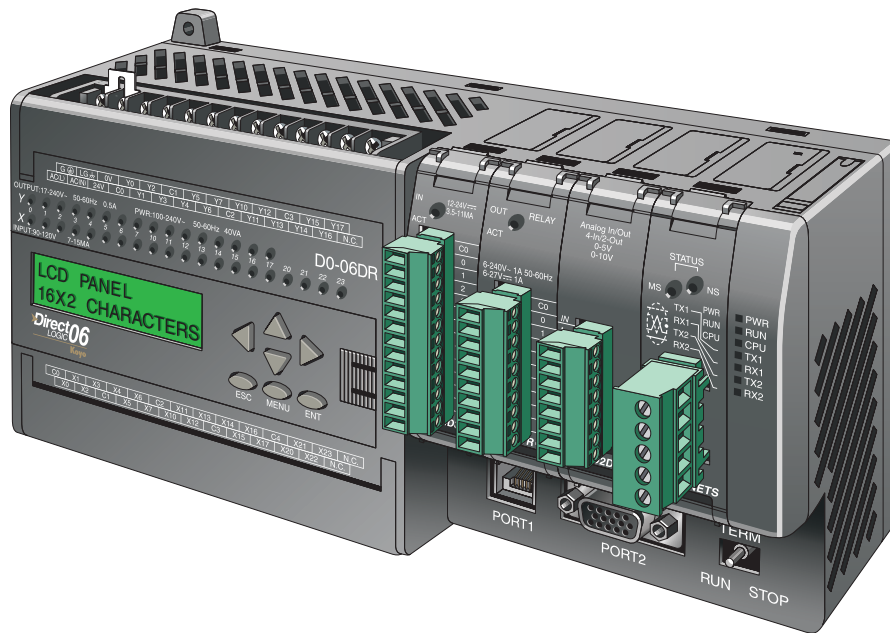


Manual del PLC DL06

Manual No.: D0-06USER-M-SP

Volumen 1 de 2



⚡ WARNING ⚡

Thank you for purchasing automation equipment from **Automationdirect.com**[®], doing business as, **AutomationDirect**. We want your new automation equipment to operate safely. Anyone who installs or uses this equipment should read this publication (and any other relevant publications) before installing or operating the equipment.

To minimize the risk of potential safety problems, you should follow all applicable local and national codes that regulate the installation and operation of your equipment. These codes vary from area to area and usually change with time. It is your responsibility to determine which codes should be followed, and to verify that the equipment, installation, and operation is in compliance with the latest revision of these codes.

At a minimum, you should follow all applicable sections of the National Fire Code, National Electrical Code, and the codes of the National Electrical Manufacturer's Association (NEMA). There may be local regulatory or government offices that can also help determine which codes and standards are necessary for safe installation and operation.

Equipment damage or serious injury to personnel can result from the failure to follow all applicable codes and standards. We do not guarantee the products described in this publication are suitable for your particular application, nor do we assume any responsibility for your product design, installation, or operation.

Our products are not fault-tolerant and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the product could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). **AutomationDirect** specifically disclaims any expressed or implied warranty of fitness for High Risk Activities.

For additional warranty and safety information, see the Terms and Conditions section of our catalog. If you have any questions concerning the installation or operation of this equipment, or if you need additional information, please call us at 770-844-4200.

This publication is based on information that was available at the time it was printed. At **AutomationDirect** we constantly strive to improve our products and services, so we reserve the right to make changes to the products and/or publications at any time without notice and without any obligation. This publication may also discuss features that may not be available in certain revisions of the product.

Trademarks

This publication may contain references to products produced and/or offered by other companies. The product and company names may be trademarked and are the sole property of their respective owners. **AutomationDirect** disclaims any proprietary interest in the marks and names of others.

**Copyright 2007, Automationdirect.com Incorporated
All Rights Reserved**

No part of this manual shall be copied, reproduced, or transmitted in any way without the prior, written consent of **Automationdirect.com Incorporated**. **AutomationDirect** retains the exclusive rights to all information included in this document.

⚡ ADVERTENCIA ⚡

Gracias por comprar equipo de automatización de Automationdirect.com™. Deseamos que su nuevo equipo de automatización opere de manera segura. Cualquier persona que instale o use este equipo debe leer esta publicación (y cualquier otra publicación pertinente) antes de instalar u operar el equipo.

Para reducir al mínimo el riesgo debido a problemas de seguridad, debe seguir todos los códigos de seguridad locales o nacionales aplicables que regulan la instalación y operación de su equipo. Estos códigos varían de área en área y usualmente cambian con el tiempo. Es su responsabilidad determinar cuales códigos deben ser seguidos y verificar que el equipo, instalación y operación estén en cumplimiento con la revisión más reciente de estos códigos.

Como mínimo, debe seguir las secciones aplicables del Código Nacional de Incendio, Código Nacional Eléctrico, y los códigos de (NEMA) la Asociación Nacional de Fabricantes Eléctricos de USA. Puede haber oficinas de normas locales o del gobierno que pueden ayudar a determinar cuales códigos y normas son necesarios para una instalación y operación segura.

Si no se siguen todos los códigos y normas aplicables, puede resultar en daños al equipo o lesiones serias a personas. No garantizamos los productos descritos en esta publicación para ser adecuados para su aplicación en particular, ni asumimos ninguna responsabilidad por el diseño de su producto, la instalación u operación.

Nuestros productos no son tolerantes a fallas y no han sido diseñados, fabricados o intencionados para uso o reventa como equipo de control en línea en ambientes peligrosos que requieren una ejecución sin fallas, tales como operación en instalaciones nucleares, sistemas de navegación aérea, o de comunicación, control de tráfico aéreo, máquinas de soporte de vida o sistemas de armamentos en las cuales la falla del producto puede resultar directamente en muerte, heridas personales, o daños físicos o ambientales severos ("Actividades de Alto Riesgo").

Automationdirect.com™ específicamente rechaza cualquier garantía ya sea expresada o implicada para actividades de alto riesgo.

Para información adicional acerca de garantía e información de seguridad, vea la sección de Términos y Condiciones de nuestro catálogo. Si tiene alguna pregunta sobre instalación u operación de este equipo, o si necesita información adicional, por favor llámenos al número 770-844-4200 en Estados Unidos.

Esta publicación está basada en la información disponible al momento de impresión. En **Automationdirect.com™** nos esforzamos constantemente para mejorar nuestros productos y servicios, así que nos reservamos el derecho de hacer cambios al producto y/o a las publicaciones en cualquier momento sin notificación y sin ninguna obligación. Esta publicación también puede discutir características que no estén disponibles en ciertas revisiones del producto.

Marcas Registradas

Esta publicación puede contener referencias a productos producidos y/u ofrecidos por otras compañías. Los nombres de las compañías y productos pueden tener marcas registradas y son propiedad única de sus respectivos dueños. Automationdirect.com™, renuncia cualquier interés propietario en las marcas y nombres de otros.

PROPIEDAD LITERARIA 2007, AUTOMATIONDIRECT.COM™ INCORPORATED
Todos los derechos reservados

No se permite copiar, reproducir, o transmitir de ninguna forma ninguna parte de este manual sin previo consentimiento por escrito de Automationdirect.com™ Incorporated. Automationdirect.com™ retiene los derechos exclusivos a toda la información incluida en este documento. Los usuarios de este equipo pueden copiar este documento solamente para instalar, configurar y mantener el equipo correspondiente. También las instituciones de enseñanza pueden usar este manual para propósitos educativos.

MANUAL DEL USUARIO DEL PLC DL06



Por favor incluya el número y la edición del manual, mostrados abajo, al comunicarse con apoyo técnico con respecto a esta publicación.

Número del Manual: D0-06USER-M-SP
Edición: Segunda Edición en español
Fecha de edición: 06/07

Historia de la publicación		
Edición	Fecha	Descripción de los cambios
1a edición en inglés	7/02	Original en inglés
Rev. A en inglés	10/02	Actualizadas algunas figuras y hechas algunas correcciones.
Rev. B en inglés	6/03	Agregados nuevos modelos de PLCs y hechas muchas correcciones.
2a edición en inglés	3/04	Agregados dos apéndices, retirados datos de módulos discretos y hechas muchas correcciones.
3a edición en inglés	12/06	Corregidas todas las tablas, muchas correcciones a los capítulos 2, 3, 4, 5, 6, y 7; El capítulo 3 (HSIO) fue movido al apéndice E y el capítulo 4 fue dividido en los capítulos 3 y 4 y se agregó el apéndice K; se agregaron las instrucciones Ibox de DirectSOFT5 al capítulo 5; ejemplo agregado de Ramp/Soad al capítulo 8; Los sistemas de numeración y las comunicaciones seriales fueron agregados como apéndice; muchas correcciones de menor importancia fueron hechas a través del manual.
1a edición en español	10/04	Traducción de la revisión 3/04 con algunas correcciones y explicaciones adicionales. Traducción hecha por Luis Miranda y Richard Pinney, ingenieros miembros del departamento de apoyo técnico de AUTOMATION DIRECT.
2a Edición en español	6/07	Se ha actualizado el manual de acuerdo a la 3a. edición en inglés con adiciones. Traducción y adiciones hechas por Luis Miranda y Hector Calcaño, ingenieros miembros del departamento de apoyo técnico de AUTOMATION DIRECT.

Notas del traductor.

1a. edición - La traducción ha sido hecha con el propósito de atender una vasta clientela que no puede leer inglés, por la experiencia en las llamadas de apoyo técnico. Se ha mantenido el formato, la paginación y los dibujos los más posible como la versión en inglés, aunque a veces ha sido necesario moverse algunas páginas por razones de espacio o necesidad de mayores explicaciones. Muchas de las palabras se han dejado en inglés, ya que es muy común usarlas en el lenguaje técnico y también, más importante, otras informaciones se encuentran solamente en inglés, tal como el programa DirectSOFT.

2a edición - Se ha agregado el capítulo 11, que no tiene un equivalente en inglés, para dar ejemplos de como ejecutar programas de un PLC basado en el concepto de un proceso. Se han seguido las modificaciones hechas en la tercera edición en inglés. Se ha aumentado el contenido del apéndice K.

VOLUMEN I:

CONTENIDO



Capítulo 1: Como comenzar

Introducción	1-2
Propósito de este manual	1-2
Manuales complementarios	1-2
Apoyo técnico	1-2
Asuntos claves en cada capítulo	1-3
Convenciones usadas	1-3
Descripción general del PLC DL06	1-4
Características del PIC DL06	1-4
Métodos de programación	1-4
Programación con <i>DirectSOFT</i> en Windows™	1-4
Programador portátil D2-HPP	1-5
Guía rápida de selección de E/S	1-5
Comienzo rápido	1-6
Pasos para diseñar un sistema	1-10
Preguntas y respuestas sobre el PLC DL06	1-12

Capítulo 2: Instalación, cableado y especificaciones

Consideraciones de seguridad	2-2
Planee con la seguridad en mente	2-2
Tres niveles de protección	2-3
Parada normal del sistema	2-3
Parada de emergencia	2-3
Aprobación de sitios peligrosos Clase 1, División 2	2-4
Explicación del panel frontal del PLC DL06	2-4
Retiro del bloque de terminales	2-5

Instrucciones generales de montaje	2-6
Dimensiones de las unidades	2-6
Gabinetes	2-6
Disposición del panel & distancias mínimas	2-7
Usando rieles de montaje DIN	2-8
Especificaciones ambientales	2-9
Aprobaciones de agencias reguladoras	2-9
Consideraciones de cableado	2-10
Protección de fusibles para la alimentación de energía	2-10
Fuente de energía externa	2-11
Planeando las rutas de cableado	2-11
Protección por fusibles de los circuitos de entradas y de salidas	2-12
Enumeración de los puntos de E/S	2-12
Estrategias de cableado del sistema	2-13
Límites de aislación del PLC	2-13
Conectando aparatos de interfaces de operación	2-14
Conectando aparatos de programación	2-14
Concepto de E/S drenadoras/ surtidoras	2-15
Concepto de terminales "comunes"	2-16
Conectando entradas y salidas a aparatos de campo de estado sólido	2-17
Sensores de entrada de estado sólido	2-17
Cargas de salidas de estado sólido	2-17
Métodos de cableado con salidas a relevador	2-19
Supresión de transientes y sobretensiones con cargas inductivas	2-20
Prolongando la vida útil de los contactos de un relevador	2-21
Métodos de cableado de entradas CC	2-22
Métodos de cableado de salidas CC	2-23
Métodos de cableado de entradas y salidas HSIO	2-24
Glosario de términos de la especificación	2-25
Diagramas eléctricos y especificaciones	2-26
Diagrama de cableado de entradas y salidas del PLC D0-06AR	2-28
Diagrama de cableado de entradas y salidas del PLC D0-06DA	2-30
Diagrama de cableado de entradas y salidas del PLC D0-06DD1	2-32
Diagrama de cableado de entradas y salidas del PLC D0-06DD2	2-34
Diagrama de cableado de entradas y salidas del PLC D0-06DR	2-36
Diagrama de cableado de entradas y salidas del PLC D0-06DD1-D	2-38

Diagrama de cableado de entradas y salidas del PLC D0-06DD2-D2-40
 Diagrama de cableado de entradas y salidas del PLC D0-06DR-D2-42

Capítulo 3: Instalación, cableado y especificaciones

Introducción3-2
 Especificaciones de la CPU3-3
 Configuración del hardware de la CPU3-4
 Diagramas de los puertos de comunicación serial3-4
 Conectando aparatos de programación3-5
 Información de configuración de la CPU3-5
 La indicación de estado del PLC3-6
 Funciones del conmutador de modo3-6
 Cambiando modos en el PLC DL063-7
 Modo de operación durante la energización del PLC3-7
 Usando una batería de respaldo3-8
 Funciones auxiliares3-9
 Borre un programa existente3-9
 Inicializando la memoria del sistema3-9
 Configuración de rangos retentivos de memoria3-10
 Usando una contraseña3-11
Operación de la CPU3-12
 El sistema operativo de la CPU3-12
 Modo Program3-13
 Modo Run3-13
 Leyendo entradas3-14
 Comunicación con aparatos periféricos y forzar E/S3-14
 Comunicación por la barra de la CPU3-15
 Actualización de la hora y fecha, relevadores especiales y memorias especiales ...3-15
 Resolviendo programas de uso3-16
 Resolviendo los algoritmos de lazos de control PID3-16
 Tiempo de respuesta de entradas y salidas3-17
 Escribiendo a las salidas3-17
 Escribiendo salidas a módulos de especialidad3-17
 Diagnósticos3-17
 Tiempo de respuesta de entradas y salidas3-17
 ¿Es importante el tiempo en su proceso?3-17
 Respuesta de entradas y salidas normal3-18

Respuesta normal máxima de entradas y salidas	3-18
Mejorando el tiempo de respuesta	3-19
Consideraciones del tiempo de barrido de la CPU	3-20
Leyendo entradas	3-20
Escribiendo a salidas	3-20
Atendiendo los aparatos periféricos	3-21
Comunicación de la barra de la CPU	3-21
Actualizando hora y fecha, relevadores especiales, memorias dedicadas	3-21
Ejecución de un programa	3-22
Sistemas numéricos del PLC	3-23
Recursos del PLC	3-23
Memoria V	3-24
Números BCD	3-24
Números hexadecimales	3-24
Mapa de memoria	3-25
Sistema de numeración octal	3-25
Direcciones discretas y de palabra	3-25
Direcciones de memoria V para las áreas de memoria discretas	3-25
Puntos de entradas (Datos tipo X)	3-26
Puntos de salidas (Datos tipo Y)	3-26
Relevadores de control (Datos tipo C)	3-26
Temporizadores y bits de estado (Datos tipo T)	3-26
Valores corrientes del temporizador (Datos del tipo V)	3-27
Contadores y bits de estado de contadores (Datos tipo CT)	3-27
Valores corrientes de un contador (Datos tipo V)	3-27
Memoria de palabra (Datos tipo V)	3-28
Las etapas (Datos tipo S)	3-28
Los relevadores especiales (Datos tipo SP)	3-28
Memoria del sistema DL06	3-39
Parámetros del sistema y de datos originales de fábrica (tipo de datos V)	3-29
Alias o apodos del PLC DL06	3-31
Mapa de memoria del DL06	3-32
Mapa de bits de entradas X o salidas Y	3-33
Mapa de bits de control del estado de etapas	3-34
Mapa de bits de relevadores de control	3-36
Mapa de bits de estado de temporizadores	3-38
Mapa de bits de estado de contadores	3-38

Mapa de bits de Remote I/O3-39

Capítulo 4: Especificaciones y operación de la CPU

Estrategias de diseño del sistema DL064-2
 Colocación de los módulos opcionales4-3
 Configuración de entradas y salidas4-4
 Consumo de corriente4-5
 Configuración de los puertos del PLC DL064-7
 Configuración ladder de los puertos del PLC DL064-9
 Comunicación con diversos protocolos4-13
 Operación de un esclavo MODBUS RTU4-14
 Operación de un maestro MODBUS RTU4-20
 Operación de un maestro MODBUS RTU con MRX y MWX4-24
 Operación con caracteres ASCII4-26

Capítulo 5: Instrucciones

Instrucciones RLL del DL065-1
Introducción5-2
Usando instrucciones booleanas5-5
 Instrucción END5-5
 Renglones simples5-5
 Contactos normalmente cerrados5-5
 Contactos en serie5-6
Salidas en el medio del renglón5-6
 Elementos en paralelo5-6
 Uniendo ramas conectadas en serie y en paralelo5-7
 Ramas en paralelo que se unen en serie5-7
 Circuitos combinación5-7
 Comparación booleana5-7
 Stack booleano5-8
 Instrucciones booleanas inmediatas5-9
Usando instrucciones booleanas5-10
Instrucciones de comparación booleanas5-26
Instrucciones de acción inmediata5-32
Instrucciones de temporizadores, contadores y Shift Register5-39

Usando temporizadores o timers	5-39
Ejemplo de uso de temporizador con los bits de estado	5-41
Ejemplo de temporizador con contactos de comparación	5-41
Ejemplo de uso de temporizador acumulador con bits de estado	5-43
Ejemplo de temporizador acumulador usando contactos de comparación	5-43
Usando contadores	5-44
Ejemplo de contador usando el bit de estado	5-46
Ejemplo de contador usando contactos de comparación	5-46
Ejemplo del contador de etapas usando el bit de estado	5-48
Ejemplo de contador de etapas usando contactos de comparación	5-48
Ejemplo de contador incremental/decremental usando el bit de estado	5-50
Ejemplo de contador UDC con contactos de comparación	5-50
Operaciones de carga y copia con el acumulador y Stack	5-52
Usando el acumulador	5-52
Copiando datos al acumulador	5-52
Cambiando los datos del acumulador	5-53
Usando el Stack del acumulador	5-54
Usando punteros	5-55
Instrucciones lógicas (Acumulador)	5-69
Instrucciones aritméticas	5-86
Instrucciones de funciones transcendentales	5-118
Instrucciones de operación con bits	5-120
Instrucciones de conversión de números (Acumulador)	5-127
Diagrama de bloque de barajada de dígitos	5-139
Instrucciones de tablas	5-141
Copie datos de un área de etiqueta de datos a la memoria V (Data Label Area)	5-143
Instrucciones de fecha y hora	5-171
Instrucciones de control de la CPU	5-173
Instrucciones de control de programa	5-175
Ejemplo de MLS/MLR	5-182
Instrucciones de interrupción	5-183
Ejemplo de programa de interrupción externa	5-185
Instrucciones de mensajes	5-186

Ejemplo de instrucción Fault	5-186
Ejemplo de data Label	5-188
Colocación directa de texto	5-197
Colocación de fecha y/o hora	5-198
Colocando datos de memorias V	5-198
Sufijos de formatos de datos para datos embutidos de memoria V	5-199
Colocación de texto desde la memoria V	5-200
Instrucciones de MODBUS RTU	5-201
Rangos de direcciones de esclavo MRX	5-202
Ejemplo de MWX	5-203
Rangos de direcciones de esclavo MWX	5-205
Rangos de direcciones de memoria del maestro MWX	5-205
Buffer de excepción response MWX	5-205
Número de Elementos MWX	5-205
Ejemplo de MWX	5-206
Instrucciones ASCII	5-207
Leyendo Texto ASCII	5-207
Escribiendo Texto ASCII	5-207
Administrando texto ASCII	5-208
Ejemplos de longitud fija de AIN	5-210
Ejemplo de longitud variable con AIN	5-214
Ejemplo de búsqueda con AFIND	5-214
Ejemplo de instrucción AFIND combinado con instrucción AEX	5-215
Ejemplo de CMPV	5-217
Ejemplo de VPRINT combinado con la instrucción PRINTV	5-222
Ejemplo de SWAPBX	5-225
Configuración de módulo Analógico Combo (ANLGCMB)IB-462	5-232
Configuración de módulo Analógico de entradas (ANLGIN)IB-460	5-234
Configuración de módulo Analógico de salidas (ANLGOUT)IB-461	5-236
Escala de 12 Bit BCD a BCD (ANSCL)IB-423	5-238
Escala de 12 Bit Binario a Binario (ANSCLB)IB-403	5-239
Filtro - BCD (FILTER) IB-422	5-240
Filtro - Binario (FILTERB) IB-402	5-242
Alarma alta/baja - BCD (HILOAL)IB-421	5-244
Alarma alta/baja - Binaria (HILOALB)IB-401	5-246
Temporizador Off Delay (OFFDTMR)IB-302	5-248
Temporizador On Delay (ONDTMR)IB-301	5-250

One Shot (ONESHOT) IB-303	5-252
Flipflop (PONOFF)IB-300	5-253
Circuito Push On / Push Off o flipflpp (PONOFF)IB-300	5-253
Move Single Word (MOVEW)IB-200	5-254
Move Double Word (MOVED)IB-201	5-255
BCD a Real con coma implicada (BCDTOR)IB-560	5-256
Doble BCD a Real con coma implicada(BCDTORD)IB-562	5-257
Función Math - BCD (MATHBCD)IB-521	5-258
Función Math - Binaria (MATHBIN)IB-501	5-260
Función Math - Real (MATHR)IB-541	5-262
Real a BCD con coma implicada y redondeo (RTOBCD)IB-561	5-263
Real to Double BCD con coma implicada y redondeo (RTOBCDD)IB-563	5-264
BCD al cuadrado (SQUARE)IB-523	5-265
Binario al cuadrado(SQUAREB)IB-503	5-266
Square Real(SQUARER)IB-543	5-267
Suma de números BCD (SUMBCD)IB-522	5-268
Suma de números Binarios (SUMBIN)IB-502	5-269
Suma de números Reales (SUMR)IB-542	5-270
Configuración de ECOM100 (ECOM100)IB-710	5-272
ECOM100 Disable DHCP (ECDHCPD)IB-736	5-274
ECOM100 Enable DHCP (ECDHCPE)IB-735	5-276
ECOM100 Query DHCP Setting (ECDHCPQ)IB-734	5-278
Envíar E-mail ECOM100 (ECEMAIL)IB-711	5-280
Configuración de Restore Default E-mail ECOM100 (ECEMRDS)IB-713	5-281
Configuración de e-mail ECOM100 (ECEMSUP)IB-712	5-286
Configuración de IP de ECOM100 (ECIPSUP)IB-717	5-290
Lea Descripción ECOM100 (ECRDDES)IB-726	5-292
ECOM100 Read Gateway Address (ECRDGWA)IB-730	5-294
ECOM100 Read IP Address (ECRDIP)IB-722	5-296
ECOM100 Read Module ID (ECRDMID)IB-720	5-298
ECOM100 Read Module Name (ECRDNAM)IB-724	5-300
ECOM100 Read Subnet Mask (ECRDSNM)IB-732	5-302
Escriba Descripción ECOM100 (ECWRDES) IB-727	5-304
ECOM100 Write Gateway Address (ECWRGWA)IB-731	5-302
ECOM100 Write IP Address (ECWRIP)IB-723	5-304
ECOM100 Write Module ID (ECWRMID)IB-721	5-310
ECOM100 Write Name (ECWRNAM)IB-725	5-312

ECOM100 Write Subnet Mask (ECWRSNM)IB-733	5-314
ECOM100 RX Network Read (ECRX)IB-740	5-316
ECOM100 WX Network Write(ECWX)IB-741	5-319
NETCFG Network Configuration (NETCFG)IB-700	5-322
Network RX Read (NETRX)IB-701	5-324
Network WX Write (NETWX)IB-702	5-327
Configuración de CTRIO (CTRIO)IB-1000	5-330
CTRIO Configuration (CTRIO)IB-1000	5-330
CTRIO Add Entry to End of Preset Table (CTRADPT)IB-1005	5-332
CTRIO Clear Preset Table (CTRCLRT)IB-1007	5-335
CTRIO Edit Preset Table Entry (CTREDPT)IB-1003	5-338
CTRIO Edit Preset Table Entry and Reload (CTREDRL)IB-1002	5-342
CTRIO Initialize Preset Table (CTRINPT)IB-1004	5-346
CTRIO Initialize Preset Table (CTRINTR)IB-1010	5-350
CTRIO Load Profile (CTRLDPR)IB-1001	5-354
CTRIO Read Error (CTRRDER)IB-1014	5-357
CTRIO Run to Limit Mode (CTRRTLTM)IB-1011	5-359
CTRIO Run to Position Mode (CTRRTPM)IB-1012	5-362
CTRIO Velocity Mode (CTRVELO)IB-1013	5-365
CTRIO Write File to ROM (CTRWFTTR)IB-1006	5-368

VOLUMEN 2:

CONTENIDO



Capítulo 6: La instrucción de tambor(DRUM)

Introducción6-2
Propósito6-2
Terminología del tambor6-2
Representación de una tabla del tambor6-3
Secuencias de salidas6-3
Transiciones de pasos6-4
Tipos de instrucción de tambor6-4
Transiciones basadas en tiempo6-4
Transiciones de evento y tiempo6-5
Transiciones de evento solamente6-6
Asignando contadores6-6
Terminación del último paso6-7
Sumario de la operación tambor (DRUM)6-8
Esquema de bloque de la instrucción de tambor6-8
Estado al energizar de la memorias de tambor6-9
Técnicas de control de tambor6-10
Las entradas de control del tambor6-10
Como hacer para que el tambor vuelva al comienzo6-11
Inicializando salidas del tambor6-11
Usando transiciones de eventos complejas6-11
La instrucción DRUM6-12
Tambor temporizado con salidas discretas (DRUM)6-12
El Tambor de eventos (EDRUM)6-14
Nemotécnicos de la instrucción DRUM con el programador D2-HPP6-16
Tambor de eventos con máscara en las salidas discretas (MDRMD)6-19
El tambor de eventos con máscara de palabra en las salidas (MDRMW)6-21

Capítulo 7: Programación por etapas RLL^{PLUS}

Introducción a la programación por etapas	7-2
Venciendo el temor de programar por etapas	7-2
Diseñando diagramas de transición de estados	7-3
Introducción a estados de procesos	7-3
Necesidad de diagramas de estado	7-3
Proceso de 2 estados	7-3
Equivalente RLL	7-4
Equivalente con etapas	7-4
Hagamos comparaciones entre RLL y RLL ^{plus}	7-5
Etapas iniciales	7-5
Qué hacen los bits de etapas	7-6
Características de la instrucción de etapas	7-6
Usando la instrucción de salto de etapas para transiciones de estados	7-7
Las instrucciones de salto JMP, SET y RESET	7-7
Ejemplo de programa de etapas: Controlador de lámpara con flip flop	7-8
Proceso de cuatro estados	7-8
Cuatro pasos para escribir un programa por etapas	7-9
1. Escriba una descripción de la operación de la aplicación	7-9
2. Dibuje un diagrama de bloques.	7-9
3. Dibuje el diagrama de transición de estados	7-9
4. Escriba el programa por etapas	7-9
Ejemplo de programa de etapas: Abridor de un portón de garaje	7-10
Ejemplo del control para abrir un portón de garaje	7-10
Dibuje el diagrama de bloques	7-10
Dibuje el diagrama de estados	7-11
Agregue una lámpara para iluminar el garaje por un corto tiempo	7-12
Modifique los diagramas de bloque y de estado	7-12
Usando un temporizador dentro de una etapa	7-13
Agregue una parada de emergencia	7-14
Transiciones exclusivas	7-14
Consideraciones de diseño del programa de etapas	7-15
Organización del programa	7-15
Cómo trabajan las instrucciones dentro de una etapa	7-16

Contador de etapas	7-17
La técnica de transición del flujo del poder	7-18
La pantalla Stage View en <i>DirectSOFT</i>	7-18
Conceptos de procesamiento paralelo	7-19
Procesos paralelos	7-19
Procesos de convergencia	7-19
Etapas de convergencia (CV)	7-19
Salto de convergencia (CVJMP)	7-20
Reglas de uso de la etapa de convergencia	7-20
Instrucciones de RLL^{PLUS} (Etapas)	7-21
Etapa (o Stage) (SG)	7-21
Etapa inicial (ISG)	7-22
El salto o JUMP (JMP)	7-22
No Salto (NJMP)	7-22
Etapa de convergencia (CV) y el salto de convergencia (CVJMP)	7-23
Llamada de bloque (BCALL)	7-25
BLOCK (BLK)	7-25
El fin de bloque (BEND)	7-25
Preguntas y respuestas acerca de la programación por etapas	7-27

Capítulo 8: Operación del control PID

Funciones de control PID en el PLC DL06	8-2
Características principales	8-2
¿Que es control PID?	8-4
Introducción al control PID con el PLC DL06	8-6
Tabla y número de lazos	8-6
Indicaciones de error de PID	8-6
Definiciones de procesos de control	8-8
Operación del control PID	8-9
Protección de Reset Windup	8-10
"Congelar" la tendencia (Bias freeze)	8-11
Ajustando el Bias	8-11
Eliminación de la acción proporcional, integral o derivativa	8-12
Modo de velocidad del algoritmo PID	8-12
Descripciones de bits de la palabra configuración de modo PID 1 (Addr + 00)	8-9

Descripciones de bits de la palabra configuración de modo PID 2 (Addr + 01) . . .	8-10
Palabra de supervisión de modo/alarmas (Addr + 06)	8-11
Transferencias sin saltos (Bumpless)	8-13
Modos de funcionamiento del control PID	8-14
Lazos de acción directa e inversa	8-14
Diez pasos para obtener un buen control de proceso	8-16
Configuración de lazos PID en el PLC DL06	8-18
Estableciendo el tamaño de la tabla de lazos y localización	8-20
Definiciones de palabras de la tabla de lazos	8-22
Descripciones de bits de la palabra configuración de modo PID 1:	8-23
Descripciones de bits de la palabra de configuración PID 2	8-24
Palabra de supervisión de modo/alarmas (Addr + 06)	8-25
Tabla de indicaciones de Ramp/Soak (Rampa y valor constante) (Addr + 33) . . .	8-25
Localización de la tabla Ramp/Soak (Rampa y valor constante) (Addr + 34)	8-26
Indicaciones de error de programación en tabla Ramp/Soak (Addr + 35)	8-26
Configure el lazo de control PID	8-27
Entre el período de muestreo	8-27
Formatos de datos de parámetros de lazos	8-28
Escogiendo formato unipolar o bipolar	8-28
Compensación de desvíos de datos	8-29
Límites del valor de referencia (SP)	8-29
Dirección remota del valor de referencia (SP)	8-29
Configuración de la variable de proceso (PV)	8-29
Configuración de la salida de control (CV)	8-33
Entradas de parámetros de control PID	8-34
Limitación de la ganancia derivativa	8-35
Selección de modo de errores	8-35
"Congelar" la tendencia (Bias freeze)	8-36
Configurando las alarmas del control PID	8-36
Supervisión de alarmas de límite	8-37
Alarmas de desvío	8-38
Alarma de tasa de cambio	8-39
Histéresis de alarmas de PV	8-39
Error de programación de alarmas	8-40
Error de cálculo Overflow o underflow	8-40
Ramp/Soak	8-41
Complete la configuración del control del lazo PID	8-41

Sintonizando lazos de control PID	8-42
Prueba de lazo abierto	8-42
Procedimiento de sintonía manual	8-44
Procedimientos alternativos de sintonía manual	8-46
Procedimiento de sintonía automática	8-47
Sintonía automática de lazo abierto	8-48
Sintonía automática de lazo cerrado	8-49
Usando PID View	8-51
Usando las funciones especiales del control PID	8-52
Como cambiar modos del lazo desde un programa ladder	8-55
Control de los modos PID con panel de operador	8-56
Efecto de los modos del PLC sobre los modos del control PID	8-56
Condiciones de cambio del modo del controlador PID	8-56
Filtro de la variable de proceso PV	8-57
El filtro analógico incorporado en el PLC DL06	8-57
Creando un filtro analógico en lógica ladder	8-58
Usando Ibox del filtro con <i>DirectSOFT5</i>	8-59
Ejemplo de FilterBV	8-59
Generador de Ramp/Soak	8-60
Introducción a Ramp/Soak	8-60
La tabla Ramp/Soak	8-61
Los indicadores de bit de la tabla Ramp/Soak	8-63
Activando el generador de Ramp/Soak	8-63
Controles de la Ramp/Soak	8-63
Supervisión del perfil de Ramp/Soak	8-64
Errores de programación de Ramp/Soak	8-64
Probando el perfil de Ramp/Soak	8-64
Ejemplo de Ramp/Soak con <i>DirectSOFT</i>	8-65
Configure el perfil Ramp/Soak con el diálogo de PID setup	8-65
Programando el control Ramp/Soak en lógica ladder	8-65
Probando el perfil de ejemplo de Ramp/Soak	8-66
Control en cascada	8-67
Introducción a lazos en cascada	8-67
Los lazos en cascada en el PLC DL06	8-68
Sintonizando lazos en cascada	8-69

Control de pulsos con amplitud modulada (PWM)	8-70
Ejemplo de programa de control PWM	8-71
El control feedforward	8-72
Ejemplo de feedforward	8-73
Ejemplo de programa ladder con control PID	8-74
Programa ladder asociado a un lazo del control PID	8-74
Consejos para búsqueda de fallas en PID	8-77
Glosario de la terminología de lazos de control PID	8-79
Bibliografía	8-81

Capítulo 9: Mantenimiento y búsqueda de problemas

Mantenimiento del sistema de hardware	9-2
Mantenimiento normal	9-2
Diagnósticos	9-2
Errores fatales	9-2
Errores no fatales	9-2
Direcciones de memoria de códigos de error	9-3
Relevadores especiales (SP) que corresponden a códigos de error	9-3
Códigos de error del PLC DL06	9-4
Códigos de errores del programa	9-5
Indicadores LED de la CPU	9-6
Indicador PWR	9-6
Indicador RUN	9-7
Indicador CPU	9-7
Problemas de comunicaciones	9-7
Búsqueda de problemas de entradas y salidas	9-8
Causas posibles	9-8
Algunos pasos de verificación rápidos	9-8
Secuencia de teclado en el programador usada para probar un punto de salida ..	9-9
Búsqueda de fallas de ruido	9-10
Problemas eléctricos de ruido electromagnético	9-10
Reducción del ruido eléctrico	9-10
Partida de una máquina y búsquedas de fallas del programa	9-11
Verificación de sintaxis	9-11

Instrucciones especiales	9-12
Verificación de referencias duplicadas	9-13
Modificaciones del programa durante el modo RUN	9-14
Ejemplo de cambio durante modo RUN	9-15
Forzando puntos de entradas y salidas	9-16
Forzar normal con acceso directo	9-18
Forzar con el bit override	9-19
Indicadores de bit override	9-19
Contactando apoyo técnico en AUTOMATION DIRECT	9-20

Capítulo 10: Panel con visor LCD

Introducción al visor LCD del DL06	10-2
Teclado del visor LCD	10-2
Instalación de encaje rápido	10-3
Prioridad de la pantalla	10-4
Navegación por el menú	10-5
Confirme PLC, nivel de versión de firmware, uso de memoria, etc.	10-6
Examinando lo que tiene en las ranuras del PLC	10-8
Menú 2, M2:SYSTEM CFG.	10-8
Menú 3, M3:MONITOR	10-10
Supervisión y cambio de valores	10-10
Supervisión de datos	10-10
Valores de memoria V	10-10
Valores del puntero	10-12
Supervisión de bits	10-13
Estado de bits	10-13
Cambiando fecha y hora	10-14
Menú 4, M4 : CALENDAR R/W	10-14
Definiendo la contraseña y bloqueando cambios	10-17
Menú 5, M5 : PASSWORD R/W	10-17
Historia de errores	10-20
Menú 6, M6 : ERR HISTORY	10-20

Teclas de prueba, Luz trasera y el zumbador	10-21
Menú 7, M7 : LCD TEST&SET	10-21
Información de memoria del PLC para el visor LCD	10-22
Sufijos de formato de datos para datos de memoria V mostrados en el visor ...	10-22
Memoria reservada para el panel de visor de LCD	10-23
Definiciones de los bits de V7742	10-24
Cambiando los mensajes originales de la pantalla	10-25
Programa ejemplo para colocar el mensaje de pantalla por defecto	10-25
Instrucción del panel con visor LCD del PLC DL06 (LCD)	10-26
Fuente del mensaje	10-26
Códigos de caracteres ASCII	10-27
Programa ejemplo: Alarma mostrada con la fecha y hora que ocurrió	10-28
Programa ejemplo: alarma con datos embutidos de la memoria V	10-29
Programa ejemplo: Texto de alarma con datos embutidos de la memoria V ...	10-30

Capítulo 11: Recopilación y ejemplos

Introducción al capítulo 11	10-2
Que es memoria V	11-2
Alimentación de energía	11-4
Familiarización con <i>DirectSOFT</i>	11-4
Conexión al PLC desde una computadora con <i>DirectSOFT</i>	11-5
Crear o modificar un programa	11-6
Ejercicios para el nuevo programador	11-13
Resumen de las particularidades de los PLCs <i>DirectLOGIC</i>	11-15
Programación de entradas y salidas análogas con PLCs DL	11-15
Una palabra sobre números negativos	11-16
Una palabra sobre programación de PID	11-17
El PLC DL06 puede tener entradas y salidas remotas	11-17
Concepto de una instalación controlada por PLC's	11-18
Ejemplos de programa ladder con PLC DL06	11-21
Ejemplo 1: Triturador de minerales con E/S discretas	11-21
Ejemplo 2: Uso de contadores	11-50
Ejemplo 3: Uso de comparación	11-50
Ejemplo 4: Uso de una interfase de operador	11-51
Ejemplo 5: Uso de E/S análogas	11-53

Ejemplo 6: Uso de comunicaciones seriales11-55
 Ejemplo 7: Uso de lazo de control PID11-58

Apéndice A: Funciones auxiliares

IntroducciónA-2
 Propósito de las funciones auxiliaresA-2
 Acceso a funciones AUX con *DirectSOFT*A-3
 Acceso a funciones AUX con el programador portátilA-3

AUX 2* — Operaciones de programa ladder RLLA-4
 AUX 21 Verifique ProgramaA-4
 AUX 22 Cambie ReferenciaA-4
 AUX 23 Suprimir partes del programaA-4
 AUX 24 Borre el programaA-4

AUX 3* — Operaciones de memoria VA-4
 AUX 31 Limpia la memoria V completaA-4

AUX 4* — Configuración de entradas y salidasA-4
 AUX 41 Muestra la configuración de E/SA-4

AUX 5* — Configuración de la CPUA-5
 AUX 51 Modifique el nombre del programaA-5
 AUX 53 Tiempo de barrido del PLCA-5
 AUX 54 Initialize ScratchpadA-5
 AUX 55 Configure el watchdog TimerA-5
 AUX 56 CPU Network AddressA-5
 AUX 57 Configure rangos retentivosA-6
 AUX 58 Pruebe las operacionesA-6
 AUX 59 Bit OverrideA-6
 AUX 5B Configuración HSIOA-7
 AUX 5D Seleccione el modo de barrido del PLCA-7

AUX 6* — Configuración del programador D2-HPPA-8
 AUX 61 Muestre el no. de versiónA-8
 AUX 62 Zumbador activado o desactivadoA-8
 AUX 65 DiagnósticosA-8

AUX 7* — Operaciones con EEPROMA-8
 Areas de memoria transferiblesA-8
 AUX 71 CPU al EEPROM del programadorA-8

AUX 72 EEPROM del programador a la CPU	A-9
AUX 73 Compare EEPROM a la CPU	A-9
AUX 74 Verifique que el EEPROM está vacío	A-9
AUX 75 Borre el EEPROM	A-9
AUX 76 Muestre el tipo de EEPROM	A-9
AUX 8* — Operaciones de contraseña	A-9
AUX 81 Modifique la contraseña	A-9
AUX 82 Destrabe la CPU	A-10
AUX 83 Trabe la CPU	A-10

Apéndice B: Códigos de error del PLC DL06

Códigos de error del PLC DL06	B-2
-------------------------------------	-----

Apéndice C: Tiempos de ejecución de cada instrucción

Introducción	C-2
Memorias de datos	C-2
Memorias de bits	C-2
Como leer las tablas	C-2
Tiempos de ejecución de instrucciones	C-3
Instrucciones booleanas	C-3
Instrucciones booleanas de comparación	C-4
Instrucciones booleanas bit of word	C-13
Instrucciones inmediatas	C-14
Temporizador, Contador y Shift Register	C-14
Instrucciones de datos de acumulador	C-16
Instrucciones lógicas	C-17
Instrucciones aritméticas	C-19
Instrucciones diferenciales	C-22
Instrucciones de bits	C-22
Instrucciones de conversión de números	C-23
Instrucciones de tablas de memorias	C-23
Instrucciones de control de la CPU	C-25
Instrucciones de control del programa	C-25
Instrucciones de interrupción	C-25
Instrucciones de red de comunicación	C-25
Instrucciones de entradas y salidas inteligentes (RX/WX)	C-26

Instrucciones de mensajes	C-26
Instrucciones RLL <i>plus</i>	C-26
Instrucciones Drum	C-26
Instrucciones de fecha y hora	C-27
Instrucciones MODBUS	C-27
Instrucciones ASCII	C-27

Apéndice D: Relevadores especiales

Relevadores especiales del PLC DL06	D-2
Relevadores de partida y de pulsos de tiempo definido	D-2
Relevadores de estado de la CPU	D-2
Supervisión del sistema	D-3
Estado del acumulador	D-3
Estado de entradas HSIO	D-4
Relevador especial de salida de pulsos HSIO	D-4
Relevador de supervisión de comunicación	D-4
Relevador de supervisión de comunicaciones de las ranuras del PLC	D-4
Relevadores especiales para módulos opcionales	D-4
Relevadores especiales de detección del contador 1 en el Modo 10	D-5
Relevadores especiales de detección del contador 2 en el Modo 10	D-6

Apéndice E: Entradas de alta velocidad y salidas de tren de pulsos

Introducción	E-2
Soluciones de control de movimiento embutidas	E-2
Disponibilidad de las características HSIO	E-2
Circuitos dedicados de entradas y salidas de alta velocidad	E-3
Diagramas de cableado para cada modo HSIO	E-3
Selección de modo de operación HSIO	E-4
Entendiendo los seis modos	E-4
Modo normal de fábrica	E-5
Configuración del modo HSIO	E-6
Configuración de las entradas X0 – X3	E-6
Modo 10: Contador	E-7
Propósito	E-7
Diagrama de bloques funcional	E-7
Diagrama de cableado	E-8

Interconexión a las entradas de conteo	E-8
Configuración del modo 10	E-9
Relevadores especiales y valores prefijados	E-9
Valores prefijados absolutos e incrementales	E-10
Dirección inicial de los datos de valores prefijados	E-11
Usando menos de 24 valores prefijados	E-11
Relevadores especiales de comparación	E-12
Cálculo de valores prefijados	E-13
Configuración de las entradas X	E-14
Escribiendo el programa de control	E-15
Ejemplo 1: Contador sin valor prefijado	E-16
Ejemplo 2: Contador con valores prefijados	E-18
Ejemplo 3: Contador con valor inicial	E-21
Búsqueda de problemas del modo 10	E-23
Síntoma: El contador no cuenta	E-23
Síntoma: El contador cuenta pero los valores prefijados no funcionan	E-23
Síntoma: El contador cuenta pero no vuelve a cero	E-23
Modo 20: Contador incremental/decremental	E-24
Propósito del modo 20	E-24
Diagrama de bloques funcional	E-24
Señal en cuadratura del encoder	E-25
Conexión de salidas de un encoder	E-25
Configuración del modo 20	E-27
Relevadores especiales y valores prefijados del modo 20	E-27
Configuración de las entradas X	E-28
Contador incremental/decremental de modo 20	E-28
Escribiendo el programa de control del modo 20	E-29
Ejemplo 1: Contador en cuadratura con interrupción	E-30
Ejemplo 2: Contador incremental y decremental con entradas normales	E-32
Ejemplo 3: Contadores en cuadratura	E-34
Búsqueda de problemas en el modo 20	E-37
Síntoma: El contador no cuenta	E-37
Síntoma: El contador cuenta en la dirección errada	E-37
Síntoma: El contador cuenta hacia arriba y abajo pero no hace reset	E-37
Modo 30: Salidas de tren de pulsos	E-38
Propósito del modo 30	E-38

Diagrama de bloques funcional	E-39
Diagrama de cableado de modo 30	E-40
Conexión de entradas de un accionamiento	E-40
Especificaciones del perfil de movimiento	E-41
Configuración de las entradas y salidas físicas	E-41
Funciones lógicas de entradas y salidas	E-41
Configuración del modo 30	E-43
Memoria de selección de velocidad o de perfil	E-43
Tabla de parámetros del perfil	E-43
Perfil trapezoidal automático	E-43
Perfil trapezoidal paso a paso	E-44
Selección de tipo de perfil	E-45
Definición del perfil trapezoidal automático	E-45
Definición del perfil trapezoidal paso a paso	E-46
Definición del control de velocidad	E-46
Operación de perfil trapezoidal automático	E-47
Ejemplo 1: Perfil trapezoidal automático sin interrupción externa	E-48
Cambie el valor de posición en cualquier momento	E-49
Ejemplo 2: Perfil trapezoidal automático con interrupción externa	E-50
Ejemplo 3: Perfil trapezoidal automático con búsqueda de "Home"	E-53
Operación del perfil trapezoidal paso a paso	E-55
Ejemplo 4: Perfil trapezoidal paso a paso	E-56
Operación del perfil de velocidad	E-59
Programa Ejemplo 5: Perfil de velocidad	E-60
Códigos de error del perfil trapezoidal automático	E-62
Búsqueda de problemas para el Modo 30	E-62
Síntoma: El motor de paso a paso no gira	E-62
Síntoma: El motor gira en la dirección equivocada	E-63
Modo 40: Interrupciones de alta velocidad	E-64
Propósito del modo 40	E-64
Esquema de bloques funcional	E-64
Configuración del modo 40	E-65
Las interrupciones y el programa ladder	E-65
Parámetros de sincronización de interrupción externa	E-66
Parámetros de interrupción por tiempo	E-66
Configuración INT temporizada / entrada X	E-66
Ejemplo 1 modo 40: Interrupción externa	E-67

Ejemplo 2 modo 40: Interrupción por tiempo	E-68
Modo 50: Entrada de captura de pulso	E-69
Propósito de modo 50	E-69
Diagrama de bloque funcional	E-69
Parámetros de sincronización de captura de pulso	E-69
Cuando usar modo de captura de pulso	3-70
Configuración del modo 50	E-70
Configuración de las entradas X	E-71
Ejemplo 1 modo 50 : Captura de pulso	E-72
Modo 60: Entradas discretas con filtro	E-73
Propósito de modo 60	E-73
Diagrama de bloques funcional	E-73
Parámetros de constante de tiempo del filtro de entrada	E-73
Configuración del modo 60	E-74
Configuración de las entradas X	E-74
Ejemplo modo 60 : Entradas filtradas	E-75
Apéndice F: Memorias del PLC	
Memorias del PLC DL06	F-2
Apéndice G: Tabla ASCII	
Tabla de conversión ASCII	G-2
Apéndice H: Pesos de los componentes	
Tablas de pesos de los productos	H-2
Apéndice I: Sistemas numéricos	
Introducción a sistemas numéricos	J-2
Sistema decimal	I-2
Sistema numérico octal	I-2
Memorias de datos y memorias para configuración del PLC	I-3
Sistema numérico binario	I-3
Sistema numérico BCD (Binary coded decimal)	I-4
Sistema numérico hexadecimal	I-5
Sistema numérico real de punto flotante	I-5

¿Cual es la convención con IEEE para punto flotante de 32 bits?	I-6
Rangos de números de punto flotante	I-7
Número en representación Gray	I-7
Valores numéricos en módulos análogos	I-8
Representación del complemento de 2	I-9
El cálculo del complemento de 2	I-10

Apéndice J: Directivas de la Unión Europea (CE)

Directivas de la Unión Europea (EU)	J-2
Países miembros	J-2
Directivas aplicables	J-2
Cumplimiento	J-2
Seguridad general	J-3
Manual de instalación especial	J-4
Otras fuentes de información	J-4
Pautas básicas de instalación de EMC	J-4
Gabinetes	J-4
Filtros de alimentación en CA	J-5
Supresores y fusibles	J-5
Aterramiento interno del gabinete	J-5
Aterramiento equipotencial	J-6
Comunicaciones y cables blindados	J-6
Cables de señales análogas y RS232	J-7
Cables multinodo	J-7
Cables blindados dentro de gabinetes	J-7
Aislamiento de una red de comunicación	J-7
Versiones alimentadas por corriente continua	J-8
Asuntos específicos al PLC DL06	J-9

Apéndice K: Introducción a comunicaciones seriales

Introducción a comunicaciones seriales	K-2
Estándar de cableado	K-2
Protocolos de comunicaciones disponibles	K-3
Concepto de transmisión de datos seriales	K-5
Comunicación con K-sequence	K-7
Comunicación con <i>DirectNET</i>	K-7
Operación de un esclavo en una red <i>DirectNET</i>	K-12

Contenido

Operación de un maestro en una red <i>DirectNET</i>K-12
Configuración del puerto como <i>DirectNET</i>K-16
Ejemplo para redes <i>DirectNET</i>K-21
Comunicación con MODBUS RTUK-29
Introducción a la comunicación MODBUS RTUK-29
Convención de direcciones en MODBUSK-32
Operación de un esclavo en una red MODBUS RTUK-35
Operación de un maestro en una red MODBUS RTUK-41
Configuración del puerto como MODBUS RTUK-45
Operación como maestro usando MRX y MWXK-50
Ejemplos para redes MODBUS RTUK-51
Respuesta de excepción en MODBUSK-52
Comunicación con ASCII non sequenceK-51
Ejemplo para Comunicación ASCIIK-59

Indice

COMO COMENZAR



En este capítulo...

Introducción	1-2
Convenciones usadas	1-3
Descripción general del PLC DL06	1-4
Métodos de programación	1-4
Guía rápida de selección de entradas y salidas	1-5
Comienzo rápido	1-6
Pasos para diseñar un sistema	1-10
Preguntas y respuestas sobre el PLC DL06	1-12

Introducción

Propósito de este manual

Gracias por comprar un PLC DL06. Este manual le muestra cómo instalar, programar y mantener todos los PLCs en la familia DL06. También le ayuda a entender cómo interconectarlos a otros dispositivos en un sistema de control. Este manual contiene información importante para el personal que instalará PLCs DL06 y para el programador de este mismo. Este manual le proporcionará información necesaria para mantener su sistema en operación. Es una versión casi idéntica a la versión original en inglés en la tercera Edición, con algunas explicaciones adicionales cuando sea necesario y la adición del capítulo 11.

Manuales adicionales

El manual D0-OPTIONS-M-SP contiene información técnica sobre los módulos opcionales disponibles para los PLCs DL06. Esta información incluye especificaciones y diagramas eléctricos que serán imprescindibles si usted usa cualquiera de los módulos opcionales de entradas o salidas o de comunicaciones. Si usted ha comprado uno de nuestros paneles de interface de operador o software de programación *DirectSOFT™*, usted debe referirse a los manuales de estos productos. Si necesita saber más de comunicación Ethernet, use el manual HX-ECOM-M. Si quiere usar un módulo H0-CTRIO, use el manual Hx-CTRIO-M. Si quiere saber más del módulo de comunicación D0-DCM, use el manual D0-DCM-MSP.

Apoyo técnico

Nos esforzamos en hacer nuestros manuales los mejores de la industria. Confiamos en sus comentarios sobre esta información para hacernos saber si estamos alcanzando nuestra meta. Si usted no puede encontrar una solución a su uso particular o si por cualquier razón usted necesita apoyo técnico, llámenos por favor al teléfono en E.E.U.U.:

770-844-4200.

Nuestro grupo de apoyo técnico trabajará con usted para contestar a sus preguntas. Este grupo está disponible de Lunes a Viernes a partir de la 9:00 de la mañana hora de Nueva York hasta las 6:00 P.M. También le pedimos que visite nuestro sitio de Internet en donde usted puede encontrar información técnica y no técnica sobre nuestros productos y nuestra compañía.

<http://www.automationdirect.com>

Si usted tiene un comentario, una pregunta o una sugerencia sobre cualesquiera de nuestros productos, servicios, o manuales, complete, por favor, la hoja de comentarios que está incluida con este manual al final de la publicación y envíela de vuelta por fax o e-mail.

Convenciones usadas



*Cuando usted vea el icono de "libreta" en el margen izquierdo, el párrafo a la derecha inmediata será una nota especial. Las notas representan información que puede hacer su trabajo más rápido o más eficiente. La palabra **NOTA**: en negrita marcará el principio del texto.*



Cuando usted vea "el icono del punto de exclamación" en el margen izquierdo, el párrafo a la derecha inmediata será una advertencia. Esta información podría prevenir lesión, pérdida de propiedad o aún la muerte en casos extremos. Cualquier advertencia en este manual se debe mirar como información crítica que se deba leer en su totalidad. La palabra **ADVERTENCIA** en negrita marcará el principio del texto.

Asuntos claves para cada capítulo.

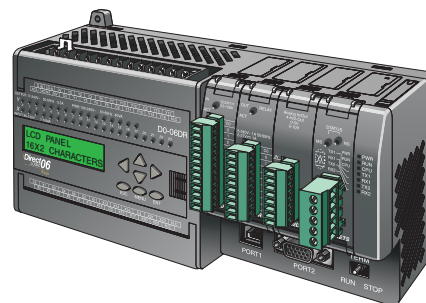
El principio de cada capítulo enumerará los asuntos claves que se pueden encontrar en ese capítulo.



Generalidades del PLC DL06

1

La familia de PLCs DL06 es una línea de productos muy versátil que combina características poderosas en un tamaño compacto. Este PLC ofrece módulos de expansión de entradas y salidas discretas y análogas, contadores de alta velocidad, matemática del punto flotante, controladores PID, programación de secuenciador de tambor, varias opciones de comunicaciones seriales y con Ethernet y un visor LCD opcional.



Características del PLC DL06

Hay nueve versiones de este PLC. Todos tienen el mismo tamaño y en general las mismas funciones de CPU. Este PLC ofrece un conjunto de instrucciones muy similares a la poderosa CPU D2-260 de la familia *DirectLogic* incluyendo instrucciones MODBUS y ASCII. Todos los PLCs DL06 tienen dos puertos de comunicación seriales que pueden ser usados para programación, para interfaces de operador, para redes, etc.

Las unidades con entradas de corriente continua tienen características de alta velocidad en cuatro puntos de entrada. Las unidades con salidas de corriente continua tienen pulsos de salidas de alta velocidad en diferentes formatos en el primero y segundo punto de salida. Más detalles están en el capítulo 4.

Vea las tablas a continuación para ver el modelo que mejor se adapta a su aplicación.

Familia del PLC DL06					
No. de artículo del DL06	Tipo de entrada discreta	Tipo de salidas discretas	Alimentación	Entradas de alta velocidad	Salidas de tren de pulsos
DO-06AA	CA	CA	95-240 VCA	No	No
DO-06AR	CA	Relevador	95-240 VCA	No	No
DO-06DA	CC	CA	95-240 VCA	Yes	No
DO-06DD1	CC	Drenadoras	95-240 VCA	Yes	Yes
DO-06DD2	CC	Surtidoras	95-240 VCA	Yes	Yes
DO-06DR	CC	Relevador	95-240 VCA	Yes	No
DO-06DD1-D	CC	Drenadoras	12-24 VCC	Yes	Yes
DO-06DD2-D	CC	Surtidoras	12-24 VCC	Yes	Yes
DO-06DR-D	CC	Relevador	12-24 VCC	Yes	No

Métodos de programación

Hay disponibles dos métodos de programación: RLL (lógica de relevadores o en inglés Relay Ladder Logic) y RLLPLUS. La programación RLLPLUS combina una característica de diagrama de flujo de programación (etapas) con el lenguaje normal RLL. El paquete de programación de *DirectSOFT™* y el programador portátil pueden programar los dos métodos.

Programando con *DirectSOFT* en Windows™

El PLC DL06 puede ser programado con la versión *DirectSOFT*, versión 4.0 o más nueva, y este es un paquete de programación basado en el sistema operativo Windows usado en PC's, lo que permite usar todas las familiares características de este programa.

DirectSOFT5 (Artículo PC-DSOFT5) programa todas las familias de PLCs *DirectLOGIC*. Se puede utilizar la versión completa de *DirectSOFT* para programar el DL05, el DL06, el DL105, el DL205, el DL305, y el DL405. (Puede ser que se necesiten actualizaciones del software para nuevos PLCs si llegan a ser disponibles). Un manual separado describe el software de programación *DirectSOFT*. Es necesaria la versión 4.0 de *DirectSOFT* o más nueva para programar el PLC DL06.

Programador portátil

Todos los PLCs DL06 tienen un puerto de programación incorporado para el uso con el programador portátil (D2-HPP), el mismo programador usado con el DL05, DL105 y las familias DL205. El programador portátil se puede utilizar para crear, modificar y eliminar errores del programa de uso. Un manual separado describe el programador portátil D2-HPP. Solamente D2-HPPs con la versión de firmware 2.0 o mejor puede programar el PLC DL06.

Guía rápida de selección de entradas y salidas

Las nueve versiones del DL06 tienen circuitos de entradas y salidas que se pueden conectar a una amplia variedad de dispositivos. En varios casos un circuito particular de entrada o de salida puede conectarse a voltajes de C.C. o C.A., o a drenadoras y surtidoras. Verifique esta tabla para encontrar el PLC DL06 adecuado a la interfase de los dispositivos en su uso.

Selección de un PLC por el tipo de entradas y salidas						
No. de artículo del DL06	ENTRADAS			SALIDAS		
	Tipo de E/S comunes	Drenadoras/ Surtidoras	Rangos de voltaje	Tipo de E/S comunes	Drenadoras/ Surtidoras	Voltaje/ Corriente Nominal*
D0-06AA	CA / 5	–	90 – 120 VCA	CA / 4	–	17 – 240 VCA, 50/60 Hz 0.5A
D0-06AR	CA / 5	–	90 – 120 VCA	Relevador / 4	Drenadoras o Surtidoras	6 – 27VDC, 2A 6 – 240 VCA, 2A
D0-06DA	CC / 5	Drenadoras or Source	12 – 24 VCC	CA / 4	–	17 – 240 VCA, 50/60 Hz 0.5A
D0-06DD1	CC / 5	Drenadoras or Source	12 – 24 VCC	CC / 4	Drenadoras	6 – 27 VDC, 0.5A (Y0–Y1) 6 – 27 VCC, 1.0A (Y2–Y17)
D0-06DD2	CC / 5	Drenadoras or Source	12 – 24 VCC	CC / 4	Surtidoras	12 – 24 VCC, 0.5A (Y0–Y1) 12 – 24 VCC, 1.0A (Y2–Y17)
D0-06DR	CC / 5	Drenadoras or Source	12 – 24 VCC	Relevador / 4	Drenadoras o Surtidoras	6 – 27VCC, 2A 6 – 240 VCA, 2A
D0-06DD1-D	CC / 5	Drenadoras or Source	12 – 24 VCC	CC / 4	Drenadoras	6 – 27 VCC, 0.5A (Y0–Y1) 6 – 27 VCC, 1.0A (Y2–Y17)
D0-06DD2-D	CC / 5	Drenadoras or Source	12 – 24 VCC	CC / 4	Surtidoras	12 – 24 VCC, 0.5A (Y0–Y1) 12 – 24 VCC, 1.0A (Y2–Y17)
D0-06DR-D	CC / 5	Drenadoras or Source	12 – 24 VCC	Relevador / 4	Drenadoras o Surtidoras	6 – 27 VCC, 2A 6 – 240 VCA, 2A

* Vea el capítulo 2, especificaciones para más información de cada modelo de PLC DL06

Partida rápida

Este ejemplo no le va a decir todo lo que usted necesita saber sobre programación y como comenzar un sistema de control complejo. Se trata solamente de darle una oportunidad de demostrarle a Ud. y a otros los pasos básicos necesarios para activar el PLC y para confirmar su operación. Vea por favor las advertencias y las notas en este manual para tener informaciones importantes que usted no debe pasar por alto.

Paso 1: Desembalar el equipo del PLC DL06

Desempaque el DL06 y recolecte las partes necesarias para construir este sistema de demostración. Los componentes recomendados son:

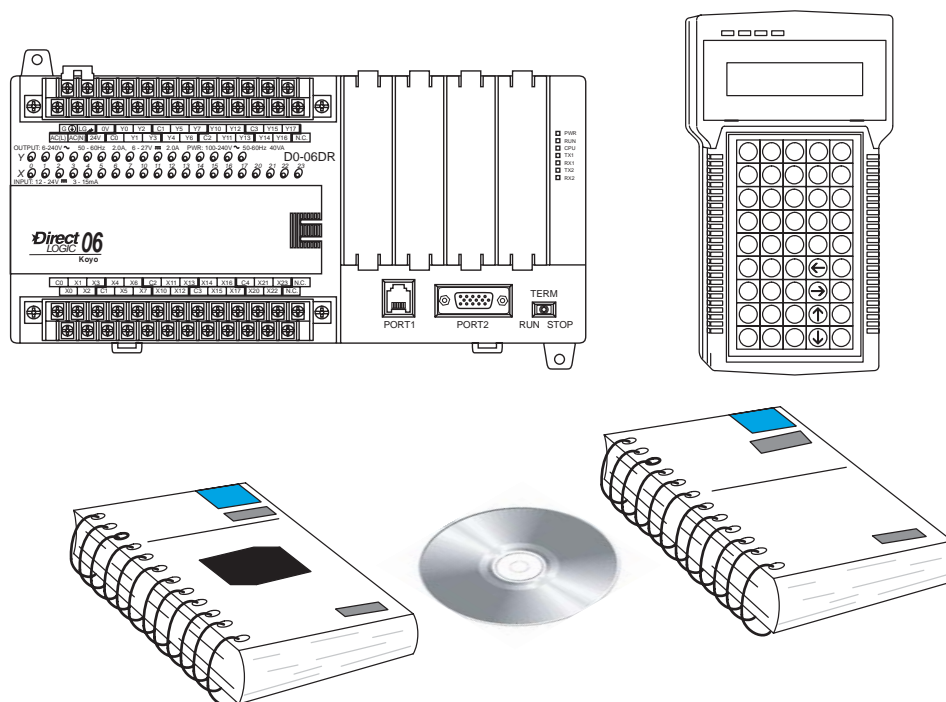
- PLC DL06
- Cable eléctrico o fuente de poder de corriente continua.
- Conmutadores de palanca (vea el paso 2 en la página siguiente).
- Cable para conexión de circuitos, sección 16 a 22 AWG
- Manual del usuario DL06 (este manual)
- Un destornillador pequeño, plano de 5/8" o tipo Philips # 1

Usted necesitará por lo menos una de las opciones de programación siguientes:

- Programa *DirectSOFT* V4.0 o mejor (PC-D5OFT5), Manual *DirectSOFT* (incluido con el software), y un cable de programación (D2-DSCBL que conecta el PLC DL06 con una computadora personal) o la versión gratuita PC-DS100

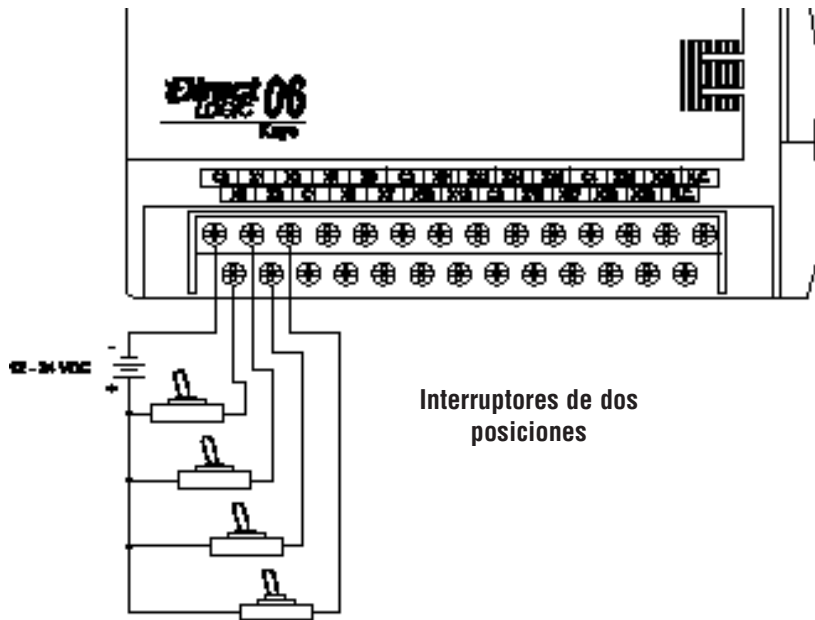
o un

- programador portátil D2-HPP, con firmware versión 2.0 o mejor, (viene con el cable de programación). Compre por favor el manual del programador D2-HPP-M en separado.

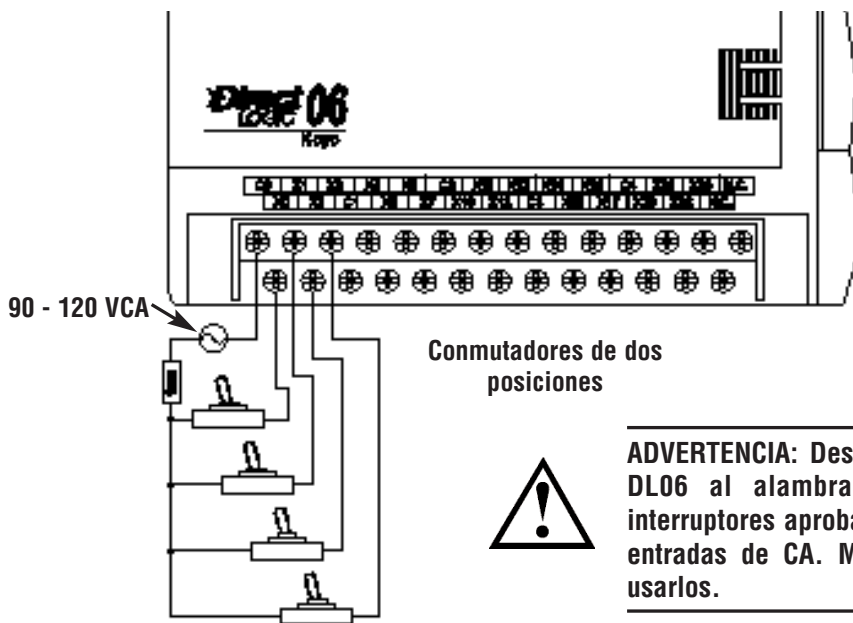


Paso 2: Conecte los conmutadores a los terminales de entrada

Para proceder con este ejercicio o seguir otros ejemplos en este manual, usted necesitará conectar uno o más conmutadores de entrada según lo mostrado en la figura de abajo. Si usted tiene entradas de corriente continua en un PLC DL06, puede utilizar la fuente de poder auxiliar de 24 VCC en el bloque de terminales de salida u otra fuente de poder externa de 12 o 24 VCC. Siga las instrucciones en la ADVERTENCIA en esta página.



**D0-06DA, D0-06DD1,
D0-06DD2, D0-06DR,
D0-DD1-D, y D0-06DR1-D**
Entradas CC

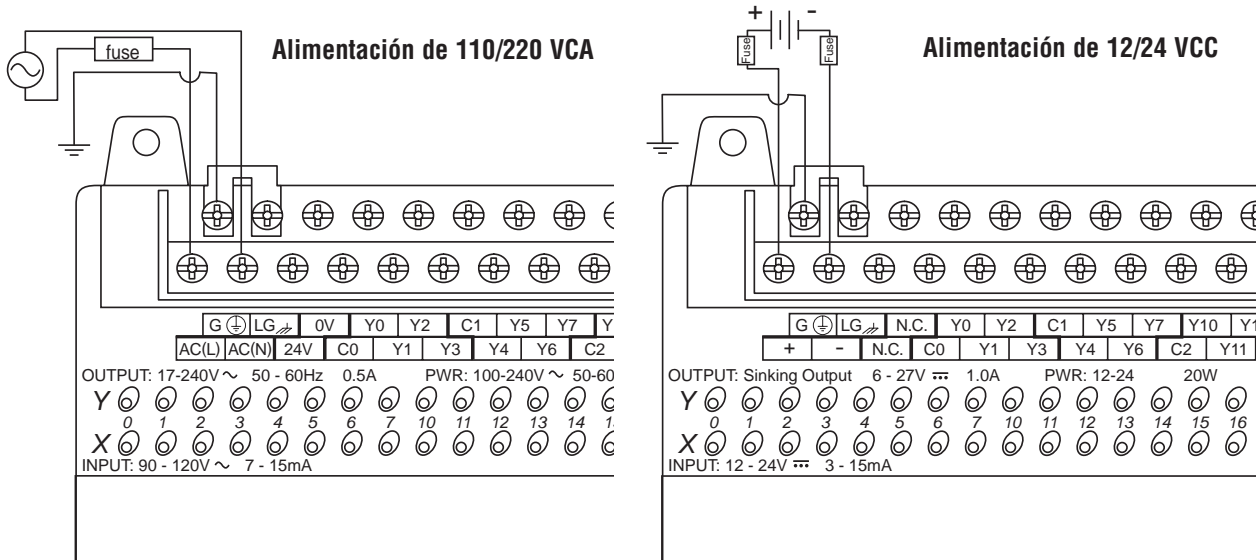


D0-06AA y D0-06AR
Sólo entradas CA

ADVERTENCIA: Desconecte la energía eléctrica y desenchufe el DL06 al alambrear los conmutadores. Utilice solamente interruptores aprobados para por lo menos 250 VCA, 1A para las entradas de CA. Monte firmemente los interruptores antes de usarlos.

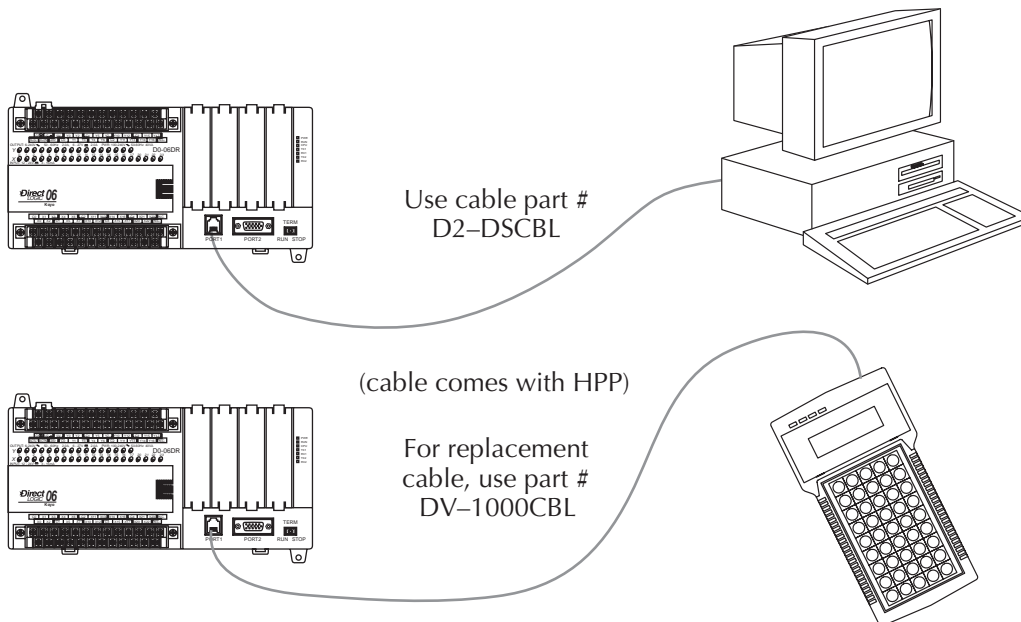
Paso 3: Conecte el cableado de potencia

Conecte el cableado de la entrada de potencia al PLC DL06. Observe todas las precauciones indicadas en este manual. Para más detalles del cableado, vea el capítulo 2 para instalación, cableado y especificaciones. Cuando el cableado esté completo, cierre la cubierta del conector. No aplique energía en este momento



Paso 4: Conecte el dispositivo de programación

La mayoría de los programadores utilizan el software de programación *DirectSOFT*, versión 4.0 o más nuevo, instalado en una computadora. Como alternativa, si necesita un dispositivo de programación portátil, el programador HPP (Por lo menos Firmware versión 2.20). Ambos dispositivos se conectan al puerto 1 del DL06 con el cable apropiado.



Nota: El programador portátil no puede crear o tener acceso a instrucciones LCD, ASCII o MODBUS.

Paso 5: Encienda la energía del sistema del PLC

Aplique energía al sistema y asegúrese que el indicador PWR en el PLC DL06 está encendido. Si no es así, apague el PLC y compruebe todo el cableado y vea la sección de localización de fallas en el capítulo 9 para más ayuda.

Paso 6: Inicialice la memoria scratchpad

Es una buena práctica que por precaución limpie la memoria del sistema (memoria scratchpad) en un DL06 nuevo. Hay dos maneras de limpiar la memoria de sistema:

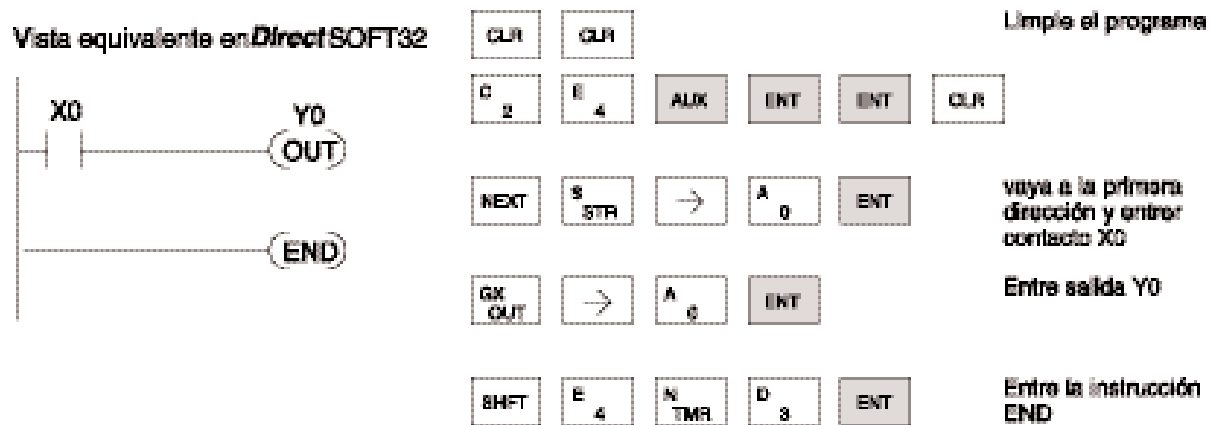
- En *DirectSOFT*, seleccione el menú PLC, luego Setup, después Initialize Scratchpad. Para información adicional, vea el manual de *DirectSOFT*.
- Usando el programador portátil, utilice la tecla AUX y ejecute AUX 54. Vea el manual del programador portátil para información adicional.

Paso 7: Entrar un programa ladder (o de escalera)

En este momento, los técnicos programadores con *DirectSOFT* deben estudiar por lo menos la clase rápida de tutoría de como comenzar en el manual *DirectSOFT*. Allí se aprenderá cómo establecer un enlace (link) de comunicación con el PLC DL06, cambiar modos de la CPU a RUN o a Program, y entrar un programa al PLC. Vea también el capítulo 11.

Si usted está aprendiendo cómo programar con el programador portátil, asegúrese que la CPU está en modo Program (el LED RUN en el frente del DL06 debe estar apagado). Si el LED RUN está encendido, utilicer la tecla MODE en el programador portátil para poner el PLC en modo Program, y luego cambie a TERM.

Apriete las teclas siguientes en el programador portátil.



Después de entrar el programa simple del ejemplo coloque el PLC en modo RUN (funcionamiento) usando la tecla MODE en el programador portátil.

El indicador de RUN en el PLC iluminará indicando que la CPU ha entrado al modo de funcionamiento. Si no es así, repita este paso, asegúrese que el programa fue hecho correctamente o vea a la guía de localización de fallas en el capítulo 9.

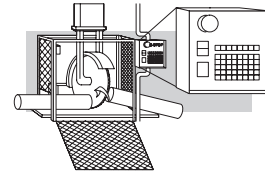
Después de que la CPU entre en el modo de funcionamiento (RUN), el indicador de estado de la salida para Y0 debe seguir el estado del interruptor en el canal X0 de entrada. Cuando el interruptor está encendido, la salida estará encendida.

1

Pasos para diseñar un sistema con suceso

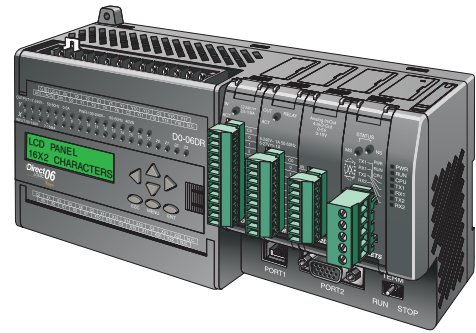
Paso 1: Repase las normas de la instalación

Siempre haga la seguridad la primera prioridad en cualquier diseño de un sistema. El capítulo 2 proporciona varias reglas que le ayudarán a diseñar un sistema más seguro, más confiable. Este capítulo también incluye reglas del cableado para las varias versiones del PLC DL06.



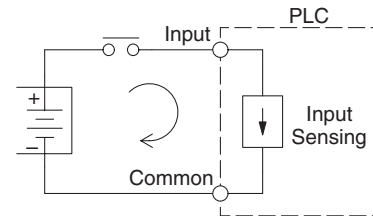
Paso 2: Entienda los procedimientos de la configuración del PLC

El PLC es el corazón de su sistema de automatización. Asegúrese de tomar un tiempo para entender todas las características y requisitos de configuración.



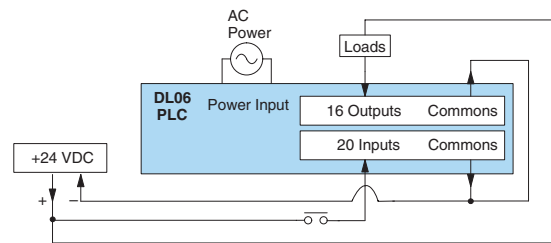
Paso 3: Repase los criterios de selección de E/S

Hay muchas consideraciones cuando usted selecciona los tipos de dispositivos de entradas y salidas. Tome un momento para entender cómo los varios tipos de sensores y de cargas pueden afectar su opción del tipo de E/S.



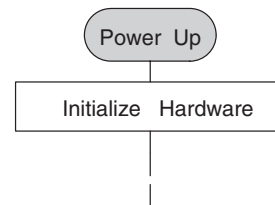
Paso 4: Escoja una estrategia de cableado del sistema

Es importante entender las varias opciones de diseño del sistema que están disponibles antes de cablear dispositivos de campo y fuentes de alimentación del lado del campo para el PLC.



Paso 5: Entienda la operación del sistema

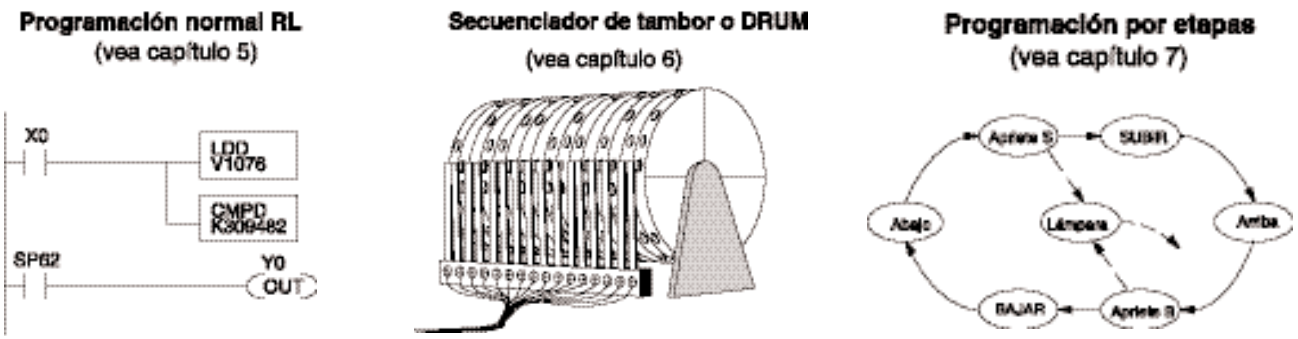
Antes de que usted comience a implementar un programa, es muy importante entender cómo el sistema DL06 procesa la información. Esto implica no solamente pasos de ejecución de programa, pero también implica las varias características de la disposición de la memoria y del modo de operación.



Paso 6: Repase los conceptos de programación

El sistema de instrucciones del PLC DL06 permite usar tres métodos de programación principales para resolver el programa de su sistema, representado en la figura abajo.

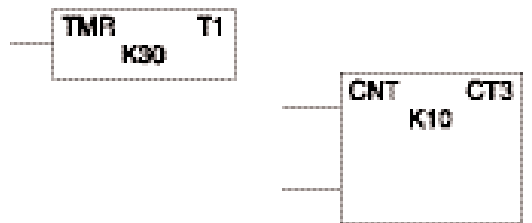
- La programación del diagrama-estilo RLL es la mejor herramienta para solucionar lógica booleana y manipulación general de la memoria en la CPU. Incluye docenas de instrucciones, que pueden ser también necesarias para aumentar las funciones de tambores y etapas.
- El secuenciador de tambor de tiempos y eventos ofrece hasta 16 pasos y ofrece tiempo y/o transiciones de eventos en cada paso. La instrucción DRUM es la mejor para un proceso repetido basado en una sola serie de pasos.
- La programación por etapas (también llamada RLLPlus) se basa en diagramas de transiciones de estado. Las etapas dividen el programa en secciones que corresponden a los estados en un organigrama o diagrama de flujo que usted dibuja para su proceso.



Después de repasar los conceptos de programación arriba, Ud. tendrá una variedad de herramientas para escribir su programa de uso.

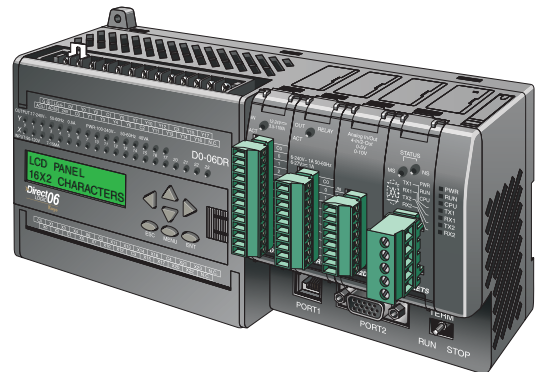
Paso 7: Escoja las instrucciones

Una vez que haya instalado el PLC y entienda los conceptos de programación principales, usted puede comenzar a escribir el programa. En este momento usted comenzará a utilizar uno de los sistemas de instrucción más poderosos disponibles en un PLC pequeño.



Paso 8: Entienda la mantención y los procedimientos de localización de fallas

Las fallas de equipo ocurren a veces cuando menos se espera. Los interruptores fallan, las cargas hacen cortocircuitos y hay necesidad de substituir piezas, etc. En la mayoría de los casos, el mayor tiempo de localización de fallas y de mantención es intentar localizar el problema. El PLC DL06 tiene muchas características incorporadas tales como códigos de error que pueden ayudarle a identificar problemas.



Preguntas y respuestas sobre PLCs DL06

¿Como es el sistema de instrucciones?

El sistema de instrucciones ladder es muy similar al de la CPU DL260. Vea el capítulo 5 para más detalles. Las instrucciones del DL06 incluyen la instrucción tambor y capacidades para establecimiento de una red ASCII o MODBUS, el visor LCD y entradas y salidas de alta velocidad. Las entradas y salidas de alta velocidad están solamente disponibles en unidades con entradas de C.C.

¿Tengo que comprar el paquete de programación completo *DirectSOFT* para programar el DL06?

Si. Se ofrece la versión de *DirectSOFT5* para todos los PLCs de la familia *DirectLOGIC* es muy económica. Hay, sin embargo, un programa gratuito que es limitado (PD-DS100) en que solamente puede programar hasta 100 palabras de código ladder. Ud puede hacer todas las otras funciones del programa. El manual del programa está disponible en formato PDF en inglés en la carpeta HELP del conjunto de archivos de *DirectSOFT5*.

¿Es el DL06 expandible?

Sí, la serie DL06 funciona como un PLC independiente. Sin embargo, las 4 ranuras de módulos opcionales le permiten ampliar el sistema con el mismo tamaño.

¿Tiene el PLC DL06 capacidad de control de movimiento?

Sí, el DL06 tiene una capacidad limitada de control de movimiento. Las funciones de alta velocidad de entradas y salidas permite contar pulsos de alta velocidad de un encoder y tener valores prefijados con interrupción o una salida de pulsos y dirección para el control de motores stepper. Hay tres tipos de perfiles de movimiento disponibles, que se explican en el capítulo 3. Además, es posible usar el módulo H0-CTRIO para agregar un eje adicional.

¿Se almacenan los programas en un EEPROM removable?

No. El PLC DL06 contiene una memoria FLASH fija para el almacenaje del programa, que se puede escribir y borrar millares de veces. Usted puede transferir los programas desde o para *DirectSOFT* en una PC.

¿Tiene el PLC DL06 fusibles para sus salidas?

No hay fusibles en el circuito de salida. Por lo tanto, recomendamos colocar un fusible en cada canal o en cada común. Vea el capítulo 2 para reglas del cableado de E/S.

¿Es el PLC DL06 aprobado por U.L.?

El PLC cumple con los requisitos de UL (Underwriters' Laboratories, Inc.), y CUL (Canadian Underwriters' Laboratories, Inc.). Vea nuestro sitio de Internet, www.Automationdirect.com, para detalles completos.

¿Cumple el PLC DL06 con las directivas de la Unión Europea (EU)?

El PLC cumple con los requisitos de las directivas de la Unión Europea (CE). Vea nuestro sitio de Internet, www.automationdirect.com, para detalles completos.

¿Qué dispositivos se pueden conectar a los puertos de comunicación del DL06?

Puerto 1: El puerto es RS-232C, fijo en 9600 baud, paridad impar, dirección 1, y utiliza el protocolo propietario K-sequence. El DL06 se puede también conectar con las redes MODBUS RTU y DirectNET como dispositivo esclavo a través del puerto 1. El puerto se comunica con los dispositivos siguientes:

- Interfaces de operador DV-1000, C-more y C-more micro, *DirectTouch*, *LookoutDirect*, *DSDData*, *KEP Direct* u *Optimation*
- *DirectSOFT* (que funciona en una computadora personal)
- Programador portátil D2-HPP
- Otros dispositivos que se comunican con protocolos K-sequence, *DirectNET* o MODBUS RTU pueden trabajar con el DL06 PLC. Contacte al vendedor del dispositivo específico para más detalles.

Puerto 2: Este es un puerto de funcionamiento múltiple. Permite usar RS-232C, RS422 o RS485, con velocidad (300-38,400bps), dirección y paridad seleccionables. También permite usar el protocolo propietario K-sequence así como los protocolos DirectNet y MODBUS RTU, ASCII In/Out (de entrada y de salida) y de non-sequence (Para imprimir a una impresora serial).

El módulo opcional también permite establecer comunicaciones seriales como maestro o esclavo. vea más detalles en el manual de este módulo.

¿Acepta el PLC DL06 entradas de 5VVC?

No, 5 Volt es más bajo que el valor mínimo ON de la entrada de C.C. Sin embargo, muchos circuitos de lógica TTL pueden adaptar las entradas si se conectan con entradas de colector abierto (drenadoras). Vea el capítulo 2 para las pautas de cableado de entradas y salidas.

INSTALACIÓN, CABLEADO Y ESPECIFICACIONES



CAPÍTULO 2

En este capítulo...

Consideraciones de seguridad	2-2
Explicación de la parte frontal del panel del PLC DL06	2-4
Instrucciones generales de montaje	2-6
Consideraciones de cableado	2-10
Estrategias de cableado del sistema	2-13
Glosario de términos de la especificación	2-25
Diagramas eléctricos y especificaciones	2-26

Pautas de seguridad

2



NOTA: Los productos con la marca **CE** ejecutan sus funciones con seguridad y cumplen con las normas relevantes como especificado por los directorios de **CE**, si es que se utilizan según su propósito y que las instrucciones en este manual sean seguidas. La protección proporcionada por el equipo puede bajar si este equipo se utiliza de una manera no especificada en este manual. Un listado de nuestros afiliados internacionales está disponible en nuestro sitio de Internet <http://www.automationdirect.com>



ADVERTENCIA: Es su responsabilidad suministrar un ambiente de funcionamiento seguro para el personal y el equipo y debe ser su meta fundamental durante el planeamiento y la instalación del sistema. Los sistemas de automatización pueden fallar y pueden dar lugar a situaciones que pueden causar lesión seria al personal o daño al equipo. No confíe sólo en el sistema de automatización para tener seguridad. Debe utilizar dispositivos electromecánicos externos, tales como relevadores o interruptores límite, que son independientes del uso del PLC para tener una protección para cualquier parte del sistema que pueda causar daños corporales. Cada uso de la automatización es diferente, de modo que pueden haber requisitos especiales para su uso particular. Asegúrese de seguir todas las normas nacionales y requisitos locales para instalación y uso apropiado de su equipo.

Planee hacer un sistema seguro de control

La mejor manera de suministrar un ambiente seguro es planear la seguridad del personal y del equipo de proceso. Se debe examinar cada aspecto del sistema para determinar qué áreas son críticas a la seguridad del operador o de la máquina. Si usted no está familiarizado con las prácticas de instalación de un sistema con PLCs o su compañía no ha establecido normas de instalación, debe obtener información adicional de otras fuentes, tales como:

- NEMA — The National Electrical Manufacturers Association, localizada en Washington, D.C. USA, publica muchos documentos que discuten los estándares para los sistemas de control industriales. Se puede pedir estas publicaciones directamente de NEMA. Algunos de éstos incluyen:
ICS 1, General Standards for Industrial Control and Systems.
ICS 3, Industrial Systems.
ICS 6, Enclosures for Industrial Control Systems.
- NEC — El código eléctrico nacional proporciona regulaciones referentes la instalación y al uso de varios tipos de equipo eléctrico. Se pueden obtener copias del manual de NEC de su distribuidor local del equipo eléctrico o de su biblioteca local.
- Agencias locales - muchas agencias locales tienen requisitos adicionales sobre y más allá de éstos descritos en el manual de NEC. Verifique con la oficina eléctrica local para más información.

Tres niveles de protección

Las publicaciones mencionadas proporcionan muchas ideas y requisitos para la seguridad del sistema. Como mínimo, se deben seguir estos reglamentos. También, usted debe utilizar las técnicas siguientes, que proporcionan tres niveles de control de sistema.

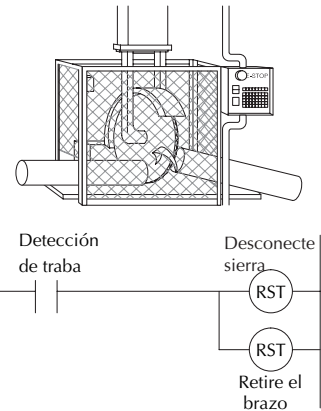
- Secuencia ordenada de la parada normal del sistema en el programa de control del PLC
- Desconexión mecánica de la energía eléctrica del módulo de salida
- Interruptor de parada de emergencia para desconectar la energía eléctrica del sistema

Parada ordenada normal del sistema

El primer nivel de detección de fallas es idealmente el programa de control del PLC que puede identificar problemas en la máquina. Usted debe planear la secuencia de parada que se debe realizar en estos casos. Estos tipos de problemas son generalmente cosas tales como piezas atoradas, etc. que no coloquen riesgo de daños corporales o daño de equipo.



ADVERTENCIA: El programa de control no debe ser la única forma de protección para ningún problema que pueda resultar en un riesgo de daños corporales o del equipo.

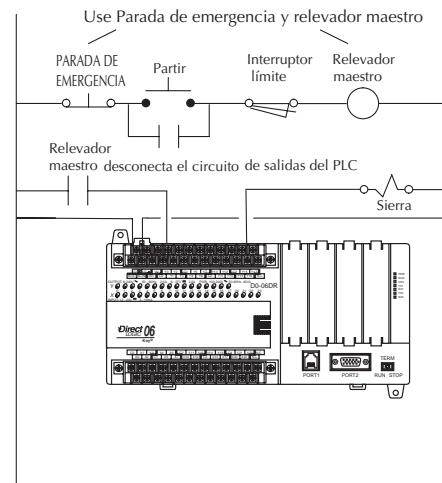
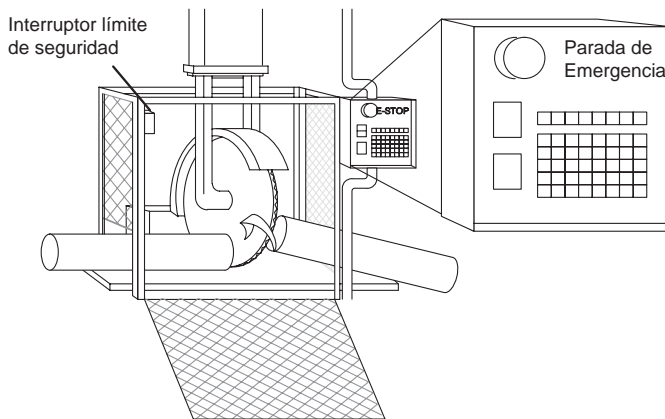


Desconexión del sistema de potencia

Usted debe utilizar también dispositivos electromecánicos, tales como relevadores de control e interruptores de límite maestros, para prevenir arranque accidental del equipo en un momento inesperado. Estos dispositivos se deben instalar en tal manera que eviten que ocurra cualquier operación de máquina. Por ejemplo, si la máquina tiene una parte atorada, el programa de control del PLC puede apagar el motor de una sierra y contraer el cenador. Sin embargo, ya que el operador debe abrir la puerta para quitar la pieza, usted debe también incluir un interruptor que desconecte todas las salidas del sistema cuando se abra la puerta en cualquier momento.

Parada de Emergencia

La maquinaria debe tener un método manual rápido de desconectar toda la energía de las salidas del sistema. El dispositivo o el interruptor de desconexión se debe rotular claramente "Parada de emergencia".



Después de una parada de emergencia o de cualquier otro tipo de interrupción de energía, puede haber requisitos que deben ser resueltos antes de que el programa de control del PLC se pueda recomenzar. Por ejemplo, puede haber valores específicos de memorias que deben ser establecidos (o el estado antes de la parada debe ser mantenido) antes de que las operaciones puedan reanudarse. En este caso, se pueden utilizar posiciones de memoria retentivas o incluir constantes en el programa de control para asegurar un punto de partida conocido.

Aprobación de area Clase 1, División 2

Este equipo es adecuado para usar en áreas clasificadas peligrosas Clase 1, Division 2, grupos A, B, C y D definidos por NEC, solamente.

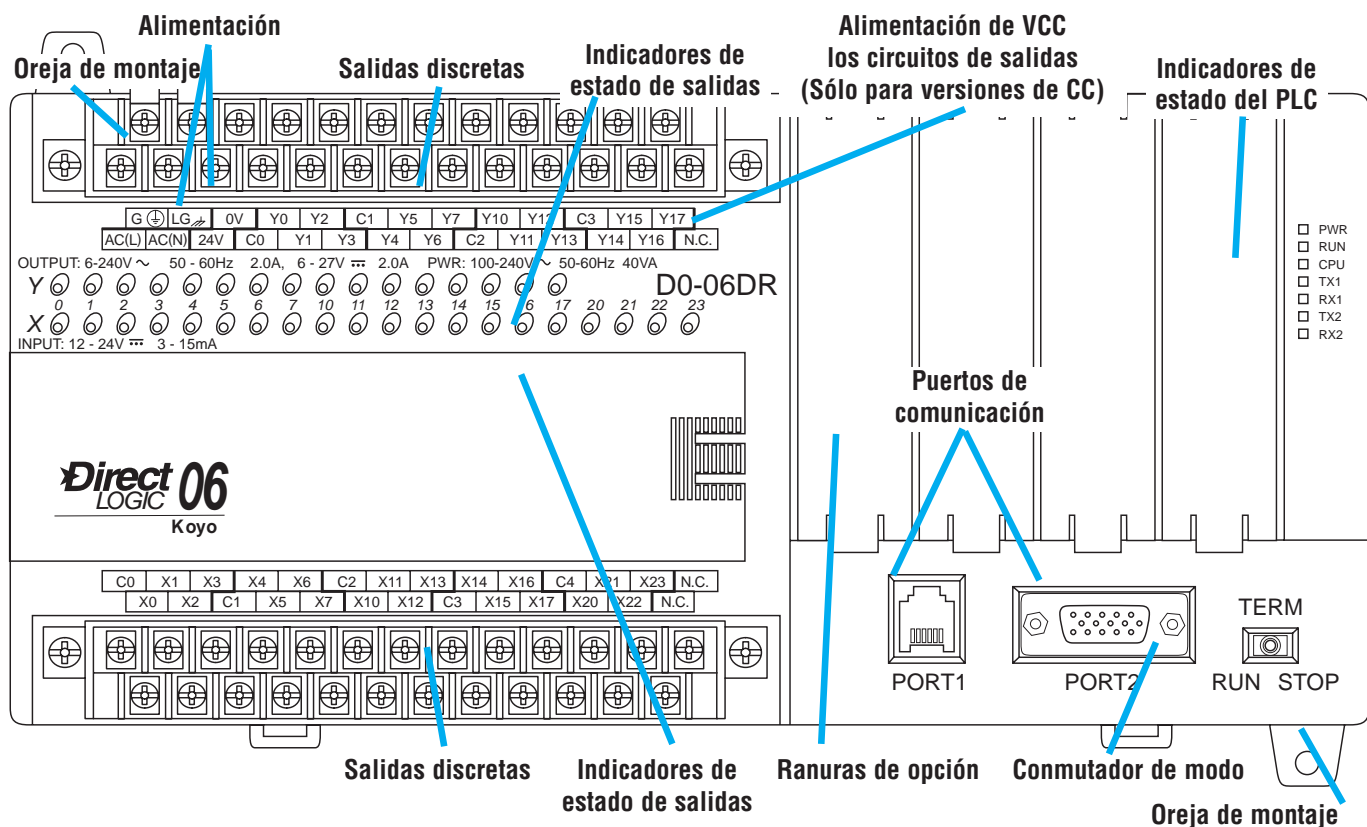
2



ADVERTENCIA: Peligro de explosión! La sustitución de componentes puede deteriorar la condición de uso para clase 1, división 2. No desconecte el equipo a menos que se haya apagado la energía o se sabe que el área no es peligrosa.

Explicación de la parte frontal del panel del PLC DL06

La mayoría de las conexiones, indicadores y etiquetas en el PLC DL06 están situados en su panel frontal. Los puertos de comunicación están situados en el frente del PLC al igual que las ranuras de tarjeta de opción y el switch selector de modo. Vea por favor la figura de abajo.



El bloque terminal de salidas y de alimentación acepta conexiones de tierra externa al chasis y a la lógica en los terminales indicados. Los terminales restantes son para los comunes y las conexiones de salida de Y0 hasta Y17. Los dieciséis terminales de salida se numeran en octal, Y0-Y7 e Y10-Y17. En unidades de salida de C.C., el terminal del extremo a la derecha acepta 24 VCC para la etapa de salida. El bloque terminal del lado de entradas permite conectar las entradas X0 hasta X23 y los comunes asociados



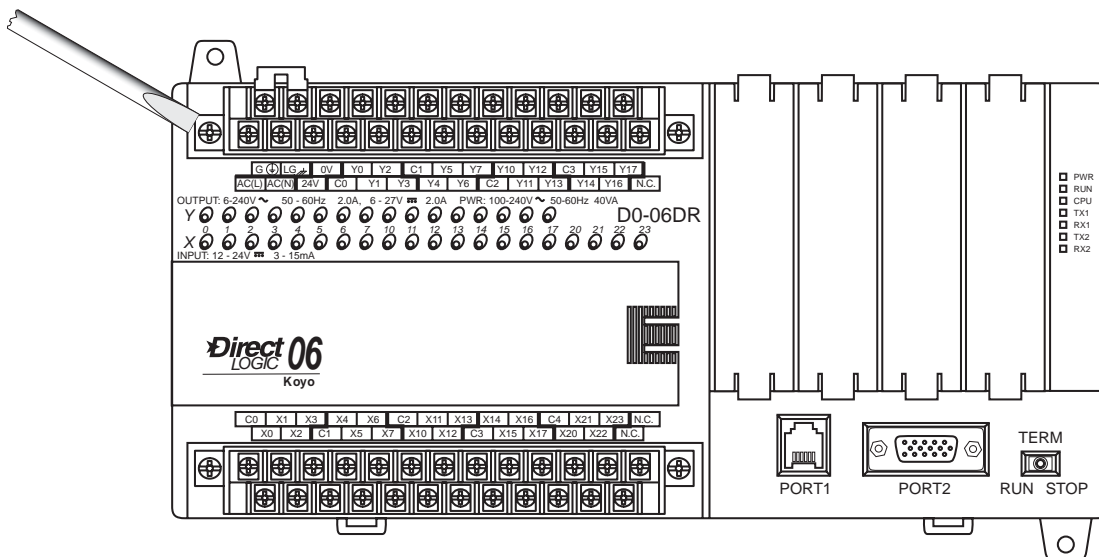
ADVERTENCIA: En algunos casos, puede haber tensión de dispositivos de campo en el bloque de terminales aunque el PLC se haya apagado. Para reducir al mínimo el riesgo de choque eléctrico, verifique que los dispositivos de campo estén desconectados antes de que se exponga o quite cualquier conector.

Removiendo el bloque de terminales

Los terminales del DL06 se dividen en dos grupos. Cada grupo tiene su propio bloque de terminales. Las salidas y el cableado de la energía están en un bloque, y el cableado de las entradas está en el otro. En algunos casos, puede ser deseable quitar el bloque de terminales para hacer fácil el cableado. El bloque de terminales está diseñado para retirarlo fácilmente con un destornillador pequeño. El dibujo de abajo muestra el procedimiento para quitar uno de los bloques de terminales.

2

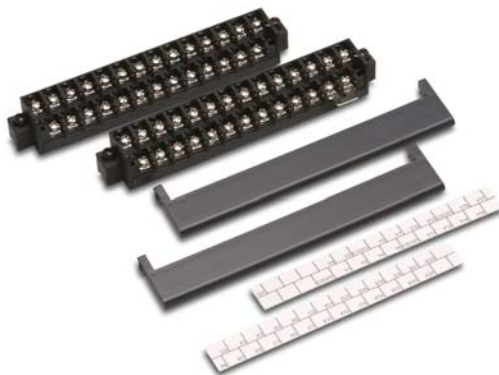
1. Afloje los tornillos de retención en cada extremo del bloque de terminales.



2. Levante desde el centro del bloque de terminales con el destornillador hasta que el bloque esté suelto.

Los bloques terminales en los PLCs DL06 tienen terminales de tornillo (tamaño M3), que aceptarán destornillador de hoja plana o Philips #1. Utilice alambre trenzado No. 16 a 22 AWG. Tenga cuidado de no apretar demasiado el tornillo; el torque necesario es 0,882 a 1,02 N-m (7,80 a 9,0 pulgada-libras).

Los bloques terminales de repuesto están disponibles en un juego de accesorios cuyo número de artículo es D0-ACC-2. Usted puede encontrar éste y otros accesorios en nuestro sitio de Internet.



Instrucciones generales de montaje

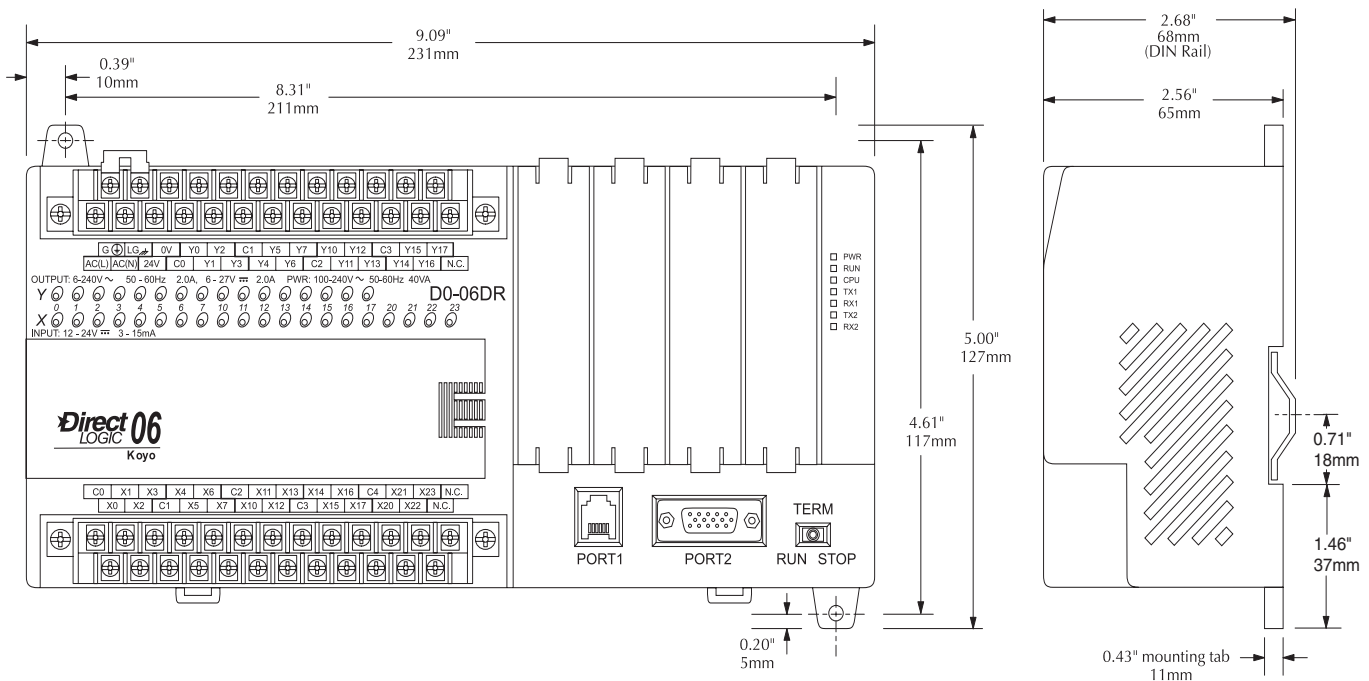
Además de las pautas de montaje del panel, otras especificaciones pueden afectar la instalación de un sistema de PLC. Considere siempre lo siguiente:

2

- Especificaciones ambientales
- Requisitos de energía
- Aprobaciones de agencias
- Dimensiones del gabinete donde será montado y de los componentes accesorios.

Dimensiones de la unidad

El diagrama siguiente muestra las dimensiones y las localizaciones del agujero de montaje para todas las versiones del DL06. Asegúrese que siga las pautas de instalación para permitir el espaciado apropiado de otros componentes.



Gabinetes

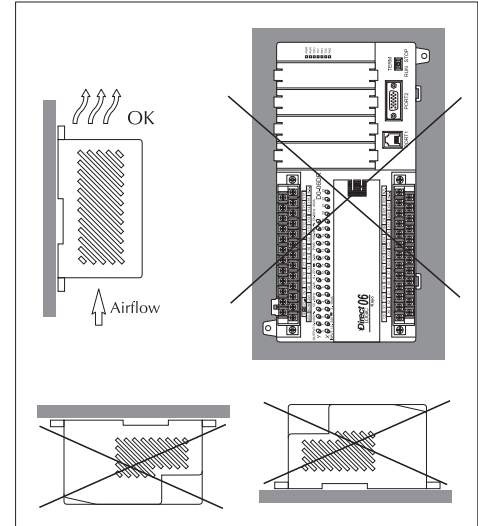
La selección de un gabinete o cubículo apropiado es importante para asegurar una operación segura y apropiada de su sistema DL06. Los usos de los sistemas DL06 varían y pueden requerir características adicionales. Las consideraciones mínimas para los gabinetes incluyen:

- Conformidad a las normas eléctricas
- Protección contra los elementos en un ambiente industrial
- Referencia común de la tierra
- Mantenimiento de la temperatura ambiente especificada
- Tener acceso al equipo
- Seguridad con acceso restringido
- Suficiente espacio para la instalación y mantenimiento apropiada del equipo

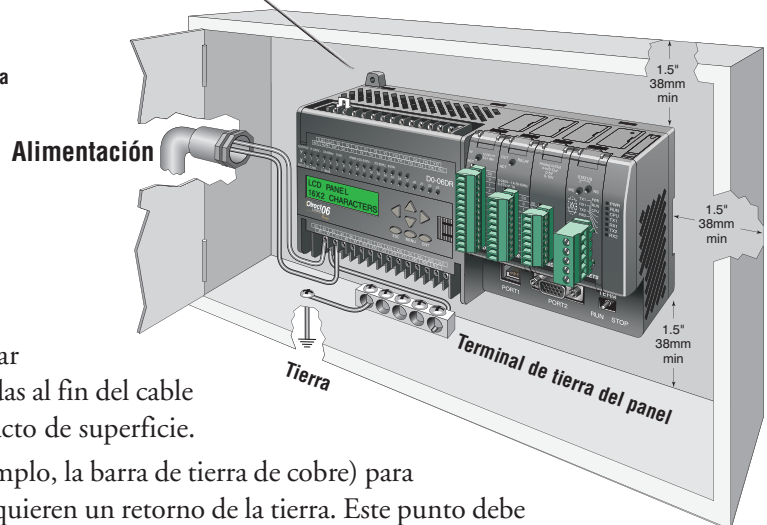
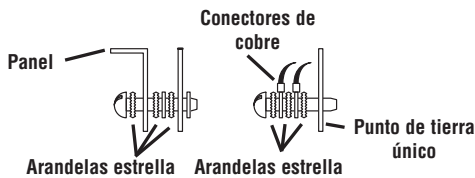
Montaje en un gabinete y distancias mínimas

Hay muchas cosas a considerar cuándo se diseña la disposición de un panel. Los siguientes asuntos corresponden al esquema mostrado. Nota: pueden haber requisitos adicionales, dependiendo de su aplicación y el uso de otros componentes en el gabinete.

1. Monte el PLC horizontalmente como mostrado debajo, para proporcionar ventilación apropiada. Usted no debe montar las unidades DL06 verticalmente, al revés, ni en una superficie horizontal plana. Si usted coloca más de una unidad en un gabinete, debe haber un mínimo de 7.2" (183 mm) entre las unidades.
2. Deje un espacio libre mínimo de 1.5" (39 mm) entre la unidad y todos los lados del gabinete. Recuerde de tener en cuenta cualquier panel de operador u otros artículos montados en la puerta.
3. Debe haber también por lo menos 3" (78 mm) de espacio libre entre la unidad y cualquier conducto de alambrado que corre paralelo a los terminales.



NOTA: Hay un requerimiento de espacio mínimo de 1.5" (38 mm) entre la pi del panel (o cualquier aparato mor en la puerta del panel) y el compor. más cercano al PLC DL06



4. El terminal de tierra en la base DL06 debe ser conectado a un solo punto de tierra. Use alambre trenzado de cobre para conseguir una baja impedancia. Se deben usar conectores de cobre o deben ser soldadas al fin del cable trenzado para asegurar un buen contacto de superficie.

5. Debe haber una sola tierra (por ejemplo, la barra de tierra de cobre) para todos componentes en el panel que requieren un retorno de la tierra. Este punto debe ser conectado a la terminación de la tierra del panel. Los tamaños mínimos de cable, la codificación de colores y las prácticas generales de la seguridad deben estar de acuerdo con códigos y estándares eléctricos apropiados para su área.

6. Una buena referencia de tierra es esencial para la operación apropiada del DL06. Se debe conectar un lado del blindaje del cable en todos los circuitos de control y de energía correctamente a una tierra adecuada. Hay varios métodos para tener una referencia adecuada de tierra, incluyendo:

- a) Instalando una barra de tierra tan cerca del panel como sea posible.
- b) Conexión de la tierra del sistema de energía.

7. Evalúe cualquier instalación donde la temperatura ambiente puede acercarse al límite superior o inferior de las especificaciones. Si usted sospecha que la temperatura del ambiente no estará dentro de la especificación de operación para el sistema DL06, se deben tomar medidas tales como instalar una refrigeración o una calefacción para obtener la temperatura ambiente dentro de los rangos de especificaciones.

8. Los sistemas DL06 se diseñan para ser accionados por 95-240 VCA o 12-24 VCC, normalmente disponibles en un ambiente industrial. La energía eléctrica en algunas áreas donde se instala el PLC no es siempre estable y las tempestades pueden causar picos de tensión. Debido a esto se recomienda usar filtros de línea para proteger el PLC DL06 de picos y ruido de RFI. El filtro Powerline de Automationdirect, para uso con 120 VCA y 240 VCA, 1-5 A es una opción excelente (vea www.automationdirect.com). Sin embargo, se puede usar cualquier filtro. Estas unidades se instalan entre la alimentación y el PLC.

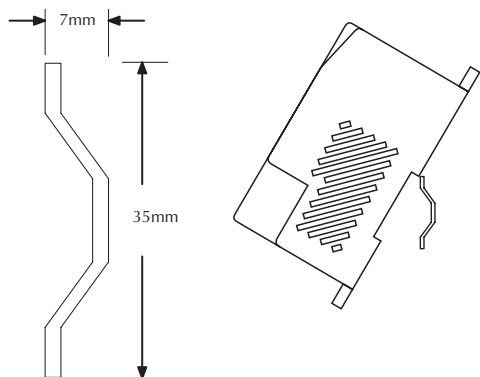


NOTA: Si usted está utilizando otros componentes en su sistema, asegúrese de ver el manual apropiado para determinar cómo pueden afectar esas unidades las dimensiones de montaje.

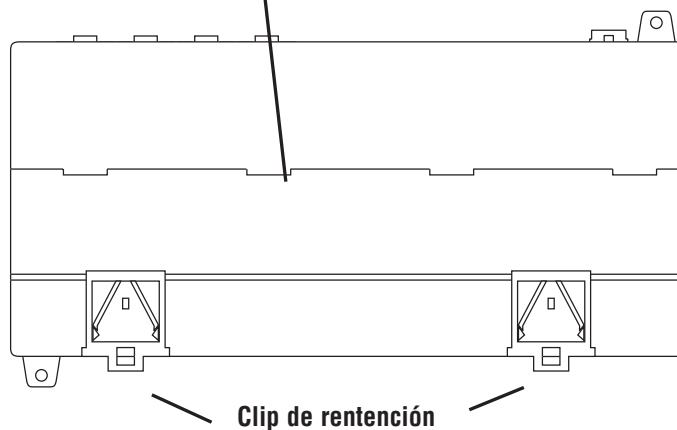
Usando rieles de montaje

Los PLCs DL06 pueden ser montados en un gabinete usando rieles DIN. Recomendamos rieles que cumplen con el estándar DIN EN 50 022. Ellos son aproximadamente 35 mm de alto, con una profundidad de 7 mm. En la parte trasera del PLC hay dos clips pequeños que permiten sujetar el PLC al riel. Para montar el PLC a un riel DIN, colóquelo en el riel y levante suavemente los clips para agarrar el riel. Para retirar el PLC, empuje los clips para abajo, levante el PLC levemente, y luego ya puede moverlo del riel.

Dimensiones del riel DIN



La ranura del riel DIN está diseñada para 35mm x 7 mm de acuerdo a DIN EN 50022





NOTA: Vea nuestro catálogo o sitio de Internet para una lista completa de piezas del sistema de conexiones **DINnector**.

Especificaciones ambientales

La tabla mostrada abajo lista los requerimientos de ambiente que se aplican generalmente a los PLCs DL06. Los rangos que para el programador portátil se muestran abajo de esta tabla. Ciertos tipos de circuito de salidas pueden tener curvas de degradación del valor nominal, dependiendo de la temperatura del ambiente y el número de salidas ON. Vea por favor la sección apropiada en el manual que pertenece a su PLC DL06 particular PLC.

* La temperatura de funcionamiento para el programador portátil y el DV-1000 es 32° a 122° F (0° a 50° C). La temperatura de almacenaje para el programador portátil y el DV-1000 es -4° a 158° F (-20° a 70° C).

**El equipo funcionará hasta una humedad relativa del 5%. Sin embargo, con frecuencia hay más problemas de electricidad estática con niveles bajos de humedad (debajo del 30%). Asegúrese de tomar precauciones adecuadas cuando toque el equipo. Considere usar ground straps (cintas de aterramiento de colocarse en el brazo), cubiertas de piso antiestáticas, etc. si usted utiliza el equipo en ambientes de baja humedad..

2

Especificaciones ambientales	
Especificación	Valores aceptables
Temperatura de almacenamiento	-4° F a 158° F (-20° C a 70° C)
Temperatura de operación del ambiente*	32° F a 131° F (0° C a 55° C)
Humedad del ambiente**	5% – 95% Humedad relativa (non-condensing)
Resistencia a vibración	MIL STD 810C, Method 514.2
Resistencia a choques	MIL STD 810C, Method 516.2
Inmunidad al ruido	NEMA (ICS3-304)
Atmósfera	No corrosive gases
Aprobaciones de agencias	UL, CE (C1D2), FCC class A

Aprobaciones de Agencias

Algunos usos requieren aprobaciones de la agencia para los componentes particulares. Se enumeran abajo las aprobaciones de agencias del PLC DL06 :

- UL (Underwriters' Laboratories, Inc.)
- CUL (Canadian Underwriters' Laboratories, Inc.)
- CE (European Economic Union)

Usos en ambientes marinos

La certificación del American Bureau of Shipping (ABS) requiere una aislación que retarde la propagación del fuego. ABS acepta cables de baja generación de humo, o el cable "plenum rated por NEC (Nivel 4 de resistencia al fuego), u otros cables resistentes a la llama. Use cables en su sistema que cumplan con una norma de retardio de propagación de llama (por ejemplo, UL, IEEE, etc) que incluyan evidencia de certificación de pruebas de cables .



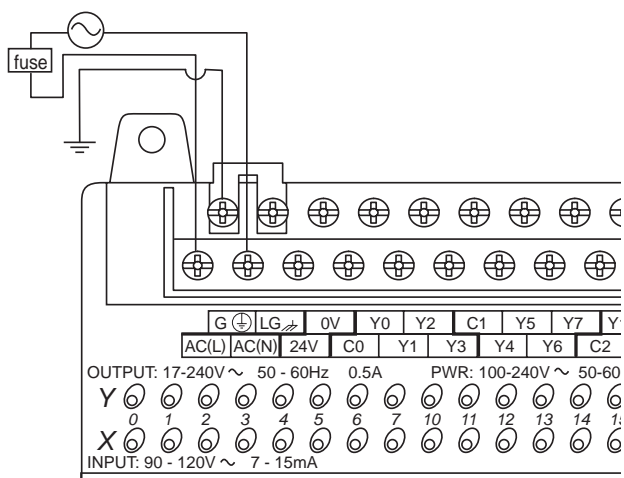
NOTA: Los cables deben ser de "baja generación de humo" de acuerdo al párrafo de arriba. Se recomienda también el uso de cable con cubierta de Teflon.

2

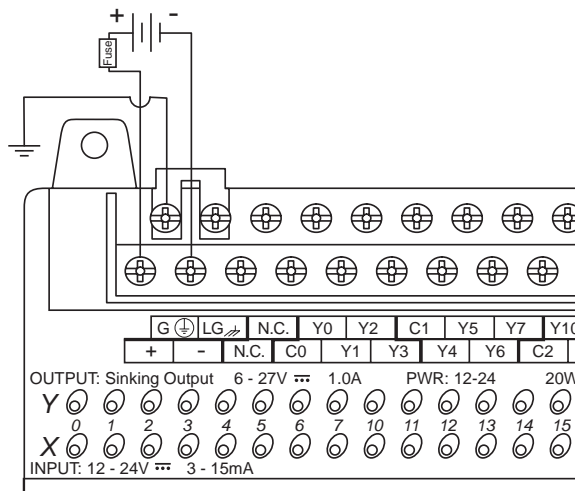
Consideraciones de cableado

Conecte el cableado de la alimentación para el DL06. Observe todas las precauciones indicadas anteriormente en este manual. Para más detalles en el cableado, vea otra parte en este capítulo 2. Cuando el cableado esté completo, cierre las cubiertas del conector. No aplique energía en este momento.

Alimentación de 110/220 VCA



Alimentación de 12/24 VCC



ADVERTENCIA: Una vez que el cableado de la alimentación esté conectado, mantenga la cubierta del bloque de terminales en la posición cerrada. Cuando la cubierta está abierta hay riesgo de choque eléctrico si se toca accidentalmente los terminales de conexión o el cableado de alimentación

Protección de fusibles en la alimentación

No hay fusibles internos en los circuitos de entrada, de modo se necesita una protección externa para asegurar la seguridad del personal de servicio y tener una operación segura del equipo. Para cumplir con las especificaciones de UL/CUL, la alimentación debe tener fusibles. Dependiendo del tipo de energía de entrada que es utilizada, siga estas recomendaciones para protección por fusibles:

Operación de 208/240VCA

Si la fuente de voltaje es un transformador o a tiene dos fases cuando conectada la unidad a una tensión entre 208 a 240 VCA, coloque fusibles en los conductores de la fase (L) y del neutro (N). El tamaño recomendado del fusible es 1,0A (rápido).

Operación de 110/125VCA

Al hacer funcionar la unidad a partir de 110/125 VCA, solamente es necesario un fusible en el conductor de la fase (L1); no es necesario un fusible en el neutro (N). El tamaño recomendado del fusible es 1,0A (rápido).

Operación con 12/24VCC

Al hacer funcionar el PLC con estos voltajes de C.C. más bajos, la sección del cable es tan importante tanto como las técnicas de fusibles adecuadas. Use conductores de gran sección para reducir al mínimo la caída de tensión en el conductor. Cada terminal de alimentación del PLC DL06 puede acomodar un alambre 16 AWG o dos alambres 18 AWG. Una falla de C.C. puede mantener un arco por algún tiempo y a una distancia mucho más grande que fallas de C.A. Típicamente, la barra principal tiene fusibles en un nivel más alto que el dispositivo de rama, que en este caso es el DL06. El tamaño recomendado del fusible para el circuito de rama el DL06 es 1,5 A (por ejemplo, un Littelfuse 312.001 o equivalente).

Alimentación

La fuente de energía debe ser capaz de suministrar voltaje y corriente que cumplan con las especificaciones individuales del PLC, de acuerdo a lo siguiente:

Especificaciones de la fuente de alimentación		
Detalle	Modelos alimentados con CA	Modelos alimentados con CC
Rango de voltaje en las entradas	110/220 VCA (95–240 VCA)	12–24 VCC (10.8–26.4 VCC)
Máxima corriente de Inrush	13 A, 1ms (95–240 VCA) 15 A, 1ms (240–264 VCA)	10A
Máxima potencia	30 VA	20 W
Aplicación de voltaje (dieléctrico)	1 minuto @ 1500 VCA entre primario, secundario y tierra	
Resistencia de aislación	> 10 MOhm a 500 VCC	



NOTA: El grado entre todos los circuitos internos es AISLAMIENTO BASICO SOLAMENTE

Planeando las rutas de cables

Las pautas siguientes proporcionan una información general en cómo alambrear las conexiones de E/S un PLC DL06. Para información específica sobre como conectar un PLC particular vea la hoja correspondiente de la especificación que aparece más adelante en este capítulo.

1. Cada conexión a un terminal del PLC DL06 puede aceptar un alambre 16 AWG o 2 alambres del tamaño 18 AWG. No exceda esta capacidad recomendada.

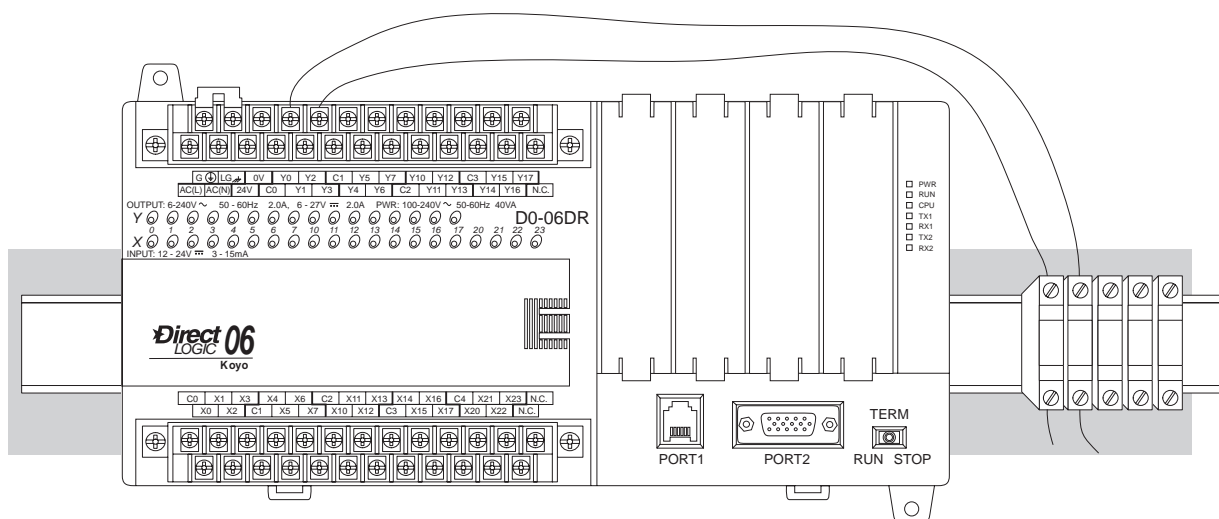


NOTA: El tamaño recomendado de cable para dispositivos de campo es 16 - 22 AWG con hebras o sólido. Apriete los tornillos del terminal con un torque máximo 7,81 lb-ft (0,88 N-m) a 9,03 libra-en (1.02 N-m)..

2. Siempre use una longitud continua de alambre. No empalme los cables.
3. Use la longitud más corta posible del alambre.
4. Use bandejas en lo posible.
5. Evite correr alambres cerca de cableado de alta energía.
6. Evite instalar cables de entradas cerca de cables de salidas en lo posible.
7. Para reducir al mínimo caídas de tensión cuando los alambres tengan una distancia considerable, use múltiples alambres para la línea de vuelta.
8. Evite correr el cableado de C.C. en la proximidad de un cableado de C.A. en lo posible.
9. Evite crear curvas agudas en los alambres.
10. Instale un filtro de EMI recomendado en la alimentación para reducir ruidos de EMI/RFI o picos de tensión.

Protección de fusibles para los circuitos de entradas y de salidas

Los circuitos de entradas y de salidas en el PLC DL06 no tienen fusibles internos. Para tener protección de su PLC, sugerimos que agregue fusibles externos en su cableado de salidas. Un fusible de fusión rápida (Fast blow), con un grado actual más bajo que el grado actual común del banco de E/S se puede conectar en cada común. O se puede agregar un fusible con un grado levemente menor que la corriente máxima por punto de cada salida. Vea las hojas de especificación del PLC en este capítulo para encontrar la corriente máxima por punto de salida o por común de salida. La adición de un fusible externo no garantiza la prevención de daños al PLC, pero aumentará la protección.



Identificación de las entradas y salidas

Todos los DL06 PLCs tienen una configuración fija de E/S. Sigue el mismo sistema de numeración octal usado en otras familias de PLCs DirectLogic, comenzando en X0 y Y0. La letra X es siempre usada para indicar entradas discretas y la letra Y se utiliza siempre para indicar salidas discretas.

La enumeración de E/S comienza en cero y no incluye los dígitos 8 o 9 ya que la dirección es octal. Las direcciones se asignan típicamente en grupos de 8 o 16, dependiendo del número de puntos en un grupo de E/S. Para el DL06 las veinte entradas utilizan los números de referencia X0 - X23. Los dieciséis puntos de salida utilizan las referencias Y0 - Y17.

Los módulos de opción no siguen el direccionamiento en forma consecutiva. Así, la primera entrada en cualquiera de las ranuras es enumerada como X100 y las próximas siguen el mismo criterio en forma consecutiva.

De la misma forma, las salidas son enumeradas Y100 y así sucesivamente.

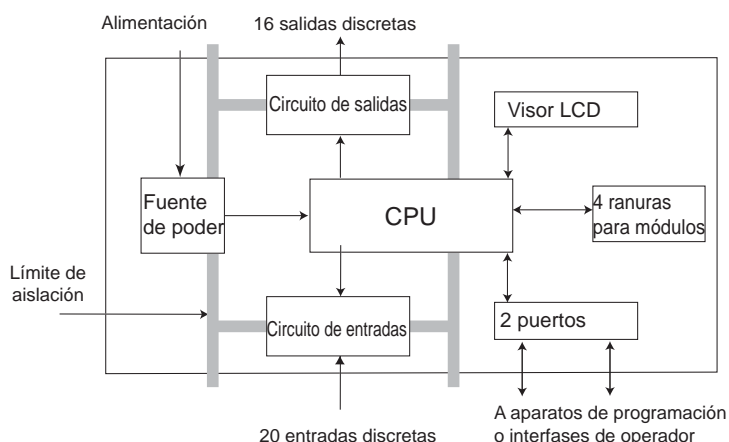
Más información sobre este asunto se encuentra en el manual D0-OPTIONS-M-SP

Estrategias de cableado del sistema

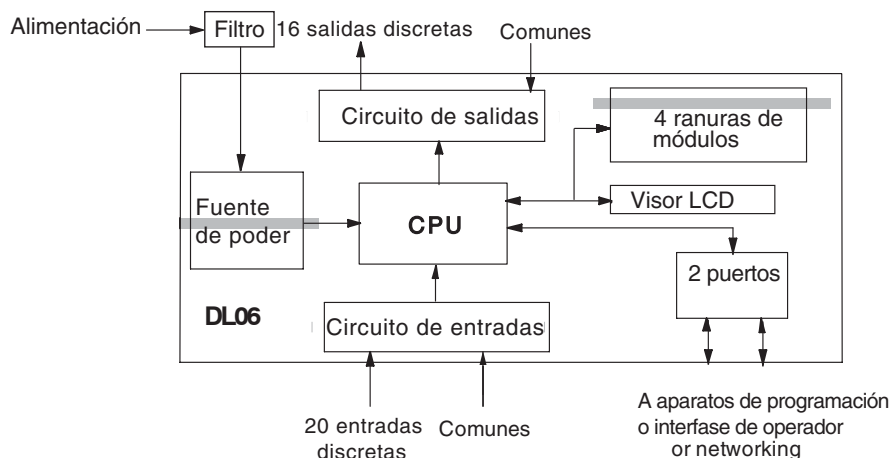
El PLC DL06 es muy flexible y funcionará en diversas configuraciones de cableado. Estudiando esta sección antes de la instalación real, Ud. puede encontrar probablemente la mejor estrategia de cableado para su sistema. Esto le ayudará a tener un costo más bajo del sistema, a evitar errores de alambrado y a problemas de seguridad.

Límites de aislación del PLC

El trazado de los circuitos del PLC se divide en tres regiones principales separadas por los límites de aislación, mostrados en el dibujo abajo. La aislación eléctrica proporciona seguridad, de modo que una falla en una área no se propague a otra. Un filtro EMI hace una aislación entre la fuente de alimentación y la fuente de poder interna. Un transformador en la alimentación proporciona aislación entre los lados primario y secundario. Los acopladores ópticos proporcionan aislación óptica en circuitos de entradas y de salidas. Esto aísla el circuito lógico del lado del campo, donde se conecta la maquinaria de la fábrica. Observe que las entradas discretas están aisladas de las salidas discretas, porque cada una se aísla del lado lógico. Los límites de aislación protegen la interfase del operador (y el operador) contra averías de la entrada de energía o averías del cableado de campo. Cuando se hace el cableado de un PLC, *es extremadamente importante evitar hacer conexiones externas que conecten los circuitos del lado lógico a cualquier otro.*

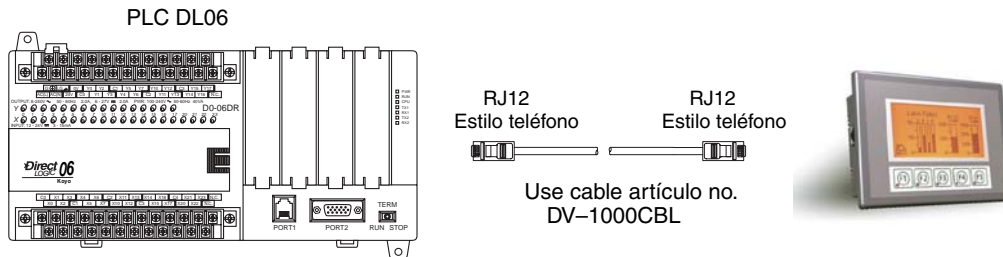


La próxima figura muestra el arreglo interno del DL06, como se ve desde la frente.

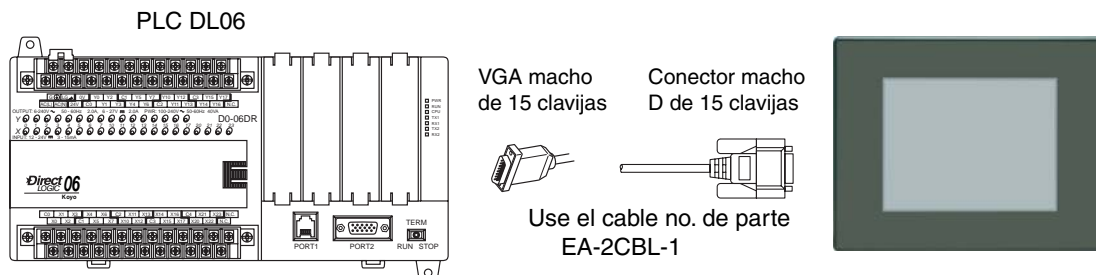


Conectando dispositivos de interfase de operador

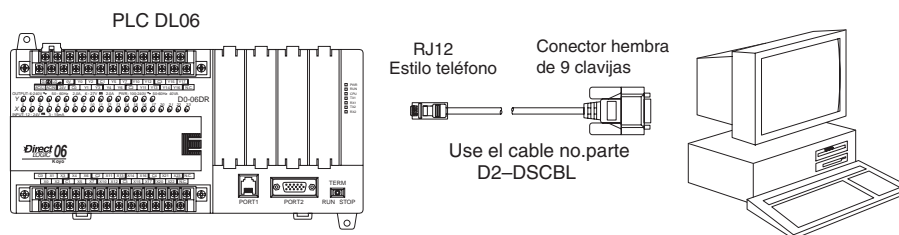
Las interfaces de operador requieren conexiones de datos y de energía. Las interfaces con un monitor de rayos catódicos requieren generalmente corriente alterna separada. Sin embargo, dispositivos pequeños de interfase de operador como la unidad popular de acceso de datos C-more micro se pueden accionar directamente desde el PLC DL06. Conecte el panel al puerto de comunicación 1 en el PLC DL06 usando el cable mostrado abajo. Un solo cable contiene los conductores de transmisión/recepción de datos y energía de +5V.



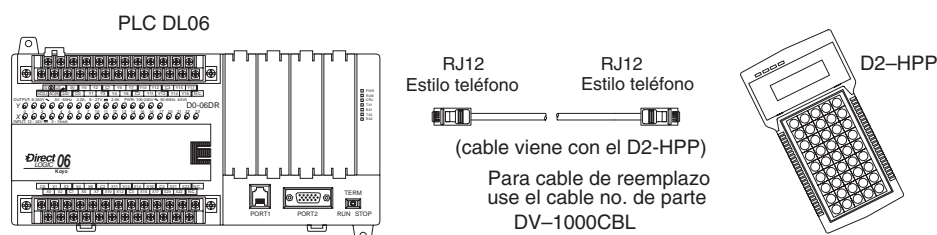
Los paneles de interfase de operador C-more requieren conexiones separadas de energía y de comunicaciones. Conecte el DL06 al conector apropiado en la parte posterior del panel de operador usando el cable mostrado abajo. Estos paneles requieren 24 VCC (rango de operación 20-30 VCC).



Conectando dispositivos de programación



Los PLCs DL06 se puede programar con un programador portátil o con *DirectSOFT* instalado en una PC. Conecte el DL06 a una PC usando el cable mostrado abajo. El programador portátil D2-HPP viene con el cable de comunicación. Para recambio, utilice el cable mostrado abajo.



Concepto de entradas y salidas surtidoras/drenadoras

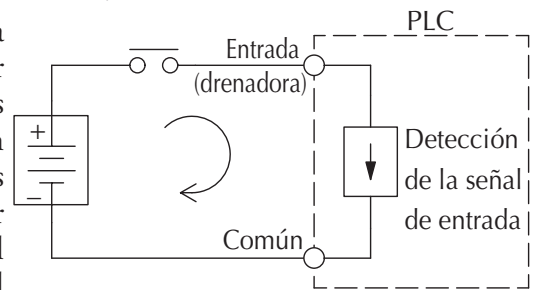
Antes de avanzar en la presentación de estrategias de alambrado, necesitamos introducir los conceptos de "drenadoras" y "surtidoras." Estos términos se aplican a circuitos típicos de entradas o salidas. Es la meta de esta sección de hacer estos conceptos fáciles de entender. Primero damos las definiciones cortas siguientes, seguido por aplicaciones prácticas.

Entradas y salidas drenadoras = Ruta para suministrar negativo (-)

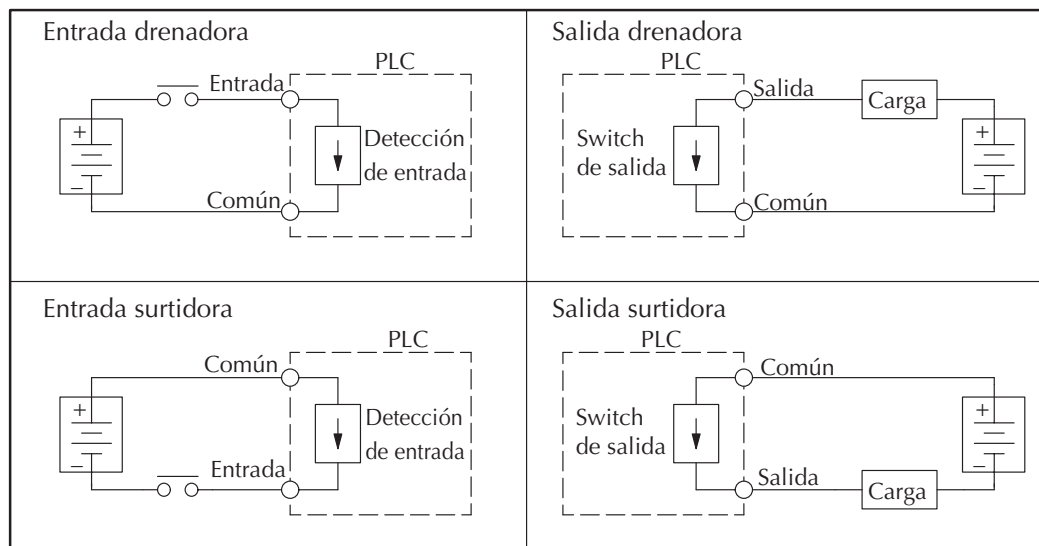
Entradas y salidas surtidoras = Ruta para suministrar positivo (+)

Note la referencia a las polaridades (+) y (-). *La terminología de drenadoras y surtidoras se aplica sólo a circuitos de entradas y salidas de corriente continua (CC).* Los puntos de entradas y las salidas que son drenadoras o surtidoras pueden conducir corriente en una dirección solamente. Esto significa que es posible conectar el aparato externo de suministro y el campo al punto de entradas y salidas de tal forma que la corriente fluya en la dirección errada y el circuito no operará. Sin embargo, podemos conectar exitosamente el aparato de suministro y el campo cada vez entendiendo lo que es una entrada "surtidora" y "drenadora."

Por ejemplo, la figura representa a la derecha una entrada que es "drenadora". Para conectar apropiadamente el suministro externo, solo tenemos que conectar la entrada para que proporcione un camino a negativo (-). De modo que comenzamos en el terminal de la entrada del PLC, seguimos por el circuito del sensor de entrada, saliendo en el terminal común y conectamos el suministro (-) al terminal común. Agregando el interruptor entre el suministro (+) y la entrada, hemos completado el circuito. La corriente fluye en la dirección de la flecha cuando el interruptor se cierra.

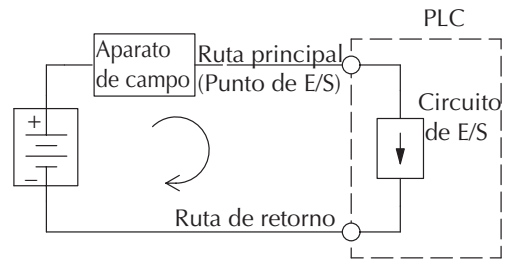


Aplicando el principio de circuito anterior a las cuatro combinaciones posibles de los tipos de entradas y salidas drenadoras y surtidoras, tenemos los cuatro circuitos mostrados abajo. Cualquier par de circuitos de entradas o salidas mostrados debajo es posible con uno de los modelos DL06.

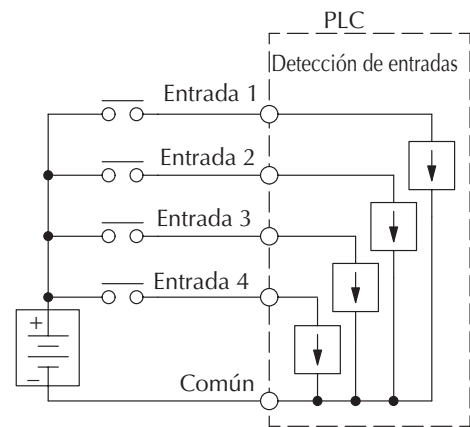


Conceptos de terminales "Comunes"

Para que opere un circuito de entradas y salidas del PLC, la corriente debe entrar en un terminal y salir en otro. Esto significa que por lo menos dos terminales se asocian con cada punto de entrada o salida. En la figura a la derecha, el terminal de entrada o salida es el camino principal para la corriente. Un terminal adicional debe proporcionar el camino de regreso a la alimentación. Éste es el terminal común.



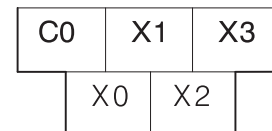
La mayoría de los puntos de entradas o salidas se agrupan en los PLCs en el camino de regreso entre dos o más entradas o salidas. La figura muestra a la derecha un grupo (o el banco) de 4 puntos de entradas que comparten un camino común de regreso. De esta manera, las cuatro entradas requieren sólo cinco terminales en vez de ocho.



Nota: En el circuito a la derecha, la corriente en el común es 4 veces cualquier corriente de entrada del canal cuando todas las entradas se energizan. Esto es especialmente importante en circuitos de salidas, donde es a veces necesario colocar un conductor de sección mayor.



La mayoría de los circuitos de entradas y salidas del PLC DL06 se agrupan en bancos que comparten un camino común de regreso. La mejor indicación del agrupamiento común está en el rótulo de alambrado en el PLC. Los grupos comunes son separados por una línea más gruesa. Una línea más delgada separa las entradas asociadas con aquel común. A la derecha, note que X0, X1, X2, y X3 comparten el común terminal C0, localizado a la izquierda de X1.



El siguiente conjunto completo de etiquetas muestra cinco bancos de cuatro entradas y cuatro bancos de cuatro salidas. Hay un común por cada banco.

G ⊕	LG	0V	Y0	Y2	C1	Y5	Y7	Y10	Y12	C3	Y15	Y17	
AC(L)	AC(N)	24V	C0	Y1	Y3	Y4	Y6	C2	Y11	Y13	Y14	Y16	N.C.

C0	X1	X3	X4	X6	C2	X11	X13	X14	X16	C4	X21	X23	N.C.
X0	X2	C1	X5	X7	X10	X12	C3	X15	X17	X20	X22	N.C.	

Este conjunto de rótulos abajo es para versiones de salidas de CC (drenadoras) tales como el D0-06DD1 y el D0-06DD1-D. Hay un común para cada grupo de cuatro salidas y un terminal designado en el lado de las salidas acepta la alimentación para las etapas de salidas.

G ⊕	LG	0V	Y0	Y2	C1	Y5	Y7	Y10	Y12	C3	Y15	Y17	
AC(L)	AC(N)	24V	C0	Y1	Y3	Y4	Y6	C2	Y11	Y13	Y14	Y16	+V

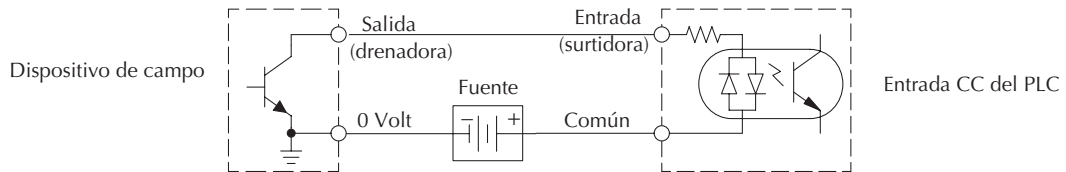
C0	X1	X3	X4	X6	C2	X11	X13	X14	X16	C4	X21	X23	N.C.
X0	X2	C1	X5	X7	X10	X12	C3	X15	X17	X20	X22	N.C.	

Conectando entradas y salidas a aparatos de campo de "estado sólido"

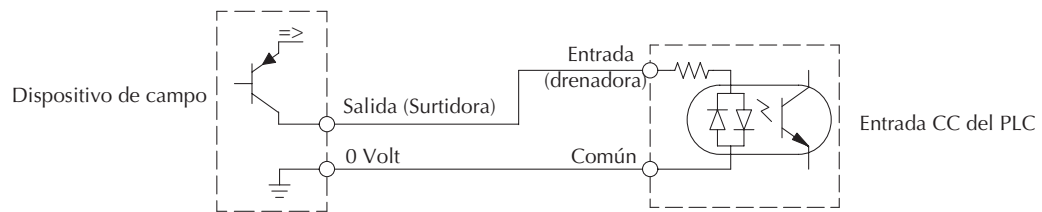
En la sección previa en conceptos de entradas y salidas drenadoras y surtidoras, discutimos circuitos de entradas y salidas C.C. que sólo permiten que la corriente fluya en un sentido. Esto es también verdad para muchos de los aparatos de campo que tienen interfases de estado sólido (transistor). En otras palabras, los aparatos de campo pueden ser también surtidores o drenadores. Cuando se conectan dos aparatos en un circuito en serie C.C. (como el caso cuando se conecta un dispositivo a una entrada o salida de un PLC C.C.), uno se debe conectar como surtidor y el otro como drenador.

Sensores de entrada de estado sólido

Las entradas del PLC DL06 son flexibles ya que detectan el flujo de corriente en cualquier dirección, así que pueden ser alambradas como o surtidoras o drenadoras. En el circuito siguiente, un aparato de campo tiene una salida de transistor "open collector" NPN. Drena la corriente del punto de entrada del PLC, que surte la corriente. La alimentación puede ser la fuente de poder auxiliar de 24VCC incluida en el PLC u otro suministro (+ 12 VCC o +24VCC), si se cumplen las especificaciones de las entradas.



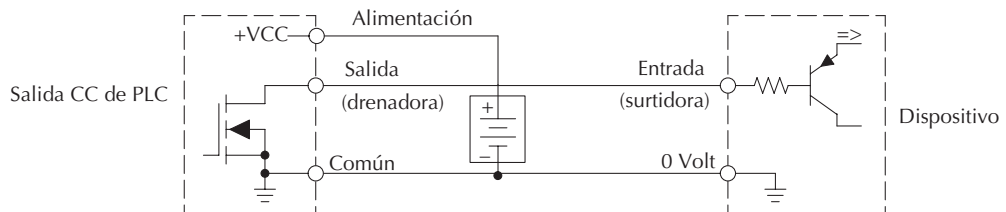
En la próxima figura, un dispositivo de campo tiene una salida de transistor PNP de emisor abierto. Surte corriente al punto de entrada del PLC, que drena corriente hasta el negativo. Ya que el dispositivo es surtidor, no se requiere ninguna alimentación adicional entre el aparato y la entrada del PLC.



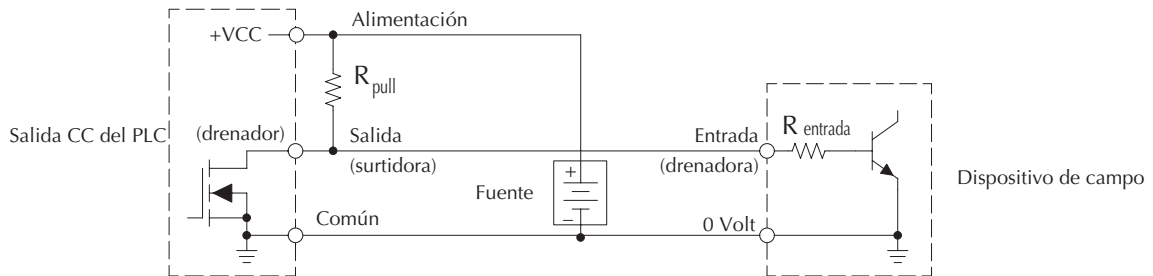
Cargas de salidas de estado sólido

A veces una aplicación requiere conectar un punto de salida del PLC a una entrada de estado sólido en un aparato. Este tipo de conexión se hace generalmente para llevar una señal de bajo nivel, no para energizar a un actuador.

La familia de PLCs DL06 C.C. ofrecen sólo salidas drenadoras o salidas surtidoras. Las dieciséis salidas tienen el mismo común eléctrico, aunque hay cuatro terminales comunes. En el circuito siguiente, el punto de salida del PLC drena corriente al común de la salida cuando energizado. Es conectado a una entrada surtidora de una entrada de un dispositivo de campo.



En el próximo ejemplo conectamos un punto de salida del PLC a la entrada drenadora de un dispositivo de campo. Esto es un poco complicado, porque la salida del PLC y la entrada del dispositivo de campo son del tipo drenadoras. Ya que el circuito debe tener un lado surtidor y un aparato drenador, le agregamos capacidad surtidora a la salida del PLC usando una resistencia pull-up. En el circuito de abajo, conectamos R_{pull} a la salida en la entrada de alimentación del circuito de salida C.C.



NOTA 1: No trate de manejar una carga grande (> 25 mA) con este método.

NOTA 2: Usando la resistencia R_{pull} para aplicar una salida surtidora tiene el efecto de invertir la lógica del punto de salida. En otras palabras, la entrada del aparato de campo se energiza cuando la salida del PLC está apagada, desde el punto de vista de la lógica ladder. Su programa debe comprender esto y debe engendrar una salida invertida. O, usted puede escoger cancelar el efecto de la inversión en otra parte, tal como en el dispositivo de campo.

Es importante escoger el valor correcto de R_{pull} . Para hacer eso, necesitamos saber la corriente nominal de entrada al dispositivo de campo (I) cuando la entrada se energiza. Si este valor no se sabe, se puede calcular como mostrado (un valor típico es 15 mA). Luego use la corriente de entrada y el voltaje de alimentación externa para calcular R_{pull} . Luego calcule la potencia de la resistencia (en Watt) P_{pull} , para dimensionar R_{pull} adecuadamente.

$V_{entrada}$ es el voltaje de la entrada cuando la señal es ON.

$$I_{entrada} = \frac{V_{entrada} \text{ (Cuando activada)}}{R_{entrada}}$$

$$R_{pull} = \frac{V_{fuente} - 0.7}{I_{entrada}} - R_{entrada} \qquad P_{pull} = \frac{V_{fuente}^2}{R_{pull}}$$

Métodos de cableado de salida a relevador

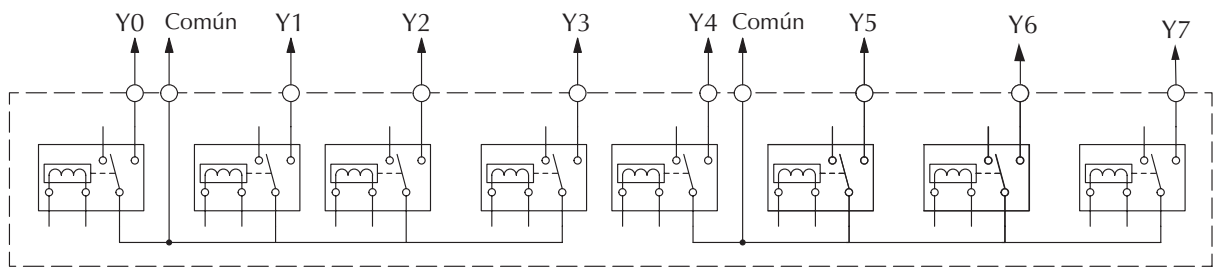
Los modelos D0-06AR y D0-06DR tienen salidas de tipo relevador. Los relevadores son mejores para las siguientes utilidades:

- Cargas que requieren corrientes más altas que lo que las salidas de estado sólido DL06 pueden entregar
- Usos sensitivos a costos
- Algunos canales de salida necesitan aislación de otras salidas (por ejemplo, cuando algunas cargas requieren C.A. mientras que otras requieren C.C.)

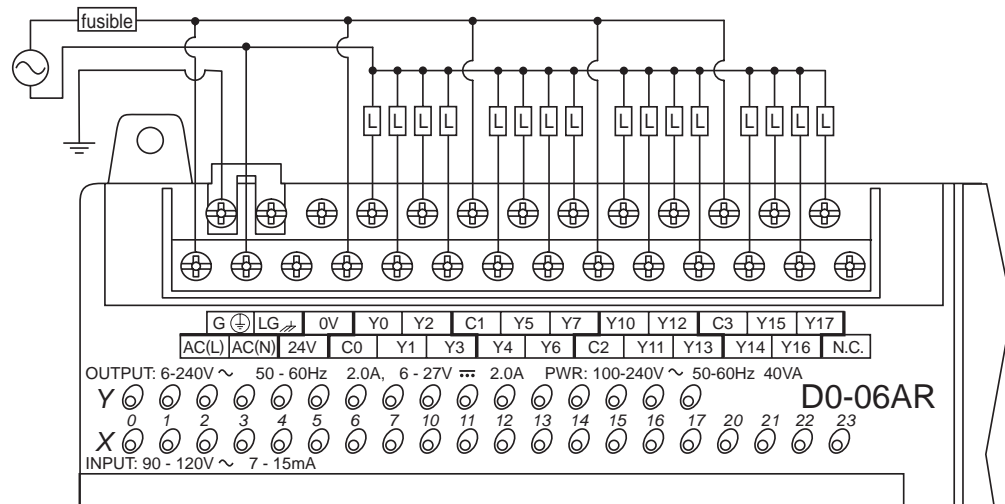
Algunos usos en los cuales NO debe usar relevadores:

- Cargas que requieren corrientes bajo 10 mA
- Cargas que se deben conmutar a alta velocidad

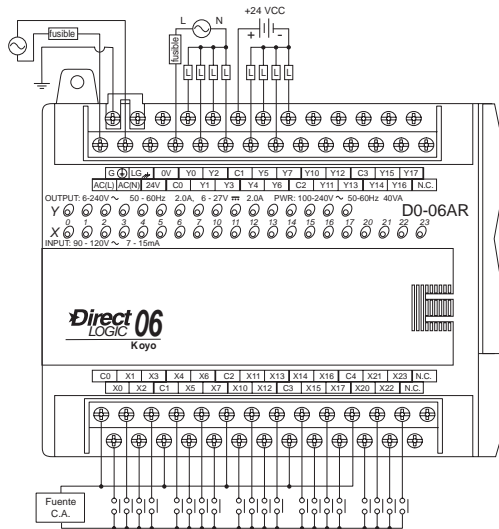
Esta sección presenta varias maneras de conectar las salidas de un relevador a las cargas. La salidas a relevador del DL06 tienen dieciséis contactos normalmente abiertos tipo SPST disponibles. Se organizan con cuatro relevadores por común. La figura de abajo muestra los relevadores y el cableado interno del PLC. Note que cada grupo está aislado del otro grupo de salidas.



En el circuito de abajo, todas las cargas utilizan la misma fuente CA que acciona el PLC DL06. En este ejemplo, todos los comunes está conectados juntos.



En el circuito en la página siguiente, las cargas en Y0-Y3 usan la misma fuente que acciona el DL06. las cargas Y4 - Y7 usan una fuente CC separada. En este ejemplo, los comunes está conectados separados de acuerdo a qué fuente de poder alimenta la carga asociada.



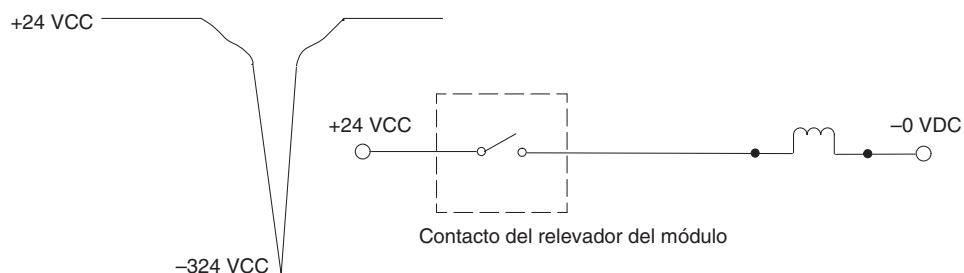
Supresión de transientes de tensión para cargas inductivas

Los aparatos de carga inductivos (aparatos con una bobina) engendran voltajes transitorios cuando se energiza con un contacto de relevador. Cuando un contacto de relevador se cierra, "rebota", lo que energiza y desenergiza la bobina hasta que pare de "rebotar". Se generan así voltajes transitorios que son mucho más grandes que la amplitud de la tensión de alimentación, especialmente con una tensión de alimentación de C.C. Cuando se abre una carga inductiva alimentada por corriente continua, la tensión de alimentación está siempre presente cuando el contacto del relevador se abre (o "rebota"). Cuando se abre una carga inductiva suministrada por C.A., como una bobina de un contactor, si el voltaje no es cero cuando el contacto del relevador se abre, la energía almacenada en el inductor es liberada cuando el voltaje al inductor es quitado repentinamente. Esta liberación de energía es la causa de los voltajes transitorios.

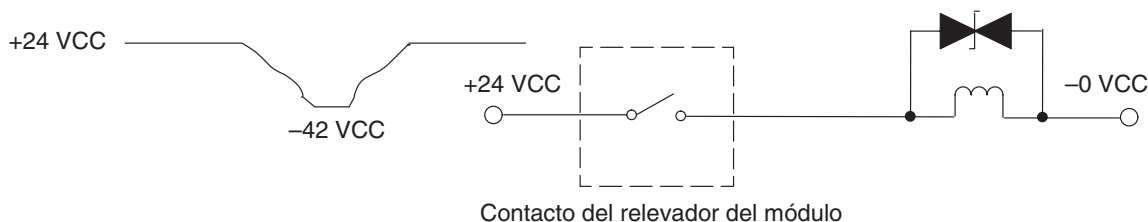
Cuando se controlan aparatos inductivos (motores, arrancadores de motores, solenoides, válvulas, etc.) con contactos de relevador, es necesario que sea conectado un aparato de supresión de sobre tensión directamente en paralelo con la bobina del aparato de campo. Si el aparato inductivo tiene conectores enchufables, el aparato de supresión de sobre tensión se puede instalar en el bloque terminal de la salida del relevador. **Esto es causa de fallas en el PLC u otros aparatos como interfaces de operador.**

Los supresores transitorios de voltaje (TVS o transorb) suministran la mejor supresión transitoria de bobinas de C.A. y CC, y tienen la respuesta más rápida con la sobre tensión más pequeña.

Varistores de óxido de metal (MOV) proporcionan la mejor próxima supresión transitoria de bobinas de C.A. y C.C. Por ejemplo, la forma de ondas en la figura debajo muestra la energía liberada cuando se abre un contacto que alimenta un solenoide de 24 VCC. Note el gran pico de voltaje.



++Esta figura muestra el mismo circuito con un transorb (TV) en paralelo con la bobina. Note que el pico de voltaje se ha reducido perceptiblemente.



2

Use la siguiente tabla para ayudarlo a seleccionar un supresor como un TVS o MOV para la aplicación de acuerdo al voltaje.

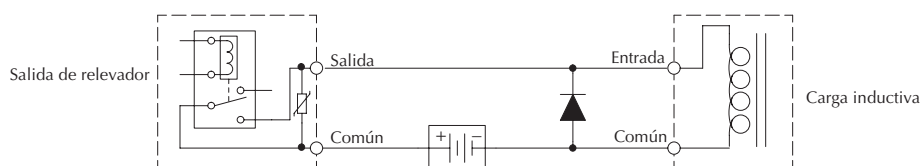
Supresores de sobretensión			
Vendedor / Catálogo	Tipo	Voltaje de la carga	Número de artículo
Automationdirect	TVS	110/120 VCA	ZL-TD8-120
	TVS	24 VCC	ZL-TD8-24
Supresores de sobrevoltage, LiteOn Diodes; del catálogo DigiKey; Teléfono: 1-800-344-4539	TVS	220/240 VCC	P6KE350CA
	TVS	12/24 VCC or VCA	P6K30CAGICT-ND
	Diode	12/24 VCC or VCA	1N4004CT-ND
Harris Metal Oxide Varistors; del catálogo Newark; Teléfono 1-800-463-9275	MOV	110/120 VCA	V150LA20C
	MOV	220/240 VCA	V250LA20C

Prolongando la vida de los contactos de un relé

Los contactos de un relé se gastan de acuerdo a la frecuencia de operación, cantidad de arcos creados en el momento de abrir o cerrar el contacto y la presencia de contaminantes atmosféricos. Hay algunas acciones para prolongar la vida de los contactos del relé, tales como encender o apagar el relé solamente cuando sea necesario y si es posible, apagando y encendiendo la carga en un momento en que haya la mínima corriente por el contacto. También, elimine voltajes de cargas inductivas de C.C. tales como contactores y solenoides.

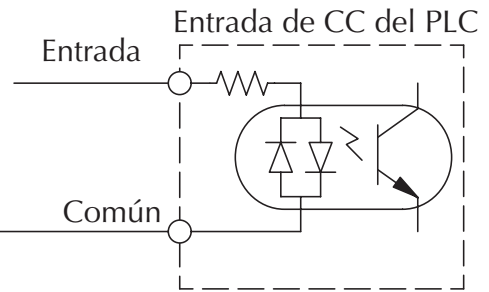
Para cargas inductivas en circuitos de C.C. recomendamos el uso de un diodo de supresión según lo mostrado en el diagrama siguiente (NO USE este circuito con una alimentación C.A.). Cuando se energiza la carga el diodo es polarizado inversamente (alta impedancia). Cuando la carga se desconecta, la energía almacenada en la bobina se libera en forma de un pico de voltaje negativo. En este momento el diodo es polarizado directamente (impedancia baja) y la corriente fluye a 0 Volt. Esto protege los contactos del relé contra el arco de alto voltaje que ocurriría cuando los contactos se están abriendo.

Coloque el diodo tan cerca como sea posible del dispositivo inductivo. Use un diodo con un voltaje inverso (PIV) de por lo menos 100 V, 3A o más grande. Utilice un tipo de recuperación rápida (tal como un tipo Schottky). No use un diodo de señal tal como 1N914, 1N941, etc. Asegúrese que el diodo está conectado correctamente antes de la operación. Si está instalado al revés, se cortocircuita la fuente cuando se energiza el relé.

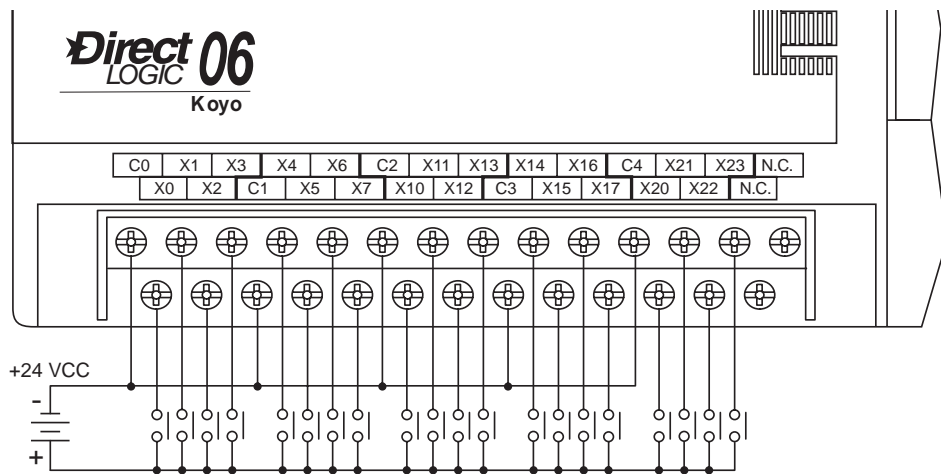


Métodos de cableado de entradas C.C.

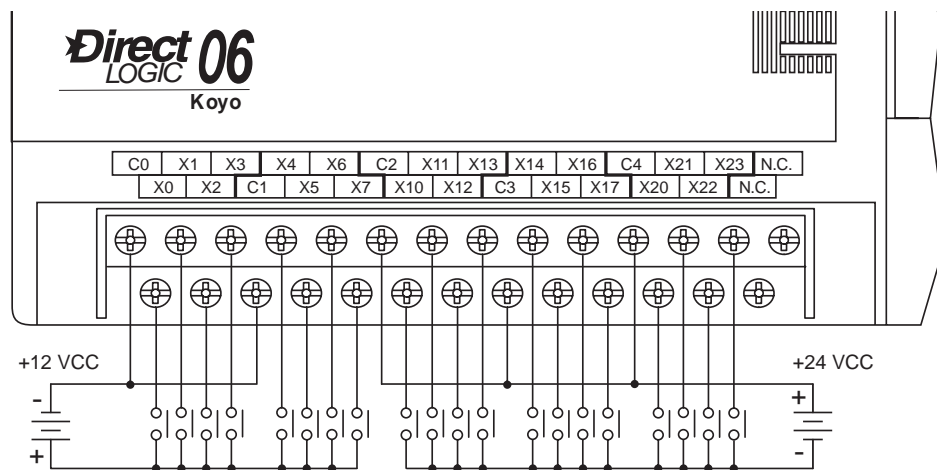
Los PLC DL06 con entradas C.C. son particularmente flexibles porque ellas pueden ser conectadas como drenadoras o surtidoras. Los diodos duales (demostrados la derecha) permiten 10.8 - 26.4 VCC. Se pretende usar en el rango de +12 VCC y +24 VCC. Se puede conectar cada grupo de entradas asociadas con el común como drenadoras y la otra mitad como surtidoras. Las entradas agrupadas por un común deben ser todas drenadoras o todas surtidoras.



En el primer y más simple ejemplo abajo, todos los comunes están conectados y todas las entradas son drenadoras.



En el ejemplo siguiente, las primeras ocho entradas son drenadoras y las últimas doce son surtidoras.

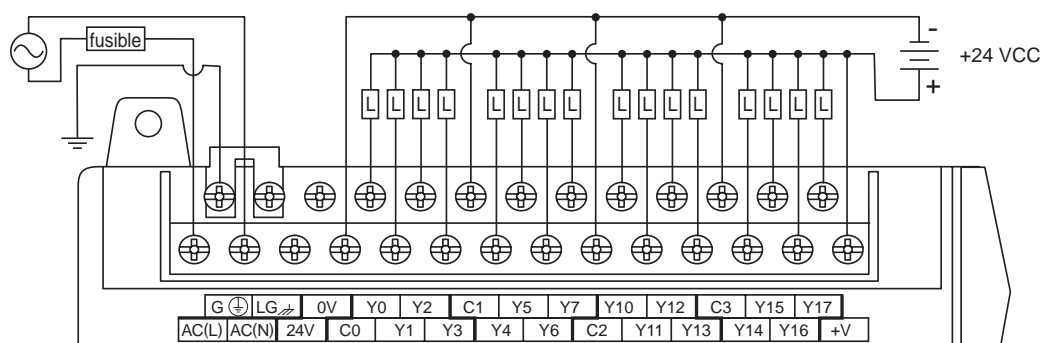


Métodos de cableado de salidas C.C.

Los circuitos de salida C.C. DL06 son transistores de alto rendimiento con resistencia baja en el estado ON y tiempos rápidos de conmutación. Observe por favor las características siguientes que son únicas en los modelos de salidas C.C.:

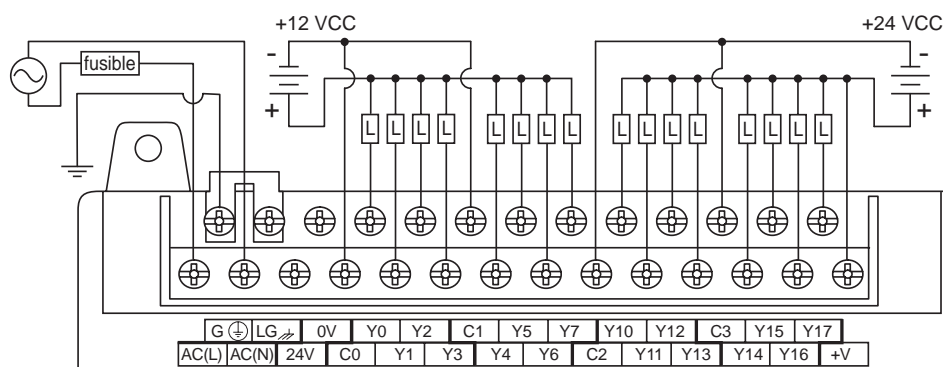
- Hay solamente un común eléctrico para las dieciséis salidas. Las dieciséis salidas pertenecen a un banco.
- Los transistores de salida son drenadoras o surtidoras solamente. Vea las especificaciones detalladas en este manual para determinar que tipo de salidas están presentes en un modelo específico.
- El circuito de salida dentro del PLC requiere energía externa. La fuente (-) debe estar conectada a un terminal común y la fuente (+) se conecta al terminal de la extrema derecha en el terminal superior (+V).

En el ejemplo abajo, las dieciséis salidas comparten una fuente común.



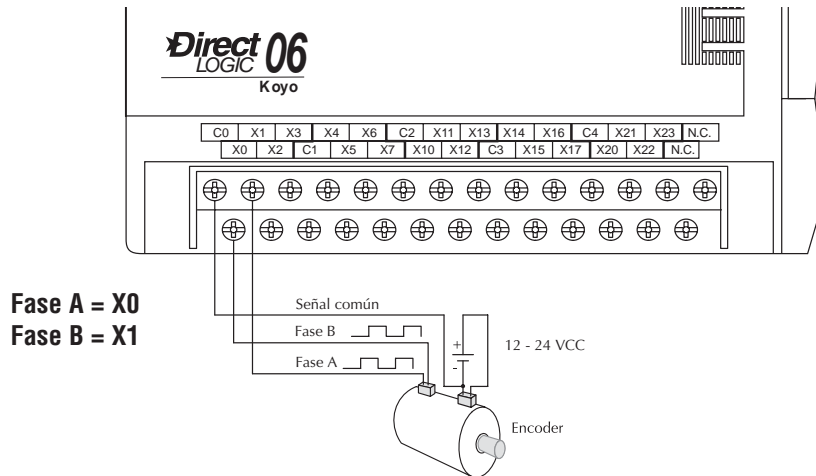
En el ejemplo siguiente de abajo, las salidas tienen fuentes "diferentes". Las primeras ocho salidas están utilizando +12 VCC, y las últimas ocho están utilizando una fuente +24 VCC. Sin embargo, usted puede dividir las salidas en cualquier número de fuentes, mientras:

- todos los voltajes de las fuentes estén dentro del rango especificado.
- todos los puntos de salida se conectan como drenadoras.
- todos los terminales de las fuente (-) están conectados juntos.

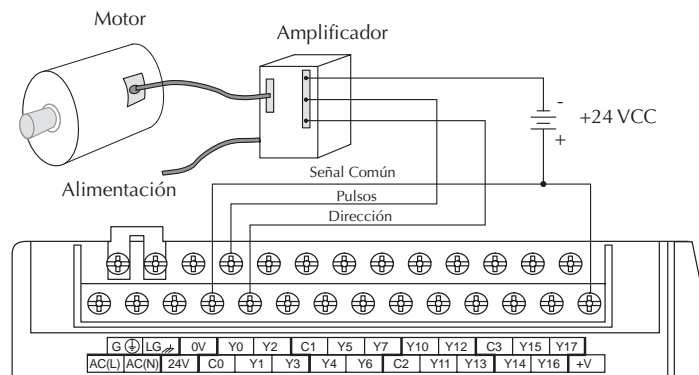


Métodos de cableado de E/S de alta velocidad (HSIO)

Las versiones DL06 con entradas o salidas C.C. contienen un circuito de alta velocidad dedicado de E/S (HSIO). La configuración de circuito es programable y procesa puntos específicos de E/S independientemente del barrido de la CPU. El apéndice E discute las opciones de programación para HSIO. Aunque que el circuito HSIO tiene seis modos, mostramos diagramas eléctricos para dos de los modos más populares de este capítulo. Las entradas de alta velocidad se conectan a X0 - X3. Configurado correctamente, el DL06 puede contar pulsos en cuadratura hasta 7 kilociclos de un encoder incremental según lo mostrado abajo.



Las versiones DL06 con el tipo de salidas CC pueden utilizar la característica de alta velocidad de salidas de un tren de pulsos. Puede generar pulsos de alta velocidad hasta 10 kilociclos para un control especializado tal como motor stepper o paso a paso o sistemas de accionamiento inteligentes. Las salidas Y0 y Y1 pueden generar señales de pulso y de dirección, o puede generar señales CW y CCW respectivamente. Vea el apéndice E para más opciones de entradas y de salidas de pulsos de alta velocidad.



Glosario de términos de la especificación y del PLC

Entradas discretas

Una de las veinte conexiones de un dispositivo de campo que el PLC convierte a una señal eléctrica en un estado binario (OFF u ON), que es leído por la CPU en cada barrido del PLC.

Salidas discretas

Una de las dieciséis conexiones de salida del PLC que convierte un resultado interno del programa (ON u OFF) para activar un dispositivo de salida. Esto permite que el programa active grandes cargas.

Común de E/S

Una conexión en los terminales de entrada o de salida que es compartida por múltiple circuitos. Es generalmente en la trayectoria de retorno de la fuente de poder del circuito de E/S.

Rango de voltaje de entrada

El rango de voltajes de operación permitido en el circuito de entrada.

Voltaje máximo de entrada

El máximo voltaje permitido en el circuito de entrada.

Nivel de voltaje ON

El nivel mínimo de voltaje en el cual la entrada se hará ON o activada.

Nivel de voltaje OFF

El nivel máximo de voltaje en el cual la entrada se hará OFF o desactivada.

Impedancia de entrada

Impedancia de entrada puede ser usada para calcular la corriente para un voltaje de funcionamiento particular.

Corriente de entrada

Corriente de funcionamiento típica para una entrada activa (ENCENDIDA).

Mínima corriente ON

Corriente mínima para que la entrada tenga un funcionamiento confiable al estar ENCENDIDA.

Máxima corriente OFF

La corriente máxima para que la entrada tenga un funcionamiento confiable al estar APAGADA.

Respuesta de OFF a ON

El tiempo que el módulo requiere en la transición del estado APAGADO a ENCENDIDO.

Respuesta de On a OFF

El tiempo que el módulo requiere en la transición del estado ENCENDIDO A APAGADO.

Indicadores de estado

Los LED que indican el estado de un punto de entrada o de salida. Todos los LED en el PLC DL06 están eléctricamente en el Lado Lógico del circuito de entrada o de salida.

Programa ladder (o de escalera)

El programa en el PLC con lógica booleana que simula un circuito eléctrico (ladder es la palabra en inglés)

Renglón en el programa ladder

Es cada una de las líneas de lógica en un programa ladder.

Diagramas eléctricos y especificaciones

El resto de este capítulo entrega información técnica detallada de los PLCs DL06. Se presenta para cada PLC un diagrama eléctrico básico, los circuitos equivalentes de E/S y tablas de especificaciones.

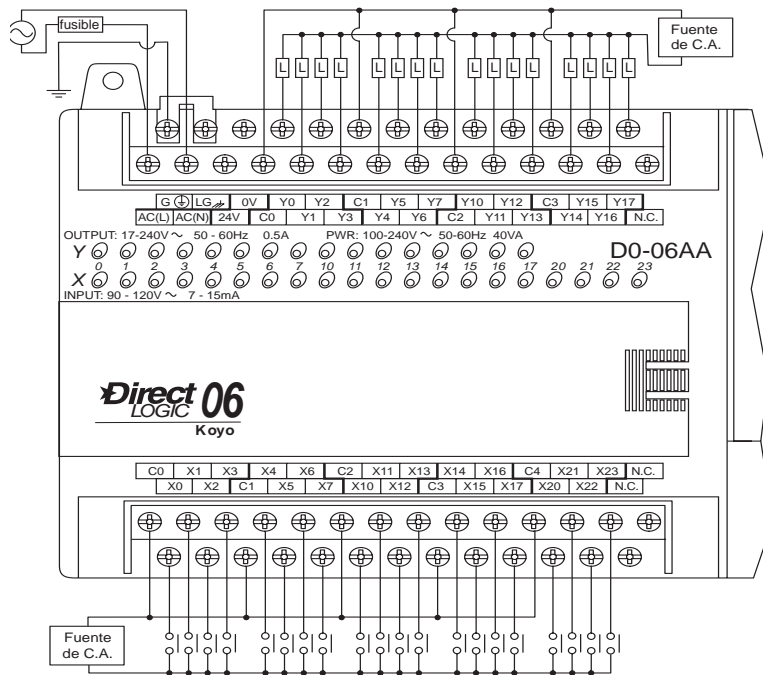
2

Diagrama eléctrico de E/S del D0-06AA

El PLC de D0-06AA tiene veinte entradas C.A. y dieciséis salidas C.A. El diagrama siguiente muestra un ejemplo típico de cableado. La conexión de alimentación CA utiliza cuatro terminales según lo mostrado.

Las entradas se organizan en cinco bancos de cuatro puntos cada uno. Cada banco tiene un terminal común. El ejemplo del cableado abajo muestra todos los comunes conectados juntos, pero se pueden usar fuentes y circuitos comunes separados. El circuito de entrada equivalente muestra un canal de un banco típico.

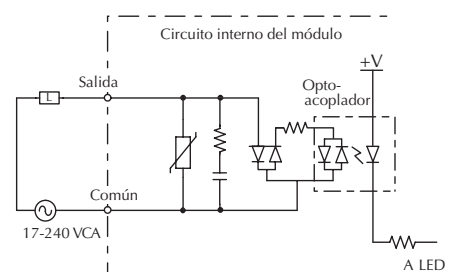
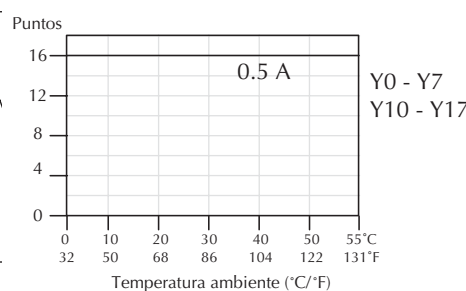
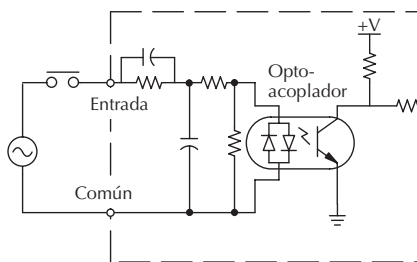
Las salidas se organizan en cuatro bancos de cuatro conmutadores triac. Cada banco tiene un terminal común.



Circuito de entrada equivalente

Curva de degradación de salidas CA.

Circuito de salida equivalente



Especificaciones generales D0-06AA	
Requerimientos de alimentación	100– 240 VCA, 40 VA máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K–Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K–Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence / print, ASCII
Tipo de cable de programación	D2–DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	–4 a 158° F (–20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3–304
Tipo de terminal	Removible
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CA	
Rango de voltaje en las entradas (Min. - Max.)	80 – 132 VCA, 47 - 63 Hz
Voltaje de operación	90 – 120 VCA, 47 - 63 Hz
Corriente en las entradas	8 mA @100 VCA a 50 Hz 10 mA @100 VCA a 60 Hz
Max. corriente en las entradas	12 mA @132 VCA a 50 Hz 15 mA @132 VCA a 60 Hz
Impedancia de entrada	14KΩ @50 Hz, 12KΩ @60Hz
Voltaje y corriente cuando ON	> 6 mA @ 75 VCA
Voltaje y corriente cuando OFF	< 2 mA @ 20 VCA
Tiempo de respuesta cuando va de OFF a ON	< 40 ms
Tiempo de respuesta cuando va de ON a OFF	< 40 ms
Indicadores de estado	Lado lógico
Comunes	4 canales/común x 5 bancos (aislados)

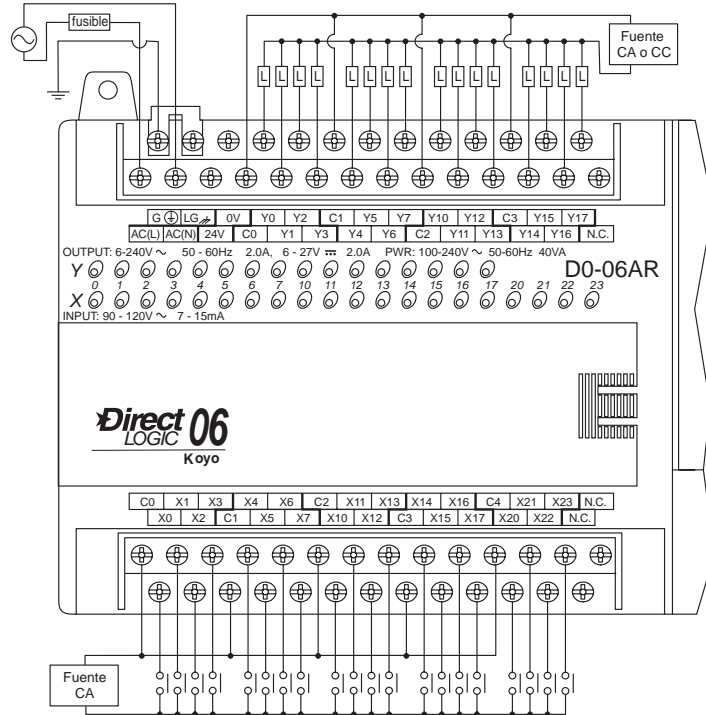
Especificaciones de salidas CA	
Rango de voltaje de salida (Min. - Max.)	15 – 264 VCA, 47 – 63 Hz
Voltaje de operación	17 – 240 VCA, 47 – 63 Hz
Caída de tensión cuando la señal es ON	1,5 VCA (>50mA) 4,0 VCA (<50mA)
Corriente máxima	0,5 A / punto, 1,5 A / común
Máxima corriente de fuga	<4 mA @ 264 VCA
Máxima corriente de inrush	10 A for 10 ms
Corriente mínima por el contacto	10 mA
Tiempo de respuesta cuando va de OFF a ON	1 ms
Tiempo de respuesta cuando va de ON a OFF	1 ms +1/2 ciclo
Indicadores de estado	Lado lógico
Comunes	4 canales / común x 4 bancos (aislados)
Fusibles	Ninguno (se recomienda colocar fusibles externos)

Diagrama eléctrico de E/S del PLC D0-06AR

El PLC de D0-06AR tiene veinte entradas C.A. y dieciséis salidas de contactos de relevador. El diagrama siguiente demuestra un ejemplo típico del cableado de campo. La conexión de alimentación C.A. utiliza cuatro terminales en la izquierda como mostrado.

Los veinte canales de entradas CA utilizan los terminales en la parte inferior. Las entradas se organizan en cinco bancos de cuatro punots cada uno. Cada banco tiene un terminal común.El ejemplo del cableado abajo muestra todo los comunes conectados juntos, pero se pueden utilizar fuentes separadas en cada común. El circuito de entrada equivalente muestra un canal de un banco típico.

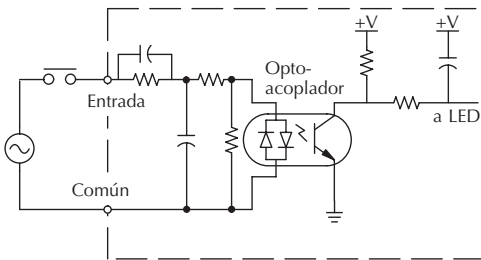
2



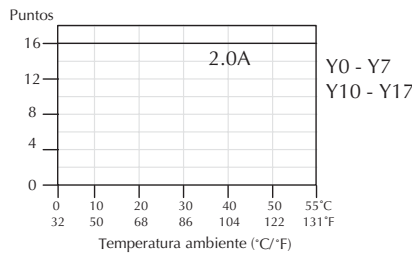
Vida típica del contacto de un relevador a temperatura ambiente

Voltaje & tipo de carga	Corriente	
	A 1A	A 2A
24 VCC resistiva	500K	250K
24 VCC inductiva	100K	50K
110 VCA resistiva	500K	250K
110 VCA inductiva	200K	100K
220 VCA resistiva	350K	200K
220 VCA inductiva	100K	50K

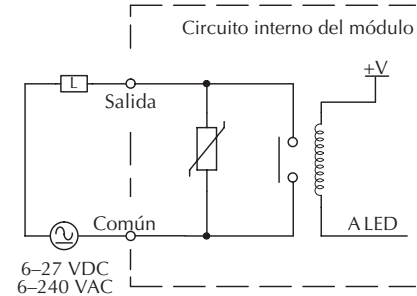
Circuito de entrada equivalente



Curva de degradación de salidas de contacto



Circuito de salida equivalente



Los dieciséis canales de salida del relevador utilizan los terminales en el bloque superior. Las salidas se organizan en cuatro bancos de cuatro contactos normalmente abiertos de relevador. Cada banco tiene un terminal común. El ejemplo del cableado en la página pasada muestra todo los comunes conectados juntos, pero pueden ser usadas fuentes separadas en cada circuito común. El circuito de salida equivalente muestra un canal de un banco típico. Los contactos de relevadores pueden conmutar voltajes CA o C.C.

Especificaciones generales del PLC D0-06AR	
Requerimientos de alimentación	100 – 240 VCA, 40 VA máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/esclavo), MODBUS (Maestro/esclavo), Non-sequence / print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removible
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CA X0-X23	
Rango de voltaje en las entradas (Min. - Max.)	80 – 132 VCA, 47 - 63 Hz
Rango del voltaje de operación	90 – 120 VCA, 47 -63 Hz
Corriente en las entradas	8 mA @ 100 VCA at 50 Hz 10 mA @ 100 VCA at 60 Hz
Max. corriente en las entradas	12 mA @ 132 VCA at 50 Hz 15 mA @ 132 VCA at 60 Hz
Impedancia de entradas	14KΩ @50 Hz, 12KΩ @60 Hz
Voltaje y corriente cuando ON	>6 mA @ 75 VCA
Voltaje y corriente cuando OFF	<2 mA @ 20 VCA
Tiempo de respuesta cuando va de OFF a ON	< 40 ms
Tiempo de respuesta cuando va de ON a OFF	< 40 ms
Indicadores LED de estado	Lado lógico
Comunes	4 canales/común x 5 bancos

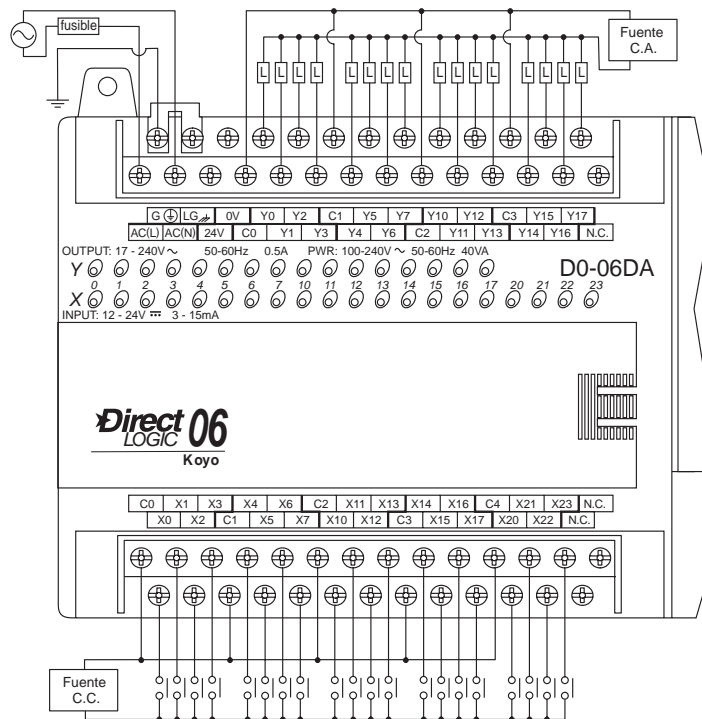
Especificaciones de salidas a relevador Y0-Y17	
Rango de voltaje de salida	(Min. – Max.) 5 – 264 VCA (47 -63 Hz), 5 – 30 VCC
Rango del voltaje de operación	6 – 240 VCA (47 -63 Hz), 6 – 27 VCC
Corriente de salida	2A / punto, 6A / común
Máxima corriente de fuga	0,1 mA @264VAC
Carga mínima recomendada	5 mA @5 VCC
Tiempo de respuesta cuando va de OFF a ON	< 15 ms
Tiempo de respuesta cuando va de ON a OFF	< 10 ms
Indicadores LED de estado	Lado lógico
Comunes	4 canales / común x 4 bancos
Fusibles	Ninguno (se recomienda colocar fusibles externos)

Diagrama de cableado de E/S del D0-06DA

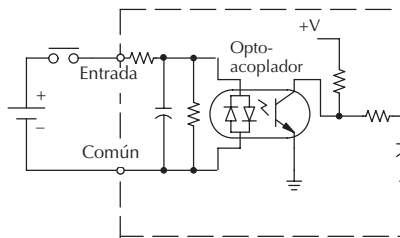
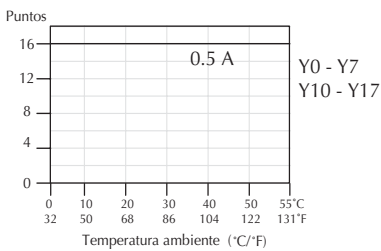
El PLC de D0-06DA tiene veinte entradas C.C. y dieciséis salidas C.A. El diagrama siguiente demuestra un ejemplo típico del cableado de campo. La conexión de energía externa C.A. utiliza cuatro terminales en la izquierda como mostrado.

Los veinte canales de entradas C.C. utilizan los terminales en la parte inferior. Las entradas se organizan en cinco bancos de cuatro y pueden ser conectadas como surtidoras o drenadoras. El ejemplo del cableado abajo muestra todo los comunes conectados juntos, pero se pueden utilizar fuentes separadas en cada común. El circuito de entrada equivalente es mostrado abajo y el circuito de alta velocidad es mostrado a la izquierda.

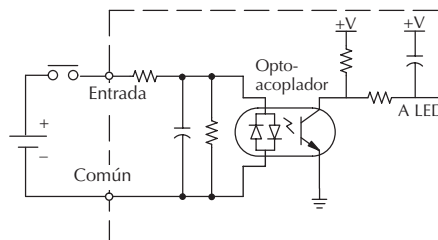
Las salidas son organizadas en cuatro bancos de 4 triacs cada uno. Cada banco tiene un común. El ejemplo del cableado abajo muestra todos los comunes conectados juntos, pero se pueden utilizar fuentes separadas en cada común.



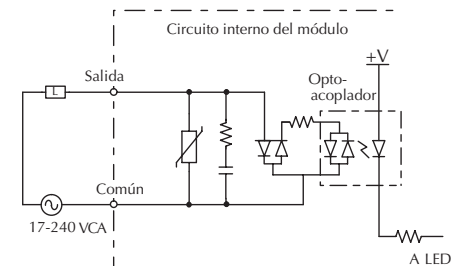
Curva de degradación para salidas CA



Entradas HSIO (X0-X3)



Entradas normales (X4-X23)



Circuito de salida equivalente

Especificaciones generales DO-06DA	
Requerimientos de alimentación	100 – 240 VCA, 40 VA máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/esclavo), MODBUS (Maestro/esclavo), Non-sequence/print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removible
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales CC X4 – X23
Rango de voltaje en las entradas	10.8 – 26.4 VCC	10.8 – 26.4 VCC
Voltaje de operación	12 – 24 VCC	12 – 24 VCC
Voltaje máximo	30 VCC (máxima frecuencia 7 kHz)	30 VCC
Ancho de pulso mínimo	70 µS	N/A
Voltaje de detección de estado ON	> 10 VCC	> 10 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Impedancia de entrada	1.8 kΩ @ 12 – 24 VCC	2.8 kΩ @ 12 – 24 VCC
Corriente mínima en estado ON	>5 mA	>4 mA
Corriente máxima en estado OFF	< 0,5 mA	<0.5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	<70 µs	2 – 8 ms, 4 ms típico
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 5 bancos	

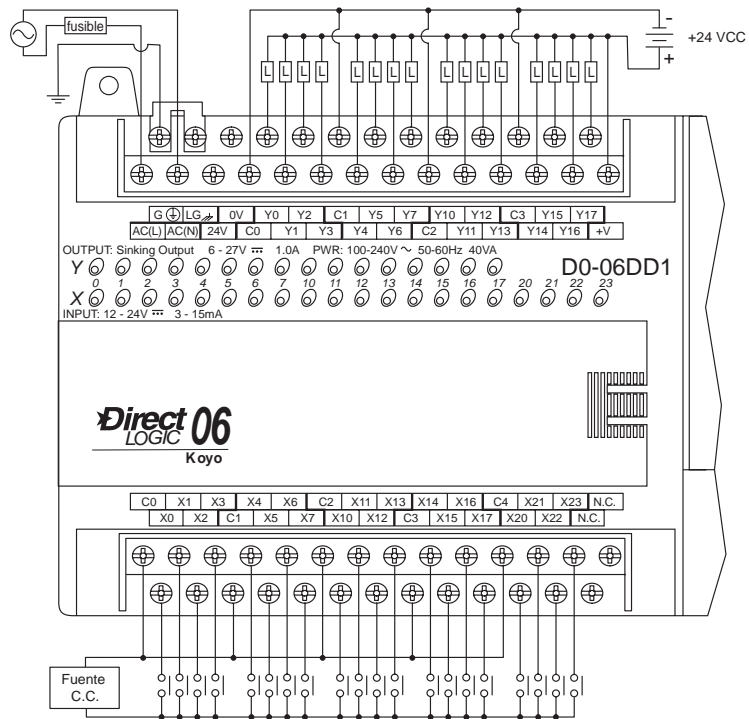
Especificaciones de salidas CA	
Rango de voltaje de salida (Min. - Max.)	15 – 264 VCA, 47 – 63 Hz
Voltaje de operación	17 – 240 VCA, 47 – 63 Hz
Caída de tensión cuando la señal es ON	1.5 VCA @ > 50mA, 4 VCA @ < 50mA
Corriente máxima	0,5 A / punto, 1,5 A / común
Máxima corriente de fuga	< 4 mA @ 264 VCA, 60Hz
Máxima corriente de inrush	10 A por 10 ms
Carga mínima	10 mA
Tiempo de respuesta cuando va de OFF a ON	1 ms
Tiempo de respuesta cuando va de ON a OFF	1 ms +1/2 ciclo
Indicadores LED de estado	Lado lógico
Comunes	4 canales / común x 4 bancos (aislados)
Fusibles	Ninguno (se recomienda colocar fusibles externos)

Diagrama eléctrico de E/S del D0-06DD1

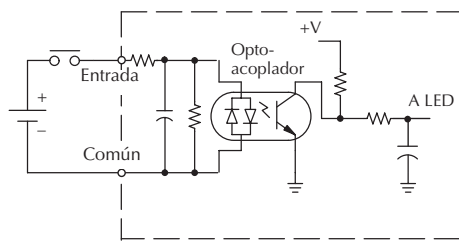
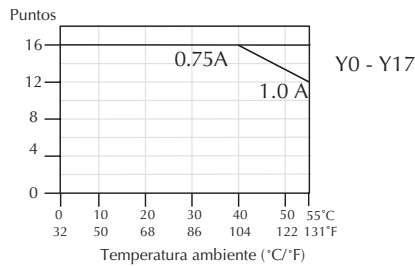
El PLC D0-06DD1 tiene veinte entradas de C.C. drenadoras y surtidoras y dieciséis salidas de C.C. drenadoras. El diagrama siguiente demuestra un ejemplo típico del cableado del campo. La conexión de alimentación C.A. utiliza cuatro terminales según lo mostrado.

Las entradas se organizan en cinco bancos de cuatro puntos. Cada banco tiene un terminal común y se puede conectar como entradas drenadoras o surtidoras. El ejemplo del cableado abajo muestra los comunes conectados, pero se pueden usar fuentes separadas en cada común.

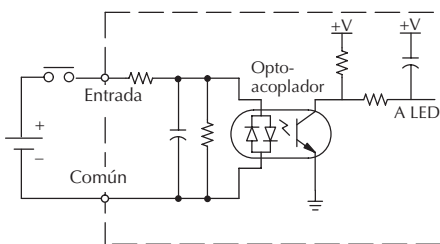
Observe los requisitos para energía externa.



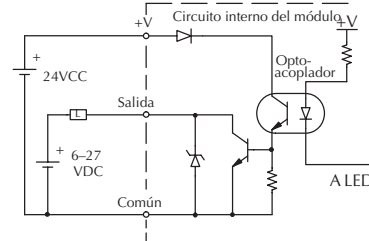
Curva de degradación de salidas CC



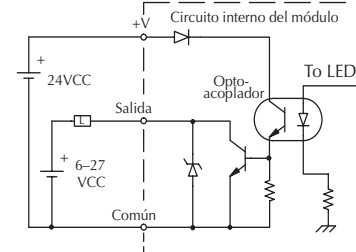
Entradas HSIO(X0-X3)



Salidas normales CC (X4-X23)



Salidas HSIO (Y0-Y1)



Salidas normales CC (Y2-Y17)

Especificaciones generales D0-06DD1		
Requerimientos de alimentación	100 – 240 VCA, 40 VA máximo,	
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)	
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence / print, ASCII	
Tipo de cable de programación	D2-DSCBL	
Temperatura de operación	32 a 131° F (0 a 55 C)	
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)	
Humedad relativa	5 a 95% (sin condensación)	
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos	
Vibración	MIL STD 810C 514.2	
Choque	MIL STD 810C 516.2	
Inmunidad al ruido	NEMA ICS3-304	
Tipo de terminal	Removable	
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo	
Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales CC X4 – X23
Min. - Max. Voltage Range	10,8 – 26,4 VCC	10,8 – 26,4 VCC
Voltaje de operación Range	12 – 24 VCC	12 – 24 VCC
Voltaje de cresta	30 VCC (máxima frecuencia 7 kHz)	30 VCC
Ancho de pulso mínimo	100 µs	N/A
Voltaje de detección de estado ON	> 10,0 VCC	> 10,0 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Max. Corriente en las entradas	6 mA @12VCC, 13mA @24VCC	4 mA @12VCC, 8,5mA @24VCC
Impedancia de entradas	1.8 Ωk @ 12 – 24 VCC	2.8 Ωk @ 12 – 24 VCC
Corriente mínima en estado ON	>5 mA	>4 mA
Corriente máxima en estado OFF	< 0,5 mA	<0,5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	<70 µs	2 – 8 ms, 4 ms típico
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 5 bancos no aislados	
Especificaciones de salidas CC		
Parámetro	Salidas de pulsos Y0 – Y1	Salidas normales Y2 – Y17
Voltaje mínimo y máximo	5 – 30 VCC	5 – 30 VCC
Voltaje de operación	6 – 27 VCC	6 – 27 VCC
Voltaje de cresta	< 50 VCC (10 kHz max. frequency)	< 50 VCC
Caída de tensión cuando es ON	0,3 VCC @ 1 A	0,3 VCC @ 1 A
Corriente máxima(resistiva)	0,5 A/pt., 1A / pto.como salida normal.	1,0 A / punto
Máxima corriente de fuga	15 µA @ 30 VCC	15 µA @ 30 VCC
Máxima corriente de inrush	2 A por 100 ms	2 A por 100 ms
Alimentación externa CC	20 - 28 VCC; 150mA máximo	20 - 28 VCC Max 280 mA (Aux. 24VCC alimenta terminal V+
Respuesta cuando va de OFF a ON	< 10µ s	< 10 µs
Respuesta cuando va de ON a OFF	< 20 µs	< 60 µs
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 4 bancos no aislados	
Fusibles	Ninguno (se recomienda colocar fusibles externos)	

Diagrama eléctrico de E/S del D0-06DD2

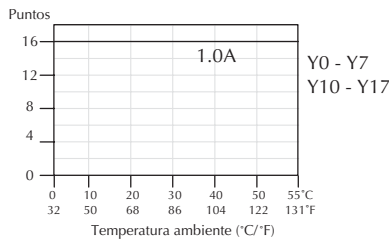
El PLC D0-06DD2 tiene veinte entradas C.C. drenadoras y surtidoras y dieciséis salidas C.C. el surtidoras. El diagrama siguiente muestra un ejemplo típico de cableado. La conexión de alimentación C.A. utiliza cuatro terminales según lo mostrado.

Las entradas se organizan en cuatro bancos de cuatro puntos cada uno. Cada banco tiene un terminal común aislado y se puede conectar como entradas drenadoras o surtidoras. El ejemplo de cableado abajo muestra todos los comunes conectados juntos, pero se pueden utilizar fuentes y comunes separados.

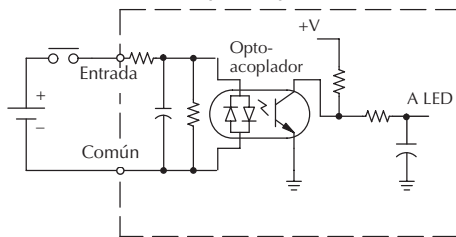
Todas las salidas comparten el mismo común. Observe el requisito para la energía externa.

2

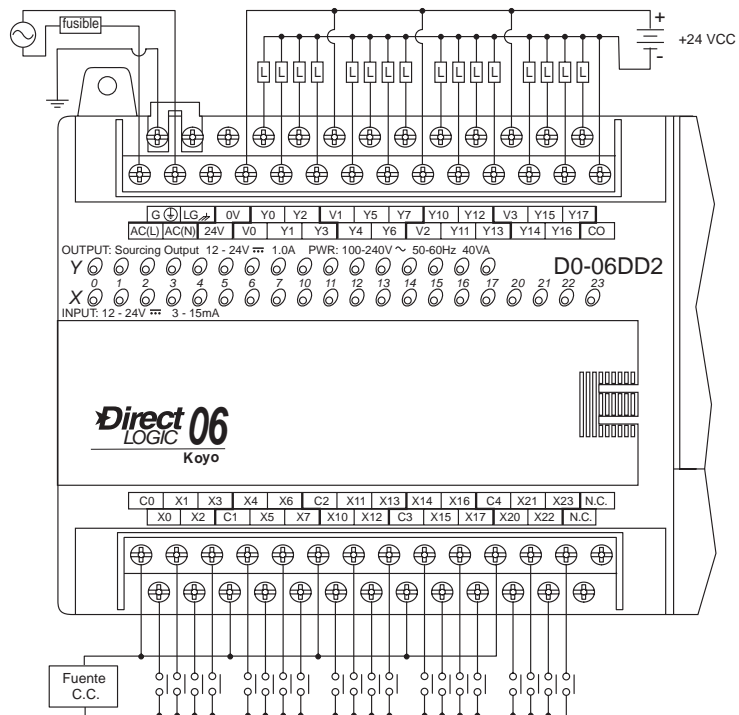
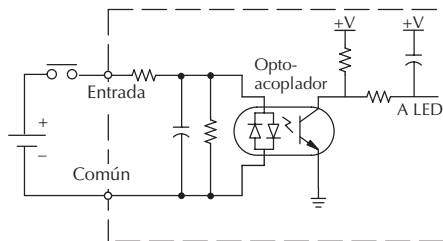
Curva de degradación de salidas CC



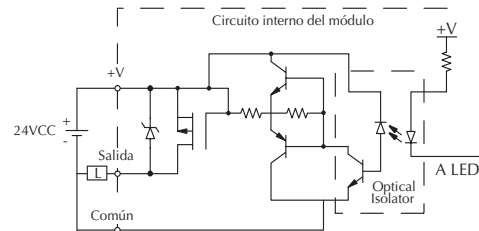
Entradas HSIO (X0-X3)



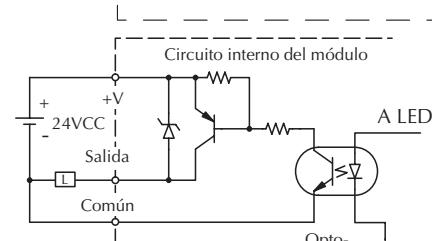
Entradas normales CC (X4-X23)



Salidas HSIO (Y0-Y1)



Salidas normales CC (Y2-Y17)



Especificaciones generales D0-06DD2	
Requerimientos de alimentación	100 – 240 VCA, 40 VA máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence/print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removable
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales CC X4 – X23
Rango de voltaje (Min. - Max.)	10.8 – 26.4 VCC	10.8 – 26.4 VCC
Rango del voltaje de operación	12 – 24 VCC	12 – 24 VCC
Voltaje de cresta	30 VCC (7 kHz máximo frequency)	30 VCC
Ancho de pulso mínimo	70 µs	N/A
Voltaje de detección de estado ON	> 10,0 VCC	> 10,0 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Max. Corriente en las entradas	6mA @12VCC, 13mA @24VCC	4mA @12VCC, 8.5mA @24VCC
Impedancia de entradas	1.8 Ωk @ 12 – 24 VCC	2.8 Ωk @ 12 – 24 VCC
Corriente mínima en estado ON	>5 mA	>4 mA
Corriente máxima en estado OFF	< 0,5 mA	<0,5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	<70 µs	2 – 8 ms, 4 ms típico
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 5 bancos no aislados	

Especificaciones de salidas CC		
Parámetro	Pulse Outputs Y0 – Y1	Standard Outputs Y2 – Y17
Rango de voltaje (Min. - Max.)	10.8 -26.4 VCC	10.8 -26.4 VCC
Rango del voltaje de operación	12-24 VCC	12-24 VCC
Voltaje de cresta	< 50 VCC (10 kHz max. frequency)	< 50 VCC
Caída de tensión cuando es ON	0,5VCC @ 1 A	1,2 VCC @ 1 A
Corriente máxima(resistiva)	0,5 A / pt., 1A / pt. como salida normal	1,0 A / punto
Máxima corriente de fuga	15 µA @ 30 VCC	15 µA @ 30 VCC
Máxima corriente de inrush	2 A for 100 ms	2 A for 100 ms
External DC power required	n/a	n/a
Respuesta cuando va de OFF a ON	< 10 µs	< 10 µs
Respuesta cuando va de ON a OFF	< 20 µs	< 0,5 µs
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 4 bancos (no aislados)	
Fusibles	Ninguno (se recomienda colocar fusibles externos)	

Diagrama eléctrico de E/S del D0-06DR

El PLC D0-06DR tiene veinte entradas C.C. y dieciséis salidas de contactos de relevador. El diagrama siguiente muestra un ejemplo típico de cableado. La conexión de alimentación de C.A. usa cuatro terminales según lo mostrado.

Las entradas se organizan en cinco bancos de cuatro entradas cada banco. Cada banco tiene un terminal común aislado y se pueden conectar como entradas drenadoras o surtidoras. El ejemplo de cableado abajo muestra todo los comunes conectados juntos pero se pueden usar fuentes y comunes separados. Se muestra abajo el circuito equivalente para las entradas normales y el circuito de entrada de alta velocidad se muestra a la izquierda.

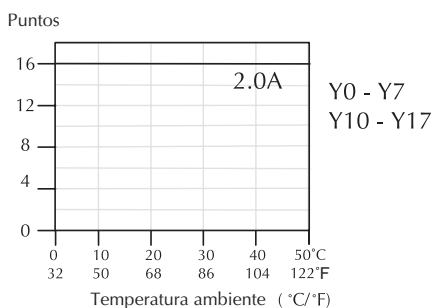
Las salidas se organizan en cuatro bancos de cuatro contactos normalmente abiertos de relevador. Cada banco tiene un terminal común. El ejemplo de cableado abajo muestra todo los comunes conectados juntos pero se pueden usar fuentes y comunes separados. El equivalente del circuito de salida muestra un canal de un banco típico. Los contactos del relevador pueden conmutar voltajes de C.A. o C.C.

2

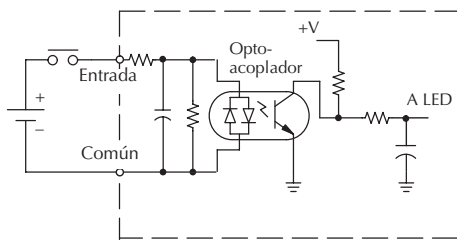
Vida típica del relevador (Operaciones) a temperatura ambiente

Voltaje & tipo de carga	Corriente	
	A 1A	A 2A
24VCC resistiva	500K	250K
24VCC inductiva	100K	50K
110VAC resistiva	500K	250K
110VAC inductiva	200K	100K
220VAC resistiva	350K	200K
220VAC inductiva	100K	50K

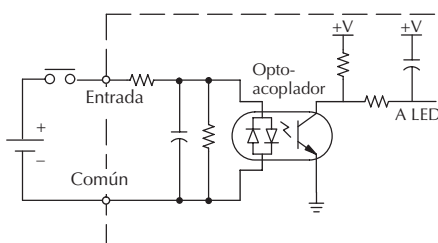
Curva de degradación de salidas de contacto



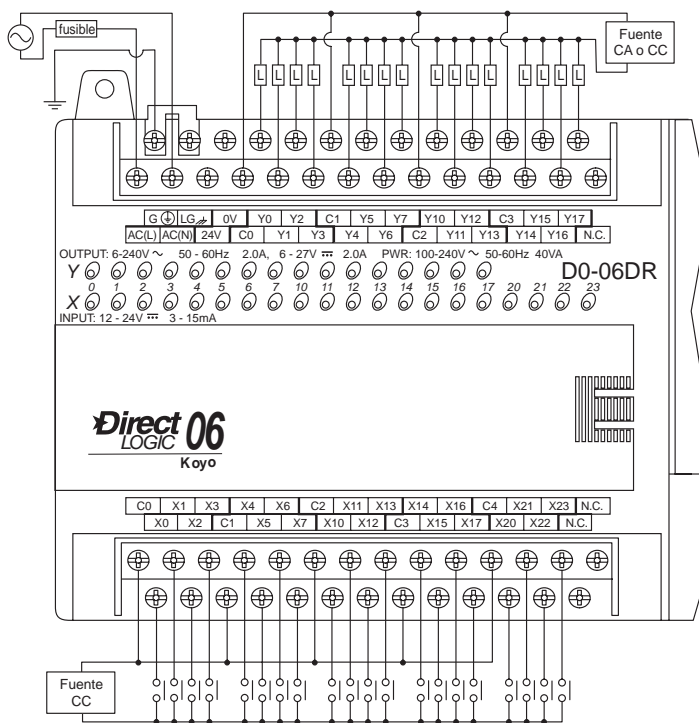
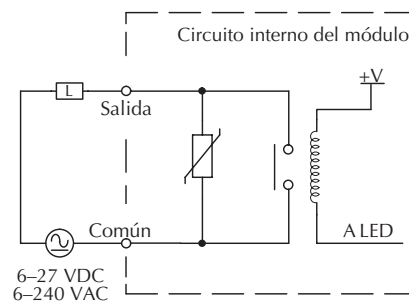
Circuito equivalente, entradas HSIO (X0-X3)



Circuito equivalente, entradas normales (X4-X23)



Circuito de salida equivalente



Especificaciones generales D0-06DR	
Requerimientos de alimentación	100 – 240 VCA, 40 VA máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence /print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removable
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales CC X4 – X23
Voltage mínimo y máximo	10,8 – 26,4 VCC	10,8 – 26,4 VCC
Voltaje de operación	12 -24 VCC	12 -24 VCC
Voltaje de cresta	30 VCC (Máxima frecuencia 7 kHz)	30 VCC
Ancho de pulso mínimo	70 µs	N/A
Voltaje de detección de estado ON	> 10 VCC	> 10 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Impedancia de entradas	1,8 kΩ @ 12 – 24 VCC	2,8 kΩ @ 12 – 24 VCC
Max. corriente en las entradas	6mA @12VCC 13mA @24VCC	4mA @12VCC 8.5mA @24VCC
Corriente mínima en estado ON	>5 mA	>4 mA
Corriente máxima en estado OFF	< 0,5 mA	<0,5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	<70 µs	2 – 8 ms, 4 ms típico
Indicadores de estado	Lado lógico	Lado lógico
Comunes	4 canales/común x 5 bancos aislados	

Especificaciones de salidas a relevador	
Rango de voltaje de salida (Min. - Max.)	5 -264 VCA (47 -63 Hz), 5 - 30 VCC
Voltaje de operación	6 -240 VCA (47 -63 Hz), 6 - 27 VCC
Corriente de salida	2A / punto, 6A / común
Voltaje máximo	264 VCA, 30 VCC
Máxima corriente de fuga	0,1 mA @264 VCA
Carga mínima recomendada	5 mA
Tiempo de respuesta cuando va de OFF a ON	< 15 ms
Tiempo de respuesta cuando va de ON a OFF	< 10 ms
Indicadores de estado	Lado lógico
Comunes	4 canales / común x 4 bancos
Fusibles	Ninguno (se recomienda colocar fusibles externos)

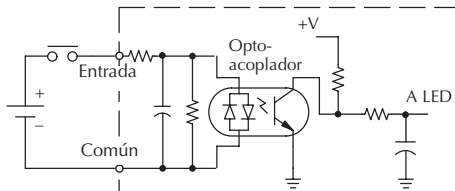
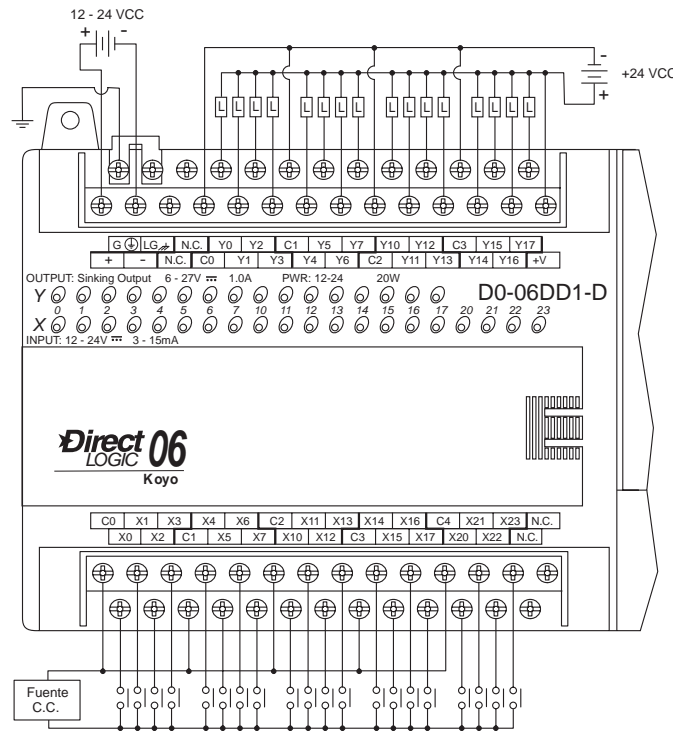
Diagrama de cableado de E/S del D0-06DD1-D

Estos PLCs tienen veinte entradas y dieciséis salidas drenadoras de C.C. El diagrama siguiente muestra un ejemplo típico de cableado. La conexión de alimentación de C.C. utiliza cuatro terminales en la izquierda según lo mostrado.

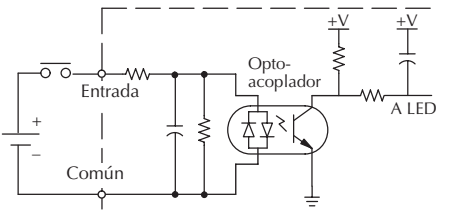
Las entradas se organizan en cinco bancos de cuatro puntos cada uno. Cada banco tiene un terminal común aislado -y se puede conectar como entradas drenadoras o surtidoras. El ejemplo del cableado abajo demuestra todo los comunes conectados juntos, pero se pueden utilizar fuentes y comunes separados.

Todas las salidas comparten realmente un común, es decir, los comunes no son aislados. Note la alimentación externa alimentando el terminal +V.

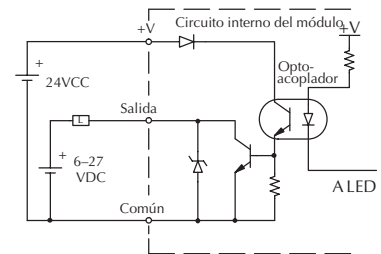
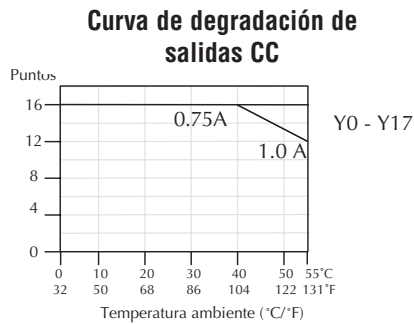
2



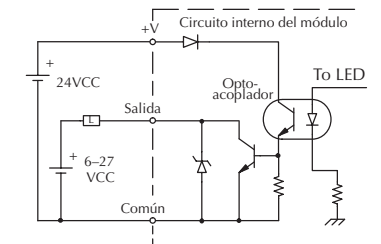
Entradas HSIO(X0-X3)



Circuito de entradas normales(X4-X23)



Salidas HSIO (Y0 - Y1)



Salidas normales CC (Y2 - Y17)

Especificaciones generales D0-06DD1-D	
Requerimientos de alimentación	12 – 24 VCC, 20 W máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence/print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removible
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales CC X4 – X23
Rango de voltaje	10.8 – 26.4 VCC	10.8 – 26.4 VCC
Rango del voltaje de operación	12 – 24 VCC	12 – 24 VCC
Voltaje de cresta	30 VCC (máxima frecuencia 7 kHz)	30 VCC
Ancho mínimo del pulso	70 µs	N/A
Voltaje de detección de estado ON	>10,0 VCC	> 10,0 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Max. Corriente en las entradas	6mA @12VCC, 13mA @24VCC	4mA @12VCC, 8.5mA @24VCC
Impedancia de entradas	1.8 kΩ @ 12 – 24 VCC	2.8 kΩ @ 12 – 24 VCC
Corriente mínima en estado ON	>5 mA	>4 mA
Corriente máxima en estado OFF	< 0,5 mA	<0,5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	<70 µs	2 – 8 ms, 4 ms típico
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales/común x 5 bancos	

Especificaciones de salidas CC		
Parámetro	Salidas de pulsos, Y0 – Y1	Salidas normales, Y2 – Y17
Rango de voltaje	5 – 30 VCC	5 – 30 VCC
rango del voltaje de operación	6 – 27 VCC	6 – 27 VCC
Voltaje de pico	< 50 VCC (10 kHz max. frequency)	< 50 VCC
Caída de tensión cuando ON	0.3 VCC @ 1 A	0.3 VCC @ 1 A
Corriente máxima(resistiva)	0,5 A/pto, 1A/pto como salida normal	1,0 A / punto
Corriente de fuga máxima	15 µA @ 30 VCC	15 µA @ 30 VCC
Corriente de inrush máxima	2 A por 100 ms	2 A por 100 ms
Corriente de una fuente externa	A 20 - 28 VCC, Máx. 150 mA	A 20 - 28 VCC, Máx. 150 mA
Tiempo de respuesta de OFF a ON	< 10 µs	< 10 µs
Tiempo de respuesta de ON a OFF	< 20 µs	< 60 µs
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 4 bancos (no aislados)	
Fusibles	Ninguno (se recomienda colocar fusibles externos)	

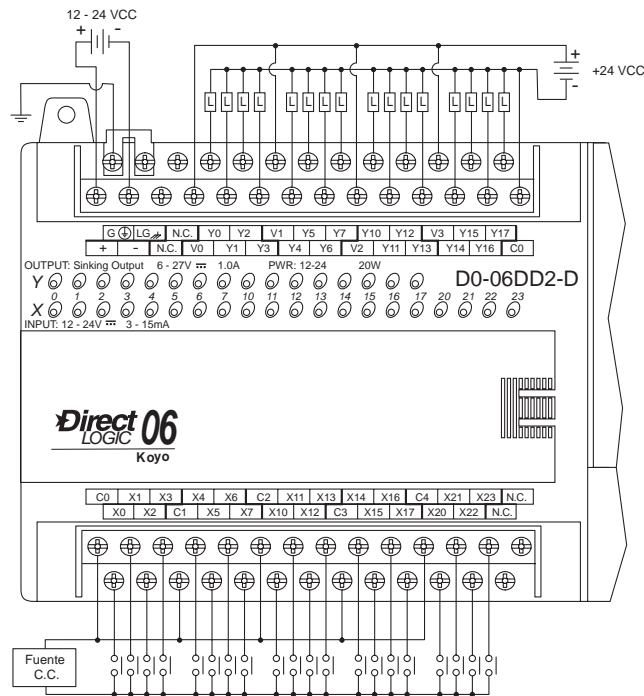
Diagrama de cableado de E/S del D0-06DD2-D

Estos PLCs tienen veinte entradas y dieciséis salidas surtidoras de C.C. El diagrama siguiente muestra un ejemplo típico de cableado. La conexión de alimentación de C.C. utiliza cuatro terminales en la izquierda según lo mostrado.

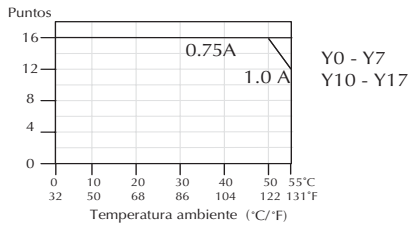
Las entradas se organizan en cinco bancos de cuatro puntos cada uno. Cada banco tiene un terminal común y se puede conectar como entradas drenadoras o surtidoras. El ejemplo del cableado abajo demuestra todo los comunes conectados juntos, pero se pueden utilizar fuentes y comunes separados.

Todas las salidas comparten realmente un común, es decir, los comunes no son aislados. Note la alimentación externa alimentando el terminal +V.

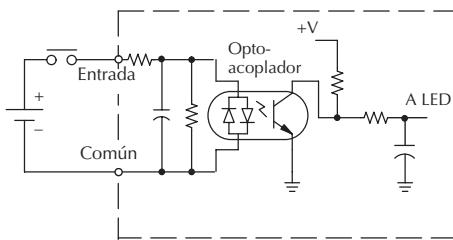
2



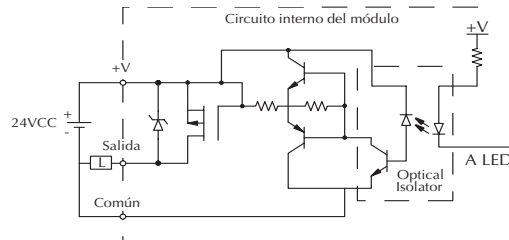
Curva de degradación de salidas CC



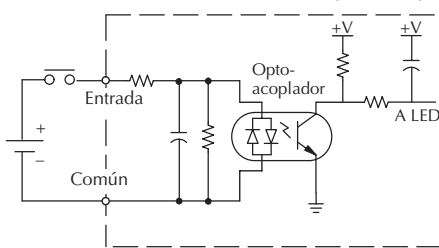
Entradas HSIO (X0-X3)



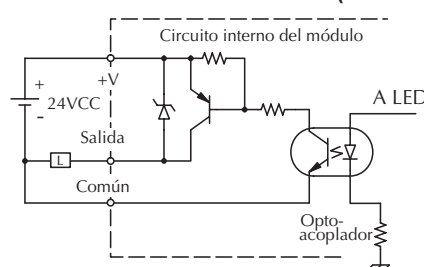
Salidas HSIO (Y0 - Y1)



Circuito de entradas normales (X4-X23)



Salidas normales CC (Y2 - Y17)



Especificaciones generales DO-06DD2-D	
Requerimientos de alimentación	12 – 24 VCC, 20 W máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original), 8 bits de datos, 1 bit stop, paridad Odd	K-Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence/print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removable
Sección del cable	Un AWG#16 o dos AWG#18, AWG#24 mínimo

Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales X4 – X23
Rango de voltage	10,8 – 26,4 VCC	10,8 – 26,4 VCC
Rango del voltaje de operación	12 – 24 VCC	12 – 24 VCC
Voltaje de cresta	30 VCC (máxima frecuencia 7 kHz)	30 VCC
Ancho de pulso mínimo	70 µs	N/A
Voltaje de detección de estado ON	>10,0 VCC	> 10,0 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Max. Corriente en las entradas	15mA @26,4VCC	11mA @26,4VCC
Impedancia de entradas	1,8 kΩ @ 12 – 24 VCC	2,8 kΩ @ 12 – 24 VCC
Corriente mínima en estado ON	5 mA	3 mA
Corriente máxima en estado OFF	0,5 mA	0,5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	<70 µs	2 – 8 ms, 4 ms típico
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales/común x 5 bancos no aislados	

Especificaciones de salidas CC		
Parámetro	Salidas de pulsos, Y0 – Y1	Salidas normales, Y2 – Y17
Min. - Max. Voltage Range	10,8 – 26,4 VCC	10,8 – 26,4 VCC
Voltaje	12 – 24 VCC	12 – 24 VCC
Voltaje de cresta	30 VCC (10 kHz max. frequency)	30 VCC
Caída de tensión cuando ON	0,5 VCC @ 1 A	1,2 VCC @ 1 A
Corriente máxima(resistiva)	0,5 A/pto, 1A/pto.como salida normal	1,0 A / punto
Corriente de fuga máxima	15 µA @ 30 VCC	15 µA @ 30 VCC
Corriente de inrush máxima	2 A por 100 ms	2 A por 100 ms
Corriente externa requerida	N/A	N/A
Respuesta cuando OFF a ON	< 10 µs	< 10 µs
Respuesta cuando ON a OFF	< 20 µs	< 0.5 ms
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales / común x 4 bancos (no aislados)	
Fusibles	Ninguno (se recomienda colocar fusibles externos)	

Diagrama de cableado de E/S del D0-06DR-D

El PLC D0-06DR-D tiene veinte entradas C.C. y dieciséis salidas de contactos de relevador. El diagrama siguiente muestra un ejemplo típico de cableado. La conexión de alimentación de C.C. utiliza tres terminales según lo mostrado.

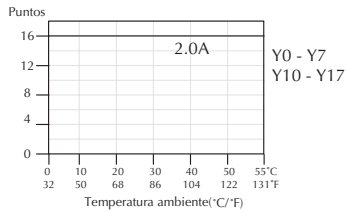
Las entradas se organizan en cinco bancos de cuatro puntos cada uno. Cada banco tiene un terminal común aislado y se puede conectar como entradas drenadoras o surtidoras. El ejemplo del cableado arriba muestra todo los comunes conectados juntos pero se pueden utilizar fuentes separadas.

Las salidas se organizan en cuatro bancos de cuatro contactos normalmente abiertos de relevador. Cada banco tiene un terminal común. El equivalente de circuito de salida muestra un canal de un banco típico. Los contactos del relevador pueden conmutar voltajes C.A. o C.C.

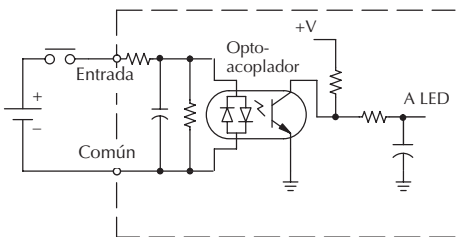
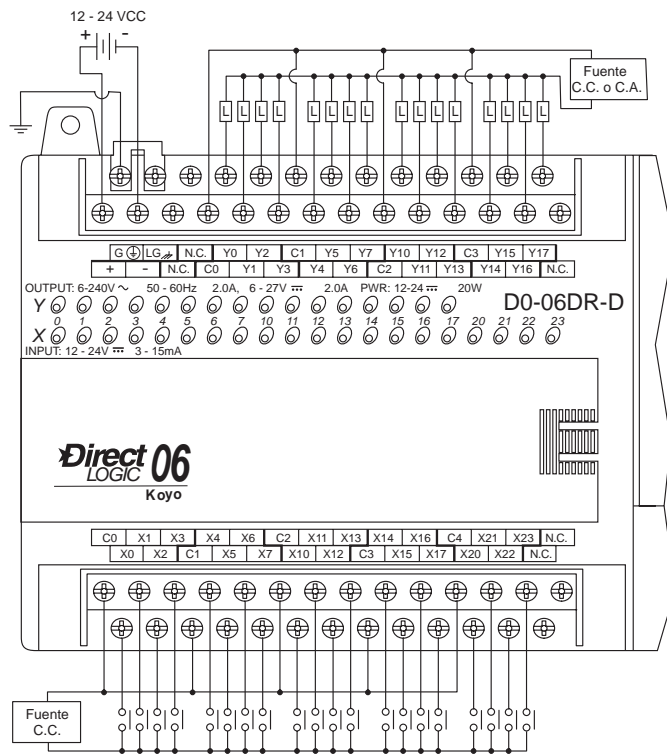
2

Vida típica del contacto (Operaciones) a temperatura ambiente

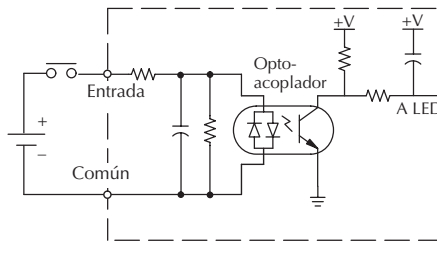
Voltaje & tipo de carga	Corriente	
	A 1A	A 2A
24VCC resistiva	500K	250K
24VCC inductiva	100K	50K
110VAC resistiva	500K	250K
110VAC inductiva	200K	100K
220VAC resistiva	350K	200K
220VAC inductiva	100K	50K



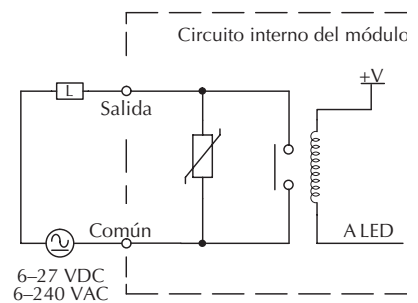
Curva de degradación de salidas a contacto



Circuito de salida HSIO (X0-X3)



Circuito de entradas normales(X4-X23)



Circuito de salida normal

Especificaciones generales D0-06DR-D	
Requerimientos de alimentación	12 – 24 VCC, 20 W máximo,
Puerto de comunicación 1, 9600 baud (Fijo), 8 data bits, 1 stop bit, odd parity	K-Sequence (Esclavo), DirectNET (Esclavo), MODBUS (Esclavo)
Puerto de comunicación 2, 9600 baud (original),, 8 data bits, 1 stop bit, odd parity	K-Sequence (Esclavo), DirectNET (Maestro/Esclavo), MODBUS (Maestro/Esclavo), Non-sequence/print, ASCII
Tipo de cable de programación	D2-DSCBL
Temperatura de operación	32 a 131° F (0 a 55 C)
Temperatura de almacenamiento	-4 a 158° F (-20 a 70 C)
Humedad relativa	5 a 95% (sin condensación)
Calidad del aire ambiente	No se permite en ambientes con gases corrosivos
Vibración	MIL STD 810C 514.2
Choque	MIL STD 810C 516.2
Inmunidad al ruido	NEMA ICS3-304
Tipo de terminal	Removible
Sección del cable	UN AWG#16 o 2 AWG#18, AWG#24 minimum

Especificaciones de entradas CC		
Parámetro	Entradas HSIO, X0 – X3	Entradas normales CC X4 – X23
Rango de voltage	10,8 – 26,4 VCC	10,8 – 26,4 VCC
Rango del voltaje de operación	12 -24 VCC	12 -24 VCC
Voltaje de cresta	30 VCC (máxima frecuencia 7 kHz)	30 VCC
Ancho de pulso mínimo	70 µs	N/A
Voltaje de detección de estado ON	> 10 VCC	> 10 VCC
Voltaje de estado OFF	< 2,0 VCC	< 2,0 VCC
Impedancia de entradas	1.8 kΩ @ 12 – 24 VCC	2.8 kΩ @ 12 – 24 VCC
Max. corriente en las entradas	6mA @12VCC; 13mA @24VCC	4mA @12VCC; 8,5mA @24VCC
Corriente mínima en estado ON	>5 mA	>4 mA
Corriente máxima en estado OFF	< 0.5 mA	<0.5 mA
Respuesta cuando va de OFF a ON	<70 µs	2 – 8 ms, 4 ms típico
Respuesta cuando va de ON a OFF	< 70 µs	2 – 8 ms, 4 ms típico
Indicadores LED de estado	Lado lógico	Lado lógico
Comunes	4 canales/común x 5 bancos aislados	

Especificaciones de salidas a relevador	
Rango de voltaje de salida (Min. - Max.)	5 -264 VCA (47 -63 Hz), 5 - 30 VCC
Voltaje de operación	6 -240 VCA (47 -63 Hz), 6 - 27 VCC
Corriente de salida	2A / punto 6A / común
Voltaje máximo	264 VCA, 30 VCC
Máxima corriente de fuga	0,1 mA @264 VCA
Carga mínima recomendada	5 mA
Tiempo de respuesta cuando va de OFF a ON	< 15 ms
Tiempo de respuesta cuando va de ON a OFF	< 10 ms
Indicadores LED de estado	Lado lógico
Comunes	4 canales/común x 4 bancos aislados
Fusibles	Ninguno (se recomienda colocar fusibles externos)

ESPECIFICACIONES Y OPERACIÓN DE LA CPU



CAPÍTULO 3

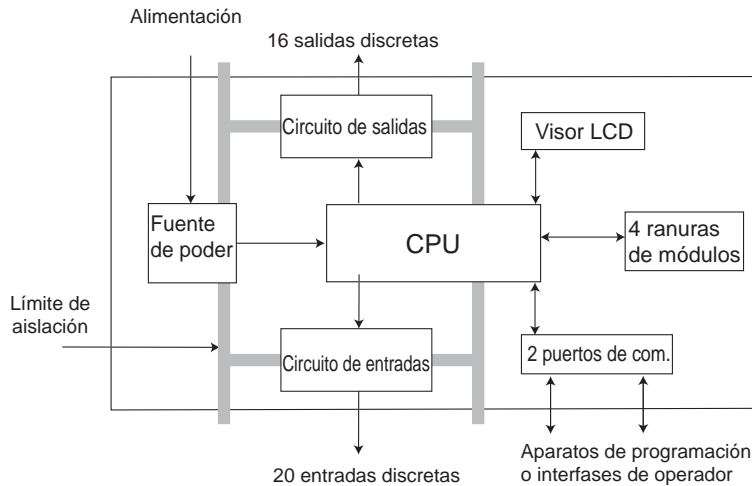
En este capítulo

Introducción	3-2
Especificaciones de la CPU	3-3
Configuración del hardware de la CPU	3-4
Usando una batería de respaldo	3-8
Operación de la CPU	3-12
Tiempo de respuesta de E/S	3-17
Consideraciones del tiempo de barrido de la CPU	3-20
Mapa de memoria	3-25
Memoria del sistema DL06	3-29
Mapa de bits de relevadores de control	3-35
Mapa de bits de estado de temporizadores	3-37
Mapa de bits de estado de contadores	3-37
Mapa de bits de Remote I/O	3-38

Introducción

La unidad central de proceso (CPU) es el corazón del PLC. Casi todas las operaciones del PLC son controladas por la CPU, así que es importante que esté configurado correctamente. Este capítulo entrega la información necesaria para entender:

- Pasos requeridos para configurar la CPU
- Operación de los programas ladder (Se usa la palabra **ladder** para describir la programación como diagramas eléctricos)
- Organización de la memoria V



Nota: La función de alta velocidad (HSIO) se compone de hardware configurable en el DL06. No se considera parte de la CPU, porque no ejecuta el programa ladder.

Características principales de la CPU DL06

El PLC DL06 tiene 14,8K palabras de memoria, con 7,6K palabras de memoria ladder y 7,6K palabras de memoria V para usuario (registros de datos). El almacenamiento del programa se hace en una memoria Flash, que es una parte de la CPU en el PLC. Además, existe una memoria RAM en la CPU que almacenará los parámetros del sistema, la memoria V y otros datos que no están en el programa de aplicación. La RAM es alimentada y es guardada por un "super condensador", almacenando los datos por varias horas en caso de falla de energía. El condensador se carga automáticamente durante la operación del PLC con energía externa.

El PLC DL06 tiene veinte puntos de entradas y dieciséis puntos de salidas discretos fijos. Tiene además 4 ranuras para colocar módulos opcionales de E/S. Especificaciones sobre estos módulos opcionales se encuentran en el manual D0-OPTIONS-M-SP.

Hay más de 220 instrucciones diferentes que están disponibles para el desarrollo de programas y para diagnóstico, así como también se puede supervisar el PLC a partir del programa de aplicación o desde una interface de operador. Los capítulos 5, 6 y 7 dan más detalles.

El PLC DL06 tiene incorporado dos puertos de comunicación, de modo que puede conectarlo fácilmente a un programador portátil, a una interface de operador o a una computadora personal sin necesitar de ningún hardware adicional e incluso puede crear redes seriales para aumentar el número de entradas y salidas. Con un módulo opcional puede crear una red de comunicación Ethernet con otros aparatos.

Especificaciones de la CPU

Especificaciones	
Característica	Detalle
Memoria total de programa (palabras)	14.8K
Memoria Ladder (palabras)	7680
Memoria V total	7616
Memoria V de usuario (palabras)	7488
Memoria V no volátil (palabras)	128
Tiempo de ejecución de un contacto	0,6 us
Tiempo de barrido típico para 1K booleano	1 - 2 ms
Programación de estilo ladder RLL	Si
Programación RLL y RLLPLUS	Si
Modificaciones del programa durante el modo RUN	Si
Tiempo de barrido	Variable o fijo
Programador portátil	Si
Programación con <i>DirectSOFT</i> para Windows	Si
Puertos de comunicación incluidos (RS232C)	Si
Memoria FLASH	Normal en la CPU
Puntos de E/S discretos disponibles locales	36
Canales locales de E/S análogas, máximo	Ninguno
E/S HSIO(cuadratura, tren de pulsos, interrupción, etc.)	Si, 2
Cantidad de puntos de entradas y salidas locales	20 entradas, 16 salidas
Cantidad de instrucciones disponibles	229
Relevadores de control internos	1024
Relevadores especiales (Definidos por el sistema)	512
Etapas en RLLPLUS	1024
Temporizadores	256
Contadores	128
Entradas y salidas inmediatas	Si
Entradas de interrupción (externas o por tiempo)	Si
Subrutinas	Si
Lazos de For/Next	Si
Operaciones aritméticas (Con enteros y con punto flotante)	Si
Instrucciones de secuenciador de tambor (Drum)	Si
Hora y fecha	Si
Diagnóstico interno	Si
Seguridad con contraseña	Si
Registro de errores del sistema	Si
Registro de errores del usuario	Si
Respaldo por batería	D2-BAT-1 opcional disponible (no viene incluida con el PLC)

Configuración del hardware de la CPU

Diagramas de los puertos de comunicación serial

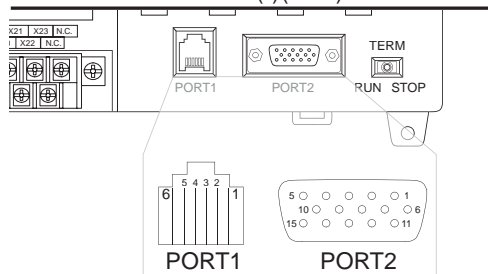
Hay cables disponibles que le permiten conectarse fácilmente desde un programador o una computadora personal al PLC DL06. Sin embargo, si usted necesita construir sus propios cables, puede usar los diagramas de clavijas mostrados abajo. El DL06 requiere un enchufe RJ-12 de 6 clavijas para el puerto 1 y un conector SVGA de 15 clavijas para el puerto 2.

El PLC DL06 tiene dos puertos de comunicación seriales incorporados. El puerto 1 (RS232C solamente) se utiliza generalmente para conectar con un programador D2-HPP, *DirectSOFT*, una interface de operador, un esclavo MODBUS o *DirectNET* solamente. The baud rate es fijo en 9600 baud en el puerto 1. El puerto 2 (RS232C/RS422/RS485) se puede usar para conectar con un D2-HPP, *DirectSOFT*, una interface del operador, un maestro o esclavo MODBUS RTU o *DirectNET* o ASCII como entrada y salida. El puerto 2 tiene un rango de velocidades a partir de 300 Baud hasta 38.4KBaud.

3

Descripciones de clavijas Puerto 1

1	0V	Conexión (-)(GND)
2	5V	Conexión (+)
3	RXD	Recibe datos (RS-232C)
4	TXD	Transmite datos (RS-232C)
5	5V	Conexión (+)
6	0V	Conexión (-)(GND)



Descripciones de clavijas Puerto 2

1	5V	Conexión (+)
2	TXD	Transmite datos (RS-232C)
3	RXD	Recibe datos (RS-232C)
4	RTS	Ready to send
5	CTS	Clear to send
6	RXD-	Recibe datos (-) (RS-422/485)
7	0V	Conexión (-) (GND)
8	0V	Conexión (-) (GND)
9	TXD+	Transmite datos (+) (RS-422/485)
10	TXD-	Transmite datos (-) (RS-422/485)
11	RTS+	Ready to send (+) (RS-422/485)
12	RTS-	Ready to send (-) (RS-422/485)
13	RXD+	Recibe datos (+) (RS-422/485)
14	CTS+	Clear to send (+) (RS-422/485)
15	CTS-	Clear to send (-) (RS-422/485)

Comunicaciones del Puerto 1

Com 1 Se conecta a HPP, DirectSOFT, interfaces de operador, etc.

6 clavijas, RS232C

Tasa de comunicación(baud): 9600 (fija)

Paridad: odd (valor original de fábrica)

Dirección de la estación: 1 (fija)

8 bits de datos

1 bit start, 1 bit stop

Asíncrono, half-duplex, DTE

Protocolo: (Seleccionable automáticamente) K-sequence (solamente esclavo), DirectNET (solamente esclavo), MODBUS (solamente esclavo)

Comunicaciones del Puerto 2

Com 2 Se conecta a HPP, DirectSOFT, interfaces de operador, etc.

15-clavijas, puerto de funciones múltiples, RS232C, RS422, RS485

Tasa de comunicación (baud): 300, 600, 1200, 2400, 4800, 9600, 19200, 38400

Paridad: odd (valor original), even, 0 (nada)

Dirección de la estación: 1 (valor original)

8 bits de datos

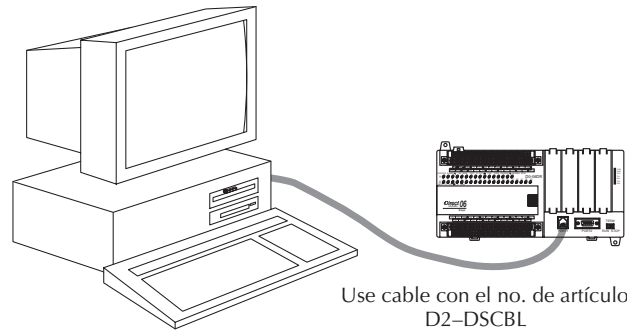
1 bit start, 1 bit stop

Asíncrono, half-duplex, DTE

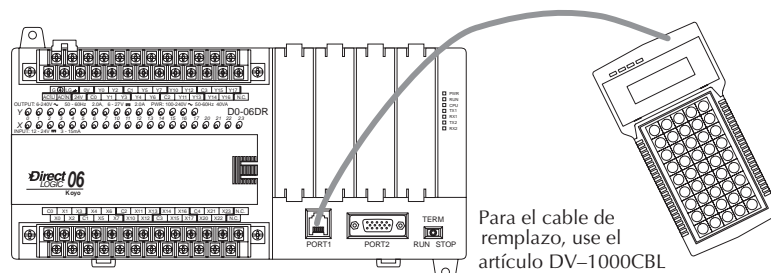
Protocolos: (selección automática) K-sequence (solamente esclavo), DirectNET (maestro/esclavo), MODBUS (maestro/esclavo), non-sequence/print/ASCII in/out

Conectando aparatos de programación

Si se usa una computadora personal IBM compatible con el paquete de *DirectSOFT™*, usted puede conectar la computadora a cualquiera de los puertos seriales del DL06. Para un ambiente de oficina de ingeniería (típico durante el desarrollo del programa), este es el método preferido para programar.



El programador D2-HPP portátil es conectado a la CPU con el cable del programador. Este aparato es ideal para mantener las instalaciones existentes o hacer pequeños cambios de programa. El programador portátil es embarcado con un cable, que tiene aproximadamente 6,5 pies (200 cm).

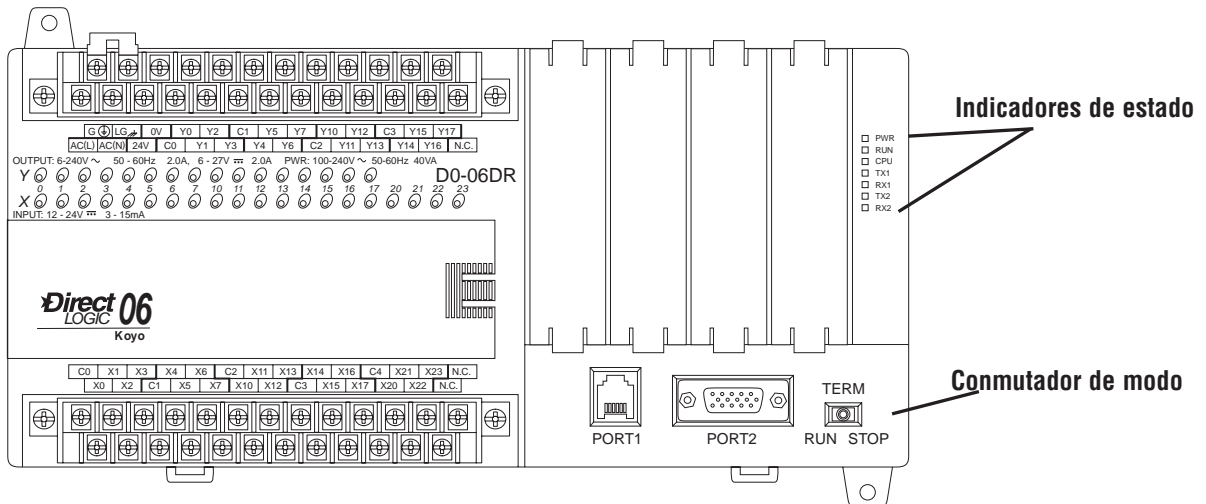


Información de configuración de la CPU

Aunque usted tenga años de experiencia usando PLCs, hay algunas cosas que se deben hacer antes de que pueda comenzar a escribir y entrar programas. Esta sección incluye algunas cosas básicas, tales como cambiar el modo de la CPU e incluyen también algunas cosas que usted nunca tendrá que usar. Aquí está una lista breve de los artículos que se discuten:

- Usando funciones auxiliares
- Como borrar el programa (y otras áreas de la memoria)
- Cómo inicializar la memoria del sistema
- Como configurar los rangos de memoria retentiva

Los párrafos siguientes le entregan la información de configuración necesaria para obtener la CPU preparada para programar. Incluyen instrucciones para cada tipo de aparato de programación que Ud. use. El manual del programador D2-HPP le enseña como usar las teclas para ejecutar todas las funciones. El manual del programa *DirectSOFT™* le da una descripción de los menús y el teclado necesario para ejecutar procedimientos de configuración con el software *DirectSOFT*.



La indicación de estado del PLC

Los LEDs en el frente el PLC, de indicación del estado de la CPU tienen funciones específicas que pueden ayudar a programar y localizar problemas

Funciones del conmutador de modo

El conmutador de modo en el PLC DL06 permite activar o desactivar cambios de modo de programa en la CPU. A menos que el conmutador de modo esté en la posición TERM, no serán permitidos cambios en los modos RUN y STOP por ningún dispositivo de interface, (programador portátil, paquete de programación *DirectSOFT* o interface de operador). Los programas pueden ser vistos o supervisados pero no se puede realizar ningún cambio. Si el conmutador está en la posición TERM y no hay contraseña de programa en efecto, se permiten todos los modos de funcionamiento y también el acceso del programa a través de dispositivos de programación o de supervisión conectados.

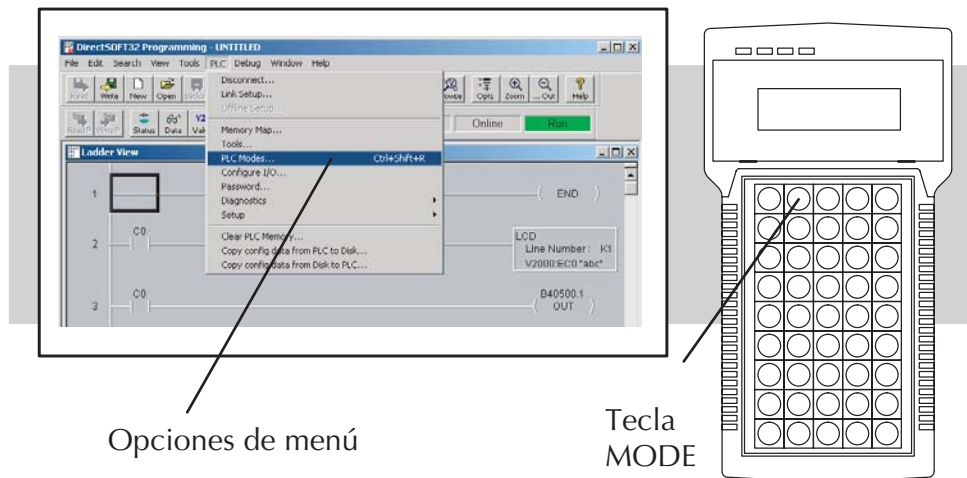
Indicador	Estado	Significado
PWR	ON	Alimentación en orden
	OFF	La alimentación ha fallado
RUN	ON	La CPU está en modo RUN
	OFF	La CPU está en modo STOP o PROGRAM
	Parpadeando	La CPU está en modo de actualización de firmware
CPU	ON	Error por diagnóstico de la CPU
	OFF	Diagnóstico de la CPU en orden
	Parpadeando	Batería con bajo voltaje
TX1	ON	Están siendo transmitidos datos por la CPU: Puerto 1
	OFF	No están siendo transmitidos datos por la CPU: Puerto 1
RX1	ON	Están siendo recibidos datos por la CPU: Puerto 1
	OFF	No están siendo recibidos datos por la CPU: Puerto 1
TX2	ON	Están siendo transmitidos datos por la CPU: Puerto 2
	OFF	No están siendo transmitidos datos por la CPU: Puerto 2
RX2	ON	Están siendo recibidos datos por la CPU: Puerto 2
	OFF	No están siendo recibidos datos por la CPU: Puerto 2

Cambiando modos en el PLC DL06.

Posición del conmutador de modo	Acción de la CPU
RUN (El programa está funcionando)	La CPU es forzada en el modo RUN si no se encuentra ningún error. No se permite ningún cambio por un dispositivo de programación o supervisión.
TERM (Terminal) RUN	Están disponibles los modos PROGRAM y TEST (PRUEBA). Son permitidos cambios de modo y de programa por un dispositivo de programación o supervisión.
STOP (parado)	La CPU es forzada al modo STOP (PARADO). No se permite ningún cambio por un dispositivo de programación.

3

Hay dos formas de cambiar el modo de la CPU. Se puede usar el conmutador de modo para seleccionar el modo de operación, o se puede colocar el conmutador en la posición TERM y se selecciona el modo usando un aparato de programación. Con el conmutador en esta posición, la CPU se puede cambiar entre los modos RUN y Program. Usted puede utilizar *DirectSOFT* o el programador D2-HPP para cambiar el modo de operación de la CPU. Con *DirectSOFT* usted usa una opción del menú en el menú del PLC o el ícono MODE. Con el programador D2-HPP, usted utiliza la tecla MODE.



Modo de operación durante la energización del PLC.

La CPU DL06 se energizará normalmente en el modo que estaba antes de la interrupción de la energía. Por ejemplo, si la CPU estaba en el modo Program cuando se desconectó la energía, la CPU se encenderá en el modo Program.



ADVERTENCIA: Cuando el super condensador se ha descargado, la memoria del sistema no puede retener el modo previo de operación. Cuando esto ocurre, el PLC puede encenderse en el Modo RUN o PROGRAM si el conmutador de modo está en la posición TERM. No hay una manera de determinar cuál modo entrará durante el encendido. Si no sigue esta advertencia se aumenta el riesgo de partida de equipo inesperadamente.

El modo como el PLC se energiza también es determinado por el estado del bit B7633.13. Si el bit está ON y el switch de modo está en la posición TERM, el PLC se va al modo RUN al energizarse. Si está OFF y el switch está en TERM, el PLC se energiza en el estado en que estaba cuando fue apagado.

Usando una batería de respaldo

Hay disponible una batería opcional de litio para mantener la memoria retentiva RAM del sistema cuando el sistema DL06 está sin energía externa. La vida típica de la batería de la CPU es cinco años, que incluye períodos normales de operación con y sin energía. Sin embargo, considere instalar una batería fresca si su batería no se ha cambiado recientemente y si el sistema va a estar apagado por un período de más de diez días.

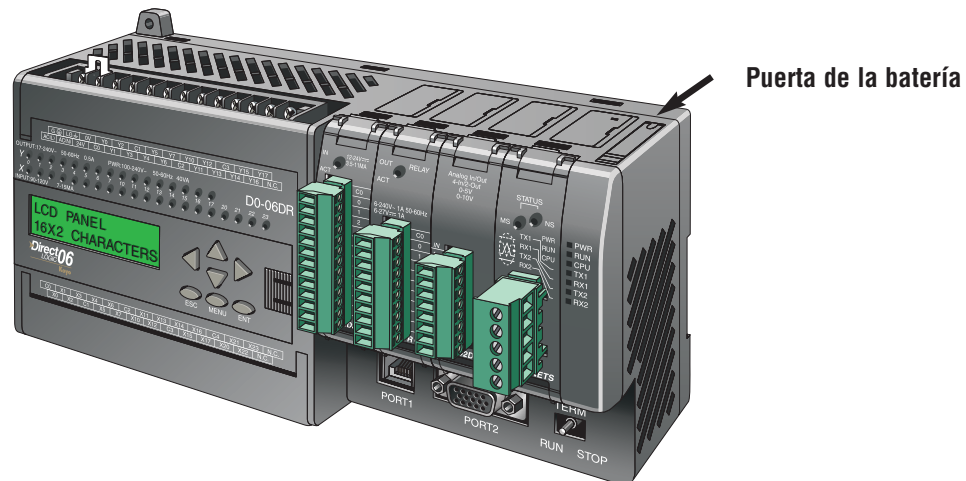


NOTA: Antes de instalar o reemplazar la batería de la CPU, salve los parámetros de memoria V y del sistema. Usted puede hacer esto usando **DirectSOFT** DirectSOFT para salvar el programa, la memoria V y los parámetros de sistema a un disco duro o flexible en una computadora personal.

3

Para instalar la batería de la CPU D2-BAT-1 en el PLC DL06 (el artículo CR 2354):

1. Apriete el clip que retiene la puerta de la batería hacia abajo y abra la puerta de la batería.
2. Coloque la batería en la ranura vertical de tipo moneda.
3. Cierre la puerta de la batería asegurándose que cierra seguramente en su lugar.
4. Anote la fecha que la batería se instaló.



ADVERTENCIA: No trate de recargar la batería o no lance una batería usada al fuego. La batería puede estallar o soltar materiales peligrosos.

La batería de respaldo

La batería de respaldo está disponible inmediatamente después que la batería se ha instalado. Se pueden configurar rangos retentivos de memoria en la CPU para mantener los datos en los elementos C, S, T, CT y la memoria V si el PLC no tiene energía. La indicación de "batería con bajo voltaje" es habilitada poniendo el bit 12 de la memoria V7633 ON (B7633.12). *El indicador LED de la CPU parpadeará si la batería esta con bajo voltaje.* El relevador SP43 se hará ON también cuando el voltaje de la batería es bajo (vea la tabla en la página 4-6). Si no se desea una función de indicación de voltaje de batería bajo, no haga ON el bit V7633.12. El super condensador mantendrá la memoria SI es configurada como retentiva, sin importar el estado de B7633.12. La batería hará lo mismo, pero por un tiempo mucho más largo.

Funciones auxiliares

Muchas tareas de configuración de la CPU implican el uso de funciones auxiliares(AUX). Las funciones AUX realizan diversas operaciones, desde limpiar la memoria del programa ladder, mostrar el tiempo de barrido, copiar programas a EEPROM en el programador D2-HPP, etc. Se dividen en categorías que afectan diversos parámetros del sistema. El apéndice A da una descripción de las funciones AUX.

Se puede tener acceso a las funciones AUX desde *DirectSOFT* o desde el programador D2-HPP. Los manuales para esos productos entregan procedimientos paso a paso para tener acceso a las funciones AUX. Algunas de estas funciones AUX se diseñan específicamente para el programador, así que no serán necesitadas (o disponibles) con el paquete *DirectSOFT*. La tabla siguiente muestra una lista de las funciones auxiliares para el programador.

Operaciones RLL - AUX 2*	
21	Verifique programa
22	Cambie referencia
23	Borre un rango de programa
24	Borre todo el programa
AUX 3* — Operaciones de memoria V	
31	Borre la memoria V
AUX 4* — Configuración de E/S	
41	Muestre al configuración de E/S
42	Diagnóstico de E/S
44	Verifique la configuración de E/S al partir
45	Seleccione la configuración
46	Configure las E/S
AUX 5* — Configuración de la CPU	
51	Modifique el nombre del programa
52	Muestre o cambie la fecha
53	Muestre el tiempo de barrido
54	Inicialice el Scratchpad
55	Configure el temporizador Watchdog
56	Configure el puerto 2 de comunicación
57	Configure rangos retentivos
58	Pruebe operaciones
59	Configuración de Override
5B	Configuración de HSIO
5C	Muestre historia de errores
5D	Configuración del control de barrido
AUX 6* — Configuración de programador D2-HPP	
61	Muestre el No. de revisión
62	Beeper On / Off
65	Haga funcionar diagnóstico
AUX 7* — Operaciones de EEPROM	
71	Copie memoria de la CPU a HPP EEPROM
72	Escriba HPP EEPROM a la CPU
73	Compare CPU a HPP EEPROM
74	Verifique que está limpio (HPP EEPROM)
75	Apague HPP EEPROM
76	Muestre el tipo de EEPROM (CPU y HPP)
AUX 8* — Operaciones de contraseña	
81	Modifique contraseña
82	Desbloquee la CPU con la contraseña
83	Bloquee la CPU con la contraseña

Borre un programa existente

Antes de que se coloque un nuevo programa, asegúrese que la memoria esté limpia. Usted puede utilizar la función AUX 24 para borrar cualquier programa que exista. Se puede también utilizar otras funciones AUX para limpiar otras áreas de memoria.

- AUX 23 — Limpie un rango del programa Ladder
- AUX 24 — Limpie todo el programa Ladder
- AUX 31 — Limpie la memoria V (de usuario, diferente de la del programa)

Inicializando la memoria del sistema

El PLC DL06 mantiene parámetros del sistema en un área de memoria designada a menudo el "scratchpad". En algunos casos, se pueden realizar cambios a la configuración del sistema que será almacenado en la memoria del sistema. Por ejemplo, si usted especifica un rango de relevadores de control (CRs) como retentivo, estos cambios se almacenan en la memoria del sistema. AUX 54 coloca la memoria del sistema a los valores originales de fábrica.



ADVERTENCIA: Puede ser que nunca tenga que usar esta función a menos que desee limpiar información de configuración que esté almacenada en la memoria del sistema. Generalmente, se necesitará solamente inicializar la memoria del sistema si está cambiando programas y el programa antiguo requeriría una configuración especial. Usted puede cargar generalmente nuevos programas sin inicializar la memoria del sistema.

Recuerde, esta función AUX volverá al valor original toda la memoria del sistema. Si ha configurado parámetros especiales tales como rangos retentivos, etc. serán borrados cuando se use AUX 54. Asegúrese de que haya considerado todas las ramificaciones de esta operación antes de que la seleccione.

3

Configuración de rangos retentivos de memoria

Los PLCs DL06 suministran ciertos rangos de memoria retentiva por defecto. Una memoria retentiva es tal que al desconectar la alimentación del PLC y luego volverla a encender varias horas más tarde, el contenido se mantiene y no se va a cero. Los rangos retentivos originales son convenientes para muchos usos, pero puede cambiarlos si su uso requiere rangos retentivos adicionales o si no necesita ningún rango retentivo. Los valores originales de fábrica son:

Area de memoria	DL06	
	Rango original	Rango disponible
Relevadores de control C	C1000 – C1777	C0 – C1777
Memoria V	V400 – V37777	V0 – V37777
Temporizadores T	Ninguno por defecto	T0 – T377
Contadores CT	CT0 – CT177	CT0 – CT177
Etapas	Ninguno por defecto	S0 – S1777

Usted puede utilizar AUX 57 para configurar los rangos retentivos. Usted puede también utilizar menús de *DirectSOFT* para seleccionar los rangos retentivos. El apéndice A contiene información detallada sobre funciones auxiliares.



ADVERTENCIA: Los PLCs DL06 no vienen con una batería. El super condensador conservará los valores en el evento de un apagón, pero solamente por un período corto, dependiendo de algunas condiciones (cerca de 4 días). Si los rangos retentivos son importantes para su uso, asegúrese de usar una batería opcional.

Recomendamos leer el apéndice F -Memorias del PLC- para que el lector conozca más sobre las características físicas de las memorias del PLC DL06.

Usando una contraseña

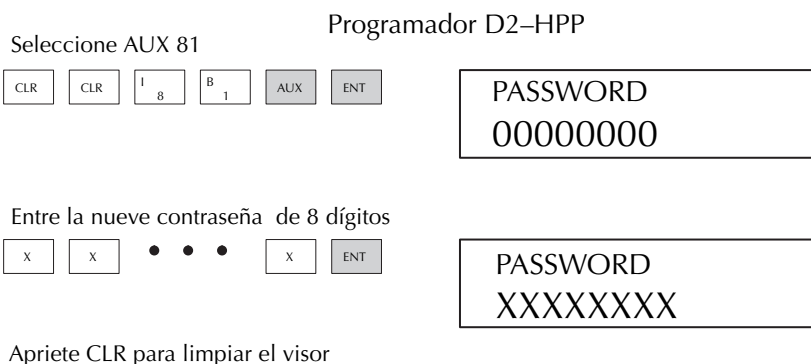
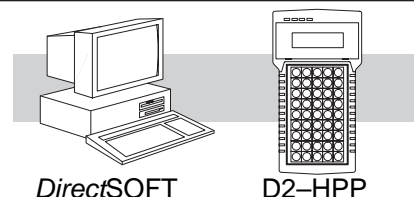
Los PLCs DL06 permiten que se utilice una contraseña para ayudar a reducir al mínimo el riesgo de cambios no autorizados del programa y/o de los datos. Una vez que usted coloque una contraseña se puede "trabar" el PLC contra acceso. Una vez que la CPU sea bloqueada debe entrar la contraseña antes de que pueda utilizar un dispositivo de programación para cambiar cualquier parámetro del sistema.

Se puede seleccionar una contraseña numérica de 8 dígitos. El PLC sale de fábrica con una contraseña 00000000. Este valor remueve la protección de contraseña. Si se ha entrado una contraseña en la CPU usted no podrá entrar los ceros para quitarla. Cuando entre la contraseña correcta, se podrá cambiar la contraseña a "ceros" para remover la protección.



ADVERTENCIA: Asegúrese de recordar su contraseña. Si se olvida de su contraseña no podrá tener acceso a la CPU. El PLC se debe enviar a AutomationDirect para remover la contraseña junto con el programa.

Usted puede utilizar el programador D2-HPP o *DirectSOFT*TM para entrar una contraseña. El diagrama siguiente le muestra cómo puede entrar una contraseña con el programador D2-HPP.



Hay tres maneras de trabar la CPU una vez que se haya incorporado la contraseña.

1. Si se desconecta la energía de la CPU, la CPU será trabada automáticamente contra acceso.
2. Si usted entra la contraseña con *DirectSOFT*, la CPU será trabada automáticamente contra acceso cuando salga de *DirectSOFT*.
3. Use AUX 83 para bloquear la CPU.

Cuando usa *DirectSOFT*, se le pedirá una contraseña si se ha bloqueado la CPU. Si usted utiliza el programador D2-HPP, usted tiene que utilizar AUX 82 para desbloquear la CPU. Cuando entre AUX 82, se le pedirá entrar la contraseña.



Nota: La protección de contraseña tiene un nivel más de protección. Este nivel le permite tener protección de contraseña para no bloquear el puerto de comunicación a una interface de operador. Esta contraseña puede ser invocada creando una contraseña con una letra mayúscula "A" seguida de siete caracteres numéricos (e.g. A1234567).

Operación de la CPU

El control apropiado para el proceso requiere una buena comprensión de cómo la CPU controla todos los aspectos de la operación del sistema. Hay cuatro áreas principales que deben ser entendidas antes de que usted cree su programa de aplicación

- **Sistema operativo de la CPU** — la CPU maneja todos aspectos de control del sistema. Una descripción general rápida es dada en la próxima sección.
- **Modos de operación de la CPU** — Los dos modos primarios de operación son el modo **Program** y el modo **RUN**.
- **Tiempos de la CPU** — Las dos áreas importantes que discutimos son el tiempo de respuesta de entradas y salidas y el tiempo de barrido de la CPU.
- **El mapa de memoria de la CPU** — los PLCs DL06 ofrecen una gran variedad de recursos, tal como temporizadores, contadores, entradas, etc. La sección del mapa de memoria muestra la organización y disponibilidad de estos tipos de datos.

El sistema operativo de la CPU

Durante la energización del PLC la CPU inicializa el hardware electrónico interno. La inicialización de memoria comienza con examinar los rangos retentivos de memoria. En general, el contenido de la memoria retentiva está guardado en la memoria y la no retentiva se inicializa a cero (salvo especificación en contrario).

Después de la tarea de energización, que ocurre una vez, la CPU comienza la actividad cíclica de barrido.

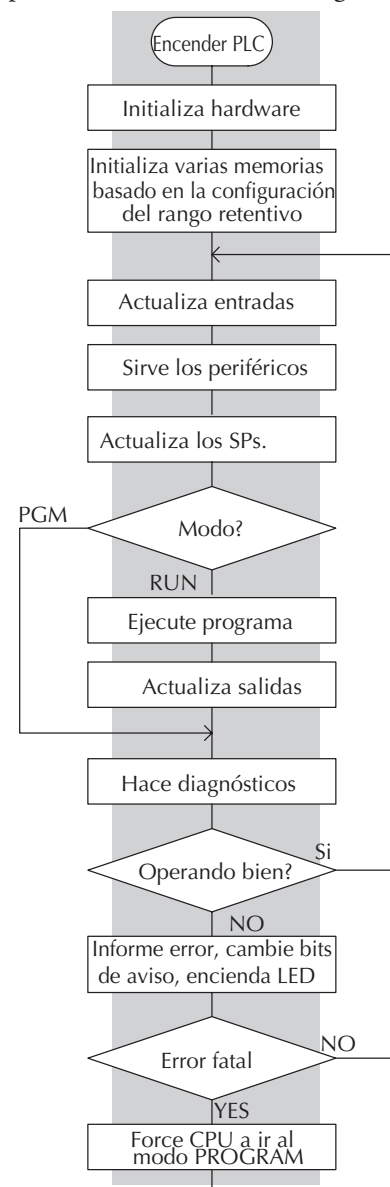
El diagrama de flujo a la derecha muestra cómo se diferencian las tareas, basado en el modo de la CPU y la existencia de cualquier error.

El "tiempo de barrido" se define como el tiempo medio de la tarea del ciclo. **Observe que la CPU está leyendo siempre las entradas, incluso durante el modo de programa.** Esto permite que las herramientas de programación supervisen el estado de las entradas en cualquier momento.

Las salidas se actualizan solamente en modo RUN. En modo de programa, están en el estado desactivado.

La detección de errores tiene dos niveles. Se informan los errores no fatales, pero la CPU permanece en su modo corriente.

Si ocurre un error fatal, la CPU es forzada al modo Program y se apagan las salidas.



Modo Program

En modo Program, la CPU no ejecuta el programa de uso ni actualiza las salidas. El uso principal del modo Program es entrar o cambiar un programa de uso. Se usa también modo Program para configurar los parámetros de la CPU, tales como características HSIO, áreas de memoria retentivas, etc..

Usted puede utilizar el conmutador de modo en el PLC o un dispositivo de programación, tal como *DirectSOFT* o el programador D2-HPP para colocar la CPU en modo Program.

Modo Run

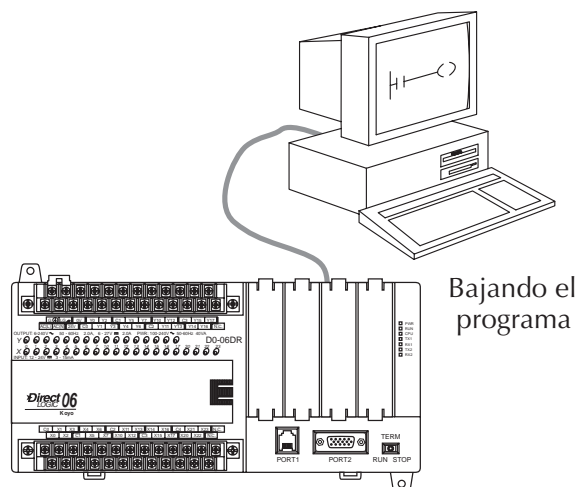
En modo RUN, la CPU ejecuta el programa de uso y actualiza el sistema de entradas y salidas. Se pueden realizar muchas operaciones durante modo RUN. Estas incluyen:

- Supervisar y cambiar el estado de E/S
- Cambiar valores prefijados de temporizadores y contadores
- Cambiar direcciones de memoria

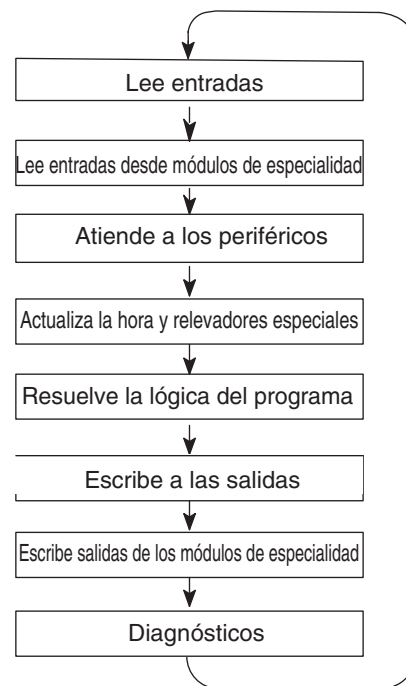
La operación del modo RUN se puede dividir en varias áreas claves. Para la mayoría de usos, algunos de estos segmentos de ejecución son más importantes que otros. Por ejemplo, se necesita entender cómo la CPU actualiza las E/S, fuerza operaciones y soluciona el programa de uso. Los segmentos restantes no son tan importante para la mayoría de los usos.

Ud. puede usar el conmutador de modo en el PLC o *DirectSOFT* o el programador D2-HPP para colocar la CPU en modo RUN.

Usted puede también modificar el programa durante modo RUN. Las modificaciones en el modo RUN no "afectan" a las salidas. En vez de apagarlas, la CPU mantiene las salidas en su estado pasado mientras acepta la nueva información del programa e ignora las entradas. Si se encuentra un error en el nuevo programa, entonces la CPU apagará a todas las salidas y entrará en el modo Program. Esta característica se discute más detalladamente en el capítulo 9.



Barrido normal del modo RUN



ADVERTENCIA: Solamente personal autorizado completamente familiar con todos los aspectos de uso debe realizar cambios al programa. Los cambios durante modo RUN se aplican inmediatamente. Considere seriamente el impacto de cualquier cambio para reducir al mínimo el riesgo de daños corporales o al equipo.

Leyendo entradas

La CPU lee el estado de todas las entradas, luego almacena el estado en la memoria imagen. Las direcciones de memoria imagen de entradas se designan con una X seguido por una dirección de memoria. Los datos de la memoria imagen son utilizados por la CPU cuando se resuelven los algoritmos del programa.

Por supuesto, el estado de una entrada puede cambiar después de que la CPU haya leído las entradas. Generalmente, el tiempo del barrido de la CPU se mide en milisegundos. Si se tiene un uso que no pueda esperar hasta la actualización siguiente de E/S, se pueden usar instrucciones inmediatas. Éstas no utilizan el estado de la memoria imagen de entradas para solucionar el programa. Las instrucciones inmediatas leen inmediatamente el estado de la entrada directamente de los módulos de E/S. Sin embargo, esto alarga el barrido del programa ya que la CPU tiene que leer el estado de la entrada de nuevo. Se incluye una lista completa de instrucciones inmediatas en el capítulo 5.

Comunicación con aparatos periféricos y forzar E/S

Después de que la CPU lea las entradas de los módulos de entrada, lee cualquier dispositivo periférico conectado. Esto es, comunicaciones con cualquier periférico. Por ejemplo, leería un dispositivo de programación para ver si cualquier estado de entradas, salidas u otro tipo de memorias necesita ser modificado.

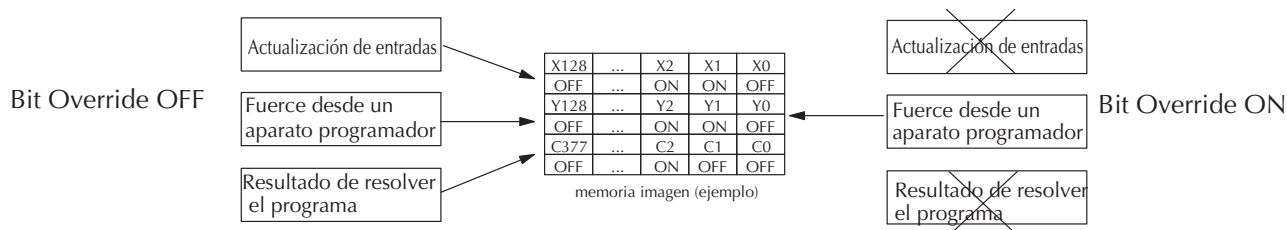
Hay dos tipos básicos de forzar E/S disponibles en el DL06:

- Forzando desde un periférico - no es permanente, bueno solamente por un barrido.
- Bit override - Mantiene una E/S en el estado corriente. Los bits válidos son X, Y, C, T, CT, y S. (estos tipos de memoria se discuten detalladamente posteriormente en este capítulo).

Forzado regular— este tipo puede cambiar temporalmente el estado de un bit. Por ejemplo, puede desear forzar una entrada a ON, aunque está realmente OFF. Esto permite que se cambie el estado que fue almacenado en la memoria imagen. Este valor será válido hasta que la dirección de la memoria imagen se escriba durante el barrido siguiente. Esto es sobre todo útil durante pruebas cuando se necesita forzar un bit a ON para accionar otro evento.

Bit Override — El modo bit override puede ser activado bit a bit usando AUX 59 con el programador D2-HPP o con una opción de menú dentro de *DirectSOFT*[™]. El bit override inhabilita básicamente cualquier cambio al bit por la CPU. Por ejemplo, si se activa el bit override para X1 y X1 está OFF en ese momento, entonces la CPU no cambiará el estado de X1. Esto significa que incluso si X1 se cambia a ON, la CPU no reconocerá el cambio. Así pues, si usted utilizara X1 en el programa, sería evaluada siempre "OFF" en este caso. Por supuesto, si X1 fuera encendido cuando el modo bit override está activado, X1 sería evaluado siempre como "ON".

Hay una ventaja cuando se utiliza la característica de bit override. El forzado regular no es desactivado cuando se activa bit override. Por ejemplo, si se activa el modo bit override en Y0 y estuviera OFF en ese momento, entonces la CPU no cambiaría el estado de Y0. Sin embargo, usted puede aún usar un dispositivo de programación para cambiar el estado. Ahora, si usted utiliza el dispositivo de programación para forzar Y0 a ON, permanecerá ON y la CPU no cambiará el estado de Y0. Si luego fuerza Y0 a OFF, la CPU mantendrá Y0 como apagada. La CPU nunca actualizará el bit con los resultados del programa o de la actualización de E/S hasta que se quite el bit override. El diagrama siguiente muestra una breve descripción de la característica del bit override. Note que la CPU no actualiza la memoria imagen cuando se activa el bit override.



ADVERTENCIA: Solamente personal autorizado completamente familiar con todos los aspectos del uso debe realizar cambios al programa. Asegúrese de considerar seriamente el impacto de cualquier cambio para reducir al mínimo el riesgo de daños corporales o del equipo.

Comunicación por la barra de la CPU

Es posible transferir datos a y desde la CPU por la barra de la CPU en la placa trasera de la base. Estos datos son más que estados de E/S. Este tipo de comunicaciones puede ocurrir solamente en la base (local) de la CPU. Hay una porción del ciclo de ejecución usado para comunicarse con estos módulos. La CPU ejecuta los pedidos de lectura y escritura durante este segmento.

Por ejemplo, los módulos opcionales tales como H0-CTRIO o F0-CP128 intercambian datos directamente con la CPU usando esta barra de comunicación en la placa trasera del PLC.

Actualización de la hora y fecha, relevadores especiales y memorias especiales.

Los PLCs DL06 tienen un reloj y calendario en tiempo real interno (hora y fecha con precisión de hasta segundos) que es accesible al programa de uso. Las direcciones especiales de memoria V llevan a cabo esta información. Esta porción de ciclo de ejecución se asegura de que estas memorias se actualizan en cada barrido. También, hay varios relevadores especiales, tales como relevadores de diagnóstico, etc., que también se actualizan durante este segmento.

Resolviendo el programa de uso

La CPU evalúa cada instrucción en el programa de uso durante este segmento del ciclo de barrido. Las instrucciones definen la relación entre las condiciones de entradas y la respuesta deseada de salidas.

La CPU utiliza el área de la memoria imagen de salida para almacenar el estado de acciones deseadas para las salidas.

Las direcciones de memoria imagen de salida se designan con una Y seguida por una dirección de memoria.

Las salidas reales son actualizadas durante el segmento de escritura de las salidas del ciclo de barrido.

Hay instrucciones de salidas inmediatas disponibles que actualizarán las salidas inmediatamente en vez de esperar hasta que venga el segmento de escritura.

Se muestra una lista completa de las instrucciones inmediatas en el capítulo 5.

Los relevadores internos de control (C), las etapas (S) y la memoria variable (V) también se actualizan en este segmento.

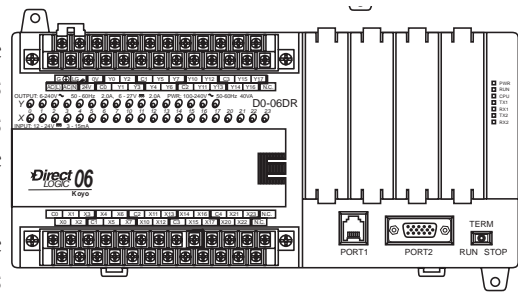
Usted puede recordar que se puede forzar varios tipos de elementos en el sistema (esto fue discutido anteriormente en este capítulo).

Si cualquiera entrada o salida o datos se han forzado en la memoria, la memoria imagen también contiene esta información.

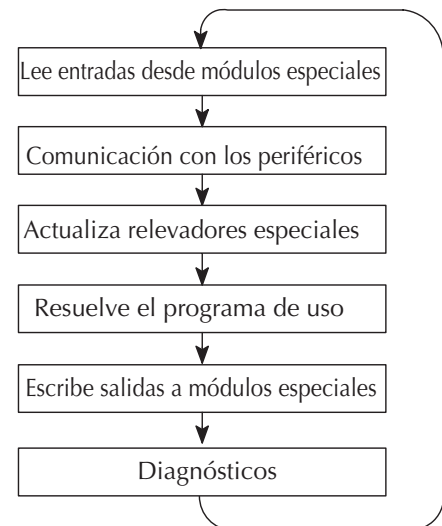
Resolviendo los algoritmos de lazos de control PID

EL PLC DL06 puede procesar hasta 8 lazos de control PID. Los cálculos de lazos funcionan como tarea separada de la ejecución de programa ladder, inmediatamente después de ella. Solamente se calculan los lazos que se han configurado y solamente según un planificador incorporado de lazos.

El tiempo de muestreo (intervalo de cálculo) de cada lazo es programable. Vea por favor el capítulo 8, operación de lazos de control PID, para más información de los efectos de cálculos de lazos de control PID sobre el tiempo total del barrido de la CPU.



Barrido normal en modo RUN



Escribiendo a las salidas

Una vez que el programa haya solucionado la lógica de las instrucciones y haya construido la memoria imagen de salidas, la CPU escribe el contenido de la memoria imagen de salidas a los bits correspondientes de salidas. Recuerde, la CPU también se aseguró de que cualquier cambio de la operación de forzar fue almacenado en la memoria imagen de salida, así que los puntos forzados siguen actualizados con el estado especificado anteriormente.

Escribiendo salidas a módulos de especialidad

Después de la CPU actualiza las salidas en las bases local y de expansión, envía la información de salida que es requerida por cualquier módulos de especialidad que estén instalados.

Diagnósticos

Durante esta parte del barrido, la CPU realiza todo el diagnóstico del sistema y otras tareas tales como calcular el tiempo de barrido y vuelve a 0 el tiempo del “watchdog”. Hay muchas condiciones de error que se detectan automáticamente. El apéndice B del manual del PLC DL06 contiene un listado de los códigos de error.

Probablemente una de las cosas más importantes que ocurre durante este segmento es el cálculo del tiempo de barrido y el control del tiempo de “watchdog”. El DL06 tiene un contador de tiempo de “watchdog” que almacena el máximo tiempo permitido para que la CPU termine el segmento de resolución del ciclo de barrido. Si se excede éste, la CPU entrará en el modo Program y apagará todas las salidas. El valor prefijado de fábrica es 200 ms. El error es informado automáticamente. Por ejemplo, el programador D2-HPP exhibiría el mensaje siguiente "E003 S/W TIMEOUT" cuando ocurre un barrido más largo que ese tiempo.

Usted puede utilizar AUX 53 para ver el tiempo mínimo, máximo y actual de barrido. Utilice AUX 55 para aumentar o para disminuir el valor del contador de tiempo de “watchdog”.

Tiempo de respuesta de entradas y salidas

¿Es importante el tiempo en su proceso?

El tiempo de respuesta de E/S es el tiempo requerido por el sistema de control para detectar un cambio en un punto de entrada y para actualizar un punto correspondiente de salida. En la mayoría de los casos, la CPU realiza esta tarea en un período de tiempo tan corto que no importa en relación con la sincronización del sistema. Sin embargo, algunos casos requieren tiempos extremadamente rápidos de actualización. En estos casos, usted puede necesitar determinar la cantidad de tiempo pasada durante los varios segmentos de la operación.

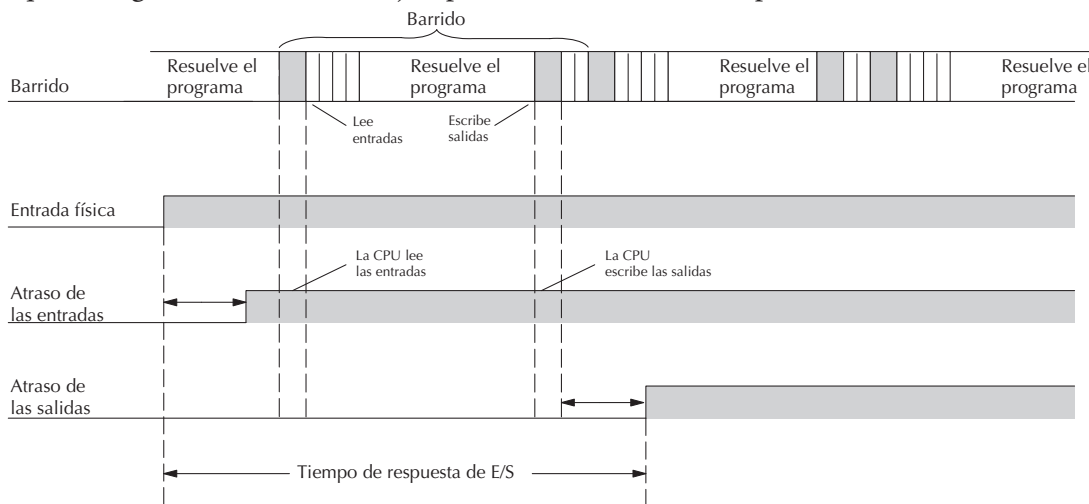
Hay cuatro factores que pueden afectar el tiempo de respuesta de entradas y salidas.

- El momento en el ciclo de barrido en que la entrada cambia estados.
- El tiempo de atraso de cuando la entrada cambia de OFF para ON
- El tiempo de barrido de la CPU
- El tiempo de atraso de cuando la salida cambia de OFF para ON

Los párrafos siguientes muestran cómo estos factores actúan entre ellos para afectar el tiempo de respuesta.

Respuesta de entradas y salidas normal

El tiempo de respuesta de entradas y salidas es más corto cuando las entradas cambian justo antes de la porción de lectura de la tabla X del ciclo de ejecución. En este caso se lee los estados de las entradas, se resuelve el programa de la aplicación, y luego se actualiza los puntos de salidas. El esquema siguiente muestra un ejemplo de la sincronización para esta situación.

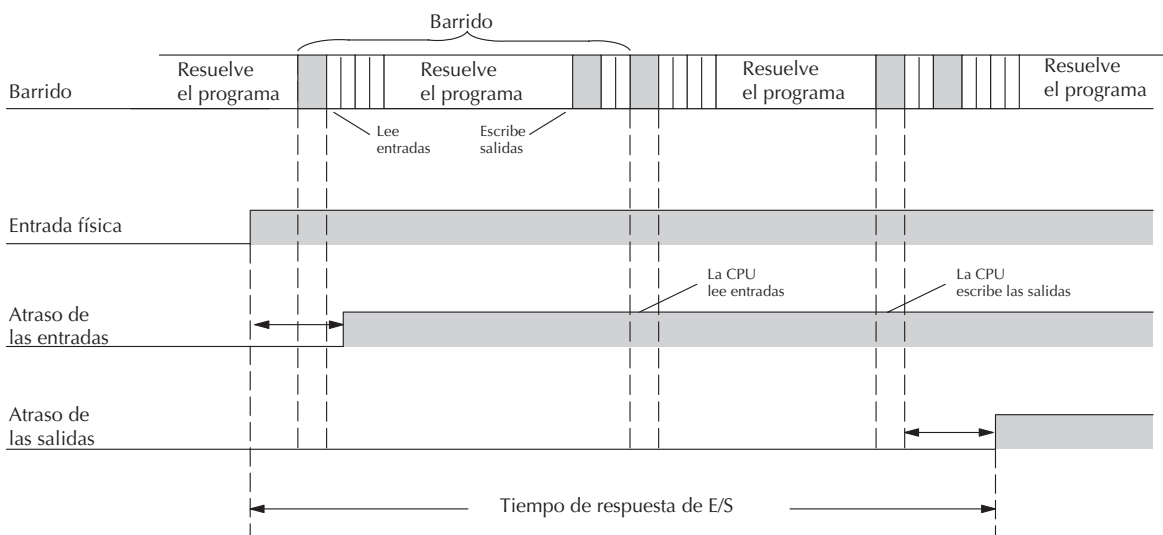


En este caso, usted puede calcular el tiempo de respuesta simplemente sumando los tiempos siguientes:

$$\text{Atraso de la entrada} + \text{tiempo de barrido} + \text{atraso de la salida} = \text{Tiempo de respuesta}$$

Respuesta normal máxima de entradas y salidas

El tiempo de respuesta de entradas y salidas es más largo cuando las entradas cambian justo después de la porción del ciclo de la ejecución que lee las entradas. En este caso el nuevo estado de las entradas no se lee hasta que el siguiente barrido. El esquema siguiente muestra un ejemplo para esta situación.



En este caso, usted puede calcular el tiempo de respuesta simplemente sumando lo siguiente:

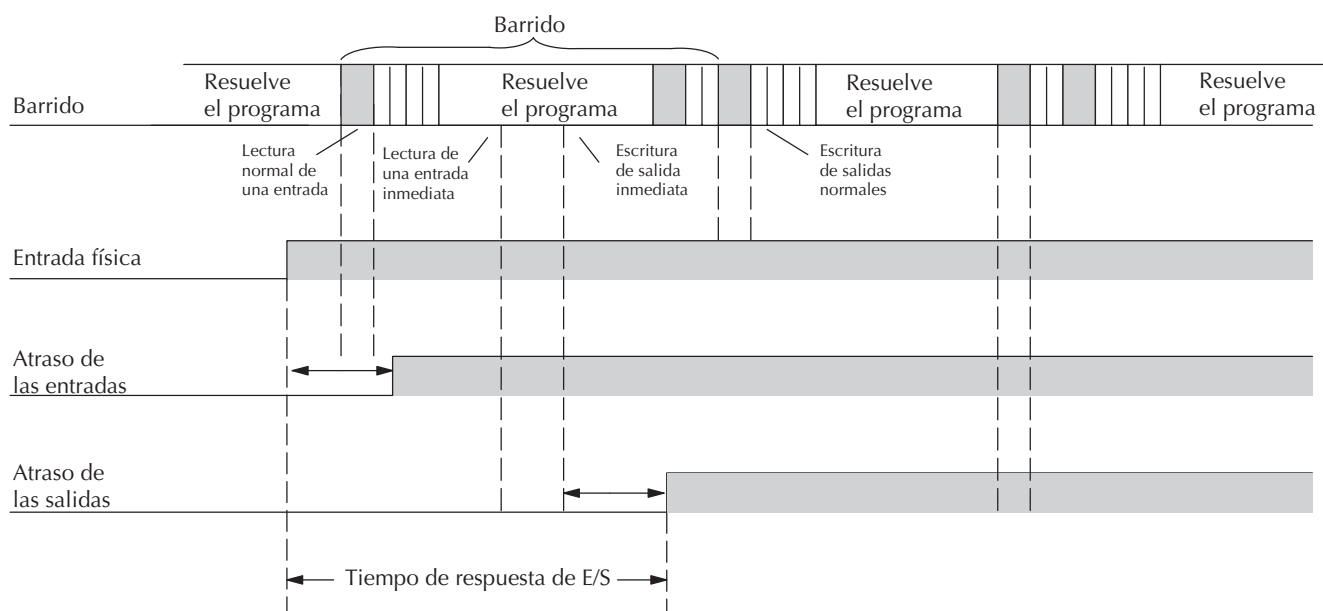
$$\text{Atraso de entradas} + (2 \times \text{tiempo de barrido}) + \text{atraso de salidas} = \text{Tiempo de respuesta}$$

Mejorando el tiempo de respuesta

Hay algunos procedimientos que usted puede hacer para ayudar a mejorar el rendimiento.

- Puede escoger instrucciones con tiempos más rápidos de ejecución
- Puede usar instrucciones inmediatas de entradas y salidas (que actualizan los puntos de entradas y salidas durante la ejecución del programa)
- Puede usar las características del modo 50-Captura de pulsos, diseñadas para operar en ambientes de alta velocidad. Vea el capítulo 3 para más detalles.
- O puede cambiar el filtro en el modo 60 a 0 ms. para X0, para X1, para X2 y para X3.

De estos tres puntos, las instrucciones inmediatas de entradas y salidas son probablemente las más importantes y útiles. El ejemplo siguiente muestra como una instrucción inmediata puede afectar el tiempo de respuesta.



En este caso, usted puede calcular el tiempo de respuesta simplemente sumando los tiempos siguientes:

$$\text{Atraso de entradas} + \text{tiempo de ejecución de la instrucción} + \text{atraso de salidas} = \text{Tiempo de respuesta}$$

El tiempo de ejecución de la instrucción podría ser calculado sumando el tiempo de la instrucción inmediata de entrada, la instrucción inmediata de salida y cualquier otra instrucción entre las dos.



NOTA: Aunque la instrucción inmediata lee la posición más actual de E/S, sólo usa los resultados para resolver aquella instrucción. No usa el estado nuevo para actualizar la memoria imagen. Por lo tanto, cualquier otra instrucción regular usará los valores de memoria imagen. Cualquier instrucción inmediata que siga conseguirá acceso a las entradas y salidas otra vez para actualizar el estado.

Consideraciones del tiempo de barrido de la CPU

El tiempo de barrido cubre todas las tareas cíclicas que son realizadas por el sistema operativo. Usted puede usar *DirectSOFT* o el programador D2-HPP para mostrar el tiempo mínimo, máximo y corriente del barrido que ha ocurrido desde la transición anterior del modo de programa al modo RUN. Esta información puede ser muy importante al evaluar el funcionamiento de un sistema. Como mostrado anteriormente, hay varios segmentos que hacen el ciclo de barrido. Cada uno de estos segmentos requiere cierto tiempo para completarse.

De todos los segmentos, los siguientes son los más importantes:

- Actualización de entradas
- Atendimento a los periféricos
- Ejecución del programa
- Actualización de las salidas
- Ejecución de interrupciones por tiempo

El segmento en el que se tiene más control es la cantidad de tiempo tomada para ejecutar el programa. Esto es porque cada instrucción toma un determinado tiempo para ejecutarse. Así pues, si hay necesidad de un barrido más rápido, debe tratar de escoger instrucciones más rápidas.

La selección del tipo de E/S y de los dispositivos periféricos puede también afectar el tiempo de barrido. Sin embargo, esta selección es dictada generalmente por el uso.

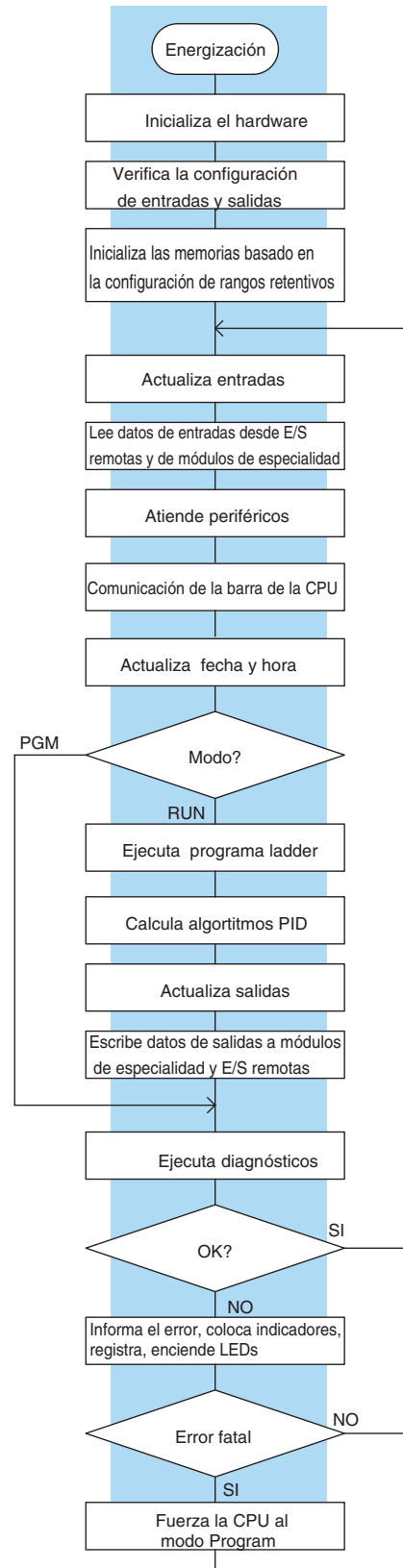
Los párrafos siguientes dan información de carácter general de cuánto tiempo pueden requerir algunos de los segmentos.

Leyendo entradas

El tiempo requerido durante cada barrido para leer el estado de las 20 entradas locales es 52.6 μ s. No confunda esto con el tiempo de respuesta de las E/S que fue discutido anteriormente.

Escribiendo a salidas

El tiempo requerido para escribir el estado de 16 salidas incorporadas es 41.1 μ s. No confunda esto con el tiempo de respuesta de E/S que fue discutido anteriormente.



Atendiendo los aparatos periféricos

Pueden ocurrir peticiones de comunicación en cualquier momento durante el barrido, pero la CPU "registra" solamente los pedidos por atendimento hasta el segmento de servicio a los periféricos en el barrido. La CPU no gasta tiempo en esto si no hay periféricos conectados.

Para registrar peticiones (en cualquier momento)		DL06
Nada está conectado	Min. & Max	0 μ s
Puerto 1	Enviar Min. / Max.	5,8/11,8 μ s
	Recibir. Min. / Max.	12,5/25,2 μ s
Puerto 2	Enviar Min. / Max.	6,2/14,3 μ s
	Recibir. Min. / Max.	14,2/31,9 μ s
LCD	Min. / Max.	4,8/49,2 μ s

Durante el segmento de atendimento a los periféricos en el barrido, la CPU analiza la petición de comunicación y responde como sea adecuado. El tiempo requerido para atender los periféricos depende del contenido de la petición.

Para atender la petición DL06	DL06
Mínimo	9 μ s
Modo Run máximo	412 μ s
Modo Program máximo.	2,5 segundos

Comunicación de la barra de la CPU

Algunos módulos de especialidad pueden también comunicarse directamente con la CPU por la barra de la CPU. Durante esta porción del ciclo la CPU completa cualquier comunicación por la barra de la CPU. El tiempo requerido depende del tipo de módulos instalados y del tipo de petición que está siendo procesada.

Actualizando la hora y la fecha, relevadores especiales, memorias especiales (o dedicadas)

La hora, la fecha y los relevadores especiales son actualizados y cargados en direcciones especiales de la memoria V durante este tiempo. Esta actualización se realiza durante los modos RUN y Program.

Modos		DL06
Modo Program	Mínimo	12.0 μ s
	Máximo	12.0 μ s
Modo Run	Mínimo	20.0 μ s
	Máximo	27.0 μ s



NOTA: El Calendario y hora es actualizado mientras haya energía en el super condensador. Si el supercondensador se descarga, la hora y la fecha se pierden.

Ejecución de un programa

La CPU procesa el programa desde la dirección 0 hasta la instrucción END. La CPU ejecuta el programa desde la izquierda a la derecha y de arriba para abajo. Mientras se evalúa cada renglón se actualiza la memoria imagen o la dirección de memoria adecuada. El tiempo requerido para resolver el programa depende del tipo y de la cantidad de instrucciones usadas, y de la cantidad de "tiempo adicional" de ejecución.

Para determinar el tiempo de ejecución sume los tiempos de ejecución de todas las instrucciones en su programa. El apéndice C tiene una lista completa de tiempos de ejecución de cada instrucción del PLC DL06. Por ejemplo, el tiempo de ejecución del programa mostrado abajo se calcula como sigue:

Instrucción	Tiempo
STR X0	0,67 µs
OR C0	0,51 µs
ANDN X1	0,51 µs
OUT Y0	1,82 µs
STRN C100	0,67 µs
LD K10	9,00 µs
STRN C101	0,67 µs
OUT V2002	9,3 µs
STRN C102	0,67 µs
LD K50	9,00 µs
STRN C103	0,67 µs
OUT V2006	1,82 µs
STR X5	0,67 µs
ANDN X10	0,51 µs
OUT Y3	1,82 µs
END	12,80 µs
<hr/>	
SUBTOTAL	51,11 µs
<hr/>	
Tiempo adicional DL06	
Mínimo	746,2 µs
Máximo	4352,4 µs

$$\text{Tiempo Total} = (\text{Tiempo de ejecución del programa} + \text{tiempo adicional}) \times 1.18$$

El programa toma solamente 51,11 µs para ejecutarse durante cada barrido. El DL06 gasta 0,18 ms en la administración de interrupción por tiempo por cada 1 ms de tiempo de barrido. El tiempo total de barrido es calculado sumando el tiempo de ejecución de programa a los gastos indirectos (mostrados arriba) y multiplicando el resultado (ms) por 1,18. El "tiempo adicional" (o gastos indirectos) incluye tareas internas y de diagnóstico.

El tiempo de barrido variará levemente a partir de un barrido al siguiente, debido a la fluctuación en las tareas de gastos indirectos.

Instrucciones de control del programa — El PLC DL06 tiene instrucciones adicionales que pueden cambiar la manera en que el programa se ejecuta. Estas instrucciones incluyen lazos de FOR/NEXT, subrutinas y rutinas de interrupción. Estas instrucciones pueden interrumpir el flujo de programa normal y afectar el tiempo de ejecución del programa. El capítulo 5 entrega información detallada en cómo funcionan estos diversos tipos de instrucciones.

Sistemas numéricos del PLC

Si usted es un nuevo usuario del PLC o está utilizando PLCs *AutomationDirect* por la primera vez, por favor tome un momento para estudiar cómo nuestros PLCs usan los números. Usted encontrará que cada fabricante de PLCs tiene sus propias convenciones relacionadas con el uso de números en el PLC. La información que usted aprenderá aquí se aplica a todos nuestros PLCs!

Como cualquier buena computadora, los PLCs almacenan y manipulan números en forma binaria: solamente 1s y 0s. Entonces ¿porqué tenemos que ocuparnos de números en tan diversas formas? Los números tienen significado, y algunas representaciones son más convenientes que otras para algunos propósitos particulares. Usamos a veces números para representar un tamaño o una cantidad de algo. Otros números se refieren a localizaciones o a direcciones, o para medir el tiempo. En ciencia unimos unidades de ingeniería a números para dar un significado particular.

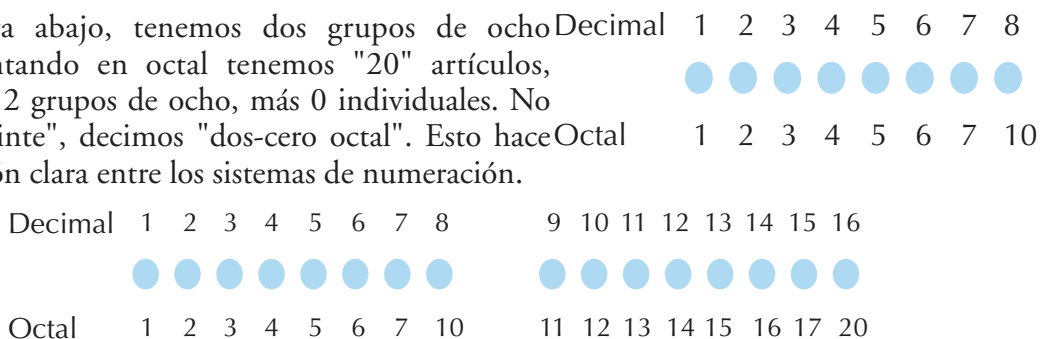
Hemos creado el apéndice I para describir los sistemas numéricos que son usados en este PLC. Por favor, lea ese apéndice para más información.

Recursos del PLC

Los PLCs tienen una cantidad fija de recursos, dependiendo del modelo y de la configuración. Usamos la palabra "recursos" para incluir la memoria variable (memoria V), puntos de E/S, temporizadores, contadores, etc. La gran mayoría de PLCs modulares le permite agregar módulos de E/S en grupos de ocho. De hecho, todos los recursos de nuestros PLCs se cuentan en octal. Es más fácil que las computadoras cuenten en grupos de ocho que diez, porque ocho es una potencia de 2.

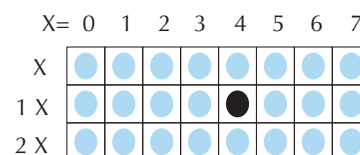
Octal significa contar en grupos de ocho cosas a la vez. En la figura a la derecha, hay ocho círculos. La cantidad en decimal es "8", pero en octal es "10" (8 y 9 son inválidos en octal). En octal, "10" significa 1 grupo de 8 más 0 (sin unidades).

En la figura abajo, tenemos dos grupos de ocho círculos. Contando en octal tenemos "20" artículos, significando 2 grupos de ocho, más 0 individuales. No decimos "veinte", decimos "dos-cero octal". Esto hace una distinción clara entre los sistemas de numeración.



Después de contar recursos del PLC, es hora de tener acceso a recursos del PLC (hay una diferencia). El sistema de instrucciones de la CPU tiene acceso a los recursos del PLC usando direcciones octales. Las direcciones octales son iguales que cantidades octales, excepto que comienzan a contar en cero. El número cero es significativo a una computadora, así que no lo saltamos.

Nuestros círculos están en un arsenal de envases cuadrados a la derecha. Para tener acceso a un recurso, la instrucción del PLC direccionará la localización usando referencias octales mostradas. Si éstos fueran contadores, "CT14" tendría acceso a la localización negra del círculo



Memoria V

La memoria variable (llamada "memoria V") almacena datos para el programa y para la configuración. Las direcciones de memoria se numeran en octal. Por ejemplo, V2073 es una localización válida, mientras que V1983 es inválido ("9" y "8" son dígitos octales inválidos).

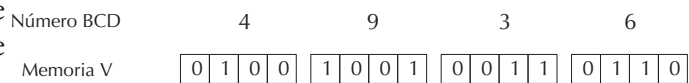
Cada dirección de memoria V es una palabra de datos, conteniendo 16 bits. Para configuración, nuestros manuales muestran cada bit de una palabra de la memoria V. El bit menos significativo (LSB) estará a la derecha y el bit más significativo (MSB) a la izquierda. Utilizamos la palabra "significativa", refiriendo al peso binario relativo de los bits.

Los datos de memoria V son binarios de 16 bits, pero raramente programamos a las memorias de datos un bit a la vez. Usamos instrucciones o herramientas que nos dejan trabajar con números decimales, octales, y hexadecimales. Todos éstos se convierten y se almacenan como binarios.

Una pregunta frecuente es "cómo puedo decir si un número es octal, BCD, o hexadecimal"? La respuesta es que no podemos responder mirando los datos... pero realmente no importa. Lo que importa es: la fuente o el mecanismo que escribe datos en una dirección de memoria y la cosa que la lee más adelante deben ambas usar el mismo tipo de datos (es decir, octal, hexadecimal, binaria, o lo que sea). La dirección de memoria V es solo una caja de almacenaje... sólo eso. No convierte ni mueve los datos.

Números BCD

Ya que los seres humanos contamos en decimal (10 dedos, 10 dedos del pie), preferimos manejar y ver datos del PLC en decimal también. Sin embargo, las computadoras son más eficientes usando números binarios puros. Una solución de compromiso entre los dos es la representación BCD. Undígito BCD va de 0 a 9 y se almacena como cuatro bits binarios (un nibble). Esto permite que cada dirección de memoria V almacene cuatro dígitos BCD, con un rango de números decimales de 0000 a 9999.



En un sentido binario puro, una palabra de 16 bits puede representar números a partir de 0 a 65535. Al almacenar números BCD, el rango se reduce a solamente 0 a 9999. Muchos datos usan números BCD, las instrucciones aritméticas (BCD) y *DirectSOFT* y el programador permiten que entremos y que veamos datos en BCD.

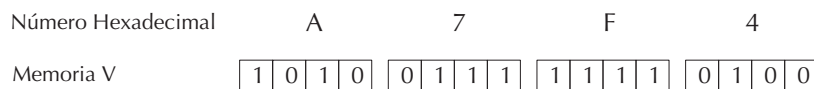
Números hexadecimales

Los números hexadecimales son similares a los números BCD, excepto que utilizan todos los valores binarios posibles en cada dígito de 4 bits. Son los números en base 16 así que necesitamos 16 dígitos.

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

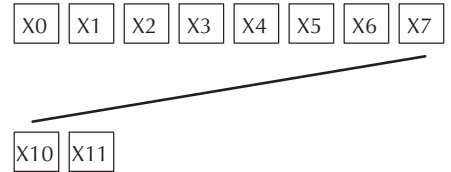
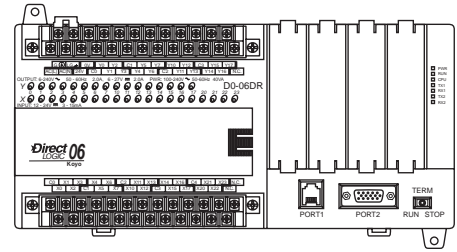
Para extender nuestros dígitos decimales 0 a 9, utilizamos A a F según lo mostrado.

Un número hexadecimal de 4 dígitos puede representar los 65536 valores en una palabra de memoria V. El rango es 0000 a FFFF (hexadecimal). Los PLCs a menudo necesitan este rango completo para datos de sensores, etc. Hexadecimal es solamente una manera conveniente para que los seres humanos veamos datos binarios completos.



Mapa de memorias

Con cualquier sistema de PLCs se tienen generalmente diversos tipos de información a procesar. Esto incluye estados de dispositivos de entradas, estado de dispositivos de salidas, varios elementos de sincronización, conteo de piezas, etc. Es importante entender cómo el sistema representa y almacena los diversos tipos de datos. Por ejemplo, usted necesita saber como el sistema identifica señales de entradas, salidas, palabras de datos, etc. Los párrafos siguientes discuten los tipos de memoria usados en los PLCs DL06. Una descripción del mapa de memoria para la CPU sigue a las descripciones de la memoria.



3

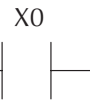
Sistema de numeración octal

Todas las direcciones de memoria y recursos se numeran en octal (base 8). Por ejemplo, el diagrama muestra cómo trabaja el sistema de numeración octal para los puntos discretos de entrada. Note que el sistema octal no contiene ningún número con los dígitos 8 o 9.

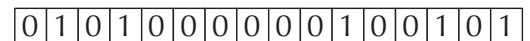
Direcciones discretas y de palabra

Usted notará dos tipos de memoria en el DL06 al observar los tipos de memoria, discretos y memoria de palabra. La memoria discreta es un bit que puede ser un 1 o un 0. La memoria de palabra es referida como memoria V (variable) y es una dirección de 16 bits usada normalmente para manipular datos o números, etc.

Discreto – On u OFF, 1 bit



Dirección de memoria V de 16 bits



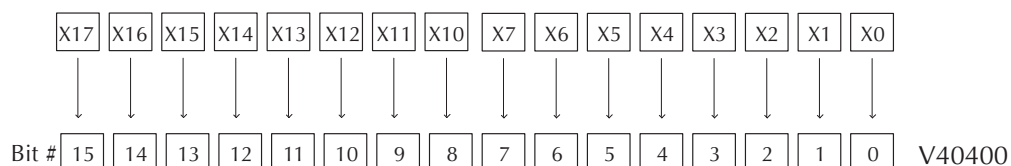
Algunas informaciones se almacenan automáticamente en la memoria V. Por ejemplo, los valores corrientes de un temporizador se almacenan en memoria V.

Direcciones de memoria V para las áreas de memoria discretas

El área de memoria discreta es usada por las entradas, las salidas, los relevadores de control, los relevadores especiales, las etapas, los bits de estado del temporizador y los bits de estado de contadores. Sin embargo, se puede tener acceso también a tipos de datos del bits como palabra de memoria V. Cada dirección de memoria V contiene 16 direcciones discretas consecutivas. Por ejemplo, el diagrama siguiente muestra cómo los puntos de entrada X corresponden a una dirección de memoria V.

Estas áreas de memoria discretas y sus rangos correspondientes de memoria V se enumeran en la tabla del área de memoria de los PLCs DL06 en las páginas siguientes.

16 puntos de entradas discretas (X)



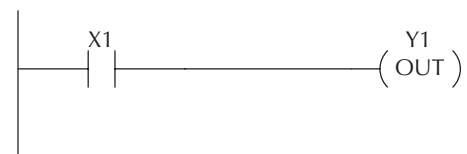
Puntos de entradas (Datos tipo X)

Los puntos discretos de entrada son denotados por un tipo de datos X. Hay 20 puntos discretos de entrada y 256 direcciones distintas de entradas disponibles con los PLCs DL06. En este ejemplo, la salida Y0 se activará cuando X0 esté activada.



Puntos de salidas (Datos tipo Y)

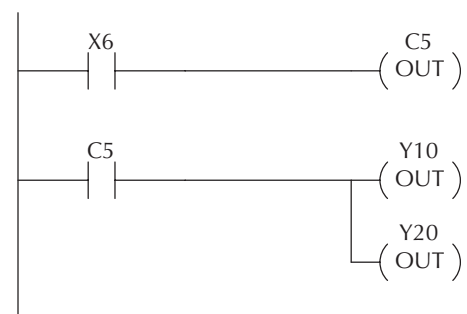
Los puntos discretos de salida son denotados por un tipo de datos Y. Hay 16 salidas distintas y 256 direcciones distintas de salidas disponibles en el PLC DL06. En este ejemplo, la salida Y1 se activará cuando X1 esté activada.



Relevadores de control (Datos tipo C)

Los relevadores de control son bits discretos que se usan normalmente para controlar el programa de usuario. Los relevadores de control no representan un aparato verdadero, esto es, no puede ser relacionado a interruptores, bobinas de salida, etc. Son internos en la CPU. A causa de esto, los relevadores de control se pueden programar como entradas o salidas discretas. Estas direcciones son usadas para programar direcciones (C) discretas de memoria o la dirección correspondiente de palabra que contiene 16 direcciones discretas consecutivas.

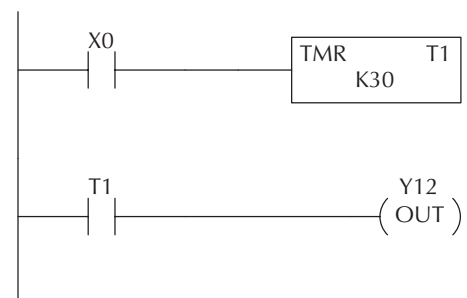
En este ejemplo, la memoria C5 se energizará cuando la entrada X6 se active. El 2o. renglón muestra un ejemplo simple de como usar un relevador de control como entrada discreta.



Temporizadores y bits de estado (Datos tipo T)

Los bits de estado de temporizadores muestran la relación entre el valor corriente (el valor corriente es el valor en el momento actual, si es que podemos imaginar que el temporizador tiene un reloj que cuenta tiempo en una base de tiempo) y el valor prefijado de un temporizador especificado.

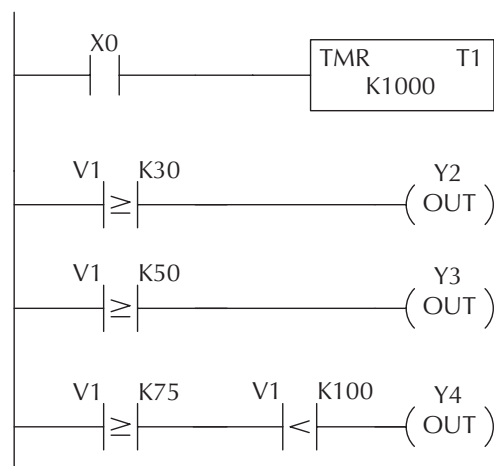
Cuando se activa la entrada X0, el temporizador T1 comienza a contar el tiempo. Cuando el temporizador alcanza el valor prefijado de 3 segundos (K 30) se activa el contacto de estado del temporizador T1. Cuando se activa T1, se activa la salida Y12. Desactivando X0 se repone el temporizador a 0.



Valores corrientes del temporizador (Datos del tipo V)

Como mencionado anteriormente, alguna información se almacena automáticamente en la memoria V. Esto es verdad para los valores corrientes asociados con temporizadores. Por ejemplo, V0 tiene el valor corriente para el temporizador 0, V1 tiene el valor corriente para el temporizador 1, etc. Estos pueden ser designados también como TA0 para el temporizador 0, y TA1 para el temporizador 1.

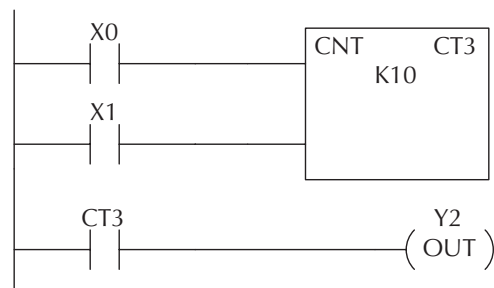
La razón primaria para esto es la flexibilidad del programa. El ejemplo adyacente muestra cómo usted puede usar contactos de comparadores para controlar varias veces intervalos de un solo temporizador.



Contadores y bits de estado de contadores (Datos tipo CT)

Los bits de estado entregan una relación entre el valor corriente y el valor prefijado de un contador especificado. El bit de estado de un contador estará ON cuando el valor corriente es igual a o mayor que el valor prefijado del contador correspondiente.

En el ejemplo, cada vez que hay una transición del contacto X0 de OFF a ON, el contador incrementa el valor corriente en uno. (Si X1 se hace ON, el valor corriente del contador vuelve a 0). Cuando el contador alcanza el valor prefijado de 10 (K10) el bit de estado CT3 prende. Cuando CT3 prende, la salida Y2 se activa.

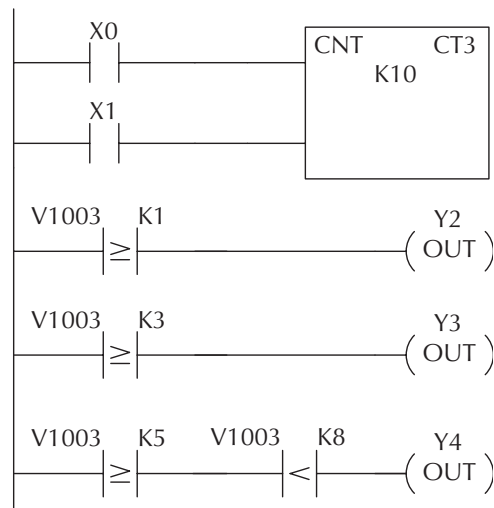


Valores corrientes de un contador (Datos tipo V)

Los valores corrientes de un contador también se almacenan automáticamente en la memoria V, como los valores de un temporizador. Por ejemplo, V1000 tiene el valor corriente del contador CT0, V1001 tiene el valor corriente del contador CT1, etc.

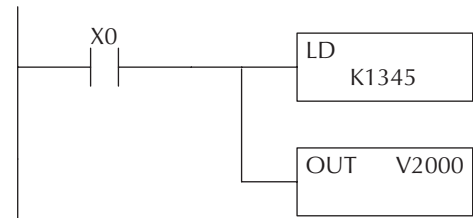
Estos pueden ser designados también como CTA0 (Contador acumulador) para el contador 0 y CTA1 para el contador 1.

La primera razón para esto es una flexibilidad de programación. El ejemplo muestra cómo se pueden usar los contactos de comparación para supervisar los valores corrientes del contador.

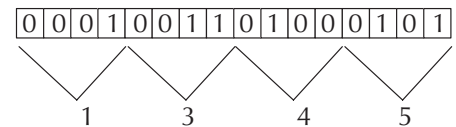


Memoria de palabra (Datos tipo V)

La memoria de palabra o también memoria V es una localización de 16 bits de memoria usada normalmente para manipular números o datos, almacenar estos, etc. Alguna información se almacena automáticamente en la memoria V. Por ejemplo, los valores corrientes de un temporizador se almacenan en la memoria V. El ejemplo muestra cómo una constante de cuatro dígitos BCD se carga en el acumulador y entonces se almacena en una localización de memoria V.



Palabra de 16 bits

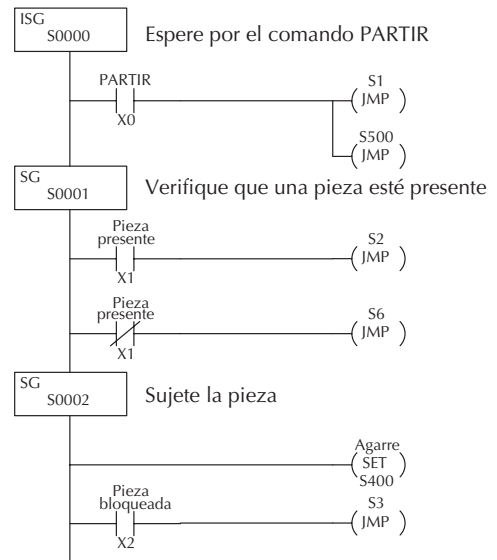


Las etapas (Datos tipo S)

Las Etapas se usan en programas RLL^{PLUS} para crear un programa estructurado, semejante a un organigrama. Cada etapa del programa denota un segmento del programa. Cuando la etapa (el segmento de programa) es activa, se ejecuta la lógica dentro de ese segmento. Si la Etapa está apagada, o inactiva, la lógica no se ejecuta y la CPU se salta a la Etapa activa siguiente. (Vea el capítulo 7 para más detalles).

Cada Etapa tiene también un bit discreto de estado que se puede usar como una entrada para indicar si la Etapa está activada o inactiva. Si la Etapa está activa, entonces el bit de estado está ON. Si la Etapa está inactiva, entonces el bit de estado está apagado. Este bit de estado puede ser prendido o apagado también por otras instrucciones, tal como las instrucciones SET y RESET. Esto permite controlar fácilmente las etapas a través del programa.

Representación Ladder



Los relevadores especiales (Datos tipo SP)

Los relevadores especiales son las localizaciones discretas de memoria con una función ya definida y dedicada. Hay muchos tipos diferentes de relevadores especiales. Por ejemplo, algunos ayudan en el desarrollo del programa, otros entregan información del estado de la operación del sistema, etc.

En este ejemplo, el relevador de control C10 se activará por 50 ms y se desactivará en los siguientes 50 ms porque SP5 es un relevador dedicado que causa un pulso de 50ms ON y 50 ms OFF.



SP4: 1 segundo
 SP5: 100 ms
 SP6: 50 ms

Memoria del sistema del PLC DL06

Parámetros del sistema y de datos originales de fábrica (tipo de datos V)

El PLC DL06 reserva varias direcciones de memoria V para almacenar parámetros del sistema o ciertos tipos de datos del sistema. Estas direcciones de memoria almacenan datos como códigos de error, datos de alta velocidad de E/S y otros tipos de información de la configuración del sistema.

Memoria del sistema	Descripción del contenido	Valores originales/Rangos
V700-V707	Configura la dirección de memoria para el módulo opcional en la ranura 1	N/A
V710-V717	Configura la dirección de memoria para el módulo opcional en la ranura 2	N/A
V720-V727	Configura la dirección de memoria para el módulo opcional en la ranura 3	N/A
V730-V737	Configura la dirección de memoria para el módulo opcional en la ranura 4	N/A
V3630-V3707	Dirección original de fábrica para los valores prefijados múltiples para contador incremental-decremental y el contador 1 o la función de captura de pulso.	N/A
V3710-V3767	Dirección original de fábrica para los valores prefijados múltiples para contador incremental-decremental y el contador 2 o la función de captura de pulso.	N/A
V7620-V7627	Direcciones para los parámetros de la interface de operador DV-1000. Configura la dirección de memoria V que contiene el valor. configura la dirección de memoria V que contiene el mensaje. Configura el número total (1 - 32) de las direcciones de memoria V a ser mostrada. Configura la dirección de la memoria V que contiene los números a ser mostrados. Configura la dirección de memoria V que contiene el código de carácter para ser mostrado. Contiene el número de la función que se puede asignar a cada tecla. Modo de operación en la energización. Cambia valores prefijados de operación.	V0 - V3760 V0 - V37601 - 32 V0 - V3760 V0 - V3760 Memoria V para X, Y, o C0, 1, 2, 3, 12; Valor original = 0000
V7630	Dirección inicial para los valores prefijados múltiples para el canal 1. El valor prefijado por defecto es 3630, que indica que el primer valor se debe obtener de V3630. Puesto que hay 24 valores prefijados disponibles, el rango es V3630 - V3707. Se puede cambiar la memoria de inicio en caso de necesidad.	Valor original: V3630 Rango: V0- V3710
V7631	Dirección inicial para los valores prefijados múltiples para el canal 2. El valor prefijado por defecto es 3710, que indica que el primer valor se debe obtener de V3710. Puesto que hay 24 valores prefijados disponibles, el rango es V3710 - V3767. Se puede cambiar la memoria de inicio en caso de necesidad.	Valor original: V3710 Rango: V0- V3710
V7632	Memoria o registro de configuración para las salidas de pulsos.	N/A
V7633	Configura el código deseado de función para el contador de alta velocidad, interrupción, la captura de pulsos, la salida de tren de pulsos y el filtro de entradas. La dirección se puede también utilizar para definir en que modo de operación va a partir el PLC cuando se energiza este.	Valor original: 0060 Rango del Byte inferior: 10 - Contador 20 - Cuadratura 30 - Salida de pulsos- 40 - Interrupción 50 - Captura de pulsos 60 - Entrada filtrada. Rango del Byte superior Bits 8-11, 14, 15: No usados, Bit 13: Parte en RUN, si el conmutador de modo está en TERM. Bit 12 es usado para activar la alarma de batería.
V7634	Memoria de configuración para funciones HSIO con la entrada X0	Valor original: 1006
V7635	Memoria de configuración para funciones HSIO con la entrada X1	Valor original: 1006
V7636	Memoria de configuración para funciones HSIO con la entrada X2	Valor original: 1006
V7637	Memoria de configuración para funciones HSIO con la entrada X3	Valor original: 1006
V7640	Dirección de memoria inicial de la tabla de lazos PID	V1200 - V7377 V10000 - V17777
V7641	Cantidad de lazos de control PID	8

Capítulo 4: Especificaciones y operación

3

Memoria del sistema	Descripción del contenido	Valores originales/Rangos
V7642	Código de error - dirección de memoria de error para la tabla de lazos PID	
V7643-V7647	Reservado	
V7650	Puerto 2: Configuración de la dirección de memoria para protocolo non procedure	V1200 – V7377 V10000 - V17777
V7653	Puerto 2: Configuración del código de terminación para protocolo non procedure	
V7655	Puerto 2: Configuración del protocolo, time-out y tiempo de atraso de respuesta	
V7656	Puerto 2: Configuración del número de de la estación, tasa de baud , bit STOP y paridad.	
V7657	Puerto 2: Configuración de código de terminación usado para indicar que se completó la configuración de parámetros	
V7660	Configuración del control de barrido: Mantiene el modo de control del barrido.	
V7661	Contador de configuración de tiempo sobrepasado: Cuenta las veces que el tiempo de barrido corriente sobrepasa el tiempo de configuración definido por el usuario.	
V7662-V7717	Reservado	
V7720-V7722	Direcciones de los parámetros de la interface de operador DV-1000.	
V7720	Puntero de valor prefijado del Titled Timer	
V7721	Puntero de valor prefijado del Titled Counter	
V7722	HiByte-Tamaño del bloque del valor prefijado del Titled Timer, LoByte-Tamaño del bloque del valor prefijado del Titled Counter	
V7723-V7726	Reservado	
V7727	Version de firmware (ejemplo : versión 1.40 => 1400)	
V7730-V7737	Reservado	
V7740	Puerto 1 y 2: Configuración del temporizador de comunicación repuesto automáticamente	Valor original: 3030
V7741-V7746	Reservado	
V7747	Esta dirección contiene un contador de 10 ms (0-99). Aumenta una vez cada 10 ms.	
V7750	Reservado	
V7751	Código de error de falla — almacena un código de 4 dígitos usado con la instrucción FAULT cuando la instrucción es ejecutada.	
V7752	Error de configuración de E/S: Código de identificación corriente de un error de ranura	
V7753	Error de configuración de E/S: Código de identificación antiguo de un error de ranura	
V7754	Error de configuración de E/S: Número de ranura con error	
V7755	Código de error — almacena el código de error fatal.	
V7756	Código de error — almacena el código de error más importante.	
V7757	Código de error — almacena el código de error menos importante	
V7760-V7762	Reservado	
V7763	Dirección de programa donde está el error de sintaxis	
V7764	Código del error de sintaxis.	
V7765	Contador de barridos — Almacena el no. total de ciclos de barrido que han ocurrido desde la última transición del modo Program a RUN.	
V7766	Contiene el número de segundos en la hora (00-59)	
V7767	Contiene el número de minutos en la hora (00-59)	
V7770	Contiene el número de horas en la hora (00-23)	
V7771	Contiene el día de la semana (Lunes, Martes, Miércoles, etc.)	
V7772	Contiene el día del mes (01, 02, etc.)	
V7773	Contiene el mes (01 a 12)	
V7774	Contiene el año (00 a 99)	
V7775	Almacena el tiempo corriente de barrido (milisegundos).	
V7776	Almacena el tiempo mínimo de barrido desde la ultima transición del modo Program a RUN (milisegundos)	
V7777	Almacena el tiempo máximo de barrido desde la ultima transición del modo Program a RUN (milisegundos)	
V37700-V37737	Para remote I/O - Esta función no es apoyada actualmente por Automation Direct	

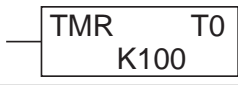
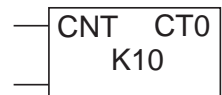

Alias (o apodos) de PLC DL06

Un alias es una manera alternativa de referirse a ciertos tipos de memoria, tales como valores corrientes de temporizadores o contadores, localizaciones de memoria V para algunos puntos de E/S, etc., que simplifica el entender la dirección de memoria. El uso de alias es opcional, pero algunos usuarios pueden encontrar que el alias es provechoso al desarrollar un programa.

La tabla de abajo muestra cómo se pueden utilizar los alias.

Alias de PLC DL06		
Dirección inicial	Alias inicial	Ejemplo
V0	TA0	V0 es el valor corriente del tiempo acumulado del temporizador 0; por lo tanto, el alias correspondiente es TA0. TA1 es el alias para V1, etc.
V1000	CTA0	V1000 es el valor corriente de conteo acumulado del contador 0; por lo tanto, el alias es CTA0. CTA1 es el alias para V1001, etc.
V40000	VGX	V40000 es la referencia de memoria de la palabra de los bits discretos GX0 hasta GX17; por lo tanto, el alias es VGX0. V40001 es la referencia de memoria de la palabra para los bits discretos GX20 hasta GX37, por lo tanto, el alias es VGX20
V40200	VGX	V40200 es la referencia de memoria de la palabra de los bits discretos GY0 hasta GY17; por lo tanto, el alias es VGY0. V40201 es la referencia de memoria de la palabra para los bits discretos Gy20 hasta GY37, por lo tanto, el alias es VGY20
V40400	VX0	V40400 es la referencia de memoria de la palabra de los bits discretos X0 hasta X17; por lo tanto, el alias es VX0. V40401 es la referencia de memoria de la palabra para los bits discretos X20 hasta X37, por lo tanto, el alias es VX20
V40500	VY0	V40500 es la referencia de memoria de la palabra de los bits discretos Y0 hasta Y17; por lo tanto, el alias es VY0. V40501 es la referencia de memoria de la palabra para los bits discretos Y20 hasta Y37, por lo tanto, el alias es VY20
V40600	VCO	V40600 es la referencia de memoria de la palabra de los bits discretos C0 hasta C17; por lo tanto, el alias es VCO. V40601 es la referencia de memoria de la palabra para los bits discretos C20 hasta C37, por lo tanto, el alias es VC20
V41000	VS0	V41000 es la referencia de memoria de la palabra de los bits discretos S0 hasta S17; por lo tanto, el alias es VS0. V41001 es la referencia de memoria de la palabra para los bits discretos S20 hasta S37, por lo tanto, el alias es VS20
V41100	VTO	V41100 es la referencia de memoria de la palabra de los bits discretos T0 hasta T17; por lo tanto, el alias es VTO. V41101 es la referencia de memoria de la palabra para los bits discretos T20 hasta T37, por lo tanto, el alias es VT20
V41140	VCT0	V41140 es la referencia de memoria de la palabra de los bits discretos CT0 hasta CT17; por lo tanto, el alias es VCT0. V41101 es la referencia de memoria de la palabra para los bits discretos CT20 hasta CT37, por lo tanto, el alias es VCT20
V41200	VSP0	V41200 es la referencia de memoria de la palabra de los bits discretos SP0 hasta SP17; por lo tanto, el alias es VSP0. V41201 es la referencia de memoria de la palabra para los bits discretos SP20 hasta SP37, por lo tanto, el alias es VSP20

Mapa de memoria del PLC DL06

Tipo de memoria	Referencia de memoria discreta (octal)	Referencia de palabra (octal)	Decimal	Símbolo
Puntos de entradas	X0 – X777	V40400 - V40437	512	X0 ┆┆
Puntos de salidas	Y0 – Y777	V40500 – V40537	512	Y0 ┆()┆
Relevadores de control	C0 – C1777	V40600 - V40677	1024	C0 C0 ┆┆ ┆()┆
Relevadores especiales	SP0 – SP777	V41200 – V41237	512	SP0 ┆┆
Temporizadores	T0 – T377	V41100 – V41117	256	
Valores corrientes de temporizadores	Ninguna	V0 – V377	256	V0 K100 ┆┆┆
Bits de estado de temporizadores	T0 – T377	V41100 – V41117	256	T0 ┆┆
Contadores	CT0 – CT177	V41140 – V41147	128	
Valores corrientes de contadores	Ninguna	V1000 – V1177	128	V1000 K100 ┆┆┆
Bits de estado de contadores	CT0 – CT177	V41140 – V41147	128	CT0 ┆┆
Palabras de datos (Vea Apéndice F)	Ninguna	V400-V677 V1200 – V7377 V10000 - V17777	192 3200 4096	Ninguno específico, usado con muchas instrucciones.
EEPROM de palabras de datos (Vea Apéndice F)	Ninguna	V7400 – V7577	128	Ninguno específico, usado con muchas instrucciones. Datos pueden ser escritos a EEPROM por lo menos 100,000 veces antes de fallar.
Etapas	S0 – S1777	V41000 – V41017	1024	
Remote I/O (no es apoyado por ADC)	GX0-GX3777 GY0-GY3777	V40000-V40177 V40200-V40377	2048 2048	GX0 GY0 ┆┆ ┆()┆
Parámetros de sistema	Ninguna	V700-V777 V7600 – V7777 V36000-V37777	64 128 1024	Ninguno específico, usado para varios propósitos

1-El PLC DL06 está limitado a 20 entradas discretas y 16 salidas discretas, o hasta 64 E y 64 S con módulos opcionales, con el hardware del PLC disponible actualmente.

Mapa de bits de entradas X o salidas Y

Esta tabla entrega una lista de puntos de entradas y salidas individuales asociados a cada bit de memoria V incluyendo las veinte entradas y 16 salidas físicas incorporadas en el PLC además de hasta 64 entradas y 64 salidas para los módulos opcionales. Las referencias disponibles reales son X0 a X777 (V40400 - V40437) y Y0 a Y777 (V40500 - V40537).

MSB		Puntos de entradas (X) y salidas (Y) del PLC DL06														LSB		Dirección entrada X	Dirección salida Y
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40400	V40500		
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40401	V40501		
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40402	V40502		
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40403	V40503		
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40404	V40504		
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40405	V40505		
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40406	V40506		
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40407	V40507		
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V40410	V40510		
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V40411	V40511		
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V40412	V40512		
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V40413	V40513		
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V40414	V40514		
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V40415	V40515		
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V40416	V40516		
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V40417	V40517		
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V40420	V40520		
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V40421	V40521		
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V40422	V40522		
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V40423	V40523		
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V40424	V40524		
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V40425	V40525		
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V40426	V40526		
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V40427	V40527		
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V40430	V40530		
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V40431	V40531		
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V40432	V40532		
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V40433	V40533		
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V40434	V40534		
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V40435	V40535		
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V40436	V40536		
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V40437	V40537		

Mapa de bits de control del estado de etapas

Esta tabla suministra una lista de bits individuales de control de etapas asociados con cada bit de la dirección de memoria V.

MSB	Bits de control de etapas del DL06														LSB	Dirección
	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41000
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41001
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41002
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41003
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41004
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41005
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41006
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41007
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V41010
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V41011
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V41012
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V41013
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V41014
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V41015
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V41016
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V41017
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V41020
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V41021
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V41022
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V41023

MSB		Bits de control de etapas (S) del PLC DL06													LSB	Dirección
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V41024
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V41025
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V41026
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V41027
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V41030
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V41031
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V41032
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V41033
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V41034
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V41035
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V41036
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V41037
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	V41040
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	V41041
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	V41042
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	V41043
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	V41044
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	V41045
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	V41046
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	V41047
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	V41050
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	V41051
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	V41052
1277	1276	1275	1274	1273	1272	1271	1270	1267	1266	1265	1264	1263	1262	1261	1260	V41053
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	V41054
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	V41055
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	V41056
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	V41057
1417	1416	1415	1414	1413	1412	1411	1410	1407	1406	1405	1404	1403	1402	1401	1400	V41060
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	V41061
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	V41062
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	V41063
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	V41064
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	V41065
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	V41066
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	V41067
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	V41070
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	V41071
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	V41072
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	V41073
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	V41074
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	V41075
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	V41076
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	V41077

Mapa de bits de relevadores de control

Esta tabla suministra una lista de relevadores de control individual asociados con cada bit de una dirección de memoria.

MSB	Relevadores de control del PLC DL06 (C)														LSB	Dirección
	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40600
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40601
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40602
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40603
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40604
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40605
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40606
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40607
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V40610
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V40611
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V40612
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V40613
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V40614
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V40615
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V40616
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V40617
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V40620
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V40621
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V40622
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V40623
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V40624
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V40625
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V40626
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V40627
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V40630
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V40631
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V40632
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V40633
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V40634
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V40635
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V40636
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V40637

MSB	Relevadores de control del PLC DL06 (C)														LSB	Dirección
	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	V40640
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	V40641
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	V40642
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	V40643
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	V40644
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	V40645
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	V40646
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	V40647
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	V40650
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	V40651
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	V40652
1277	1276	1275	1274	1273	1272	1271	1270	1267	1266	1265	1264	1263	1262	1261	1260	V40653
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	V40654
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	V40655
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	V40656
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	V40657
1417	1416	1415	1414	1413	1412	1411	1410	1407	1406	1405	1404	1403	1402	1401	1400	V40660
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	V40661
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	V40662
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	V40663
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	V40664
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	V40665
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	V40666
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	V40667
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	V40670
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	V40671
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	V40672
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	V40673
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	V40674
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	V40675
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	V40676
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	V40677

Mapa de bits de estado de temporizadores

Esta tabla suministra una lista de contactos discretos de temporizadores asociados con cada bit de las direcciones de memoria.

MSB	Contactos de temporizadores (T) del PLC DL06														LSB	Dirección
	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41100
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41101
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41102
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41103
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41104
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41105
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41106
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41107
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V41110
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V41111
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V41112
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V41113
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V41114
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V41115
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V41116
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V41117

Mapa de bits de estado de contadores

Esta tabla suministra una lista de contactos discretos de contadores asociados con cada bit de las direcciones de memoria.

MSB	Contactos de contadores (CT) del PLC DL06														LSB	Dirección
	15	14	13	12	11	10	9	8	7	6	5	4	3	2		
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V41140
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V41141
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V41142
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V41143
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V41144
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V41145
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V41146
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V41147

Mapa de bits de Remote I/O

Esta tabla suministra una lista de contactos discretos de Remote I/O (entradas y salidas remotas del sistema Koyo) asociados con cada bit de las direcciones de memoria. Estas memorias pueden ser usadas como memoria de usuario del tipo V.

MSB	Puntos de E/S Remotas (GX) y (GY)															LSB	Dirección GX	Dirección GY
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1			
017	016	015	014	013	012	011	010	007	006	005	004	003	002	001	000	V40000	V40200	
037	036	035	034	033	032	031	030	027	026	025	024	023	022	021	020	V40001	V40201	
057	056	055	054	053	052	051	050	047	046	045	044	043	042	041	040	V40002	V40202	
077	076	075	074	073	072	071	070	067	066	065	064	063	062	061	060	V40003	V40203	
117	116	115	114	113	112	111	110	107	106	105	104	103	102	101	100	V40004	V40204	
137	136	135	134	133	132	131	130	127	126	125	124	123	122	121	120	V40005	V40205	
157	156	155	154	153	152	151	150	147	146	145	144	143	142	141	140	V40006	V40206	
177	176	175	174	173	172	171	170	167	166	165	164	163	162	161	160	V40007	V40207	
217	216	215	214	213	212	211	210	207	206	205	204	203	202	201	200	V40010	V40210	
237	236	235	234	233	232	231	230	227	226	225	224	223	222	221	220	V40011	V40211	
257	256	255	254	253	252	251	250	247	246	245	244	243	242	241	240	V40012	V40212	
277	276	275	274	273	272	271	270	267	266	265	264	263	262	261	260	V40013	V40213	
317	316	315	314	313	312	311	310	307	306	305	304	303	302	301	300	V40004	V40214	
337	336	335	334	333	332	331	330	327	326	325	324	323	322	321	320	V40015	V40215	
357	356	355	354	353	352	351	350	347	346	345	344	343	342	341	340	V40016	V40216	
377	376	375	374	373	372	371	370	367	366	365	364	363	362	361	360	V40007	V40217	
417	416	415	414	413	412	411	410	407	406	405	404	403	402	401	400	V40020	V40220	
437	436	435	434	433	432	431	430	427	426	425	424	423	422	421	420	V40021	V40221	
457	456	455	454	453	452	451	450	447	446	445	444	443	442	441	440	V40022	V40222	
477	476	475	474	473	472	471	470	467	466	465	464	463	462	461	460	V40023	V40223	
517	516	515	514	513	512	511	510	507	506	505	504	503	502	501	500	V40024	V40224	
537	536	535	534	533	532	531	530	527	526	525	524	523	522	521	520	V40025	V40225	
557	556	555	554	553	552	551	550	547	546	545	544	543	542	541	540	V40026	V40226	
577	576	575	574	573	572	571	570	567	566	565	564	563	562	561	560	V40027	V40227	
617	616	615	614	613	612	611	610	607	606	605	604	603	602	601	600	V40030	V40230	
637	636	635	634	633	632	631	630	627	626	625	624	623	622	621	620	V40031	V40231	
657	656	655	654	653	652	651	650	647	646	645	644	643	642	641	640	V40032	V40232	
677	676	675	674	673	672	671	670	667	666	665	664	663	662	661	660	V40033	V40233	
717	716	715	714	713	712	711	710	707	706	705	704	703	702	701	700	V40034	V40234	
737	736	735	734	733	732	731	730	727	726	725	724	723	722	721	720	V40035	V40235	
757	756	755	754	753	752	751	750	747	746	745	744	743	742	741	740	V40036	V40236	
777	776	775	774	773	772	771	770	767	766	765	764	763	762	761	760	V40037	V40237	

MSB	Puntos de E/S Remotas (GX) y (GY)														LSB	Dirección GX	Dirección GY
	15	14	13	12	11	10	9	8	7	6	5	4	3	2			
1017	1016	1015	1014	1013	1012	1011	1010	1007	1006	1005	1004	1003	1002	1001	1000	V40040	V40240
1037	1036	1035	1034	1033	1032	1031	1030	1027	1026	1025	1024	1023	1022	1021	1020	V40041	V40241
1057	1056	1055	1054	1053	1052	1051	1050	1047	1046	1045	1044	1043	1042	1041	1040	V40042	V40242
1077	1076	1075	1074	1073	1072	1071	1070	1067	1066	1065	1064	1063	1062	1061	1060	V40043	V40243
1117	1116	1115	1114	1113	1112	1111	1110	1107	1106	1105	1104	1103	1102	1101	1100	V40044	V40244
1137	1136	1135	1134	1133	1132	1131	1130	1127	1126	1125	1124	1123	1122	1121	1120	V40045	V40245
1157	1156	1155	1154	1153	1152	1151	1150	1147	1146	1145	1144	1143	1142	1141	1140	V40046	V40246
1177	1176	1175	1174	1173	1172	1171	1170	1167	1166	1165	1164	1163	1162	1161	1160	V40047	V40247
1217	1216	1215	1214	1213	1212	1211	1210	1207	1206	1205	1204	1203	1202	1201	1200	V40050	V40250
1237	1236	1235	1234	1233	1232	1231	1230	1227	1226	1225	1224	1223	1222	1221	1220	V40051	V40251
1257	1256	1255	1254	1253	1252	1251	1250	1247	1246	1245	1244	1243	1242	1241	1240	V40052	V40252
1277	1276	1275	1274	1273	1272	1271	1270	1267	1266	1265	1264	1263	1262	1261	1260	V40053	V40253
1317	1316	1315	1314	1313	1312	1311	1310	1307	1306	1305	1304	1303	1302	1301	1300	V40054	V40254
1337	1336	1335	1334	1333	1332	1331	1330	1327	1326	1325	1324	1323	1322	1321	1320	V40055	V40255
1357	1356	1355	1354	1353	1352	1351	1350	1347	1346	1345	1344	1343	1342	1341	1340	V40056	V40256
1377	1376	1375	1374	1373	1372	1371	1370	1367	1366	1365	1364	1363	1362	1361	1360	V40057	V40257
1417	1416	1415	1414	1413	1412	1411	1410	1407	1406	1405	1404	1403	1402	1401	1400	V40060	V40260
1437	1436	1435	1434	1433	1432	1431	1430	1427	1426	1425	1424	1423	1422	1421	1420	V40061	V40261
1457	1456	1455	1454	1453	1452	1451	1450	1447	1446	1445	1444	1443	1442	1441	1440	V40062	V40262
1477	1476	1475	1474	1473	1472	1471	1470	1467	1466	1465	1464	1463	1462	1461	1460	V40063	V40263
1517	1516	1515	1514	1513	1512	1511	1510	1507	1506	1505	1504	1503	1502	1501	1500	V40064	V40264
1537	1536	1535	1534	1533	1532	1531	1530	1527	1526	1525	1524	1523	1522	1521	1520	V40065	V40265
1557	1556	1555	1554	1553	1552	1551	1550	1547	1546	1545	1544	1543	1542	1541	1540	V40066	V40266
1577	1576	1575	1574	1573	1572	1571	1570	1567	1566	1565	1564	1563	1562	1561	1560	V40067	V40267
1617	1616	1615	1614	1613	1612	1611	1610	1607	1606	1605	1604	1603	1602	1601	1600	V40070	V40270
1637	1636	1635	1634	1633	1632	1631	1630	1627	1626	1625	1624	1623	1622	1621	1620	V40071	V40271
1657	1656	1655	1654	1653	1652	1651	1650	1647	1646	1645	1644	1643	1642	1641	1640	V40072	V40272
1677	1676	1675	1674	1673	1672	1671	1670	1667	1666	1665	1664	1663	1662	1661	1660	V40073	V40273
1717	1716	1715	1714	1713	1712	1711	1710	1707	1706	1705	1704	1703	1702	1701	1700	V40074	V40274
1737	1736	1735	1734	1733	1732	1731	1730	1727	1726	1725	1724	1723	1722	1721	1720	V40075	V40275
1757	1756	1755	1754	1753	1752	1751	1750	1747	1746	1745	1744	1743	1742	1741	1740	V40076	V40276
1777	1776	1775	1774	1773	1772	1771	1770	1767	1766	1765	1764	1763	1762	1761	1760	V40077	V40277

MSB	Puntos de E/S Remotas (GX) y (GY)														LSB	Dirección GX	Dirección GY
	15	14	13	12	11	10	9	8	7	6	5	4	3	2			
2017	2016	2015	2014	2013	2012	2011	2010	2007	2006	2005	2004	2003	2002	2001	2000	V40100	V40300
2037	2036	2035	2034	2033	2032	2031	2030	2027	2026	2025	2024	2023	2022	2021	2020	V40101	V40301
2057	2056	2055	2054	2053	2052	2051	2050	2047	2046	2045	2044	2043	2042	2041	2040	V40102	V40302
2077	2076	2075	2074	2073	2072	2071	2070	2067	2066	2065	2064	2063	2062	2061	2060	V40103	V40303
2117	2116	2115	2114	2113	2112	2111	2110	2107	2106	2105	2104	2103	2102	2101	2100	V40104	V40304
2137	2136	2135	2134	2133	2132	2131	2130	2127	2126	2125	2124	2123	2122	2121	2120	V40105	V40305
2157	2156	2155	2154	2153	2152	2151	2150	2147	2146	2145	2144	2143	2142	2141	2140	V40106	V40306
2177	2176	2175	2174	2173	2172	2171	2170	2167	2166	2165	2164	2163	2162	2161	2160	V40107	V40307
2217	2216	2215	2214	2213	2212	2211	2210	2207	2206	2205	2204	2203	2202	2201	2200	V40110	V40310
2237	2236	2235	2234	2233	2232	2231	2230	2227	2226	2225	2224	2223	2222	2221	2220	V40111	V40311
2257	2256	2255	2254	2253	2252	2251	2250	2247	2246	2245	2244	2243	2242	2241	2240	V40112	V40312
2277	2276	2275	2274	2273	2272	2271	2270	2267	2266	2265	2264	2263	2262	2261	2260	V40113	V40313
2317	2316	2315	2314	2313	2312	2311	2310	2307	2306	2305	2304	2303	2302	2301	2300	V40114	V40314
2337	2336	2335	2334	2333	2332	2331	2330	2327	2326	2325	2324	2323	2322	2321	2320	V40115	V40315
2357	2356	2355	2354	2353	2352	2351	2350	2347	2346	2345	2344	2343	2342	2341	2340	V40116	V40316
2377	2376	2375	2374	2373	2372	2371	2370	2367	2366	2365	2364	2363	2362	2361	2360	V40117	V40317
2417	2416	2415	2414	2413	2412	2411	2410	2407	2406	2405	2404	2403	2402	2401	2400	V40120	V40320
2437	2436	2435	2434	2433	2432	2431	2430	2427	2426	2425	2424	2423	2422	2421	2420	V40121	V40321
2457	2456	2455	2454	2453	2452	2451	2450	2447	2446	2445	2444	2443	2442	2441	2440	V40122	V40322
2477	2476	2475	2474	2473	2472	2471	2470	2467	2466	2465	2464	2463	2462	2461	2460	V40123	V40323
2517	2516	2515	2514	2513	2512	2511	2510	2507	2506	2505	2504	2503	2502	2501	2500	V40124	V40324
2537	2536	2535	2534	2533	2532	2531	2530	2527	2526	2525	2524	2523	2522	2521	2520	V40125	V40325
2557	2556	2555	2554	2553	2552	2551	2550	2547	2546	2545	2544	2543	2542	2541	2540	V40126	V40326
2577	2576	2575	2574	2573	2572	2571	2570	2567	2566	2565	2564	2563	2562	2561	2560	V40127	V40327
2617	2616	2615	2614	2613	2612	2611	2610	2607	2606	2605	2604	2603	2602	2601	2600	V40130	V40330
2637	2636	2635	2634	2633	2632	2631	2630	2627	2626	2625	2624	2623	2622	2621	2620	V40131	V40331
2657	2656	2655	2654	2653	2652	2651	2650	2647	2646	2645	2644	2643	2642	2641	2640	V40132	V40332
2677	2676	2675	2674	2673	2672	2671	2670	2667	2666	2665	2664	2663	2662	2661	2660	V40133	V40333
2717	2716	2715	2714	2713	2712	2711	2710	2707	2706	2705	2704	2703	2702	2701	2700	V40134	V40334
2737	2736	2735	2734	2733	2732	2731	2730	2727	2726	2725	2724	2723	2722	2721	2720	V40135	V40335
2757	2756	2755	2754	2753	2752	2751	2750	2747	2736	2735	2734	2733	2732	2731	2730	V40136	V40336
2777	2776	2775	2774	2773	2772	2771	2770	2767	2766	2765	2764	2763	2762	2761	2760	V40137	V40337

Capítulo 3: Especificaciones y operación

3

MSB	Puntos de E/S Remotas (GX) y (GY)															LSB	Dirección GX	Dirección GY
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1			
3017	3016	3015	3014	3013	3012	3011	3010	3007	3006	3005	3004	3003	3002	3001	3000	V40140	V40340	
3037	3036	3035	3034	3033	3032	3031	3030	3027	3026	3025	3024	3023	3022	3021	3020	V40141	V40341	
3057	3056	3055	3054	3053	3052	3051	3050	3047	3046	3045	3044	3043	3042	3041	3040	V40142	V40342	
3077	3076	3075	3074	3073	3072	3071	3070	3067	3066	3065	3064	3063	3062	3061	3060	V40143	V40343	
3117	3116	3115	3114	3113	3112	3111	3110	3107	3106	3105	3104	3103	3102	3101	3100	V40144	V40344	
3137	3136	3135	3134	3133	3132	3131	3130	3127	3126	3125	3124	3123	3122	3121	3120	V40145	V40345	
3157	3156	3155	3154	3153	3152	3151	3150	3147	3146	3145	3144	3143	3142	3141	3140	V40146	V40346	
3177	3176	3175	3174	3173	3172	3171	3170	3167	3166	3165	3164	3163	3162	3161	3160	V40147	V40347	
3217	3216	3215	3214	3213	3212	3211	3210	3207	3206	3205	3204	3203	3202	3201	3200	V40150	V40350	
3237	3236	3235	3234	3233	3232	3231	3230	3227	3226	3225	3224	3223	3222	3221	3220	V40151	V40351	
3257	3256	3255	3254	3253	3252	3251	3250	3247	3246	3245	3244	3243	3242	3241	3240	V40152	V40352	
3277	3276	3275	3274	3273	3272	3271	3270	3267	3266	3265	3264	3263	3262	3261	3260	V40153	V40353	
3317	3316	3315	3314	3313	3312	3311	3310	3307	3306	3305	3304	3303	3302	3301	3300	V40154	V40354	
3337	3336	3335	3334	3333	3332	3331	3330	3327	3326	3325	3324	3323	3322	3321	3320	V40155	V40355	
3357	3356	3355	3354	3353	3352	3351	3350	3347	3346	3345	3344	3343	3342	3341	3340	V40156	V40356	
3377	3376	3375	3374	3373	3372	3371	3370	3367	3366	3365	3364	3363	3362	3361	3360	V40157	V40357	
3417	3416	3415	3414	3413	3412	3411	3410	3407	3406	3405	3404	3403	3402	3401	3400	V40160	V40360	
3437	3436	3435	3434	3433	3432	3431	3430	3427	3426	3425	3424	3423	3422	3421	3420	V40161	V40361	
3457	3456	3455	3454	3453	3452	3451	3450	3447	3446	3445	3444	3443	3442	3441	3440	V40162	V40362	
3477	3476	3475	3474	3473	3472	3471	3470	3467	3466	3465	3464	3463	3462	3461	3460	V40163	V40363	
3517	3516	3515	3514	3513	3512	3511	3510	3507	3506	3505	3504	3503	3502	3501	3500	V40164	V40364	
3537	3536	3535	3534	3533	3532	3531	3530	3527	3526	3525	3524	3523	3522	3521	3520	V40165	V40365	
3557	3556	3555	3554	3553	3552	3551	3550	3547	3546	3545	3544	3543	3542	3541	3540	V40166	V40366	
3577	3576	3575	3574	3573	3572	3571	3570	3567	3566	3565	3564	3563	3562	3561	3560	V40167	V40367	
3617	3616	3615	3614	3613	3612	3611	3610	3607	3606	3605	3604	3603	3602	3601	3600	V40170	V40370	
3637	3636	3635	3634	3633	3632	3631	3630	3627	3626	3625	3624	3623	3622	3621	3620	V40171	V40371	
3657	3656	3655	3654	3653	3652	3651	3650	3647	3646	3645	3644	3643	3642	3641	3640	V40172	V40372	
3677	3676	3675	3674	3673	3672	3671	3670	3667	3666	3665	3664	3663	3662	3661	3660	V40173	V40373	
3717	3716	3715	3714	3713	3712	3711	3710	3707	3706	3705	3704	3703	3702	3701	3700	V40174	V40374	
3737	3736	3735	3734	3733	3732	3731	3730	3727	3726	3725	3724	3723	3722	3721	3720	V40175	V40375	
3757	3756	3755	3754	3753	3752	3751	3750	3747	3746	3745	3744	3743	3742	3741	3740	V40176	V40376	
3777	3776	3775	3774	3773	3772	3771	3770	3767	3766	3765	3764	3763	3762	3761	3760	V40177	V40377	

DISEÑO Y CONFIGURACIÓN DEL SISTEMA



En este capítulo

Estrategias de diseño del sistema DL06	4-2
Colocación de los módulos opcionales	4-3
Configuración de entradas y salidas	4-4
Consumo de corriente	4-5
Configuración de los puertos del PLC DL06	4-7
Configuración ladder de los puertos del PLC DL06	4-9
Comunicación con diversos protocolos	4-13
Operación de un esclavo MODBUS RTU	4-14
Operación de un maestro MODBUS RTU	4-20
Operación de un maestro MODBUS RTU con MRX y MWX	4-24
Operación con caracteres ASCII	4-26

Estrategias de diseño del sistema DL06

Configuraciones del sistema de entradas y salidas (E/S)

Los PLCs DL06 ofrecen diversas configuraciones de E/S. Escoja la configuración que sea correcta para su aplicación, y tenga presente que los PLCs DL06 tienen la capacidad de poder agregar entradas y salidas con el uso de las tarjetas opcionales. Aunque no hay Remote I/O, se puede usar Terminator I/O y también hay muchas tarjetas opcionales disponibles. Por ejemplo:

- Varios módulos de E/S de corriente alterna y corriente continua
- Módulos de combinación de E/S
- Módulos análogos de E/S
- Módulos análogos combinación de E/S

Se puede desarrollar un sistema DL06 usando diversos arreglos usando los módulos opcionales. Vea nuestro manual de usuario de los módulos opcionales DL05/06 (D0-OPTIONS-MSP) en el sitio de Internet, www.automationdirect.com para más información detallada de la selección.

Configuraciones de redes

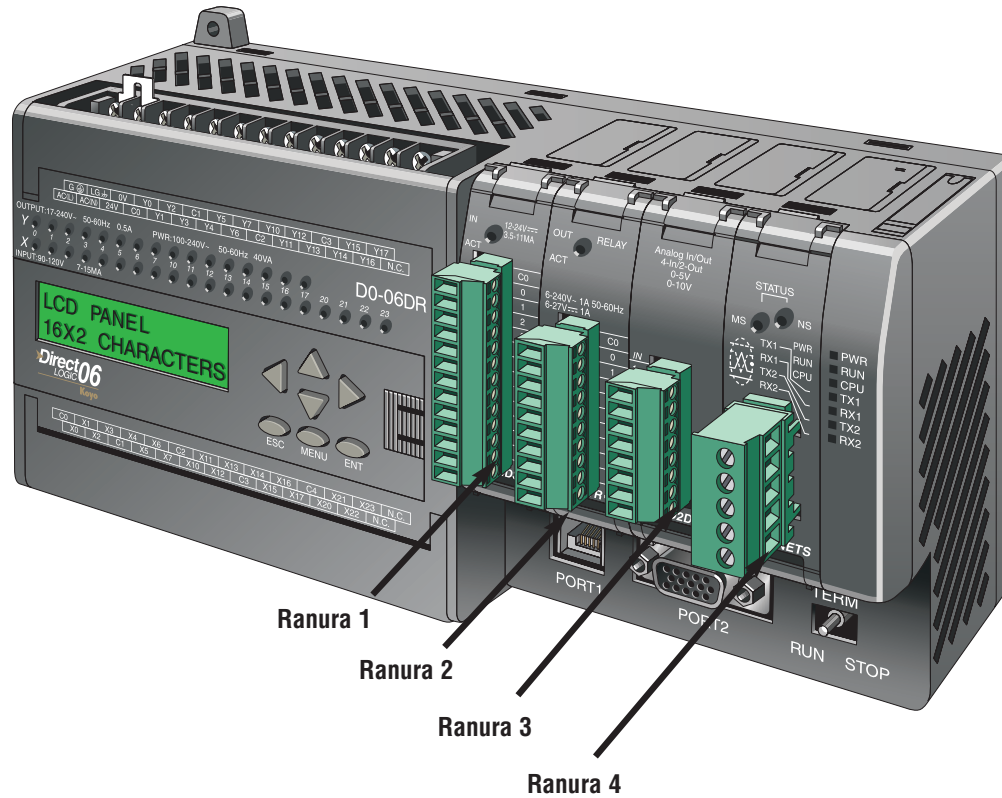
El PLC DL06 ofrece las formas siguientes de establecer una red:

- **El módulo de comunicaciones de Ethernet** — conecta un DL06 con las redes de alta velocidad punto a punto o peer-to-peer. conecta un DL06 con redes de alta velocidad punto a punto o cualquier PLC puede iniciar comunicaciones con cualquier otros PLC u otras interfaces de operador, tales como C-more, al usar los módulos de ECOM.
- **Los módulos de comunicaciones de datos** — Los módulos de comunicaciones de datos conectan un DL06 con dispositivos usando DeviceNet o Profibus para conectarse con controladores maestros, así como también un módulo serial D0-DCM.
- **El puerto de comunicaciones 1** — El DL06 tiene un conector RJ12 de 6 clavijas en el puerto 1 que soporta (como esclavo) protocolos K-sequence, MODBUS RTU o *DirectNET*.
- **El puerto de comunicaciones 2** — El DL06 tiene un conector de 15 clavijas en el puerto 2 que soporta los protocolos *DirectNET* y MODBUS RTU maestro/esclavo, o el protocolo K-sequence como esclavo (las instrucciones MRX y MWX permiten que usted use direcciones MODBUS nativas en su programa ladder sin necesidad de realizar conversiones octal a decimal). El puerto 2 se puede también usar comunicaciones ASCII IN/OUT.

Colocación de los módulos opcionales

Enumeración de las ranuras

El PLC DL06 tiene cuatro ranuras, que se numeran como sigue:

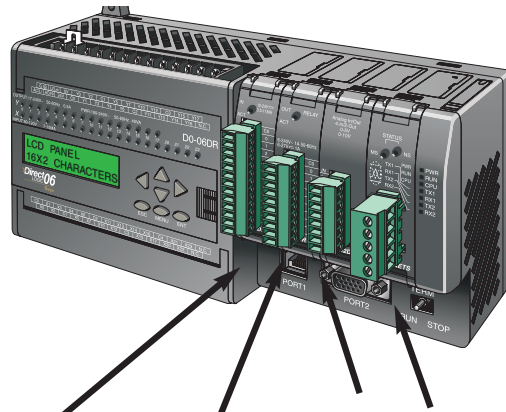


Configuración de entradas y salidas

Configuración automática de E/S

Los PLCs DL06 detectan automáticamente cualquier módulo instalado de E/S (módulos incluyendo los de especialidad) durante la enrgización, y establecen la configuración correcta y direcciones de E/S. Esto se aplica a los módulos situados en la base local. Para la mayoría de los usos, usted nunca tendrá que cambiar la configuración.

Las direcciones de E/S utilizan enumeración octal, comenzando en X100 y Y100 en la ranura mas a la izquierda. Las direcciones se asignan en grupos de 8, o 16 dependiendo del número de los puntos para el módulo de E/S. Los módulos discretos de entradas y de salidas se pueden mezclar en cualquier orden. El diagrama siguiente muestra a convención de la enumeración de E/S para un sistema de ejemplo. El programador portátil y *DirectSOFT* proporcionan las funciones AUX. que permiten que usted configure automáticamente el I/O. Por ejemplo, con el programador, el comando 46 AUX ejecuta una configuración automática, que permite que el PLC examine los módulos instalados y determine la configuración y la dirección. Con *DirectSOFT*, sería usado el menú PLC Configure I/O.



Automático

Ranura 1 8 entradas X100–X107	Ranura 2 16 salidas Y100–Y117	Ranura 3 16 entradas X110–X127	Ranura 4 8 entradas X130–X137
-------------------------------------	-------------------------------------	--------------------------------------	-------------------------------------

Manual

Ranura 1 8 entradas X100–X107	Ranura 2 16 salidas Y100–Y117	Ranura 3 16 entradas X200–X217	Ranura 4 8 entradas X120–X127
-------------------------------------	-------------------------------------	--------------------------------------	-------------------------------------

Configuración Manual de E/S

Puede que nunca llegue a ser necesario, pero los PLCs DL06 permiten asignar direcciones manuales de E/S para cualquier ranura o ranuras de E/S. Usted puede modificar manualmente una configuración para igualar una enumeración arbitraria de E/S. Por ejemplo, dos módulos de entrada adyacentes pueden comenzar direcciones en el PLC X100 y X200. Use el menú de *DirectSOFT PLC Configure I/O* para configurar la opción de asignar la dirección manual de E/S. En la configuración automática, las direcciones se asignan en los límites de 8 puntos. La configuración manual, sin embargo, asume que todos los módulos son por lo menos 16 puntos, así que usted puede asignar solamente las direcciones que son un múltiplo de 20 (octal). Usted puede todavía usar los módulos de 8 puntos, pero serán asignadas 16 direcciones y las ocho direcciones superiores no serán usadas.



ADVERTENCIA: Si usted configura manualmente una ranura de E/S, la dirección de E/S de otros módulos puede cambiar. Esto es porque los DL06 no permiten que usted asigne direcciones duplicadas de E/S. Usted debe corregir siempre cualquier error de la configuración de E/S antes de colocar el PLC en modo RUN. Los errores sin corregir pueden causar una operación imprevisible de la máquina, que puede dar lugar a un riesgo de daños corporales o daño al equipo.

Consumo de corriente

El DL06 tiene cuatro ranuras de módulos opcionales. Para determinar si la combinación de módulos seleccionados tendrá suficiente energía, usted necesitará realizar un cálculo del consumo de corriente.

Corriente suministrada

La corriente es suministrada a partir de dos fuentes, de la fuente de alimentación interna de la unidad y si fuera requerido, de una fuente externa (a ser suministrada por el cliente). Los PLCs DL06 alimentados por corriente alterna tienen una fuente de poder interna que proveen una cantidad limitada de corriente en 24VCC. La salida 24VCC se puede utilizar para accionar dispositivos externos.

Para un cálculo de consumo de corriente, comience considerando la corriente suministrada por la unidad. Todas las fuentes de poder internas del PLC DL06 tienen la misma capacidad a 5VCC. Hay un balance entre la corriente a 5VCC y la corriente a 24VCC. La cantidad de corriente 5VCC disponible depende de la cantidad de corriente 24VCC que es utilizada, y la cantidad de corriente 24VCC disponible depende de la cantidad de corriente 5VDC consumida.

Hay algunos aparatos que pueden consumir 5 VCC desde el puerto 1. La capacidad de consumo de la clavija de 5 VCC es del orden de 250 mA.

Consumo requerido por la unidad

Debido a las diversas configuraciones de E/S disponibles en la familia DL06, la corriente consumida por la unidad varía de modelo a modelo. Reste la cantidad de corriente requerida por la unidad de la cantidad de corriente suministrada por la unidad. Asegúrese de restar los valores de corriente en los voltajes de 5VCC y 24VCC.

Consumo requerido por los módulos opcionales

Luego reste la cantidad de corriente requerida por los módulos opcionales que usted está planeando utilizar. Una vez más recuerde de restar los valores de corriente en los voltajes de 5 VCC y 24 VCC. Si su análisis del presupuesto de corriente muestra corriente disponible de sobra, usted debe tener una configuración realizable.

Corriente suministrada por el DL06		
No. de parte	5 VCC (mA)	24 VCC (mA)
D0-06xx	<1500 mA	300 mA
	<2000 mA	200 mA
D0-06xx-D	1500 mA	ninguna

Si la carga en 5VCC es menos que 2000 mA, pero más que 1500mA, entonces la corriente disponible de la fuente de poder de 24VCC es 200 mA. Si la carga en 5VCC es menos que 1500 mA, la corriente disponible en 24VCC es 300 mA

Corriente requerida por el DL06		
No. de parte	5 VCC (mA)	24 VCC (mA)
D0-06AA	800 mA	ninguna
D0-06AR	900 mA	ninguna
D0-06DA	800 mA	ninguna
D0-06DD1	600 mA	280 mA, nota 1
D0-06DD2	600 mA	ninguna
D0-06DR	950 mA	ninguna
D0-06DD1-D	600 mA	280 mA, nota 1
D0-06DD2-D	600 mA	ninguna
D0-06DR-D	950 mA	ninguna

Ejemplo de cálculo del consumo			
Fuente de corriente		5VCC (mA)	24VCC (mA)
D0-06DD1 (seleccione A o B)	A	1500 mA	300 mA
	B	2000 mA	200 mA
Corriente necesaria		5VCC (mA)	24VCC (mA)
D0-06DD1		600 mA	280 mA, nota 1
D0-16ND3		35 mA	0
D0-10TD1		150 mA	0
D0-08TR		280 mA	0
F0-4AD2DA-2		100 mA	0
D0-06LCD		50 mA	0
Total Used		1215 mA	280 mA
Lo que queda	A	285 mA	20 mA
	B	785 mA	nota 2



NOTA 1: La fuente auxiliar de 24VCC se usa para alimentar el terminal V+ de las salidas de D0-06DD1/-D.
NOTA 2: Si la fuente auxiliar de 24VCC se usa para alimentar salidas drenadoras, use la opción A de la tabla de arriba.

Corriente consumida por los módulos opcionales del PLC DL06		
No. de parte	5 VDC (mA)	24 VDC (mA)
D0-07CDR	130 mA	ninguna
D0-08CDD1	100 mA	ninguna
D0-08TR	280 mA	ninguna
D0-10ND3	35 mA	ninguna
D0-10ND3F	35 mA	ninguna
D0-10TD1	150 mA	ninguna
D0-10TD2	150mA	ninguna
D0-16ND3	35 mA	ninguna
D0-16TD1	200 mA	ninguna
D0-16TD2	200mA	ninguna
D0-DCM	250 mA	ninguna
D0-DEVNETS	45 mA	ninguna
F0-04TRS	250 mA	ninguna
F0-08NA-1	5 mA	ninguna
F0-04AD-1	50 mA	ninguna
F0-04AD-2	75 mA	ninguna
F0-2AD2DA-2	50 mA	30 mA
F0-4AD2DA-1	100 mA	40 mA
F0-4AD2DA-2	100 mA	ninguna
F0-04RTD	70 mA	ninguna
F0-04THM	30 mA	ninguna
F0-CP128	150 mA	ninguna
H0-PSCM	530 mA	ninguna
H0-ECOM	250 mA	ninguna
H0-CTRIO	250 mA	ninguna



NOTA: Vea el manual de opciones DL05/DL06 para datos de los módulos para su proyecto.

Corriente consumida por otros aparatos		
No. de parte	5 VCC (mA)	24 VCC (mA)
D0-06LCD	50 mA	ninguna
D2-HPP	200 mA	ninguna
DV1000	150 mA	ninguna
EA1-S3ML/(-N)	210 mA	ninguna

Configuración de los puertos de comunicación del DL06

Esta sección describe cómo configurar los puertos al establecer una red de PLCs para MODBUS, DirectNET o ASCII. Esto permitirá que conecte el sistema del PLC DL06 directamente a redes.

Los dispositivos maestros MODBUS en la red deben ser capaces de generar comandos MODBUS de lectura o de escritura de los datos. Para detalles en el protocolo de MODBUS, vea a la guía de referencia del protocolo de Gould MODBUS (P1-MBUS-300 Rev B). Si una versión más reciente está disponible, compruebe con su distribuidor de MODBUS antes de pedir la documentación.

Para más detalles en DirectNET, pida el manual DirectNET, artículo DA-DNET-M o descárguelo del sitio de Internet de AUTOMATION DIRECT. Vea también el apéndice K.



Nota: Para más información sobre el protocolo de MODBUS vea el sitio de Internet del grupo Schneider en: www.schneiderautomation.com. Para más información sobre el protocolo de DirectNET, baje el manual sin costo desde nuestro sitio de Internet: www.automationdirect.com. Seleccione Manual/Doc>Online manuals>Misc.>DA-DNET-M.

Comunicaciones Puerto 1	
Com 1	Se conecta a HPP, DirectSOFT32, interfaces de operador, etc. 6 clavijas, RS232C Tasa de comunicación(baud): 9600 (fija) Paridad: odd (valor original de fábrica) Dirección de la estación: 1 (fija) 8 bits de datos 1 bit start, 1 bit stop Asíncrono, half-duplex, DTE Protocolo: (Seleccionable automáticamente) K-sequence (solamente esclavo), DirectNET (solamente esclavo), MODBUS (solamente esclavo)

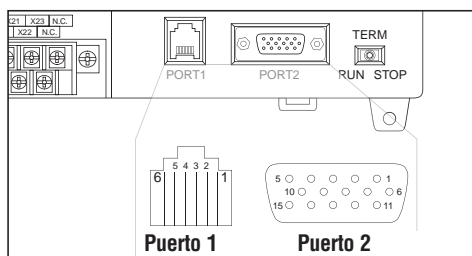
Comunicaciones Puerto 2	
Com 2	Se conecta a HPP, DirectSOFT32, interfaces de operador, etc. 15-clavijas, puerto de funciones múltiples, RS232C, RS422, RS485 Tasa de comunicación (baud): 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 Paridad: odd (valor original), even, 0 (nada) Dirección de la estación: 1 (valor original) 8 bits de datos 1 bit start, 1 bit stop Asíncrono, half-duplex, DTE Protocolos: (selección automática) K-sequence (solamente esclavo), DirectNET (maestro/esclavo), MODBUS (maestro/esclavo), non-sequence/print/ASCII in/out

Descripciones de clavijas Puerto 1

1	0V	Conexión (-)(GND)
2	5V	Conexión (+)
3	RXD	Recibe datos (RS-232C)
4	TXD	Transmite datos (RS-232C)
5	5V	Conexión (+)
6	0V	Conexión (-)(GND)

Descripciones de clavijas Puerto 2

1	5V	Conexión (+)
2	TXD	Transmite datos (RS-232C)
3	RXD	Recibe datos (RS-232C)
4	RTS	Ready to send
5	CTS	Clear to send
6	RXD-	Recibe datos (-) (RS-422/485)
7	0V	Conexión (-) (GND)
8	0V	Conexión (-) (GND)
9	TXD+	Transmite datos (+) (RS-422/485)
10	TXD-	Transmite datos (-) (RS-422/485)
11	RTS+	Ready to send (+) (RS-422/485)
12	RTS-	Ready to send (-) (RS-422/485)
13	RXD+	Recibe datos (+) (RS-422/485)
14	CTS+	Clear to send (+) (RS-422/485)
15	CTS-	Clear to send (-) (RS-422/485)



Seleccionando una especificación de red

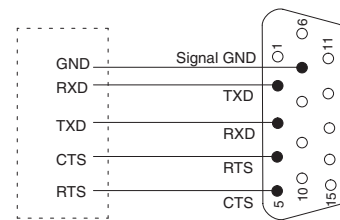
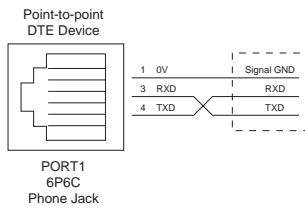
El puerto de funcionamiento múltiple del PLC DL06 le da la opción de usar las especificaciones RS-232C, RS-422, o RS-485. Primero, determine si la red será un tipo de dos conductores de RS-232C, un tipo de 4 conductores RS-422, o un tipo de 2 o 4 conductores RS-485.

La especificación RS-232C es simple de implementar para redes de distancias relativamente cortas (15 metros máximo) y la comunicación debe estar solamente entre dos dispositivos. RS-422 y RS-485 son aplicables para redes que cubren largas distancias (1000 metros máximo) y para redes de varios nodos.

Nota: Se necesitan resistencias de terminación en ambos finales de la red RS-422 y RS-485. Es necesario seleccionar resistencias de valor igual a la impedancia del cable (entre 100 y 500 ohm).

Red del tipo RS-232

Normalmente, las señales RS-232 se utilizan para distancias más cortas (15 metros de máximo), para comunicaciones entre dos dispositivos.

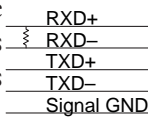


Red del tipo RS-422

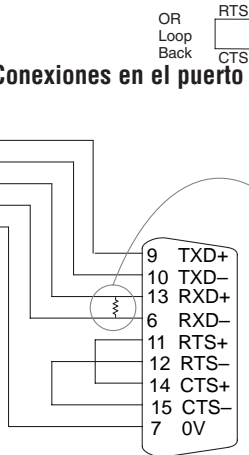
Las señales RS-422 se usan en grandes distancias (1000 metros máximo).

Utilice resistencias de terminación en ambos extremos del cableado de la red RS-422.

Conexiones en el puerto 1



Conexiones en el puerto 2

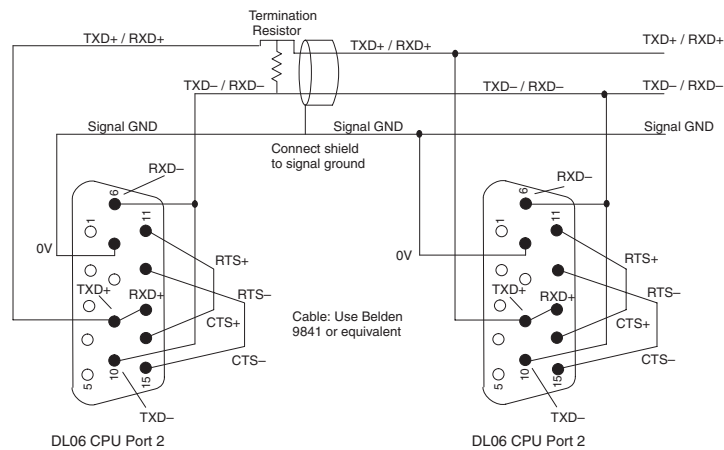


The recommended cable for RS422 is Belden 9729 or equivalent.

Red del tipo RS-485

Las señales RS-485 se usan en distancias grandes (1000 metros máximo) y para redes de varios nodos.

Use resistencias de terminación en ambos extremos del cableado de la red, con un valor igual a la impedancia del cable (entre 100 y 500 Ohm).



Configuración de los puertos para comunicación

El puerto 1 tiene una configuración fija. El puerto 2 es programado por medio de un cuadro de diálogo en el menú de *DirectSOFT*. Alternativamente, es posible programar la configuración del puerto 2 con diagrama ladder. Esto puede ser útil en el caso que se quiera mantener la configuración cada vez que el PLC se encienda, y es independiente de entrar la configuración con el menú de *DirectSOFT*.

El PLC DL06 usa las palabras reservadas V7655, V7656 y V7657 y estas son la definiciones que deben ser colocadas en esas palabras (Otras CPUs usan otras memorias reservadas):

La palabra V7655 está formada de los siguientes grupos de bits: 0yyy
0TTT mmmm mxxx

yyy = El atraso de RTS ON
TTT = Communication time out
mmmmm = tipo de protocolo
xxx = TRS OFF delay

RST ON delay yyy	TimeOut TTT	Protocolo mmmmm	RTS OFF delay xxx
000= 0 ms	000 = 100 %	10000 = K-sequence	000= 0 ms
001 = 2 ms	001 = 120 %	01000 = DirectNet	001 = 2 ms
010 = 5 ms	010 = 150 %	00100 = MODBUS	010 = 5 ms
011 = 10 ms	011 = 200 %	00010 = Non sequence	011 = 10 ms
100 = 20 ms	100 = 500 %		100 = 20 ms
101 = 50 ms	101 = 1000 %		101 = 50 ms
110 = 10 ms	110 = 2000 %		110 = 10 ms
111 = 500 ms	111 = 5000 %		111 = 500 ms

La palabra V7656 está formada de los siguientes grupos de bits: pps0 ebbb xaaa aaaa

pp = Parity
s = Stop bits
e = Supresión de eco
bbb= Tasa de Baud
x = Modo del protocolo
aaaaaa = Dirección del nodo

Paridad pp	Bits S	Supresión de eco e	Tasa de baud bbb	Modo protocolo x	Dirección del nodo aaaaaa
00= none	0 = 1 bit	0 = RS-232/422	000= 300	0 = hexadec	1 -90= k-sequence
10 = Odd	1 = 2 bits	1 = 485	001 = 600	1 = ASCII	1-90 <i>DirectNET</i>
11 = Even			010 = 1200		1-247 MODBUS
			011 = 2400		
			100 = 4800		
			101 = 9600		
			110 = 19200		
			111 = 38400		

La palabra V7657 para el protocolo non sequence: está formada de los siguientes grupos de bits:

pps0 ebbb aaaa aaaa

pp = Parity
s = Stop bits
e = Supresión de eco
bbb = Tasa de Baud
aaaaaaa= Dirección del esclavo

Paridad pp	Bits S	Supresión de eco e	Tasa de baud bbb	Modo protocolo aaaaaa
00= none	0=1 bit	0=RS-232/422	000= 300	01110000 = No flow control
10 = Odd	1=2 bits	1=485	001 = 600	01110001 = Xon/Xoff flow control
11 = Even			010 = 1200	01110010 = RTS flow control
			011 = 2400	01110011 = Xon/Xof y RTS flow control
			100 = 4800	
			101 = 9600	
			110 = 19200	
			111 = 38400	

La **palabra V7657**: La CPU aceptará los valores de configuración cuando se escribe un valor de "configuración completa" en esta memoria. Para el DL06 es K0500. Cuando la CPU verifica el valor, cambiará el número "5" por una "A" si los valores de configuración son aceptados o una "E" si hay un error.

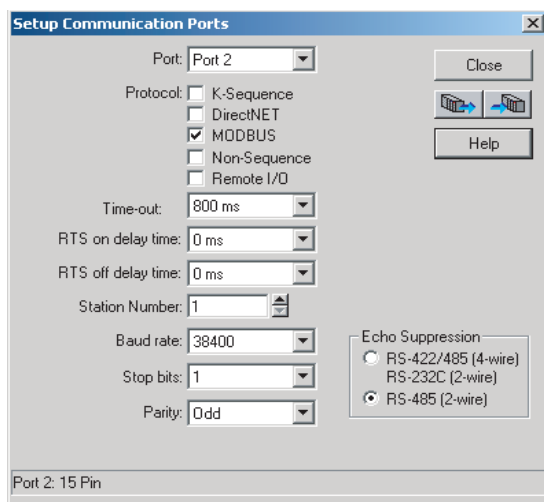
Adicionalmente, para el protocolo Non-sequence la palabra **V7650** es una memoria de almacenamiento temporal de datos ASCII que llegan al PLC. Coloque este parámetro referido a una localización no usada.

Configuración del puerto como MODBUS RTU

En *DirectSOFT*, haga clic en el menú PLC, luego Setup, luego "Set up Sec. Comm Port".

- **Port**: De la lista del número del puerto, escoja "Port 2".

- **Protocol**: Haga clic en el cuadro a la izquierda de "MODBUS" (use AUX 56 en el HPP, y seleccione "MBUS"), y luego verá el cuadro de diálogo abajo:



- **Timeout**: El período que el puerto esperará después que envíe un mensaje para obtener una respuesta antes de detectar un error.

- **RTS ON / OFF Delay Time**: Especifica el tiempo que espera el PLC DL06 para mandar datos después que la señal TRS se ha hecho ON. Especifica el tiempo que espera el PLC DL06 después de mandar datos para hacer OFF la señal TRS. *Cuando se usa el DL06 en una red multinodo, el tiempo de demora RTS ON debe ser colocado a por lo menos 5 ms y el tiempo de Demora RST OFF debe ser colocado a por lo menos 2ms. Si usted encuentra problemas, el tiempo se puede aumentar.*

- **Station Number(Número de la estación)**: Para hacer el puerto de la CPU un maestro de MODBUS, escoja "1". El rango posible para números de esclavo de MODBUS es de 1 a 247, pero el DL06 solo permite esclavos 1 a 99. Cada esclavo debe tener un número único. Durante la energización el puerto es automáticamente un esclavo, a menos que y hasta que el DL06 ejecute las instrucciones de la red de la lógica ladder que usan el puerto como un maestro. Después, el puerto vuelve al modo esclavo hasta que la lógica ladder use el puerto otra vez.

- **Baud Rate(Velocidad en Baud)**: Las tasas disponibles de baud incluyen 300, 600, 1200, 2400, 4800, 9600, 19200, y 38400 Baud. Escoja una tasa más alta de baud inicialmente, y baje el valor si experimenta errores de datos o problemas de ruido en la red. *Importante: Usted debe configurar las tasas de Baud de todos aparatos en la red al mismo valor.*

- **Stop Bits(Bits de Parada)**: Escoja 1 o 2 bits de parada para el uso en el protocolo.

- **Parity**: Escoja ninguno, par o impar (0 (nada) , Odd, Even) para verificar errores.

- **Echo Suppression**: Seleccione el botón de radio apropiado basado en la configuración del cableado usado en el puerto 2.

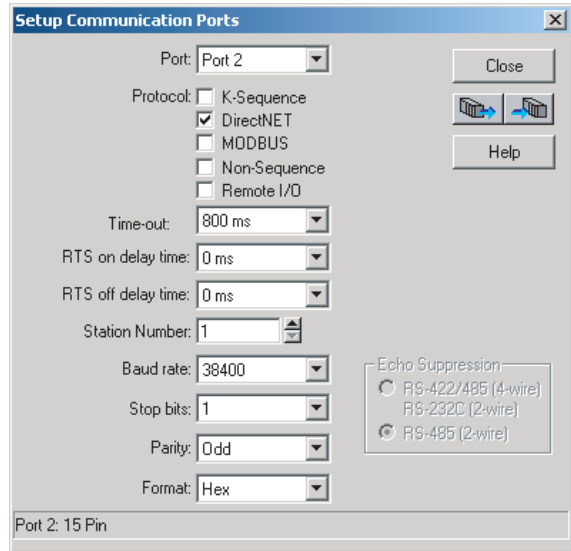


Luego haga clic en el botón indicado para enviar la configuración del puerto a la CPU y haga click en CLOSE.

Configuración del puerto como *DirectNET*

En *DirectSOFT*, escoja el menú PLC, luego Setup, luego “Setup Second. Comm Port”.

- **Port:** De la lista de números de puertos, escoja "Port 2".
- **Protocol:** haga clic en el cuadro a la izquierda de "DirectNET" (use AUX 56 en el HPP, luego seleccione "DNET"), y luego ud. verá el cuadro de diálogo de la figura abajo:
- **Timeout:** El período que el puerto esperará después que envíe un mensaje para obtener una respuesta antes de detectar un error.
- **RTS ON / OFF Delay Time:** El RTS ON delay time especifica el tiempo que espera el PLC DL06 para mandar datos después que la señal TRS se ha hecho ON. El RTS OFF Delay Time Especifica el tiempo que espera el PLC DL06 después de mandar datos para hacer OFF la señal TRS. El tiempo de atraso RTS ON debe ser colocado a por lo menos 5 ms y el tiempo de Demora RST OFF debe ser colocado a por lo menos 2ms. Si usted encuentra problemas, el tiempo se puede aumentar.



- **Station Number** (Número de esclavo): Para hacer el puerto de la CPU un maestro de *DirectNET*, escoja "1". El rango posible para números de esclavo de *DirectNET* es de 1 a 90. Cada esclavo debe tener un número único. Durante la energización el puerto es automáticamente un esclavo, a menos que y hasta que el DL06 ejecute las instrucciones de la red de la lógica ladder que usan el puerto como un maestro. Después, el puerto vuelve al modo esclavo hasta que la lógica ladder use el puerto otra vez.
- **Baud Rate** (La velocidad en Baud): Las tasas disponibles de baud incluyen 300, 600, 1200, 2400, 4800, 9600, 19200, y 38400 Baud. Escoja una tasa más alta de baud inicialmente, y baje el valor si experimenta errores de datos o problemas de ruido en la red. Importante: *Usted debe configurar las tasas de Baud de todos aparatos en la red al mismo valor.*
- **Stop Bits:** Escoja 1 o 2 bits de parada para el uso en el protocolo.
- **Parity:** Escoja ninguno, par, o impar (0 (nada), Odd o Even respectivamente) para verificar errores.
- **Format:** (Formato): escoja entre formatos hexadecimal o ASCII

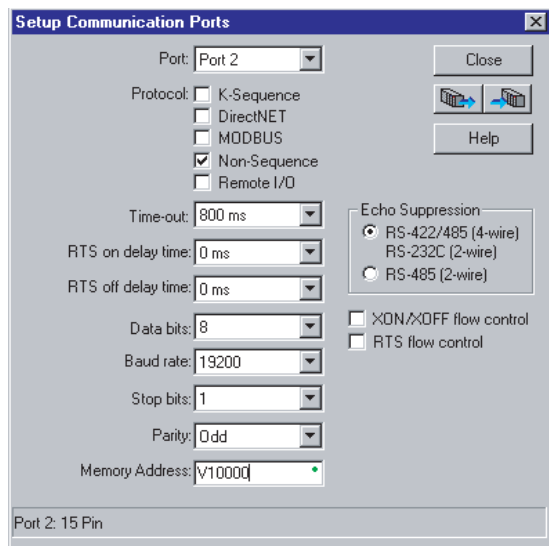


Luego haga clic el botón indicado para mandar la configuración del Puerto a la CPU, y luego haga clic en CLOSE.

Configuración del puerto como Non-Sequence (ASCII)

Configurando el puerto 2 en el DL06 para Non-Sequence permite que la CPU use el puerto 2 para leer o escribir secuencias naturales ASCII usando las instrucciones ASCII. Vea las instrucciones ASCII In/Out y la instrucción PRINT en el capítulo 5.

En *DirectSOFT*, escoja el menú PLC, luego SETUP y luego "Set Up Sec Comm Port" y luego verá el cuadro de diálogo de la figura de abajo.



- **Port:** De la lista de números de puertos escoja "Port 2".
- **Protocol:** Haga clic en el cuadro de verificación a la izquierda de "Non-Sequence".
- **Timeout:** El período que el puerto esperará después que envíe un mensaje para obtener una respuesta antes de detectar un error.
- **RTS On Delay Time:** tiempo que espera el PLC para mandar datos después que la señal TRS se ha hecho ON.
- **RTS Off Delay Time:** tiempo que espera el PLC DL06 después de mandar datos para hacer OFF la señal TRS.
- **Data Bits:** Seleccione 7 o 8 bits y hágalo igual a los bits de datos especificados para los aparatos conectados.
- **Baud Rate:** Las tasas disponibles de baud incluyen 300, 600, 1200, 2400, 4800, 9600, 19200, y 38400 Baud. Escoja una tasa más alta de baud inicialmente, y baje el valor si experimenta errores de datos o problemas de ruido en la red. Importante: *Usted debe configurar las tasas de Baud de todos aparatos en la red al mismo valor.*
- **Stop Bits:** Escoja 1 o 2 bits de parada que debe ser los mismos que los de los aparatos conectados.
- **Parity:** Escoja paridad none, even, o odd para verificación de error. Asegúrese de hacer igual la paridad especificada en los aparatos conectados.
- **Echo Suppression:** Seleccione el botón de radio adecuado basado en la configuración usada en el puerto 2 (RS-232C, RS-422 o RS-485).
- **Xon/Xoff Flow controls:** Escoja esta selección si Ud. tiene el puerto 2 conectado para el control de flujo con hardware (Xon/Xoff) con las señales RTS y CTS conectada entre los dispositivos.
- **RTS Flow controls:** Escoja esta selección si ud. tiene la señal RTS del puerto 2 cableada entre los aparatos.
- **Memory address:** Escoja una dirección de memoria para usar como buffer para almacenamiento de datos ASCII.



Luego haga clic en el botón indicado para enviar la configuración del puerto a la CPU y haga clic en CLOSE.

Comunicación con K-Sequence

El protocolo K-Sequence se puede usar para comunicación con *DirectSOFT*, una interface de operador o cualquier otro dispositivo que puedan ser un maestro con K-Sequence. El PLC DL06 puede ser un esclavo K-Sequence en el puerto 1 o el puerto 2. El PLC DL06 no puede ser un maestro K-Sequence.

Para utilizar el puerto 2 para comunicaciones K-Sequence usted primero necesita configurar el puerto usando *DirectSOFT* o lógica ladder según lo descrito previamente.

Comunicación con *DirectNET*

La red es controlada por una estación maestra que da órdenes de intercambio de datos a estaciones individuales de esclavo en una red serial. (No se puede transmitir "simultáneamente" un mensaje a todos los esclavos). Las órdenes se pueden usar para enviar los datos a las estaciones esclavas o enviar los datos desde las estaciones esclavas. Las estaciones esclavas sólo responden a pedidos de la estación maestra y no pueden iniciar comunicaciones.

Esta red usa el protocolo de comunicaciones *DirectNET*, pero usted no tiene que entender el protocolo para construir las configuraciones de redes más comunes.

Para más detalles en relación con *DirectNET*, ordene el manual de *DirectNET* desde AutomationDirect. El número de parte DA-DNET-M o también puede bajarlo desde el sitio de Internet de AUTOMATIONDIRECT www.automationdirect.com, en forma gratuita. Vea también otros conceptos y un ejemplo de comunicaciones en el apéndice K.

El protocolo *DirectNET* se puede usar para comunicarse a otro PLC o a otros dispositivos que puedan usar el protocolo *DirectNET*. El PLC DL06 puede ser usado como maestro en el puerto 2 o como esclavo usando el puerto 1 o el puerto 2 o aún el módulo D0-DCM.

Muchos paneles de operador, incluso algunos de los que vende AUTOMATIONDIRECT, se pueden usar como la estación maestra para comunicarse con uno o más esclavos. Un panel de operador debe tener un driver que permite el protocolo *DirectNET*. También es posible usar un PC con el driver adecuado. De hecho la comunicación de *DirectSOFT* para las CPUs D3-330 y D3-340 es *DirectNET*. Está disponible la descripción del protocolo *DirectNET*, en el manual nombrado anteriormente, para poder crear un driver e incluso tiene ejemplos en BASIC para poder generar comandos desde un PC.

Comunicación con MODBUS RTU

Se puede usar el protocolo MODBUS RTU para comunicación con cualquier dispositivo que utilice el protocolo MODBUS RTU. El protocolo es muy común y es probablemente la cosa más cercana a un protocolo estándar "industrial" en existencia. El PLC DL06 puede ser un esclavo de MODBUS RTU en el puerto 1 o el puerto 2, y puede ser un maestro MODBUS RTU en el puerto 2. El estándar RS 485 se puede utilizar solamente en el puerto 2 para el protocolo MODBUS RTU.

La forma de conectar el maestro o los esclavos es muy similar a la forma hecha con *DirectNET* en el caso de RS-232 or RS-422. Vea ejemplos en el apéndice K.

Comunicación con ASCII

Se puede usar el protocolo Non-sequence para leer y escribir datos ASCII. Vea la sección del capítulo 5 relacionada con las intrucciones ASCII.

Operación de un esclavo en MODBUS RTU

Esta sección describe cómo otros dispositivos en una red pueden comunicarse con un puerto del PLC DL06 que usted ha configurado como un esclavo MODBUS. Un anfitrión de MODBUS debe utilizar el protocolo MODBUS RTU para comunicarse con el DL06 como esclavo. El software del anfitrión debe enviar un código de una función de MODBUS y dirección de MODBUS para especificar una memoria del PLC que el DL06 comprenda. No se requiere ninguna lógica ladder en el PLC para permitir usar el esclavo de MODBUS.

Códigos de función MODBUS en el PLC

El código de función MODBUS determina si la transferencia es escritura o lectura y si va a transferir un bit de datos o un grupo de ellos. El PLC DL06 permite usar los códigos de función de MODBUS descritos debajo.

Códigos MODBUS	Función	Tipos de datos disponibles
01	Lea un grupo de bobinas	Y, CR, T, CT
02	Lea un grupo de entradas	X, SP
05	Active / desactive una bobina solamente	Y, CR, T, CT
15	Active / desactive un grupo de bobinas	CR, T, CT
03, 04	Lea un valor desde una o más memorias	V
06	Escriba un valor a una memoria	V
16	Escriba un valor a una o más memorias	V

El sistema MODBUS usa convenciones de direcciones diferentes que las direcciones de los PLCs *DirectLOGIC*. Con *DirectNET* la convención de dirección es única. Otros PLCs *DirectLOGIC* deben hacer un cruzamiento a las direcciones de MODBUS de acuerdo a la tabla ejemplo a continuación.

	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17	Memoria V
X0	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063	40400
X20	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	40401
X40	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	40402
X60	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	40403
X100	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	40404
X120	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	40405
X140	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	40406

Determinando la dirección de MODBUS

Hay dos formas típicas en que la mayoría de las convenciones de software anfitrión le permite especificar una dirección de memoria de un PLC. Estas son:

- Especificando el tipo de datos y la dirección de MODBUS
- Especificando la dirección de MODBUS solamente.

Si el software anfitrión requiere el tipo de datos y dirección

Muchos paquetes de software anfitrión permiten que usted especifique el tipo de datos de MODBUS y la dirección de MODBUS que corresponde a la dirección de memoria del PLC. Éste es el método más fácil, pero no todos los paquetes permiten que usted lo haga de esta manera.

La ecuación usada para calcular la dirección depende del tipo de datos del PLC que usted está utilizando. Los tipos de memoria del PLC están divididos en dos categorías para este propósito.

- Discreto – X, SP, Y, CR, S, T, C (contactos)
- Palabra – V, valor corriente del temporizador, valor corriente del contador.

En cualquier caso, usted convierte básicamente la dirección octal del PLC a decimal y suma la dirección apropiada de MODBUS (si es requerido). La tabla abajo muestra la ecuación exacta usada para cada grupo de datos.

Tipo de memoria del DL06	Cantidad (Decimal)	Rango del PLC (Octal)	Rango direcciones MODBUS (Decimal)	Tipo de datos MODBUS
Para tipos de datos discretos Convierta la dirección del PLC a decimal + inicio del rango + tipo de datos				
Entradas (X)	512	X0 – X777	2048 – 2559	Entrada
Relevadores especiales(SP)	512	SP0 – SP777	3072 – 3583	Entrada
salidas (Y)	512	Y0 – Y777	2048 – 2559	Bobina
Relevadores de control (CR)	1024	C0 – C1777	3072 – 4095	Bobina
Contactos de temporizador (T)	256	T0 – T377	6144 – 6399	Bobina
Contactos de contador (CT)	128	CT0 – CT177	6400 – 6527	Bobina
Bits de estado de etapas(S)	1024	S0 – S1777	5120 – 6143	Bobina
Para tipos de datos de palabras Convierta la dirección del PLC a decimal + tipo de datos				
Valor corriente de T (V)	256	V0 – V377	0 – 255	Entrada
Valor corriente de CT (V)	128	V1000 – V1177	512 – 639	Input register
Memoria V, datos usuario (V)	3200	V1200 – V7377	640 – 3839	Holding Register
	4096	V10000 - V17777	4096 - 8191	Holding Register
Memoria V, no-volátil (V)	128	V7400 – V7577	3840 – 3967	Holding Register

Capítulo 4: Diseño y configuración del sistema

Los ejemplos siguientes muestran cómo generar la dirección y el tipo de datos MODBUS para los anfitriones que necesitan este formato.

Ejemplo 1: V2100

Encuentre la dirección de MODBUS para la dirección V2100.

1. Encuentre la memoria V en la tabla.
2. Convierta V2100 a decimal (1088).
3. Use el tipo de datos de MODBUS de la tabla.

Holding Reg 1088

Datos de memoria (V)	3200	V1200 – V7377	640 – 3839	Holding Register
----------------------	------	---------------	------------	------------------

Ejemplo 2: Y20

Encuentre la dirección de MODBUS para la salida Y20.

1. Encuentre las salidas Y en la tabla.
2. Convierta Y20 a decimal (16).
3. Sume la dirección inicial del rango (2048).
4. Use el tipo de datos de MODBUS de la tabla.

Bobina 2064

Salidas (V)	256	Y0 – Y377	2048 - 2303	Bobina
-------------	-----	-----------	-------------	--------

Ejemplo 3: Valor corriente de T10

Encuentre la dirección de MODBUS para obtener el valor corriente del temporizador T10.

1. Encuentre valores corrientes de temporizador en la tabla.
2. Convierta T10 a decimal (8).
3. Use el tipo de datos de MODBUS de la tabla.

Memoria de entrada 8

Valores corrientes de T (V)	128	V0 – V177	0 - 127	Memoria de entrada
-----------------------------	-----	-----------	---------	--------------------

Ejemplo 4: Relevador de control C54

Encuentre la dirección de MODBUS para el relai de control C54.

1. Encuentre relevadores de control en la tabla.
2. Convierta C54 a decimal (44).
3. Sume la dirección inicial del rango (3072).
4. Use el tipo de datos de MODBUS de la tabla.

Bobina 3116

Relevadores de control (CR)	512	C0 – C77	3072 – 3583	Bobina
-----------------------------	-----	----------	-------------	--------

Si su software de anfitrión de MODBUS SOLAMENTE necesita una dirección

Algunos software anfitriones no permiten que se especifique el tipo y la dirección de datos de MODBUS. En ese caso, usted debe especificar una dirección solamente. Este método requiere otro paso determinar la dirección, pero sigue siendo bastante simple. Básicamente MODBUS también separa los tipos de datos por los rangos de dirección también. Esto significa que solamente una dirección puede describir realmente el tipo de datos y de dirección. Esto se refiere a menudo como "sumando el offset". Una cosa importante aquí es que dos modos de dirección diferentes pueden estar disponibles en su paquete de software de anfitrión. Éstos son:

- Modo 484
- Modo 584/984

Recomendamos que utilice el modo de dirección de 584/984 si su software anfitrión le permite elegir. Esto es, porque el modo de 584/984 permite el acceso a una cantidad más grande de direcciones de memoria dentro de cada tipo de datos. Si su software apoya solamente el modo 484, entonces puede haber algunas posiciones de memoria del PLC que serán inasequibles. La ecuación real usada para calcular la dirección depende del tipo de datos del PLC que usted está utilizando. Los tipos de I memoria del PLC están divididos en dos categorías para este propósito.

- Discretas – X, SP, Y, CR, S, T (contactos), CT (contactos)
- Palabra – V, valor corriente del temporizador, valor corriente del contador,

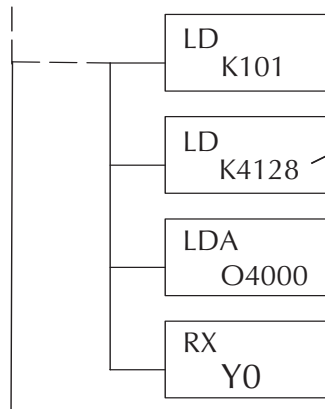
En cualquier caso, usted convierte básicamente la dirección octal del PLC a decimal y suma las direcciones apropiadas de MODBUS (de acuerdo a lo requerido). La tabla de abajo muestra la ecuación exacta usada para cada grupo de datos.

Tipo de de datos discretos				
Tipo de memoria del DL06	Rango del PLC (Octal)	(Modo 484)	Dirección (Modo 584/984)	Tipo datos MODBUS
Entradas globales(GX)	GX0-GX1746	1001 - 1999	10001 - 10999	Input
	GX1747-GX3777	---	11000 - 12048	Input
Entradas (X)	X0 – X1777	---	12049 - 13072	Input
Relevadores especiales (SP)	SP0 – SP777	---	13073 - 13584	Input
Salidas globales (GY)	GY0 - GY3777	1 - 2048	1 - 2048	Output
Salidas (Y)	Y0 – Y1777	2049 - 3072	2049 - 3072	Output
Relevadores de control (CR)	C0 – C3777	3073 - 5120	3073 - 5120	Output
Contactos de temporizadores (T)	T0 – T377	6145 - 6400	6145 - 6400	Output
Contactos de contadores (CT)	CT0 – CT377	6401 - 6656	6401 - 6656	Output
Bits de estado de etapas (S)	S0 – S1777	5121 - 6144	5121 - 6144	Output

Tipos de de datos de palabra			
Memorias	Rango del PLC (Octal)	Input/Holding (484 Mode)*	Input/Holding (584/984 Mode)*
Memoria V (Temporizadores)	V0 - V377	3001/4001	30001/40001
Memoria V (Contadores)	V1000 - V1177	3513/4513	30513/40513
Memoria V (Palabras de datos)	V1200 - V1377	3641/4641	30641/40641
	V1400 - V1746	3769/4769	30769/40769
	V1747 - V1777	---	31000/41000
	V2000 - V7377	---	41025
	V10000 - V17777	---	44097

* MODBUS: Función 04

Los PLCs DL05/06 y las CPUs DL250-1/260, DL350 y DL450 permiten usar la función 04, lea el registro de entrada (dirección 30001). Para utilizar la función 04, coloque el número "4" en la posición más significativa (4xxx), cuando no usa la instrucción MRX. Se deben entrar cuatro dígitos para que la intrucción trabaje correctamente con este modo.



La constante posible máxima es 4128. Esto es debido al número máximo de 128 bytes que puede permitir la instrucción RX/WX. El valor de 4 en la posición más significativa de la palabra hará que la instrucción RX use la función 04 (rango 30001).



1. Vea el manual de usuario del PLC si es que no usa el PLC DL06, para el tamaño correcto de la memoria de su PLC. Algunas de las direcciones mostradas arriba puede ser que no pertenezcan a su CPU particular.



2. Hay un programa automatizado en EXCEL con la conversión de dirección de MODBUS para PLCs **DirectLOGIC** y es el archivo **modbus_conversion.xls** que se encuentra en el sitio de Internet www.automationdirect.com y que puede ser bajado gratuitamente. (Referencia : Apoyo técnico>Página inicial de apoyo técnico>Notas técnicas y de aplicaciones > Communications> AN-MISC-010)

Ejemplo 1: V2100 con modo 584/984

Encuentre la dirección MODBUS para la dirección V2100 Dirección PLC(Dec.) + modo

- Encuentre la memoria en la tabla V2100 = 1088 decimal
- Convierta V2100 a decimal (1088). 1088 + 40001 = **41089**
- Sume la dirección inicial MODBUS para el modo (40001).

Para tipo de datos de palabra....		Dirección del PLC(Dec.)		+	Dirección del modo	
Valores corrientes de T (V)	128	V0 – V177	0 – 127	3001	30001	Input Register
Valores corrientes de CT(V)	128	V1200 – V7377	512 – 639	3001	30001	Input Register
Memoria, datos de usuario (V)	1024	V2000 – V3777	1024 – 2047	4001	40001	Holding Register

Ejemplo 2: Y20 con modo 584/984

Encuentre la dirección MODBUS para la salida Y20. Dir. PLC (Dec.) + Direc. inicial + modo

- Encuentre las salidas Y en la tabla. Y20 = 16 decimal
- Convierta Y20 a decimal (16). 16 + 2048 + 1 = **2065**
- Sume la dirección inicial para el rango (2048).
- Sume la dirección MODBUS para el modo (1).

Salidas (Y)	320	Y0 - Y477	2048 - 2367	1	1	Bobina
Relevadores de control (CR)	256	C0 - C377	3072 - 3551	1	1	Bobina
Contactos de tempor. (T)	128	T0 - T177	6144 - 6271	1	1	Bobina

Ejemplo 3: Valor corriente de T10 con el modo 484

Encuentre la dirección MODBUS para obtener el valor corriente de T10. Dirección del PLC (Dec.) + Modo
TA10 = 8 decimal

- Encuentre el valor corriente en la tabla. 8 + 3001 = **3009**
- Convierta T10 a decimal (8).
- Sume la dirección inicial MODBUS para el modo (3001).

Para tipos de datos de palabras....		dirección dePLC(Dec.)		+	dirección del modo	
Valores corrientes de T (V)	128	V0 – V177	0 – 127	3001	30001	Input Register
Valores corrientes de CT(V)	128	V1200 – V7377	512 – 639	3001	30001	Input Register
Memoria, datos de usuario (V)	1024	V2000 – V3777	1024 – 2047	4001	40001	Holding Register

Ejemplo 4: C54 con el modo 584/984

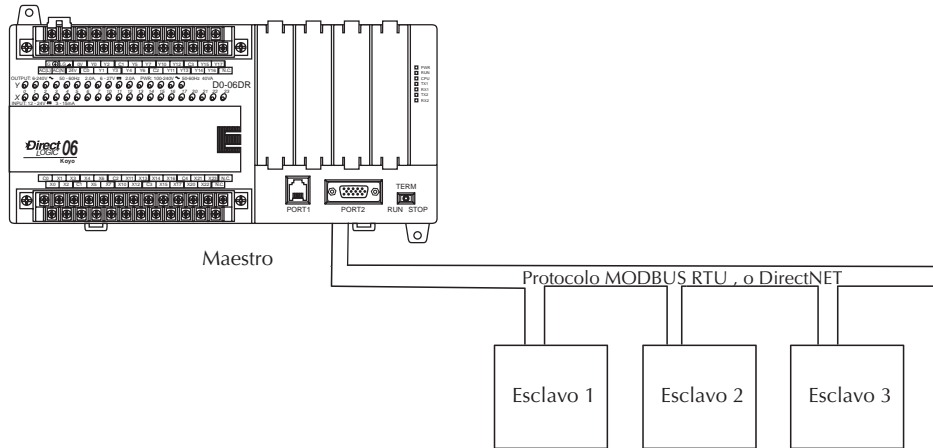
Encuentre la dirección MODBUS para C54. Dirección del PLC (Dec.)+Dir. inicial+ modo

- Encuentre relevadores de control en la tabla. C54 = 44 decimal
- Convierta C54 a decimal (44). 44 + 3072 + 1 = **3117**
- Sume la dirección inicial para el rango (3072).
- Sume la dirección MODBUS para el modo (1).

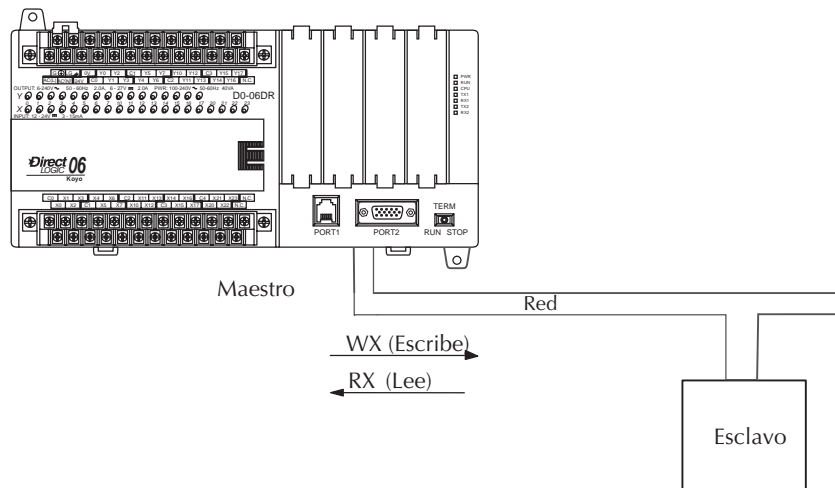
Salidas (Y)	320	Y0 - Y477	2048 - 2367	1	1	Bobina
Relevadores de control (CR)	256	C0 - C377	3072 - 3551	1	1	Bobina
Contactos de tempor. (T)	128	T0 - T177	6144 - 6271	1	1	Bobina

Operación del maestro en una red MODBUS RTU

Esta sección describe cómo el PLC DL06 puede comunicarse en una red de MODBUS como maestro (también es válido para *DirectNET*). Para las redes de MODBUS, use el protocolo MODBUS RTU, que se debe interpretar por todos los esclavos en la red. MODBUS y *DirectNet* son redes de un maestro y múltiples esclavos. El maestro es el único miembro de la red que puede iniciar peticiones en la red. Esta sección le enseña cómo diseñar la lógica requerida para operación del maestro.



Al usar el PLC DL06 como maestro, se usan instrucciones simples para iniciar las peticiones. La instrucción WX inicia las operaciones de escritura a la red y RX inicia operaciones de lectura de la red. Antes de ejecutar WX o RX, necesitamos cargar los datos relacionados con la operación de lectura o escritura en el stack del acumulador. Cuando se ejecuta la instrucción WX o RX, usa la información en el stack combinado con datos en la instrucción para definir totalmente la tarea, que va al puerto correspondiente.

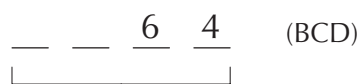
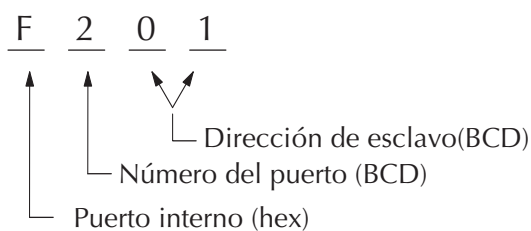


Paso 1: Identifique el numero del puerto maestro y el número de cada esclavo

La primera instrucción LD identifica el número del puerto maestro en la red DL06 y la dirección del esclavo con el cual se harán las transferencias de datos. Esta instrucción puede direccionar hasta 99 esclavos en MODBUS (o 90 esclavos de *DirectNET*).

El formato de palabra se muestra a la derecha. El "F2" en el byte superior indica el uso del puerto correcto del PLC DL06, que es el puerto 2. Recuerde que el puerto 2 es el único puerto en el PLC DL06 que puede ser maestro.

El byte más bajo contiene el número de la dirección del esclavo en BCD (01 a 99).



↑
No. de bytes a transferir



Paso 2: Cargue el no. de bytes a transferir:

La segunda instrucción LD determina el número de bytes que se transferirán entre el maestro y el esclavo en la instrucción siguiente WX o RX. El valor a ser cargado está en formato BCD, de 1 a 128 bytes.

El número de bytes especificados depende también del tipo de datos que quiere obtener. Por ejemplo, los puntos de entrada DL06 pueden obtenerse por memorias V o como localizaciones de entrada X. Sin embargo, si usted sólo quiere X0 - X27, usted tendrá que usar el tipo de datos de entrada X porque las ubicaciones de memoria V pueden ser obtenidas en incrementos de 2 bytes.

La siguiente tabla muestra los rangos de bytes para los varios tipos de productos *DirectLOGIC*TM.

Memoria DL05 / 06 / 205 / 350 / 405	Bits por unidad	Bytes
Memoria V del valor corriente de temporizador o contador	16 16	2 2
Entradas (X, SP)	8	1
salidas (Y, C, etapas, bits de T/CT)	8	1
Memoria Scratch Pad	8	1
Estado de diagnóstico	8	1

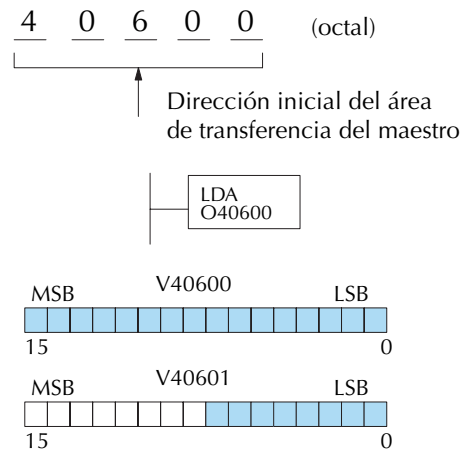
Memoria DL330 / 340	Bits por unidad	Bytes
Memorias de datos	8	1
Acumulador de T/CT	16	2
E/S, relevadores internos, bits de shift register, bits de T/CT, bits de etapas	1	1
Memoria Scratch Pad	8	1
Estado de diagnóstico (5 palabras R/W)	16	10

Paso 3: Especifique la memoria del maestro

La tercera instrucción en el programa RX o WX es una instrucción LDA. Su propósito es cargar la dirección inicial del área de memoria a ser transferida. Es entrado como un número octal y la instrucción LDA la convierte a hexadecimal y coloca el resultado en el acumulador.

Para una instrucción WX, la CPU DL06 manda el número de bytes previamente especificado al área de memoria en la dirección inicial especificada en la instrucción LDA.

Para una instrucción RX, la CPU DL06 lee el número de bytes previamente especificados del esclavo, colocando los datos recibidos en el área de memoria en la dirección inicial especificada en la instrucción LDA.

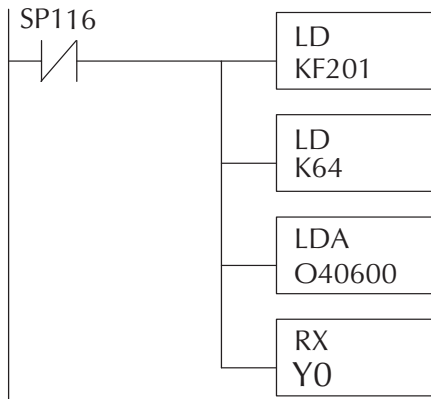


NOTA: Ya que las palabras de memoria V son siempre de 16 bits, no siempre se puede usar la palabra entera. Por ejemplo, si sólo se especifica 3 bytes y se lee las salidas Y del esclavo, sólo se obtiene 24 bits de datos. En este caso, sólo los 8 bits menos significativos de la última dirección de palabra se modificarán. Los restantes 8 bits no son afectados.

Paso 4: Especifique la memoria del esclavo

La última instrucción en nuestro programa es la instrucción WX o RX. Use WX para escribir al esclavo, y RX para leer desde el esclavo. Todas las cuatro instrucciones se muestran a la derecha. En la última instrucción, usted debe especificar la dirección inicial y un tipo válido de datos para el esclavo.

- Esclavos de *DirectNET* – especifique la misma dirección en la instrucción WX y RX como la dirección nativa del esclavo.
- Esclavos MODBUS DL405, DL205, o DL06 – especifique la misma dirección en la instrucción WX y RX como la dirección nativa del esclavo.
- Esclavos MODBUS 305 – use la siguiente tabla para convertir direcciones DL305 a direcciones MODBUS.



Correspondencia de la memoria de las CPUs DL305 a MODBUS (excluyendo la CPU 350)					
Tipo de memoria del PLC	Dirección base del PLC	Dirección base MODBUS	Tipo de memoria PLC	Dirección base del PLC	Dirección base MODBUS
Valores corrientes de TMR/CNT	R600	V0	Bits de estado de TMR/CNT	CT600	GY600
Puntos de E/S	IO 000	GY0	Relev. control	CR160	GY160
Registros de datos	R401, R400	V100	Shift Registers	SR400	GY400
Bits estado de etapas (D3-330P)	S0	GY200			

Comunicaciones desde un programa ladder

Típicamente las comunicaciones de red durarán más que 1 barrido de la CPU. El programa debe esperar que termine la transmisión de los datos en la comunicación antes de comenzar la próxima transacción.

El Puerto 2, que puede ser un maestro, tiene dos contactos de relevador especial asociados con el.

Uno indica " Puerto Ocupado " (SP116), y el otro indica "Puerto con error de comunicación" (SP117).

El ejemplo adyacente muestra el uso de estos contactos para una red con un maestro que sólo lee un aparato (RX). El bit de "Puerto ocupado" está ON mientras el PLC se comunica con el esclavo. Cuando el bit está apagado el programa puede iniciar el próximo pedido de la red. El bit "Puerto con error de comunicación" se activa cuando el PLC ha detectado un error. El uso de este bit es opcional. Cuando se usa, debe ser adelante de cualquier instrucción de red ya que el bit de error es repone cuando se ejecuta una de las instrucciones RX o WX.

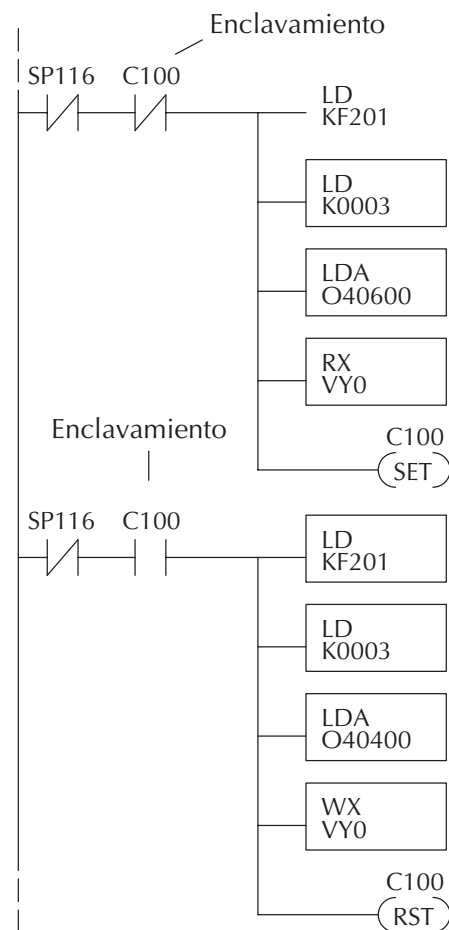
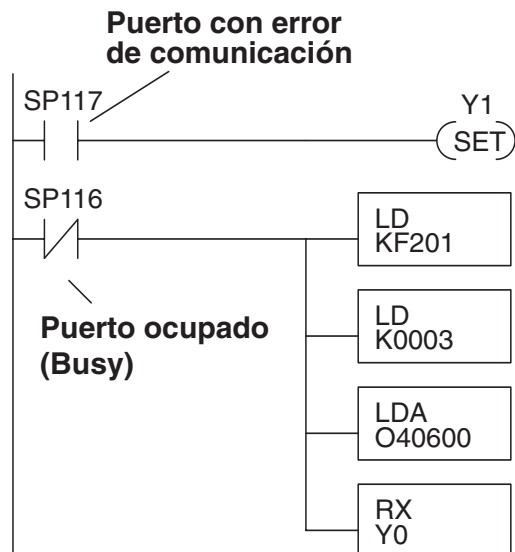
Enclavamientos múltiples para leer y escribir

Si usted usa varias instrucciones WX y RX en el programa ladder, usted tiene que enclavar las rutinas para asegurarse que todas las rutinas se ejecutan completas.

Si usted no usa el enclavamiento, entonces la CPU sólo ejecuta la primera rutina. Esto es porque cada puerto puede hacer sólo una transacción a la vez.

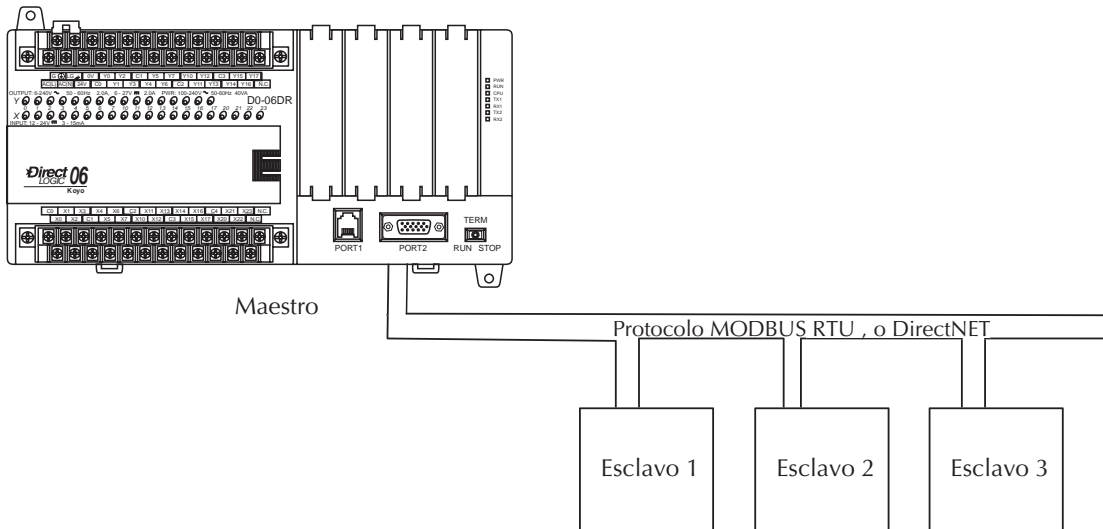
En el ejemplo a la derecha después que se ejecuta la instrucción RX, C100 se activa. Cuando el puerto ha terminado la tarea de comunicación, la segunda rutina se ejecuta y C100 es desactiva.

Si usted usa programación de etapas RLL^{PLUS}, Ud. puede colocar cada rutina en una etapa separada del programa para asegurarse que la ejecución salte de etapa a etapa para permitir sólo que una de ellas sea activada.



Operación como maestro en una red (Usando instrucciones MRX y MWX)

Esta sección describe cómo el DL06 puede comunicarse en una red MODBUS RTU como un maestro usando las instrucciones MRX y MWX. Estas instrucciones permiten que usted entre direccionamiento nativo MODBUS en su programa de lógica ladder sin necesidad de realizar las conversiones octal a decimal. MODBUS es una sola red de un maestro y múltiples esclavos. El maestro es el único miembro de la red que puede iniciar peticiones en la red. Esta sección le enseña cómo diseñar la lógica requerida para la operación del maestro.



Códigos de función MODBUS posibles de usar

El código de función MODBUS determina si la transferencia de datos es lectura o escritura y si se tiene acceso a un solo punto de referencia o a un grupo de ellas. El DL06 permite usar los códigos de función MODBUS descritos abajo.

Código de función MODBUS	Función	Tipos de datos en el DL06
01	Lea un grupo de bobinas	Y, CR, T, CT
02	Lea un grupo de entradas	X, SP
05	Active o desactive una bobina (solamente esclavo)	Y, CR, T, CT
15	Active o desactive un grupo de bobinas	Y, CR, T, CT
03, 04	Lea un valor desde uno o mas registros	V
06	Lea un valor desde solo un registro (sólo esclavo)	V
07	Lea un estado de excepción (Exception status)	V
08	Diagnósticos	V
16	Escriba un valor a un grupo de registros	V

Ejemplo de MRX/MWX en *DirectSOFT*

Vea un ejemplo en el capítulo 5, después de la instrucción MWX. El puerto 2 del DL06 tiene dos contactos de relevadores especiales asociados a él (véa el apéndice D para relevadores especiales). Uno indica el "puerto ocupado" (SP116) y el otro indica "Error de comunicación del puerto" (SP117).

El bit "puerto ocupado" está encendido mientras el PLC se comunica con el esclavo. Cuando el bit está OFF, el programa puede iniciar la petición siguiente de la red.

El bit "error del puerto" se hace ON cuando el PLC ha detectado un error; el uso de este bit es opcional. Cuando es usado, debe estar delante de cualquier bloque de instrucción de red puesto que se reajusta el bit del error cuando se ejecuta una instrucción MRX o MWX.

Las comunicaciones de red durarán típicamente más que un barrido de la CPU. El programa debe esperar que se termine la transferencia de datos antes de comenzar la transacción siguiente.

4

Enclavamientos múltiples de lectura y escritura

Si usted está utilizando lecturas y escrituras múltiples en el programa ladder, usted tiene que enclavar las rutinas para asegurarse de que todas las rutinas sean ejecutadas. Si usted no utiliza enclavamientos, entonces la CPU ejecutará solamente la primera rutina. Esto es porque cada puerto puede manejar solamente una transacción en un tiempo dado.

Ud. puede usar un contador para hacer el enclavamiento, o una instrucción shift register.

Si usted está utilizando la programación por etapas, usted puede poner cada rutina en una etapa separada del programa para asegurar la ejecución y la conmutación adecuadas de etapa a etapa permitiendo que solamente uno de ellos sea activo a la vez.

Lea más sobre estas instrucciones en el capítulo 5. Allí se muestra un ejemplo incluyendo enclavamiento entre instrucciones de modo que solamente una instrucción se ejecute en un momento dado, de la misma forma que son usadas las instrucciones RX y MX.

Operación con caracteres ASCII

Vea detalles de como establecer comunicación ASCII en el capítulo 5 y también en el apéndice K.

INSTRUCCIONES RLL DEL PLC DL06



En este capítulo

Introducción	5-2
Usando instrucciones booleanas	5-5
Instrucciones booleanas	5-10
Instrucciones de comparación booleanas	5-26
Instrucciones de acción inmediata	5-32
Instrucciones de temporizadores, contadores y Shift Register ...	5-39
Operaciones de carga y copia con el acumulador y Stack	5-52
Instrucciones lógicas (Acumulador)	5-69
Instrucciones aritméticas	5-86
Instrucciones de funciones transcendentales	5-118
Instrucciones de operación con bits	5-120
Instrucciones de conversión de números (Acumulador)	5-127
Instrucciones de tablas	5-141
Instrucciones de fecha y hora	5-171
Instrucciones de control de la CPU	5-173
Instrucciones de control de programa	5-175
Instrucciones de Interrupción	5-183
Instrucciones de mensajes	5-186
Instrucciones de MODBUS RTU	5-201
Instrucciones de texto ASCII	5-210
Instrucciones de tipo Intelligent Box (IBox)	5-230

Introducción

Los PLCs DL06 ofrecen una amplia variedad de instrucciones para realizar diversos tipos de operaciones. Este capítulo le muestra cómo utilizar cada instrucción normal de lógica ladder de relevadores (RLL). Además de estas instrucciones, usted puede también necesitar referirse a las instrucciones de tambor (DRUM) en el capítulo 6, o a las instrucciones de programación por etapas en el capítulo 7.

Hay dos formas de encontrar rápidamente la instrucción que usted necesita.

- Si sabe la categoría de la instrucción (booleana, comparativos booleanos, etc.) use el título en la parte superior de las páginas para encontrar las páginas que discuten las instrucciones en esa categoría.
- Si usted sabe el nombre individual de la instrucción, utilice el índice siguiente para encontrar la página que discute la instrucción.

5

Instrucción	Página	Instrucción	Página
Accumulating Fast Timer (TMRAF)	5-42	And Store (AND STR)	5-16
Accumulating Timer (TMRA)	5-42	And with Stack (ANDS)	5-72
Add (ADD)	5-86	Arc Cosine Real (ACOSR)	5-119
Add Binary (ADDB)	5-99	Arc Sine Real (ASINR)	5-118
Add Binary Double (ADDBD)	5-100	Arc Tangent Real (ATANR)	5-119
Add Binary Top of Stack (ADDBS)	5-114	ASCII Clear Buffer (ACRB)	5-228
Add Double (ADDD)	5-87	ASCII Compare (CMPV)	5-220
Add Formatted (ADDF)	5-106	ASCII Constante (ACON)	5-187
Add Real (ADDR)	5-88	ASCII Extract (AEX)	5-219
Add to Top (ATT)	5-162	ASCII Find (AFIND)	5-216
Add Top of Stack (ADDS)	5-110	ASCII Input (AIN)	5-212
And (AND)	5-14	ASCII Print from V-memory (PRINTV)	5-226
And Bit-of-Word (AND)	5-15	ASCII Print to V-memory (VPRINT)	5-221
And (AND)	5-31	ASCII Swap Bytes (SWAPB)	5-227
AND (AND logical)	5-69	ASCII to HEX (ATH)	5-134
And Double (ANDD)	5-70	Binary (BIN)	5-127
And Formatted (ANDF)	5-71	Binary Coded Decimal (BCD)	5-128
And If Equal (ANDE)	5-28	Binary to Real Conversion (BTOR)	5-131
And If Not Equal (ANDNE)	5-28	Compare (CMP)	5-81
And Immediate (ANDI)	5-33	Compare Double (CMPD)	5-82
AND Move (ANDMOV)	5-167	Compare Formatted (CMPF)	5-83
And Negative Differential (ANDND)	5-22	Compare Real Number (CMPR)	5-85
And Not (ANDN)	5-14	Compare with Stack (CMPS)	5-84
And Not Bit-of-Word (ANDN)	5-15	Cosine Real (COSR)	5-118
And Not (ANDN)	5-31	Contador (CNT)	5-45
And Not Immediate (ANDNI)	5-33	Data Label (DLBL)	5-187
And Positive Differential (ANDPD)	5-22	Date (DATE)	5-171

Instrucción	Página	Instrucción	Página
Decode (DECO)	5-126	Load Accumulator Indexed from Data Constantes (LDSX)	5-62
Decrement (DEC)	5-98	Load Address (LDA)	5-60
Decrement Binary (DECB)	5-105	Load Double (LDD)	5-58
Degree Real Conversion (DEGR)	5-133	Load Formatted (LDF)	5-59
Disable Interrupts (DISI)	5-184	Load Immediate (LDI)	5-37
Divide (DIV)	5-95	Load Immediate Formatted (LDIF)	5-38
Divide Binary (DIVB)	5-104	Load Label (LDLBLE)	5-142
Divide Binary by Top OF Stack (DIVBS)	5-117	Load Real Number (LDR)	5-63
Divide by Top of Stack (DIVS)	5-113	Master Line Reset (MLR)	5-181
Divide Double (DIVD)	5-96	Master Line Set (MLS)	5-181
Divide Formatted (DIVF)	5-109	MODBUS Read from Network (MRX)	5-204
Divide Real (DIVR)	5-97	MODBUS Write to Network (MWX)	5-207
Enable Interrupts (ENI)	5-183	Move Block (MOVBLK)	5-189
Encode (ENCO)	5-125	Move (MOV)	5-141
End (END)	5-173	Move Memory Cartridge (MOVMC)	5-142
Exclusive Or (XOR)	5-77	Multiply (MUL)	5-92
Exclusive Or Double (XORD)	5-78	Multiply Binary (MULB)	5-103
Exclusive Or Formatted (XORF)	5-79	Multiply Binary Top of Stack (MULBS)	5-116
Exclusive OR Move (XORMOV)	5-167	Multiply Double (MULD)	5-93
Exclusive Or with Stack (XORS)	5-80	Multiply Formatted (MULF)	5-108
Fault (FAULT)	5-186	Multiply Real (MULR)	5-94
Fill (FILL)	5-146	Multiply Top of Stack (MULS)	5-112
Find (FIND)	5-147	No Operation (NOP)	5-173
Find Block (FINDB)	5-169	Not (NOT)	5-19
Find Greater Than (FDGT)	5-148	Numerical Constante (NCON)	5-187
For / Next (FOR) (NEXT)	5-176	Or (OR)	5-12
Goto Label (GOTO) (LBL)	5-175	Or (OR)	5-30
Goto Subroutine (GTS) (SBR)	5-178	Or (OR logical)	5-73
Gray Code (GRAY)	5-138	Or Bit-of-Word (OR)	5-13
HEX to ASCII (HTA)	5-135	Or Double (ORD)	5-74
Increment (INC)	5-98	Or Formatted (ORF)	5-75
Increment Binary (INCB)	5-105	Or If Equal (ORE)	5-27
Interrupt (INT)	5-183	Or Immediate (ORI)	5-32
Interrupt Return (IRT)	5-183	OR Move (ORMOV)	5-167
Interrupt Return Conditional (IRTC)	5-183	Or Negative Differential (ORND)	5-21
Invert (INV)	5-129	Or Not (ORN)	5-12
LCD	5-200	Or Not (ORN)	5-30
Load (LD)	5-57	Or Not Bit-of-Word (ORN)	5-13
Load Accumulator Indexed (LDX)	5-61	Or Not Immediate (ORNI)	5-32

Capítulo 5: Instrucciones

Instrucción	Página	Instrucción	Página
Or Out (OROUT)	5-17	Shuffle Digits (SFLDGT)	5-139
Or Out Immediate (OROUTI)	5-34	Sine Real (SINR)	5-118
Or Positive Differential (ORPD)	5-21	Source to Table (STT)	5-156
Or Store (ORSTR)	5-16	Square Root Real (SQRTR)	5-119
Or with Stack (ORS)	5-76	Etapas Contador (SGCNT)	5-47
Out (OUT)	5-17	Stop (STOP)	5-173
Out (OUT)	5-18	Store (STR)	5-10
Out Bit-of-Word (OUT)	5-64	Store (STR)	5-29
Out Double (OUTD)	5-64	Store Bit-of-Word (STRB)	5-11
Out Formatted (OUTF)	5-65	Store If Equal (STRE)	5-26
Out Immediate (OUTI)	5-34	Store If Not Equal (STRNE)	5-26
Out Immediate Formatted (OUTIF)	5-35	Store Immediate (STRI)	5-32
Out Indexed (OUTX)	5-67	Store Negative Differential (STRND)	5-20
Out Least (OUTL)	5-68	Store Not (STRN)	5-29
Out Most (OUTM)	5-68	Store Not (STRN)	5-10
Pause (PAUSE)	5-25	Store Not Bit-of-Word (STRNB)	5-11
Pop (POP)	5-65	Store Not Immediate (STRNI)	5-32
Positive Differential (PD)	5-19	Store Positive Differential (STRPD)	5-20
Print Message (PRINT)	5-190	Subroutine Return (RT)	5-178
Radian Real Conversion (RADR)	5-133	Subroutine Return Conditional (RTC)	5-178
Read from Intelligent I/O Module (RD)	5-194	Subtract (SUB)	5-89
Read from Network (RX)	5-196	Subtract Binary (SUBB)	5-101
Real to Binary Conversion (RTOB)	5-132	Subtract Binary Double (SUBBD)	5-102
Remove from Bottom (RFB)	5-153	Subtract Binary Top of Stack (SUBBS)	5-115
Remove from Table (RFT)	5-159	Subtract Double (SUBD)	5-90
Reset (RST)	5-23	Subtract Formatted (SUBF)	5-107
Reset Bit-of-Word (RST)	5-24	Subtract Real (SUBR)	5-91
Reset Immediate (RSTI)	5-36	Subtract Top of Stack (SUBS)	5-111
Reset Watch Dog Timer (RSTWT)	5-174	Sum (SUM)	5-120
Rotate Left (ROTL)	5-123	Swap (SWAP)	5-170
Rotate Right (ROTR)	5-124	Table Shift Left (TSHFL)	5-165
RSTBIT	5-144	Table Shift Right (TSHFR)	5-165
Segment (SEG)	5-137	Table to Destination (TTD)	5-150
Set (SET)	5-23	Tangent Real (TANR)	5-118
Set Bit-of-Word (SET)	5-24	Ten's Complement (BCDCPL)	5-130
Set Immediate (SETI)	5-36	Time (TIME)	5-172
SETBIT	5-144	Timer (TMR) and Timer Fast (TMRF)	5-40
Shift Left (SHFL)	5-121	Up Down Contador (UDC)	5-49
Shift Register (SR)	5-51	Write to Intelligent I/O Module (WT)	5-195
Shift Right (SHFR)	5-122	Write to Network (WX)	5-198

Usando Instrucciones booleanas

¿Ud. se ha preguntado porqué muchos fabricantes de PLC siempre citan el tiempo de barrido para un programa booleano de 1K al usar las instrucciones booleanas? Simple. La mayoría de los programas utilizan muchas instrucciones booleanas. El PLC trabaja con estas instrucciones que son simples, diseñadas para unir contactos de entradas y salidas en serie o en paralelo, en varias combinaciones. Ya que el programa *DirectSOFT* le permite usar símbolos gráficos para construir el programa, usted no tiene que saber la abreviatura o el mnemotécnico de las instrucciones. Sin embargo, pueden ser útiles cuando vea el listado mnemotécnico de un programa. Estos mnemotécnicos también se usan como una variante con el programador portátil.

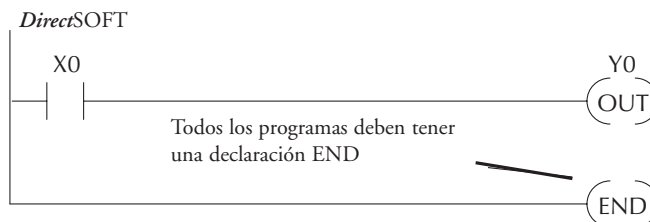
DS5	Implied
HPP	Usado

Muchas de las instrucciones en este capítulo no son instrucciones usadas in *DirectSOFT*, pero son implicadas. Esto quiere decir que no son comandos desde el teclado. Sin embargo, pueden ser vistas en Mneumonic View (nemotécnicos) del programa cuando un programa en *DirectSOFT* ha sido desarrollado y a sido aceptado (compilado). Cada instrucción listada en este capítulo tendrá una pequeña tabla como en la figura adyacente para indicar como se usa la instrucción con *DirectSOFT* y el programador HPP.

Los siguientes párrafos muestran como estas instrucciones son usadas para construir programas ladder simples.

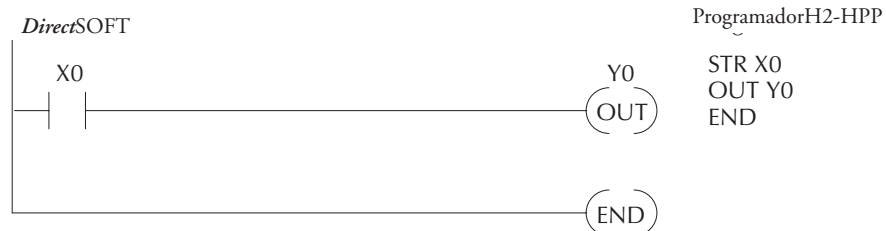
Instrucción END

Todos los programas DL06 deben tener una declaración END como instrucción final. Esto le dice a la CPU que éste es el final del programa. Normalmente, cualquiera instrucción colocada después de la instrucción END no es ejecutada. Hay excepciones a esto tal como rutinas de interrupción, etc.



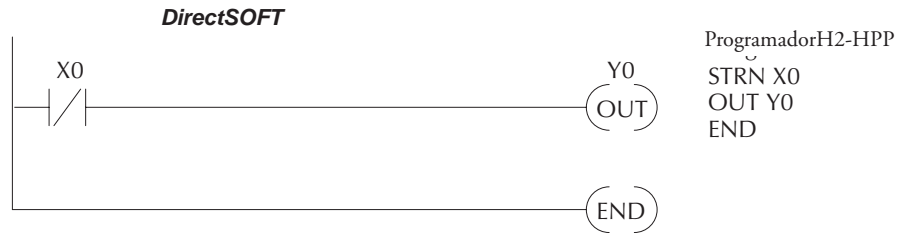
Renglones simples

Siempre se debe usar un contacto para iniciar un renglón (**rung** en inglés) que contiene contactos y bobinas (con algunas excepciones). La instrucción booleana que hace esto se llama STORE o instrucción STR y el símbolo es un contacto normalmente abierto. La salida es representada por la instrucción OUT cuyo símbolo es una bobina. El ejemplo siguiente muestra cómo entrar un solo contacto y una sola bobina de salida en un renglón.



Contactos Normalmente Cerrados

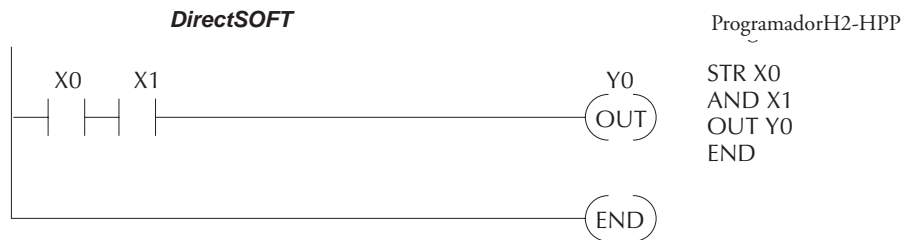
Los contactos normalmente cerrados son muy comunes. Estos se hacen con las instrucciones **Store**, **Not**, o **STRN**. El siguiente ejemplo muestra un simple renglón con un contacto normalmente cerrado.



5

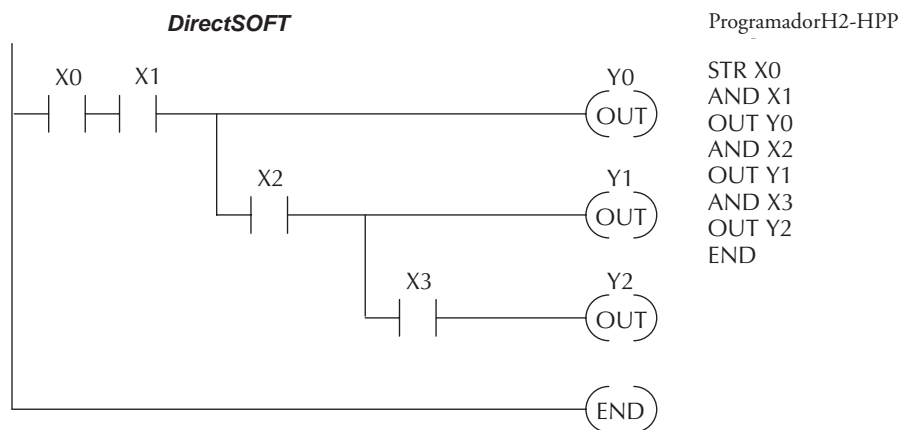
Contactos en serie

Use la instrucción **AND** para unir dos o más contactos en serie. El ejemplo siguiente muestra dos contactos en serie y una salida en una bobina. Las instrucciones usadas serían el **STR X0**, **AND X1** seguidos por **OUT Y0**.



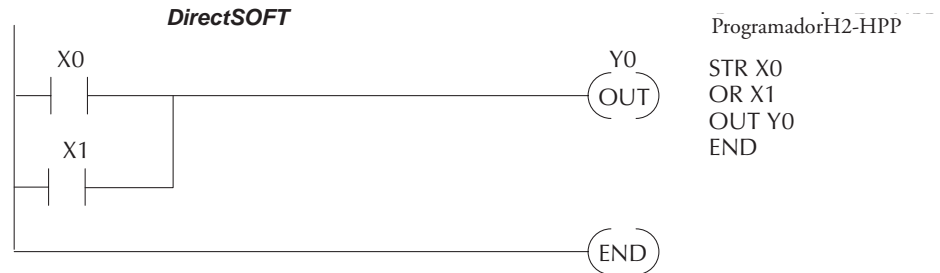
Salidas en el medio del renglón

A veces es necesario usar salidas en el medio del renglón para obtener salidas adicionales que son condiciones a otros contactos. No se deben colocar más instrucciones en un renglón después de una rama que conecta a una salida. El ejemplo siguiente muestra cómo se puede utilizar la instrucción **AND** para continuar un renglón con más salidas condicionales.



Elementos en paralelo

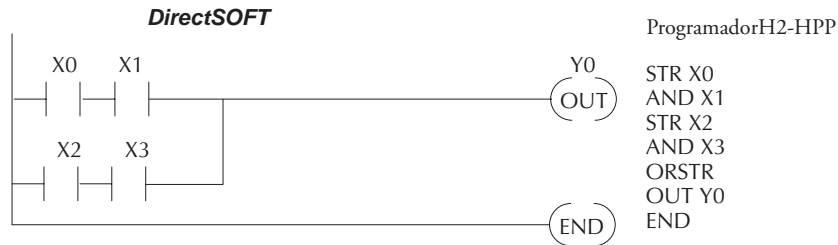
Usted puede también tener que unir contactos en paralelo. La instrucción OR permite hacer esto. El ejemplo siguiente muestra dos contactos en paralelo y una sola salida. Las instrucciones serían el STR X0, OR X1, seguidos por OUT Y0.



5

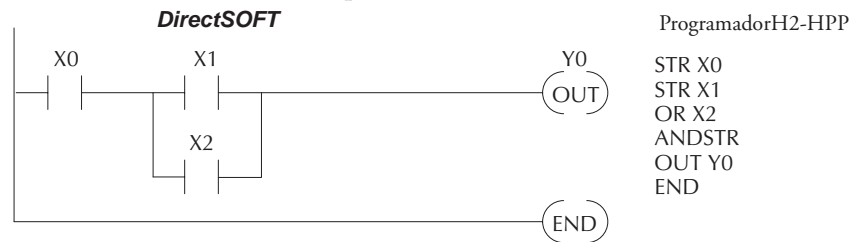
Uniendo ramas conectadas en serie y en paralelo

A menudo es necesario unir varios grupos de elementos en serie en paralelo. La instrucción OR STORE (ORSTR) permite esta operación. El ejemplo siguiente muestra un circuito con elementos en serie unidos en paralelo.



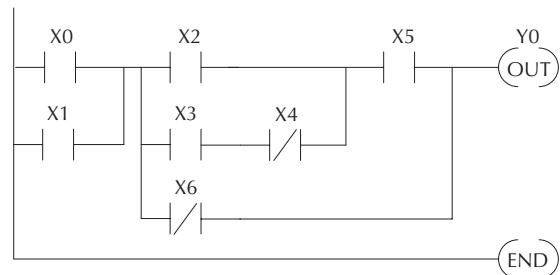
Ramas en paralelo que se unen en serie

Usted puede también unir una o más ramas paralelas en serie. La instrucción AND STORE (ANDSTR) permite esta operación. El ejemplo siguiente muestra un circuito simple con ramas de contactos en serie con contactos en paralelo.



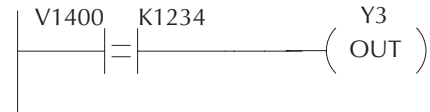
Circuitos combinación

Usted puede combinar varios tipos de ramas en serie y paralelas para solucionar la mayoría de problemas de lógica. El ejemplo siguiente muestra un circuito simple de combinación.



Comparación booleana

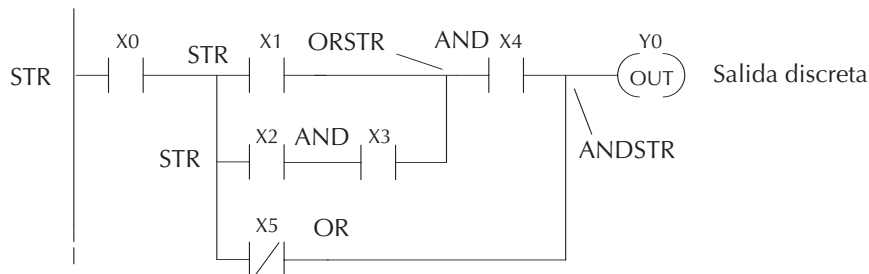
Algunos fabricantes de PLCs hacen realmente difícil el hacer una comparación simple de dos números. Algunos requieren mover los datos a varios lugares antes de que se pueda realizar realmente la comparación. Los PLCs DL06 tienen instrucciones booleanas comparativas que le permiten solucionar rápida y fácilmente este problema. La comparación booleana permite evaluación de dos valores de 4 dígitos usando contactos booleanos (los valores deben ser del mismo tipo, tal como, BCD, decimal. etc). Las evaluaciones válidas son: igual a, no igual a, igual a o mayor que, y menor que. En el ejemplo siguiente cuando el valor en la dirección de memoria V1400 es igual al valor constante 1234, Y3 se energizará.



Stack booleano

Hay límites de cuántos elementos usted puede incluir en un renglón. Esto es porque el PLC DL06 usa una memoria o stack booleano de 8 niveles para evaluar los varios elementos de lógica. El stack booleano es un área de almacenamiento temporal que soluciona la lógica en el renglón. Cada vez que el programa encuentra una instrucción STR, la instrucción se pone en el nivel superior del stack. Cualquiera otras instrucciones STR ya en el stack booleano se van hacia abajo un nivel. Las instrucciones ANDSTR y ORSTR combinan niveles del stack booleano cuando se encuentran en la lógica. Ocurrirá un error durante la compilación del programa si la CPU encuentra un renglón que use más que los 8 niveles del stack.

El ejemplo siguiente muestra cómo se usa el stack para solucionar lógica booleana.



STR X0	
1	STR X0
2	
3	
4	
5	
6	
7	
8	

STR X1	
1	STR X1
2	STR X0
3	
4	
5	
6	
7	
8	

STR X2	
1	STR X2
2	STR X1
3	STR X0
4	
5	
6	
7	
8	

AND X3	
1	X2 AND X3
2	STR X1
3	STR X0
4	
5	
6	
7	
8	

ORSTR	
1	X1 o (X2 AND X3)
2	STR X0
3	

AND X4	
1	X4 AND {X1 o (X2 AND X3)}
2	STR X0
3	

ORNOT X5	
1	NOT X5 OR X4 AND {X1 OR (X2 AND X3)}
2	STR X0
3	

4	
5	
6	
7	
8	

4	
5	
6	
7	
8	

4	
5	
6	
7	
8	

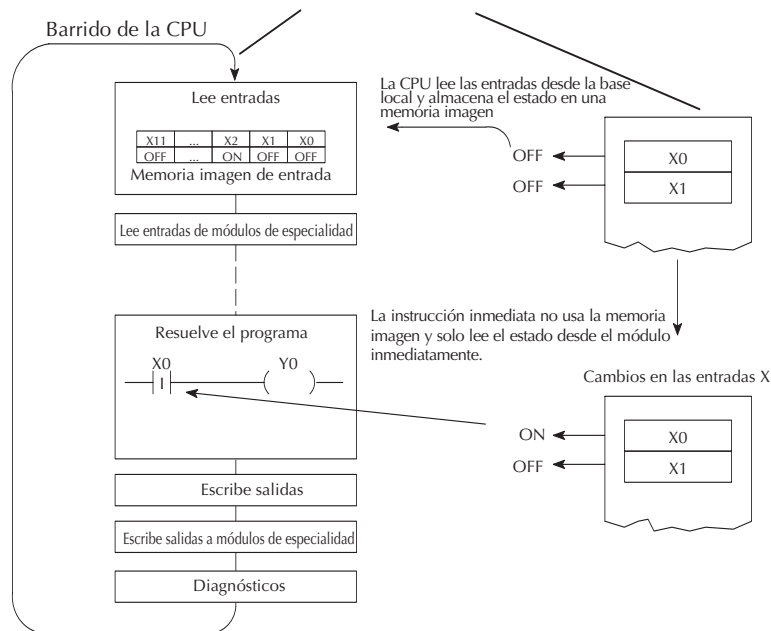
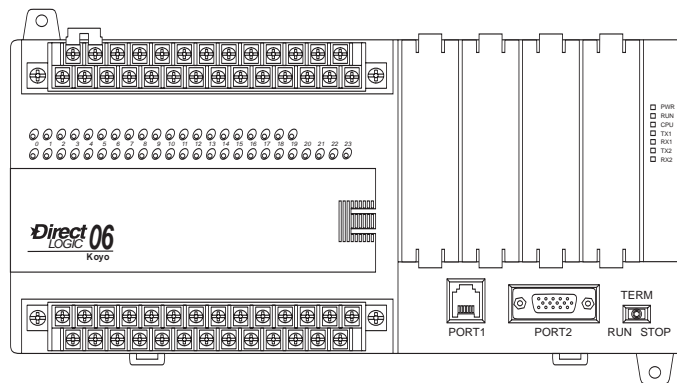
ANDSTR

Instrucciones booleanas inmediatas

El PLC DL06 puede terminar generalmente un ciclo de operación en una cuestión de milisegundos. Sin embargo, en algunos casos no se puede esperar algunos milisegundos hasta que ocurra la actualización siguiente de E/S. EL PLC DL06 tiene funciones de entradas y salidas inmediatas, que son instrucciones booleanas especiales que permiten leer directamente a las entradas y escribir directamente a las salidas durante la porción de la ejecución del programa del ciclo de la CPU. Recuerde que esto se hace normalmente durante la porción de la actualización de las entradas o de las salidas del ciclo de la CPU. Las instrucciones inmediatas demoran más para ejecutarse porque se interrumpe la ejecución del programa mientras la CPU lee o escribe las E/S. Esta función normalmente no se hace hasta que las entradas sean leídas o las salidas sean escritas en la porción del ciclo de la CPU.



NOTA: Aunque la instrucción inmediata de entrada lee el estado más corriente del punto de entrada, solamente usa los resultados para solucionar esa instrucción. No usa el nuevo estado para actualizar la memoria imagen. Por lo tanto, cualquier instrucción regular que siga usará los valores de la memoria imagen. Cualquier instrucción inmediata que siga accederá a las E/S otra vez para actualizar el estado. La instrucción

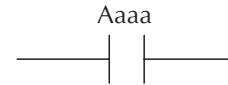


Instrucciones booleanas

La instrucción Store(STR)

DS5	Usado
HPP	Usado

Comienza un nuevo renglón o una rama adicional en un renglón con un contacto normalmente abierto. El estado del contacto será el mismo estado como el punto de la memoria imagen asociada o localización de memoria.



La instrucción Store Not (STRN)

DS5	Usado
HPP	Usado

Comienza un nuevo renglón o una rama adicional en un renglón con un contacto normalmente cerrado. El estado del contacto será opuesto al estado como la memoria imagen asociada o localización de memoria.

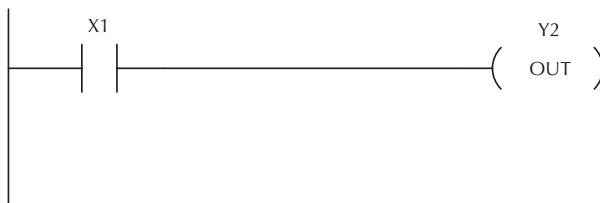


5

Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Temporizador T	0-377
Contador C CT	0-177
Relevadores especialesl SP	0-777

En el ejemplo siguiente, cuándo la entrada X1 está ON, se activará la salida Y2.

DirectSOFT

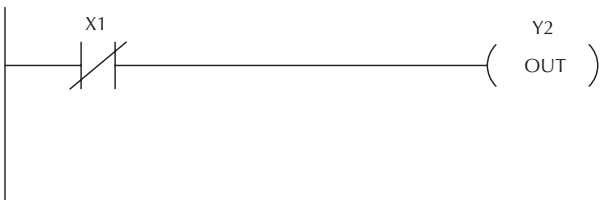


Programador D2-HPP

\$ STR	→	B 1	ENT
GX OUT	→	C 2	ENT

En el siguiente ejemplo, cuándo la entrada X1 está OFF, se activará la salida Y2.

DirectSOFT



Programador D2-HPP

SP STRN	→	B 1	ENT
GX OUT	→	C 2	ENT

Instrucción Store Bit-of-Word (STRB)

DS5	Usado	La instrucción STRB comienza un nuevo renglón o una rama adicional en un renglón con un contacto normalmente abierto. El estado del contacto será el mismo estado como el bit referenciado en la localización asociada de la memoria.
HPP	Usado	



Instrucción Store Not Bit-of-Word (STRNB)

DS5	Usado	Comienza un nuevo renglón o una rama adicional en un renglón con un contacto normalmente cerrado. El estado del contacto será opuesto al estado del bit referenciado en la localización asociada de la memoria.
HPP	Usado	



	Tipo de operando de datos	Rango del DL06	
		aaa	bb
.....	A		
Memoria V	B	Vea el mapa de memoria	0 a 15
Puntero	PB	Vea el mapa de memoria	0 a 15

5

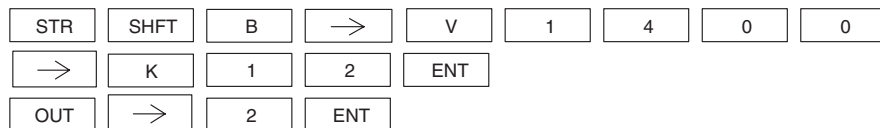
En el ejemplo siguiente de la instrucción STRB, cuando el bit 12 de la memoria V1400 está ON, la salida Y2 se activará. Note que en *DirectSOFT* se usa “B”1400.12.

En el ejemplo de STRNB, cuando el bit 12 de la memoria V1400 está apagado, se activará la salida Y2.

DirectSOFT



Programador D2-HPP

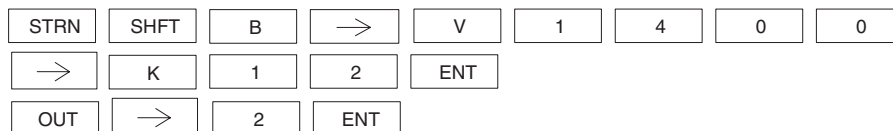


En el ejemplo de STRNB, cuando el bit 12 de la memoria V1400 está apagado, se activará la salida Y2.

DirectSOFT



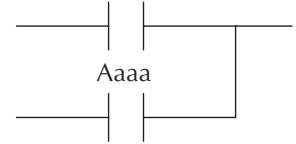
Programador D2-HPP



La instrucción OR lógica (OR)

DS5	Implied
HPP	Usado

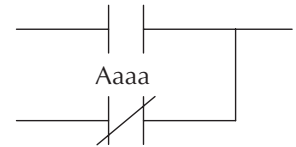
La instrucción OR hace un OR lógico con un contacto normalmente abierto en paralelo con otro contacto en un renglón. El estado del contacto será el mismo estado como el punto asociado de memoria imagen. Vea el ejemplo abajo para entender el significado .



La instrucción ORN lógica (ORN)

DS5	Implied
HPP	Usado

La instrucción lógica ORN hace un OR lógico con un contacto normalmente cerrado en paralelo con otro contacto en un renglón. El estado del contacto será opuesto al estado del punto asociado de memoria imagen.

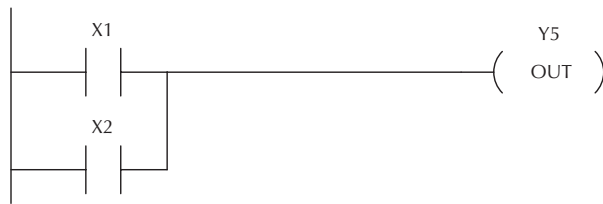


5

Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Temporizador T	0-377
Contador CT	0-177
Relevadores especiales! SP	0-777

En el siguiente ejemplo, cuándo la entrada X1 o X2 está ON, se activará la salida Y5.

DirectSOFT



Programador D2-HPP

\$ STR	→	B 1	ENT
Q OR	→	C 2	ENT
GX OUT	→	F 5	ENT

En el siguiente ejemplo, cuándo la entrada X1 está ON o X2 está apagada, se activará la salida Y5.

DirectSOFT



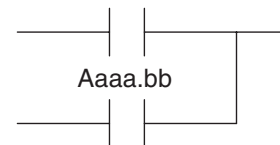
Programador D2-HPP

\$ STR	→	B 1	ENT
R ORN	→	C 2	ENT
GX OUT	→	F 5	ENT

La instrucción Or Bit-of-Word (OR)

DS5	Implied
HPP	Usado

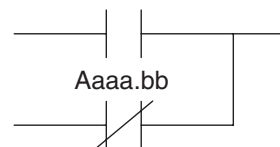
La instrucción OR hace un OR lógico de un contacto normalmente abierto en paralelo a otro contacto en un renglón. El estado del contacto será el mismo estado que el bit referido en la dirección de memoria asociada.



La instrucción Or Not Bit-of-Word (ORN)

DS5	Implied
HPP	Usado

La instrucción ORN hace un OR lógico de un contacto normalmente cerrado en paralelo a otro contacto en un renglón. El estado del contacto será el estado opuesto al bit referido en la dirección de memoria asociada.

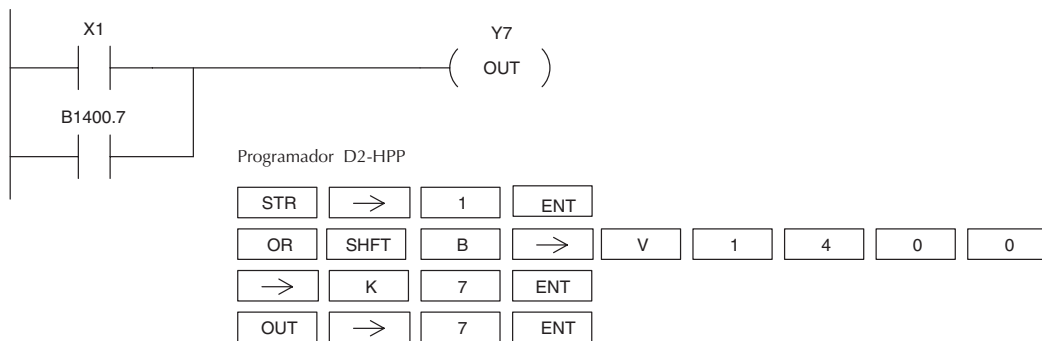


5

Tipo de operando de datos	A	Rango del DL06	
		aaa	bb
Memoria	B	Vea el mapa de memoria	0 a 15
Puntero	PB	Vea el mapa de memoria	0 a 15

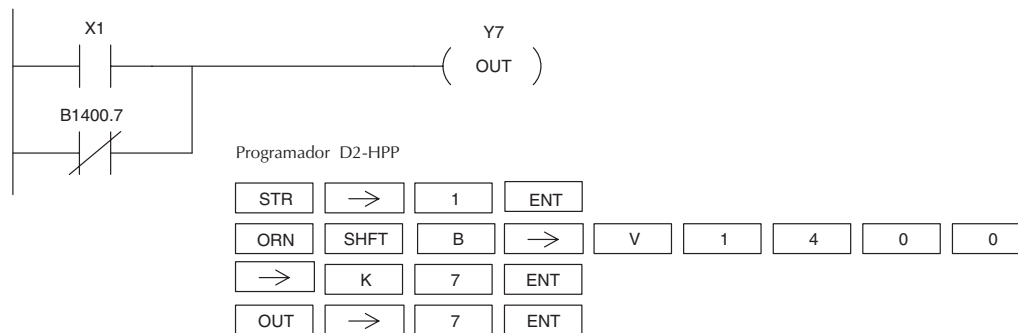
En el ejemplo siguiente de la instrucción OR, cuando la entrada X1 o el bit 7 de V1400 está activado, se energizará la salida Y5. Note que en *DirectSOFT* se usa “B”1400.7.

DirectSOFT



En el ejemplo siguiente de la instrucción OR, cuando la entrada X1 está activada o el bit 7 de V1400 no está activado, se energizará la salida Y5.

DirectSOFT



La instrucción AND lógica (AND)

DS5	Implied
HPP	Usado

La instrucción AND lógica hace la función AND lógica en un contacto normalmente abierto en serie con otro contacto en un renglón. El estado del contacto será el mismo estado que el de la entrada física asociada de memoria imagen.



La instrucción ANDN lógica (ANDN)

DS5	Implied
HPP	Usado

La instrucción ANDN lógica hace la función AND lógica en un contacto normalmente cerrado en serie con otro contacto en un renglón. El estado del contacto será opuesto al estado de la entrada física asociada de memoria imagen.

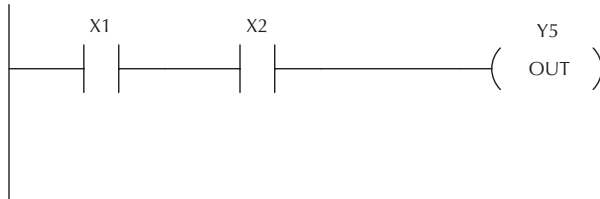


5

Tipo de operando de datos		Rango del DL06
..... A		aaa
Entradas	X	0-777
Salidas	Y	0-777
Relevadores de control	C	0-1777
Etapas	S	0-1777
Temporizador	T	0-377
Contador	CT	0-177
Relevadores especiales	SP	0-777

En el siguiente ejemplo de AND, cuándo las entradas X1 y X2 están ON, se activará la salida Y5.

DirectSOFT

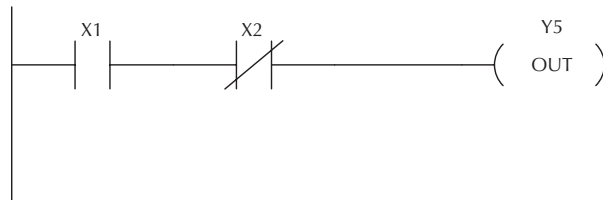


Programador D2-HPP

\$ STR	→	B 1	ENT
V AND	→	C 2	ENT
GX OUT	→	F 5	ENT

En el siguiente ejemplo de ANDN, cuándo la entrada X1 está ON y X2 está apagada, se activará la salida Y5.

DirectSOFT



Programador D2-HPP

\$ STR	→	B 1	ENT
W ANDN	→	C 2	ENT
GX OUT	→	F 5	ENT

La instrucción AND Bit-of-Word (AND).

DS5	Implied
HPP	Usado

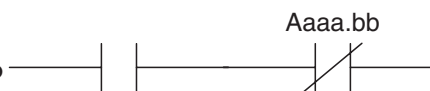
(Bit of Word significa bit de palabra) La instrucción AND hace un AND lógico de un contacto normalmente abierto en serie con otro contacto en un renglón. El estado del contacto será el mismo estado que el bit referido en la dirección de memoria asociada.



La instrucción And Not Bit-of-Word (ANDN)

DS5	Implied
HPP	Usado

La instrucción ANDN hace un AND lógico de un contacto normalmente cerrado en serie con otro contacto en un renglón. El estado del contacto será opuesto del estado del bit referido en la dirección de memoria asociada.

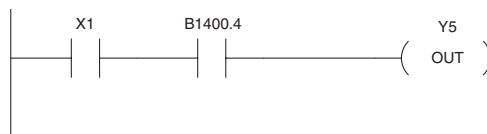


Tipo de operando de datos		Rango del DL06	
.....	A	aaa	bb
Memoria	B	Vea el mapa de memoria	0 a 15
Puntero	PB	Vea el mapa de memoria	0 a 15

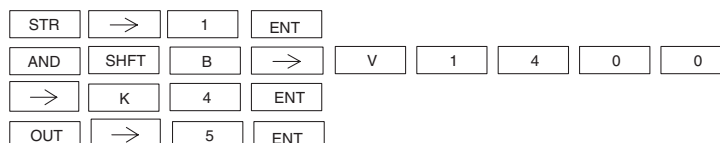
5

En el ejemplo siguiente de Bit of Word, cuando la entrada X1 y el bit 4 de V1400 están ON, se energizará la salida Y5. Note que en *DirectSOFT* se usa “B”1400.4.

DirectSOFT

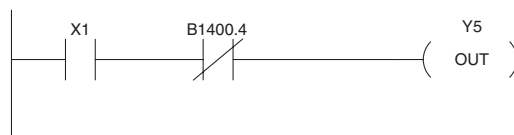


Programador D2-HPP

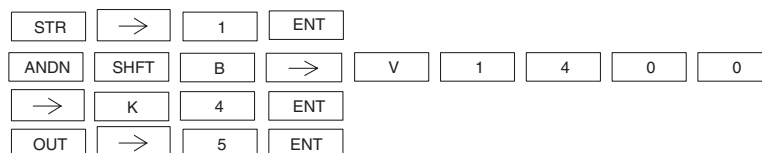


En el siguiente ejemplo de And Not Bit-of-Word, cuando la entrada X1 está ON y el bit 4 de V1400 está OFF, se energizará la salida Y5.

DirectSOFT



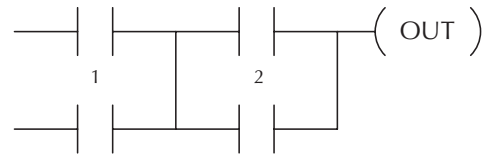
Programador D2-HPP



La instrucción AND Store (AND STR)

DS5	Implied
HPP	Usado

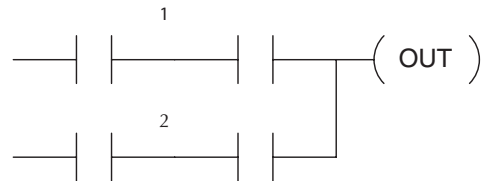
La instrucción ANDSTR hace una función AND lógica con dos ramas de un renglón en serie. Ambas ramas deben comenzar con la instrucción STR.



La instrucción OR Store (OR STR)

DS5	Implied
HPP	Usado

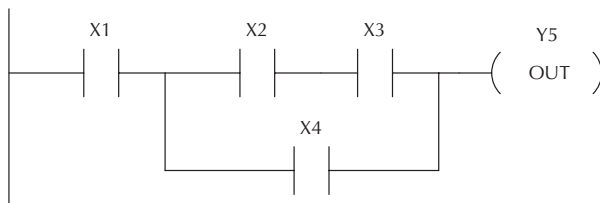
La instrucción ORSTR hace una función OR lógica con dos ramas de un renglón en paralelo. Ambas ramas deben comenzar con la instrucción STR.



5

En el siguiente ejemplo, la rama compuesta de los contactos X2, X3, y X4 se ha operado AND con la rama compuesta del contacto X1.

DirectSOFT

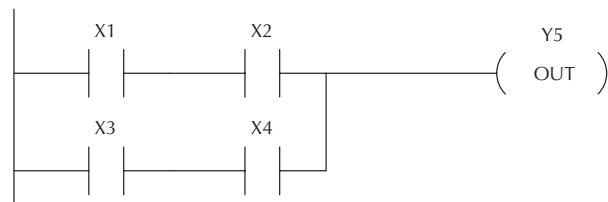


Programador D2-HPP

\$ STR	→	B 1	ENT
\$ STR	→	C 2	ENT
V AND	→	D 3	ENT
Q OR	→	E 4	ENT
L ANDST	ENT		
GX OUT	→	F 5	ENT

En el siguiente ejemplo OR , la rama compuesta de los contactos X1 y X2 se han operado OR con la rama compuesta de los contactos X3 y X4.

DirectSOFT



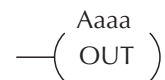
Programador D2-HPP

\$ STR	→	B 1	ENT
V AND	→	C 2	ENT
\$ STR	→	D 3	ENT
V AND	→	E 4	ENT
M ORST	ENT		
GX OUT	→	F 5	ENT

La instrucción OUT (OUT)

DS5	Usado
HPP	Usado

La instrucción OUT contiene el estado del renglón (ON/OFF) y deja salir el estado discreto (ON/OFF) al punto especificado de la memoria imagen.

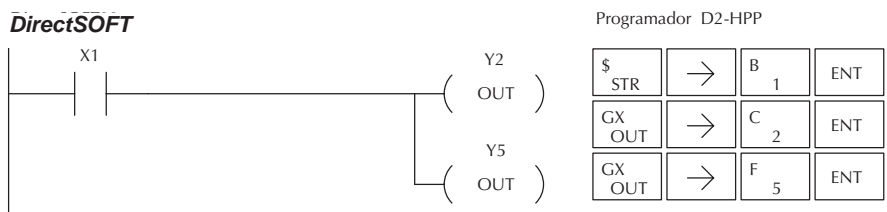


No debe usarse más de una instrucción OUT que referencie la misma localización discreta ya que sólo la última instrucción OUT en el programa controlará el punto físico de salida. En vez de eso, use la instrucción OROUT.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777

5

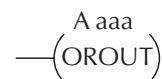
En el siguiente ejemplo Out, cuándo la entrada X1 está ON, se activarán las salidas Y2 y Y5.



La instrucción Or Out (OROUT)

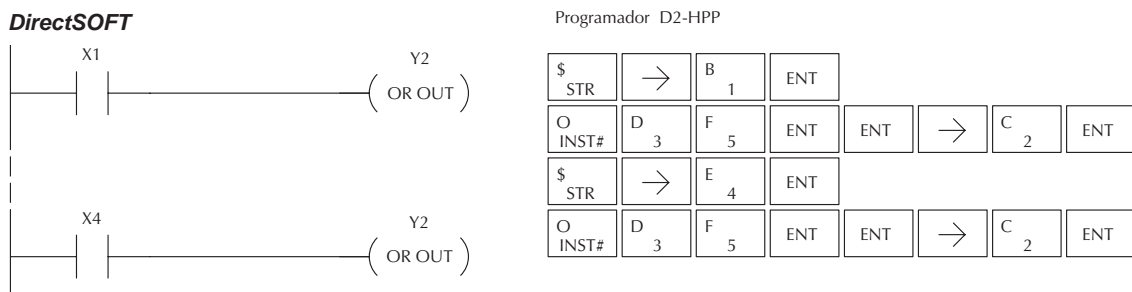
DS5	Usied
HPP	Usado

La instrucción OROUT permite que más de un renglón de lógica discreta controle una sola salida. Pueden ser usadas múltiples instrucciones OROUT que referencian la misma bobina de salida, ya que todos los contactos que controlan la salida son operados con la función OR. Si el estado de cualquier renglón está ON, la salida estará también ON.



Tipo de operando de datos	Rango del DL06
A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777

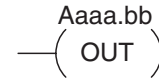
En el ejemplo siguiente, cuando una o las dos salidas X1 o X4 están ON, en cualquier parte del programa, se energizará la salida Y2 .



La instrucción Out Bit-of-Word (OUT)

DS5	Usado
HPP	Usado

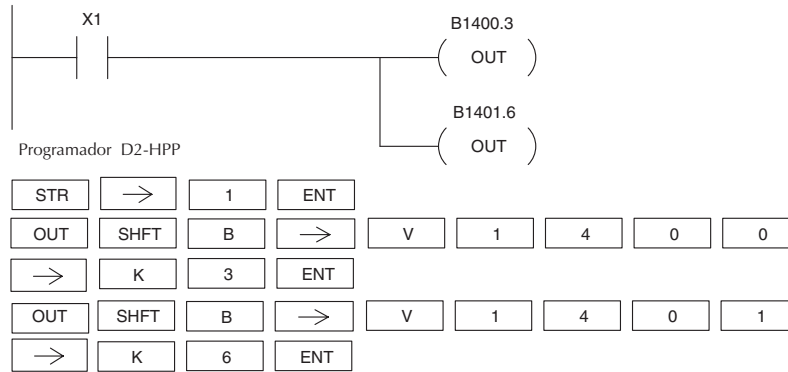
La instrucción OUT tiene el estado del renglón (ON/OFF) y produce el estado discreto (ON/OFF) del bit especificado en la dirección de memoria referida. Generalmente no deben ser usadas múltiples instrucciones OUT que se refieren al mismo bit de la misma palabra puesto que solamente la última instrucción en el programa controlará el estado del bit.



Tipo de operando de datos		Rango del DL06	
		aaa	bb
.....	A		
Memoria	B	Vea el mapa de memoria	0 a 15
Puntero	PB	Vea el mapa de memoria	0 a 15

NOTA: Si la palabra Bit-of-Word se entra como V1400.3 en DirectSOFT, ser'a convertida a B1400.3. :a

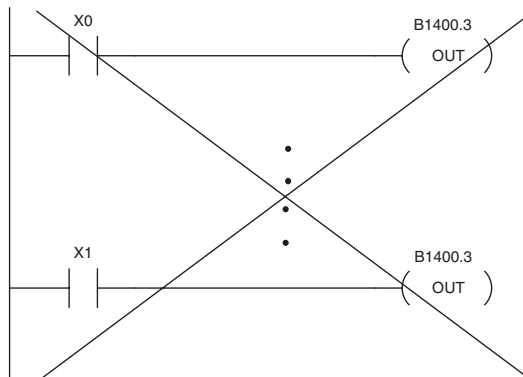
DirectSOFT



palabra Bit-of-Word puede ser también ingresada como B1400.3.

En el ejemplo siguiente de la instrucción OUT, cuando la entrada X1 está encendida, el bit 3 de V1400 y el bit 6 de V1401 se activarán.

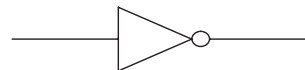
El ejemplo siguiente de Out Bit-of-Word contiene dos instrucciones Out Bit-of-Word usando el mismo bit en la misma palabra de memoria. El estado final del bit 3 de V1400 es controlado en última instancia por el último renglón de lógica en el que es referido, es decir, X1 va a forzar el estado lógico controlado por X0. *Para evitar esta situación, no deben ser usadas múltiples instrucciones Out Bit-of-Word que usan la misma dirección en la programación.*



La instrucción Not (NOT)

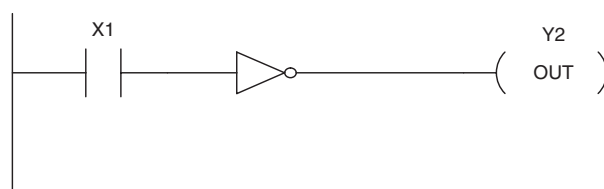
DS5	Usado
HPP	Usado

La instrucción NOT invierte el estado del renglón en el punto de la instrucción.

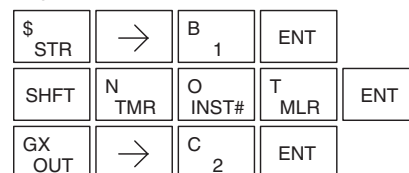


En el ejemplo siguiente cuando X1 está apagado, Y2 se activará. Esto es porque la instrucción NOT invierte el estado del renglón.

DirectSOFT



Programador D2-HPP

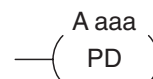


NOTE: DirectSOFT Release 1.1i and later supports the use of the NOT instrucción. The above example renglón is merely intended to show the visual representation of the NOT instrucción. The NOT instrucción can only be selected in DirectSOFT from the Instrucción Browser. The renglón cannot be created or displayed in DirectSOFT versions earlier than 1.1i.

La instrucción Positive Differential (PD)

DS5	Usado
HPP	Usado

La instrucción PD se conoce típicamente como "one shot". Cuando la lógica de entrada produce una transición de OFF para ON, la salida se



Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777

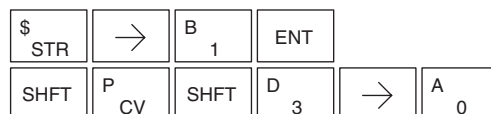
activará por un barrido de la CPU.

En el ejemplo siguiente, cada vez que X1 hace una transición de OFF para ON, C0 se activará

DirectSOFT



Programador D2-HPP



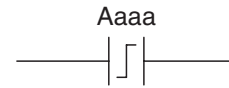
por un barrido.

La instrucción Store Positive Differential (STRPD)

DS5	Usado
HPP	Usado

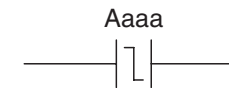
La instrucción STRPD comienza un nuevo renglón o una rama adicional en un renglón con un contacto. El contacto se cierra en un barrido de la CPU cuando el estado del punto asociado de memoria imagen hace una transición de OFF para ON.

Después, el contacto permanece abierto hasta que haya otra transición de OFF para ON (el símbolo dentro del contacto representa la transición). Esta función se llama a veces "one shot".



La instrucción Store Negative Differential (STRND)

La instrucción STRND comienza un nuevo renglón o una rama adicional en un renglón con un contacto. El contacto se cierra en un barrido de la CPU cuando el estado del punto asociado de memoria imagen hace una transición de ON para OFF. Luego el contacto permanece abierto hasta que haya otra transición de ON para OFF (el símbolo dentro del contacto representa la transición).



5

DS5	Usado
HPP	Usado



NOTE: When using DirectSOFT, these instrucciones can only be entered from the Instrucción Browser.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Temporizador T	0-377
Contador CT	0-177

En el ejemplo siguiente, cada vez que X1 hace la transición de OFF para ON, Y4 se activará por un barrido.

DirectSOFT

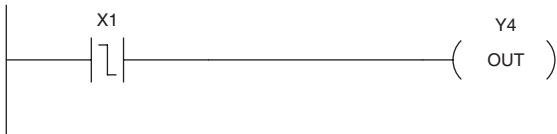


Programador D2-HPP



En el ejemplo siguiente, cada vez que X1 hace la transición de ON para OFF, la salida Y4 se activará por un barrido.

DirectSOFT



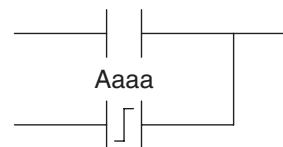
Programador D2-HPP



La instrucción Or Positive Differential (ORPD)

DS5	Implied
HPP	Usado

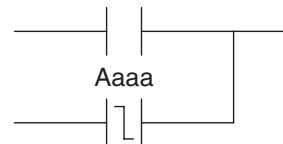
La instrucción ORPD hace un OR lógico de un contacto en paralelo a otro contacto en un renglón. El estado del contacto estará abierto hasta que el punto asociado de la memoria imagen hace una transición de OFF para ON, cerrándose en un barrido de la CPU. Después de eso, sigue abierto hasta otra transición.



La instrucción Or Negative Differential (ORND)

DS5	Implied
HPP	Usado

La instrucción ORND hace un OR lógico de un contacto en paralelo a otro contacto en un renglón. El estado del contacto estará abierto hasta que el punto asociado de la memoria imagen hace una transición de ON para OFF, cerrándose en un barrido de la CPU. Después de eso, sigue abierto hasta otra transición.

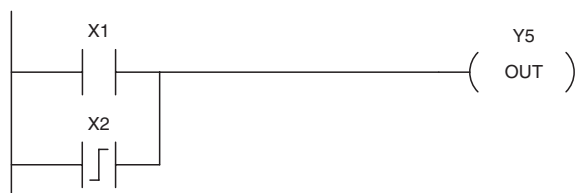


5

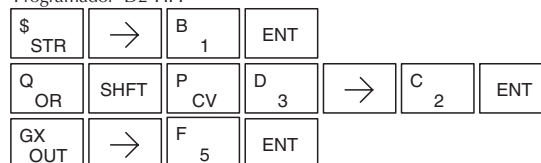
Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Temporizador T	0-377
Contador CT	0-177

En el ejemplo siguiente, se activará la salida Y5 cuando X1 está ON o por un barrido de la CPU cuando haya una transición en X2 desde OFF a ON

DirectSOFT



Programador D2-HPP

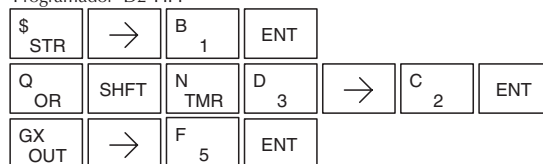


En el ejemplo siguiente, se activará la salida Y5 cuando X1 está ON o también por un barrido de la CPU cuando haya una transición en X2 desde ON a OFF.

DirectSOFT



Programador D2-HPP



La instrucción And Positive Differential (ANDPD)

DS5	Implied
HPP	Usado

La instrucción ANDPD hace la función AND lógica entre un contacto normalmente abierto en serie con otro contacto en un renglón. El estado del contacto estará abierto hasta que el punto asociado de la memoria imagen haga una transición de OFF para ON, cerrándolo por un barido de la CPU. Después de eso, sigue abierto hasta otra transición de OFF para ON.



La instrucción And Negative Differential (ANDND)

DS5	Implied
HPP	Usado

La instrucción ANDND hace la función AND lógica entre un contacto normalmente abierto en serie con otro contacto en un renglón. El estado del contacto estará abierto hasta que el punto asociado de la memoria imagen haga una transición de ON para OFF, cerrándolo por un barrido de la CPU. Después de eso, sigue abierto hasta otra transición de ON para OFF.

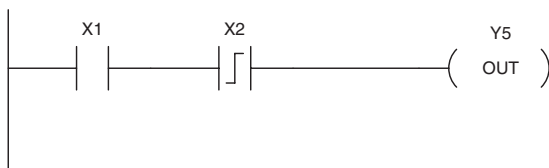


5

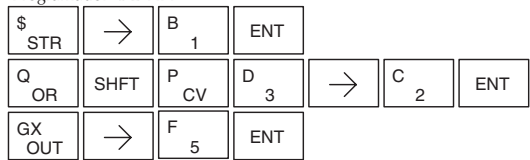
Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Temporizador T	0-377
Contador CT	0-177

En el ejemplo siguiente, se activará Y5 cuando X1 está ON y al mismo tiempo en un barrido de la CPU cuando haya una transición en X2 desde OFF para ON.

DirectSOFT

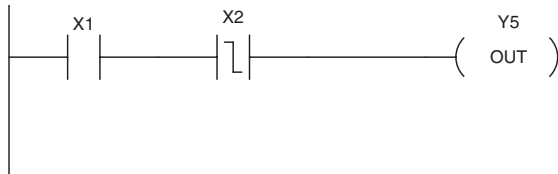


Programador D2-HPP

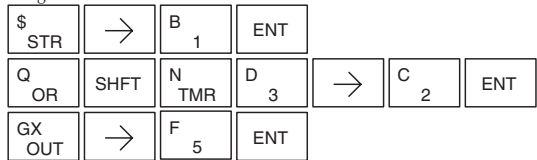


En el ejemplo siguiente, se activará Y5 cuando X1 está ON y al mismo tiempo en un barrido de la CPU cuando haya una transición en X2 desde ON a OFF.

DirectSOFT



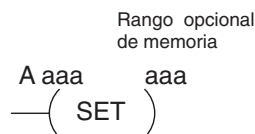
Programador D2-HPP



La instrucción Set (SET)

DS5	Usado
HPP	Usado

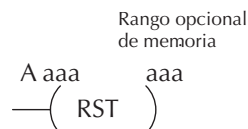
La instrucción SET coloca ON o prende un punto de memoria imagen o un rango consecutivo de memorias imagen. Una vez que la memoria se hace ON permanecerá así hasta que sea vuelta a OFF por la instrucción RESET. No es necesario que el renglón que controle la instrucción SET permanezca ON.



La instrucción Reset (RST)

DS5	Usado
HPP	Usado

Esta instrucción vuelve a 0, a OFF o apaga un punto de memoria imagen o un rango consecutivo de memorias imagen. Una vez que la localización de memoria es OFF no es necesario que el renglón permanezca ON.



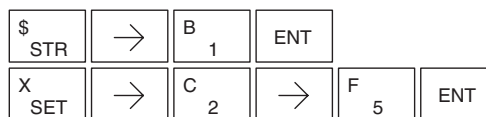
Tipo de operando de datos	Rango del DL06
..... A	aaa
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Temporizador T	0-377
Contador CT	0-177

En el ejemplo siguiente cuando X1 está ON, Y2 hasta Y5 se activarán o se harán ON y permanecerán energizadas.

DirectSOFT

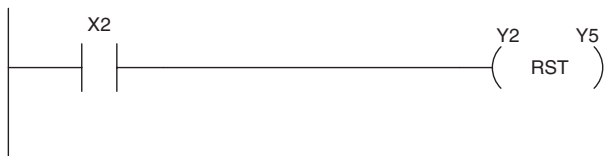


Programador D2-HPP

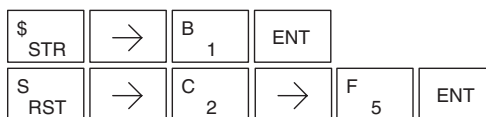


En el ejemplo siguiente cuando X1 está ON, las salidas Y2 hasta Y5 será vueltas a OFF y permanecerán desenergizadas.

DirectSOFT



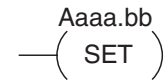
Programador D2-HPP



La instrucción Set Bit-of-Word (SET)

DS5	Usado
HPP	Usado

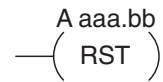
La instrucción SET activa un bit en una dirección de memoria V. Una vez que el bit se haga ON seguirá ON hasta que se repone a OFF usando la instrucción RST. No es necesario que el renglón que controla la instrucción SET permanezca activado.



La instrucción Reset Bit-of-Word (RST)

DS5	Usado
HPP	Usado

La instrucción RST repone a OFF un bit en una dirección de memoria V. Una vez que el bit se haga OFF no es necesario que el renglón que controla la instrucción RST permanezca activado.



5

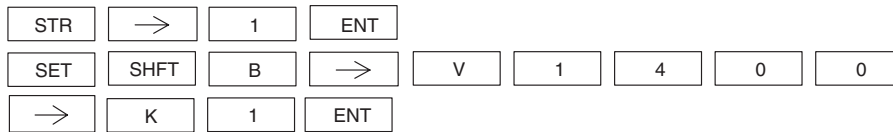
Tipo de operando de datos	Rango del DL06	
	aaa	bb
..... A		
Memoria V B	Vea el mapa de memoria	0 a 15
Puntero PB	Vea el mapa de memoria	0 a 15

En el ejemplo siguiente cuando X1 se activa ON, el bit 1 en V1400 se va al estado ON.

DirectSOFT



Programador D2-HPP

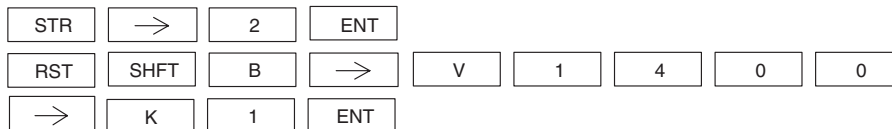


En el ejemplo siguiente cuando X2 se activa ON, el bit 1 en V1400 se va al estado OFF.

DirectSOFT



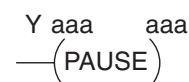
Programador D2-HPP



La instrucción Pause (PAUSE)

DS5	Usado
HPP	Usado

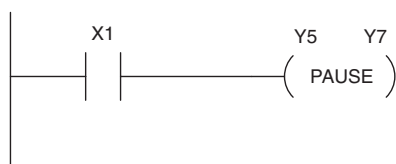
La instrucción Pause incapacita la actualización de salidas en un rango de salidas. El programa ladder continuará funcionando y actualizando la memoria imagen. Sin embargo, las salidas en el rango especificado en la instrucción Pause serán apagadas en los puntos de salidas (Colocadas OFF).



Tipo de operando de datos	Rango del DL06
..... A	aaa
Salidas Y	0-777

En el ejemplo siguiente, cuándo X1 está ON, se apagarán las salidas Y5 hasta Y7. La ejecución del programa ladder no se afectará.

DirectSOFT



Ya que el programador D2-HPP no tiene una tecla específica de Pause, usted puede utilizar el número correspondiente de la instrucción para la entrada (# 960), o puede teclear cada letra del comando.

Programador D2-HPP



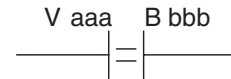
En algunos casos, usted puede querer que ciertos puntos de salida en el rango especificado en la instrucción Pause funcionen normalmente. En ese caso, use AUX 58 para cancelar la instrucción Pause.

Instrucciones de comparación booleanas

La instrucción Store If Equal (STRE)

DS5	Implied
HPP	Usado

La instrucción STRE comienza una rama nueva o adicional en un renglón con un contacto de comparación normalmente abierto. El contacto estará ON cuándo el valor contenido en Vaaa es igual al valor contenido en Bbbb.



La instrucción Store If Not Equal (STRNE)

DS5	Implied
HPP	Usado

La instrucción STRNE comienza una rama nueva o adicional en un renglón con un contacto de comparación normalmente abierto. El contacto estará ON cuándo el valor de Vaaa no es igual a Bbbb.



5

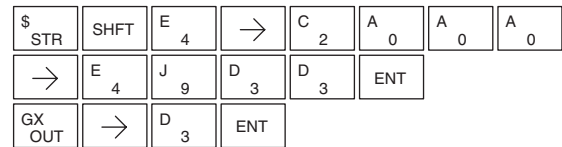
Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... B		
Memoria V..... V	Vea el mapa de memoria	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria	Vea el mapa de memoria
Constante..... K	—	0-9999

En el ejemplo siguiente, cuando el valor BCD en la memoria V2000 es igual a 4933, se activará la salida Y3.

DirectSOFT



Programador D2-HPP

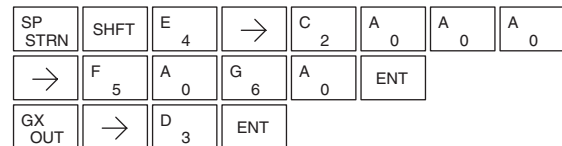


En el ejemplo siguiente, cuando el valor BCD en la memoria V2000 no sea igual a 5060, se activará la salida Y3.

DirectSOFT



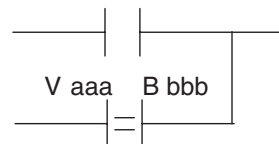
Programador D2-HPP



La instrucción Or If Equal (ORE)

DS5	Implied
HPP	Usado

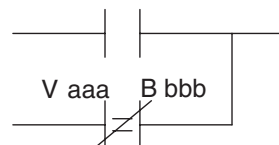
La instrucción ORE conecta un contacto comparativo normalmente abierto en paralelo con otro contacto. El contacto estará encendido cuando $V_{aaa} = B_{bbb}$.



La instrucción Or If Not Equal (ORNE)

DS5	Implied
HPP	Usado

La instrucción ORNE conecta un contacto comparativo normalmente cerrado en paralelo con otro contacto. El contacto estará encendido cuando V_{aaa} no es igual a B_{bbb} .

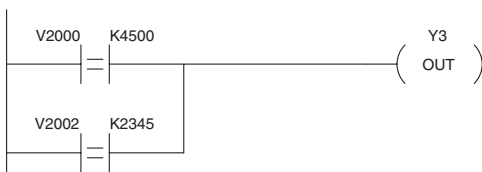


Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... B		
Memoria V	V	V
Puntero	P	P
Constante	K	0-9999

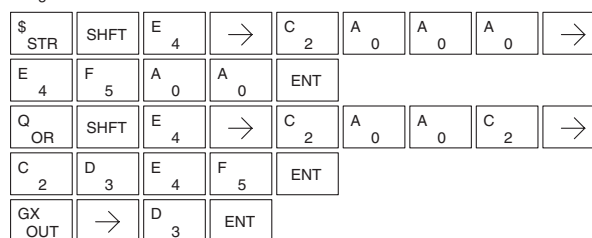
5

En el ejemplo siguiente, cuando el valor en la dirección de memoria V2000 es igual a 4500 o V2002 es igual a 2500, se energizará la salida Y3.

DirectSOFT

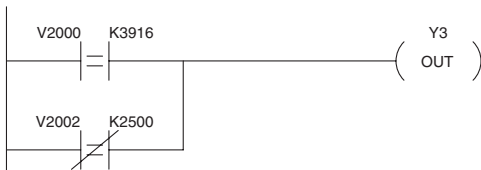


Programador D2-HPP

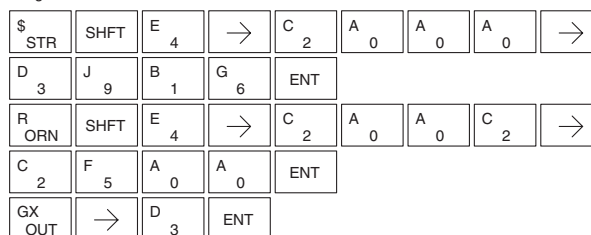


En el ejemplo siguiente, cuando el valor en la dirección de memoria V2000 es igual a 3916 o V2002 es diferente a 2500, se energizará la salida Y3.

DirectSOFT



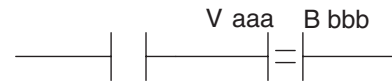
Programador D2-HPP



La instrucción And If Equal (ANDE)

DS5	Implied
HPP	Usado

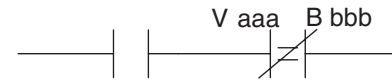
La instrucción ANDE conecta un contacto comparativo normalmente abierto en serie con otro contacto. El contacto estará encendido cuando Vaaa = Bbbb.



La instrucción And If Not Equal (ANDNE)

DS5	Implied
HPP	Usado

La instrucción ANDNE conecta un contacto comparativo normalmente cerrado en serie con otro contacto. El contacto estará encendido cuando Vaaa no es igual a Bbbb.

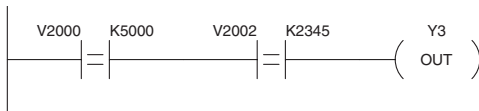


5

Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... B		
Memoria V..... V	Vea el mapa de memoria	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria	Vea el mapa de memoria
Constante..... K	—	0-9999

En el ejemplo siguiente, cuando el valor BCD en la dirección de memoria V2000 es igual a 5000 o V2002 es igual a 2345, se energizará la salida Y3.

DirectSOFT

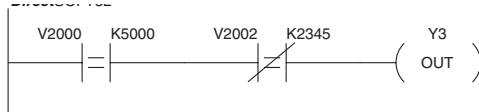


Programador D2-HPP

\$	SHFT	E 4	→	C 2	A 0	A 0	A 0	→
F 5	A 0	A 0	A 0	ENT				
V	SHFT	E 4	→	C 2	A 0	A 0	C 2	→
C 2	D 3	E 4	F 5	ENT				
GX	→	D 3	ENT					

En el ejemplo siguiente, cuando el valor BCD en la dirección de memoria V2000 es igual a 5000 o V2002 es diferente a 2345, se energizará la salida Y3.

DirectSOFT



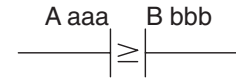
Programador D2-HPP

\$	SHFT	E 4	→	C 2	A 0	A 0	A 0	→
F 5	A 0	A 0	A 0	ENT				
V	SHFT	E 4	→	C 2	A 0	A 0	C 2	→
C 2	D 3	E 4	F 5	ENT				
GX	→	D 3	ENT					

La instrucción Comparative Store (STR)

DS5	Implied
HPP	Usado

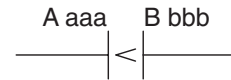
La instrucción de comparación STR comienza una rama nueva o adicional en un renglón con un contacto de comparación normalmente abierto. El contacto estará ON cuándo aaa es igual a o mayor que Bbbb.



La instrucción Store Not (STRN)

DS5	Implied
HPP	Usado

La instrucción de comparación STRN comienza una rama nueva o adicional en un renglón con un contacto de comparación normalmente cerrado. El contacto estará ON cuándo aaa sea menor que Bbbb.



Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... A/B		
Memoria V	V	Vea el mapa de memoria
Puntero	p	Vea el mapa de memoria
Constante	K	0-9999
Timer	TA	0-377
Contador	CTA	0-177

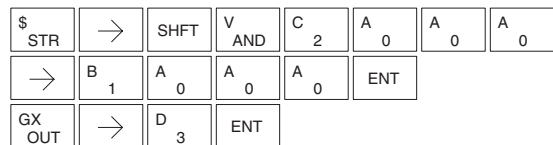
5

En el ejemplo siguiente, cuando el valor en la dirección de memoria V2000 es mayor o igual a 1000, se energizará la salida Y3..

DirectSOFT



Programador D2-HPP

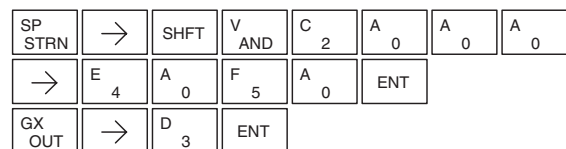


En el ejemplo siguiente, cuando el valor en la dirección de memoria V2000 es menor que 4050, se energizará la salida Y3.

DirectSOFT



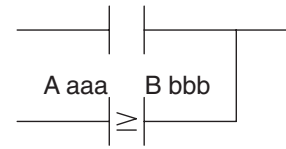
Programador D2-HPP



La instrucción Or comparativa(OR)

DS5	Implied
HPP	Usado

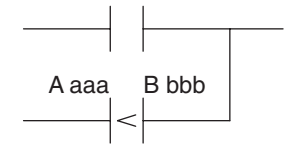
La instrucción OR comparativa conecta un contacto comparativo normalmente abierto en paralelo con otro contacto. El contacto será encendido cuando Aaaa es igual o mayor que Bbbb.



La instrucción Or Not compArativa(ORN)

DS5	Implied
HPP	Usado

La instrucción ORN comparativa conecta un contacto comparativo normalmente cerrado en paralelo con otro contacto. El contacto estará encendido cuando Aaaa es menor que Bbbb.

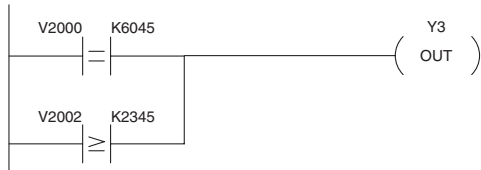


5

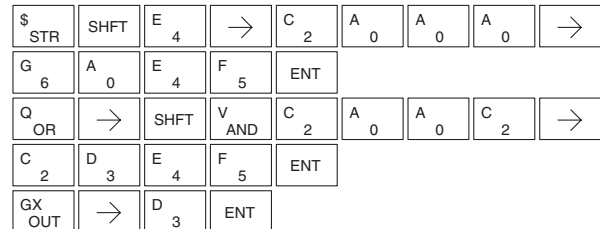
Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... A/B		
Memoria V	V	Vea el mapa de memoria
PUnterO	p	Vea el mapa de memoria
Constante	K	0-9999
Temporizador	TA	0-377
Contador	CTA	0-177

En el ejemplo siguiente, cuando el valor BCD en la dirección de memoria V2000 = 6045 o V2002 ≥ 2345, se energizará la salida Y3.

DirectSOFT

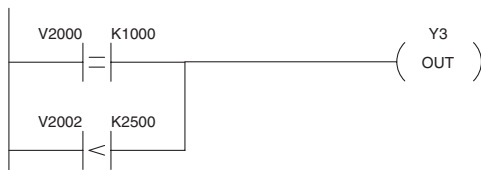


Programador D2-HPP

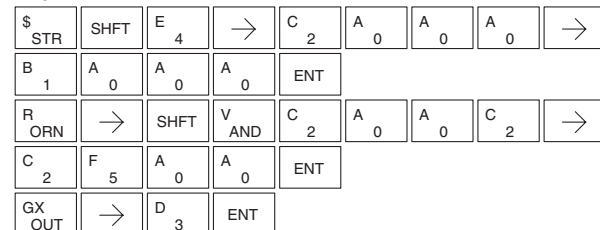


En el ejemplo siguiente, cuando el valor BCD en la dirección de memoria V2000 = 1000 o V2002 es menor que 2500, se energizará la salida Y3.

DirectSOFT



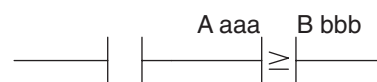
Programador D2-HPP



La instrucción And (AND)

DS5	Implied
HPP	Usado

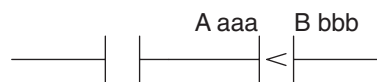
La instrucción de comparación AND conecta un contacto comparativo normalmente abierto en serie con otro contacto. El contacto estará activado a cuando Aaaa es igual o mayor que Bbbb.



La instrucción And Not (ANDN)

DS5	Implied
HPP	Usado

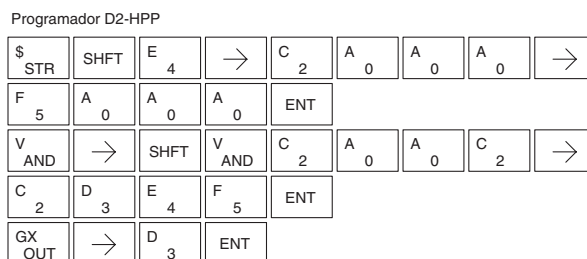
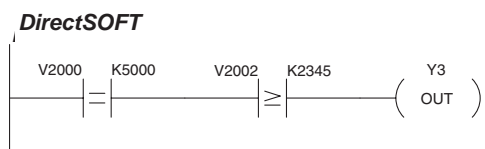
La instrucción de comparación ANDN conecta un contacto comparativo normalmente cerrado en serie con otro contacto. El contacto estará activado cuando Aaaa sea menor que Bbbb.



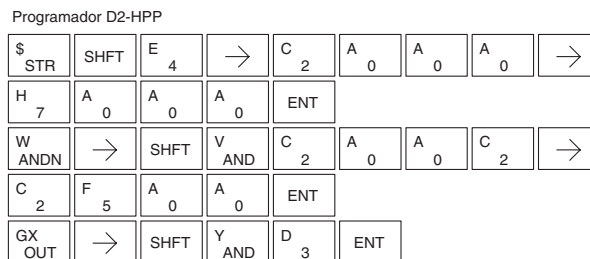
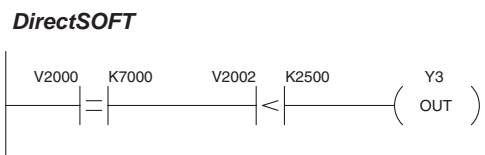
Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... A/B		
Memoria V	V	Vea el mapa de memoria
Puntero	p	Vea el mapa de memoria
Constante	K	0-9999
Temporizador	TA	0-377
Contador	CTA	0-177

5

En el ejemplo siguiente, cuando el valor en la dirección de memoria V2000 es igual a 5000 y V2002 es mayor o igual a 2345, se energizará la salida Y3.



En el ejemplo siguiente, cuando el valor en la dirección de memoria V2000 es igual a 7000 y V2002 es menor que 2500, se energizará la salida Y3.

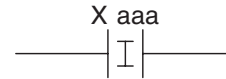


Instrucciones de acción inmediata

La instrucción Store Immediate (STRI)

DS5	Implied
HPP	Usado

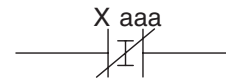
La instrucción STRI comienza una rama nueva o adicional en un renglón. El estado del contacto será el mismo que el estado del punto asociado de la entrada *en el momento que la instrucción se ejecuta*. La memoria imagen no se actualiza.



La instrucción Store Not Immediate (STRNI)

DS5	Implied
HPP	Usado

La instrucción STRNI comienza una rama nueva o adicional en un renglón. El estado del contacto será opuesto al estado del punto asociado de la entrada *en el momento que se ejecuta la instrucción*. La memoria imagen no se actualiza.



5

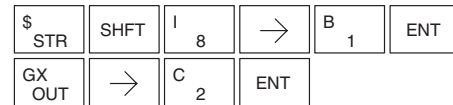
Tipo de operando de datos	Rango del DL06
	aaa
Entradas X	0-777

En el ejemplo siguiente, cuando X1 está ON, se activará la salida Y2.

DirectSOFT



Programador D2-HPP

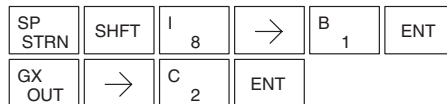


En el ejemplo siguiente, cuando X1 está OFF, se activará la salida Y2.

DirectSOFT



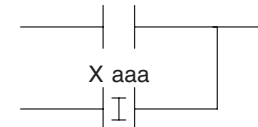
Programador D2-HPP



La instrucción Or Immediate (ORI)

DS5	Implied
HPP	Usado

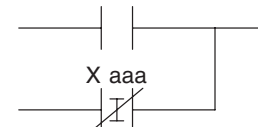
La instrucción ORI conecta dos contactos en paralelo. El estado del contacto será igual que el estado del punto asociado de la entrada *en el momento que se ejecuta la instrucción*. La memoria imagen no es actualizada.



La instrucción Or Not Immediate (ORNI)

DS5	Implied
HPP	Usado

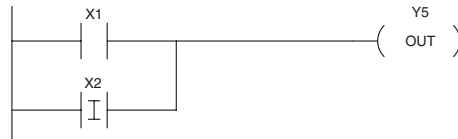
La instrucción ORNI conecta dos contactos en paralelo. El estado del contacto será opuesto al estado del punto asociado de la entrada *en el momento que se ejecuta la instrucción*. La memoria imagen no es actualizada.



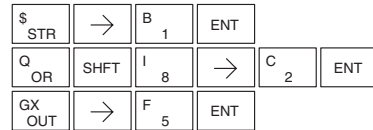
Tipo de operando de datos	Rango del DL06
	aaa
Entradas X	0-777

En el ejemplo siguiente, cuando X1 o X2 están encendidas, se energizará la salida Y5.

DirectSOFT

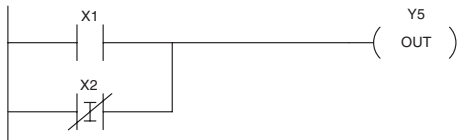


Programador D2-HPP

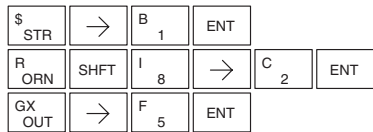


En el ejemplo siguiente, cuando X1 está encendida o X2 está apagada, se energizará Y5.

DirectSOFT



Programador D2-HPP



La instrucción And Immediate (ANDI)

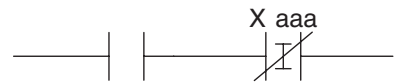
DS5	Implied
HPP	Usado

La instrucción ANDI conecta dos contactos en serie. El estado del contacto será igual que el estado del punto asociado de entrada *en el momento que se ejecuta la instrucción*. La memoria imagen no es actualizada.

La instrucción And Not Immediate (ANDNI)

DS5	Implied
HPP	Usado

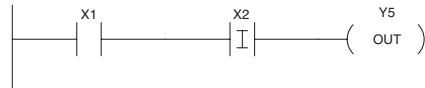
La instrucción ANDNI conecta dos contactos en serie. El estado del contacto será opuesto al estado del punto asociado de entrada *en el momento que se ejecuta la instrucción*. La memoria imagen no es actualizada.



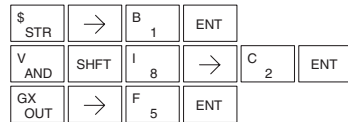
Tipo de operando de datos	Rango del DL06
	aaa
Entradas X	0-777

En el ejemplo siguiente, cuando X1 y X2 están encendidas, se energizará Y5.

DirectSOFT

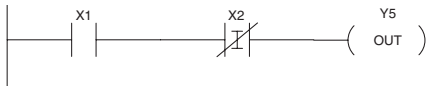


Programador D2-HPP

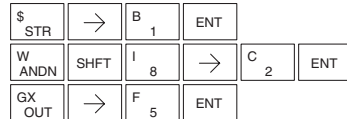


En el ejemplo siguiente, cuando X1 está encendida y X2 está apagada, se energizará Y5.

DirectSOFT



Programador D2-HPP

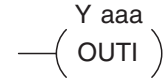


La instrucción Out Immediate (OUTI)

DS5	Usado
HPP	Usado

La instrucción Inmediata OUTI refleja el estado del renglón (ON/OFF) y las salidas del estado discreto (ON/OFF) en el punto especificado de la salida del módulo y la memoria imagen, *en el momento que se ejecuta la instrucción.*

Si se usan múltiples instrucciones OUTI que se refieren al mismo punto discreto es posible que el estado de la salida del módulo cambie múltiples veces en un barrido de la CPU. Vea OR OUT Inmediato.

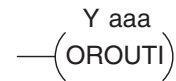


La instrucción Or Out Immediate (OROUTI)

DS5	Usado
HPP	Usado

La instrucción OROUTI ha sido diseñada para usar más de un renglón de lógica discreta para controlar una sola salida. Se puede usar múltiples instrucciones OROUT con la misma bobina de salida, desde que todos los contactos de control de la salida se operan OR juntos.

Si el estado de cualquier renglón está ON *en el momento que se ejecuta la instrucción*, la salida estará también ON.



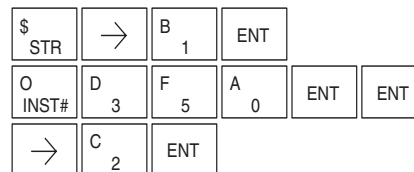
Tipo de operando de datos	Rango del DL06
	aaa
SalidasY	0-777

En el ejemplo siguiente, cuándo X1 está ON, prenderá el punto Y2 de la salida en el módulo de salida. Para entrar la instrucción en el programador D2-HPP, puede usar el número de la instrucción #350 como se muestra, o teclee cada letra del comando.

DirectSOFT

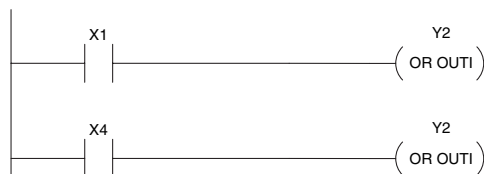


Programador D2-HPP

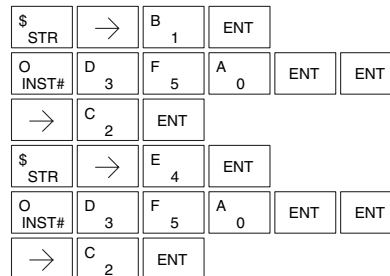


En el ejemplo siguiente, cuándo X1 o X4 están ON, se activará la salida Y2.

DirectSOFT



Programador D2-HPP



La instrucción Load Immediate Formatted (LDIF)

DS5	Usado
HPP	Usado

La instrucción LDIF carga un valor binario de 1 hasta 32 bits en el acumulador. El valor refleja el estado actual del módulo (s) de la entrada(s) *en el momento que la instrucción se ejecuta*. Los bits del acumulador que no son usados por la instrucción son colocados en OFF.

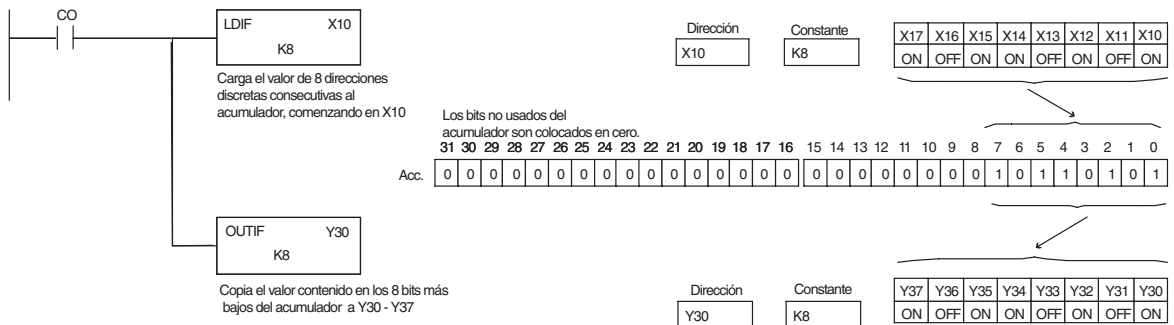


Tipo de operando de datos	Rango del DL06
	aaa
Salidas Y	0-777
Constante K	1-32

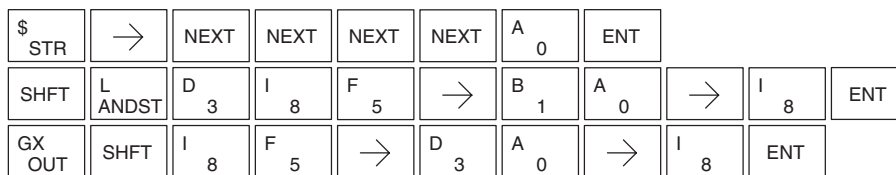
5

En el ejemplo siguiente, cuándo C0 está ON, el modelo binario de X10-X17 se carga en el acumulador usando la instrucción LDIF. La instrucción OUTIF se usa para copiar el número especificado de bits en el acumulador a las salidas especificadas en el módulo de salidas físicas, tales como Y30-Y37.

Esta técnica es útil para copiar rápidamente un conjunto de valores de entradas a salidas (sin esperar el barrido de la CPU).



Programador D2-HPP



La instrucción Set Immediate (SETI)

DS5	Usado
HPP	Usado

La instrucción SET Immediate (SETI) coloca una salida física o un rango de salidas en la memoria imagen y el punto (s) correspondiente(s) de la salida *en el momento en que se ejecuta la instrucción*. Una vez que las salidas se configuran ON no es necesario que el renglón permanezca ON. La instrucción RSTI se puede usar para poner las salidas en OFF.



La instrucción Reset Immediate (RSTI)

DS5	Usado
HPP	Usado

La instrucción RSTI vuelve a 0 u OFF inmediatamente o apaga una salida o un rango de salidas en la memoria imagen y el o los puntos de las salidas *en el momento en que se ejecuta la instrucción*. Una vez que las salidas son colocadas en OFF no es necesario que el renglón permanezca ON.



5

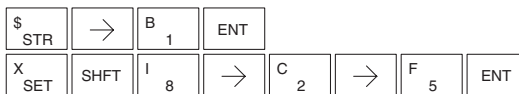
Tipo de operando de datos	Rango del DL06
	aaa
SalidasY	0-777

En el ejemplo siguiente, cuándo X1 está ON, se colocará ON Y2 hasta Y5 en la memoria imagen y en los puntos correspondientes de salidas físicas.

DirectSOFT

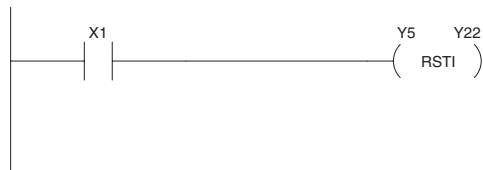


Programador D2-HPP

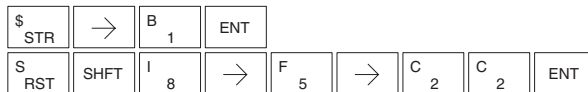


En el ejemplo siguiente, cuándo X1 está ON, Y5 hasta Y22 se colocará OFF en la memoria imagen y en el o los módulos correspondiente de salidas físicas.

DirectSOFT



Programador D2-HPP



La instrucción Load Immediate (LDI)

DS5	Usado
HPP	Usado

La instrucción LDI carga un valor de 16 bits de la memoria en el acumulador. El rango válido de direcciones incluye todos los puntos de entrada en la base local. El valor refleja el estado actual de los puntos de entrada *en el momento que se ejecuta la instrucción*. Esta instrucción se puede usar en vez de la instrucción de LDIF que requiere usted especificar el número de puntos de entrada.



Tipo de operando de datos	Rango del DL06
	aaa
EntradasV	40400-40437

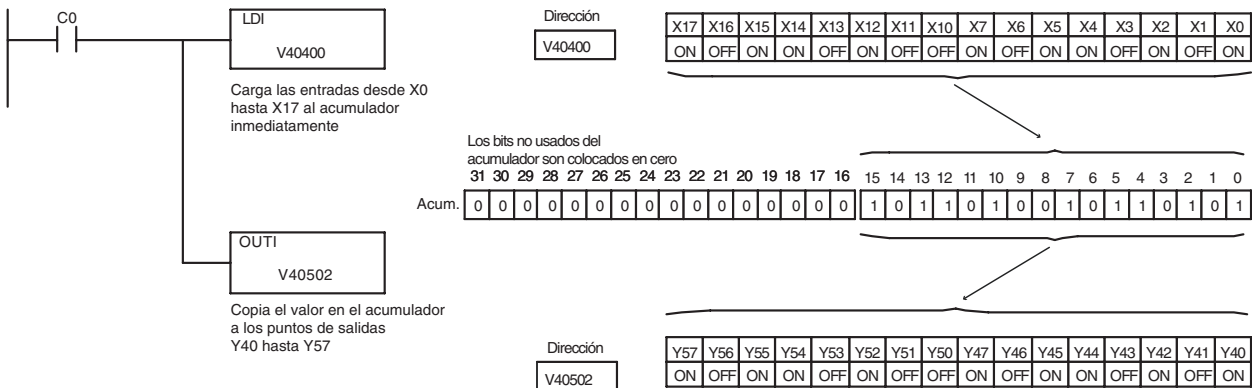
5

En el ejemplo siguiente, cuándo C0 está ON, se carga en el acumulador el modelo binario de X0-X17, usando la instrucción LDI.

La instrucción OUTI es usada para copiar los 16 bits en el acumulador a puntos de salidas, tales como Y40-Y57.

Esta técnica es útil para copiar rápidamente un valor de entradas a puntos de salida (sin esperar que ocurra un barrido de la CPU).

DirectSOFT



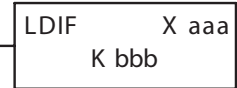
Programador D2-HPP

\$ STR	→	NEXT	NEXT	NEXT	NEXT	A 0	ENT			
SHFT	L ANDST	D 3	I 8	→	E 4	A 0	E 4	A 0	A 0	ENT
GX OUT	SHFT	I 8	→	NEXT	E 4	A 0	F 5	A 0	C 2	ENT

La instrucción Load Immediate Formatted (LDIF)

DS5	Usado
HPP	Usado

La instrucción LDIF carga un valor binario de 1 hasta 32 bits en el acumulador. El valor refleja el estado actual del módulo(s) de la entrada(s) en el momento que la instrucción se ejecuta. Los bits del acumulador que no son usados por la instrucción son colocados en OFF.



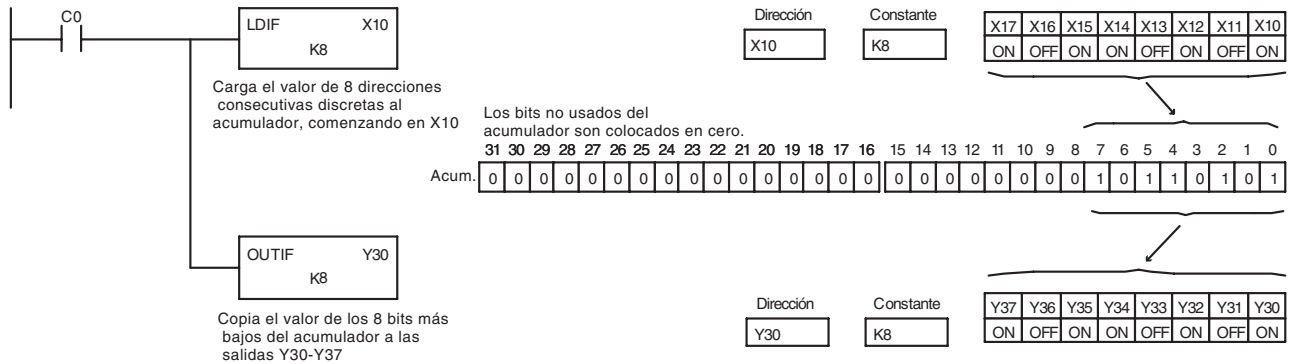
Tipo de operando de datos	Rango del DL06	
	aaa	bbb
Entradas X	0-777	--
Constantes K	--	1-32

5

En el ejemplo siguiente, cuándo C0 está ON, el modelo binario de X10-X17 se carga en el acumulador usando la instrucción LDIF. La instrucción OUTIF se usa para copiar el número especificado de bits en el acumulador a las salidas especificadas en el módulo de salidas físicas, tales como Y30-Y37.

Esta técnica es útil para copiar rápidamente un conjunto de valores de entradas a salidas (sin esperar el barrido de la CPU).

DirectSOFT



Programador D2-HPP

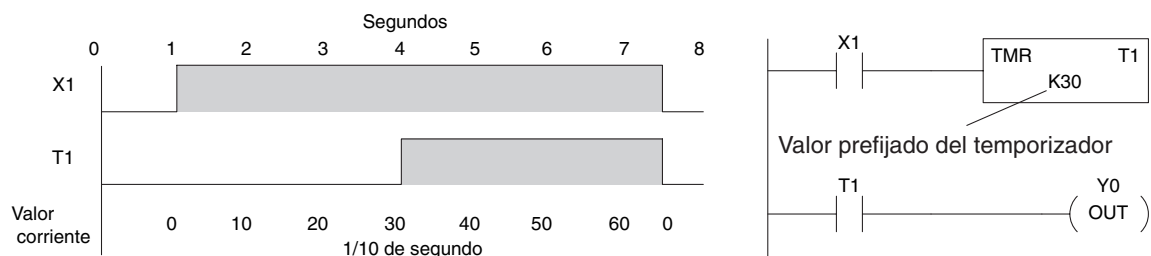
\$	STR	→	NEXT	NEXT	NEXT	NEXT	A	0	ENT		
SHFT	L	ANDST	D	I	F	→	B	A	→	I	ENT
GX	SHFT	I	F	→	D	A	→	I	ENT		
	OUT	8	5		3	0		8			

Instrucciones de temporizador, contadores y shift registers

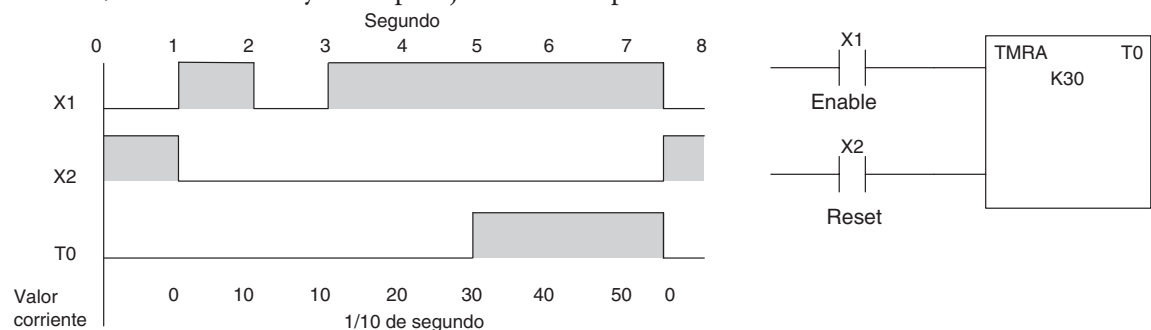
Usando temporizadores o timers

Los temporizadores se usan para medir el tiempo de un evento por una cantidad de tiempo deseada. El temporizador de una entrada medirá el tiempo mientras la entrada está activada. Cuando la entrada cambia de activada a desactivada (ON a OFF) el valor corriente del temporizador se va a 0. Hay bases de tiempo de un décimo de segundo y un centésimo de segundo disponibles, con un tiempo máximo de 999,9 y 99,99 segundos respectivamente. Hay un bit discreto asociado a cada temporizador para indicar que el valor corriente es igual a o mayor que el valor prefijado. El diagrama que mide el tiempo abajo muestra la relación entre la entrada del temporizador, el bit discreto asociado, el valor actual, y el valor prefijado del temporizador.

5



Hay algunos usos que necesitan un temporizador acumulador, queriendo decir que tiene la capacidad de medir el tiempo, parar y después reanudar de donde paró. El temporizador acumulador trabaja en forma similar al temporizador regular, pero se requieren dos entradas. La entrada "enable" parte y para el temporizador. Cuando el temporizador para, se mantiene el tiempo transcurrido. Cuando el temporizador comienza otra vez, el conteo de tiempo continúa a partir del tiempo transcurrido. Cuando se activa la entrada "reset", el tiempo transcurrido es apagado y el temporizador comenzará en 0 cuando se parte nuevamente. Hay bases de tiempo de un décimo de segundo y un centésimo de segundo disponibles con un tiempo máximo de 9999999,9 y 999999,99 segundos respectivamente. El diagrama que mide el tiempo abajo muestra la relación entre la entrada del temporizador, reset del temporizador, bit discreto asociado, valor corriente y valor prefijado del temporizador.

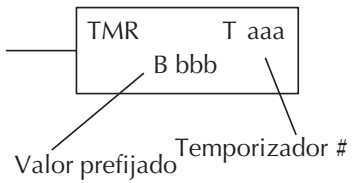


.NOTA: No se usa la coma decimal en este temporizador, pero hay una coma implicada. Los valores corriente y prefijado de todo los tipos de temporizadores están en formato BCD.

Las instrucciones temporizador (TMR) y temporizador rápido (TMRF)

DS5	Usado
HPP	Usado

La instrucción TMR es un temporizador de una entrada con base de tiempo de 0,1 segundo que cuenta tiempo hasta un máximo de 999,9 segundos. La instrucción TMRF es un temporizador de una entrada con base de tiempo de 0,01 segundo que cuenta tiempo hasta un máximo de 99,99 segundos. Estos temporizadores se activan si la lógica de entrada es verdadera (ON) y serán vueltos a 0 si la lógica de entrada es falsa (OFF).



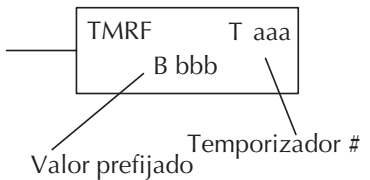
Especificaciones de la instrucción

La Referencia del temporizador (Taaa): Especifica el número del temporizador.

Valor Prefijado (Preset value) (Bbbb): un valor Constantee (K) o una localización de memoria, expresadas en BCD.

Valor corriente: Se refiere al valor de conteo del tiempo en unidades de base de tiempo, y se puede ver en la dirección de memoria T* asociada, valor expresado en BCD. Por ejemplo, el valor corriente del temporizador para T3 se va a la memoria V3.

Bit de estado (Status bit): El bit de estado indica si el temporizador ya alcanzó el valor prefijado de tiempo. Se encuentra en la dirección asociada de memoria T. Estará ON si el valor corriente es igual a o mayor que el valor prefijado del temporizador específico. Por ejemplo, el bit de estado para el Temporizador 2 es T2.



5



NOTA: La constante de valor prefijado (K) del temporizador puede ser cambiada usando un Programador Portátil, aún cuando la CPU está en el modo RUN. Por lo tanto, una memoria en el valor prefijado es requerida solamente si el programa ladder debe cambiar el valor prefijado.

Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... A/B		
Temporizadores T	0-777	—
Memoria V para valores prefijado V	—	400-677 1200-7377 7400-7577 10000-17777
Punteros (solo valor prefijado) P	—	400-677 1200-7377 7400-7577 10000-17777
Constantes (solo valor prefijado) K	—	0-9999
Bits de estado de temporizadores T/V	0-377 o V41100-41117	
Valores corrientes de temporizadores V/T*	0-377	



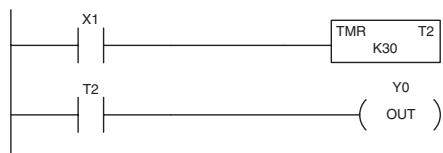
*NOTA: * Con el programador D2-HPP, los bits discretos de estado y el valor corriente del temporizador se obtienen con la misma referencia. DirectSOFT usa referencias separadas, tal como "T2" para el bit de estado y "TA2" para el valor corriente del temporizador T2.*

Usted puede realizar funciones cuando el temporizador alcanza el valor prefijado especificado usando el bit de estado. O, usando contactos de comparación para realizar funciones en intervalos diferentes de tiempo, basado en un temporizador. Los ejemplos siguientes muestran

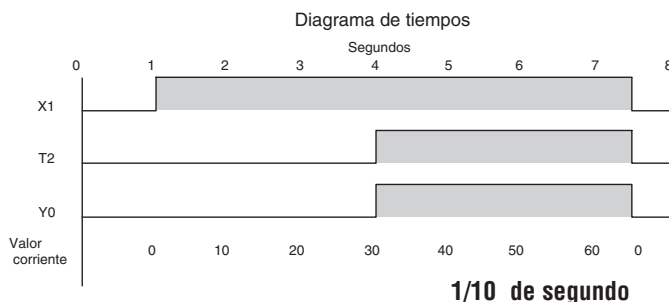
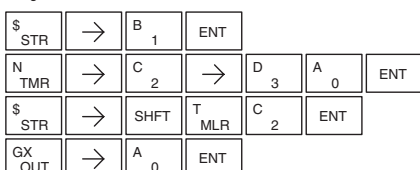
Ejemplo de uso de temporizador con los bits de estado

En el ejemplo siguiente, se usa un temporizador con un valor prefijado de 3 segundos. El bit de estado del temporizador (T2) prenderá cuando el temporizador ha cronometrado por 3 segundos. El temporizador es vuelto a 0 cuándo X1 se apaga, haciendo OFF el bit de estado y coloca en 0 el valor corriente del temporizador.

DirectSOFT



Programador D2-HPP

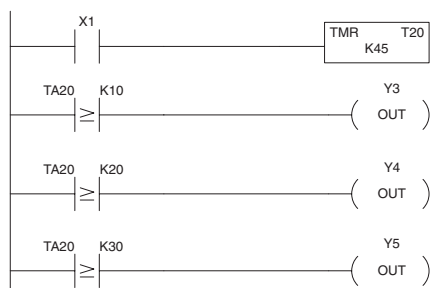


Ejemplo de temporizador con contactos de comparación

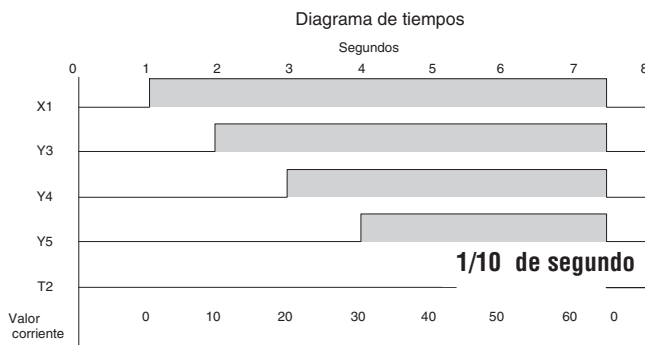
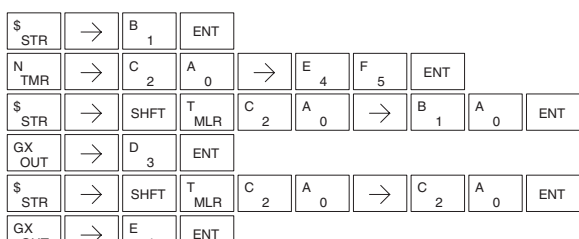
En el ejemplo siguiente, se usa un temporizador con un valor prefijado de 4,5 segundos. Los contactos de comparación se usan para activar Y3, Y4, y Y5 en un intervalo de un segundo respectivamente. Cuando X1 se apaga, el temporizador vuelve a 0 y los contactos de comparación se abren con lo cual Y3, Y4 y Y5 se apagarán.

DirectSOFT

Direct SOFT32



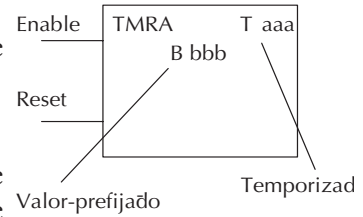
Programador D2-HPP



La Instrucción temporizador acumulador (TMRA)

DS5	Usado
HPP	Usado

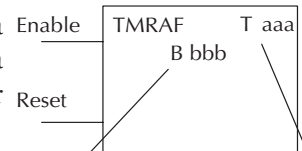
La instrucción TMRA es un temporizador de base de tiempo 0,1 segundo con dos entradas, que cuenta hasta a un máximo de 9999999,9 segundos.



Temporizador acumulador rápido (TMRAF)

DS5	Usado
HPP	Usado

La instrucción TMRAF es un temporizador de base de tiempo de 0,01 segundo con dos entradas que cuenta hasta un máximo de 999999,99 segundos.



Cada uno de estos temporizadores usa 2 palabras de memoria. Estos temporizadores tienen dos entradas, un Enable y un RESET. El temporizador comienza a contar el tiempo cuando la entrada Enable está ON y para el conteo cuando está OFF (Sin volver a cero el valor corriente). La entrada RESET coloca en 0 el valor corriente del temporizador.

La referencia del temporizador (Taaa): Especifica el número del temporizador.

Valor prefijado (Preset value) (Bbbb): un valor constante (K) o una memoria V, en BCD.

Valor corriente: se refiere al valor de conteo del tiempo, y se puede ver en la memoria T* asociada. Por ejemplo, el valor corriente para T3 se va a la memoria V3, y está en BCD.

Bit de estado discreto: El bit de estado indica si el temporizador ya alcanzó el valor prefijado de tiempo. Se encuentra en la dirección asociada de memoria T. Estará ON si el valor corriente es igual a o mayor que el valor prefijado del temporizador específico. Por ejemplo, el bit de estado para el temporizador 2 es T2.

5



NOTA: El TMRA usa dos direcciones consecutivas de memoria para el valor de 8 dígitos y por lo tanto dos direcciones consecutivas de temporizador. Por ejemplo, si es usado TMRA 1, el próximo número disponible del temporizador es TMRA 3.

Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... A/B		
Timers T	0-777	—
Memoria V para valores prefijados V	—	400-677 1200-7377 7400-7577 10000-17777
Punteros (solamente valores prefijados). P	—	400-677 1200-7377 7400-7577 10000-17777
Constantes (solamente valores prefijados) ... K	—	0-99999999
Bits de estado del temporizador T/V	0-377 or V41100-41117	
Valores corrientes del temporizador V /T*	0-377	



*NOTA: * Con el programador D2-HPP, los bits de estado y el valor corriente del temporizador se obtienen con la misma referencia. DirectSOFT separa las referencias, tal como "T2" para el bit de estado y "TA2" para el valor corriente del temporizador T2.*

Los ejemplos siguientes muestran dos métodos de programar los temporizadores. Uno ejecuta la función cuando el temporizador alcanza el valor prefijado usando de valor del bit de estado y el otro

Ejemplo de temporizador acumulador con bits de estado

En el ejemplo siguiente, un temporizador acumulador es usado con un valor prefijado de 3 segundos. El bit de estado temporizador (T6) prenderá cuando el temporizador ha medido un tiempo en total por 3 segundos (30 x 0,1 segundo) y activará Y7.

Note en este ejemplo que el temporizador cuenta el tiempo por 1 segundo, para por 1 segundo y luego reanuda el conteo del tiempo. El temporizador volverá a 0 cuándo C10 prende, haciendo OFF el bit de estado y coloca en 0 el valor corriente del temporizador.

DirectSOFT

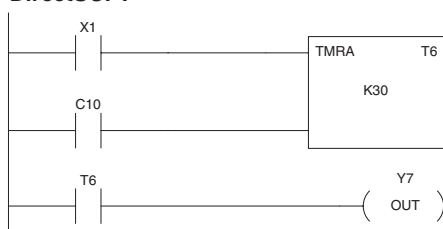
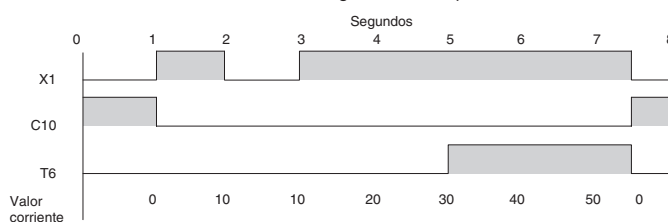


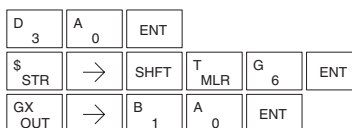
Diagrama de tiempos



Programador D2-HPP



Programador D2-HPP (continuación)



Ejemplo de temporizador acumulador usando contactos de comparación

En el ejemplo siguiente, un temporizador se usa con un valor prefijado de 4,5 segundos. Los contactos de comparación se usan para activar las salidas Y3, Y4 y Y5 en intervalos de un segundo respectivamente. Los contactos de comparación se apagarán cuando el valor corriente del temporizador vuelve a 0.

DirectSOFT32

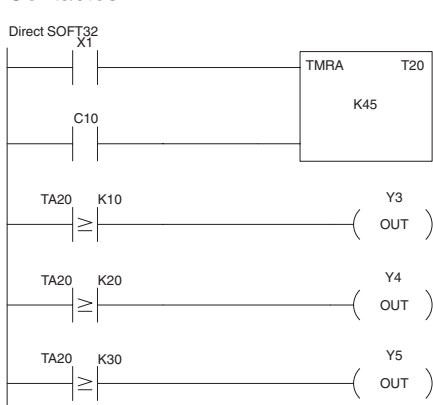
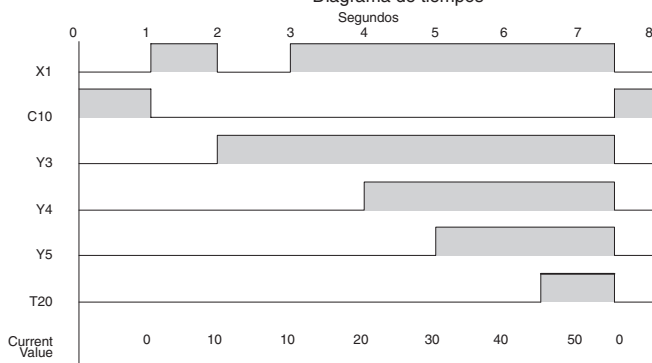
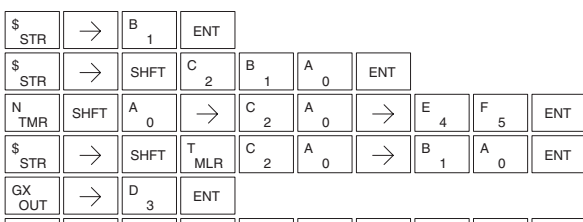


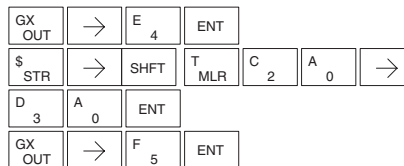
Diagrama de tiempos



Programador D2-HPP



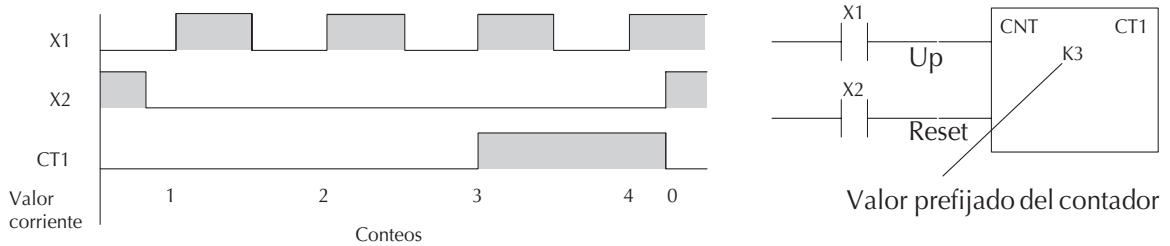
Programador D2-HPP (continuación)



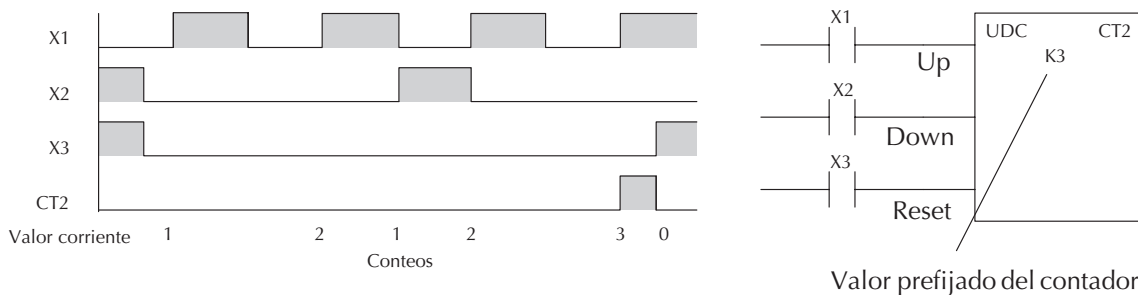
Usando Contadores

Los contadores se utilizan para contar eventos. Los contadores disponibles son contadores ascendentes, contadores incrementales/decrementales y contadores de etapas (usados con programas RLL^{PLUS}).

El contador ascendente (CNT) tiene dos entradas, una entrada de conteo (UP) y una entrada RESET. El valor de conteo máximo es 9999. El diagrama de tiempos abajo muestra la relación entre la entrada, el reset, el bit de estado asociado, el valor corriente y el valor prefijado del contador.

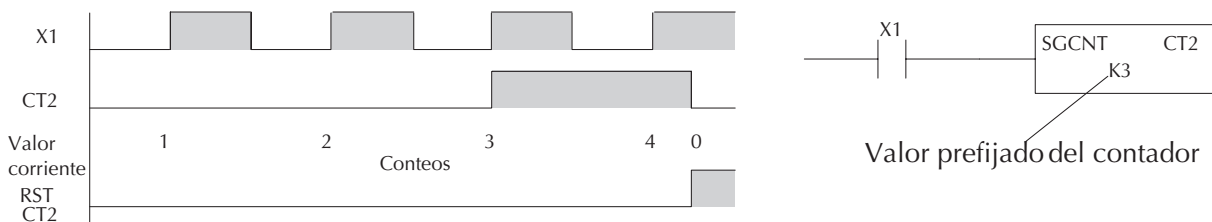


El contador incremental/decremental (UDC) tiene tres entradas, una entrada para contar ascendiendo (UP), otra para contar descendiendo (Down) y el reset. El valor de conteo máximo es 99999999. El diagrama de tiempos abajo muestra la relación entre las entradas, reset, bit de estado asociado, valor corriente y valor prefijado del contador.



Nota: El contador UDC usa dos memorias consecutivas para el valor de 8 dígitos, y por lo tanto, 2 contadores. Por ejemplo si se usa UDC CT1, el próximo contador disponible será CT3.

El contador de etapas (SGCNT) tiene una entrada de conteo y es vuelto a cero por la instrucción RST. Esta instrucción es útil cuando la programación se usa la programación estructurada RLL^{PLUS}. El valor de cuenta máximo es 9999. El diagrama de tiempos abajo muestra la relación entre la entrada, el bit de estado asociado, el valor corriente, el valor prefijado



La instrucción Contador (CNT)

DS5	Usado
HPP	Usado

El Contador es una instrucción de dos entradas que incrementa el valor corriente cuando hay una transición lógica de la entrada COUNT de OFF para ON. Cuando la entrada RESET del contador está ON el contador vuelve a 0. Cuando el valor corriente es igual al valor prefijado, el bit de estado del contador se hace ON y el contador continúa contando hasta un conteo máximo de 9999. El valor máximo se mantendrá hasta que el contador sea vuelto a 0.

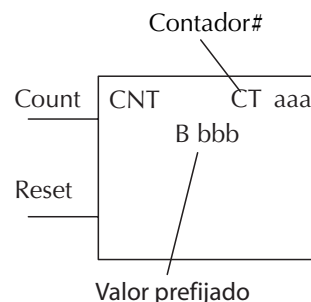
Especificaciones de la instrucción

Referencia del contador (CTaaa): Especifica el número del contador.

Valor prefijado (Bbbb): Una constante (K) o una dirección de memoria, expresado en BCD.

Valor corriente: Los valores corrientes del contador son obtenidos en el contenido de la memoria de CT* asociada, expresado en BCD. La localización de memoria es el número del contador + 1000. Por ejemplo, el valor contador corriente para CT3 está en la dirección de memoria V1003.

El bit de estado: El bit de estado es accesado referenciándose a la dirección asociada de memoria de CT. Estará ON si el valor es igual o mayor que el valor prefijado. Por ejemplo el bit de estado discreto para el contador 2 es CT2.



5



NOTE: A Memoria preset is required if the ladder program or OIP must change the preset.

Tipo de operando de datos	Rango del DL06	
	aaa	bbb
Contadores CT	0-177	—
Memoria V (solamente valor prefijado) V	—	400-677 1200-7377 7400-7577 10000-17777
Punteros (solamente valor prefijado) P	—	400-677 1200-7377 7400-7577 10000-17777
Constantees (solamente valor prefijado) K	—	0-9999
Bits de estado del contador CT/V	0-177 o V41140-41147	
Valores corrientes del contador V /CT*	1000-1177	



*NOTA: * Con el programador D2-HPP, los bits de estado y el valor corriente del contador se obtienen con la misma referencia. DirectSOFT separa las referencias, tal como "CT2" para el bit de estado y "CTA2" para el valor corriente del contador CT2.*

Ejemplo de contador usando el bit de estado

En el ejemplo siguiente, cuando X1 hace una transición de OFF para ON, el valor corriente del contador CT2 se incrementará en uno. Cuando el valor corriente llega al valor prefijado de 3, el bit de estado del contador CT2 prenderá y se activará Y7. Cuando la entrada RESET C10 prende, el bit de estado del contador se apagará y el valor corriente será 0. El valor corriente para el contador CT2 se tendrá en la memoria V1002.

DirectSOFT

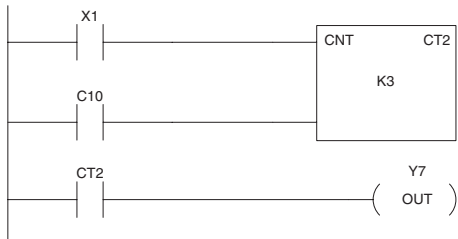
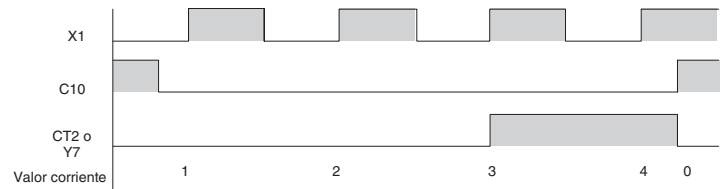
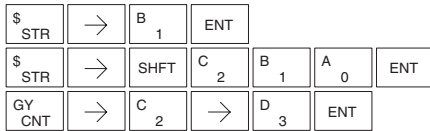


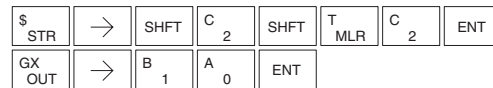
Diagrama del contador



Programador D2-HPP



Programador D2-HPP (cont.)



Ejemplo de contador usando contactos de comparación

En el ejemplo siguiente, cuando X1 hace una transición de OFF para ON, el el valor corriente del contador CT2 se incrementará en 1. Los contactos de comparación se usan para activar las salidas Y3, Y4, y Y5 en conteos diferentes. Cuando el contacto de entrada RESET C10 se cierra, el bit de estado se apagará y el valor corriente del contador volverá a 0 y los contactos de comparación se apagarán.

DirectSOFT

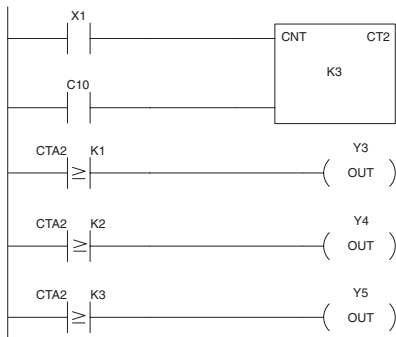
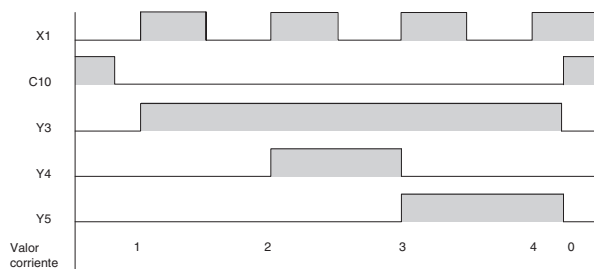
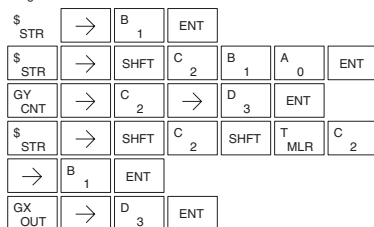


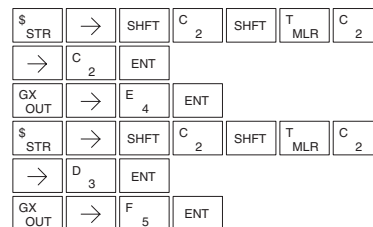
Diagrama del contador



Programador D2-HPP



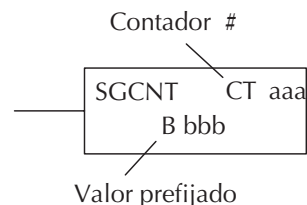
Programador D2-HPP(cont)



La instrucción Contador de Etapas (SGCNT)

DS5	Usado
HPP	Usado

El contador de etapas es un contador de una entrada que incrementa cuando hay una transición lógica de la entrada de OFF para ON. Este contador difiere de otros contadores ya que tendrá su valor actual hasta que se use la instrucción RST (es decir, no tiene una entrada de reset, como los contadores CNT o UDC). El contador de etapas está diseñado para uso en programas RLL^{PLUS}, pero puede ser usado en programas de lógica ladder de relevador. Cuando el valor actual es igual al valor prefijado, el bit contador de estado prende y el contador continúa contando hasta un conteo máximo de 9999. El valor máximo se mantendrá hasta que el contador será vuelto a 0.



Especificaciones de la instrucción

Referencia del contador (CTaaa): Especifica el número del contador.

Valor prefijado (Bbbb): Una constante (K) o una dirección de memoria, en BCD.

Valor corriente: Los valores corrientes del contador son obtenidos en el contenido de la memoria del CT* asociada, en BCD. La dirección de memoria es el número del contador + 1000. Por ejemplo, el valor corriente del contador CT3 se va a la dirección de memoria V1003.

El Bit de estado: El bit de estado es accedido referenciándose a la localización asociada de memoria de CT. Estará ON si el valor es igual o mayor que el valor prefijado. Por ejemplo el bit de estado discreto para el contador 2 es CT2.



NOTA: Al usar un contador dentro de etapas, las etapas deben estar activas por un barrido antes de que la entrada al contador haga una transición de 0-1. Si no es así, no hay transición verdadera y el contador no contará.



NOTA: Solamente se requiere un memoria de valor predefinido si el programa ladder o una interface de operador debe cambiar el valor.

Tipo de operando de datos	Rango del DL06	
	aaa	bbb
Contadores A/B CT	0-177	—
Memoria V (solamente valor prefijado) V	—	400-677 1200-7377 7400-7577 10000-17777
Punteros (solamente valor prefijado) P	—	400-677 1200-7377 7400-7577 10000-17777
Constantes (solamente valor prefijado) K	—	0-9999
Bits de estado del contador CT/V	0-177 o V41140-41147	
Valores corrientes del contador V /CT*	1000-1177	



NOTA: * Con el programador D2-HPP, los bits de estado y el valor corriente del contador se obtienen con la misma referencia. DirectSOFT separa las referencias, tal como "CT2" para el bit de estado y "CTA2" para el valor corriente del contador CT2.

Ejemplo del contador de etapas usando el bit de estado

En el ejemplo siguiente, cuándo X1 hace una transición de OFF para ON, el valor corriente del contador de etapas CT7 incrementará en 1. Cuándo el valor corriente alcanza 3, el bit de estado del contador CT7 prenderá y se activará Y7. El bit de estado del contador CT7 permanecerá ON hasta que el contador sea vuelto a 0 usando la instrucción RST. Cuándo el contador es vuelto a 0, el bit de estado del contador se apagará y el valor corriente será 0. El valor corriente para el contador CT7 se obtendrá en la memoria V1007.

DirectSOFT

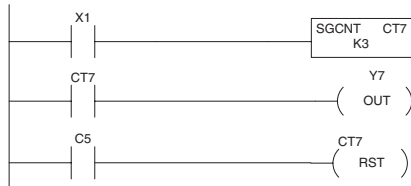
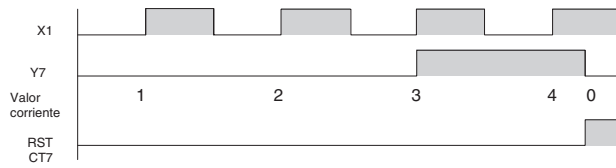
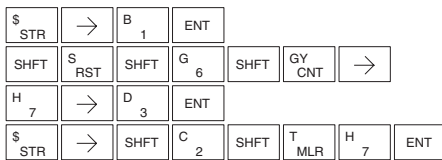


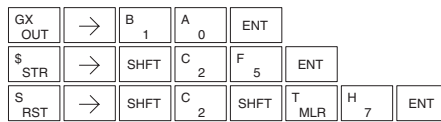
Diagrama del contador



Programador D2-HPP



Programador D2-HPP (cont.)



Ejemplo de contador de etapas usando contactos de comparación

En el ejemplo siguiente, cuándo X1 hace una transición de OFF para ON, el valor corriente del contador CT2 incrementará en 1. Los contactos de comparación se usan para activar Y3, Y4 y Y5 en conteos diferentes. Aunque esto no se muestre en el ejemplo, cuando el contador usa la instrucción RST, el bit de estado del contador se apagará y el valor corriente será 0. El valor corriente y el valor corriente para el contador CT2 se mantendrá en la memoria V1002 (o CTA2).

DirectSOFT32

DirectSOFT

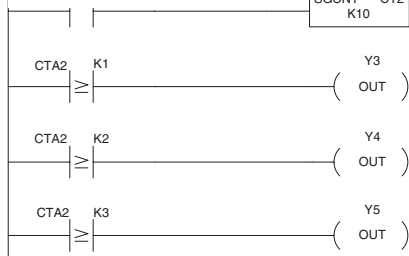
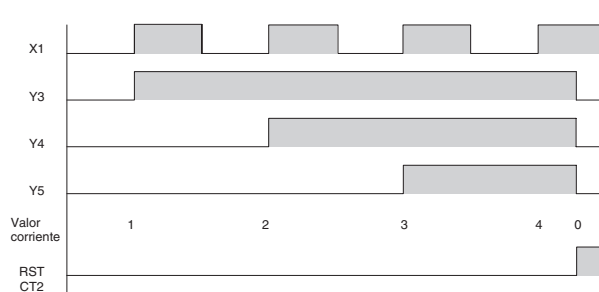
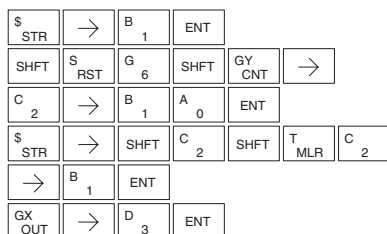


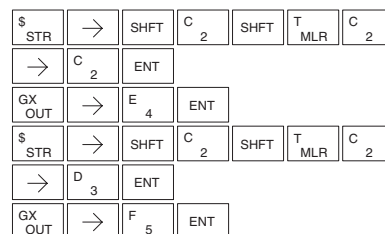
Diagrama de contador



Programador D2-HPP



Programador D2-HPP (cont.)



La instrucción Up Down Counter (UDC)

DS5	Usado
HPP	Usado

El contador UDC cuenta subiendo el conteo en la transición de falso para verdadero (OFF a ON) en la entrada UP y cuenta hacia abajo en cada transición de OFF para ON en la entrada Down. El contador vuelve a 0 cuando la entrada RESET está ON. El rango de conteo es 0-99999999. La entrada de conteo que no se usa debe estar apagada para que la entrada activa de conteo pueda funcionar.

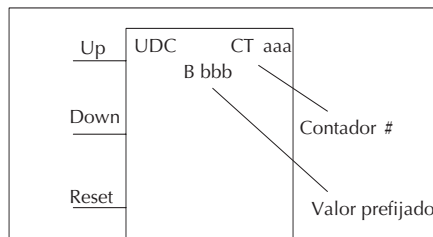
Especificación de la instrucción

Referencia del contador (CTaaa): Especifica el número del contador.

Valor prefijado(Bbbb): Valor constantee (K) o dos direcciones consecutivas de memoria V, en BCD.

Valor corriente: El valor corriente de conteo es un valor de palabra doble que se puede acceder referenciando las direcciones de memoria de CT* asociadas, en BCD. La dirección de memoria V es el número del contador + 1000. Por ejemplo, el valor corriente para el contador CT5 está en las memorias V1005 y V1006.

El bit de estado de contador: El bit de estado es accesado al referenciar la dirección asociada de memoria del contador CT. Opera estando ON si el valor es igual a o mayor que el valor prefijado. Por ejemplo el bit de estado discreto para el contador 12 sería CT12.



Atención: El contador UDC usa dos direcciones de memoria para el valor corriente de 8 dígitos. Esto es, el contador UDC usa dos direcciones de memorias consecutivas. Si se usa el contador UDC CT1 en un programa, el próximo contador disponible en ese programa es CT3.



NOTA: UDC usa dos memorias consecutivas para el valor de 8 dígitos, por lo tanto dos localizaciones consecutivas de temporizador. Por ejemplo, si se usa UDC CT1, el número disponible siguiente es CT3.



NOTA: Solamente se requiere un memoria de valor predefinido si el programa ladder o una interface de

Tipo de operando de datos	Rango del DL06	
	aaa	bbb
Contadores A/B CT	0-177	—
Memoria V (solamente valor prefijado) V	—	400-677 1200-7377 7400-7577 10000-17777
Punteros (solamente valor prefijado) P	—	400-677 1200-7377 7400-7577 10000-17777
Constantes (solamente valor prefijado) K	—	0-9999
Bits de estado del contador CT/V	0-177 o V41140-41147	
Valores corrientes del contador V /CT*	1000-1177	

operador debe cambiar el valor.

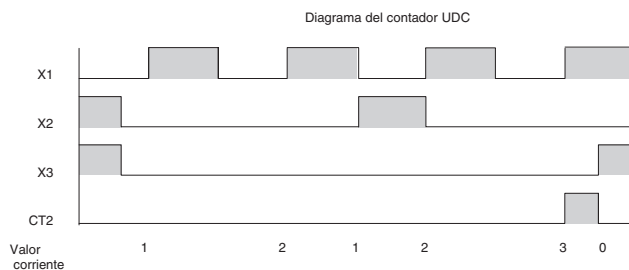
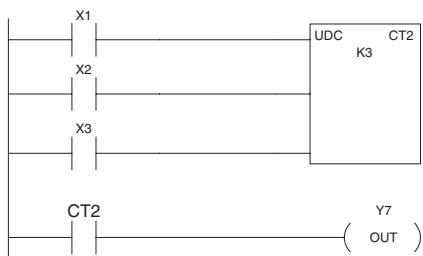


*NOTA: * * Con el programador D2-HPP, los bits de estado y el valor corriente del contador se obtienen con la misma referencia. DirectSOFT usa referencias diferentes, tal como "CT2" para el bit de*

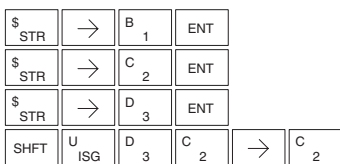
Ejemplo de contador incremental/decremental usando el bit de estado

En el ejemplo siguiente, si X2 y X3 están apagados, cuándo X1 pase de OFF para ON el valor corriente del contador incrementará en 1. Si X1 y X3 están apagados el valor corriente del contador decrece en 1 cuándo X2 pasa de OFF para ON. Cuándo el valor de conteo alcanza el valor prefijado de 3, el bit de estado del contador prenderá. Cuándo X3 prende, el bit de estado del contador se apagará y el valor actual se hará 0.

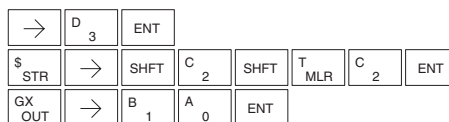
DirectSOFT



Programador D2-HPP



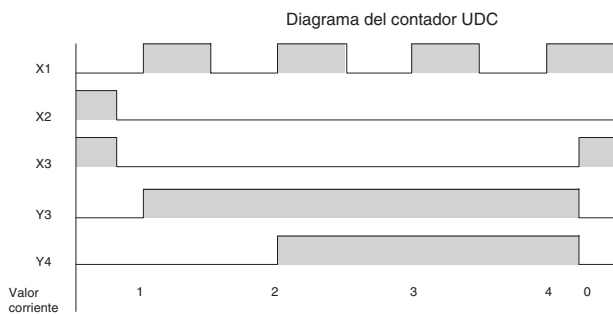
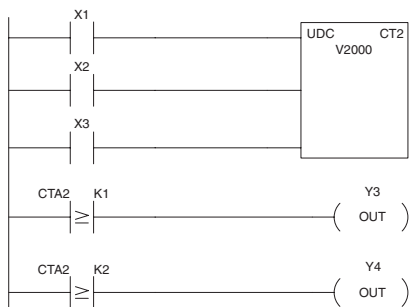
Programador D2-HPP (cont)



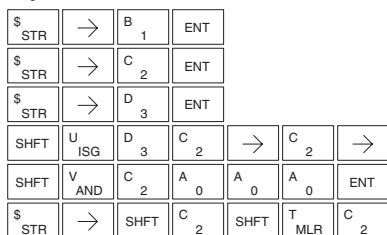
Ejemplo de contador UDC con contactos de comparación

En el ejemplo siguiente, si X2 y X3 están apagados, cuándo X1 pase de OFF para ON el valor corriente del contador incrementará en 1. Si X1 y X3 están apagados el valor corriente del contador decrece en 1 cuándo X2 pasa de OFF para ON. Cuándo el valor de conteo alcanza el valor prefijado de 3, el bit de estado del contador prenderá. Cuándo X3 prende, el bit de estado del contador se apagará y el valor actual se hará 0.

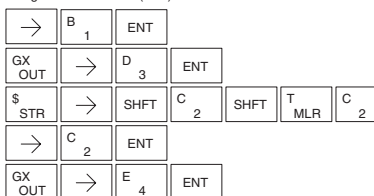
DirectSOFT



Programador D2-HPP



Programador D2-HPP (cont)



Operaciones de carga y copia del acumulador y stack

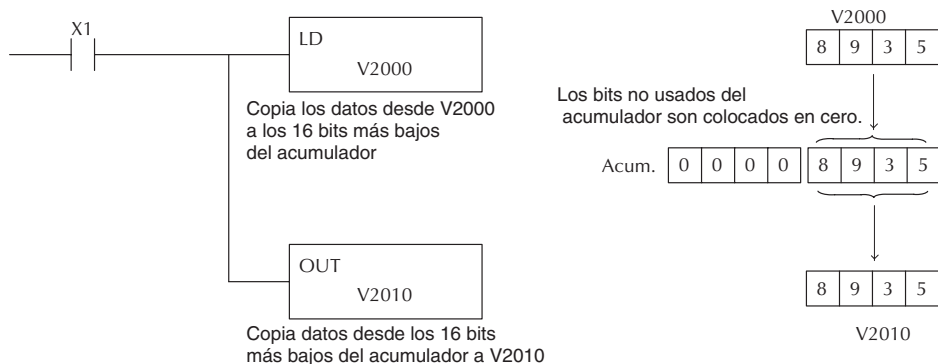
Usando el acumulador

El acumulador en la unidad de procesamiento central (CPU) del PLC DL06 es una memoria intermedia (RAM) de 32 bits que se usa como una localización de almacenamiento temporaria para datos que se copian o son manipulados de alguna manera. Por ejemplo, usted tiene que usar el acumulador para realizar operaciones aritméticas tales como sumar, restar, multiplicar, etc. Ya que hay 32 bits, usted puede operar con un número de 8 dígitos BCD o datos ASCII sobre cualquier tipo de datos. *El acumulador es vuelto a 0 al fin de cada barrido de la CPU, es decir, el acumulador no retiene información.*

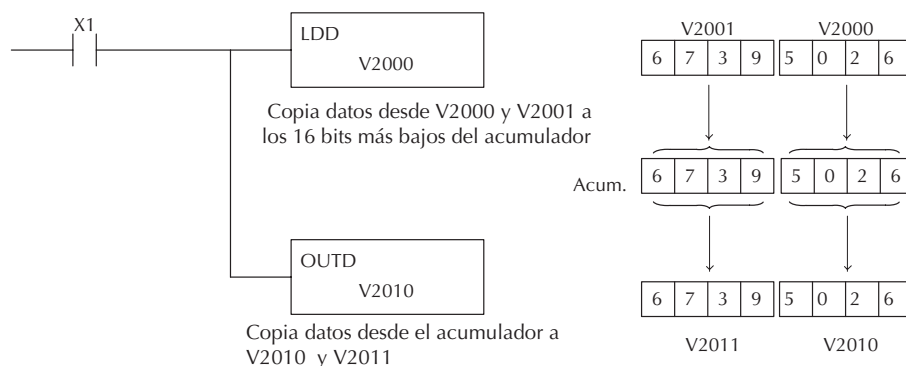
5

Copiando datos al acumulador

Las instrucciones LD y OUT y sus variaciones se usan para copiar datos de una dirección de memoria V al acumulador o para copiar los datos del acumulador a una memoria V. El ejemplo siguiente copia los datos de la memoria V2000 a la memoria V2010.



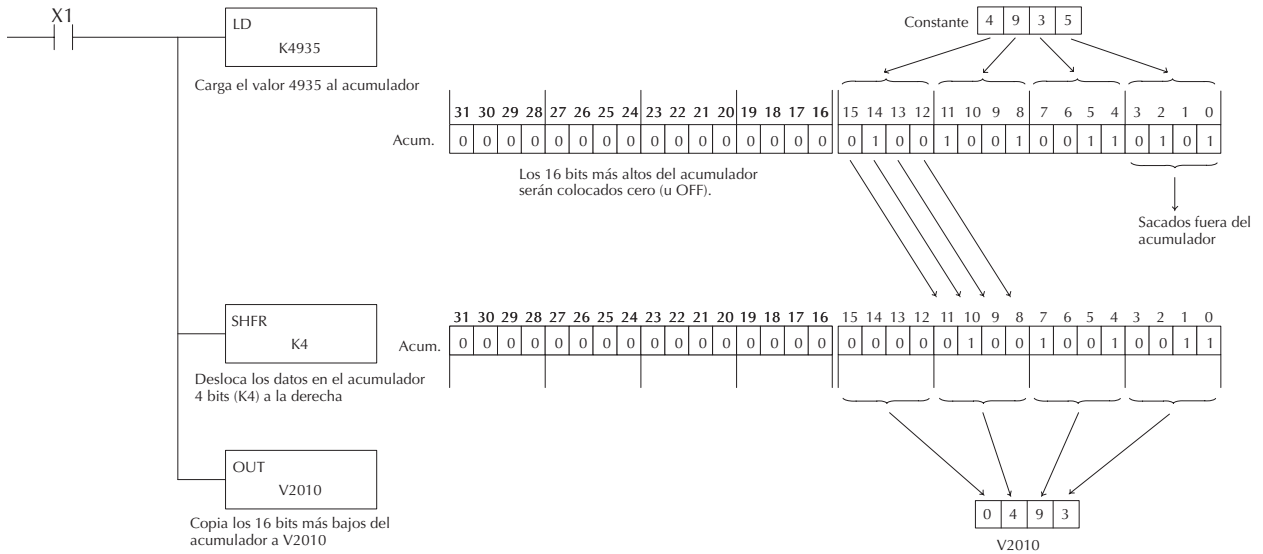
Ya que el acumulador es de 32 bits y las direcciones de memoria V son de 16 bits, las instrucciones LDD y OUTD (o las variaciones de las mismas) usan dos direcciones consecutivas de memoria V o una constante de 8 dígitos BCD para copiar los datos al o desde una dirección de memoria V al acumulador. Por ejemplo si usted quiere copiar los datos de V2000 y V2001 para V2010 y V2011 la manera más eficiente de realizar esta función sería como sigue:



Cambiando los datos del acumulador

Las instrucciones que manipulan datos también usan el acumulador. El resultado de los datos manipulados se queda en el acumulador. Los datos que tenía el acumulador antes de hacer la operación correspondiente se pierden en el acumulador.

El ejemplo siguiente copia la constante 4935 en el acumulador, disloca a la derecha los datos en 4 bits y copia el resultado a V2010.

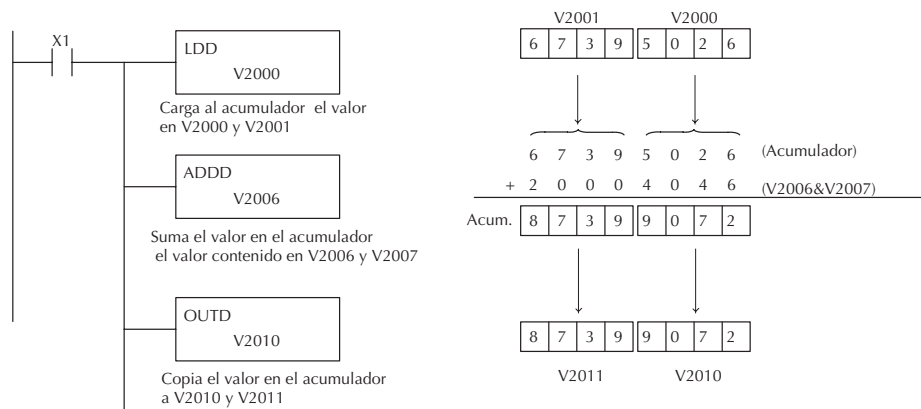


5

Algunas instrucciones de manipulación de datos usan 32 bits. Usan dos direcciones consecutivas de memoria V o una constante de 8 dígitos BCD para manipular los datos en el acumulador.

En el ejemplo siguiente, cuando X1 está ON, se carga el valor en V2000 y V2001 en el acumulador usando la instrucción LDD.

El valor en el acumulador se suma al valor en V2006 y V2007 usando la instrucción ADDD. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.



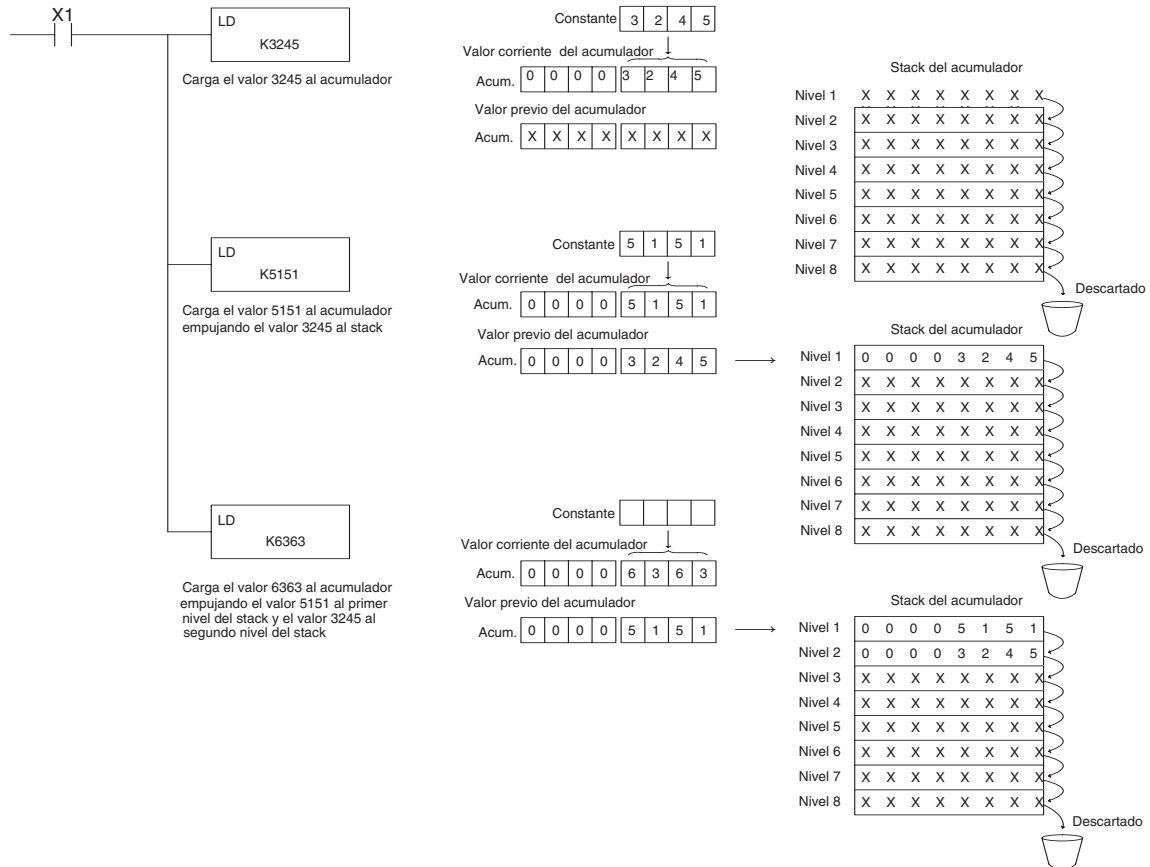
Usando el Stack del acumulador

El stack del acumulador (Una pila de memorias) es usado por instrucciones que requieren más de un parámetro para ejecutar una función o para una función definida por el usuario. El Stack del acumulador se usa cuando se ejecuta más de una instrucción LD sin el uso de una instrucción OUT. El contenido del stack vuelve a 0 al fin de cada barrido.

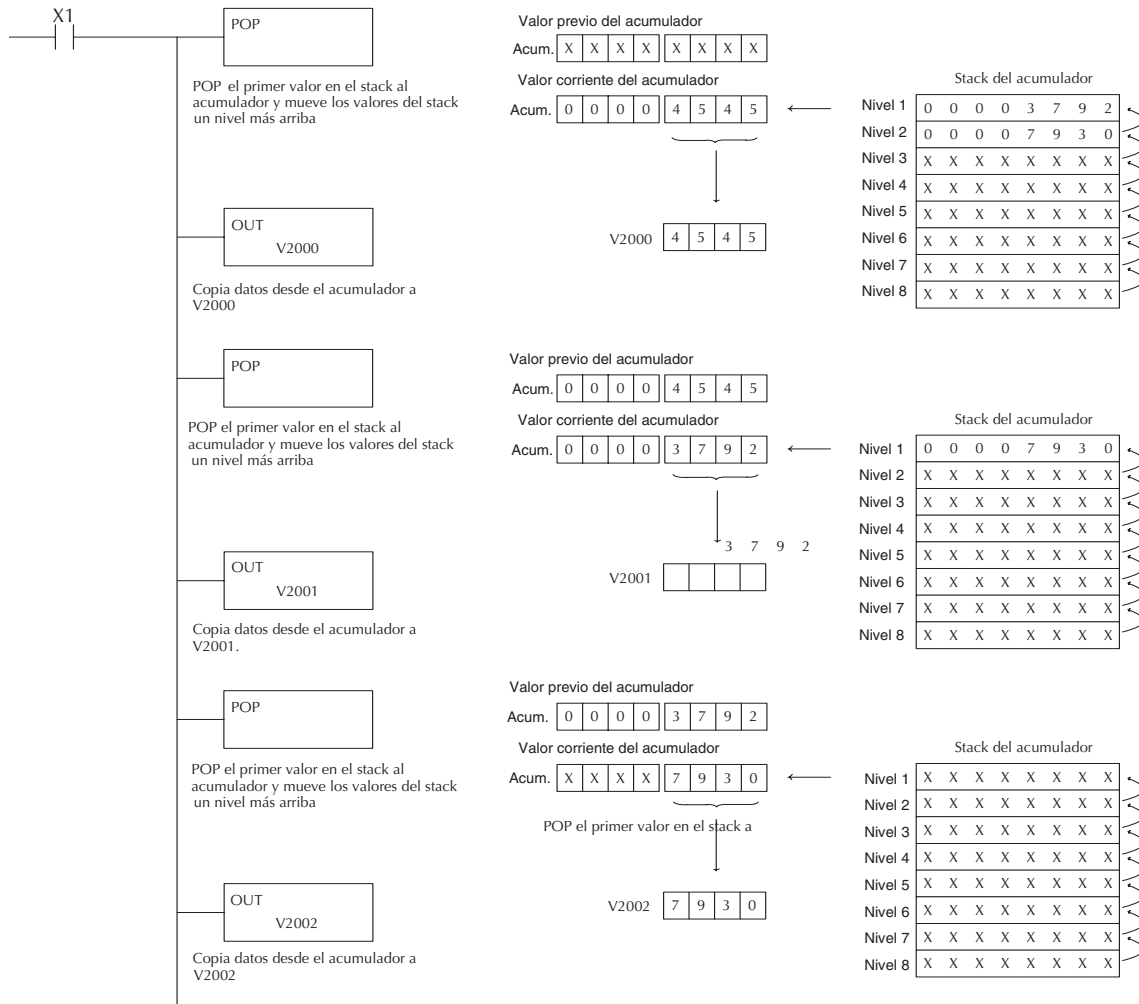
La primera instrucción LD en el barrido coloca un valor en el acumulador. Cada instrucción LD después, sin el uso de la instrucción OUT, coloca un valor en el acumulador y el valor que estaba en el acumulador se coloca en el Stack del acumulador.

La instrucción OUT anula la instrucción previa de LD y no coloca el valor que estaba en el acumulador en el Stack del acumulador cuando se ejecuta la próxima instrucción LD. Cada vez que un valor se coloca en el acumulador amontona los otros valores en el Stack y se empujan hacia abajo una dirección de memoria.

El acumulador tiene ocho niveles de profundidad (ocho registros de 32 bits). Si hay un valor en la octava localización cuando un valor nuevo se coloca en el Stack, el valor en la octava localización sale fuera del Stack y no se puede recuperar, es decir, se pierde



La instrucción POP rota los valores hacia arriba por el Stack al acumulador. Cuando se ejecuta la instrucción POP el valor que estaba en el acumulador se limpia y el valor que estaba encima del Stack pasa al acumulador. Los valores en el Stack se desplazan una posición hacia arriba en el Stack.



Usando punteros

Muchas de las instrucciones del PLC de la serie DL06 permitirán usar los punteros de la memoria V como un operando (comúnmente conocido como direccionamiento indirecto). Los punteros permiten que las instrucciones obtengan los datos de direcciones de memoria V indicadas por el valor del puntero.

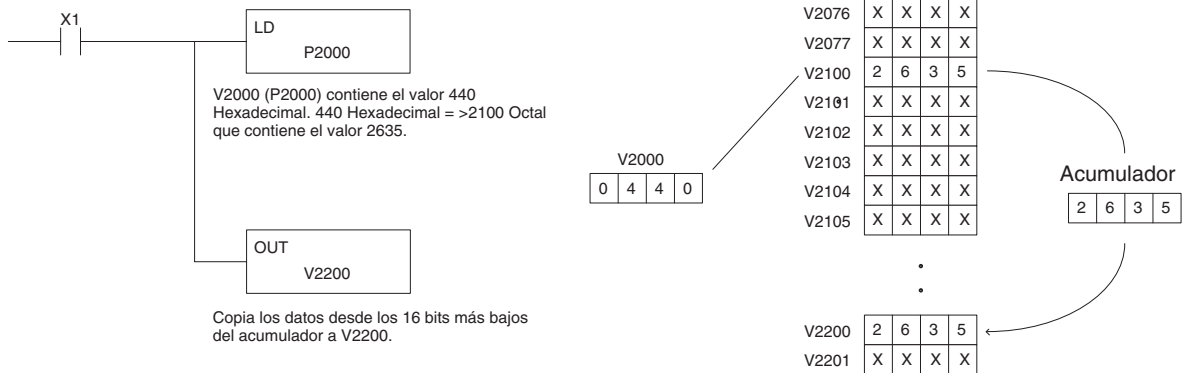


NOTA: La dirección de memoria DL06 V está en octal. Sin embargo, el puntero se refiere a una dirección de memoria V con valores hexadecimales. Use la instrucción LDA para transformar una dirección a la dirección de puntero. Esta instrucción realiza la conversión Octal a Hexadecimal automáticamente.

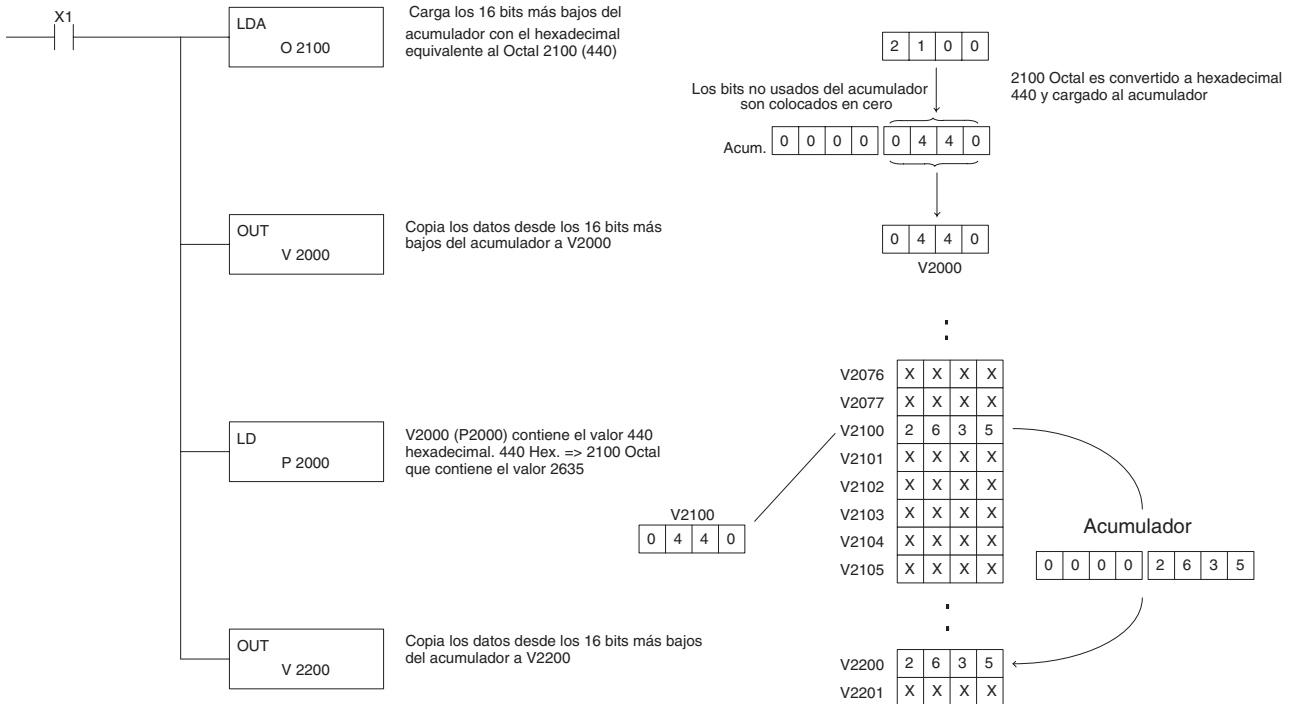
En el ejemplo siguiente usaremos un operando del puntero en una instrucción LD. La dirección de memoria V2000 es usada como localización del puntero. V2000 contiene el valor 440 que la CPU ve como el equivalente hexadecimal de la memoria octal V2100. La CPU copiará los datos de V2100 que en este ejemplo contiene el valor [2635] en la palabra más baja del acumulador.

Capítulo 5: Instrucciones de Acumulador/Stack Load y salidas de datos (OUT)

5



El ejemplo siguiente es idéntico al de arriba con una excepción. La instrucción LDA convierte automáticamente la dirección octal a hexadecimal.



La instrucción Load (LD)

DS5	Usado
HPP	Usado

La instrucción Load (LD) es una instrucción de 16 bits que carga o copia el valor (Aaaa), que es una dirección de memoria V o una constante de 4 dígitos BCD/Hexadecimal, en los 16 bits más bajos del acumulador. Los 16 bits más altos del acumulador son forzados a 0.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P
Constante	K
	0-FFFF

Indicadores	Descripción
SP53	Está ON cuando el puntero está fuera del rango disponible.
SP70	ON si el valor en el acumulador por cualquier instrucción es negativo.
SP76	ON cuando cualquier instrucción carga un valor 0 al acumulador.

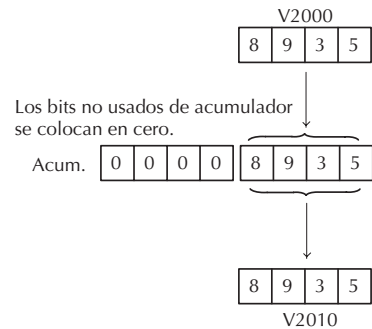
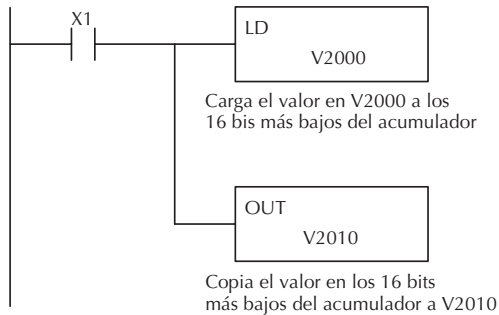
5



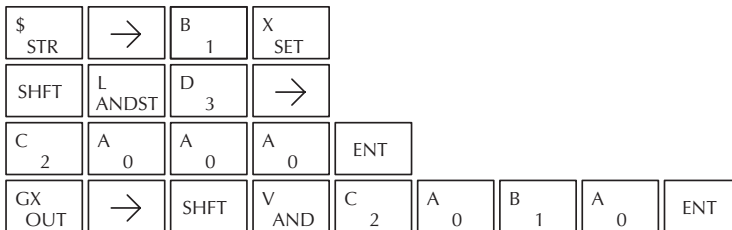
NOTA: Dos instrucciones consecutivas LD colocarán el valor de la primera instrucción LD en el Stack del acumulador.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V2000 al acumulador y luego se copia a V2010.

DirectSOFT



Programador D2-HPP



La instrucción Load Double (LDD)

DS5	Usado
HPP	Usado

La instrucción LDD es una instrucción de 32 bits que carga o copia el valor (Aaaa), que es: o dos direcciones consecutivas de memoria V o una constante de 8 dígitos BCD/Hexadecimal, en el acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0–FFFF

Indicadores	Descripción
SP53	Está ON cuando el puntero está fuera del rango disponible.
SP70	ON si el valor en el acumulador por cualquier instrucción es negativo.
SP76	ON cuando cualquier instrucción carga un valor 0 al acumulador.

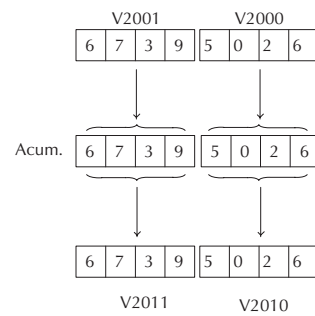
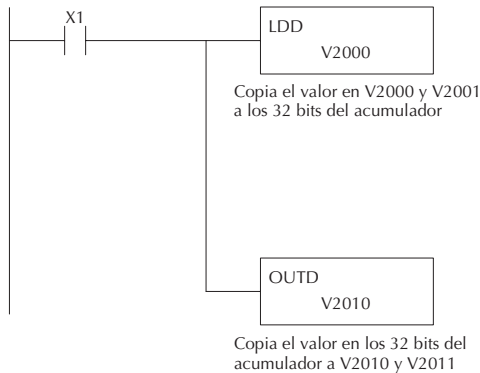
5



NOTA: Dos instrucciones LDD consecutivas colocarán el valor de la primera instrucción LDD en el Stack del acumulador.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor de 32 bits en V2000 y V2001 en el acumulador y es copiado a V2010 y V2011.

DirectSOFT



Programador D2-HPP

\$	STR	→	B	1	ENT
SHFT	L	ANDST	D	3	→
C	A	A	A	0	ENT
GX	SHFT	D	3	→	
C	A	B	A	0	ENT

La instrucción Load Formatted (LDF)

DS5	Usado
HPP	Usado

La instrucción LDF carga o copia un conjunto de 1 a 32 bits consecutivos de direcciones discretas de memoria en el acumulador. La instrucción requiere una dirección (Aaaa) de inicio y el número de bits (Kbbb) a ser cargado. Los bits no usados del acumulador se colocan en 0.



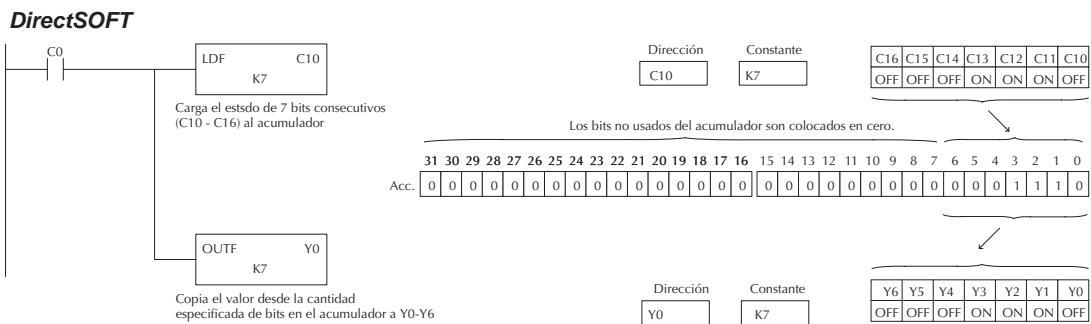
Tipo de operando de datos	Rango del DL06	
	aaa	bbb
Entradas X	0-777	—
Salidas Y	0-777	—
Relevadores de control C	0-1777	—
Bits de Etapas S	0-1777	—
Bits de temporizadores T	0-377	—
Bits de contadores CT	0-177	—
Relevadores especiales SP	0-777	—
Constante K	—	1-32

Indicadores	Descripción
SP70	On anytime the value in the accumulator is negative.
SP76	On when any instrucción loads a value of zero into the accumulator.

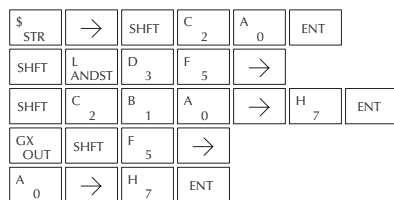


NOTA: Dos instrucciones consecutivas LDF colocarán el valor de la primera instrucción en el Stack del acumulador.

En el ejemplo siguiente, cuándo C0 está ON, el conjunto de bits de C10-C16 (7 bits) será copiado al acumulador usando la instrucción LDF. Los 7 bits más bajos del acumulador son



Programador D2-HPP



La instrucción Load Address (LDA)

DS5	Usado
HPP	Usado

La instrucción LDA es una instrucción de 16 bits. Convierte cualquier valor octal (o dirección) al valor del equivalente hexadecimal y lo carga (o copia) al acumulador. Esta instrucción es útil cuando se requiere un parámetro de dirección ya que todas las direcciones para el sistema DL06 están en octal.



Tipo de operando de datos	Rango del DL06
	aaa
Octal Address 0	Vea el mapa de memoria

Indicadores	Descripción
SP70	On anytime the value in the accumulator is negative.
SP76	On when any instrucción loads a value of zero into the accumulator.

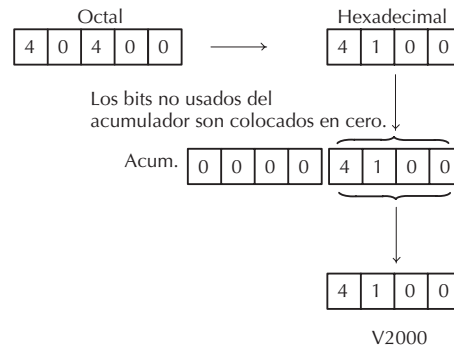
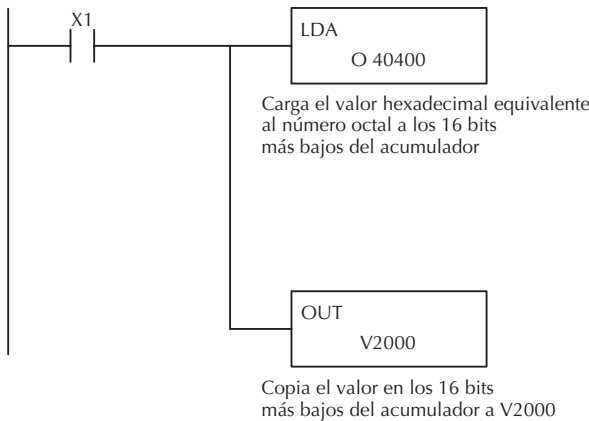
5



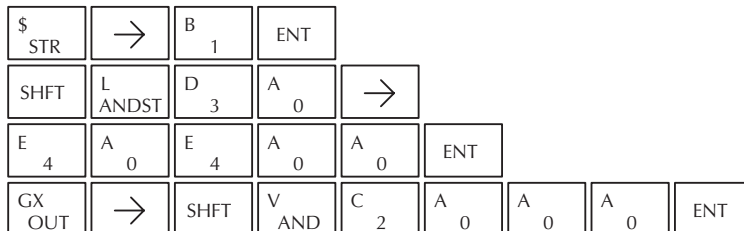
NOTA: Dos instrucciones consecutivas LDA colocan el contenido de la primera instrucción en el stack del acumulador.

En el ejemplo siguiente cuando X1 está ON, el número octal 40400 será convertido a un 4100 hexadecimal y cargado en el acumulador usando la instrucción LDA. El valor en los 16 bits más bajos del acumulador es copiado a V2000 usando la instrucción OUT.

DirectSOFT



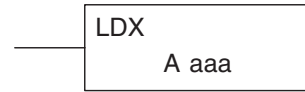
Programador D2-HPP



La instrucción Load Accumulator Indexed (LDX)

DS5	Usado
HPP	Usado

Esta instrucción de 16 bits especifica una dirección de la fuente (la memoria V) que será dislocada por el valor en la primera dirección del Stack. Esta instrucción LDX interpreta el valor en la primera dirección del Stack como hexadecimal. El valor en la dirección ya dislocada (la dirección de la fuente + el desvío) es cargado en los 16 bits más bajos del acumulador. Los 16 bits más altos del acumulador son forzados a 0.



Sugerencia: — La instrucción LDA se puede usar para convertir una dirección de octal a una dirección hexadecimal y cargar el valor en el acumulador

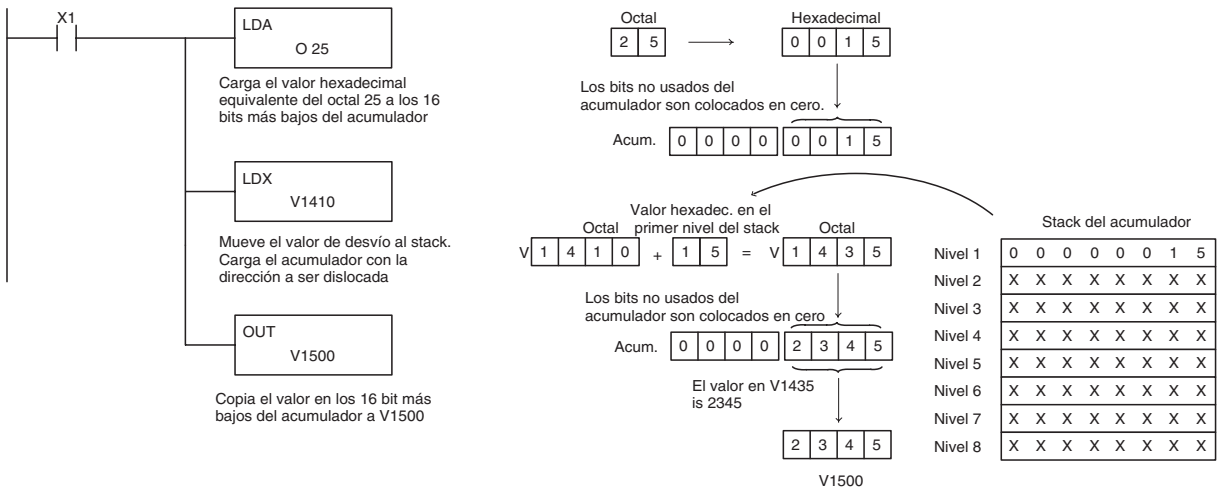
Tipo de operando de datos	Rango del DL06	
	A	aaa
Memoria	V	Vea el mapa de memoria
Puntero	P	Vea el mapa de memoria

Indicadores	Descripción
SP53	Está ON cuando el puntero está fuera del rango disponible.
SP70	ON si el valor en el acumulador por cualquier instrucción es negativo.
SP76	ON cuando cualquier instrucción carga un valor 0 al acumulador.



NOTA: Dos instrucciones consecutivas de la instrucción LDX colocarán el valor de la primera instrucción en el Stack del acumulador.

En el ejemplo siguiente cuando X1 está ON, el equivalente hexadecimal del octal 25 será cargado al acumulador (este valor se colocará en el Stack cuando se ejecuta la instrucción LDX). La dirección de memoria V1410 se suma al valor en el primer el nivel del Stack y el valor de esta dirección es cargado en los 16 bits más bajos del acumulador usando la instrucción LDX. El valor en los 16 bits más bajos del acumulador es copiado a V1500 usando la instrucción OUT.



Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	A	0	→	C	2	F	5	ENT				
SHFT	L	ANDST	D	3	X	SET	→	B	1	E	4	B	1	A	0	ENT

La instrucción Load Accumulator Indexed from Data Constantes (LDSX)

DS5	Usado
HPP	Usado

La instrucción LDSX es una instrucción de 16 bits. La instrucción especifica un Area de Data Label (DLBL) (de Etiqueta de Datos) donde se almacenan constantes numéricas o ASCII. Este valor se carga en los 16 bits más bajos del acumulador.



La instrucción LDSX usa el valor en el primer nivel del Stack del acumulador como un "desvío" para determinar cuál constante numérica o ASCII dentro del Area DLBL se carga en el acumulador. La instrucción de LDSX interpreta el valor en el primer nivel del Stack del acumulador como un valor hexadecimal.

Sugerencia: — La instrucción LDA se puede usar para convertir octal a hexadecimal y cargar el valor en el acumulador.

5

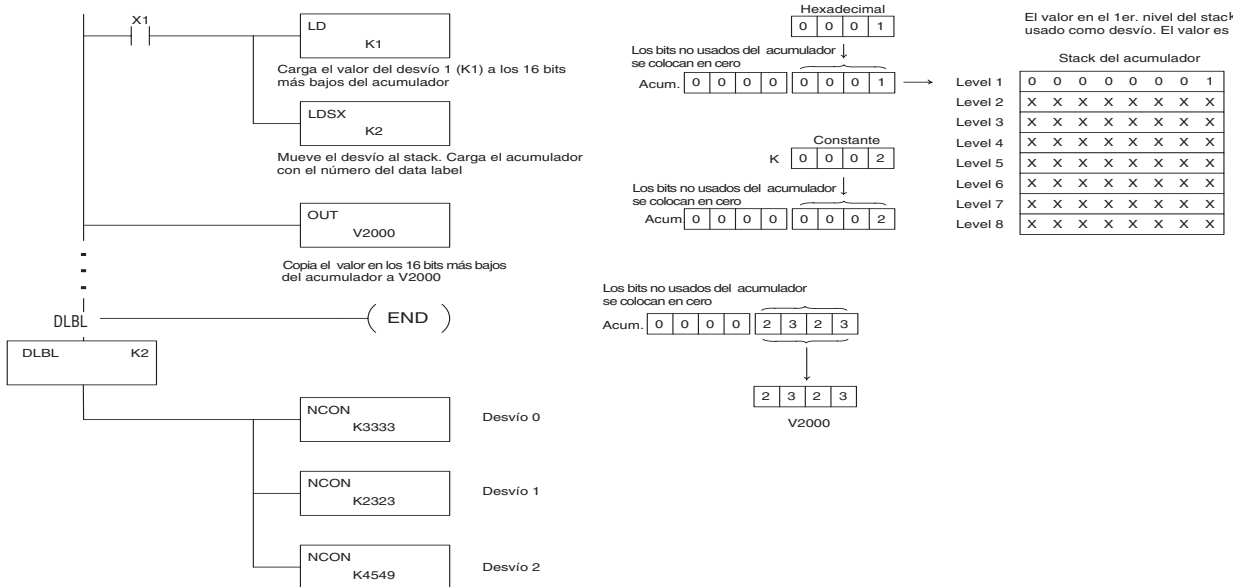
Tipo de operando de datos	Rango del DL06
Constante K	aaa 1-FFFF

Indicadores	Descripción
SP53	Está ON cuando el puntero está fuera del rango disponible.
SP70	ON si el valor en el acumulador por cualquier instrucción es negativo.
SP76	ON cuando cualquier instrucción carga un valor 0 al acumulador.



NOTA: Dos instrucciones consecutivas LDSX colocan el contenido de la primera instrucción en el stack del acumulador.

En el ejemplo siguiente cuando X1 está ON, se carga un desvío de 1 en el acumulador. Este valor se colocará en el primer nivel del Stack del acumulador cuando se ejecuta la instrucción LDSX. La instrucción LDSX especifica el área DLBL K2 donde se encuentran las constantes numéricas en el programa y carga el valor constante, indicado por el valor de desvío en el Stack, en los 16 bits más bajos del acumulador.

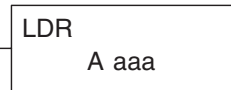


\$ STR	→	B 1	ENT	Programador D2-HPP						
SHFT	L ANDST	D 3	→	SHFT	K JMP	B 1	ENT			
SHFT	L ANDST	D 3	S RST	X SET	→	C 2	ENT			
SHFT	E 4	N TMR	D 3	ENT						
SHFT	D 3	L ANDST	B 1	L ANDST	→	C 2	ENT			
SHFT	N TMR	C 2	O INST#	N TMR	→	D 3	D 3	D 3	D 3	ENT
SHFT	N TMR	C 2	O INST#	N TMR	→	C 2	D 3	C 2	D 3	ENT
SHFT	N TMR	C 2	O INST#	N TMR	→	E 4	F 5	E 4	J 9	ENT
GX OUT	→	SHFT	V AND	C 2	A 0	A 0	A 0	ENT		

La instrucción Load Real Number (LDR)

DS5	Usado
HPP	N/A

La instrucción LDR carga un número real contenido en dos direcciones consecutivas de la memoria V o en una constante de 8 dígitos en el acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Real Constante R	-3.402823E+38 to + -3.402823E+38

Indicadores	Descripción
SP70	On anytime the value in the accumulator is negative.
SP76	On when any instrucción loads a value of zero into the accumulator.

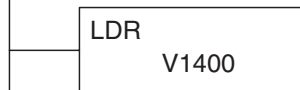
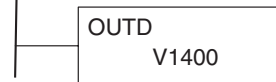
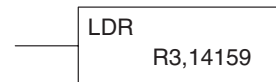
DirectSOFT le permite que entre los números reales directamente, usando una "R" como prefijo que indica un ingreso de número real. Usted puede entrar una constante tal como Pi(π), mostrado en el ejemplo a la derecha.

Para entrar números negativos, use un signo menos (-) después de la "R".

Para números muy grandes o números muy pequeños, se puede usar la notación exponencial. El número a la derecha es 5,3 millones. La instrucción OUTD lo copia a V1400 y V1401.

¡Estos números reales están en el formato de punto flotante IEEE de 32 bits, de modo que ocupan dos direcciones de memoria V, a pesar de que el número puede ser muy grande o pequeño! Si usted ve un número real almacenado en hexadecimal, binario o aún BCD, el número mostrado será muy difícil de descifrar. Así como todos los otros tipos de números, usted debe seguir las direcciones del número real en la memoria, de modo que puedan ser leídos en otra parte con las instrucciones apropiadas

El ejemplo previo encima almacenó un número real en V1400 y V1401. Suponga que ahora queremos recuperar ese número. Use solamente LDR con el tipo de datos V, como se muestra a la derecha. Luego podríamos realizar las operaciones matemáticas reales o convertirlo a un número binario.



La instrucción Out de bloque (OUT)

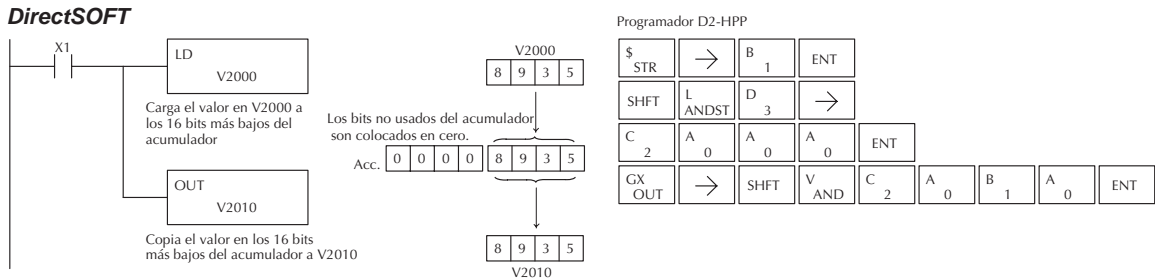
DS5	Usado
HPP	Usado

Es una instrucción de 16 bits que copia el valor en los 16 bits más bajos contenido en el acumulador a una localización especificada de memoria V (Aaaa).



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Indicadores	Descripción
SP53	ON si la CPU no puede resolver la lógica

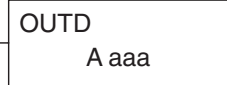
En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 se carga en los 16 bits más bajos del acumulador usando la instrucción LD. Luego se copia el valor en los 16 bits más bajos del acumulador a V2010 con la instrucción OUT de bloque.



La instrucción Out Double (OUTD)

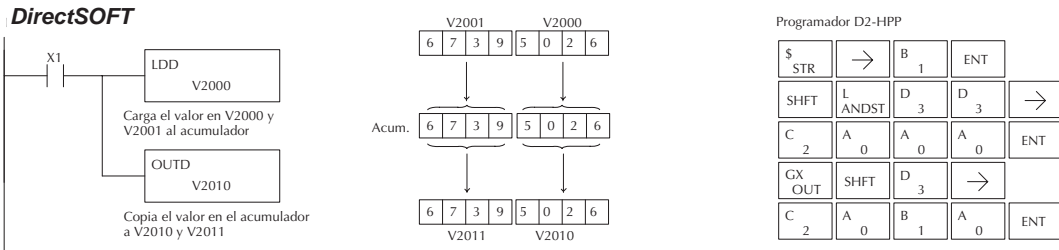
DS5	Usado
HPP	Usado

La instrucción OUT Doble es una instrucción de 32 bits que copia el valor en el acumulador a dos direcciones consecutivas de la memoria V en una localización (Aaaa) especificada.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Indicadores	Descripción
SP53	On if CPU cannot solve the logic.

En el ejemplo siguiente, cuándo X1 está ON, el valor de 32 bits en V2000 y V2001 se carga en el acumulador usando la instrucción LDD. El valor en el acumulador es colocado en V2010 y V2011 usando la instrucción OUTD.

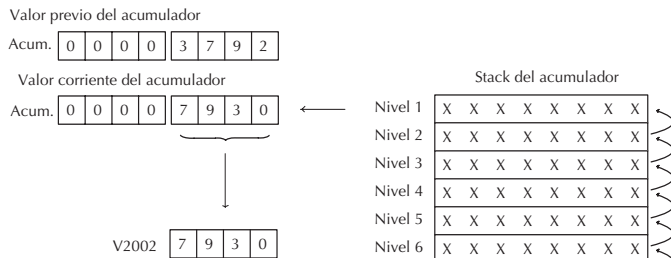
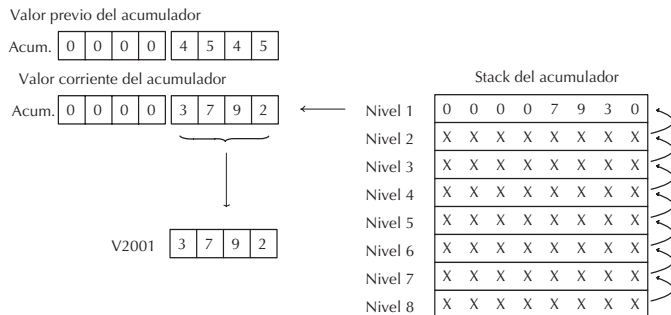
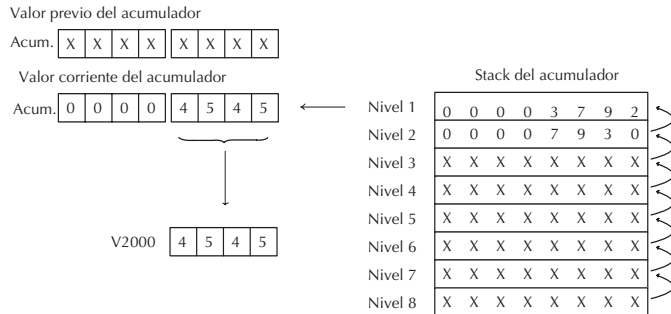
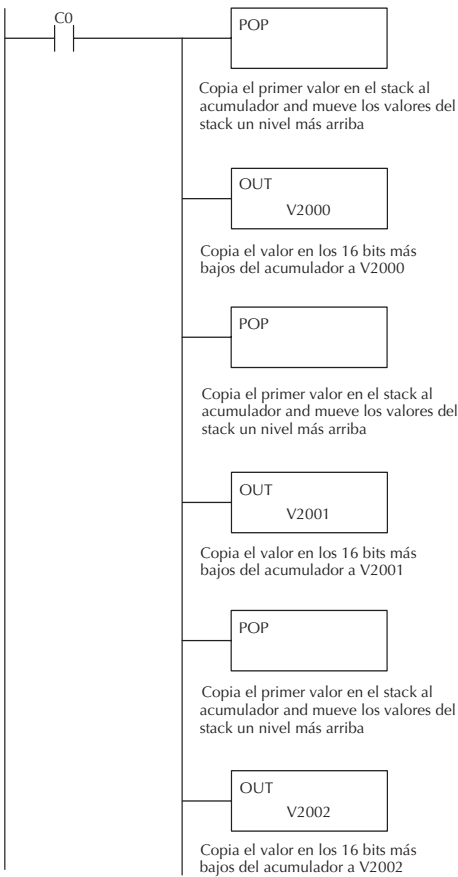


La instrucción Pop continuada

En el ejemplo siguiente, cuándo C0 está ON, el valor 4545 que estaba encima del Stack se mueve al acumulador usando la instrucción POP. El valor es copiado a V2000 usando la instrucción OUT. El próximo POP mueve el valor 3792 al acumulador y copia el valor a V2001. El último POP mueve el valor 7930 al acumulador y copia el valor a V2002 con la instrucción OUTD. Note que si el valor en el Stack usa más de 16 bits (4 dígitos) debe usarse la instrucción OUTD y deben ser asignadas 2 direcciones de memoria V para cada OUTD.

5

DirectSOFT



Programador D2-HPP

\$ STR	→	SHFT	C 2	A 0	ENT				
SHFT	P CV	SHFT	O INST#	P CV	ENT				
GX OUT	→	SHFT	V AND	C 2	A 0	A 0	A 0	ENT	
SHFT	P CV	SHFT	O INST#	P CV	ENT				
GX OUT	→	SHFT	V AND	C 2	A 0	A 0	B 1	ENT	
SHFT	P CV	SHFT	O INST#	P CV	ENT				
GX OUT	→	SHFT	V AND	C 2	A 0	A 0	C 2	ENT	

La instrucción Out Indexed (OUTX)

La instrucción OUTX es una instrucción de 16 bits. Copia un valor de 16 bits o de 4 dígitos desde el primer nivel del Stack del acumulador hasta una dirección cambiada por un número de desvío que es el valor en el acumulador (la memoria V + el desvío). Esta instrucción interpreta el valor del desvío como un número hexadecimal. Los 16 bits más altos del acumulador son forzados a 0.

DS5	Usado
HPP	Usado



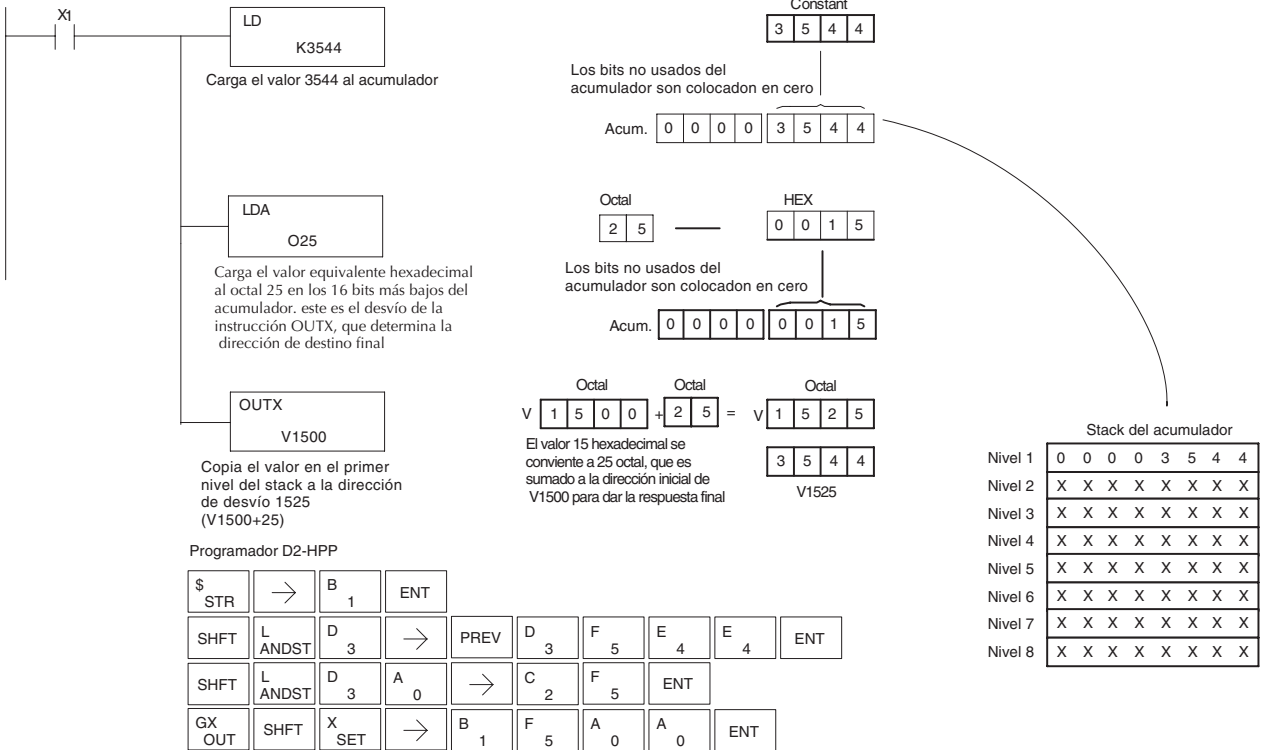
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero..... P	P
	Vea el mapa de memoria
	Vea el mapa de memoria
Indicadores	Descripción
SP53	ON si la CPU no puede resolver la lógica

5

En el ejemplo siguiente, cuándo X1 está ON, la constante 3544 es cargada al acumulador. Este es el valor que será copiado a la memoria V de destino con desvío (V1525). El valor 3544 será colocado en el Stack cuando se ejecuta la instrucción LDA. Recuerde, dos instrucciones consecutivas LDA colocan el valor de la primera instrucción LD en el Stack. La instrucción LDA convierte el valor 25 octal a 15 hexadecimal y coloca el valor en el acumulador.

La instrucción OUTX copia el valor 3544 que está en el primer nivel del Stack del acumulador a V1525.

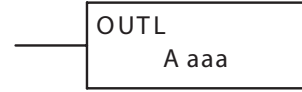
DirectSOFT



La instrucción Out Least (OUTL)

DS5	Usado
HPP	Usado

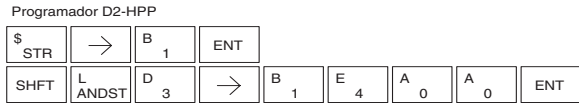
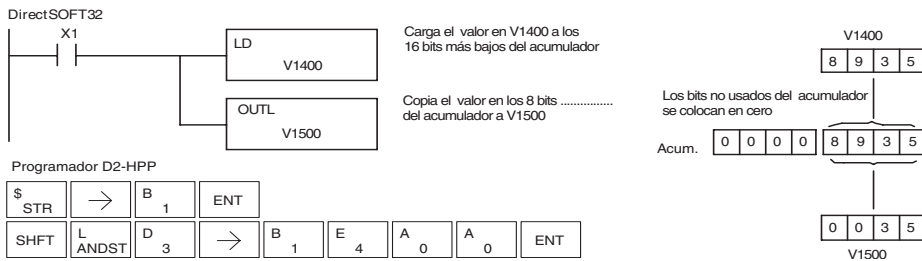
La instrucción OUTL copia el valor en los 8 bits más bajos del acumulador a los 8 bits más bajos de la memoria especificada (en otras palabras, copia el byte más bajo de la palabra más baja del acumulador).



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 se carga en los 16 bits más bajos del acumulador usando la instrucción LD. El valor en los 8 bits más bajos del acumulador es copiado a V1500 usando la instrucción OUTL.

DirectSOFT



La instrucción Out Most (OUTM)

DS5	Usado
HPP	Usado

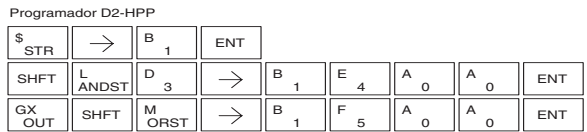
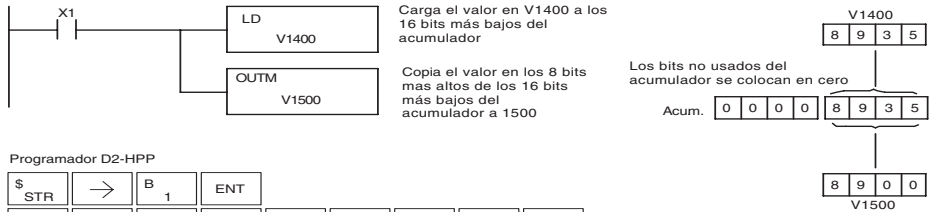
La instrucción OUTM copia el valor en los 8 bits más altos de la palabra más baja del acumulador a los 8 bits más altos de la memoria especificada (en otras palabras, copia el byte más alto de la palabra más baja del acumulador).



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 se carga en los 16 bits más bajos del acumulador usando la instrucción LD. El valor en los 8 bits más altos de los 16 bits más bajos del acumulador es copiado a V1500 usando la instrucción OUTM.

DirectSOFT

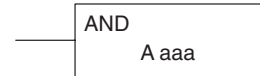


Las instrucciones lógicas con el acumulador

La instrucción And de bloque (AND)

DS5	Usado
HPP	Usado

La instrucción AND es una instrucción de 16 bits lógica que hace la función AND del valor en los 16 bits más bajos del acumulador con una localización especificada de memoria V (Aaaa). El resultado se va al acumulador. Una indicación discreta del estado con un relevador especial SP indica si el resultado es cero.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero..... P	

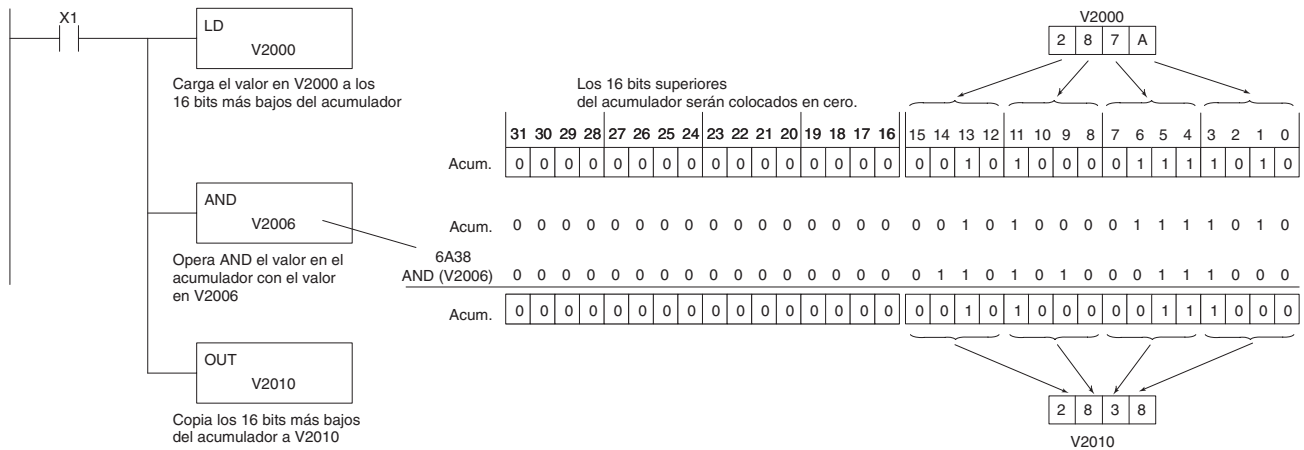
Indicadores	Descripción
SP63	Está ON si el resultado de la instrucción en el acumulador es 0.
SP70	ON cuando el valor en el acumulador es negativo.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 se carga en el acumulador usando la instrucción LD. El valor en el acumulador es operado AND con el valor en V2006 usando la instrucción AND. El valor en los 16 bits más bajos del acumulador es copiado a V2010 usando la instrucción OUT.

DirectSOFT



Programador D2-HPP

\$ STR	→	B 1	ENT					
SHFT	L ANDST	D 3	→	C 2	A 0	A 0	A 0	ENT
V AND	→	SHFT	V AND	C 2	A 0	A 0	G 6	ENT
GX OUT	→	SHFT	V AND	C 2	A 0	B 1	A 0	ENT

La instrucción And with Stack (ANDS)

DS5	Usado
HPP	Usado

La instrucción ANDS es una instrucción de 32 bits que hace la función lógica AND entre el valor en el acumulador con el valor del primer nivel del Stack del acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador se remueve del Stack y todos los valores son movidos para arriba un nivel. Indicaciones de estado discretas SP indican si el resultado del ANDS es cero o un número negativo (el bit más significativo está ON).



Indicadores	Descripción
SP63	ON si el resultado en el acumulador es 0.
SP70	ON si el resultado en el acumulador es negativo

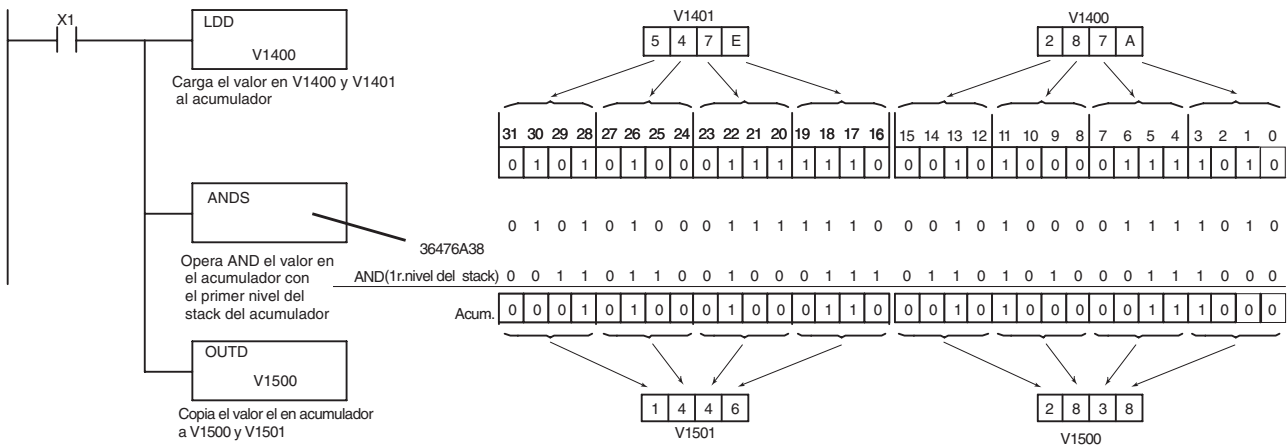
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

5



En el ejemplo siguiente cuando X1 está ON, el valor binario en el acumulador hace la función AND con el valor binario en el primer nivel del Stack del acumulador. El resultado se va al acumulador. El valor de 32 bits luego es copiado a V1500 y V1501.

DirectSOFT



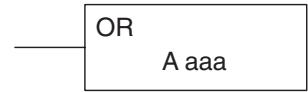
Programador D2-HPP

\$	STR	→	B	1	ENT									
SHFT	L	ANDST	D	3	→	B	1	E	4	A	0	A	0	ENT
V	AND	SHFT	S	RST	ENT									
GX	OUT	SHFT	D	3	→	B	1	F	5	A	0	A	0	ENT

La instrucción de bloque Or (OR)

DS5	Usado
HPP	Usado

La instrucción OR es una instrucción de 16 bits que hace la función lógica OR entre el valor en los 16 bits más bajos del acumulador con una localización especificada de memoria V (Aaaa). El resultado se va al acumulador. La indicación de estado discreta SP indica si el resultado de la función OR es cero.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V..... V	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria

Indicadores	Descripción
SP63	ON si el resultado en el acumulador es 0.
SP70	ON cuando el valor en el acumulador es negativo.

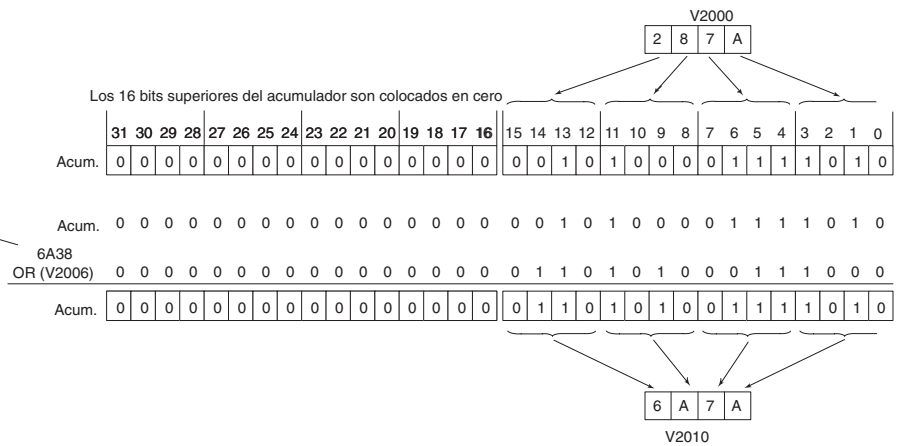
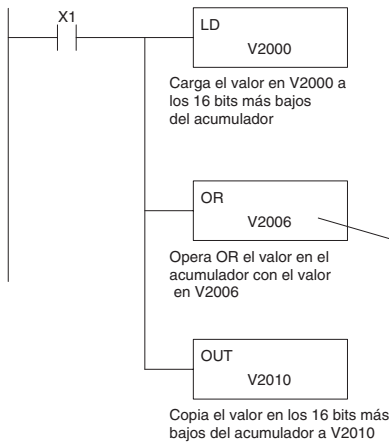
5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 se carga en el acumulador usando la instrucción LD. El valor en el acumulador es operado con V2006 usando la instrucción OR. El valor en los 16 bits más bajos del acumulador es copiado a V2010 usando la instrucción OUT.

DirectSOFT



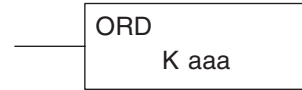
Programador D2-HPP

\$ STR	→	B 1	ENT						
SHFT	L ANDST	D 3	→	C 2	A 0	A 0	A 0	ENT	
Q OR	→	SHFT	V AND	C 2	A 0	A 0	G 6	ENT	
GX OUT	→	SHFT	V AND	C 2	A 0	B 1	A 0	ENT	

La instrucción Or Double (ORD)

DS5	Usado
HPP	Usado

ORD es una instrucción de 32 bits que hace la función OR entre el valor en el acumulador con el valor (Aaaa), que es dos direcciones consecutivas de memoria V o una constante de 8 dígitos (max.). El resultado se va al acumulador. Indicaciones de estado discretas SP indican si el resultado de la operación ORD es cero o un número negativo (el bit más significativo está ON).



Tipo de operando de datos	Rango del DL06
	aaa
Memoria V V	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria
Constante K	0-FFFFFFF

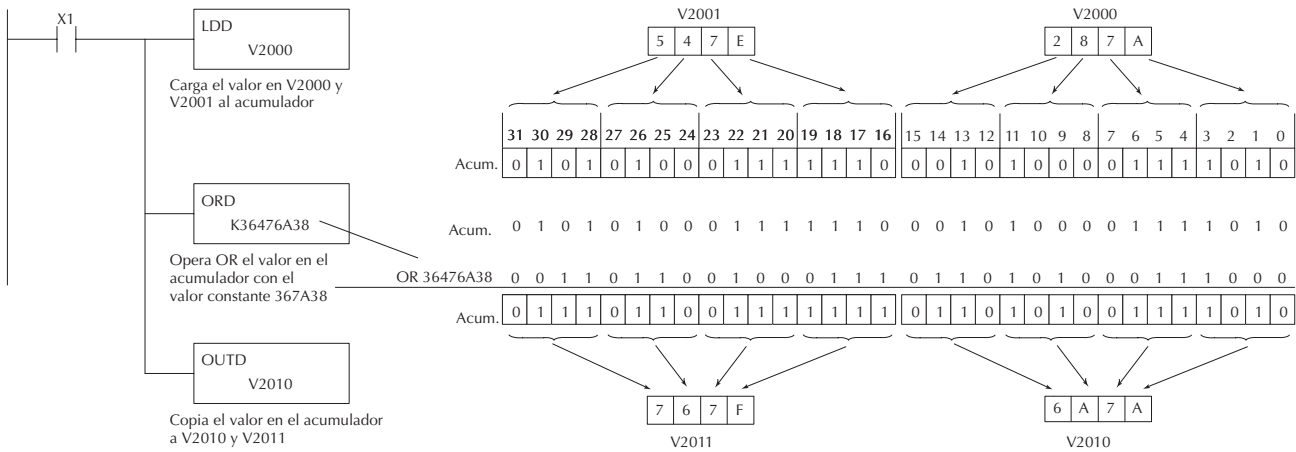
Indicadores	Descripción
SP63	ON si el resultado en el acumulador es 0.
SP70	ON si el resultado en el acumulador es negativo.



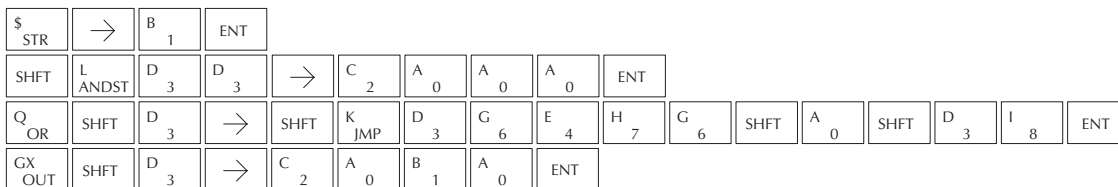
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 y V2001 se carga en el acumulador usando la instrucción LDD. El valor en el acumulador es operado OR con 36476A38 usando la instrucción ORD. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.

DirectSOFT



Programador D2-HPP



La instrucción Or with Stack (ORS)

DS5	Usado
HPP	Usado

La instrucción ORS es una instrucción de 32 bits que opera OR lógicamente el valor en el acumulador con el primer nivel del Stack del acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador se quita del Stack y todos los valores son movidos un nivel para arriba. Indicaciones de estado discretas SP indican si el resultado del OR con el Stack es cero o un número negativo (el bit más significativo está ON).



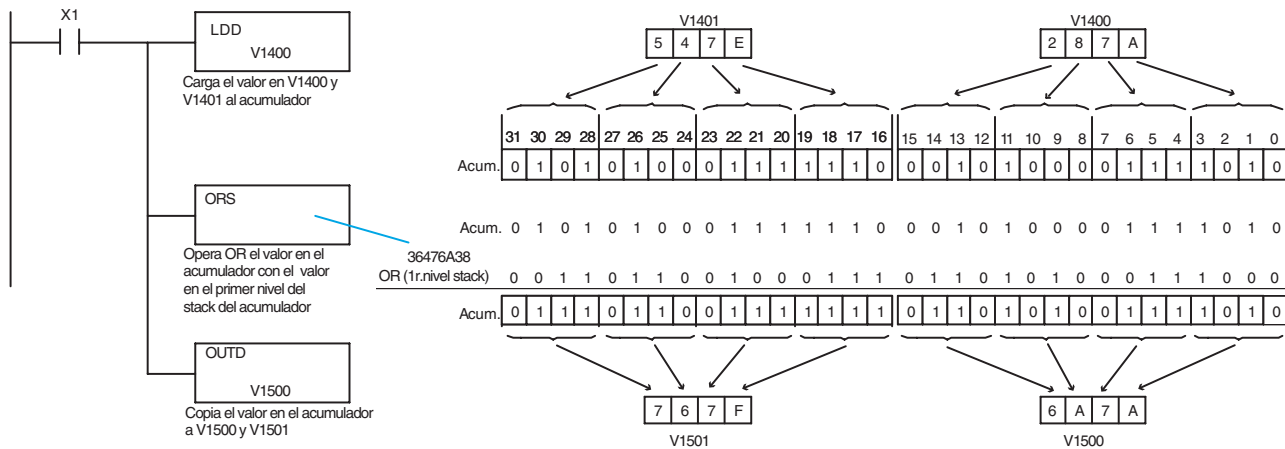
Indicadores	Descripción
SP63	ON si el resultado en el acumulador es 0.
SP70	ON cuando el valor en el acumulador es negativo.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente cuando X1 está ON, el valor binario en el acumulador será operado OR con el valor binario en el primer nivel del Stack. El resultado se va al acumulador.

DirectSOFT



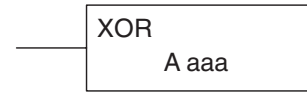
Programador D2-HPP

\$	STR	→	B	1	ENT									
SHFT	L	ANDST	D	3	→	B	1	E	4	A	0	A	0	ENT
Q	OR	SHFT	S	RST	ENT									
GX	OUT	SHFT	D	3	→	B	1	F	5	A	0	A	0	ENT

La instrucción Exclusive Or (XOR)

DS5	Usado
HPP	Usado

La instrucción XOR es una instrucción de 16 bits que realiza un OR exclusivo entre el valor en los 16 bits más bajos del acumulador y una localización especificada de memoria V (Aaaa). El resultado se va al acumulador. La indicación de estado discreta SP indica si el resultado del XOR es cero.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P

Indicadores	Descripción
SP63	ON si el resultado en el acumulador es 0.
SP70	ON cuando el valor en el acumulador es negativo.

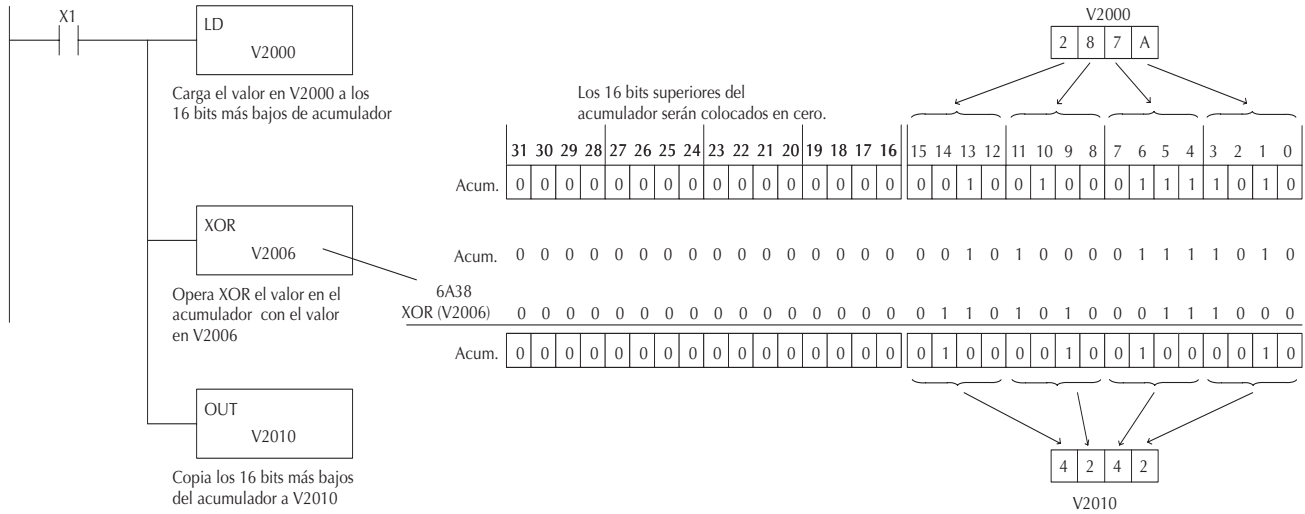
5



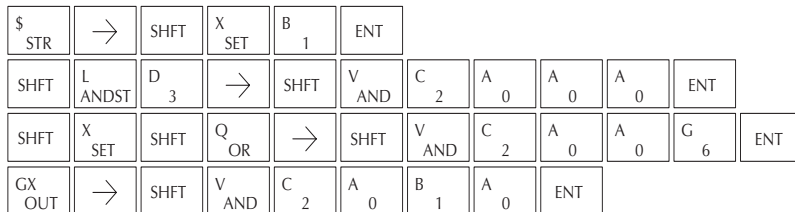
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 se carga en el acumulador usando la instrucción LD. El valor en el acumulador es operado con V2006 usando la instrucción XOR. El valor en los 16 bits más bajos del acumulador es copiado a V2010 usando la instrucción OUT.

DirectSOFT



Programador D2-HPP



La instrucción Exclusive Or Double (XORD)

DS5	Usado
HPP	Usado

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 se carga en el acumulador usando la instrucción LD. El valor en el acumulador es operado con V2006 usando la instrucción XOR, es decir, hace un OR exclusivo entre el acumulador y V2006. El valor en los 16 bits más bajos del acumulador es copiado a V2010 usando la instrucción OUT).



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero.....	P
Constante	K
	0-FFFFFFFF

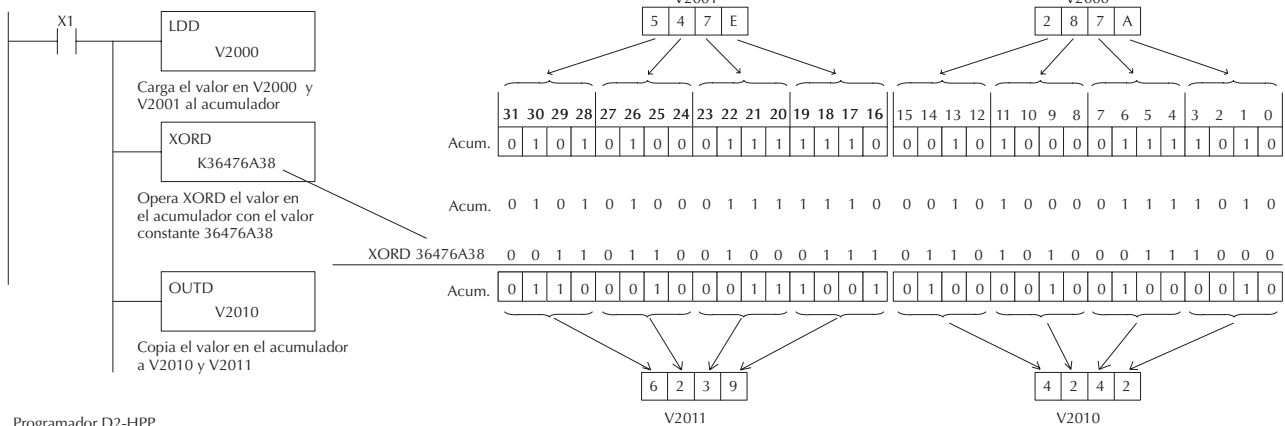
Indicadores	Descripción
SP63	ON si el resultado en el acumulador es 0.
SP70	ON cuando el valor en el acumulador es negativo.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 y V2001 se carga en el acumulador usando la instrucción LDD. El valor en el acumulador es operado con un OR exclusivo con 36476A38 usando la instrucción XORD. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.

DirectSOFT



Programador D2-HPP

\$	STR	→	B	1	ENT				
SHFT	L	D	D	→	C	A	A	A	ENT
	ANDST	3	3		2	0	0	0	
SHFT	X	Q	SHFT	D	→	SHFT	K		
	SET	OR		3		JMP			
D	G	E	H	G	SHFT	A	SHFT	D	I
3	6	4	7	6		0		3	8
ENT									
GX	SHFT	D	→	C	A	B	A		ENT
OUT		3		2	0	1	0		

La instrucción Exclusive Or with Stack (XORS)

DS5	Usado
HPP	Usado

La instrucción XORS es una instrucción de 32 bits que realiza un OR exclusivo del valor en el acumulador con el primer nivel del Stack del acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador se quita del Stack y todos los valores son movidos un nivel para arriba. Indicaciones de estado discretas SP indican si el resultado de la instrucción XORS es cero o un número negativo (el bit más significativo está ON). Recuerde que el stack se hace cero al fin de cada barrido.



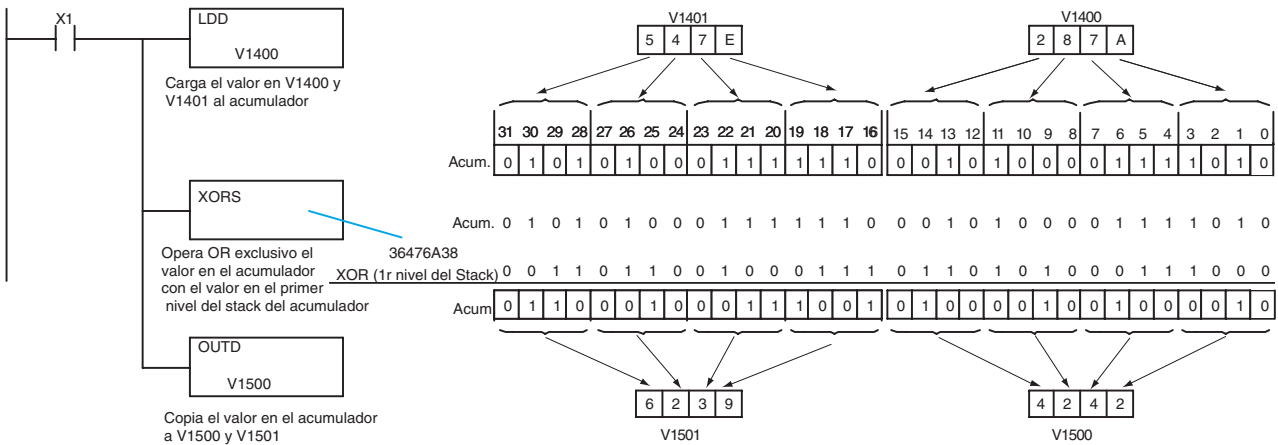
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

5

Indicadores	Descripción
SP63	ON si el resultado en el acumulador es cero.
SP70	ON si el resultado en el acumulador es negativo

En el ejemplo siguiente cuando X1 está ON, el valor binario en el acumulador será operado OR exclusivo con el valor binario en el primer nivel del Stack del acumulador. El resultado residirá en el acumulador. La instrucción OUTD copia el valor en el acumulador a V1500.

DirectSOFT



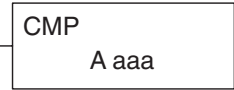
Programador D2-HPP

\$ STR	→	B 1	ENT						
SHFT	L ANDST	D 3	D 3	→	B 1	E 4	A 0	A 0	ENT
SHFT	X SET	Q OR	SHFT	S RST	ENT				
GX OUT	SHFT	D 3	→	B 1	F 5	A 0	A 0	ENT	

La instrucción Compare (CMP)

DS5	Usado
HPP	Usado

La instrucción CMP es una instrucción de 16 bits que compara el valor en los 16 bits más bajos del acumulador con el valor en una localización especificada de memoria V (Aaaa). La indicación SP correspondiente del estado será prendida indicando el resultado de la comparación.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero.....	P
	Vea el mapa de memoria
	Vea el mapa de memoria

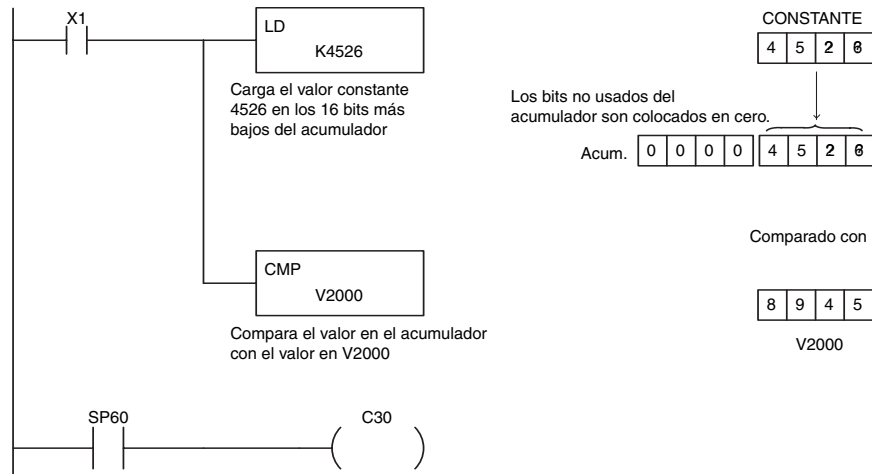
Indicadores	Descripción
SP60	ON si el resultado en el acumulador es menor que el valor de la instrucción.
SP61	ON si el resultado en el acumulador es igual al valor de la instrucción.
SP62	ON si el resultado en el acumulador es mayor que el valor de la instrucción.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente cuando X1 está ON, la constante 4526 es cargada en los 16 bits más bajos del acumulador usando la instrucción LD. El valor en el acumulador es comparado con el valor BCD en V2000 usando la instrucción CMP. La indicación SP correspondiente del estado será prendida indicando el resultado de la comparación. En este ejemplo, si el valor en el acumulador es menor que el valor especificado en la instrucción CMP, SP60 prenderá activando C30.



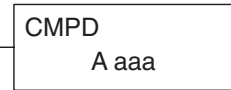
Programador D2-HPP

\$ STR	→	B 1	ENT							
SHFT	L ANDST	D 3	→	SHFT	K JMP	E 4	F 5	C 2	G 6	ENT
SHFT	C 2	SHFT	M ORST	P CV	→	C 2	A 0	A 0	A 0	ENT
\$ STR	→	SHFT	SP STRN	G 6	A 0	ENT				
GX OUT	→	SHFT	C 2	D 3	A 0	ENT				

La instrucción Compare Double (CMPD)

DS5	Usado
HPP	Usado

La instrucción CMPD es una instrucción de 32 bits que compara el valor en el acumulador con el valor (Aaaa), que es dos direcciones consecutivas de memoria V o una constante de 8 dígitos (máximo). La indicación SP correspondiente del estado será activada indicando el resultado de la comparación.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0-FFFFFFF

Indicadores	Descripción
SP60	ON si el resultado en el acumulador es menor que el valor de la instrucción.
SP61	ON si el resultado en el acumulador es igual al valor de la instrucción.
SP62	ON si el resultado en el acumulador es mayor que el valor de la instrucción.

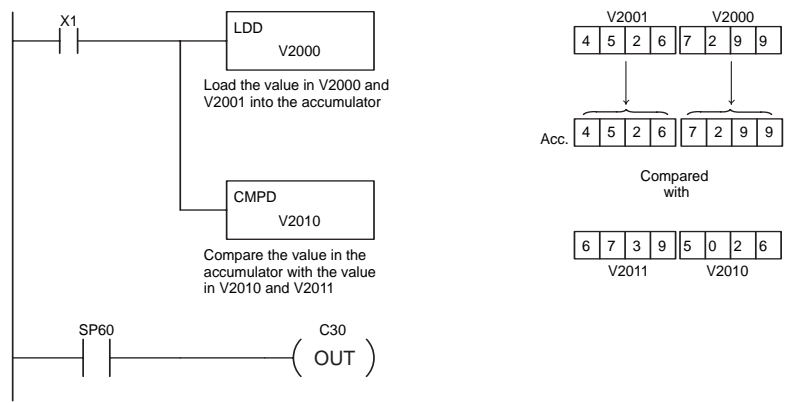
5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente cuando X1 está ON, el valor en V2000 y V2001 se carga al acumulador usando la instrucción LDD. El valor en el acumulador es comparado con el valor en V2010 y V2011 usando la instrucción CMPD. La indicación SP correspondiente del estado será prendida indicando el resultado de la comparación.

En este ejemplo, si el valor en el acumulador es menor que el valor especificado en la instrucción, SP60 prenderá activando C30.



Handheld Programmer Keystrokes

\$ STR	→	B 1	ENT											
SHFT	L ANDST	D 3	D 3	→	C 2	A 0	A 0	A 0	ENT					
SHFT	C 2	SHFT	M ORST	P CV	D 3	→	C 2	A 0	B 1	A 0	ENT			
\$ STR	→	SHFT	SP STRN	G 6	A 0	ENT								
GX OUT	→	SHFT	C 2	D 3	A 0	ENT								

La instrucción Compare Formatted (CMPF)

DS5	Usado
HPP	Usado

La instrucción CMPF compara el valor en el acumulador con un número especificado de bits consecutivos (1-32). La instrucción requiere una localización (Aaaa) de inicio y el número de bits (Kbbb) a ser comparado. La indicación correspondiente del estado SP será prendida indicando el resultado de la comparación.



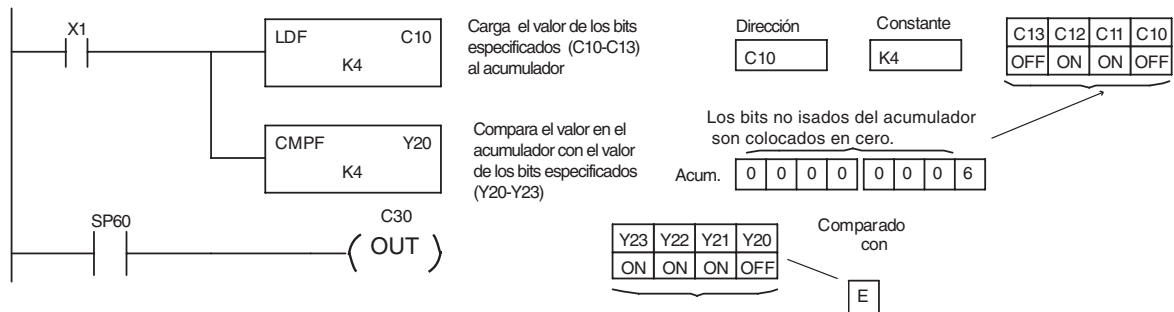
Tipo de operando de datos	Rango del DL06	
..... A/B	aaa	bbb
Entradas X	0-777	-
Salidas Y	0-777	-
Relevadores de control C	0-1777	-
Bits de etapas S	0-1777	-
Bits de temporizadores T	0-377	-
Bits de contadores CT	0-177	-
Relevadores especiales SP	0-777	-
Constante K	-	1-32

Indicadores	Descripción
SP60	ON si el resultado en el acumulador es menor que el valor de la instrucción..
SP61	ON si el resultado en el acumulador es igual al valor de la instrucción.
SP62	ON si el resultado en el acumulador es mayor que el valor de la instrucción.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON la instrucción LDF carga el valor binario de C10-C13 en el acumulador. La instrucción CMPF compara el valor en el acumulador al valor en Y20-Y23 (hexadecimal E). La indicación SP correspondiente del estado será prendida indicando el resultado de la comparación. . En este ejemplo, si el valor en el acumulador es menor que el valor especificado en la instrucción, SP60 prenderá activando C30.



La instrucción Compare with Stack (CMPS)

DS5	Usado
HPP	Usado

La instrucción CMPS es una instrucción de 32 bits que compara el valor en el acumulador con el valor en el primer nivel del Stack del acumulador. La indicación correspondiente del estado SP será prendida indicando el resultado de la comparación. Esto no afecta el valor en el acumulador. Recuerde que el stack se hace 0 al fin de cada barrido.

CMPS

Indicadores	Descripción
SP60	ON si el resultado en el acumulador es menor que el valor de la instrucción.
SP61	ON si el resultado en el acumulador es igual que el valor de la instrucción.
SP62	ON si el resultado en el acumulador es mayor que el valor de la instrucción.

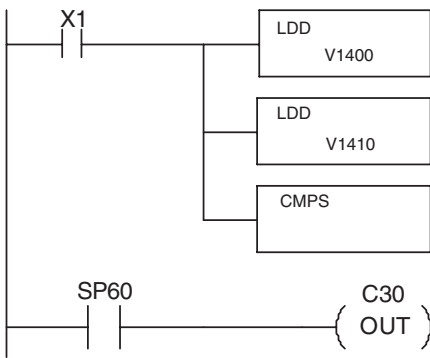
5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente cuando X1 está ON, el valor en V1400 y V1401 se carga en el acumulador usando la instrucción LDD. El valor en V1410 y V1411 se carga en el acumulador usando la instrucción LDD. El valor que se cargó en el acumulador desde V1400 y V1401 se coloca en el primer nivel del Stack cuando la segunda instrucción LDD es ejecutada. El valor en el acumulador es comparado con el valor en el primer nivel del Stack del acumulador usando la instrucción CMPS. La indicación SP correspondiente del estado será prendida indicando el resultado de la comparación. En este ejemplo, si el valor en el acumulador es menor que el valor en el Stack, SP60 prenderá, activando C30.

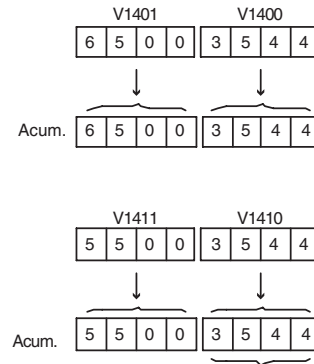
DirectSOFT



Carga el valor en V1400 y V1401 al acumulador

Carga el valor en V1410 y V1411 al acumulador

Compara el valor en el acumulador con el valor en el primer nivel del stack del acumulador



Comparado con el primer nivel del stack

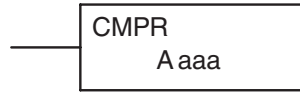
Programador D2-HPP

\$ STR	→	B 1	ENT						
SHFT	L ANDST	D 3	D 3	→	B 1	E 4	A 0	A 0	ENT
SHFT	L ANDST	D 3	D 3	→	B 1	E 4	B 1	A 0	ENT
SHFT	C 2	SHFT	M ORST	P CV	S RST	ENT			
\$ STR	PREV	G 6	A 0	ENT					
GX OUT	→	NEXT	NEXT	NEXT	SHFT	C 2	D 3	A 0	ENT

La instrucción Compare Real Number (CMPR)

DS5	Usado
HPP	Usado

La instrucción CMPR compara un valor del número real en el acumulador con dos direcciones consecutivas de memoria V que contienen un número real. La indicación correspondiente del estado SP será prendida indicando el resultado de la comparación. Ambos números a ser comparados tienen 32 bits.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante R	-3.402823E+ 038 hasta + -3.402823E+ 038

Indicadores	Descripción
SP60	ON si el resultado en el acumulador es menor que el valor de la instrucción..
SP61	ON si el resultado en el acumulador es igual que el valor de la instrucción.
SP62	ON cuando el valor en el acumulador es mayor que el valor de la instrucción.
SP71	ON en cualquier momento que la memoria V especificada por un puntero (P) no es válida

5



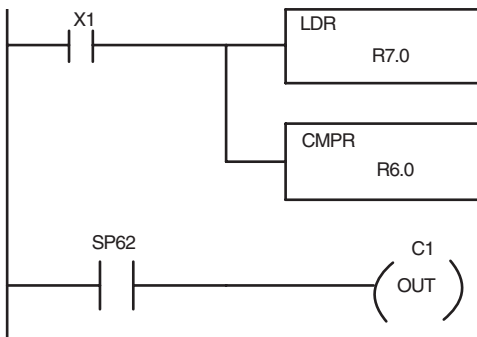
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.



NOTA: El número real no es absolutamente preciso; permite un rango desde negativo hasta positivo, pero no es muy preciso ya que solo representa 23 bits de resolución.

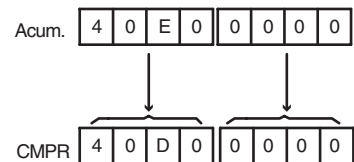
En el ejemplo siguiente cuando X1 está ON, la instrucción LDR carga la representación real del número 7,0 en el acumulador. La instrucción CMPR compara el contenido del acumulador con la representación real del número 6,0. Ya que 7 > 6, la indicación discreta SP correspondiente del estado es activada (el relevador especial SP62) activando el relevador de control C1.

DirectSOFT



Carga la representación del número real del decimal 7 al acumulador

Compara el valor con la representación del número real del decimal 6

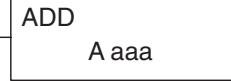


Instrucciones aritméticas

La instrucción Add (ADD)

DS5	Usado
HPP	Usado

ADD es una instrucción de 16 bits que suma un valor BCD en el acumulador con un valor BCD en una dirección de memoria V (Aaaa). **No se puede usar una constante K como parámetro en la instrucción.** El resultado se va al acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria

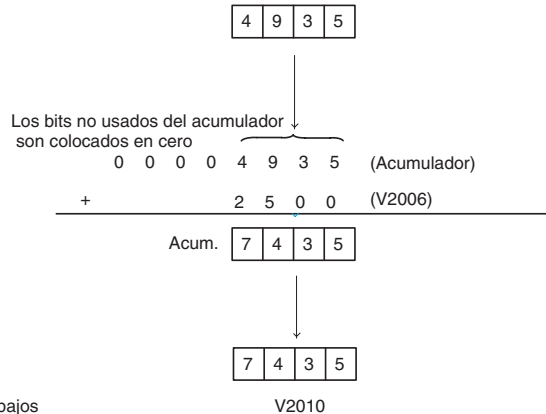
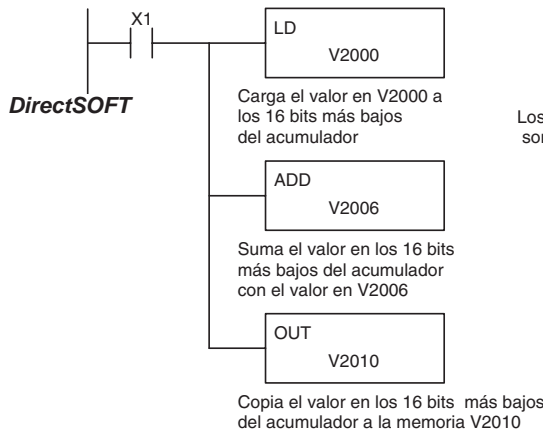
Indicadores	Descripción
SP63	ON cuando el resultado de la operación causa que el valor en el acumulador sea cero.
SP66	ON cuando el resultado de la operación de 16 bits resulta en un "pasa para" .
SP67	ON cuando el resultado de la operación de 32 bits resulta en un "pasa para" .
SP70	ON en cualquier momento que el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra uno de tipo diferente.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V2000 en el acumulador usando la instrucción LD. El valor en los 16 bits más bajos del acumulador es sumado al valor en V2006 usando la instrucción ADD. El valor en el acumulador es copiado a V2010 usando la instrucción OUT.



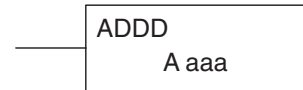
Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	→	C	2	A	0	A	0	A	0	ENT		
SHFT	A		D	3	D	3	→	C	2	A	0	A	0	G	6	ENT
GX	OUT	→	SHFT	V	AND	C	2	A	0	B	1	A	0	ENT		

La instrucción Add Double (ADDD)

DS5	Usado
HPP	Usado

ADDD es una instrucción de 32 bits que suma el valor BCD en el acumulador con un valor BCD (Aaaa), que son 2 direcciones consecutivas de memoria V o una constante de 8 dígitos (max) BCD. El resultado se va al acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero.....	P
Constante	K
	0-99999999

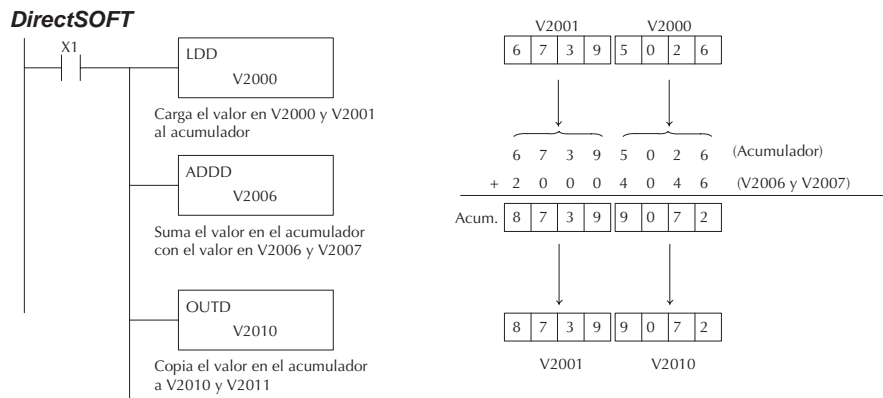
Indicadores	Descripción
SP63	ON cuando el resultado de la operación causa que el valor en el acumulador sea cero.
SP66	ON cuando el resultado de la operación de 16 bits resulta en un "pasa para" .
SP67	ON cuando el resultado de la operación de 32 bits resulta en un "pasa para".
SP70	ON en cualquier momento que el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra uno de tipo diferente.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 y V2001 se carga en el acumulador usando la instrucción LDD. El valor en el acumulador se suma con el valor en V2006 y V2007 usando la instrucción ADDD. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.



Programador D2-HPP

\$	STR	→	B	1	ENT													
SHFT	L	ANDST	D	3	D	3	→	C	2	A	0	A	0	A	0	ENT		
SHFT	A		D	3	D	3	D	3	→	C	2	A	0	A	0	G	6	ENT
GX	OUT	SHFT	D	3	→	SHFT	V	AND	C	2	A	0	B	1	A	0	ENT	

La instrucción Add Real (ADDR)

DS5	Usado
HPP	Usado

La instrucción ADDR suma un número real en el acumulador con una constante real o un número real que ocupa dos direcciones consecutivas de memoria V. El resultado se va al acumulador. Ambos números deben estar de acuerdo al formato de punto flotante IEEE de 32 bits.

ADDR
A aaa

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero. P	Vea el mapa de memoria
Constante R	-3.402823E+ 38 to + -3.402823E+ 38

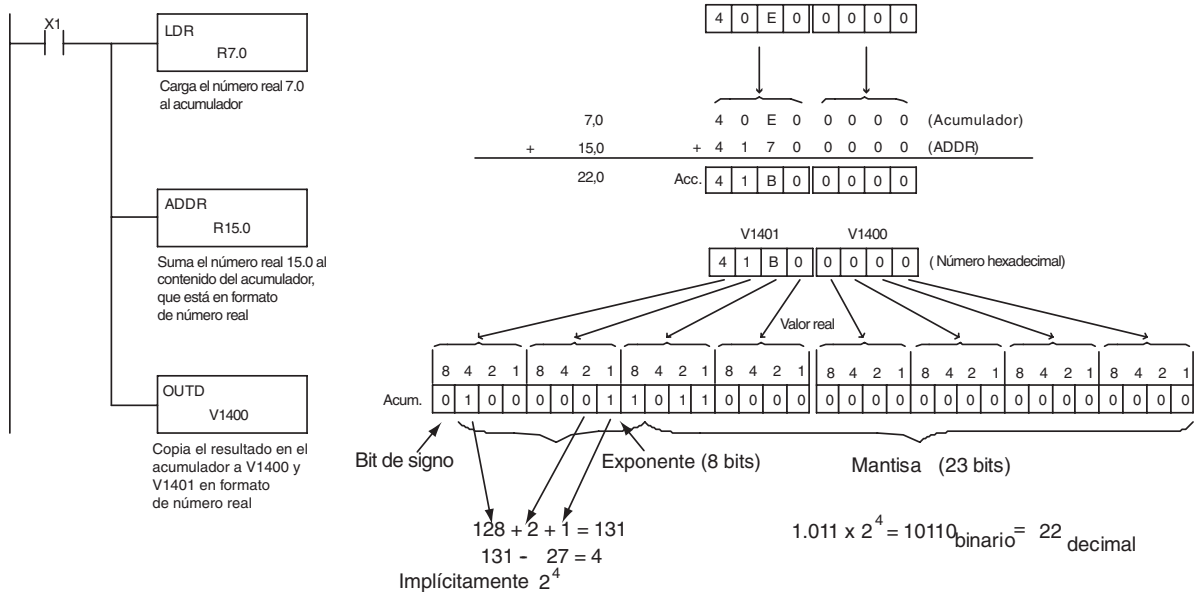
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP71	ON en cualquier momento que la memoria V especificada por un puntero (P) no es válida.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP73	ON cuando una suma o sustracción con signo da como resultado un bit de signo incorrecto.
SP74	ON cuando una operación de punto flotante resulta en un error underflow.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

Este ejemplo muestra la convención de punto flotante IEEE de 32 bits

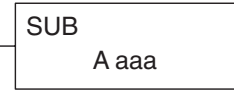


NOTA: El programador D2-HPP no permite entrar números reales con conversión automática al formato IEEE de 32 bits. Ud debe utilizar DirectSOFT en este caso, para usar esta función.

La instrucción Subtract (SUB)

DS5	Usado
HPP	Usado

SUB (Resta) es una instrucción de 16 bits que resta el valor BCD (Aaaa) en una dirección de memoria V del valor BCD en los 16 bits más bajos del acumulador. El resultado se va al acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de resta de 16 bits pide un "préstamo".
SP65	ON cuando la instrucción de resta de 32 bits pide un "préstamo".
SP70	ON cuando el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

5

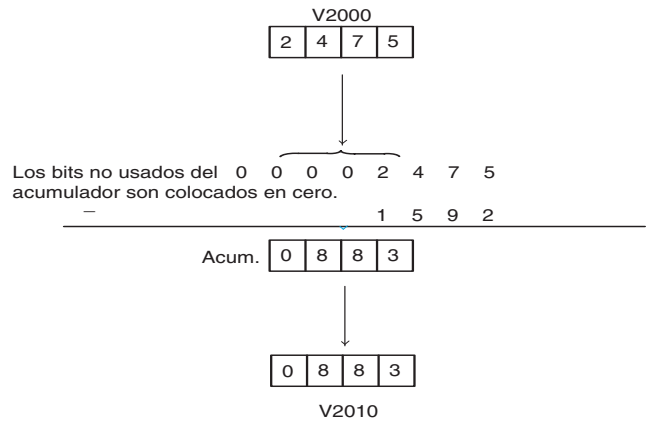
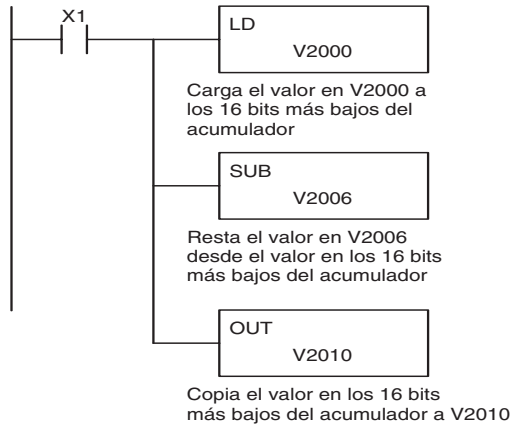


NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

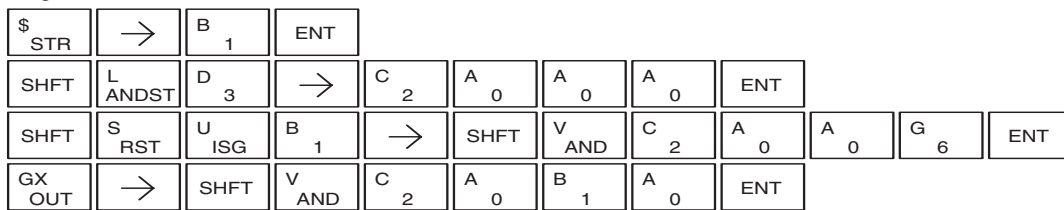
En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 se carga en el acumulador usando la instrucción LD.

El valor en V2006 se resta del valor en el acumulador usando la instrucción SUB. El valor en el acumulador es copiado a V2010 usando la instrucción OUT.

DirectSOFT



Programador D2-HPP



La instrucción Subtract Double (SUBD)

DS5	Usado
HPP	Usado

Resta Doble SUBD es una instrucción de 32 bits que resta el valor BCD (Aaaa), que puede ser 2 direcciones consecutivas de memoria V o una constante de 8 dígitos (máximo), desde el valor BCD en el acumulador.

SUBD
A aaa

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0-99999999

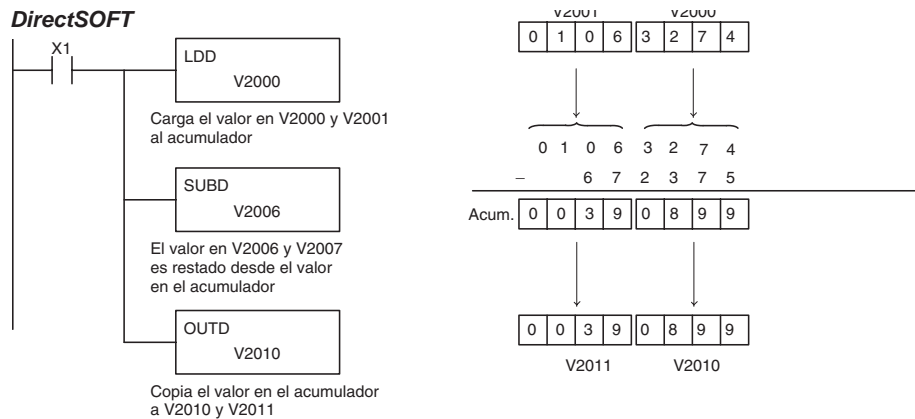
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de resta de 16 bits pide un "préstamo".
SP65	ON cuando la instrucción de resta de 32 bits pide un "préstamo".
SP70	ON cuando el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V2000 y V2001 en el acumulador usando la instrucción LDD. El valor en V2006 y V2007 se resta del valor en el acumulador. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.



Programador D2-HPP

\$	STR	→	B	1	ENT										
SHFT	L	D	D	→	C	A	A	A	ENT						
SHFT	S	SHFT	U	B	D	→	C	A	A	G	ENT				
GX	OUT	SHFT	D	→	C	A	B	A	ENT						

La instrucción Subtract Real (SUBR)

DS5	Usado
HPP	N/A

La instrucción SUBR resta un número real en el acumulador de una constante real o un número real que ocupa 2 direcciones consecutivas de memoria V. El resultado se va al acumulador. Ambos números deben seguir el formato de punto flotante IEEE de 32 bits.

SUBR
A aaa

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante R	-3.402823E + 38 hasta +3.402823E + 38

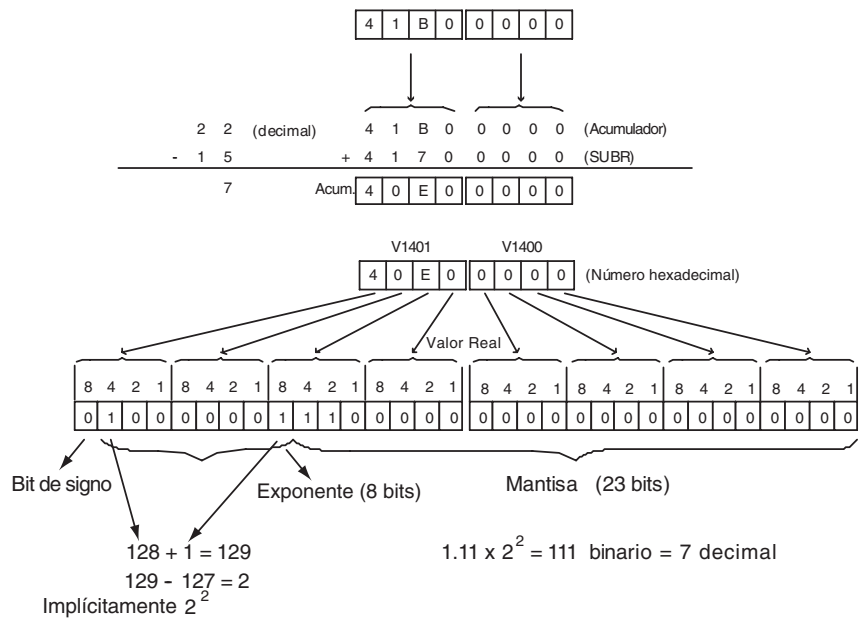
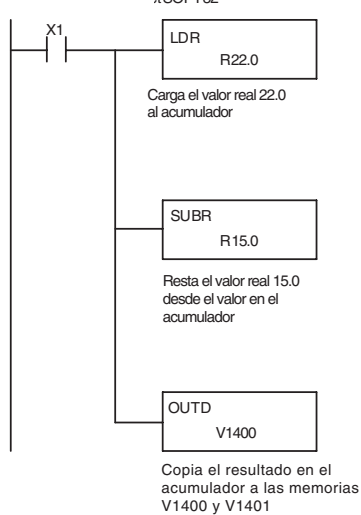
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP71	ON en cualquier momento que la memoria V especificada por un puntero (P) no es válida.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP73	ON cuando una suma o sustracción con signo da como resultado un bit de signo incorrecto.
SP74	On cuando una operación de punto flotante resulta en un error de underflow.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT

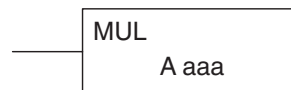


NOTA: El programador D2-HPP no permite entrar números reales con conversión automática al formato IEEE de 32 bits. Ud debe usar DirectSOFT en este caso, para usar esta función.

La instrucción Multiply (MUL)

DS5	Usado
HPP	Usado

MUL es una instrucción de 16 bits que multiplica el valor BCD (Aaaa), que es una dirección de memoria V o una constante de 4 dígitos (max.) por el valor BCD en los 16 bits más bajos del acumulador. El resultado puede ser de hasta 8 dígitos y se va al acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P
Constante	K
	0-9999

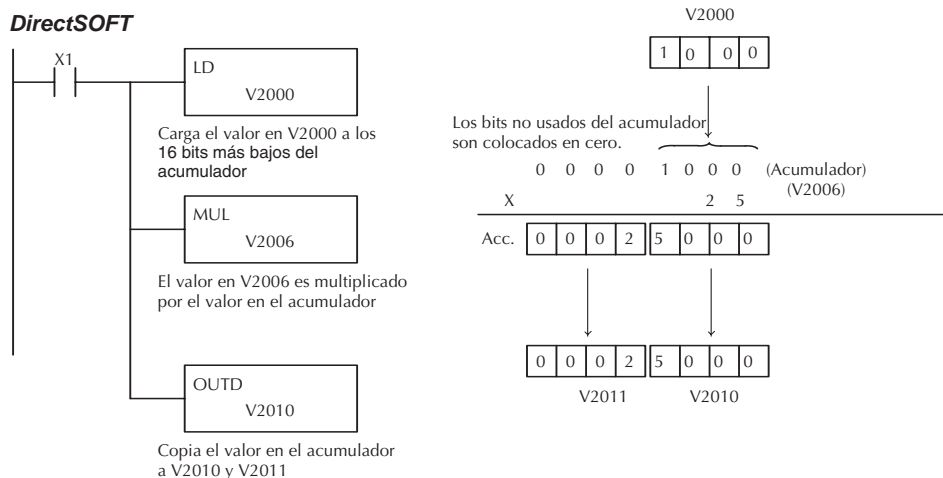
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V2000 al acumulador usando la instrucción LD. El valor en V2006 es multiplicado por el valor en el acumulador. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.



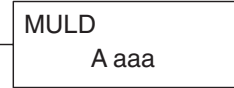
Programador D2-HPP

\$ STR	→	B 1	ENT						
SHFT	L ANDST	D 3	→	C 2	A 0	A 0	A 0	ENT	
SHFT	M ORST	U ISG	L ANDST	→	C 2	A 0	A 0	G 6	ENT
GX OUT	SHFT	D 3	→	C 2	A 0	B 1	A 0	ENT	

La instrucción Multiply Double (MULD)

DS5	Usado
HPP	Usado

MULD es una instrucción de 32 bits que multiplica el valor de 8 dígitos BCD en el acumulador por el valor de 8 dígitos BCD en 2 direcciones consecutivas de memoria V especificadas en la instrucción. Los 8 dígitos más bajos del resultado se van al acumulador. Los dígitos superiores del resultado se van al Stack del acumulador.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

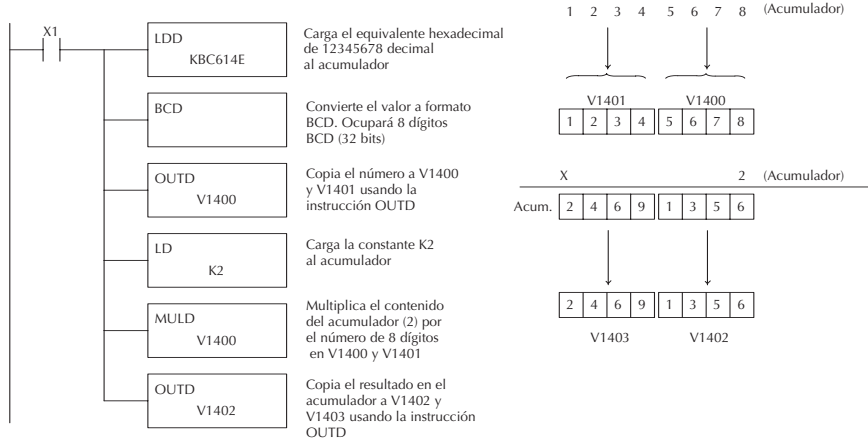
5



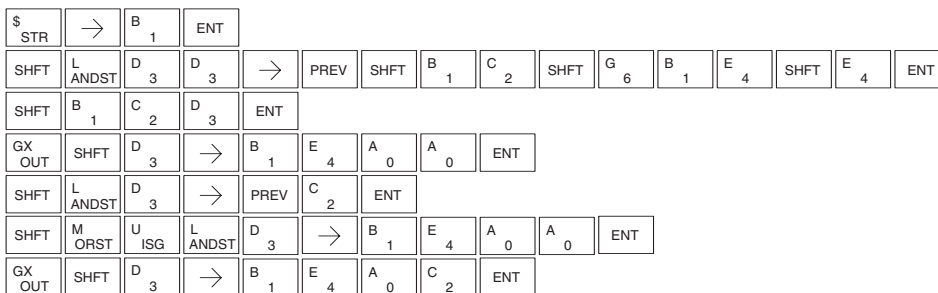
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, la constante hexadecimal Kbc614e se carga en el acumulador. Cuándo es convertido a BCD el número es "12345678". Esos números se almacenan en V1400 y V1401. Después de cargar la constante K2 en el acumulador, se multiplica por 12345678, que es 24691356.

DirectSOFT



Programador D2-HPP



La instrucción Multiply Real (MULR)

DS5	Usado
HPP	Usado

La instrucción MULR multiplica un número real en el acumulador con una constante real o un número real que ocupa dos direcciones consecutivas de memoria V. El resultado se va al acumulador. Ambos números deben estar de acuerdo al formato de punto flotante IEEE.

MULR
A aaa

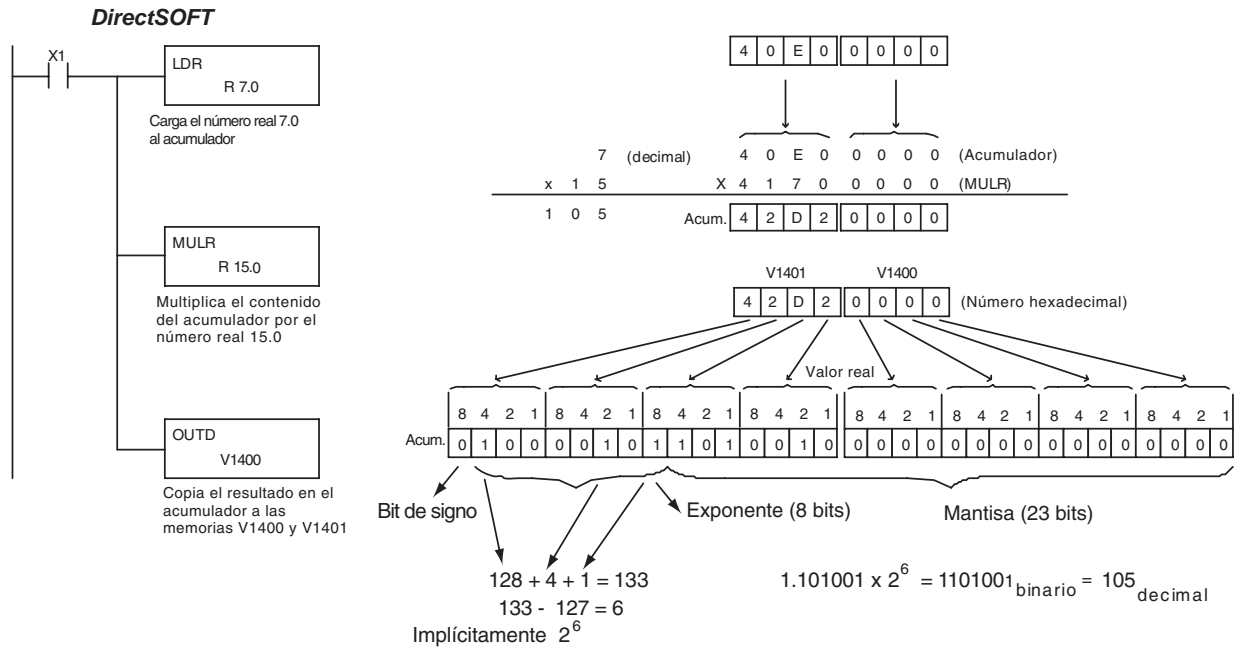
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante real R	-3.402823E +38 to + -3.402823E +38

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP71	ON en cualquier momento que la memoria V especificada por un puntero (P) no es válida.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP73	ON cuando una suma o sustracción con signo da como resultado un bit de signo incorrecto.
SP74	On cuando una operación de punto flotante resulta en un error de underflow.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

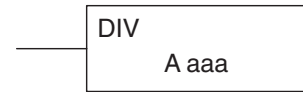


NOTE: The current HPP does not support real number entry with automatic conversion to the 32-bit IEEE format. You must use DirectSOFT for this feature.

La instrucción Divide (DIV)

DS5	Usado
HPP	Usado

DIV es una instrucción de 16 bits que divide el valor BCD en el acumulador por un valor BCD (Aaaa), que es una localización de memoria V o una constante de 4 dígitos (max.) La primera parte del cociente se va al acumulador y el resto se va al primer nivel del Stack.



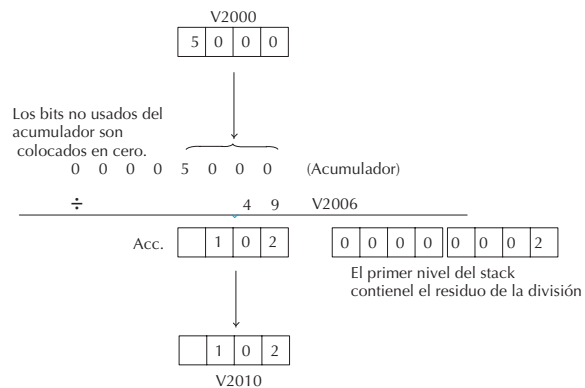
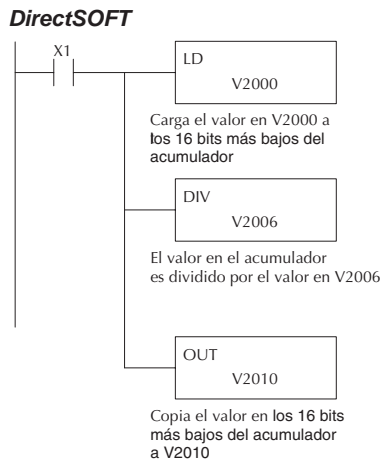
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0-9999

Indicadores	Descripción
SP53	ON cuando el valor del operando es más grande de lo que puede aceptar el acumulador.
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V2000 al acumulador usando la instrucción LD. El valor en el acumulador será dividido por el valor en V2006 usando la instrucción DIV. El valor en el acumulador es copiado a V2010 usando la instrucción OUT.



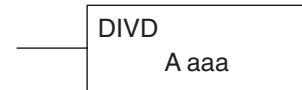
Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	→	C	2	A	0	A	0	A	0	ENT		
SHFT	D	3	I	8	V	AND	→	C	2	A	0	A	0	G	6	ENT
GX	OUT	→	SHFT	V	AND	C	2	A	0	B	1	A	0	ENT		

La instrucción Divide Double (DIVD)

DS5	Usado
HPP	Usado

DIVD es una instrucción de 32 bits que divide el valor BCD en el acumulador por un valor BCD (Aaaa), que se debe obtener de 2 direcciones consecutivas de memoria V. (No se puede usar una constante como el parámetro de la instrucción) La primera parte del cociente se va al acumulador y el resto se va al primer nivel del Stack.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria

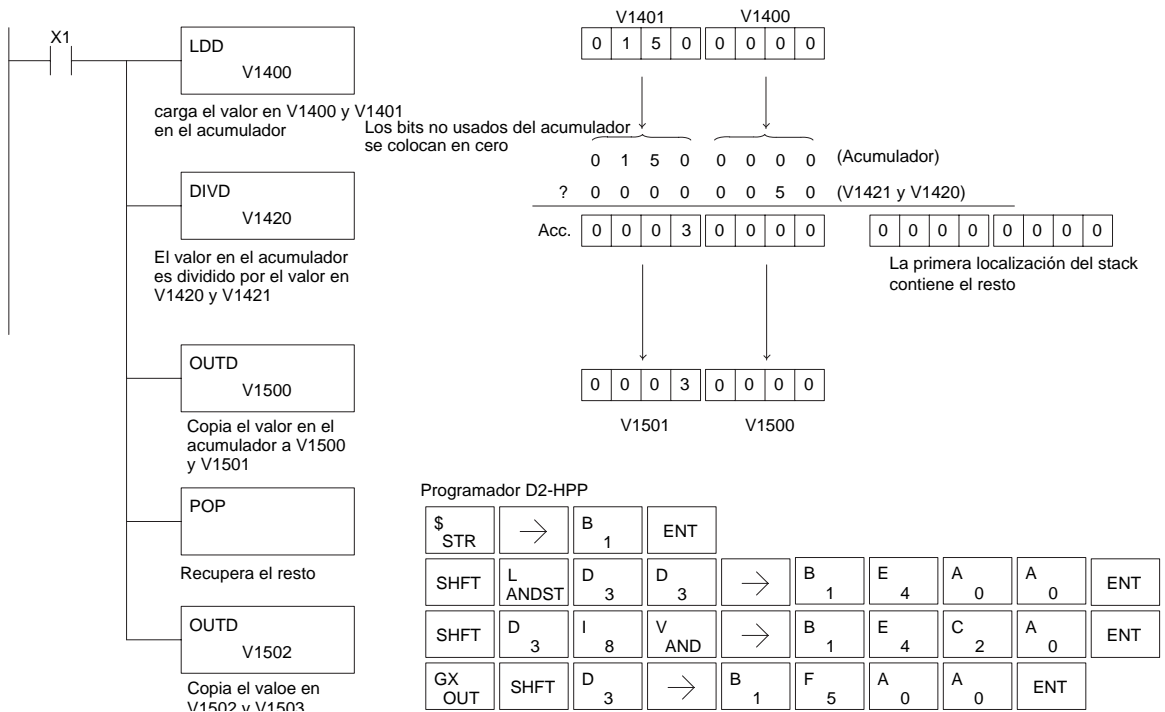
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

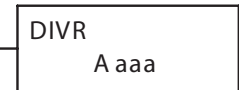
En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El valor en el acumulador es dividido por el valor en V1420 y V1421 usando la instrucción DIVD. La primera parte del cociente se va al acumulador y el resto se va al primer nivel del Stack . El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.



La instrucción Divide Real (DIVR)

DS5	Usado
HPP	N/A

La instrucción DIVR divide un número real en el acumulador por una constante real o por un número real que ocupa 2 direcciones consecutivas de memoria V. El resultado se va al acumulador. Ambos números deben estar de acuerdo al formato punto flotante IEEE.



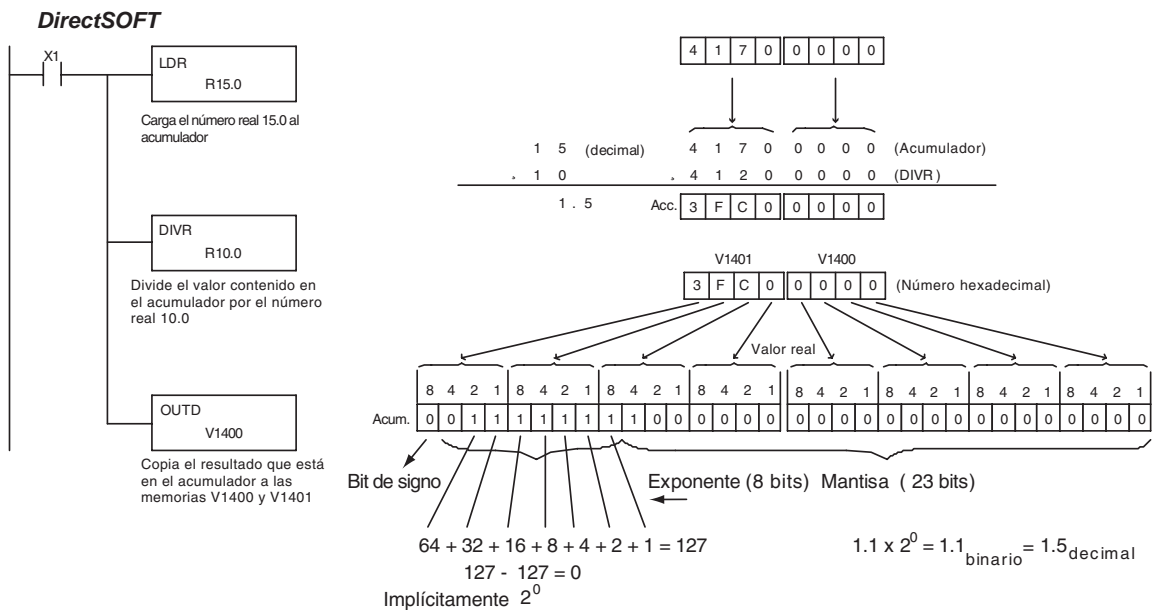
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante Real R	-3.402823E + 38 hasta + -3.402823E + 38

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP71	ON en cualquier momento que la memoria V especificada por un puntero (P) no es válida.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP74	ON cuando una operación de punto flotante resulta en un error underflow.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

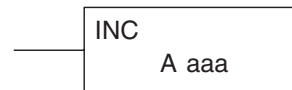


NOTA: El programador D2-HPP no permite entrar números reales con conversión automática al formato IEEE de 32 bits. Ud debe usar DirectSOFT en este caso, para usar esta función.

La instrucción Increment (INC)

DS5	Usado
HPP	Usado

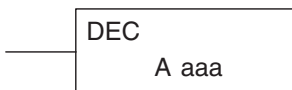
La instrucción INC incrementa un valor BCD en "1" en una dirección especificada de memoria V cada vez que se ejecuta la instrucción.



La instrucción Decrement (DEC)

DS5	Usado
HPP	Usado

La instrucción DEC decrementa en "1" un valor BCD en una dirección especificada de memoria V cada vez que se ejecuta la instrucción.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria

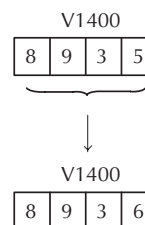
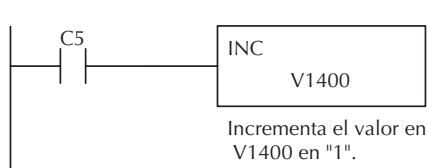
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

5

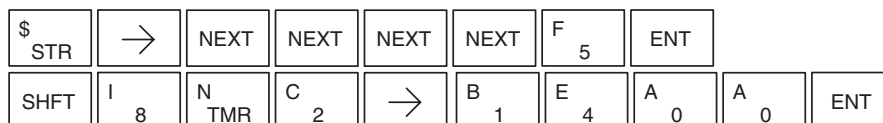


NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT

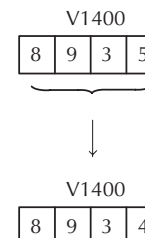
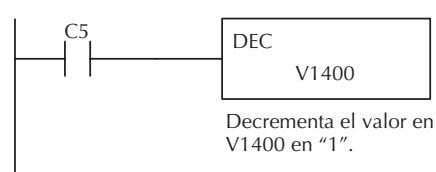


Programador D2-HPP

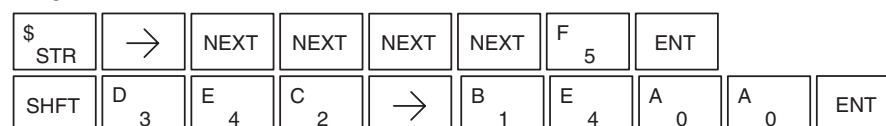


En el ejemplo siguiente, cuándo C5 está ON, el valor contenido en V1400 aumenta en 1.

DirectSOFT



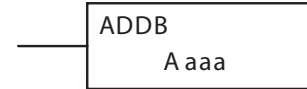
Programador D2-HPP



La instrucción Add Binary (ADDB)

DS5	Usado
HPP	Usado

ADDB es una instrucción de 16 bits que suma el valor binario en los 16 bits más bajos del acumulador con el valor (Aaaa) binario que es una localización de memoria V o una constante de 16 bits. El resultado puede ser de hasta de 32 bits y se va al acumulador. Note que se puede usar el complemento de 2 para expresar números negativos. Vea el apéndice J para más explicaciones.



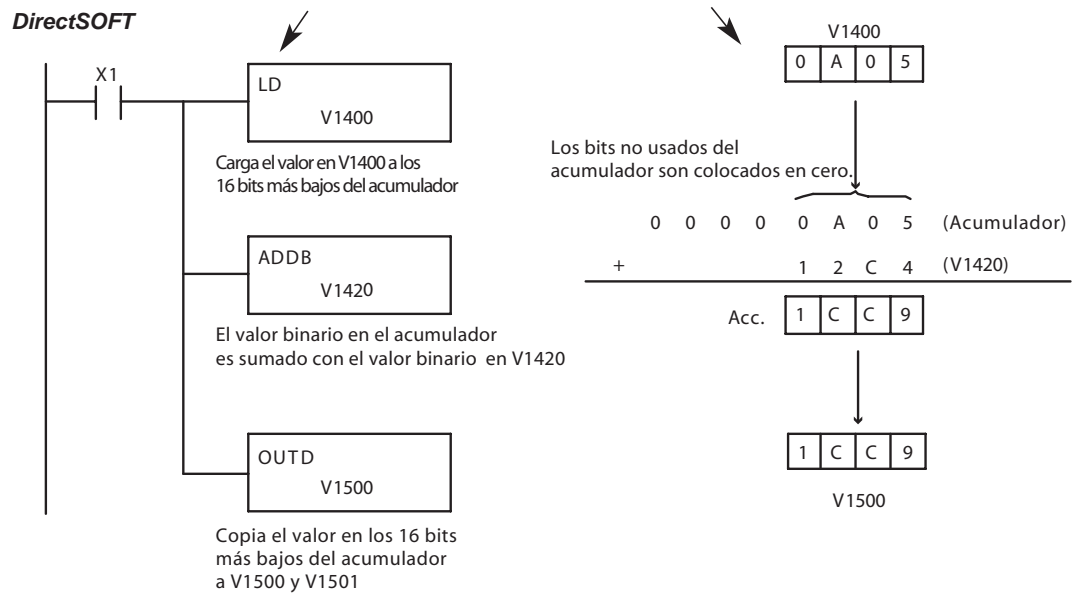
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P
Constante	K
	0-FFFF, h=65636

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP66	ON cuando la instrucción de 16 bits de suma resulta en un "pasa para".
SP67	ON cuando la instrucción de 32 bits de suma resulta en un "pasa para".
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON si una suma o resta con signo resulta con el bit de un signo incorrecto.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el siguiente ejemplo, cuando X1 está ON, el valor en V1400 se carga en el acumulador usando la instrucción LD. El valor binario en el acumulador es sumado al valor binario en

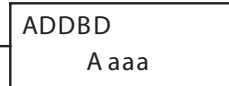


V1420 usando la instrucción ADDB. El valor en el acumulador es copiado a V1500 y V1501

La instrucción Add Binary Double (ADDBD)

DS5	Usado
HPP	Usado

ADDBD es una instrucción de 32 bits que suma el valor binario en el acumulador con el valor (Aaaa), que corresponde a dos localizaciones consecutivas de memoria V o una constante binaria de 32 bits. El resultado reside en el acumulador. Note que se puede usar el complemento de 2 para expresar números negativos. Vea el apéndice J para más explicaciones.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0-FFFF FFFF

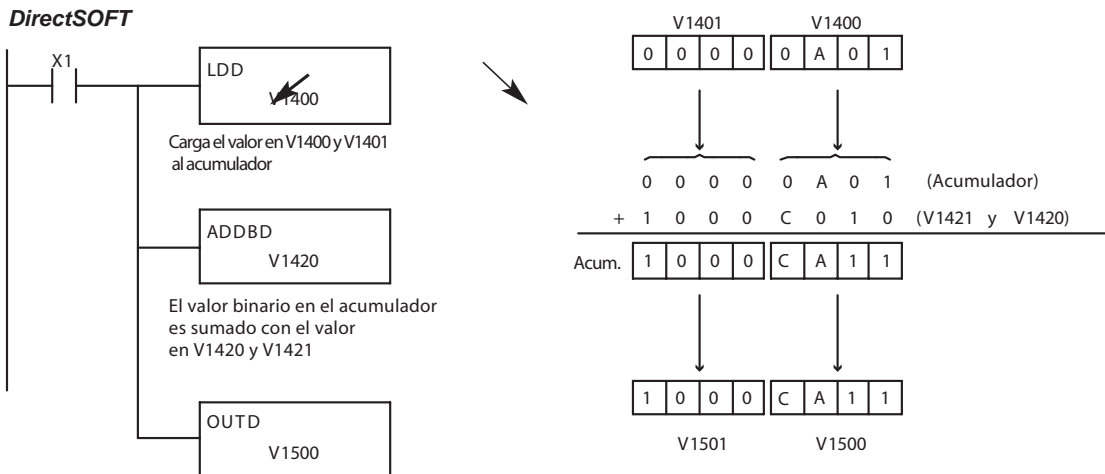
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP66	ON cuando la instrucción de 16 bits de suma resulta en un "pasa para".
SP67	ON cuando la instrucción de 32 bits de suma resulta en un "pasa para".
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON si una suma o resta con signo resulta con el bit de un signo incorrecto.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El valor binario en el acumulador se suma con el valor binario en V1420 y V1421 usando la instrucción ADDBD. El valor en el acumulador es copiado a V1500

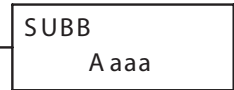


y V1501 usando la instrucción OUTD.

La instrucción Subtract Binary (SUBB)

DS5	Usado
HPP	Usado

SUBB es una instrucción de 16 bits que resta el valor (Aaaa) binario que es una dirección de memoria V o una constante del valor binario en el acumulador. El resultado se va al acumulador. Note que se puede usar el complemento de 2 para expresar números negativos. Vea el apéndice I para más explicaciones.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P
Constante	K
	0-FFFF, h=65636

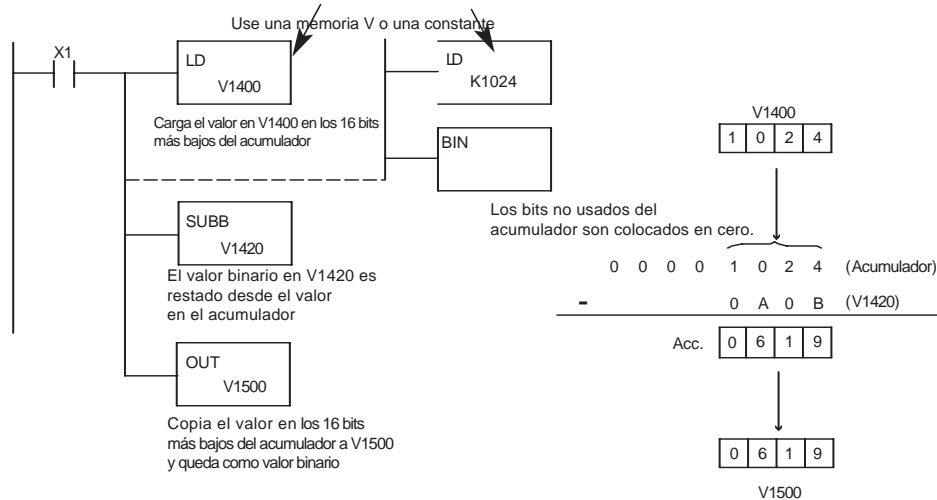
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de 16 bits de resta resulta en un "préstamo".
SP65	ON cuando la instrucción de 32 bits de resta resulta en un "préstamo".
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON si una suma o resta con signo resulta con el bit de un signo incorrecto.

5

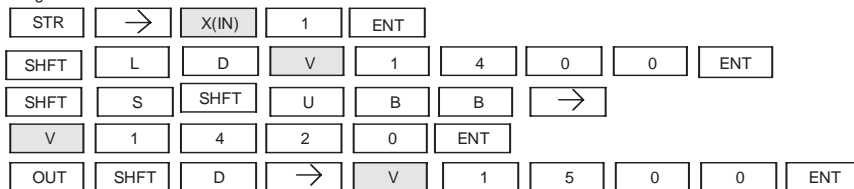


NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 se cargará al acumulador usando la instrucción LD. El valor binario en V1420 es restado del valor binario en el acumulador usa la instrucción SUBB. El valor en el acumulador es copiado a V1500 usando la instrucción OUT.



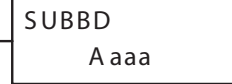
Programador D2-HPP



La instrucción Subtract Binary Double (SUBBD)

DS5	Usado
HPP	Usado

SUBBD es una instrucción de 32 bits que resta el valor (Aaaa) binario que son 2 direcciones consecutivas de memoria V o una constante binaria de 32 bits, del valor binario en el acumulador. El resultado se va al acumulador. Note que el complemento de 2 se puede usar para expresar números negativos.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0-FFFF FFFF

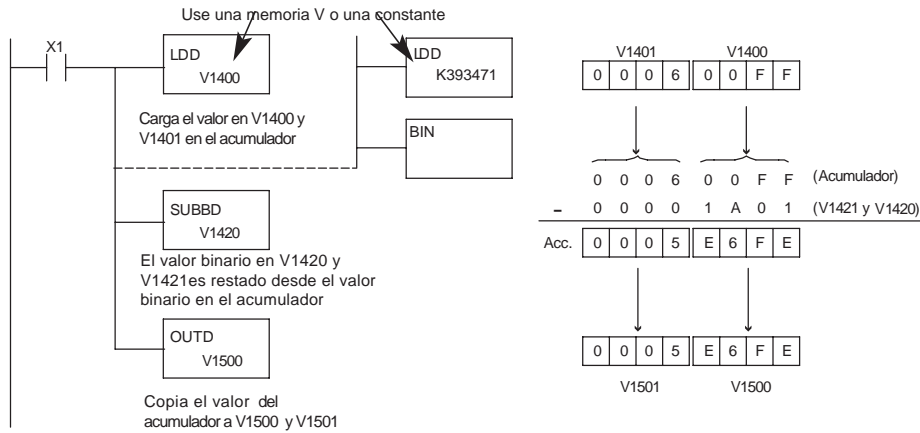
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de 16 bits de resta resulta en un "préstamo".
SP65	ON cuando la instrucción de 32 bits de resta resulta en un "préstamo"
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON si una suma o resta con signo resulta con el bit de un signo incorrecto.

5

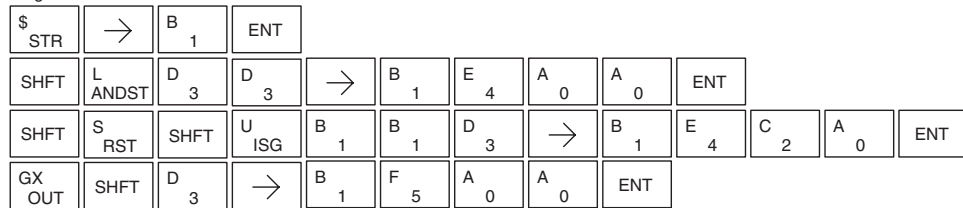


NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se cargará en el acumulador usando la instrucción LDD. El valor binario en V1420 y V1421 es restado del valor binario en el acumulador usando la instrucción SUBBD. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.



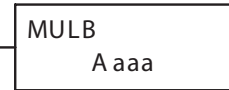
Programador D2-HPP



La instrucción Multiply Binary (MULB)

DS5	Usado
HPP	Usado

MULB es una instrucción de 16 bits que multiplica el valor (Aaaa) binario, que es una dirección de memoria V o una constante binaria de 16 bits, por el valor binario en el acumulador. El resultado puede llegar a ser de hasta de 32 bits y se va al acumulador. Note que se puede usar el complemento de 2 para expresar números negativos. Vea el apéndice J para más explicaciones.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0-FFFF

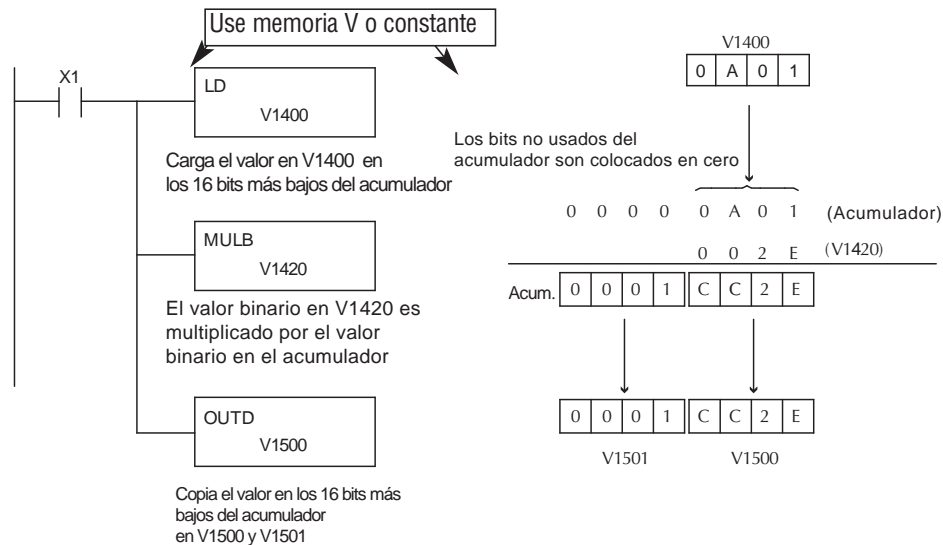
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

5

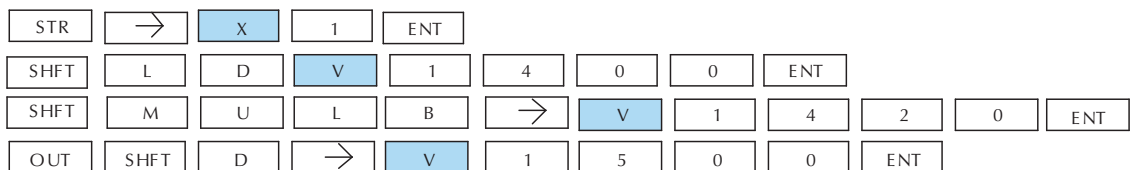


NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuando X1 está ON, el valor en V1400 se carga al acumulador usando la instrucción LD. Luego el valor binario en V1420 es multiplicado por el valor binario en el acumulador usando la instrucción MULB. El valor en el acumulador es copiado a V1500 usando la instrucción OUTD.



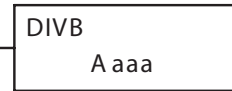
Programador D2-HPP



La instrucción Divide Binary (DIVB)

DS5	Usado
HPP	Usado

DIVB es una instrucción de 16 bits que divide el valor binario en el acumulador por un valor (Aaaa) binario, que es una dirección de memoria V o una constante binaria de 16 bits. La primera parte del cociente se va al acumulador y el residuo se va al primer nivel del stack. Note que se puede usar el complemento de 2 para expresar números negativos. Vea el apéndice J para más explicaciones.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P
Constante	K
	0-FFFF

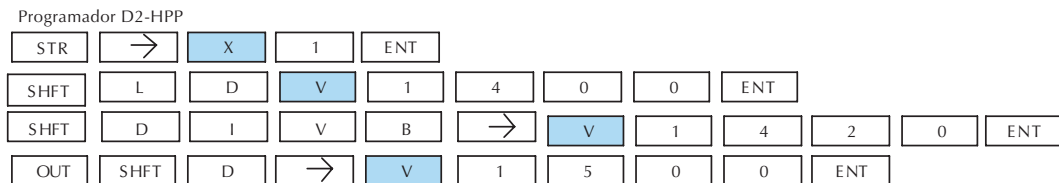
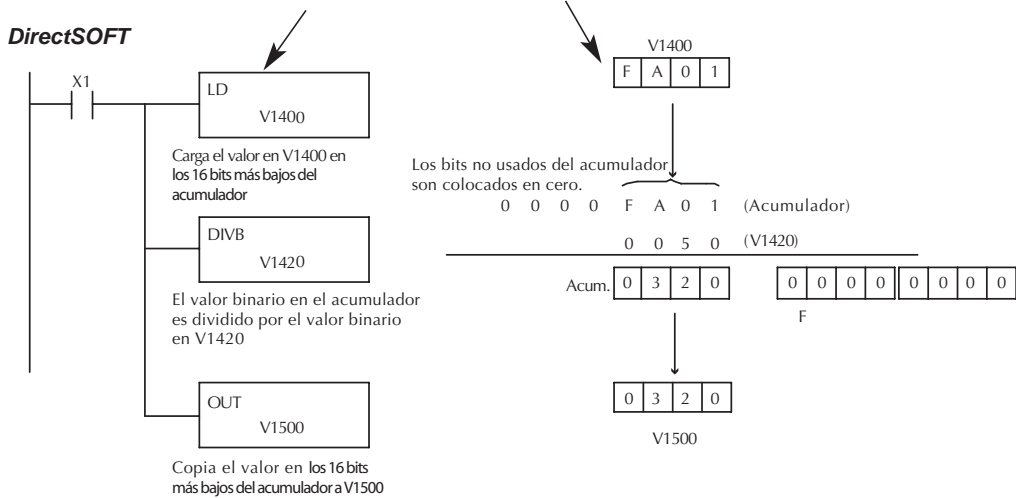
Indicadores	Descripción
SP53	ON cuando el valor del operando es mayor que lo que puede trabajar el acumulador.
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

5



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

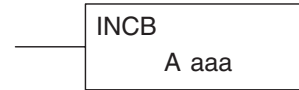
En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V1400 al acumulador usando la instrucción LD. El valor binario en el acumulador es dividido por el valor binario en V1420 usando la instrucción DIVB. El valor en el acumulador es copiado a V1500 usando la instrucción OUT.



La instrucción Increment Binary (INCB)

DS5	Usado
HPP	Usado

La instrucción INCB incrementa un valor binario en "1" en una dirección especificada de memoria V cada vez que se ejecuta la instrucción.

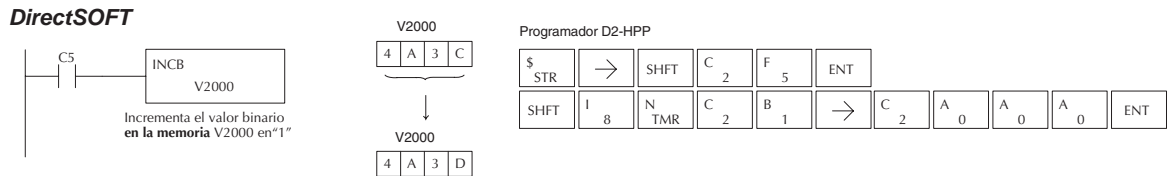


Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.

5

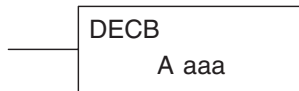
En el ejemplo siguiente cuando C5 está ON, el valor binario en V2000 es aumentado en 1.



La instrucción Decrement Binary (DECB)

DS5	Usado
HPP	Usado

La instrucción DECB decrementa en "1" un valor binario en una dirección especificada de la memoria V, cada vez que la instrucción se ejecuta



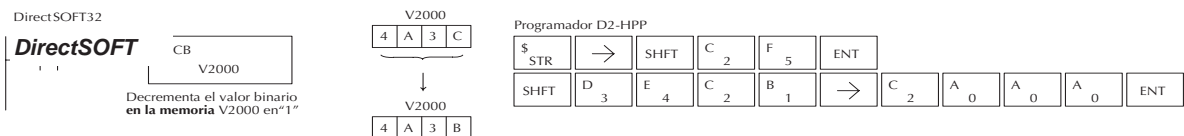
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria

Indicadores	Descripción
SP63	On cuando el resultado de la instrucción causa que el valor en el acumulador sea cero.



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP.

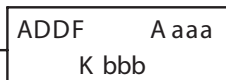
En el ejemplo siguiente cuando C5 está ON, el valor binario en V2000 es disminuido en 1.



La instrucción Add Formatted (ADDF)

DS5	Usado
HPP	Usado

ADDF es una instrucción de 32 bits que suma el valor BCD en el acumulador con el valor BCD (Aaaa), que es un rango de bits discretos. El rango (Kbbb) especificado puede ser 1 a 32 bits consecutivos. El resultado se va al acumulador.



Tipo de operando de datos		Rango del DL06	
A		aaa	bbb
Entradas	X	0-777	—
Salidas	Y	0-777	—
Relevadores de control	C	0-1777	—
Bits de etapas	S	0-1777	—
Bits de temporizadores	T	0-377	—
Bits de contadores	CT	0-177	—
Relevadores especiales	SP	0-137 320-717	—
Global I/O	GX	0-3777	—
Constante	K	—	1-32

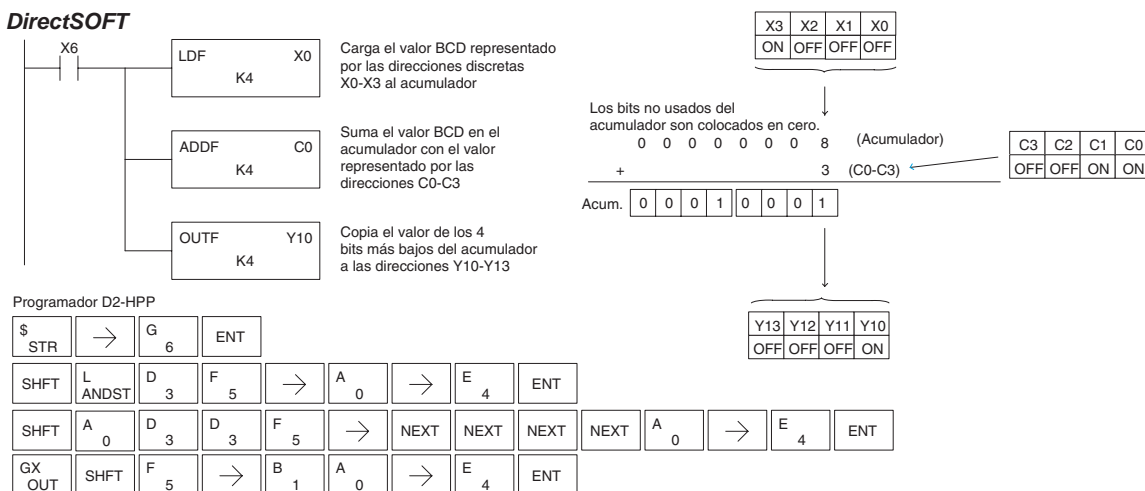
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP66	ON cuando la instrucción de suma de 16 bits resulta en un "pasa para".
SP67	ON cuando la instrucción de suma de 32 bits resulta en un "pasa para".
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON si se espera un número BCD y se encuentra un número diferente de BCD.
SP75	ON si se ejecuta una instrucción BCD y se encuentra un número diferente de BCD.

5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

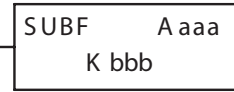
En el ejemplo siguiente, cuándo X6 está ON, el valor BCD formado por las direcciones discretas X0-X3 se carga en el acumulador usando la instrucción LDF. El valor BCD formado por las direcciones discretas C0-C3 se suma al valor en el acumulador usando la instrucción ADDF. El valor en los 4 bits más bajos del acumulador es copiado a Y10-Y13 usando la instrucción OUTF.



La instrucción Subtract Formatted (SUBF)

DS5	Usado
HPP	Usado

SUBF es una instrucción de 32 bits que resta el valor BCD (Aaaa), que es un rango de bits distintos del valor BCD en el acumulador. El rango (Kbbb) especificado puede ser 1 a 32 bits consecutivos. El resultado se va al acumulador.



Tipo de operando de datos	Rango del DL06	
	aaa	bbb
..... A		
Entradas X	0-777	—
Salidas Y	0-777	—
Relevadores de control C	0-1777	—
Bits de etapas S	0-1777	—
Bits de temporizadores T	0-377	—
Bits de contadores CT	0-177	—
Relevadores especiales SP	0-137 320-717	—
Global I/O GX	0-3777	—
Constante K	—	1-32

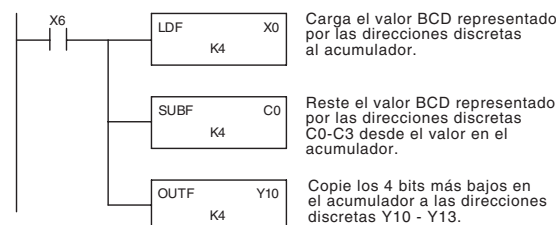
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de resta de 16 bits resulta en un "préstamo".
SP65	ON cuando la instrucción de resta de 32 bits resulta en un "préstamo".
SP70	ON cuando el valor en el acumulador es un número negativo.
SP73	On cuando hay una instrucción de suma o resta que resulta en un bit de signo incorrecto.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.



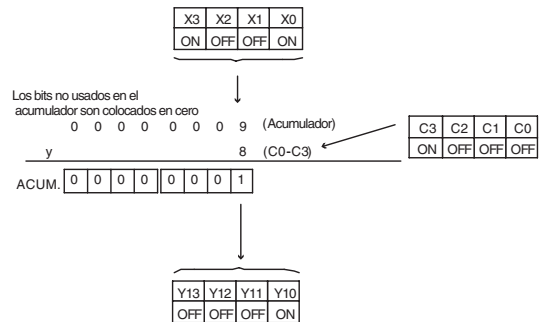
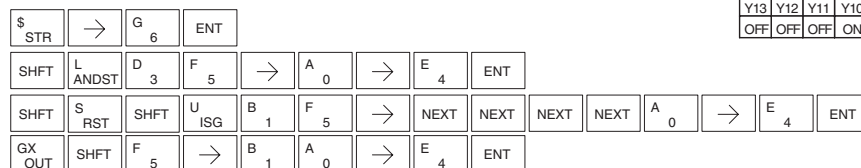
NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X6 está ON, el valor BCD formado por las direcciones discretas X0-X3 se carga al acumulador usando la instrucción LDF. El valor BCD formado por las direcciones discretas C0-C3 se resta del valor en el acumulador usando la instrucción SUBF. El valor en los 4 bits más bajos del acumulador es copiado a Y10-Y13 usando la instrucción OUTF.

DirectSOFT



Programador D2-HPP



La instrucción Multiply Formatted (MULF)

DS5	Usado
HPP	Usado

MULF es una instrucción de 16 bits que multiplica el valor BCD en el acumulador por el valor BCD (Aaaa) que es un rango de bits discretos. El rango (Kbbb) especificado puede ser 1 a 16 bits consecutivos. El resultado se va al acumulador.



Tipo de operando de datos	Rango del DL06	
	aaa	bbb
Entradas X	0-777	—
Salidas Y	0-777	—
Relevadores de control C	0-1777	—
Bits de etapas S	0-1777	—
Bits de temporizadores T	0-377	—
Bits de contadores CT	0-177	—
Relevadores especiales SP	0-137 320-717	—
Global I/O GX	0-3777	—
Constante K	—	1-16

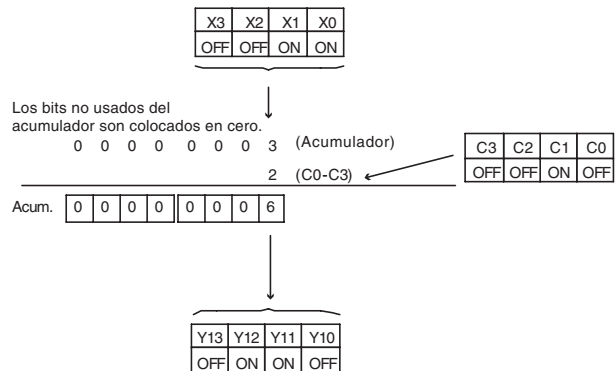
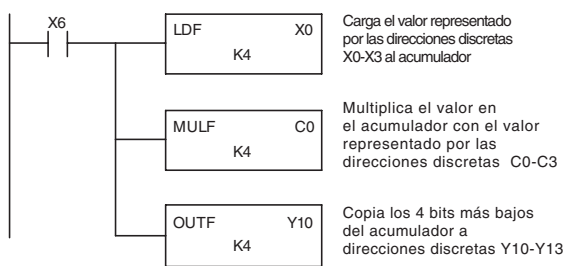
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es un número negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X6 está ON, el valor formado por las direcciones discretas X0-X3 se carga al acumulador usando la instrucción LDF. El valor formado por las direcciones discretas C0-C3 es multiplicado por el valor en el acumulador usando la instrucción MULF. El valor en los 4 bits más bajos del acumulador es copiado a Y10-Y13 usando la instrucción OUTF.

DirectSOFT



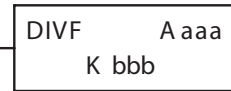
Programador D2-HPP



La instrucción Divide Formatted (DIVF)

DS5	Usado
HPP	Usado

DIVF es una instrucción de 16 bits que divide el valor BCD en el acumulador por el valor BCD (Aaaa), que es un rango de bits discretos. El rango (Kbbb) especificado puede ser 1 a 16 bits consecutivos. La primera parte del cociente se va al acumulador y el residuo se va al primer nivel del Stack.



Tipo de operando de datos		Rango del DL06	
A		aaa	bbb
Entradas	X	0-777	—
Salidas	Y	0-777	—
Relevadores de control	C	0-1777	—
Bits de etapas	S	0-1777	—
Bits de temporizadores	T	0-377	—
Bits de contadores	CT	0-177	—
Relevadores especiales	SP	0-137 320-717	—
Global I/O	GX	0-3777	—
Constante	K	—	1-16

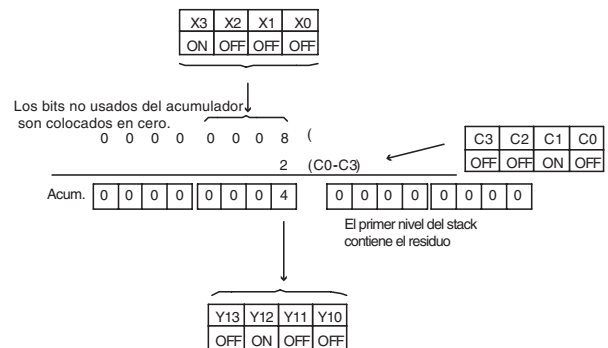
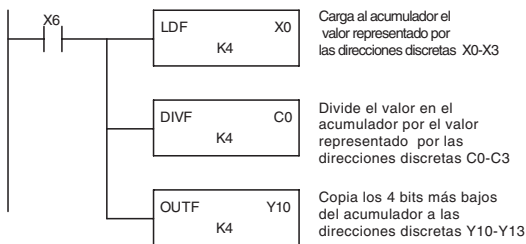
Indicadores	Descripción
SP53	ON cuando el valor del operando es más grande de lo que puede trabajar el acumulador.
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es un número negativo (MSB es 1).
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X6 está ON, se carga el valor formado por las direcciones discretas X0-X3 al acumulador usando la instrucción LDF. El valor en el acumulador es dividido por el valor formado por las direcciones discretas C0-C3 usando la instrucción DIVF. El valor en los 4 bits más bajos del acumulador es copiado a Y10-Y13 usando la instrucción OUTF.

DirectSOFT



Programador D2-HPP



La instrucción Add Top of Stack (ADDS)

DS5	Usado
HPP	Usado

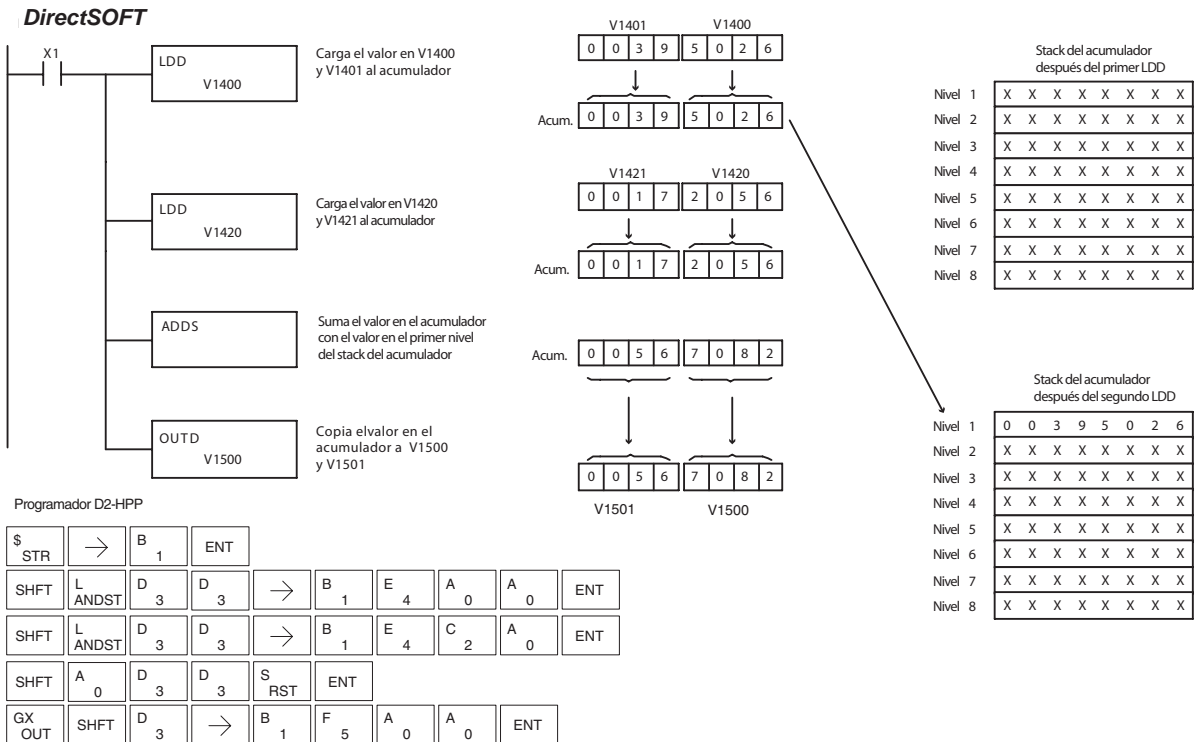
ADDS es una instrucción de 32 bits que suma el valor BCD en el acumulador con el valor BCD en el primer nivel del Stack del acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador es removido y todos los valores del Stack se mueven un nivel para arriba.

ADDS

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP66	ON cuando la instrucción de suma de 16 bits da un resultado con "pasa para".
SP67	ON cuando la instrucción de suma de 32 bits da un resultado con "pasa para".
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON cuando una suma o resta con signo resulta con un bit de signo incorrecto.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

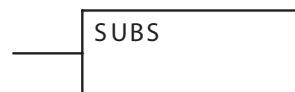
En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El valor en V1420 y V1421 se carga al acumulador usando la instrucción LDD, empujando el valor previamente cargado al acumulador en el Stack del acumulador. El valor en el primer nivel del Stack se suma con el valor en el acumulador usando la instrucción ADDS. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.



La instrucción Subtract Top of Stack (SUBS)

DS5	Usado
HPP	Usado

SUBS es una instrucción de 32 bits que resta el valor BCD en el primer nivel del Stack del acumulador del valor BCD en el acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador es removido y todos los valores del Stack se mueven un nivel hacia arriba.

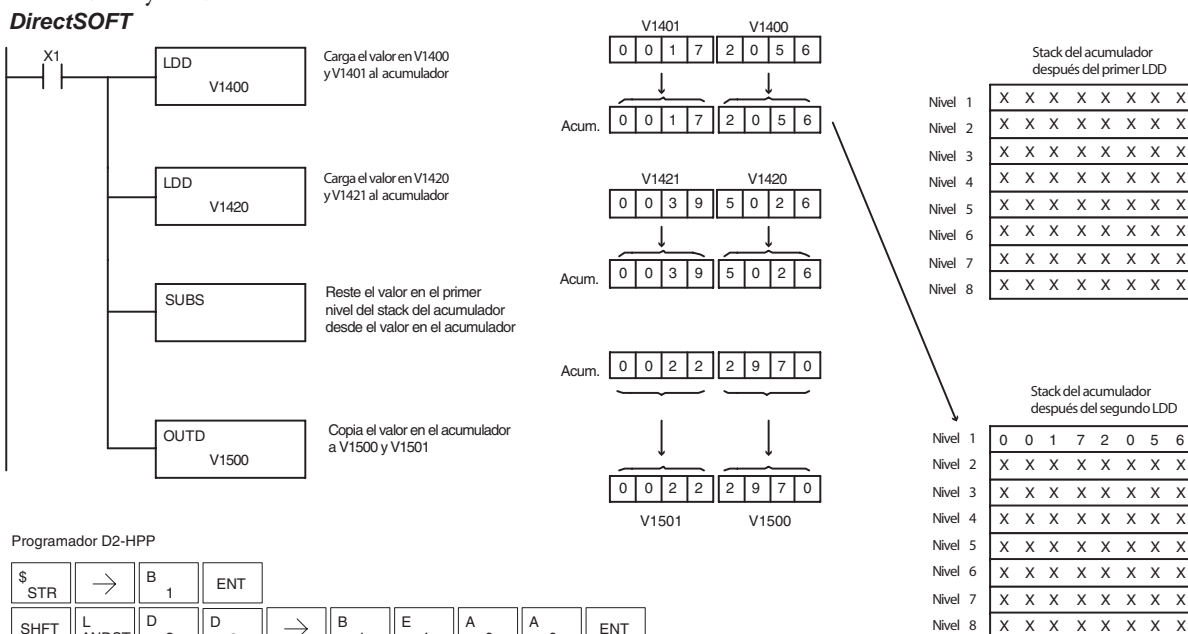


Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de resta de 16 bits resulta en un "préstamo".
SP65	ON cuando la instrucción de resta de 32 bits resulta en un "préstamo".
SP70	ON cuando el valor en el acumulador es un número negativo.
SP73	ON cuando una suma o resta con signo resulta con un bit de signo incorrecto.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El valor en V1420 y V1421 se carga al acumulador usando la instrucción LDD, empujando el valor previamente cargado en el acumulador en el Stack del acumulador. El valor BCD en el primer nivel del Stack del acumulador se resta del valor BCD en el acumulador usando instrucción SUBS. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.



La instrucción Multiply Top of Stack (MULS)

DS5	Usado
HPP	Usado

MULS es una instrucción de 16 bits que multiplica un valor de 4 dígitos BCD en el primer nivel del Stack del acumulador por un valor de 4 dígitos BCD en el acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador es removido y todos valores del Stack se mueven un nivel hacia arriba.

MULS

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es un número negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.

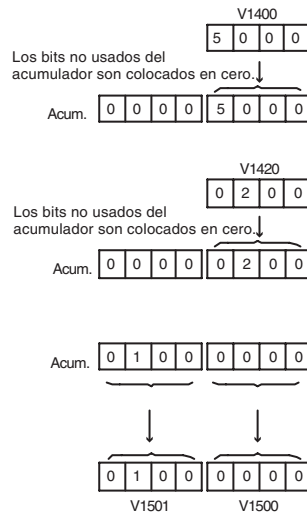
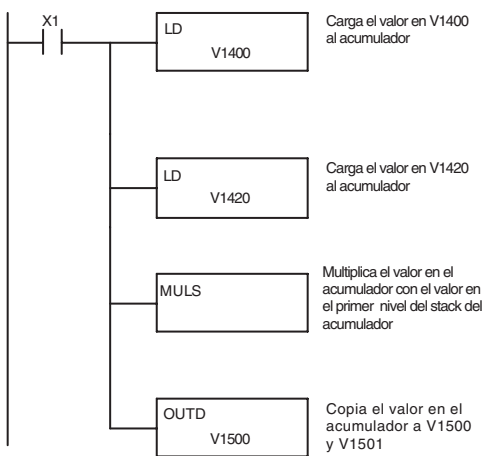
5



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V1400 al acumulador usando la instrucción LD. El valor en V1420 se carga al acumulador usando la instrucción LD, empujando el valor previamente cargado en el acumulador al Stack del acumulador. El valor BCD en el primer nivel del Stack del acumulador es multiplicado por el valor BCD en el acumulador usando la instrucción MULS. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.

DirectSOFT



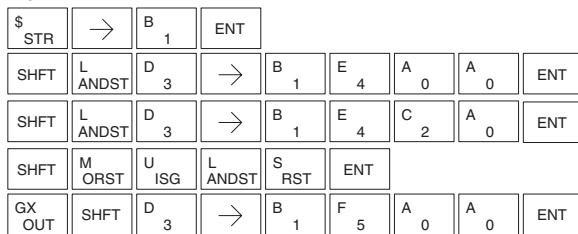
Stack del acumulador después del primer LDD

Nivel 1	X	X	X	X	X	X	X
Nivel 2	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X

Stack del acumulador después del segundo LDD

Nivel 1	0	0	0	0	5	0	0	0
Nivel 2	X	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X	X

Programador D2-HPP



La instrucción Divide by Top of Stack (DIVS)

DS5	Usado
HPP	Usado

DIVS es una instrucción de 32 bits que divide el valor de 8 dígitos BCD en el acumulador por un valor de 4 dígitos BCD en el primer nivel del Stack del acumulador. El resultado se va al acumulador y el residuo se va al primer nivel del Stack del acumulador.

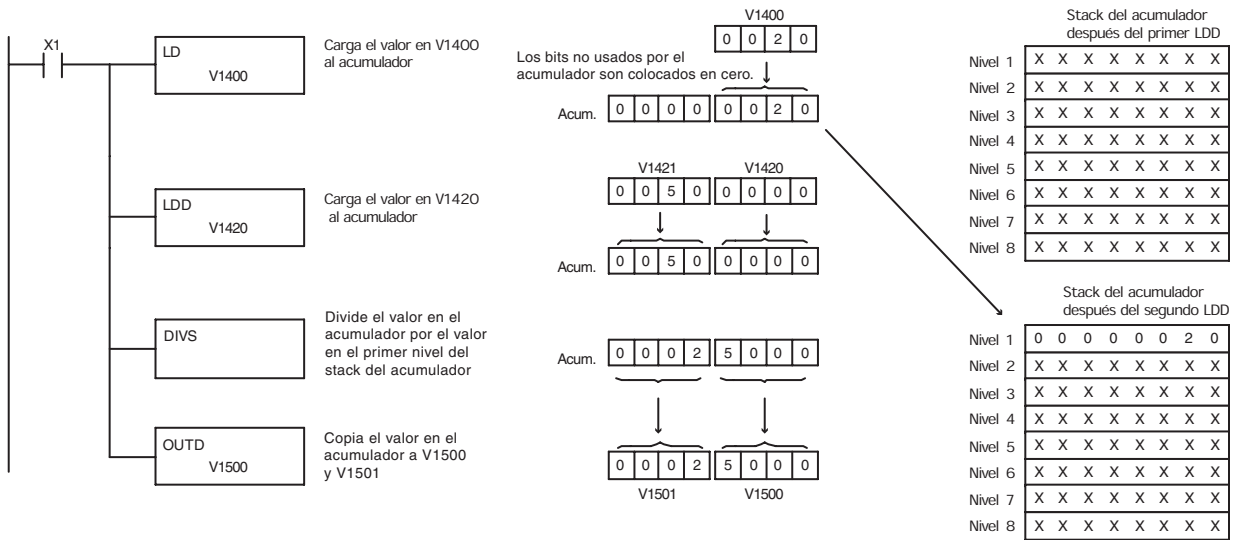
DIVS

Indicadores	Descripción
SP53	ON cuando el valor en el operando es más grande de lo que el acumulador puede trabajar.
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es un número negativo.
SP75	ON si se espera un número BCD y se encuentra un número diferente de BCD.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, la instrucción LD carga el valor que está en V1400 al acumulador. El valor en V1420 se carga al acumulador usando la instrucción LDD, empujando el valor previamente cargado en el acumulador al Stack del acumulador. El valor BCD en el acumulador es dividido por el valor BCD en el primer nivel del Stack del acumulador usando la instrucción DIVS. Luego se copia el valor en el acumulador a V1500 y V1501 usando la instrucción OUTD.



Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	→	B	1	E	4	A	0	A	0	ENT		
SHFT	L	ANDST	D	3	D	3	→	B	1	E	4	C	2	A	0	ENT
SHFT	D	3	I	8	V	AND	S	RST	ENT							
GX	OUT	SHFT	D	3	→	B	1	F	5	A	0	A	0	ENT		

El residuo se val al primer nivel del stack

Nivel 1	0	0	0	0	0	0	0	0	0
Nivel 2	X	X	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X	X	X

La instrucción Add Binary Top of Stack (ADDDBS)

DS5	Usado
HPP	Usado

La instrucción ADDDBS es una instrucción de 32 bits que suma el valor binario en el acumulador con el valor binario en el primer nivel del Stack del acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador se elimina y todos valores del Stack se mueven un nivel hacia arriba.



Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP66	ON cuando la instrucción de suma de 16 bits da un resultado con "pasa para".
SP67	ON cuando la instrucción de suma de 32 bits da un resultado con "pasa para".
SP70	ON cuando el valor en el acumulador es negativo.
SP73	ON cuando una suma o resta con signo resulta con un bit de signo incorrecto.

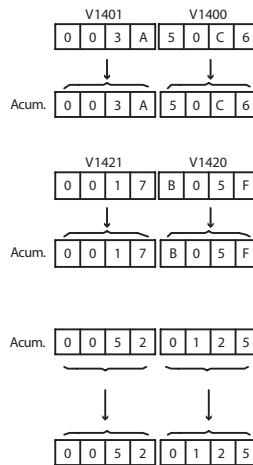
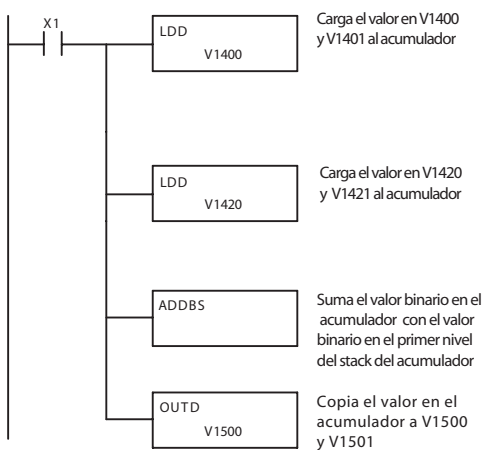
5



NOTA: Las indicaciones de estado discretas SP son sólo válidas hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El valor en V1420 y V1421 se carga al acumulador usando la instrucción LDD, empujando el valor previamente cargado en el acumulador al Stack del acumulador. El valor binario en el primer nivel del Stack del acumulador se suma con el valor binario en el acumulador usando la instrucción ADDDBS. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD. Double instrucción.

DirectSOFT



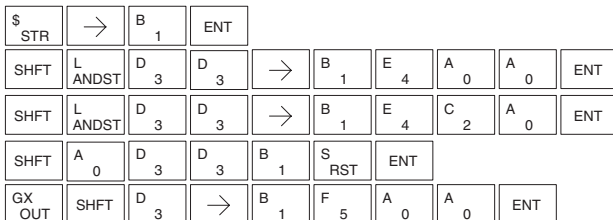
Stack del acumulador después del primer LDD

Nivel 1	X	X	X	X	X	X	X
Nivel 2	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X

Stack del acumulador después del segundo LDD

Nivel 1	0	0	3	A	5	0	C	6
Nivel 2	X	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X	X

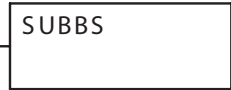
Programador D2-HPP



La instrucción Subtract Binary Top of Stack (SUBBS)

DS5	Usado
HPP	Usado

SUBBS es una instrucción de 32 bits que resta el valor binario en el primer nivel del Stack del acumulador del valor binario en el acumulador. El resultado se va al acumulador. El valor en el primer nivel del Stack del acumulador se pierde y todas direcciones del Stack se mueven un nivel hacia arriba.



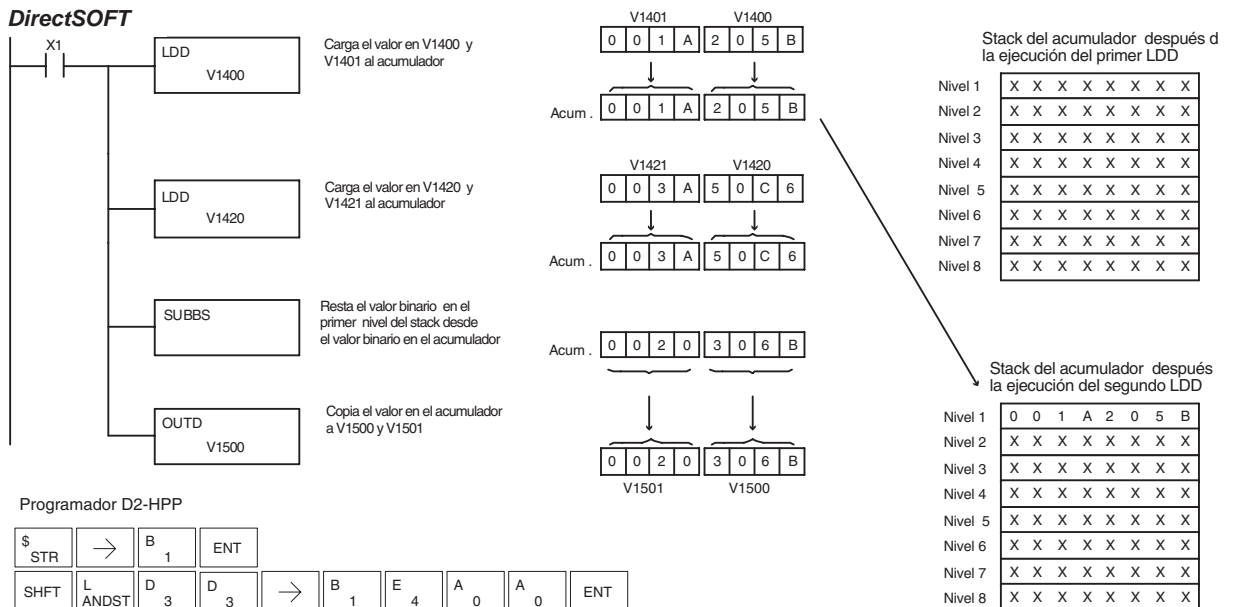
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP64	ON cuando la instrucción de resta de 16 bits resulta en un "préstamo".
SP65	ON cuando la instrucción de resta de 32 bits resulta en un "préstamo".
SP70	ON en cualquier momento que el valor en el acumulador es negativo.
SP73	ON cuando una suma o resta con signo resulta con un bit de signo incorrecto.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El valor en V1420 y V1421 se carga al acumulador usando la instrucción LDD, empujando el valor previamente cargado en el acumulador en el Stack del acumulador.

El valor binario en el primer nivel del Stack del acumulador se resta del valor binario en el acumulador que usa la instrucción SUBBS. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.



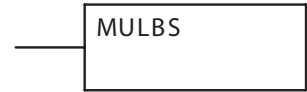
Programador D2-HPP

\$	STR	→	B	1	ENT							
SHFT	L	ANDST	D	3	→	B	1	E	4	A	0	ENT
SHFT	L	ANDST	D	3	→	B	1	E	4	C	2	ENT
SHFT	S	RST	SHFT	U	ISG	B	1	S	RST	ENT		
GX	OUT	SHFT	D	3	→	B	1	F	5	A	0	ENT

La instrucción Multiply Binary Top of Stack (MULBS)

DS5	Usado
HPP	Usado

MULBS es una instrucción de 16 bits que multiplica el valor binario de 16 bits en el primer nivel del Stack del acumulador por el valor binario de 16 bits en el acumulador. El resultado se va al acumulador y puede ser de 32 bits (8 dígitos máximos.) El valor en el primer nivel del Stack del acumulador se pierde y todas direcciones del Stack se mueven un nivel hacia arriba.



5



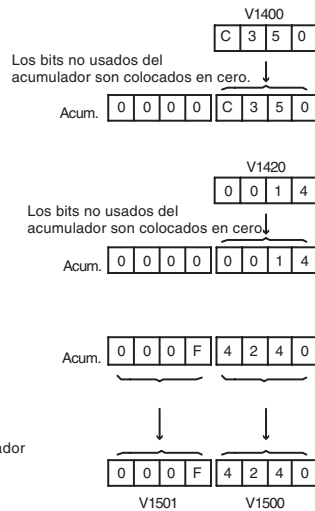
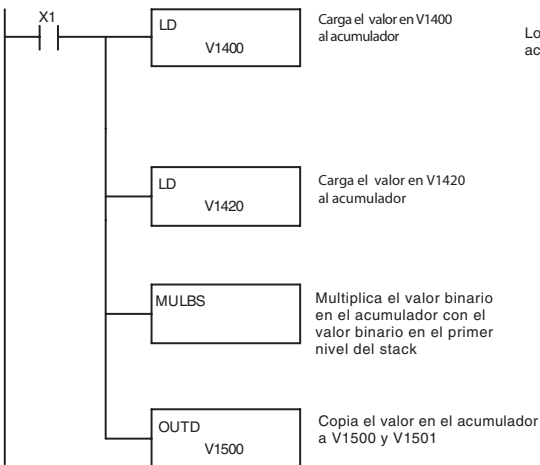
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	On cuando el valor en el acumulador es negativo.

NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, la instrucción LD mueve el valor en V1400 al acumulador. El valor en V1420 se carga al acumulador usando la instrucción LD, empujando el valor previamente Cargado en el acumulador al Stack. El valor binario en el primer nivel del Stack es multiplicado por el valor binario en el acumulador usando la instrucción MULBS.

La instrucción OUTD copia el valor en el acumulador a V1500 y V1501.

DirectSOFT



Stack del acumulador después del primer LD

Nivel 1	X	X	X	X	X	X	X
Nivel 2	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X

Stack del acumulador después del segundo LD

Nivel 1	0	0	0	0	C	3	5	0
Nivel 2	X	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X	X

Programador D2-HPP

\$	STR	→	B	1	ENT								
SHFT	L ANDST	D	3	→	B	1	E	4	A	0	A	0	ENT
SHFT	L ANDST	D	3	→	B	1	E	4	C	2	A	0	ENT
SHFT	M ORST	U	ISG	L ANDST	B	1	S	RST	ENT				
GX OUT	SHFT	D	3	→	B	1	F	5	A	0	A	0	ENT

La instrucción Divide Binary by Top OF Stack (DIVBS)

DS5	Usado
HPP	Usado

Esta es una instrucción de 32 bits que divide el valor binario de 32 bits en el acumulador por el valor binario de 16 bits en el primer nivel del stack del acumulador.



El resultado reside en el acumulador y el resto reside en el primer nivel del stack del acumulador.

Indicadores	Descripción
SP53	ON cuando el valor del operando es más grande que lo que puede aceptar el acumulador
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

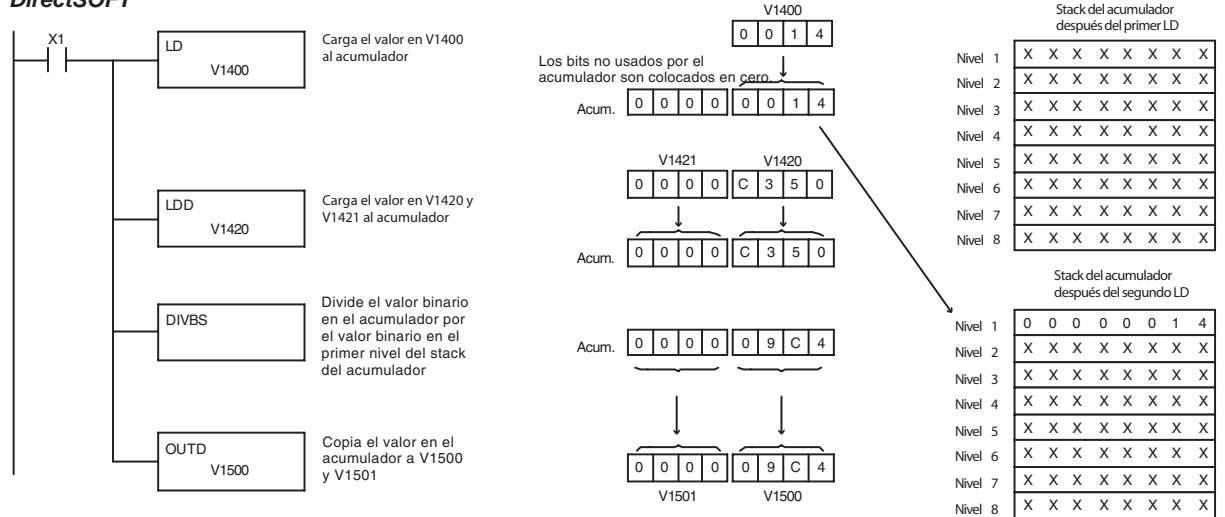


NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuando X1 está ON, se carga el valor en V1400 en el acumulador usando la instrucción LD. El valor en V1420 y V1421 también se carga en el acumulador usando la instrucción LDD, empujando el valor cargado previamente en el acumulador sobre el stack del acumulador. El valor binario en el acumulador es dividido por el valor binario en el primer nivel del stack del acumulador usando la instrucción DIVBS.

El valor en el acumulador se copia a V1500 y a V1501 usando la instrucción OUTD.

DirectSOFT



Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	→	B	1	E	4	A	0	A	0	ENT		
SHFT	L	ANDST	D	3	D	3	→	B	1	E	4	C	2	A	0	ENT
SHFT	D	3	I	8	V	AND	B	1	S	RST	ENT					
GX	OUT	SHFT	D	3	→	B	1	F	5	A	0	A	0	ENT		

El residuo se queda en el primer nivel del stack

Nivel 1	0	0	0	0	0	0	0	0	0
Nivel 2	X	X	X	X	X	X	X	X	X
Nivel 3	X	X	X	X	X	X	X	X	X
Nivel 4	X	X	X	X	X	X	X	X	X
Nivel 5	X	X	X	X	X	X	X	X	X
Nivel 6	X	X	X	X	X	X	X	X	X
Nivel 7	X	X	X	X	X	X	X	X	X
Nivel 8	X	X	X	X	X	X	X	X	X

Funciones transcendentales

El PLC DL06 permite ejecutar funciones numéricas especiales para complementar su capacidad de procesar números reales. Las funciones transcendentales incluyen el seno, coseno, y tangente trigonométricos y también sus inversos (arcoseno, arcocoseno y arcotangente). La función de raíz cuadrada también se agrupa con estas otras funciones.

Las instrucciones transcendentales funcionan en un número real localizado en el acumulador (no puede ser BCD o binario). El resultado de la operación reside en el acumulador. La función de raíz cuadrada funciona en el rango completo de números reales positivos. Las funciones de seno, coseno y tangente requieren números expresados en radianes. Usted puede trabajar con ángulos expresados en grados primero convirtiéndolos a radianes con la instrucción radián (RADR) y luego ejecutando la función trigonométrica. Todas las funciones transcendentales utilizan los bits de indicación siguientes:

Indicadores	Descripción
SP53	ON cuando el valor del operando es más grande que de lo que el acumulador puede aceptar.
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP73	ON cuando el valor en el acumulador es negativo.
SP75	ON cuando se ejecuta una instrucción de número real y fue encontrado un número que no es real.

La instrucción Sine Real (SINR)

DS5	Usado	La instrucción SINR calcula el seno del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	SINR
HPP	N/A		

La instrucción Cosine Real (COSR)

DS5	Usado	La instrucción COSR calcula el coseno del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	COSR
HPP	N/A		

La instrucción Tangent Real (TANR)

DS5	Usado	La instrucción TANR calcula la tangente del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	TANR
HPP	N/A		

La instrucción Arc Sine Real (ASINR)

DS5	Usado	La instrucción ASINR calcula el arcoseno del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	ASINR
HPP	N/A		

La instrucción Arc Cosine Real (ACOSR)

DS5	Usado	La instrucción ACOSR calcula el arcocoseno del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	ACOSR
HPP	N/A		

La instrucción Arc Tangent Real (ATANR)

DS5	Usado	La instrucción ATANR calcula el arcotangente del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	ATANR
HPP	N/A		

La instrucción Square Root Real (SQRTR)

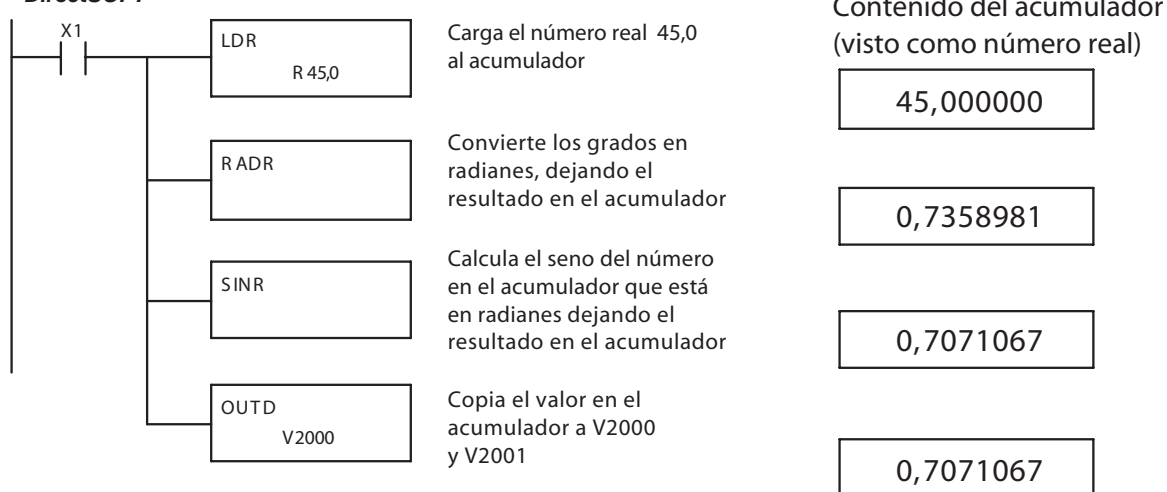
DS5	Usado	La instrucción SQRTR calcula la raíz cuadrada del número real almacenado en el acumulador. El resultado se va al acumulador. El número original y el resultado deben estar en el formato de 32 bits IEEE.	SQRTR
HPP	N/A		



NOTA: La función raíz cuadrada puede ser útil en varias situaciones. Sin embargo, si se trata de hacer la función de extracción de raíz para un instrumento medidor de flujo del tipo de placa orificio como PV para un lazo de PID, note que el lazo de PID ya tiene una función de extracción de raíz cuadrada incluida.

El ejemplo siguiente toma el seno de 45 grados. Ya que estas funciones trascendentales operan sólo con números reales, hacemos una instrucción LDR (Carga real) con el operando 45,0. Las funciones trigonométricas operan sólo con radianes, así que se debe convertir los grados a radianes usando la instrucción RADR. Después de usar la instrucción SINR (Seno Real), se usa la instrucción OUTD para mover el resultado del acumulador a la memoria V. El resultado es

DirectSOFT



de 32 bits, y se necesita la instrucción OUTD para moverlo.

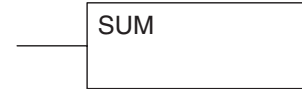
NOTA: El D2-HPP no permite el uso de números reales con la conversión automática al formato de 32 bit de IEEE. Usted debe utilizar DirectSOFT para entrar números reales, usando la instrucción LDR.

Instrucciones de operación con bits

La instrucción Sum (SUM)

DS5	Usado
HPP	Usado

La instrucción SUM cuenta el número de bits que son "1" en el acumulador. El resultado en hexadecimal se va al acumulador.



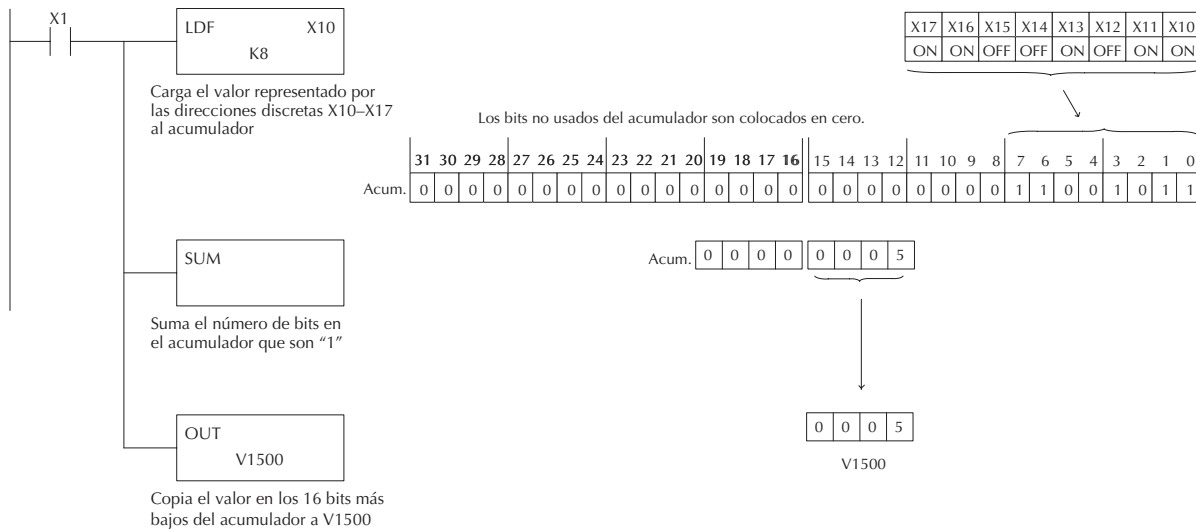
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.

En el ejemplo siguiente, cuando X1 está ON, se carga el valor formado por las direcciones discretas X10-X17 al acumulador usando la instrucción LDF. Luego es contado el número de bits del acumulador que son "1s", usando la instrucción SUM. El valor en el acumulador es copiado a V1500 usando la instrucción OUT.

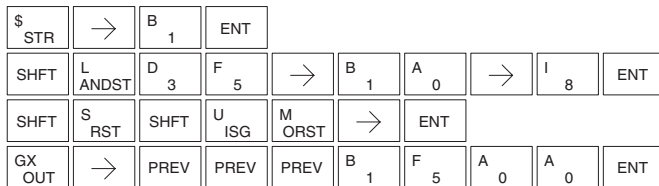


NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT



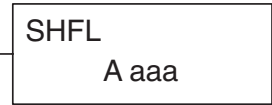
Programador D2-HPP



La instrucción Shift Left (SHFL)

DS5	Usado
HPP	Usado

SHFL es una instrucción de 32 bits que desplaza los bits en el acumulador un número especificado de lugares (Aaaa) a la izquierda, es decir, en la dirección desde el bit menos significativo al más significativo. Las posiciones vacías se llenan con ceros y los bits que son desplazados fuera del acumulador se pierden.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Constante	K
	Vea el mapa de memoria
	1-32

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

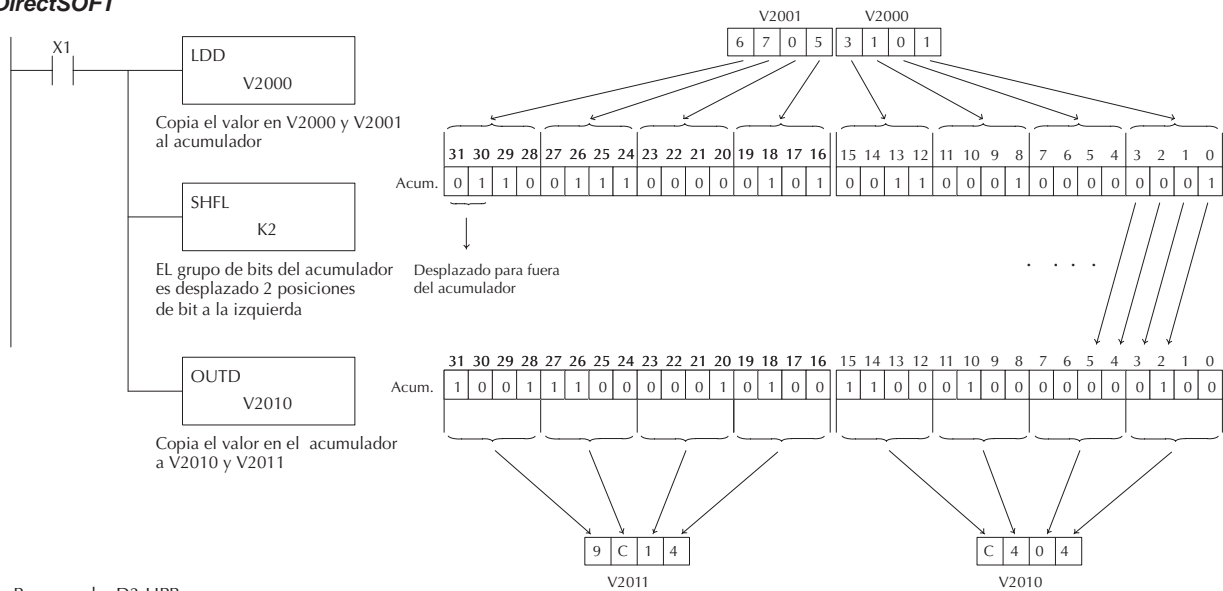
5

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 y V2001 se carga al acumulador usando la instrucción LDD. El conjunto de bits en el acumulador se desplaza 2 bits a la izquierda usando la instrucción SHFL. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT



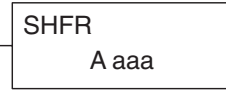
Programador D2-HPP

\$ STR	→	B 1	ENT						
SHFT	L ANDST	D 3	D 3	→	C 2	A 0	A 0	A 0	ENT
SHFT	S RST	SHFT	H 7	F 5	L ANDST	→	C 2	ENT	
GX OUT	SHFT	D 3	→	C 2	A 0	B 1	A 0	ENT	

La instrucción Shift Right (SHFR)

DS5	Usado
HPP	Usado

SHFR es una instrucción de 32 bits que desplaza los bits en el acumulador un número especificado de lugares (Aaaa) a la derecha, es decir, en la dirección desde el bit más significativo al menos significativo. Las posiciones vacías se llenan con ceros y los bits que son desplazados fuera del acumulador se pierden.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Constante K	1-32

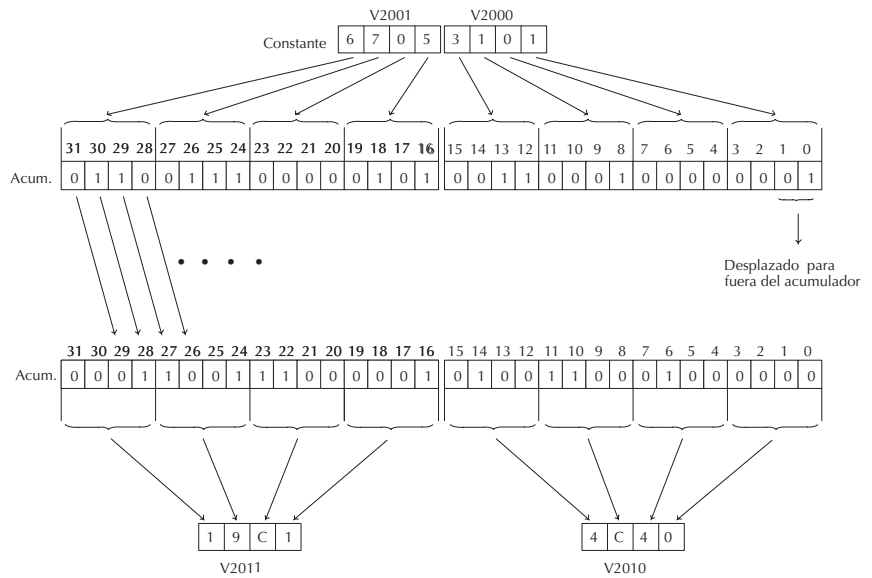
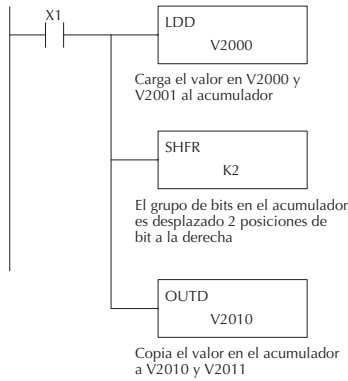
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 y V2001 se carga al acumulador usando la instrucción LDD. El conjunto de bits en el acumulador se cambia de 2 bits a la derecha usando la instrucción SHFR. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT



Programador D2-HPP

\$	→	B	ENT						
STR		1							
SHFT	L	D	D	→	C	A	A	A	ENT
	ANDST	3	3		2	0	0	0	
SHFT	S	SHFT	H	F	R	→	C	ENT	
	RST		7	5	ORN		2		
GX	SHFT	D	→	C	A	B	A	ENT	
OUT		3		2	0	1	0		

DS5	Usado
HPP	Usado

ROTL es una instrucción de 32 bits que desplaza los bits en el acumulador un número (Aaaa) especificado de lugares a la izquierda y los que se perderían se van al extremo derecho, es decir, los bits se desplazan en la dirección desde el bit menos significativo al más significativo.



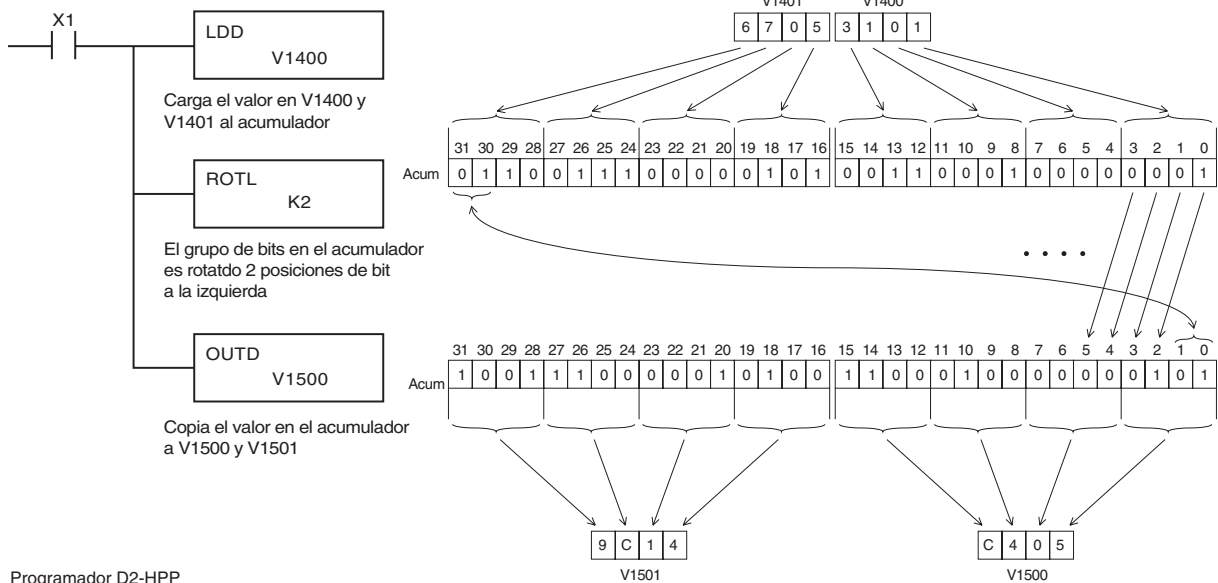
Tipo de operando de datos		Rango del DL06
..... A		aaa
Memoria V	V	Vea el mapa de memoria
Constante	K	1-32
Indicadores	Descripción	
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.	
SP70	ON cuando el valor en el acumulador es negativo.	

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El conjunto de bits en el acumulador se mueve 2 bits a la izquierda usando la instrucción ROTL. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT



Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	D	3	→	B	1	E	4	A	0	A	0	ENT
SHFT	R	ORN	O	INST#	T	MLR	L	ANDST	→	C	2	ENT				
GX	OUT	SHFT	D	3	→	B	1	F	5	A	0	A	0	ENT		

La instrucción Rotate Right (ROTR)

DS5	Usado
HPP	Usado

ROTR es una instrucción de 32 bits que desplaza los bits en el acumulador un número (Aaaa) especificado de lugares a la derecha es decir, los bits se desplazan en la dirección desde el bit más significativo al menos significativo.



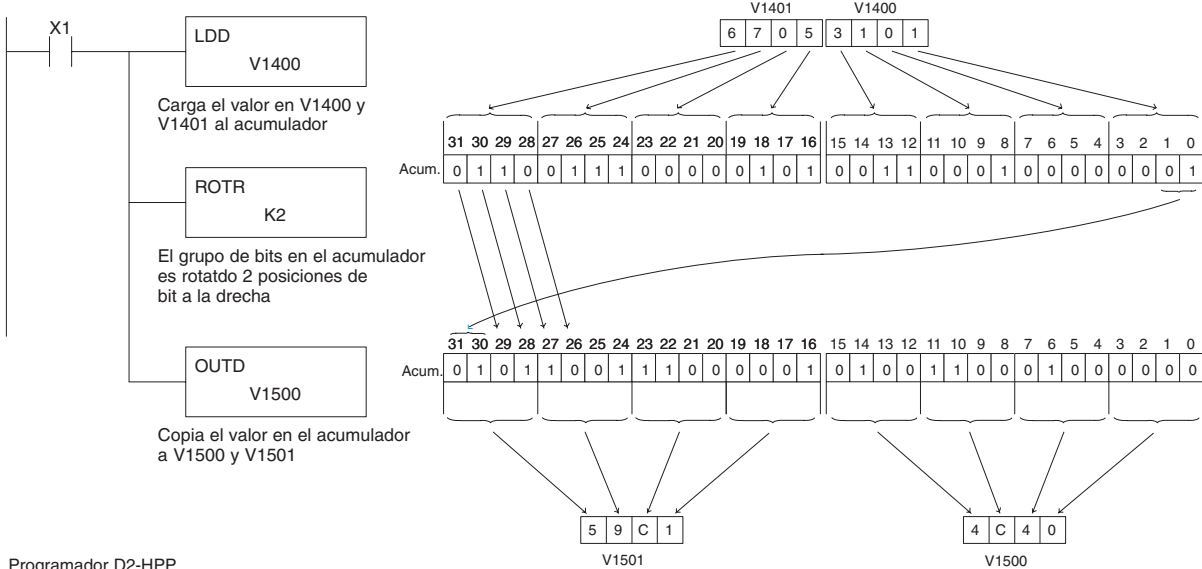
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Constante K	1-32

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

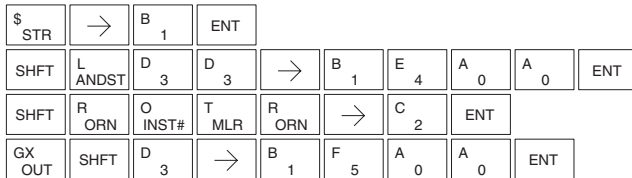
5

En el ejemplo siguiente, cuando X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. El conjunto de bits en el acumulador desplaza 2 bits a la derecha usando la instrucción ROTR. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.

DirectSOFT



Programador D2-HPP



La instrucción Encode (ENCO)

DS5	Usado
HPP	Usado

La instrucción ENCO es una instrucción de 16 bits que codifica la posición del bit en el acumulador que tiene un valor de 1 y retorna la representación binaria apropiada de 5 bits. Si el bit más significativo está en 1 (Bit 31), la instrucción ENCO colocaría el valor 1F hexadecimal (decimal 31) en el acumulador. Si el valor a ser codificado es 0000 o 0001, la instrucción colocará un cero en el acumulador. Si el valor a ser codificado tiene más de un conjunto de posiciones de bit en "1", el bit menos significativo con un "1" será codificado y SP53 se hará ON.



Indicadores	Descripción
SP53	ON cuando el valor del operando es más grande de lo que puede procesar el acumulador.

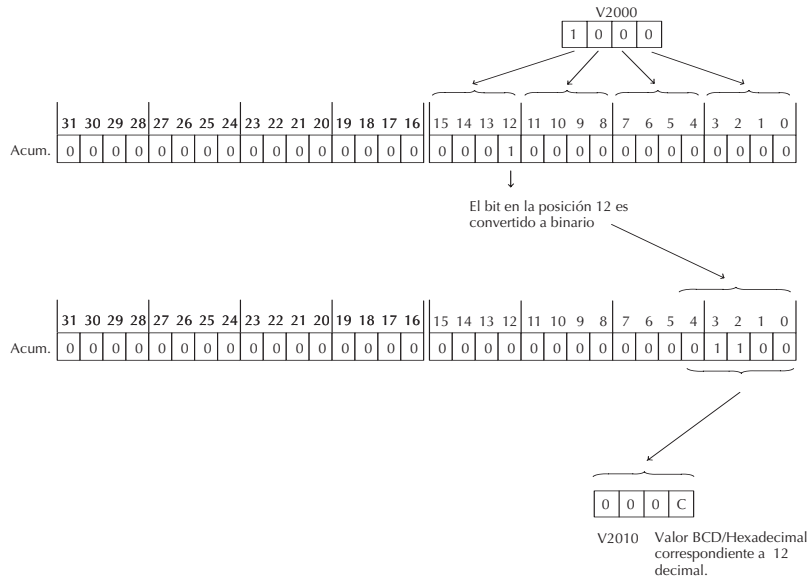
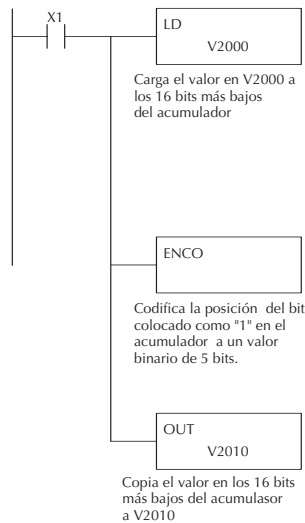
5



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor en V2000 al acumulador usando la instrucción LD. La posición del bit que está en "1" (posición 12) en el acumulador es codificada como el valor binario correspondiente usando la instrucción ENCO. El valor en los 16 bits más bajos del acumulador es copiado a V2010 usando la instrucción OUT.

DirectSOFT



Programador D2-HPP

\$ STR	→	B 1	ENT					
SHFT	L ANDST	D 3	→	C 2	A 0	A 0	A 0	ENT
SHFT	E 4	N TMR	C 2	O INST#	ENT			
GX OUT	→	SHFT	V AND	C 2	A 0	B 1	A 0	ENT

La instrucción Decode (DECO)

DS5	Usado
HPP	Usado

La instrucción DECO decodifica un valor binario de 5 bits en el rango de 0-31 (0-1F hexadecimal) en el acumulador poniendo la posición apropiada del bit en "1".



Si el acumulador contiene el valor F (hexadecimal), el bit 15 será colocado como "1" en el acumulador. Los demás bits serán 0.

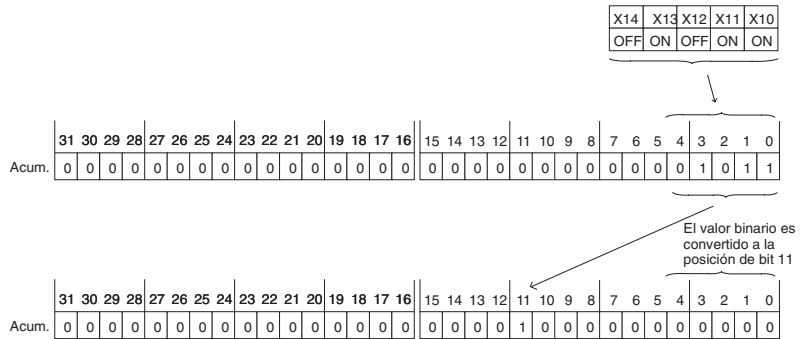
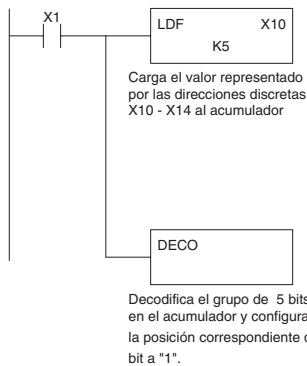
Si el valor para ser decodificado es más que 31, el número es dividido por 32 hasta que el valor sea menor que 32 y entonces el valor se decodifica.

En el ejemplo siguiente cuando X1 está ON, el valor formado por las direcciones discretas X10-X14 se carga al acumulador usando la instrucción LDF.

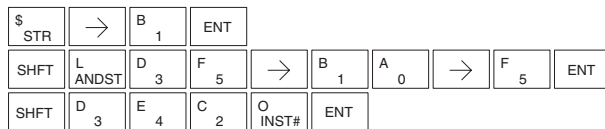
Los cinco bits en el acumulador son decodificados poniendo la posición correspondiente de bit en "1" usando la instrucción DECO.

5

DirectSOFT



Programador D2-HPP



Capítulo 5: Instrucciones de conversión de formatos

La instrucción Binary Coded Decimal (BCD)

DS5	Usado
HPP	Usado

La instrucción BCD convierte un valor binario en el acumulador al valor equivalente BCD. El resultado se va al acumulador.

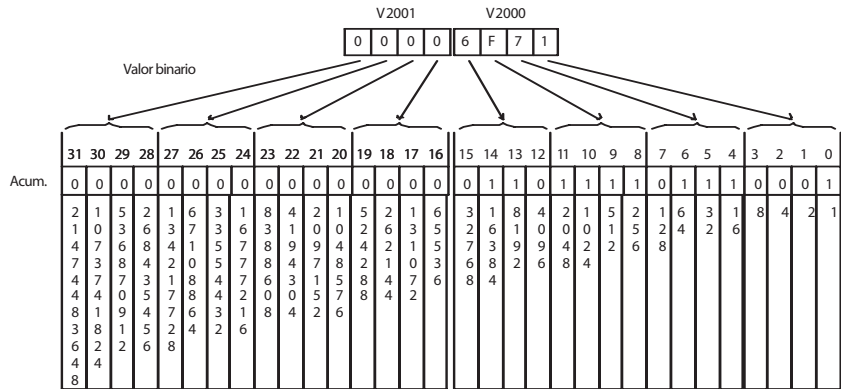
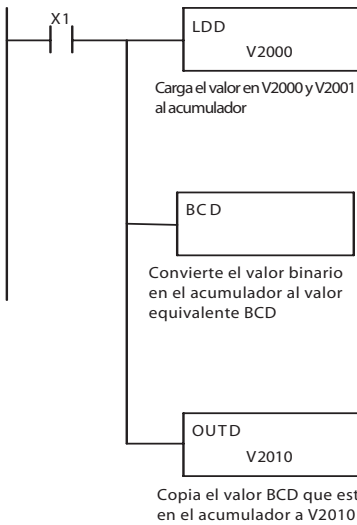


Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

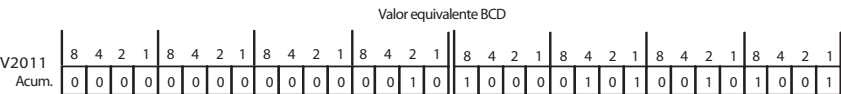
En el ejemplo siguiente, cuándo X1 está ON, el valor binario (hexadecimal) en V2000 y V2001 se carga al acumulador usando la instrucción LDD. El valor binario en el acumulador es convertido al valor equivalente BCD usando la instrucción BCD. El valor BCD en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.

5

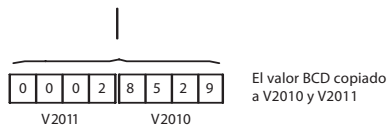
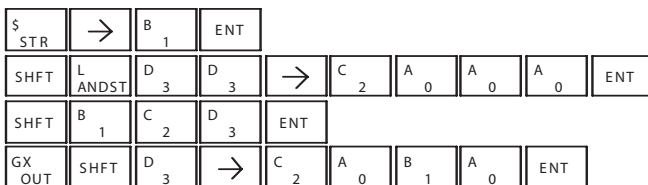
DirectSOFT



$$16384 + 8192 + 2048 + 1024 + 512 + 256 + 64 + 32 + 16 + 1 = 28529$$



Programador D2-HPP



La instrucción Invert (INV)

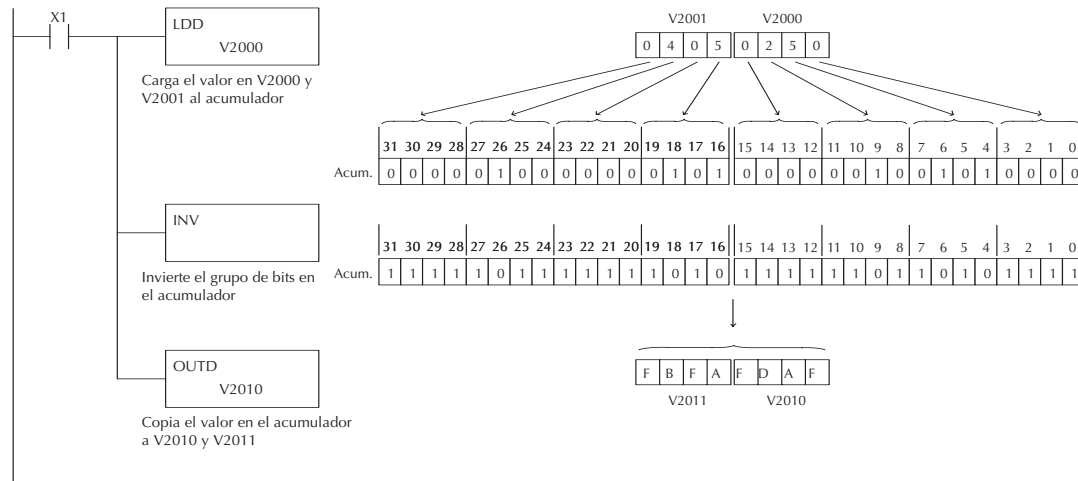
DS5	Usado
HPP	Usado

La instrucción INV invierte o toma el complemento de uno del valor de 32 bits en el acumulador. El resultado se va al acumulador. Esto es, cada bit que es cero pasa a ser uno y cada bit que es uno pasa a ser cero, en la misma posición de la palabra.

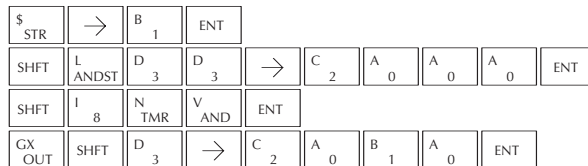


En el ejemplo siguiente, cuándo X1 está ON, el valor en V2000 y V2001 se carga al acumulador usando la instrucción LDD. El valor en el acumulador se invierte usando la instrucción INV. El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.

DirectSOFT



Programador D2-HPP



La instrucción Ten's Complement (BCDCPL)

DS5	Usado
HPP	Usado

La instrucción BCDCPL toma el complemento de 10's (BCD) del acumulador con 8 dígitos. El resultado se va al acumulador. El cálculo para esta instrucción es:



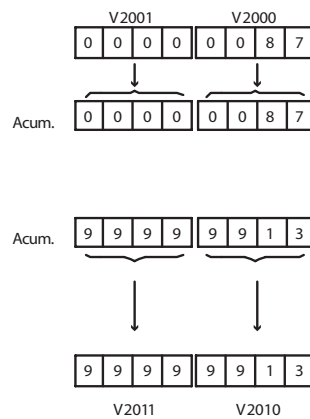
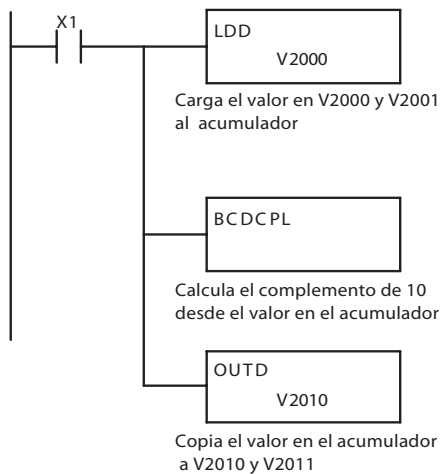
$$\begin{array}{r}
 10000000 \\
 -\text{acumulador} \\
 \hline
 \text{valor del complemento de 10}
 \end{array}$$

En el ejemplo siguiente cuando X1 está ON, el valor en V2000 y V2001 se carga a acumulador.

Se calcula entonces el complemento de 10 del acumulador con los 8 dígitos usando la instrucción BCDCPL.

El valor en el acumulador es copiado a V2010 y V2011 usando la instrucción OUTD.

DirectSOFT



Programador D2-HPP

\$	STR	→	B	1	ENT											
SHFT	L	ANDST	D	3	D	3	→	C	2	A	0	A	0	A	0	ENT
SHFT	B	1	C	2	D	3	C	2	P	CV	L	ANDST	ENT			
GX	OUT	SHFT	D	3	→	C	2	A	0	B	1	A	0	ENT		

La instrucción Binary to Real Conversion (BTOR)

DS5	Usado
HPP	Usado

La instrucción BTOR convierte un valor binario en el acumulador al formato de número real equivalente (punto flotante). El resultado se va al acumulador. El número binario y el número real pueden usar los 32 bits del acumulador.



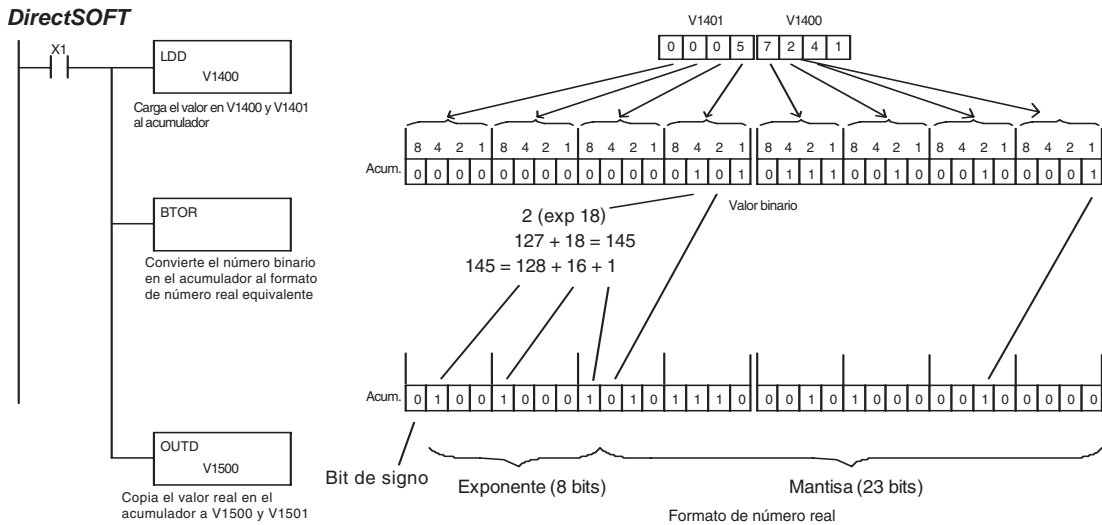
NOTA: Esta instrucción sólo trabaja con valores binarios. No trabajará con valores decimales con signo.

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

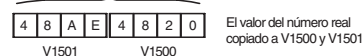
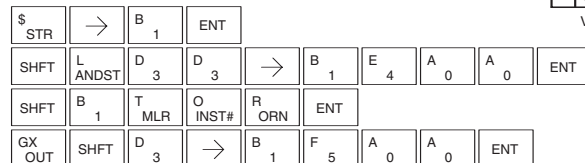
5

En el ejemplo siguiente, cuando X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. La instrucción BTOR convierte el valor binario en el acumulador al formato real equivalente del número. El peso binario del MSB (El bit más significativo) es convertido al exponente real del número sumándolo a 127 (decimal). Luego los bits restantes son copiados a la mantisa como es mostrado en el diagrama. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD.

El programador D2-HPP mostraría el valor binario en V1500 y V1501 como un valor hexadecimal.



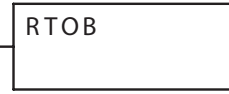
Programador D2-HPP



La instrucción Real to Binary Conversion (RTOB)

DS5	Usado
HPP	Usado

La instrucción RTOB convierte un número real en el acumulador a un valor binario. El resultado se va al acumulador. El número binario y el número real pueden usar los 32 bits del acumulador. El valor real es truncado a un número entero.



NOTA₁: La porción decimal del resultado será truncada.

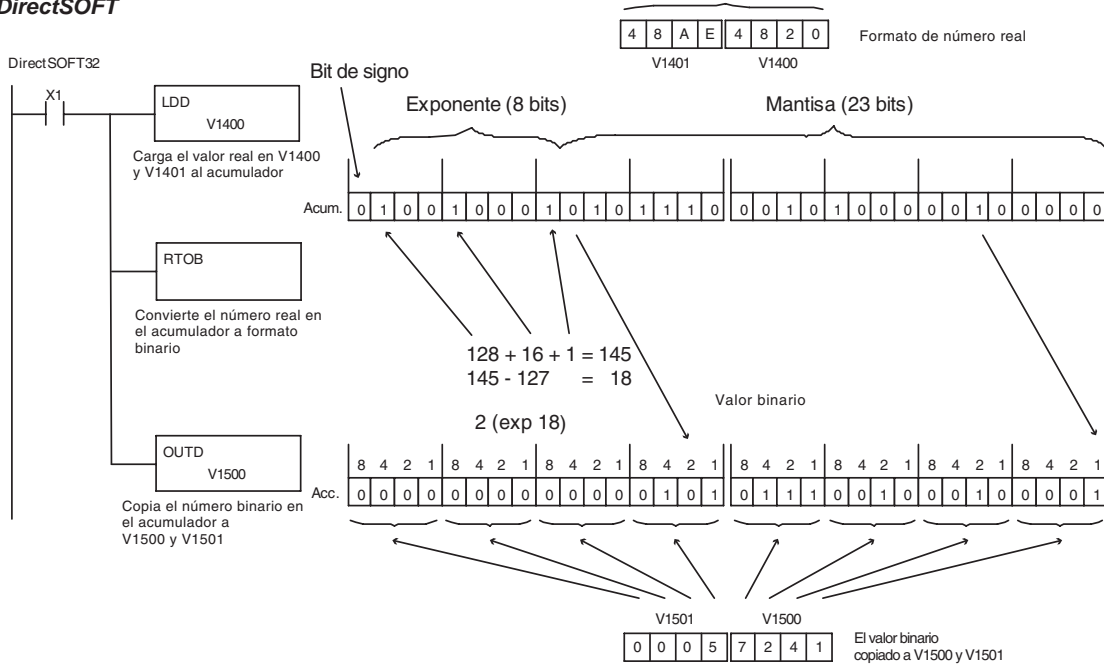
NOTA₂: si el número real es negativo, se torna en un valor decimal con signo.

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP73	ON cuando una suma o resta con signo resulta en un bit de signo incorrecto.
SP75	ON cuando un número no puede ser convertido a binario.

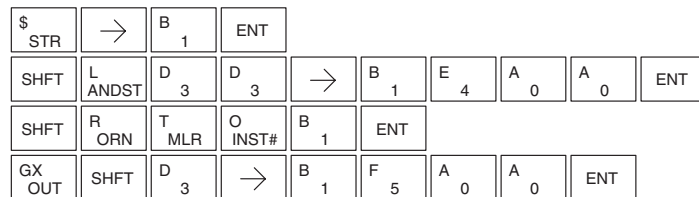
5

En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 y V1401 se carga al acumulador usando la instrucción LDD. La instrucción de RTOB convierte el valor real en el acumulador al formato equivalente de numeración binaria. El valor en el acumulador es copiado a V1500 y V1501 usando la instrucción OUTD. El programador D2-HPP mostraría el valor binario en V1500 y V1501 como un valor hexadecimal.

DirectSOFT



Programador D2-HPP



La instrucción Radian Real Conversion (RADR)

DS5	Usado
HPP	N/A

RADR convierte el valor real del grado almacenado en el acumulador al número real equivalente en radianes. El resultado se va al acumulador.



La instrucción Degree Real Conversion (DEGR)

DS32	Usado
HPP	N/A

La instrucción DEGR convierte el valor real de radián almacenado en el acumulador al número real equivalente en grados. El resultado se va al acumulador.



Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.
SP72	ON cuando el valor en el acumulador es un número de punto flotante inválido.
SP73	ON cuando una suma o resta con signo resulta en un bit de signo incorrecto.
SP75	ON cuando un número no puede ser convertido a formato binario (antes era SP74)

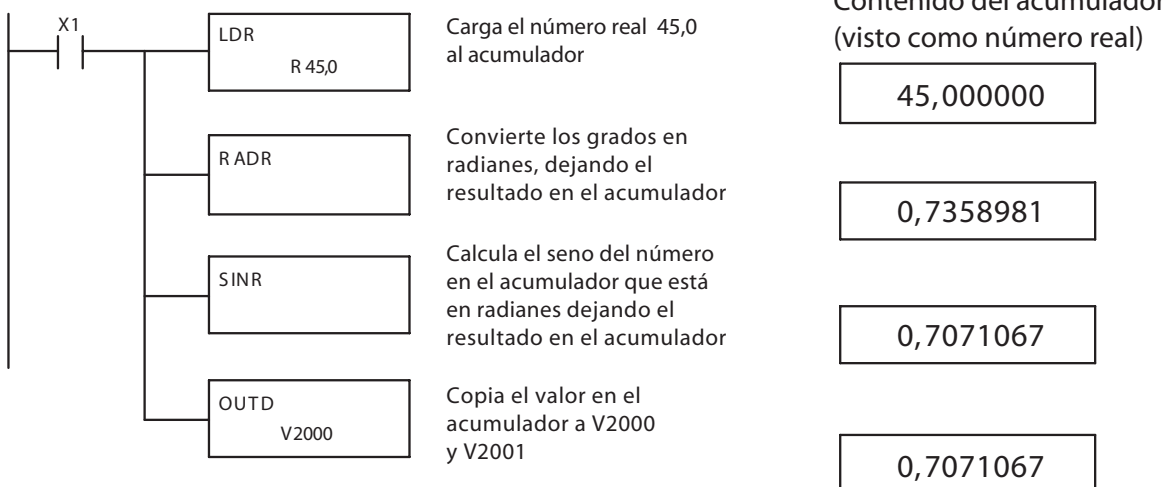
Las dos instrucciones descritas encima convierten números reales en el acumulador del formato de grado al formato de radián, y vice-versa. En el formato de grado, un círculo contiene 360 grados. En el formato de radián, un círculo contiene aproximadamente 6.28 radianes (2π). Estos convierten números positivos y negativos reales y ángulos de más de un círculo completo. Estas funciones son muy útiles cuando son combinadas con las funciones trigonométricas trascendentales (vea la sección en instrucciones aritméticas).



NOTA: El programador D2-HPP no permite entrar números reales con conversión automática al formato de 32 bits IEEE. Usted debe usar DirectSOFT para entrar números reales, usando la instrucción LDR.

El ejemplo siguiente calcula el seno de 45 grados. Ya que las funciones trascendentales operan sólo con números reales, se hace un LDR (Cargue real) 45,0. Las funciones trigonométricas operan sólo en radianes, así que debemos convertir los grados a radianes usando la instrucción RADR. Después de usar la instrucción SINR (Seno Real), se usa la instrucción OUTD para copiar el resultado del acumulador a la memoria V. El resultado es de 32 bits de ancho, requiriendo el OUTD para moverlo.

DirectSOFT



La instrucción ASCII a HEX (ATH)

DS5	Usado
HPP	N/A

La instrucción ATH convierte una tabla de valores de ASCII a una tabla de valores hexadecimales. Los valores de ASCII son dos dígitos y sus equivalentes hexadecimales solamente son un dígito. Esto significa que una tabla ASCII de cuatro direcciones de memoria V sólo requiere dos direcciones de memoria V para la tabla equivalente hexadecimal. Los parámetros de la función son cargados en el Stack del acumulador y en el acumulador por dos instrucciones adicionales.



Abajo están listados los pasos necesarios de programar una función de tabla de ASCII a hexadecimal.

El ejemplo en la página siguiente muestra un programa para la función de conversión ASCII a hexadecimal.

- Paso 1: — Cargue el número de direcciones de memoria V para la tabla ASCII en el primer nivel del Stack del acumulador.
- Paso 2: — Cargue la dirección de memoria V de inicio para la tabla ASCII en el acumulador. Este parámetro debe ser un valor en hexadecimal.
- Paso 3: — Especifique la dirección de memoria V (Vaaa) de inicio para la tabla hexadecimal en la instrucción de ATH.

Tipo de operando de datos	Rango del DL06
	aaa
Memoria V V	Vea el mapa de memoria

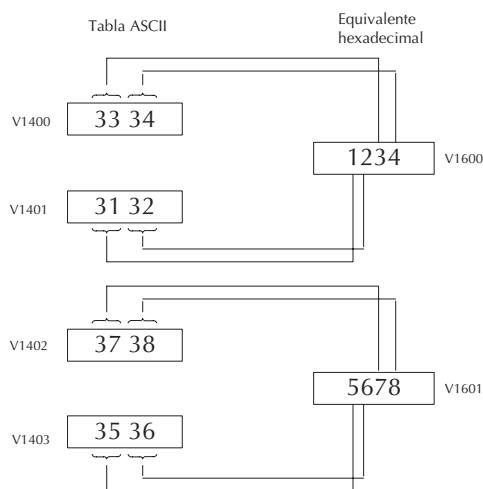
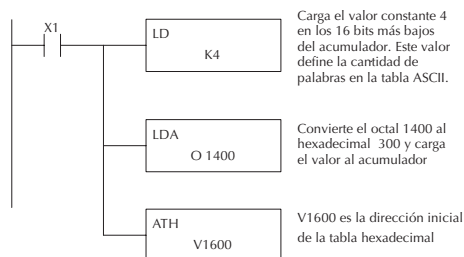
Indicadores	Descripción
SP53	ON cuando el valor del operando es más grande de lo que puede procesar el acumulador.

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

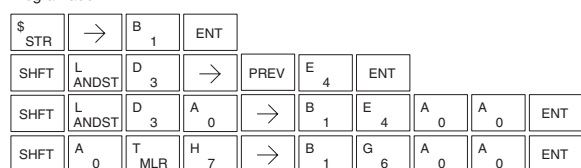
En el ejemplo en la página siguiente, cuándo X1 está ON la constante (K4) es cargada en el acumulador usando la instrucción LD y se colocará en el primer nivel del Stack del acumulador cuando se ejecuta la próxima instrucción LD. La localización de inicio para la tabla ASCII (V1400) es cargada en el acumulador usando la instrucción LDA. La dirección inicial para la tabla hexadecimal (V1600) es especificada en la instrucción ATH. La tabla de abajo lista valores válidos ASCII para la conversión ATH.

Valores ASCII válidos para la conversión ATH			
Valor ASCII	Valor hexadecimal	Valor ASCII	Valor hexadecimal
30	0	38	8
31	1	39	9
32	2	41	A
33	3	42	B
34	4	43	C
35	5	44	D
36	6	45	E
37	7	46	F

DirectSOFT



Programador D2-HPP

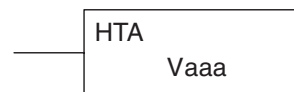


La instrucción HEX a ASCII (HTA)

DS5	Usado
HPP	N/A

La instrucción HTA convierte una tabla de valores hexadecimales a una tabla especificada de valores ASCII. Los valores hexadecimales son de un dígito y sus equivalentes ASCII son de dos dígitos.

Esto significa que una tabla hexadecimal de dos direcciones de memoria V requeriría cuatro direcciones de memoria V para la tabla equivalente de ASCII. Los parámetros de la función son cargados en el Stack del acumulador y el acumulador por dos instrucciones adicionales. Abajo están listados los pasos necesarios para programar la función de transformación de la tabla hexadecimal a ASCII. El ejemplo en la página siguiente muestra un programa para la función de conversión hexadecimal a ASCII.



- Paso 1: Cargue el número de direcciones de memoria V en la tabla hexadecimal al primer nivel del Stack del acumulador.
- Paso 2: Cargue la localización de la memoria V de inicio para la tabla hexadecimal al acumulador. Este parámetro debe ser un valor hexadecimal.
- Paso 3: Especifique la localización de memoria V (Vaaa) de inicio para la tabla ASCII en la instrucción HTA.

Sugerencia: — Se puede usar la instrucción LDA para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria, para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

Tipo de datos del operando	Rango en el DL06
	aaa
Memoria V V	Vea el mapa de memorias

La instrucción Segment (SEG)

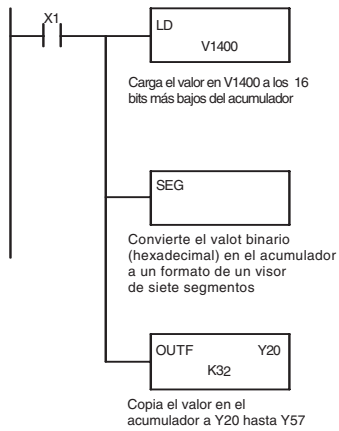
DS5	Usado
HPP	Usado

La instrucción SEGMENT convierte un valor hexadecimal de 4 dígitos en el acumulador a un formato de visor de 7 segmentos. El resultado se va al acumulador.

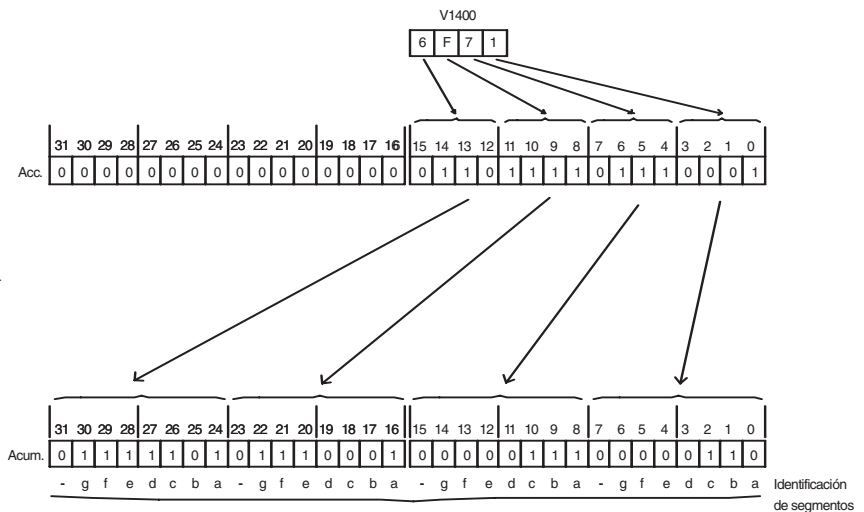


En el ejemplo siguiente, cuándo X1 está ON, el valor en V1400 se carga en los 16 bits más bajos del acumulador usando la instrucción LD. El valor binario (hexadecimal) en el acumulador es convertido a un formato de siete segmentos usando la instrucción SEGMENT. El modelo de bits en el acumulador es copiado a Y20-Y57 usando la instrucción OUTF.

DirectSOFT



Identificación de segmentos



Y57	Y56	Y55	Y54	Y53	Y24	Y23	Y22	Y21	Y20
OFF	ON	ON	ON	ON			OFF	OFF	ON	ON	OFF



Programador D2-HPP

\$	STR	→	B	1	ENT										
L	ANDST	D	3	→	B	1	E	4	A	0	A	0	ENT		
SHFT	S	RST	SHFT	E	4	G	6	ENT							
GX	OUT	SHFT	F	5	→	C	2	A	0	→	D	3	C	2	ENT

La instrucción Gray Code (GRAY)

DS5	Usado
HPP	Usado

La instrucción GRAY convierte un valor de código Gray de 16 bits a un valor BCD. La conversión BCD requiere 10 bits del acumulador. Los 22 bits superiores son colocados en "0". Esta instrucción está diseñada para uso con aparatos (típicamente encoders) que usan el código Gray, como los encoders absolutos.



La instrucción GRAY convertirá directamente un número de código GRAY a un número BCD para aparatos que tienen una resolución de 512 o 1024 conteos por revolución. Si un aparato que tiene una resolución de 360 conteos por revolución lo deberá ser usada debe restar un valor BCD de 76 del valor convertido para obtener el resultado apropiado. Para un aparato que tiene una resolución de 720 conteos por revolución usted debe restar un valor BCD de 152.

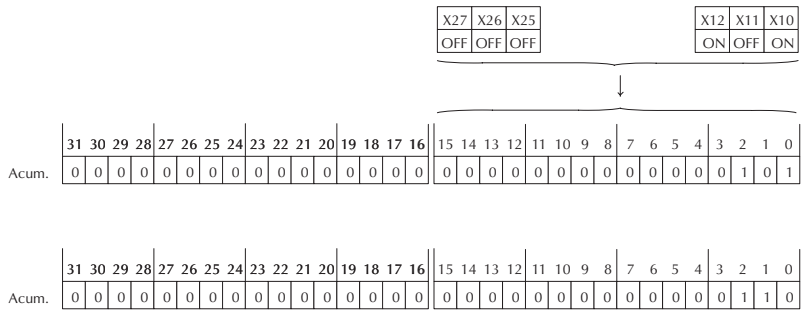
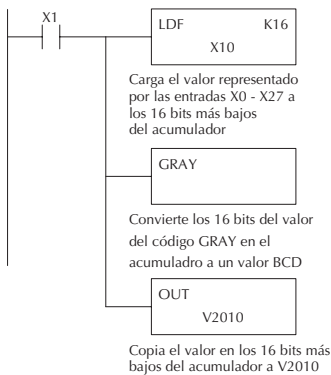
Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

En el ejemplo siguiente, cuándo X1 está ON el valor binario representado por X10-X27 es cargado al acumulador usando la instrucción LDF. El valor del código GRAY en el acumulador es convertido a BCD usando la instrucción GRAY. El valor en los 16 bits más bajos del acumulador es copiado a V2010.

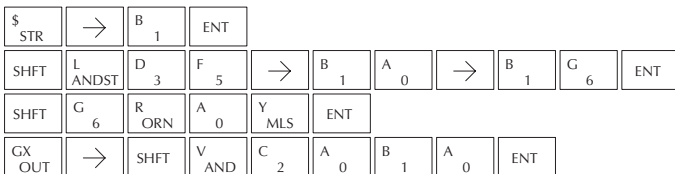


NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

DirectSOFT



Programador D2-HPP

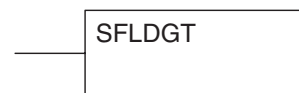


1000000001	1022
1000000000	1023

La instrucción Shuffle Digits (SFLDGT)

DS5	Usado
HPP	Usado

La instrucción SFLDGT baraja un máximo de 8 dígitos re- arreglándolos en una orden especificada. Esta función requiere que los parámetros sean cargados al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios para usar la función SFLDGT.



Paso 1: Cargue el valor (dígitos) para ser barajados en el primer nivel del Stack del acumulador.

Paso 2: Cargue la orden en que los dígitos serán barajados en el acumulador.

Paso 3: Use la instrucción SFLDGT.



NOTA: Si el número especificado para especificar el orden contiene un 0 o 9 hasta F la posición correspondiente será colocada como 0.

5

Indicadores	Descripción
SP63	ON cuando el resultado de la instrucción hace que el valor en el acumulador sea 0.
SP70	ON cuando el valor en el acumulador es negativo.

Diagrama de bloque de barajada de dígitos

Hay un máximo de 8 dígitos que se pueden barajar.

Las posiciones de los bits en el primer nivel del Stack del acumulador definen los dígitos a ser barajados.

Ellos corresponden a las posiciones de bits en el acumulador que define la orden que los dígitos se barajarán.

Los dígitos se barajan y el resultado se va al acumulador.

Dígitos a ser barajados
(en el primer nivel del stack)

9	A	B	C	D	E	F	0
---	---	---	---	---	---	---	---



1	2	8	7	3	6	5	4
---	---	---	---	---	---	---	---

Orden especificada (en el acumulador)

Posiciones de bits 8 7 6 5 4 3 2 1

B	C	E	F	0	D	A	9
---	---	---	---	---	---	---	---

Resultado (en el acumulador)

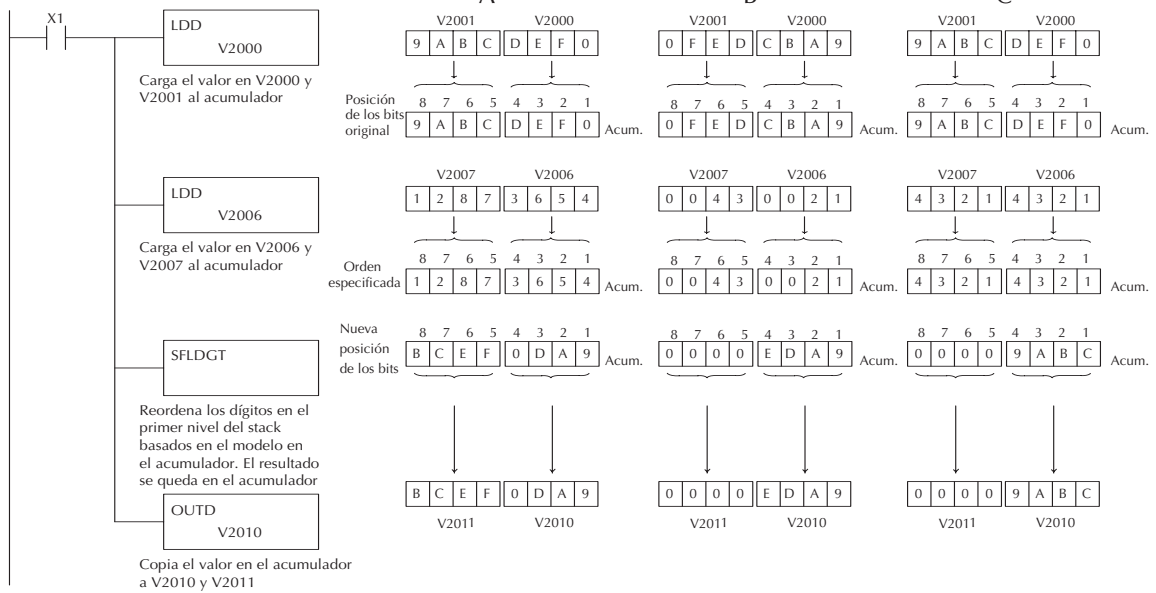
En el ejemplo siguiente cuando X1 está ON, el valor en el primer nivel del Stack del acumulador será reorganizado en la orden especificada por el valor en el acumulador.

El ejemplo A muestra cómo los dígitos siendo barajados trabajan, cuándo 0 o 9 hasta F no se usan, cuándo se especifica la orden que los dígitos deben ser barajados. También, no hay números duplicados en la orden especificada.

El ejemplo B muestra cómo los dígitos siendo barajados trabajan, cuando se usa un 0 o de 9 hasta F, cuándo se especifica la orden que los dígitos deberán ser barajados. Note que cuando se ejecuta la instrucción SFLDGT, las posiciones de bits en la primera dirección del Stack que tenía un 0 o de 9 hasta F correspondiente en el acumulador (orden especificado) son puestos a "0".

El ejemplo C muestra cómo los dígitos siendo barajados trabajan, cuando se usan números duplicados, especificando la orden en que los dígitos deberán ser barajados. Note que cuando se ejecuta la instrucción SFLDGT, se usa el número duplicado más significativo en la orden especificada en el resultado.

DirectSOFT



Programador D2-HPP

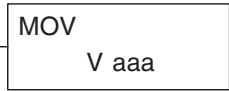
\$	STR	→	B	1	ENT							
SHFT	L	ANDST	D	3	D	3	→	C	A	A	A	ENT
SHFT	L	ANDST	D	3	D	3	→	C	A	A	G	ENT
SHFT	S	RST	SHFT	F	5	L	ANDST	D	G	T	MLR	ENT
GX	OUT	SHFT	D	3	→	C	A	B	A			ENT

Instrucciones de tablas (Tablas son simplemente memorias consecutivas)

La instrucción Move (MOV)

DS5	Usado
HPP	Usado

La instrucción MOV copia los valores de una tabla de memoria V a otra tabla de memoria V de una misma longitud. La tabla original queda intacta. Los parámetros de función se cargan en el primer nivel del Stack del acumulador y en el acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios para programar la función MOV.



- Paso 1 Cargue el número de direcciones de memoria V a ser copiados al primer nivel del Stack del acumulador. Este parámetro es un valor hexadecimal (máx. kFFE, 4096 decimal)
- Paso 2 Cargue la dirección inicial de memoria V de las direcciones a ser copiadas al acumulador. Este parámetro es un valor hexadecimal.
- Paso 3 Use la instrucción MOV que especifica donde está la dirección inicial de memoria V (Vaaa) en la tabla de destino.

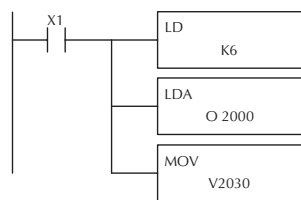
Sugerencia: — Para parámetros que necesitan valores hexadecimales cuando se refieren a direcciones de memoria, se puede usar la instrucción LDA para convertir una dirección de octal al equivalente hexadecimal y cargar el valor en el acumulador.

Tipo de operando de datos		Rango del DL06	
		aaa	
Memoria V	V	Vea el mapa de memoria	
Puntero	P	Vea el mapa de memoria	
Indicadores		Descripción	
SP53		ON cuando el valor del operando es más grande que lo que el acumulador puede usar	

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constantee (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en el primer nivel del Stack después que se ejecuta la instrucción LDA. La dirección octal 2000 (V2000), que es la dirección inicial para la tabla fuente, se carga al acumulador. La dirección de la tabla de destino (V2030) es especificada en la instrucción MOV.

Esta es la única instrucción que permite escribir datos a memoria no-volátil.

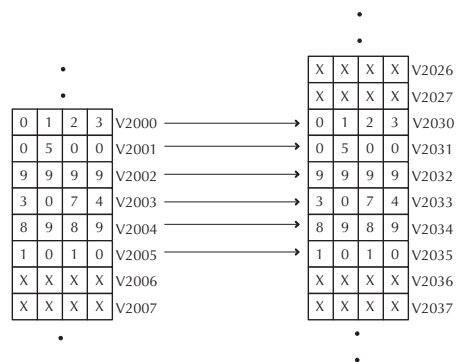
DirectSOFT



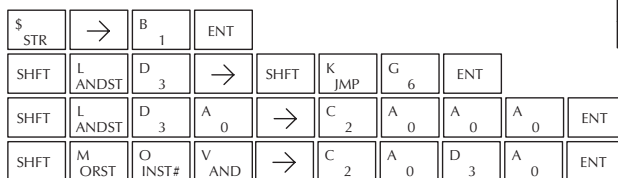
Carga el valor constante 6 hexadecimal en los 16 bits más bajos del acumulador

Convierte el octal 2000 al hexadecimal 400 y carga el valor al acumulador

Copia la tabla especificada a una tabla que comienza en la dirección V2030



Programador D2-HPP

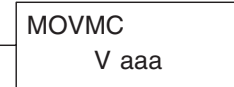


La instrucción Move Memory Cartridge (MOVMC)

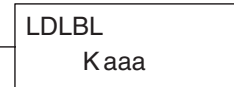
La instrucción Load Label (LDLBL)

DS5	Usado
HPP	Usado

Las instrucciones MOVMC y LDLBL son usadas para copiar los datos en la memoria ladder de un programa a memoria V. La instrucción LDLBL se usa con la instrucción MOVMC cuando se copian los datos de memoria ladder en un programa a la memoria V.



Para copiar los datos de la memoria ladder a la memoria V, se cargan los parámetros de la función a los primeros dos niveles del Stack del acumulador y a acumulador con dos instrucciones adicionales.



Esta instrucción está relacionada con las instrucciones DLBL, ACON y NCON. Le recomendamos que vea las definiciones de estas instrucciones.

Abajo están listados los pasos necesarios para programar las funciones MOVMC y LDLBL.

- Paso 1: Cargue el número de palabras a ser copiado en el segundo nivel del Stack del acumulador.
- Paso 2: Cargue el desplazamiento del área de data label en la memoria ladder y el comienzo del bloque de memoria V en el primer nivel del Stack .
- Paso 3: Cargue la etiqueta de datos de fuente (LDLBL Kaaa) al acumulador cuando se copian los datos de memoria ladder a la memoria V. Esto es la fuente de la localización del valor.
- Paso 4: la instrucción MOVMC que especifica el destino en la memoria V (Vaaa). Esto es el destino de copia.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria



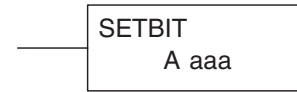
NOTA: Para más información sobre LDLBL, vea la página 5-187.



ADVERTENCIA: El desplazamiento para este uso de la instrucción comienza en 0, pero puede ser cualquier número que no dé lugar a datos fuera del área de datos de la fuente que es copiada en la tabla de destino. Cuando un desplazamiento está fuera de los límites de la información de la fuente, entonces serán transferidos valores desconocidos de datos en la tabla de destino.

La instrucción SETBIT

DS5	Usado	La instrucción SETBIT pone un solo bit en "1" dentro de un rango de direcciones de memoria V.
HPP	Usado	



La instrucción RSTBIT

DS5	Usado	La instrucción RSTBIT coloca un solo bit en "0" en un rango de localizaciones de memoria V.
HPP	Usado	



La descripción siguiente se aplica a las instrucciones SETBIT y RSTBIT.

- Paso 1: Cargue la longitud de la tabla (el número de direcciones de memoria V) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 hasta FF, que es 255 decimal.
- Paso 2: Cargue la dirección inicial de memoria V de la tabla al acumulador. Este parámetro debe ser un valor hexadecimal. Usted puede usar la instrucción LDA para convertir una dirección octal a hexadecimal.
- Paso 3: Coloque la instrucción SETBIT o RSTBIT. Esto especifica la referencia del número del bit que usted quiere hacer "1" o "0". El número del bit está en octal, y el primer bit en la tabla es el número "0".

Sugerencia: — Recuerde que cada dirección de memoria V contiene 16 bits. Así, los bits de la primera palabra de la tabla se numeran de 0 a 17 octal. Por ejemplo, si la longitud de tabla es de seis palabras, entonces 6 palabras = (6x16 bits) = 96 bits (decimal) o 140 octal. El rango permisible de números de referencia de bits sería 0 a 137 octal. SP 53 se hará ON si el bit especificado está fuera del rango de la tabla.

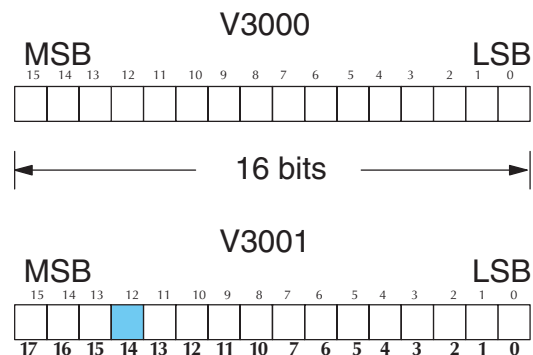
Tipo de operando de datos	Rango del DL06
Memoria V V	aaa Vea el mapa de memoria

Indicadores	Descripción
SP53	ON cuando el número de bit referenciado en el Set Bit o en el Reset Bit excede el rango de la tabla.



NOTA: Indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP.

Por ejemplo, suponga que tenemos una tabla que comienza en V3000 que tiene dos palabras, como mostrado a la derecha. Cada palabra en la tabla contiene 16 bits, o 0 a 17 en octal. Para poner el bit 12 en la segunda palabra, usamos su referencia octal (bit 14). Entonces calculamos la dirección octal de bit desde el comienzo de la tabla, de modo que $17 + 14 = 34$ octal. El programa siguiente muestra cómo poner el bit (como mostrado) en "1".



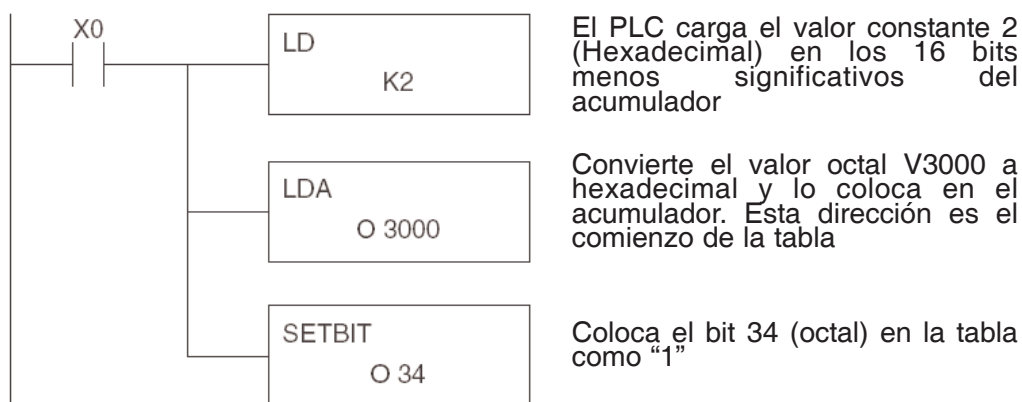
En este ejemplo usaremos la entrada X0 para disparar o activar la operación SETBIT.

Primero, cargamos la longitud de la tabla (2 palabras) al Stack del acumulador.

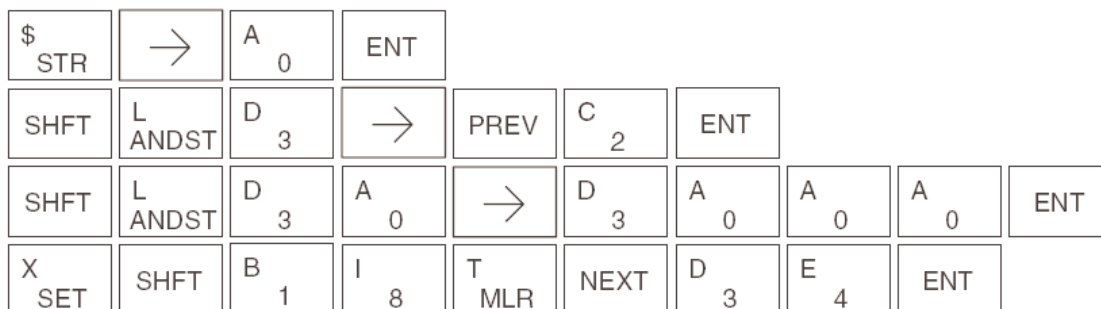
Luego, cargamos la dirección inicial en el acumulador. Ya que V3000 es un número octal lo tenemos que convertir a hexadecimal usando la instrucción LDA.

Finalmente, usamos la instrucción SETBIT (o RSTBIT) y especificamos la dirección octal del bit (bit 34), referenciada de la tabla.

DirectSOFT



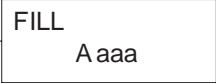
Programador D2-HPP



La instrucción Fill (FILL)

DS5	Usado
HPP	Usado

La instrucción FILL llena una tabla de hasta 255 direcciones de memoria V con un valor (Aaaa), que es una dirección de memoria V o una constante de 4 dígitos. Los parámetros de la función son cargados al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios para programar la función FILL.



Paso 1:— Cargue la cantidad de direcciones de memoria V a ser llenadas al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, de 0 a FF, que es 255 decimal.

Paso 2:— Cargue la dirección inicial de memoria V de la tabla en el acumulador. Este parámetro debe ser un valor hexadecimal.

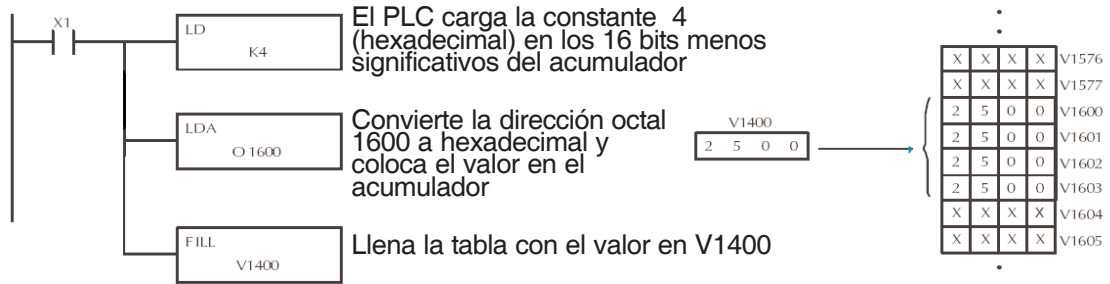
Paso 3:— Coloque la instrucción FILL que especifica el valor para llenar la tabla.

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

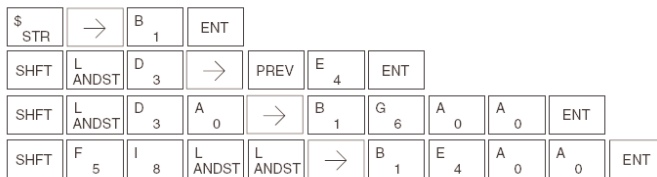
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Constante K	0–FF

Indicadores	Descripción
SP53	On si la dirección de memoria está fuera de rango

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor (K4 constante) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en el primer nivel del Stack del acumulador cuando se ejecuta la instrucción LDA. La dirección octal 1600 (V1600) es la dirección inicial de la tabla y se carga en el acumulador usando la instrucción LDA. El valor para llenar la tabla con (V1400) es especificado en la instrucción FILL.



Programador D2-HPP



La instrucción Find (FIND)

DS5	Usado
HPP	Usado

La instrucción Find se usa para buscar un valor especificado en una tabla de memoria V de hasta 255 direcciones. Los parámetros de la función FIND son cargados en el primer y segundo nivel del Stack del acumulador y del acumulador con tres instrucciones adicionales. Abajo están listados los pasos necesarios para programar la función Find.

```

FIND
  A aaa
    
```

Paso 1: Cargue la longitud de la tabla (el número de direcciones de memoria V) en el segundo nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, de 0 hasta FF, que es 255 decimal.

Paso 2: Cargue la dirección de la memoria V de inicio de la tabla en el primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal.

Paso 3: Cargue la cantidad de memorias a ser saltadas de la dirección inicial para comenzar la búsqueda. Este parámetro debe ser un valor hexadecimal.

Paso 4: Coloque la instrucción Find especificando el primer valor a ser encontrado en la tabla.

Resultado: — El resultado es expresado en hexadecimal y muestra cuantas memorias V hay entre el inicio de la tabla y la memoria encontrada. El desvío de la dirección inicial a la primera dirección de la memoria V que contiene el valor de búsqueda es vuelto al acumulador. SP53 se pondrá ON si se especifica una dirección fuera de la tabla en el desvío, o el valor no es encontrado. Si el valor no se encuentra serán colocados 0s en el acumulador.

Sugerencia: — Para parámetros que requieran valores hexadecimales cuando se refieran a direcciones de memoria se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

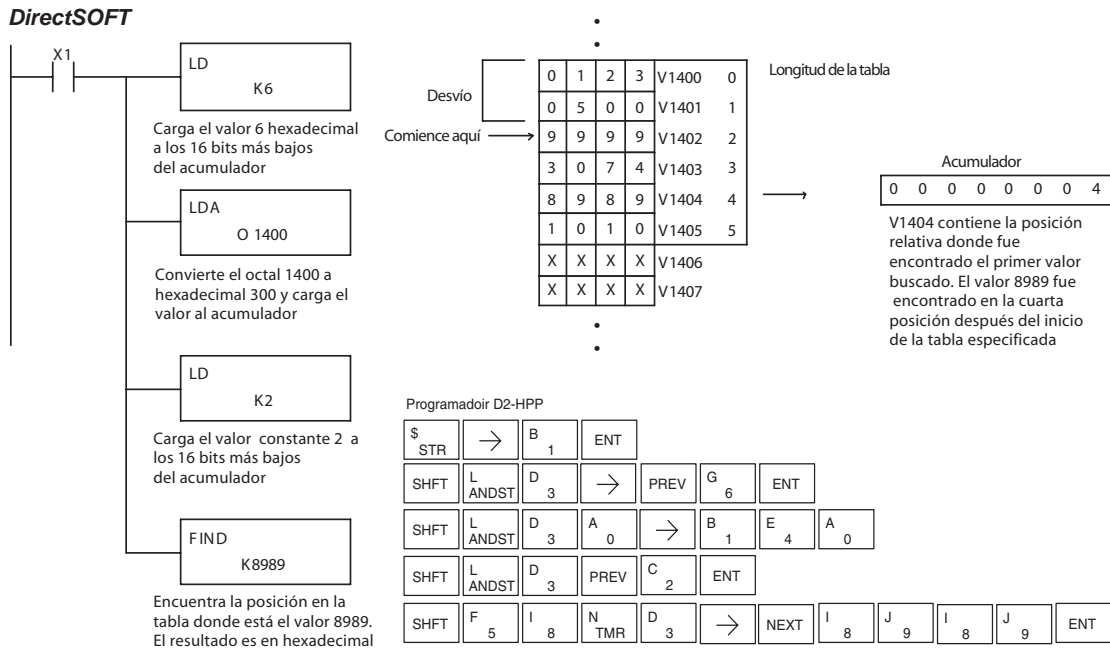
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Constante K	0–FF

Indicadores	Descripción
SP53	ON si no hay un valor en la tabla que sea igual a valor de búsqueda.



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP.

En el ejemplo siguiente, cuando X1 está encendido, el valor de la constante K6 se carga en el acumulador usando el instrucción..Este valor especifica la longitud de la tabla y se coloca en la segunda localización del stack cuando se ejecuta la instrucción siguiente. La dirección octal 1400 (V1400) es la localización inicial de la tabla y se carga en el acumulador. Este valor se pone en el primer nivel del stack del acumulador cuando se ejecuta el instrucción siguienteLD. El desplazamiento (offset K2) se carga en los 16 bits más bajos del acumulador usando la instrucción LD. El valor que se encontrará en la tabla se especifica en el instrucción FIND. Si se encuentra un valor igual al valor de la búsqueda, el desplazamiento (desde el inicio de la tabla) donde se encuentra el valor residirá en el acumulador.



La instrucción Find Greater Than (FDGT)

La instrucción FDGT se usa para buscar la primera ocurrencia de un valor en una tabla de memoria V que es más grande que el valor (Aaaa) especificado, que puede ser una dirección de memoria V o una constante de 4 dígitos. Los parámetros de la función son cargados en el primer nivel del Stack del acumulador y el acumulador por dos instrucciones adicionales. Abajo están listados los pasos necesarios para programar la instrucción FDGT.



- Paso 1: Cargue la longitud de la tabla (hasta 255 direcciones) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, de 0 hasta FF.
- Paso 2: Cargue la dirección inicial de la tabla en el acumulador. Este parámetro debe ser un valor hexadecimal.
- Paso 3: Coloque la instrucción FDGT que especifica el valor prefijado de búsqueda. El resultado es expresado en hexadecimal.

Resultado:— El desvío de la dirección inicial a la primera dirección de memoria V que contiene el valor más grande de búsqueda es vuelto al acumulador. SP53 se pondrá ON si el valor no se encuentra y se colocarán "0s" en el acumulador.

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria la instrucción LDA se puede usar para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.



NOTA: Esta instrucción no tiene una cantidad de memorias a ser saltadas tal como el que se usa para la instrucción FIND.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Constante K	0-FF

Indicadores	Descripción
SP53	ON si no hay un valor en la tabla que sea mayor que el valor de búsqueda

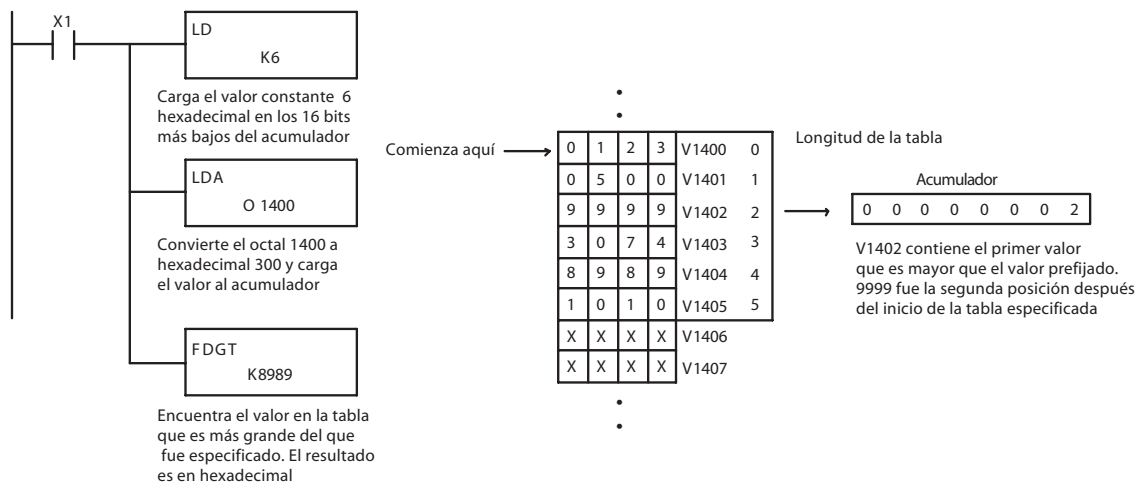


NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP. El puntero para esta instrucción comienza en 0 y se va al acumulador.

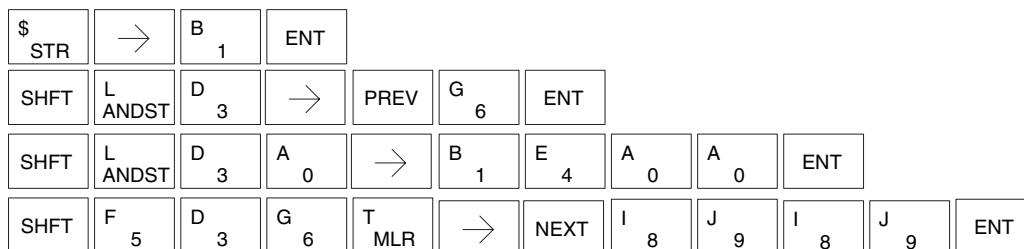
En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constantee (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en la primera localización del Stack después que se ejecuta la instrucción LDA. La dirección octal 1400 (V1400) es la dirección inicial de la tabla y se carga al acumulador. El valor prefijado de búsqueda se especifica en la instrucción Find. Si se encuentra un valor mayor que el valor prefijado de búsqueda, el desvío (de la dirección inicial de la tabla) donde el valor se localiza residirá en el acumulador.

Si no hay un valor en la tabla que sea mayor que el valor buscado, se almacena un cero en el acumulador y SP53 se hará ON.

DirectSOFT



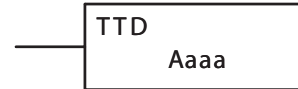
Programador D2-HPP



La instrucción Table to Destination (TTD)

DS5	Usado
HPP	Usado

La instrucción TTD copia un valor de una tabla de memoria V a una dirección de memoria V e incrementa el puntero de la tabla en 1. La primera dirección de memoria V en la tabla contiene el puntero de la tabla que indica la próxima dirección a ser copiada en la tabla. La instrucción será ejecutada una vez por barrido si el renglón es verdadero u ON. El puntero de la tabla vuelve a 1 cuando el valor se hace igual a la última dirección en la tabla. Los parámetros de la función son cargados al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales.



5

Abajo están listados los pasos necesarios para programar la instrucción TTD.

- Paso 1: Cargue la longitud de la tabla de datos (el número de direcciones de memoria V) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 hasta FF, que es 255 decimal..
- Paso 2: Cargue la dirección de memoria V inicial para la tabla en el acumulador. (Recuerde, para esta instrucción la dirección inicial de la tabla se usa como el puntero de la tabla.) Este parámetro debe ser un valor hexadecimal.
- Paso 3: Coloque la instrucción TTD especificando el destino de la memoria V (Vaaa)

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

Sugerencia:— La instrucción se ejecutará cada barrido si la lógica de la entrada está ON. Si usted no quiere que la instrucción ejecute en más que un barrido, se debe usar una instrucción one shot (PD) en la lógica de entrada.

Sugerencia: — La localización del puntero debe ser puesta al valor donde comenzará la operación de tabla. Se debe usar el relevador especial SP0 o una instrucción one shot (PD) de modo que el valor sólo sea puesto en un barrido y no afecte la operación de la instrucción.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

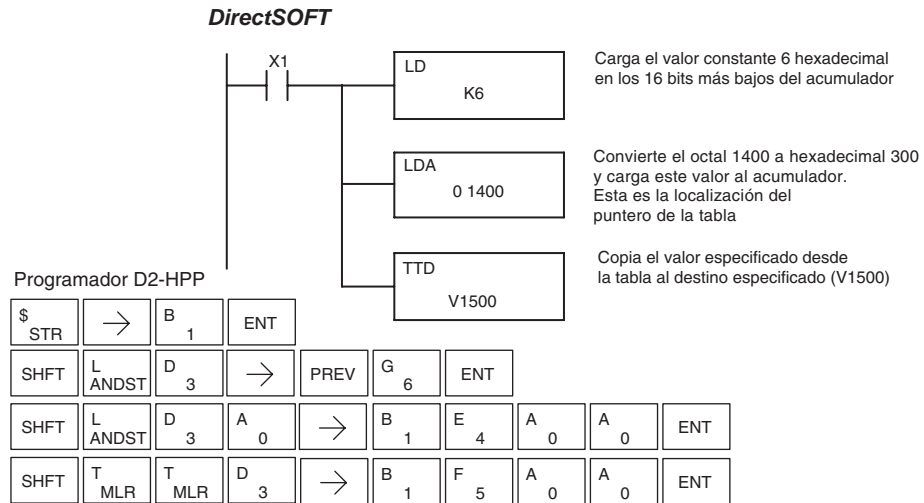
Indicadores	Descripción
SP56	ON cuando el puntero de la tabla llega a la longitud de la tabla.



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP o en el fin del barrido. El puntero para esta instrucción comienza en 0 y se hace 1 cuando se llega a la longitud de la tabla. Note que el puntero se vuelve a "1" en esta ocasión y no a 0.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constante (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en la primera localización del Stack después que se ejecuta la instrucción LDA. La dirección octal 1400 (V1400) es la dirección inicial de la tabla fuente y se carga en el acumulador.

Recuerde, V1400 se usa como la localización de un puntero, y no es realmente parte de la tabla fuente de datos. La localización de destino (V1500) es especificada en la instrucción TTD. El puntero de la tabla (V1400 en este caso) será aumentado en "1" después de cada ejecución de la instrucción TTD.



Es importante entender cómo se numeran las direcciones de la tabla. Si examina la tabla del ejemplo, usted notará que la primera dirección de datos, V1401, se usará cuando el puntero es igual a 0, y nuevamente cuando el puntero es igual a seis. ¿Por qué? Porque el puntero es sólo igual a 0 antes de la primera ejecución. De allí en adelante, incrementa de uno a seis y luego vuelve a 1.

Tabla				
V1401	0	5	0	0
V1402	9	9	9	9
V1403	3	0	7	4
V1404	8	9	8	9
V1405	1	0	1	0
V1406	2	0	4	6
V1407	X	X	X	X

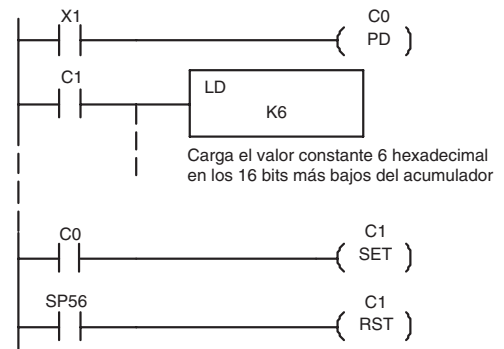
Puntero de la tabla					
0	6	0	0	0	0
1					
2					
3					
4					
5					

Destino			
X	X	X	X

También, nuestro ejemplo usa un contacto normal de entrada (X1) para controlar la ejecución. Ya que el barrido de la CPU es extremadamente rápido y el puntero aumenta en 1 automáticamente, la tabla hace el ciclo por todas las direcciones muy rápidamente.

Si esto es un problema, se tiene la opción de usar SP56 en unión con una instrucción one shot (PD) y un enclavamiento (C1 por ejemplo) para permitir a la tabla pasar por todas direcciones una vez y luego para. La lógica mostrada aquí no es necesaria, es solamente un método opcional.

DirectSOFT (Ejemplo de enclavamiento usando SP56)

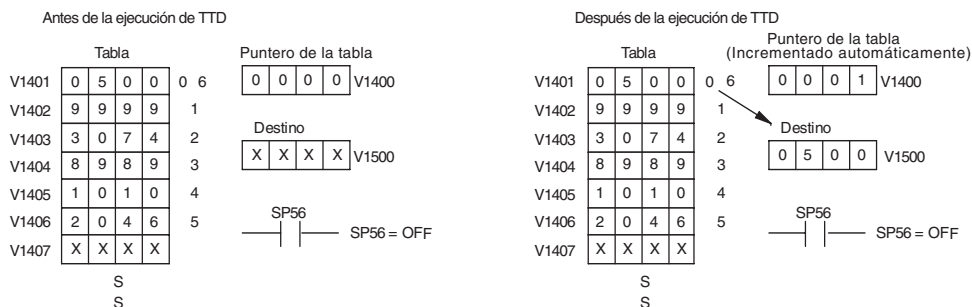


Ya que los relevadores especiales vuelven a 0 al fin del barrido, este enclavamiento debe estar justamente después de la instrucción TTD en el programa

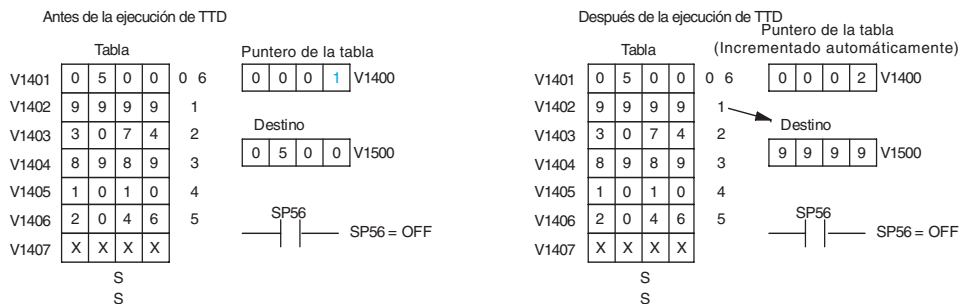
Capítulo 5: Instrucciones normales RLL - Instrucciones de tablas

El diagrama en esta página muestra los resultados barrido por barrido de la ejecución del programa del ejemplo. Note como el puntero automáticamente salta de 0 a 6 y luego comienza en 1 en vez de 0. También, note como SP56 es ON solamente hasta el fin del barrido

Barrido N

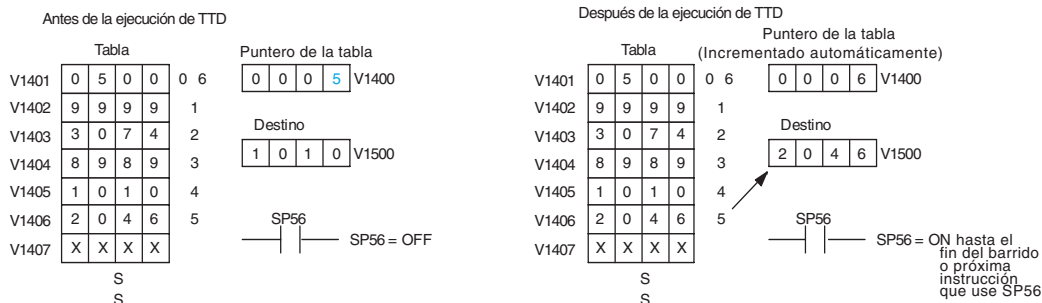


Barrido N+1

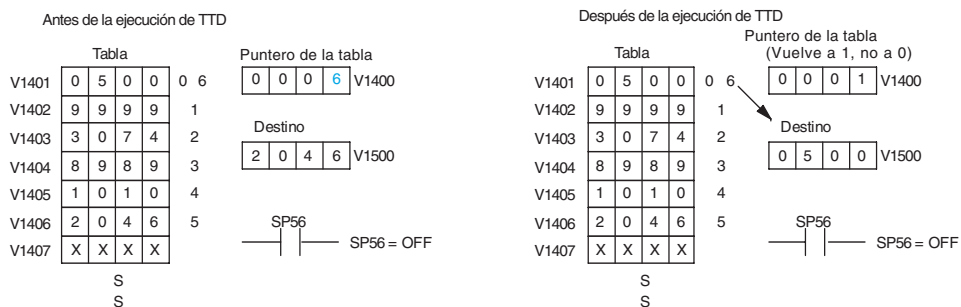


S
S
S

Barrido N+5



Barrido N+6



La instrucción Remove from Bottom (RFB)

DS5	Usado
HPP	Usado

La instrucción RFB copia un valor del fondo de una tabla de memoria V a una dirección de memoria V y decrementa un puntero de la tabla en "1". La primera localización de memoria V en la tabla contiene el puntero de la tabla que indica la próxima localización en la tabla a ser copiada. La instrucción se ejecutará una vez por barrido si el renglón es verdadero. La instrucción parará la operación cuando el puntero es igual a 0. Los parámetros de la función son cargados al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios para programar la instrucción RFB.



- Paso 1:— Cargue la longitud de la tabla (la cantidad de direcciones de memoria V) en el primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 hasta FF.
- Paso 2:— Cargue la dirección de la memoria V inicial de la tabla al acumulador. (Recuerde, para esta instrucción, la primera dirección de la tabla se usa como el puntero de la tabla). Este parámetro debe ser un valor hexadecimal.
- Paso 3:— Coloque la instrucción RFB que especifica la dirección (Vaaa) de la memoria V de destino.

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

Sugerencia:— La instrucción se ejecutará cada barrido si la lógica de entrada está ON. Si usted no quiere que la instrucción se ejecute más que un barrido, se debe usar una instrucción one shot (PD) en la lógica de entrada.

Sugerencia: — La localización del puntero debe ser puesta al valor donde comenzará la operación de tabla. Se debe usar el relevador SP0 especial o la instrucción one shot (PD) de modo que el valor sólo sea puesto en un barrido y no afecte la operación de la instrucción.

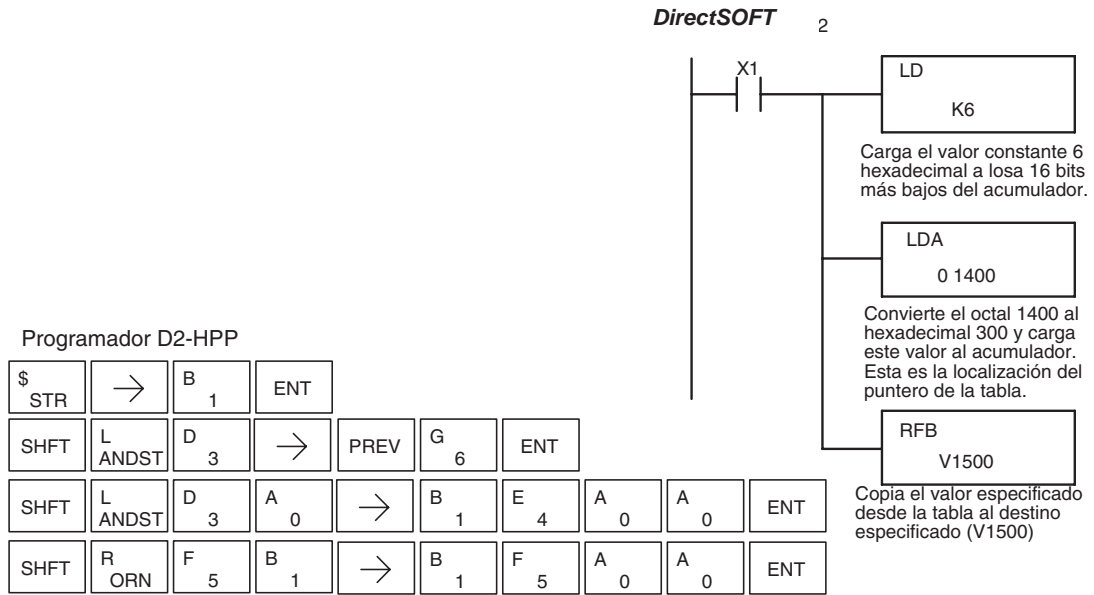
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

Indicadores	Descripción
SP56	ON cuando el valor del puntero de la tabla es igual a 0



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP o en el fin del barrido. El puntero para esta instrucción puede ser colocado para iniciar en cualquier parte de la tabla. No es colocado automáticamente. Ud. debe colocar un valor en el puntero en algún lugar del programa ladder.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constante (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en el primer nivel del Stack después que se ejecuta la instrucción LDA. La dirección octal 1400 (V1400) es la dirección inicial de la tabla fuente y se carga en el acumulador. Recuerde, V1400 se usa como la localización de un puntero y no es realmente parte de la fuente de datos de la tabla. La dirección del destino (V1500) es especificada en la instrucción TFB. El puntero de la tabla (V1400 en este caso) será decrementado en "1" después de cada ejecución de la instrucción RFB.



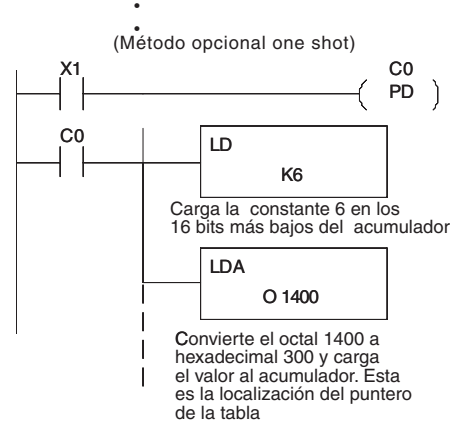
Es importante entender cómo se numeran las direcciones de la tabla. Si usted examina la tabla del ejemplo, usted verá que la primera localización de datos, V1401, se usará cuando el puntero es igual a uno. La segunda localización de datos, V1402, se usará cuando el puntero es igual a dos, etc.

V1401	0	5	0	0	1
V1402	9	9	9	9	2
V1403	3	0	7	4	3
V1404	8	9	8	9	4
V1405	1	0	1	0	5
V1406	2	0	4	6	6
V1407	X	X	X	X	

0	0	0	0	V1400
---	---	---	---	-------

X	X	X	X	V1500
---	---	---	---	-------

También, nuestro ejemplo usa un contacto normal de entrada (X1) para controlar la ejecución. Ya que el barrido de la CPU es extremadamente rápido y el puntero decrementa el valor automáticamente, la tabla recorre las direcciones en un ciclo muy rápido. Si esto es un problema para su aplicación, usted tiene la opción de usar una instrucción one shot (PD) para quitar un valor cada vez que el contacto de entrada hace la transición de OFF para ON.

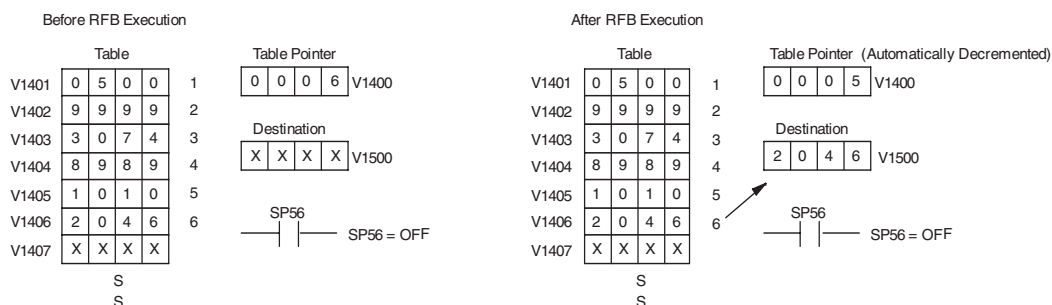


Capítulo 5: Instrucciones normales RLL - Instrucciones de tablas

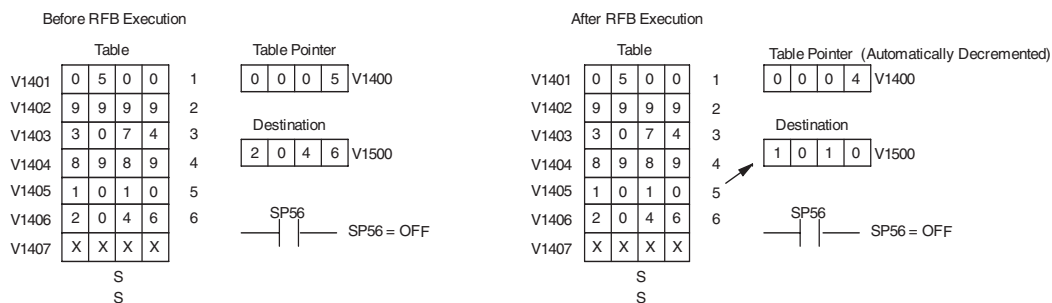
El esquema siguiente muestra los resultados de la ejecución barrido por barrido para nuestro programa del ejemplo. Advierta cómo el puntero automáticamente decrece de 6 a 0. También, note cómo SP56 es sólo ON hasta el fin del barrido.

Example of Execution

Scan N

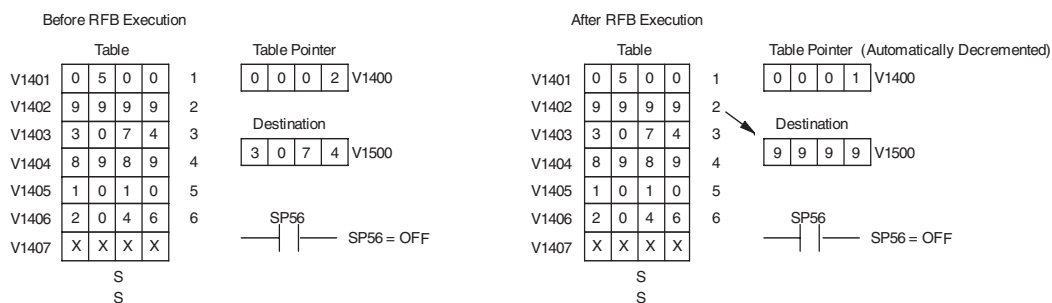


Scan N+1

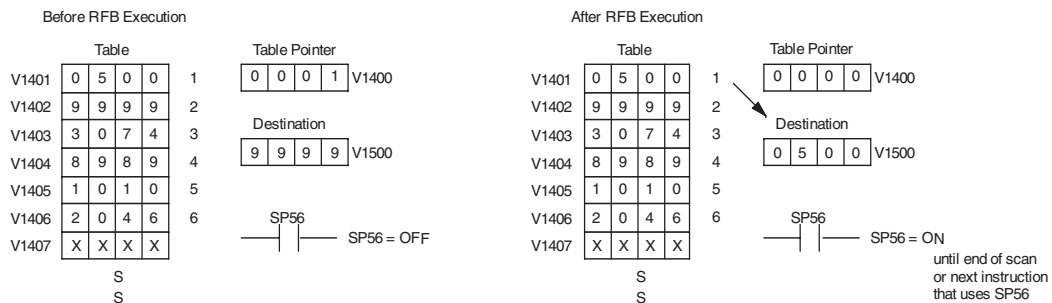


S
S
S

Scan N+4



Scan N+5



La instrucción Source a Table (STT)

DS5	Usado
HPP	Usado

La instrucción SST copia un valor de memoria V a una tabla de memoria V e incrementa el puntero de la tabla en 1. Cuando el puntero de la tabla alcanza el fin de la tabla, vuelve a 1. La primera dirección de memoria V en la tabla contiene el puntero de la tabla que indica la próxima dirección en la tabla que almacenará un valor. La instrucción se ejecutará una vez por barrido si el renglón es verdadero.



Los parámetros de la función se cargan al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios de programar la instrucción STT.

- Paso 1: Cargue la longitud de la tabla (la cantidad de direcciones de memoria V) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 a FF.
- Paso 2: Cargue la dirección inicial de la memoria V en la tabla al acumulador. (Recuerde, la dirección inicial de la tabla se usa como el puntero de la tabla.) Este parámetro debe ser un valor hexadecimal.
- Paso 3: Coloque la instrucción STT especificando la dirección (Vaaa) de la memoria V fuente. Aquí es de donde se moverá el valor.

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

Sugerencia:— La instrucción se ejecutará cada barrido si la lógica de entrada está ON. Si usted no quiere que la instrucción se ejecute por más que un barrido, se debe usar instrucción one shot (PD) en la lógica de entrada.

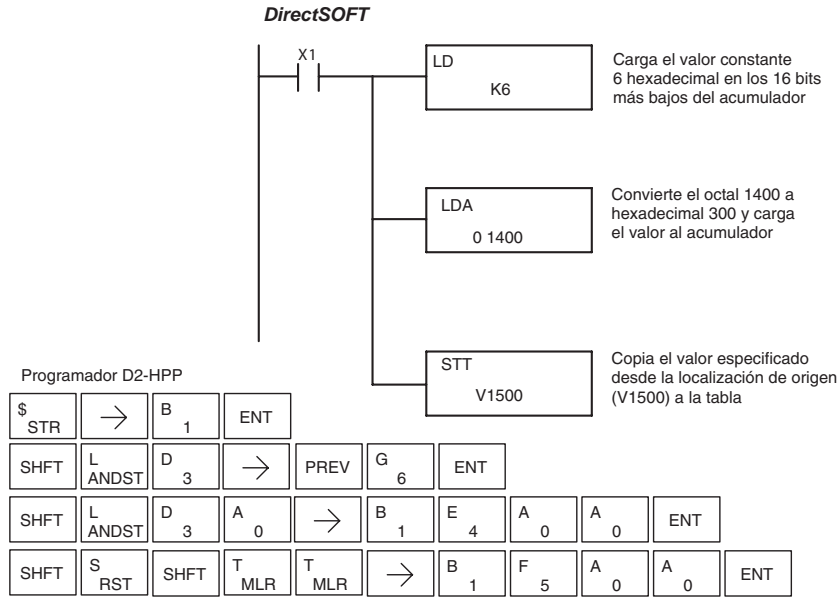
Sugerencia: — El valor contador de la tabla debe ser configurado para indicar el punto de partida de la operación. También, debe ser configurado a un valor que esté dentro de la longitud de la tabla. Por ejemplo, si la tabla es de 6 palabras, entonces el rango admisible de los valores que podrían estar en el puntero debe estar entre 0 y 6. Si el valor está fuera de este rango, los datos no se moverán. También, se debe usar una instrucción one shot (PD) de modo que el valor sólo sea puesto en un barrido y no afecte la operación de la instrucción.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Indicadores	Descripción
SP56	On cuando el puntero de la tabla es igual a la longitud de la tabla.



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP o en el fin del barrido. El puntero para esta instrucción comienza en 0 y vuelve a 1 automáticamente cuando se alcanza la longitud de

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constante (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en la primera localización del Stack después que se ejecuta la instrucción LDA. La dirección octal 1400 (V1400), que es la de la tabla de destino y el puntero de la tabla, se carga al acumulador. La dirección de la fuente de datos (V1500) es especificada en la instrucción STT. El puntero de la tabla será aumentado en "1" cada vez que se ejecuta la instrucción.

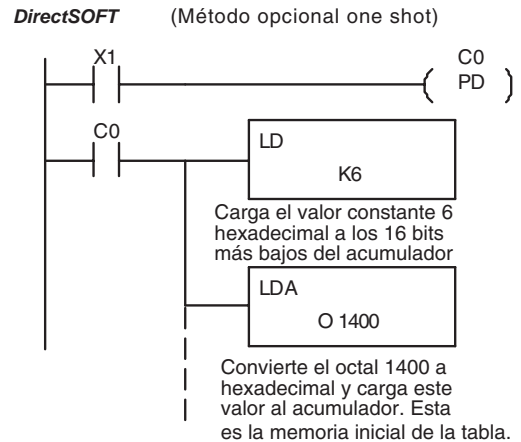


Es importante entender cómo se numeran las direcciones de tabla. Si usted examina la tabla del ejemplo, usted notará que la primera dirección de almacenamiento de datos, V1401, se usará cuando el puntero es igual a 0 y otra vez cuando el puntero es igual a seis. ¿Por qué? Porque el puntero es sólo igual a 0 antes de la primera ejecución. De allí en adelante incrementa de 1 a 6 y entonces vuelve a 1.

También, nuestro ejemplo usa un contacto normal de entrada (X1) para controlar la ejecución. Ya que el barrido de la CPU es extremadamente rápido y el puntero se incrementa automáticamente, los datos de la fuente se cambiarían a todas las direcciones de tabla muy rápidamente. Si esto es un problema para su aplicación, usted tiene la opción de usar una instrucción one shot (PD) para mover un valor cada vez que el contacto de entrada hace la transición de OFF para ON.

	Tabla	
V1401	X X X X	0 6
V1402	X X X X	1
V1403	X X X X	2
V1404	X X X X	3
V1405	X X X X	4
V1406	X X X X	5
V1407	X X X X	

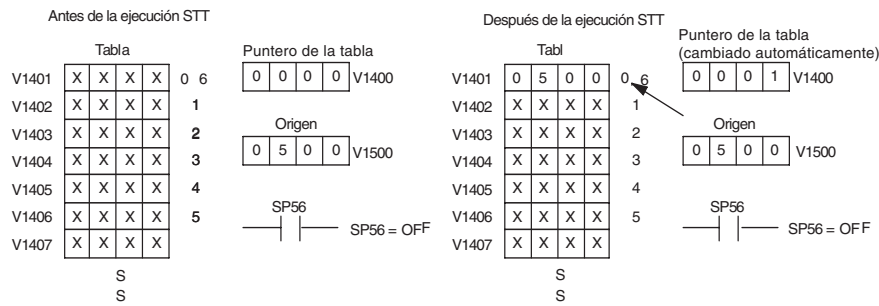
Puntero de la tabla	0 0 0 0	V1400
Origen de los datos	0 5 0 0	V1500



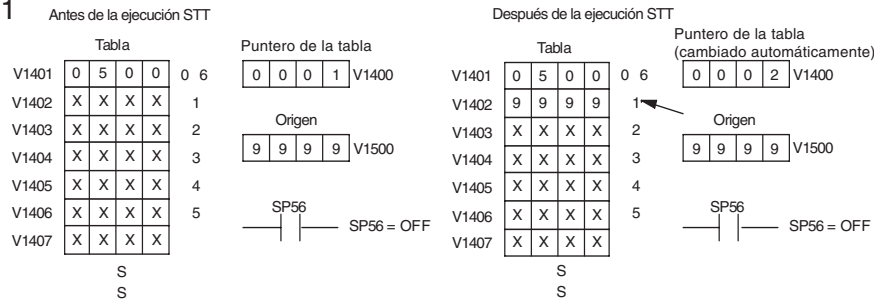
Capítulo 5: Instrucciones normales RLL - Instrucciones de tablas

El esquema siguiente muestra barrido por barrido los resultados de la ejecución para el programa ejemplo. Note cómo el puntero automáticamente hace un ciclo de 0 - 6, y luego comienza de nuevo en 1 en vez de 0. También, note cómo es afectado SP56 por la ejecución. Aunque el ejemplo no lo muestre, asumimos que hay otra parte del programa que cambia el valor en V1500 (la fuente de datos) antes de la ejecución de la instrucción STT. Esto no es necesario pero hace más fácil de ver cómo la fuente de datos se copia en la tabla.

Barrido N

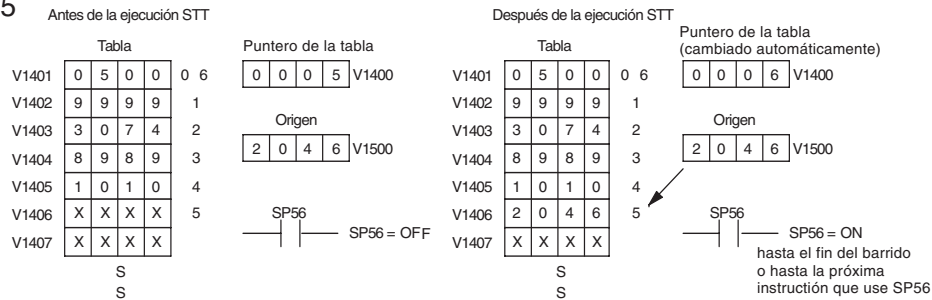


Barrido N+1

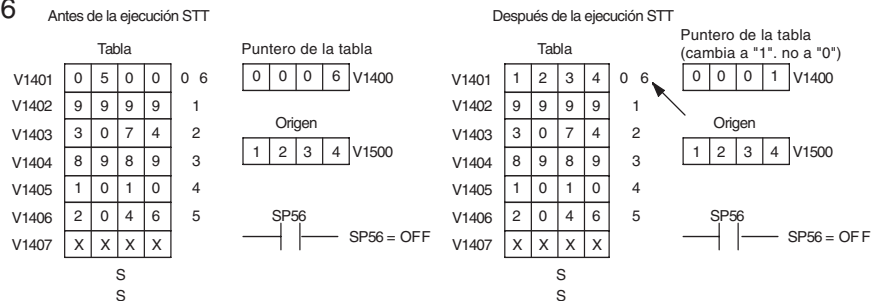


S
S
S

Barrido N+5



Barrido N+6



La instrucción Remove from Table (RFT)

DS5	Usado
HPP	Usado

La instrucción RFT remueve un valor de una tabla y lo almacena en una dirección de memoria V. Cuando un valor se remueve de la tabla todos los otros valores se mueven para arriba 1 nivel. La primera dirección de memoria V en la tabla contiene el contador de la longitud de la tabla. El valor corriente del contador de la tabla disminuye en 1 cada vez que se ejecuta la instrucción. Si el contador de la longitud es cero o mayor que la longitud máxima de la tabla (especificada en el primer nivel del Stack del acumulador) la instrucción no se ejecutará y SP56 se hará ON.



La instrucción se ejecutará una vez por barrido si que el renglón fuera verdadero. Los parámetros de la instrucción son cargados al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios de programar la instrucción RFT.

5

- Paso 1: Cargue la longitud de la tabla (el número de direcciones de memoria V) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 a FF.
- Paso 2: Cargue la localización de la memoria V de inicio a la tabla en el acumulador. (Recuerde, la localización de inicio de la tabla se usa como el contador de longitud de tabla). Este parámetro debe ser un valor hexadecimal.
- Paso 3: Coloque la instrucción RFT que especifica la dirección de la memoria de destino (Vaaa). Esto es, donde el valor en la tabla será movido.

Sugerencia: — Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

Sugerencia:— La instrucción se ejecutará cada barrido si el renglón está ON. Si usted no quiere que la instrucción ejecute más que un barrido, se debe usar la instrucción PD (one shot) en la lógica del renglón.

Sugerencia: — El valor del contador de la tabla se debe definir para indicar el punto de partida de la operación. También, debe ser colocado a un valor que esté dentro de la longitud de la tabla. Por ejemplo, si la tabla es de 6 palabras de longitud, entonces el rango admisible de los valores que podrían estar en el contador de la tabla debe estar entre 1 y 6. Si el valor está fuera de este rango o es 0, los datos no se moverán de la tabla. También, se debe usar una instrucción one shot (PD) de modo que el valor sólo sea colocado en un barrido y no afecte la operación de la instrucción.

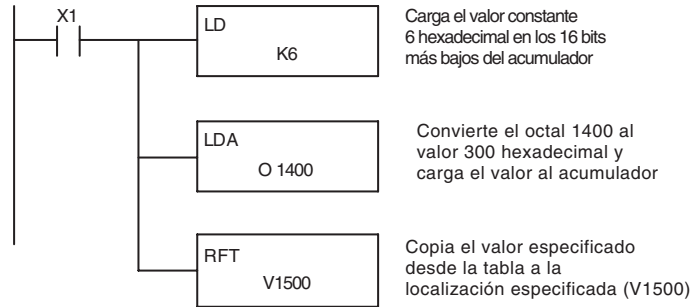
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

Indicadores	Descripción
SP56	ON cuando el valor corriente del contador de la tabla es igual a 0



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP o en el fin del barrido. El puntero para esta instrucción puede comenzar en cualquier lugar en la tabla. No es colocado automáticamente. Usted tiene que cargar un valor en el puntero en algún lugar en su programa.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constante (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en la primera localización del Stack después que se ejecuta la instrucción LDA. La dirección octal 1400 (V1400) es la dirección inicial de la tabla fuente y se carga en el acumulador. La localización (V1500 del destino) es especificado en la instrucción RFT. El contador de la tabla será disminuido en "1" después que se ejecuta la instrucción.



Carga el valor constante 6 hexadecimal en los 16 bits más bajos del acumulador

Convierte el octal 1400 al valor 300 hexadecimal y carga el valor al acumulador

Copia el valor especificado desde la tabla a la localización especificada (V1500)

Programador D2-HPP

\$ STR	→	B 1	ENT							
SHFT	L ANDST	D 3	→	PREV	G 6	ENT				
SHFT	L ANDST	D 3	A 0	→	B 1	E 4	A 0	A 0	ENT	
SHFT	R ORN	F 5	T MLR	→	B 1	F 5	A 0	A 0	ENT	

Ya que el contador de la tabla especifica el rango de los datos que se sacarán de la tabla, es importante entender cómo se numeran las direcciones de la tabla. Si usted examina la tabla del ejemplo, usted notará que las direcciones de datos se numeran desde encima de la tabla. Por ejemplo, si el contador de la tabla comenzó en 6, entonces todas las seis direcciones se afectarían durante la ejecución de la instrucción.

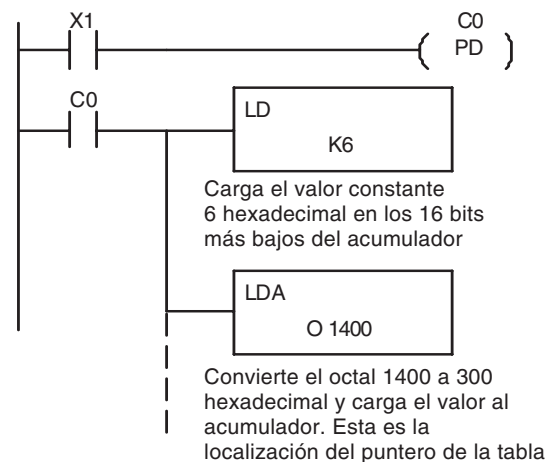
	Tabla				
V1401	0	5	0	0	1
V1402	9	9	9	9	2
V1403	3	0	7	4	3
V1404	8	9	8	9	4
V1405	1	0	1	0	5
V1406	2	0	4	6	6
V1407	X	X	X	X	

Contador de tabla				
0	0	0	6	V1400

Destino				
X	X	X	X	V1500

También, nuestro ejemplo usa un contacto normal de entrada (X1) para controlar la ejecución. Ya que el barrido es extremadamente rápido, y el puntero se decrementa automáticamente, los datos se sacarían de la tabla muy rápidamente. Si esto es un problema para su aplicación, usted tiene una opción de usar una instrucción one shot (PD) para sacar un valor cada vez en la transición del contacto de entrada de OFF para ON.

DirectSOFT (Método opcional "one-shot")



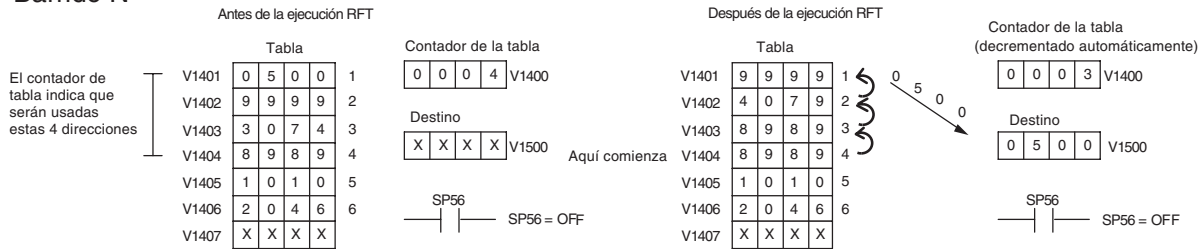
Carga el valor constante 6 hexadecimal en los 16 bits más bajos del acumulador

Convierte el octal 1400 a 300 hexadecimal y carga el valor al acumulador. Esta es la localización del puntero de la tabla

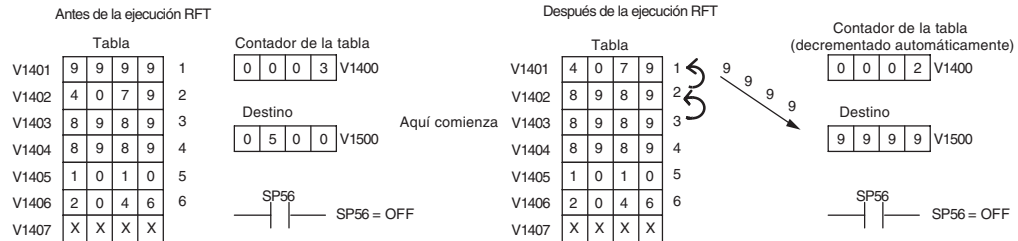
Capítulo 5: Instrucciones normales RLL - Instrucciones de tablas

El esquema siguiente muestra barrido por barrido los resultados de la ejecución para el programa del ejemplo. En el ejemplo mostramos el valor corriente del contador de la tabla en 4 inicialmente. (Recuerde, usted puede poner el valor corriente del contador de la tabla a cualquier valor que este dentro del rango de la tabla). El contador de la tabla automáticamente decrece de 4 hasta 0 cuando se ejecuta la instrucción. Note cómo las últimas dos posiciones de la tabla, 5 y 6, no son movidas para arriba por la tabla. También, note cómo SP56, que se hace ON cuando el contador de la tabla es cero, es ON sólo hasta que el fin del barrido.

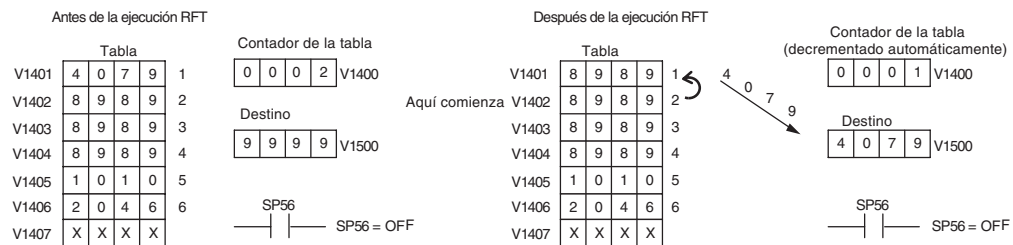
Barrido N



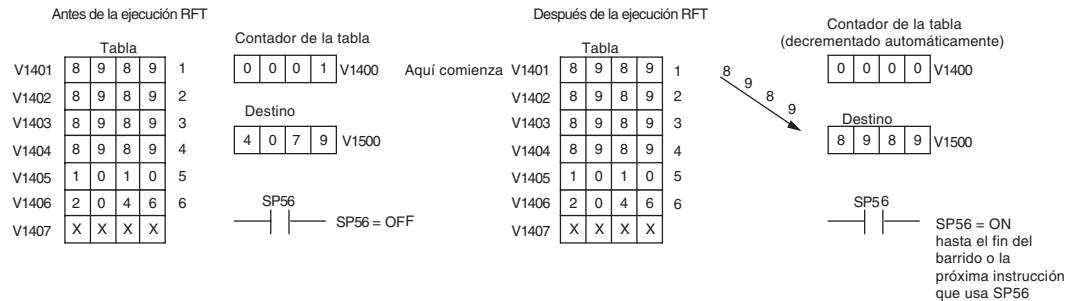
Barrido N+1



Barrido N+2



Barrido N+3



La instrucción Add a Top (ATT)

DS5	Usado
HPP	Usado

La instrucción ATT lleva un valor a la memoria inicial de una tabla de memoria desde una dirección de memoria V. Cuando el valor es agregado a la tabla todos los otros valores se corren hacia abajo 1 localización.



La instrucción se ejecutará una vez por barrido si el renglón fuera ON o verdadero. Los parámetros de la función son cargados al primer nivel del Stack del acumulador y al acumulador con dos instrucciones adicionales. Abajo están listados los pasos necesarios para programar la instrucción ATT.

- Paso 1: Cargue la longitud de la tabla (cantidad de direcciones de memoria V) al primer nivel del Stack. Este parámetro debe ser un valor hexadecimal, 0 a FF.
- Paso 2: Cargue la dirección de memoria V inicial de la tabla al acumulador. (Recuerde, para esta instrucción, la dirección inicial de la tabla se usa como el contador de longitud de la tabla). Este parámetro debe ser un valor hexadecimal.
- Paso 3: Coloque la instrucción ATT que especifica la dirección de la memoria origen (Vaaa). Esto es, desde donde se moverá el valor.

Sugerencia:- La instrucción se ejecutará en cada barrido si el renglón fuera verdadero. Si usted no quiere que la instrucción se ejecute en más que un barrido, se debe usar una instrucción one shot (PD) en la lógica de activación.

Sugerencia: - Para parámetros que requieran valores en hexadecimal cuando se refieran a direcciones de memoria, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

Sugerencia: - Se debe definir el valor del contador de la tabla para indicar el punto de partida de la operación. También, debe ser puesto a un valor que esté dentro de la longitud de la tabla. Por ejemplo, si la tabla es de longitud de 6 palabras, entonces el rango admisible de los valores que podrían estar en el contador de la tabla debe estar entre 1 y 6. Si el valor está fuera de este rango o es 0, los datos no se irán a la tabla. También, se debe usar una instrucción one shot (PD) de modo que el valor sólo sea colocado en un barrido y no afecte la operación de la instrucción.

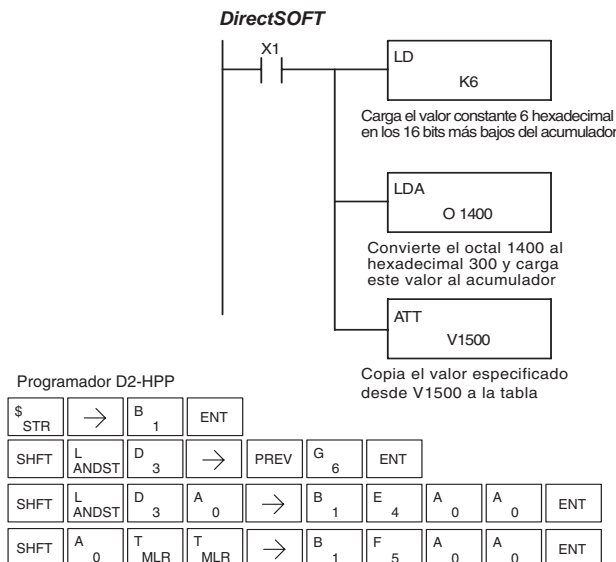
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

Indicadores	Descripción
SP56	ON cuando el valor corriente del contador de la tabla es igual a 0



NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP o en el fin del barrido. El puntero para esta instrucción puede comenzar en cualquier lugar en la tabla. No es colocado automáticamente. Usted tiene que cargar un valor en el puntero en algún lugar en su programa.

En el ejemplo siguiente, cuándo X1 está ON, se carga el valor constante (K6) al acumulador usando la instrucción LD. Este valor especifica la longitud de la tabla y se coloca en la primera localización del Stack después que se ejecuta la instrucción LDA. La dirección octal 1400 (V1400), que es la dirección inicial de la tabla de destino y contador de tabla, se carga en el acumulador. La dirección de la tabla origen (V1500) es especificada en la instrucción ATT. El contador de la tabla será aumentado en "1" después que se ejecuta la instrucción.



En la instrucción ATT, el contador de la tabla determina el número de las adiciones de valores que se pueden hacer antes la instrucción pare de ejecutarse. Es útil entender cómo el sistema usa este contador para controlar la ejecución. Por ejemplo, si el contador de la tabla está colocado en 2 y la longitud de la tabla es 6 palabras, entonces allí pueden haber solamente 4 adiciones de datos antes de que la ejecución se pare. Esto puede ser calculado fácilmente por:

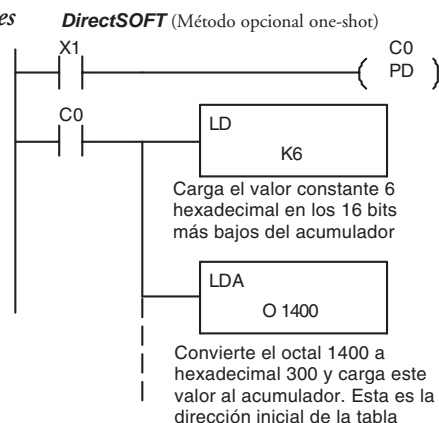
Tabla		Contador de tabla
V1401	0 5 0 0	0 0 0 2
V1402	9 9 9 9	V1400
V1403	3 0 7 4	Origen del dato
V1404	8 9 8 9	X X X X
V1405	1 0 1 0	V1500
V1406	2 0 4 6	
V1407	X X X X	

(Ejemplo: 6 - 2 = 4)

Longitud de tabla - contador de tabla = número de ejecuciones

También, el ejemplo usa un contacto normal de entrada (X1) para controlar la ejecución. Ya que el barrido es extremadamente rápido y el contador de la tabla se incrementa automáticamente, los datos pasan a la tabla muy rápidamente.

Si esto es un problema para su aplicación, tiene la opción de usar una instrucción one shot (PD) para agregar solamente un valor cada vez que el contacto hace la transición de OFF para ON.

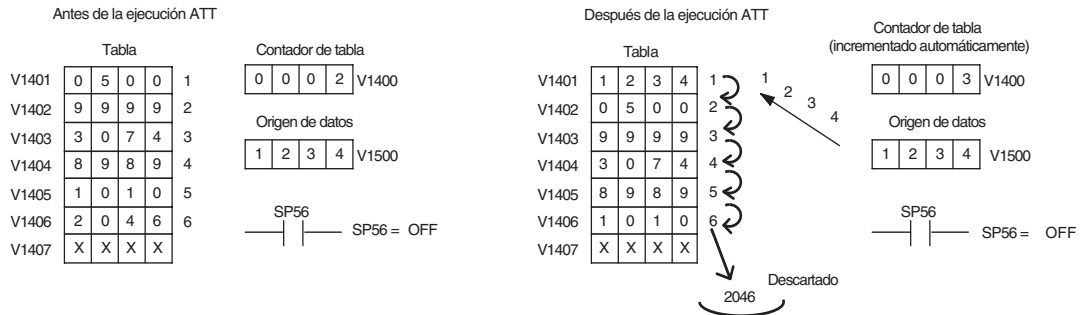


Capítulo 5: Instrucciones normales RLL - Instrucciones de tablas

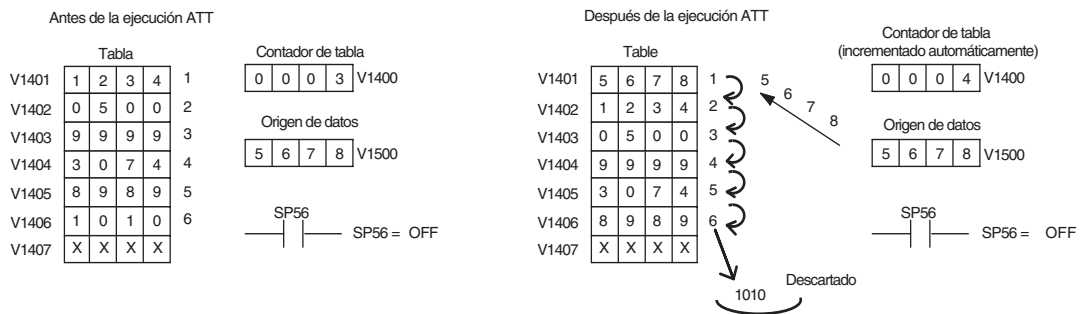
El esquema siguiente muestra barrido por barrido los resultados de la ejecución para el programa del ejemplo. El contador de la tabla es configurado como 2 inicialmente, e incrementará automáticamente de 2 hasta 6 cuando se ejecuta la instrucción. Note cómo SP56 se hace ON cuando el contador de la tabla es 6, que es igual a la longitud de la tabla. Además, aunque el ejemplo no lo muestre, asumimos que hay otra parte del programa que cambia el valor en V1500 (el origen de datos) antes de la ejecución de la instrucción de ATT.

Ejemplo de ejecución

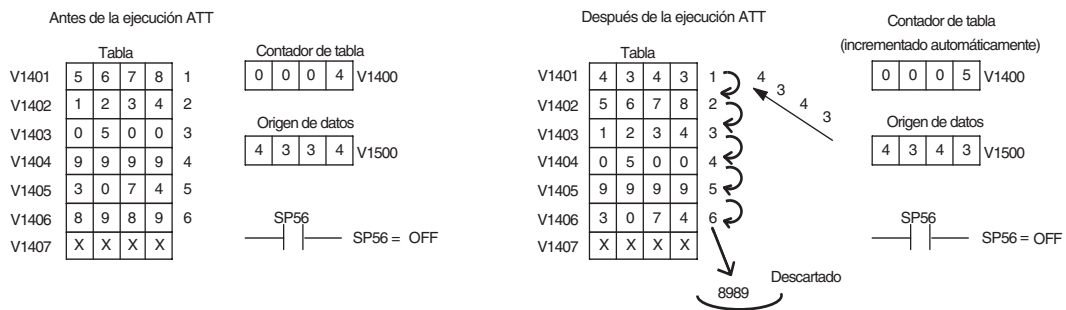
Barrido N



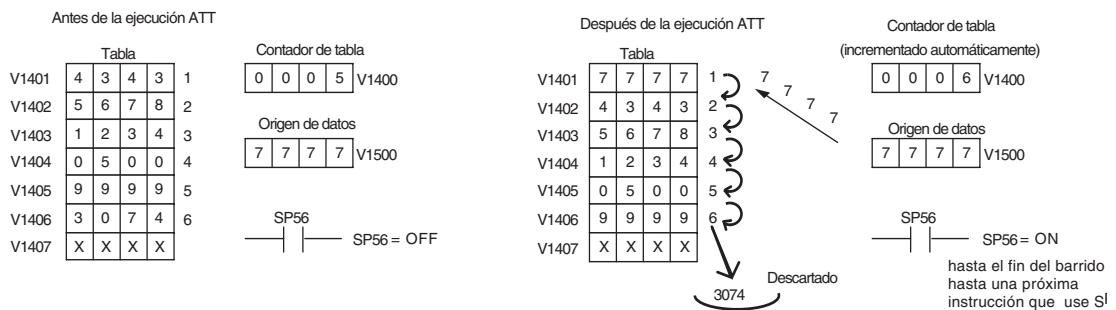
Barrido N+1



Barrido N+2



Barrido N+3



La instrucción Table Shift Left (TSHFL)

DS5	Usado
HPP	Usado

La instrucción TSHFL mueve todos los bits en una tabla de memoria a la izquierda el número especificado de posiciones de bit, esto es, desde el bit menos al más significativo.



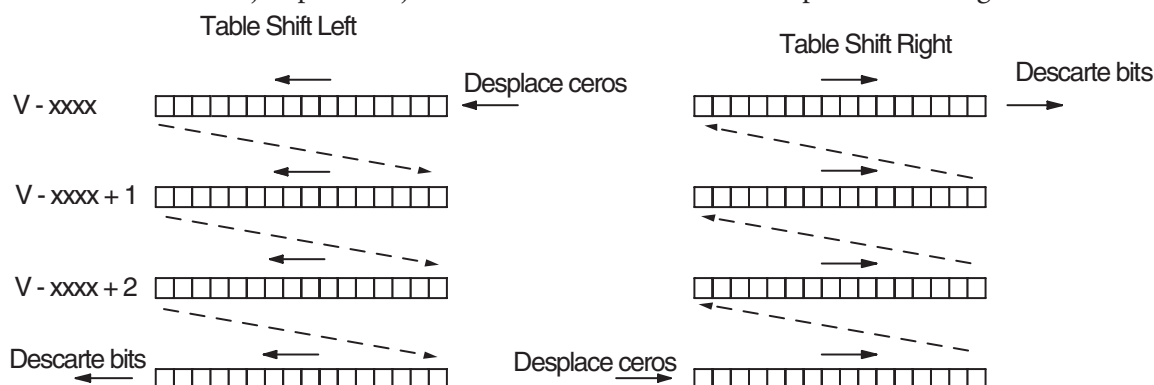
La instrucción Table Shift Right (TSHFR)

DS5	Usado
HPP	Usado

La instrucción TSHFR mueve todos los bits en una tabla de memoria V a la derecha, un número especificado de posiciones de bit, esto es, desde el bit más al menos significativo.



La descripción siguiente se aplica a ambas instrucciones. Una tabla es solamente un rango de direcciones de memoria V. Las instrucciones TSHFL y TSHFR mueven los bits serialmente a lo largo de la tabla entera. Los bits se mueven saliendo del fin de una palabra y hasta el fin opuesto de una palabra adyacente. Al final de la tabla los bits son sacados o se desplazan ceros a la tabla. Las tablas del ejemplo debajo son arbitrariamente de cuatro palabras de largo.



- Paso 1: Cargue la longitud de la tabla (cantidad de direcciones de memoria V) en el primer nivel del stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 hasta FF
- Paso 2: Cargue la localización de la memoria V de inicio de la tabla al acumulador. Este parámetro debe ser un valor hexadecimal. Usted puede usar la instrucción LDA para convertir una dirección de octal a hexadecimal.
- Paso 3: Coloque la instrucción que desee. Esta especifica el número de posiciones de bits que desea desplazar la tabla entera. El número de posiciones de bits debe estar en octal.

Sugerencia: — Recuerde que cada dirección de memoria V contiene 16 bits, de modo que los bits de la primera palabra de la tabla se numeran de 0 a 17 octal. Si usted quiere desplazar la tabla entera 20 bits, eso es 24 octal. SP 53 será ON si el número de bits a ser desplazado es más grande que los bits totales contenidos dentro de la tabla.

El relevador especial SP67 será colocado ON si el último bit desplazado (justamente antes de eliminarlo) es un "1".

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

Indicadores	Descripción
SP53	ON cuando el número de bits a ser desplazados es más grande que el número de bits en la tabla
SP67	ON cuando el último bit que se desplazó es un "1" (antes de que sea eliminado)



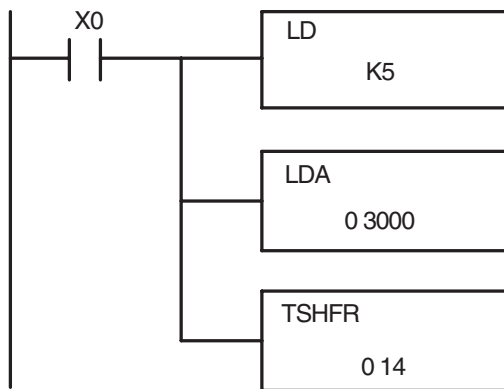
NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP. o hasta el fin del barrido.

La tabla del ejemplo contiene a la derecha los datos BCD como mostrado (para propósitos de demostración). Suponga que queremos hacer una movida de bits a la derecha de 3 dígitos BCD (12 bits). La conversión al octal de 12 bits es 14 octal. Usando instrucción TSHFR y especificando un desplazamiento a la derecha con el octal 14, tenemos la tabla resultante mostrada a la derecha. Note que se ha sacado la sucesión 2-3-4 de la secuencia y se ha desplazado la sucesión 0-0-0 en la parte inferior.

V 3000	V 3000
1 2 3 4	6 7 8 1
5 6 7 8	1 2 2 5
1 1 2 2	3 4 4 1
3 3 4 4	5 6 6 3
5 5 6 6	0 0 0 5

El ejemplo siguiente ladder asume que los datos en V3000 a V3004 ya existen, como mostrado arriba. Usaremos la entrada X0 para provocar la operación. Primero, cargaremos la longitud de tabla (5 palabras) al Stack del acumulador. Luego cargamos la dirección de inicio al acumulador. Ya que V3000 es un número octal lo tenemos que convertir a hexadecimal usando la instrucción LDA. Finalmente, usamos la instrucción TSHFR y especificamos el número de bits para ser desplazados (12 decimal), que es 14 octal.

DirectSOFT



Carga el valor constante 5 hexadecimal a los 16 bits más bajos del acumulador

Convierte el octal 3000 to hexadecimal y carga el valor al acumulador. Este es el inicio de la tabla

Hace una operación SHIFT RIGHT con 12 bits, el cual es 14 octal.

Programador D2-HPP

\$ STR	→	A 0	ENT										
SHFT	L ANDST	D 3	→	PREV	F 5	ENT							
SHFT	L ANDST	D 3	A 0	→	D 3	A 0	A 0	A 0	ENT				
SHFT	T MLR	SHFT	S RST	H 7	F 5	R ORN	→	NEXT	B 1	E 4	ENT		

La instrucción AND Move (ANDMOV)

DS5	Usado
HPP	Usado

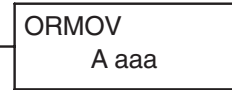
La instrucción ANDMOV copia los datos de una tabla a la dirección especificada de memoria, haciendo la operación AND de cada palabra con los datos de acumulador cuando se procesa.



La instrucción OR Move (ORMOV)

DS5	Usado
HPP	Usado

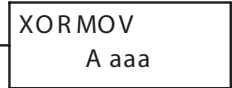
La instrucción OR MOVE copia los datos de una tabla a la dirección especificada de memoria, haciendo la operación OR de cada palabra con el valor contenido en el acumulador cuando se procesa.



La instrucción Exclusive OR Move (XORMOV)

DS5	Usado
HPP	Usado

La instrucción Exclusive OR Move copia los datos de una tabla a la dirección especificada de memoria, haciendo una operación OR exclusiva de cada palabra con el valor del acumulador cuando se procesa.



La descripción siguiente se aplica a las instrucciones ANDMOV, ORMOV y XORMOV.

Estas instrucciones copian los datos de una tabla a otra tabla en la localización especificada, ejecutan una operación lógica en cada palabra con el contenido del acumulador y crean entonces la otra tabla.

Paso 1: Cargue la longitud de la tabla (el número de direcciones de memoria V) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 hasta FF.

Paso 2: Cargue la dirección inicial de la memoria V de la tabla en el acumulador. Este parámetro debe ser un valor hexadecimal. Usted puede usar la instrucción LDA para convertir una dirección octal a uno hexadecimal.

Paso 3: Cargue el valor BCD/hexadecimal que expresa el conjunto de bits a ser el operando AND en el acumulador que será combinado lógicamente con el contenido de la tabla durante la operación.

Paso 4: Coloque una de las instrucciones ANDMOV, ORMOV o XORMOV. La que sea escogida especifica la dirección inicial de la copia de la tabla original. Esta tabla nueva será automáticamente de la misma longitud que la tabla original.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria

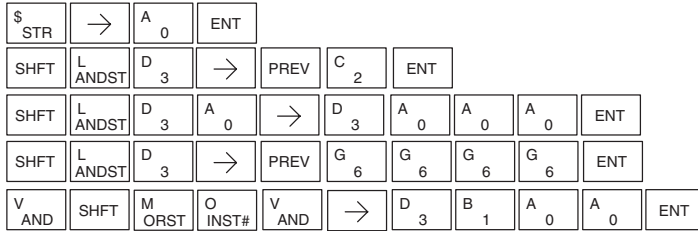
La tabla del ejemplo contiene a la derecha los datos BCD como mostrado (para propósitos de demostración). Suponga que queremos mover una tabla de dos palabras localizada en V3000 y hacer la operación AND con K6666. La copia de la tabla en V3100 muestra el resultado de la operación AND con cada palabra.



El programa en esta página realiza el ejemplo de la operación de ANDMOV anterior. Asume que los datos en la tabla en V3000 - V3001 ya existen. Primero cargamos la longitud de la tabla (dos palabras) en el acumulador. Luego cargamos la dirección de inicio de la tabla origen, usando la instrucción LDA. Luego cargamos los datos en el acumulador para ser operados AND con la tabla. En la instrucción ANDMOV se especifica el destino de la tabla, V3100.

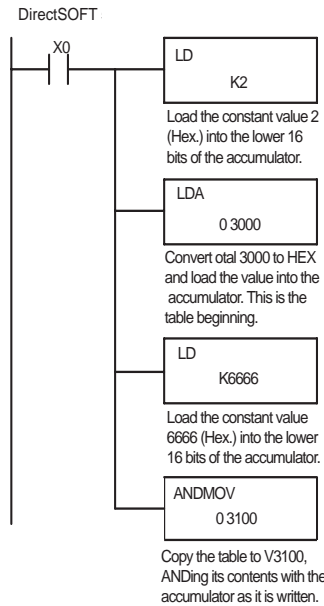
5

Programador D2-HPP

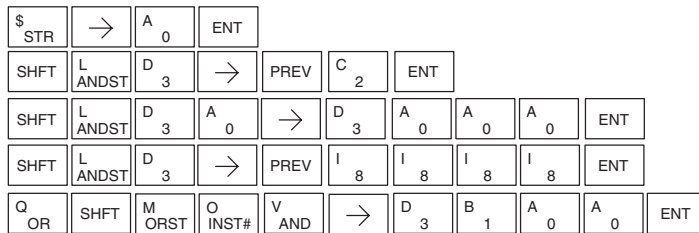


El ejemplo de la derecha muestra una tabla de dos palabras en V3000 y lógica OR con K8888. La copia de la tabla en V3100 muestra el resultado de la operación OR con cada palabra.

El programa de abajo realiza el ejemplo de ORMOV arriba. Asume que los datos en la tabla en V3000 - V3001 ya existen. Primero cargamos la longitud de la tabla (dos palabras) en el acumulador. Luego cargamos la dirección de inicio de la tabla fuente, usando la instrucción LDA. Luego cargamos los datos en el acumulador para ser operados OR con la tabla. En la instrucción ORMOV se especifica el destino de la tabla, V3100.



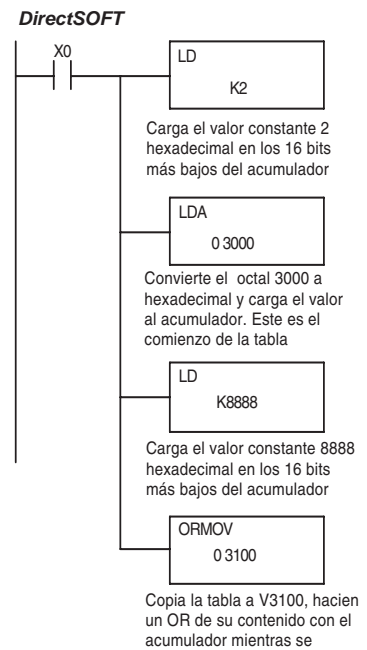
Programador D2-HPP



El ejemplo a la derecha muestra una tabla de dos palabras en V3000 y hace una operación XOR lógico con K3333. La copia de la tabla en V3100 muestra el resultado de la operación XOR para cada palabra.

El ejemplo del programa ladder para el XORMOV es similar al de arriba para el ORMOV. Use sin embargo la instrucción XORMOV.

En el programador usted debe usar la tecla SHFT y deletrear "XORMOV" explícitamente..



La instrucción Find Block (FINDB)

DS5	Usado
HPP	N/A

La instrucción FINDB busca una ocurrencia de un bloque especificado de valores en una tabla de memoria V. Los parámetros de la instrucción son cargados al primer y segundo nivel del Stack del acumulador y el acumulador por tres instrucciones adicionales. Si el bloque se encuentra, su dirección inicial se almacenará en el acumulador. Si el bloque no se encuentra, el relevador especial SP53 se hará ON.



Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V..... V	Vea el mapa de memoria
Puntero..... P	Vea el mapa de memoria

Indicadores	Descripción
SP56	ON cuando la instrucción FINDB fue ejecutada pero no encontró el bloque de datos

5

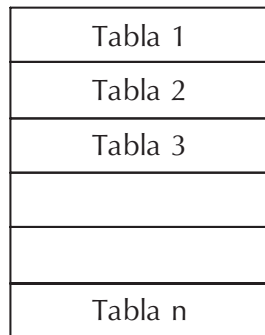


NOTA: Las indicaciones de estado discretas SP son válidas solamente hasta que se ejecute otra instrucción que use los mismos relevadores especiales SP.

Los pasos necesarios para programar la instrucción **FINDB** están listados abajo.

- Paso 1: Cargue el número de byte en el bloque a ser localizado. Este parámetro debe ser un valor hexadecimal, 0 a FF, que es 255 decimal..
- Paso 2: Cargue la longitud de una tabla (el número de palabras) a ser buscada. FINDB buscará múltiples tablas que están adyacentes en la memoria V. Este parámetro debe ser un valor hexadecimal, 0 hasta FF.
- Paso 3: Cargue la localización final para todas las tablas en el acumulador. Este parámetro debe ser un valor hexadecimal. Usted puede usar la instrucción LDA para convertir una dirección octal a hexadecimal.
- Paso 4: Cargue la dirección inicial de la tabla para todas las tablas en el acumulador. Este parámetro debe ser un valor hexadecimal.
- Paso 5: Coloque la instrucción FINDB. Esta especifica la dirección inicial del bloque de los datos que usted trata de localizar.

Dirección inicial



Dirección final

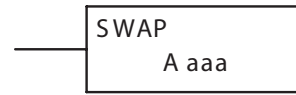
Dirección inicial



La instrucción Swap (SWAP)

DS5	Usado
HPP	Usado

Esta instrucción SWAP intercambia datos en dos tablas de igual longitud.



Paso 1: Cargue la longitud de las tablas (la cantidad de direcciones de memoria V) al primer nivel del Stack del acumulador. Este parámetro debe ser un valor hexadecimal, 0 hasta FF. Recuerde que las tablas deben ser de igual longitud.

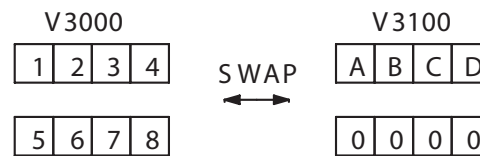
Paso 2: Cargue la dirección de la memoria V de inicio de la primera tabla al acumulador. Este parámetro debe ser un valor hexadecimal. Usted puede usar la instrucción de LDA para convertir una dirección octal a hexadecimal.

Paso 3: Coloque la instrucción SWAP. Esta especifica la dirección inicial de la segunda tabla. Este parámetro debe ser un valor hexadecimal.

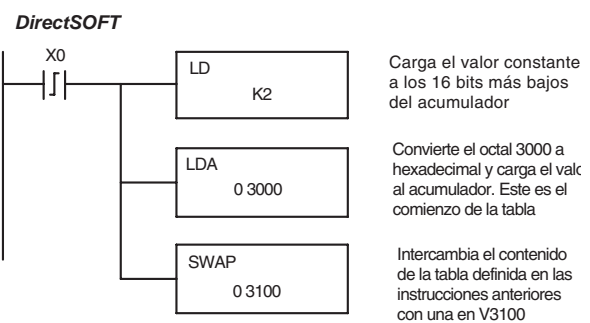
Sugerencia: — El intercambio de datos ocurre dentro de un solo barrido. Si la instrucción ejecuta en múltiples barridos, será difícil de saber el contenido real de cualquier tabla en algún tiempo particular. De modo que solo ejecuta esta instrucción en un solo barrido.

Tipo de operando de datos	Rango del DL06
	aaa
Memoria V V	Vea el mapa de memoria

El ejemplo a la derecha muestra una tabla de 2 palabras comenzando en V3000. Haremos la función SWAP con otra tabla de 2 palabras comenzando en V3100. El programa ladder para esto es mostrado abajo.



El programa del ejemplo adyacente usa un contacto PD (dispara por un barrido en la transición de OFF para ON de X0). Primero, cargamos la longitud de las tablas (dos palabras) al acumulador. Luego cargamos la dirección de la primera tabla (V3000) en el acumulador usando la instrucción LDA, convirtiendo la dirección de octal a hexadecimal. **Note que no importa cuál tabla declaramos "primero", porque los resultados de intercambio serán los mismos.**



Programador D2-HPP

\$	STR	SHFT	P	CV	D	3	→	A	0	ENT									
SHFT	L	ANDST	D	3	→	PREV	C	2	ENT										
SHFT	L	ANDST	D	3	A	0	→	D	3	A	0	A	0	A	0	ENT			
SHFT	S	RST	SHFT	W	ANDN	A	0	P	CV	→	D	3	B	1	A	0	A	0	ENT

Instrucciones de fecha y hora

La instrucción Date (DATE)

DS5	Usado
HPP	Usado

La instrucción DATE puede ser usada para poner la fecha en la CPU. La instrucción requiere dos direcciones consecutivas de memoria V (Vaaa) para ajustar la fecha. Si los valores especificados en las direcciones no son válidos, la fecha no se ajustará en la CPU. La fecha actual se puede leer desde 4 memorias V consecutivas V (V7771 hasta V7774).



En el ejemplo siguiente, cuándo C0 está ON, el valor constantee (K94010301) es cargado en el acumulador usando la instrucción LDD (C0 debe ser un contacto de una instrucción One Shot (PD)). El valor en el acumulador es copiado a V2000 usando la instrucción OUTD. La instrucción DATE usa el valor en V2000 para ajustar la fecha en la CPU.

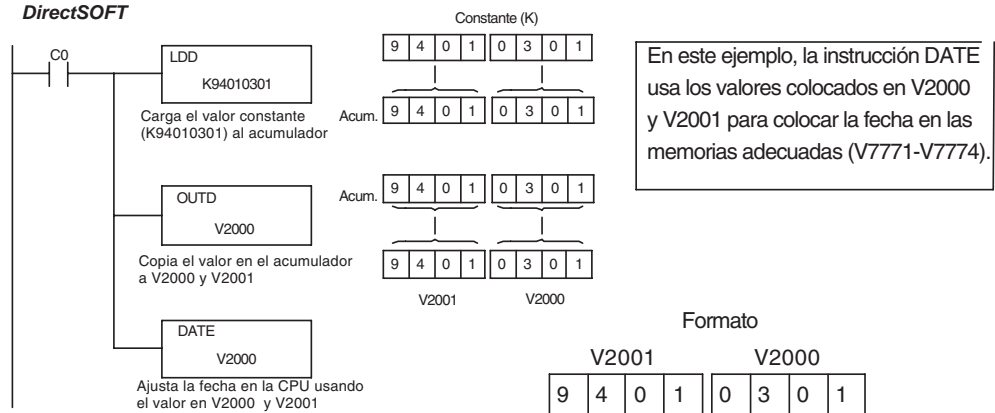
5

Fecha	Rango	Memoria V(BCD) (Sólo para lectura)
Añ	0-99	V7774
Mes	1-12	V7773
Día	1-31	V7772
Día de la semana	0-06	V7771

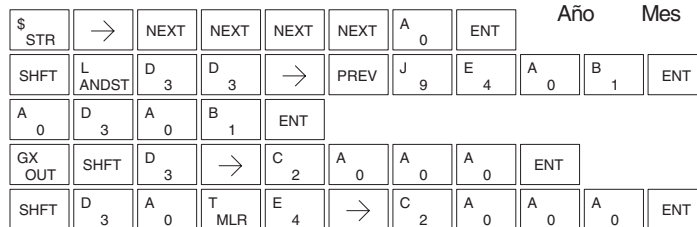
Los valores a entrar como día de la semana son:
0=Domingo, 1=Lunes, 2=Martes, 3=Miércoles, 4=Jueves, 5=Viernes, 6=Sábado

Tipo de operando de datos	Rango del DL06
	aaa
Memoria V V	Vea el mapa de memoria

DirectSOFT



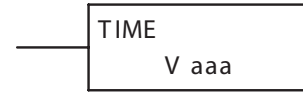
Programador d2-HPP



La instrucción Time (TIME)

DS5	Usado
HPP	Usado

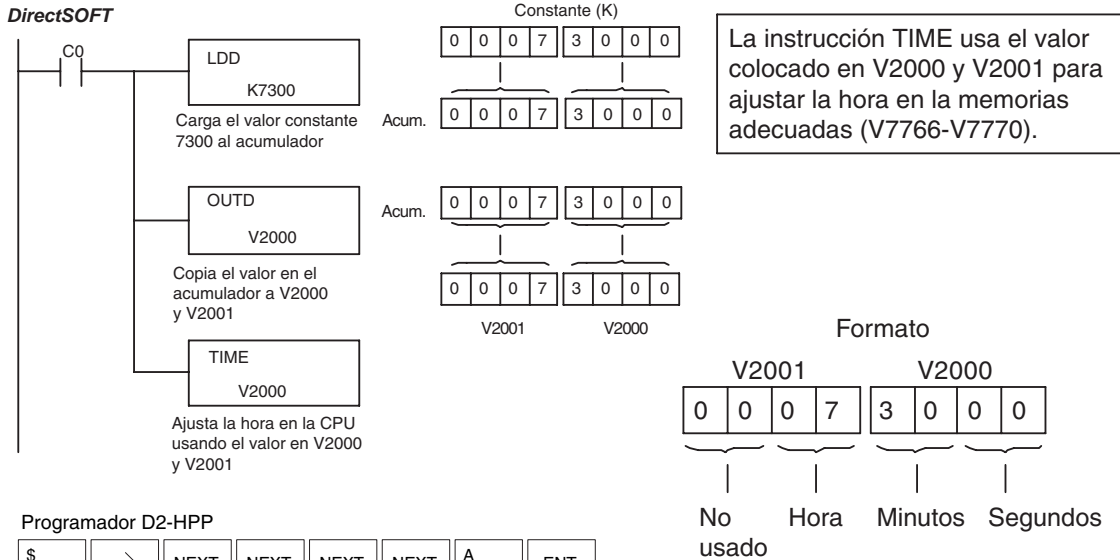
La instrucción TIME se puede usar para ajustar la hora (24 horas) en la CPU. La instrucción requiere dos direcciones consecutivas de memoria V (Vaaa) que se usan para ajustar la hora, minutos y segundos. Si los valores en las direcciones especificadas no son válidos, el tiempo no se ajustará. El tiempo actual se puede leer en las direcciones de memoria V7747 y V7766-V7770.



Hora	Rango	Dirección de Memoria V (BCD) (Sólo para lectura)
1/100 segundos (10ms)	0-99	V7747
Segundos	0-59	V7766
Minutos	0-59	V7767
Hora	0-23	V7770

Tipo de operando de datos	Rango del DL06
Memoria V V	aaa
	Vea el mapa de memoria

En el ejemplo siguiente, cuándo C0 está ON, se carga el valor constantee (K73000) al acumulador usando la instrucción LDD. (C0 debe ser un contacto de una instrucción one shot (PD)). El valor en el acumulador es copiado a V2000 usando la instrucción OUTD. La instrucción TIME usa el valor en V2000 para ajustar la hora en la CPU.



Programador D2-HPP

\$	STR	→	NEXT	NEXT	NEXT	NEXT	A	0	ENT			
SHFT	L	ANDST	D	D	→	PREV	H	D	A	A	A	ENT
A	D	A	B	ENT								
GX	OUT	SHFT	D	→	C	A	A	A	ENT			
SHFT	T	MLR	SHFT	I	M	E	→	C	A	A	A	ENT

Instrucciones de control de la CPU

La instrucción No Operation (NOP)

DS5	Usado
HPP	Usado

La instrucción NOP es una dirección de memoria vacía (no programada) .

—(NOP)

DirectSOFT



Programador D2-HPP



La instrucción End (END)

DS5	Usado
HPP	Usado

La instrucción END marca el punto de terminación del barrido del programa normal. ES NECESARIO COLOCAR una instrucción END al fin del cuerpo principal del programa. Si se omite la instrucción END ocurrirá un error y la CPU no entrará en Modo Run. Las etiquetas de datos, los programas de subrutina s e interrupción se colocan después la instrucción END. La instrucción END no es condicional; por lo tanto, no se coloca ningún contacto de entrada.

—(END)

DirectSOFT



Programador D2-HPP



La instrucción Stop (STOP)

DS5	Usado
HPP	Usado

La instrucción STOP cambia el modo operacional de la CPU, del modo RUN a Program (STOP). Esta instrucción se usa típicamente para parar la operación del PLC en una condición de error.

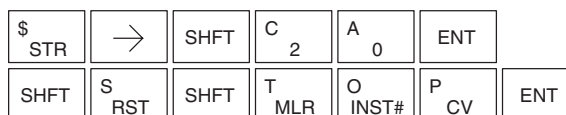
—(STOP)

En el ejemplo siguiente, cuándo C0 prende, la CPU detiene la operación y cambia el modo a Program.

DirectSOFT



Programador D2-HPP



Indicadores	Descripción
SP16	On cuando el PLC DL06 pasa al modo TERM_PRG
SP53	On cuando la instrucción STOP es ejecutada..

La instrucción Reset Watch Dog Timer (RSTWT)

DS5	Usado
HPP	Usado

La instrucción RSTWT coloca el temporizador de barrido de la CPU a 0. El ajuste original del temporizador de watchdog es 200 milisegundos.

—(RSTWT)

Watch dog timer es un temporizador que supervisa que el barrido no exceda el valor prefijado. El barrido de la CPU muy rara vez excede 200 ms, pero es posible que suceda.

Los lazos For/Next, subrutinas, rutinas de interrupción e instrucciones de tablas se pueden programar de tal forma que puede ser que el barrido llegue a ser más largo que 200 ms.

Cuándo se usa una o más instrucciones de una forma que podría exceder el watch dog timer, esta instrucción se puede usar para reponer este temporizador.

Un error (E003 de tiempo muerto de software) ocurrirá y la CPU entrará el modo de programa si el tiempo de barrido excede el valor prefijado en este temporizador . Es muy importante la colocación de la instrucción RSTWT en el programa.

La instrucción se tiene que ejecutar antes que el tiempo de barrido exceda el ajuste del watch dog timer.

Si el tiempo de barrido es continuamente más largo que el temporizador watchdog, el valor de tiempo muerto se puede aumentar permanentemente del valor normal de 200 ms con la función auxiliar apropiada en su paquete de programación. Esto elimina la necesidad de la instrucción RSTWT.

En el ejemplo siguiente el temporizador watchdog de la CPU será repuesto a 0 cuando la instrucción de RSTWT se ejecuta. Vea la instrucción For/Next para un ejemplo detallado.

DirectSOFT



Programador D2-HPP



Instrucciones de control de programa

La instrucción Goto Label (GOTO) (LBL)

DS5	Usado
HPP	Usado

Estas instrucciones se saltan todas instrucciones entre el Goto y la instrucción correspondiente de LBL. El valor del operando para el Goto y la instrucción correspondiente de LBL es el mismo. La lógica entre Goto y la instrucción de LBL no se ejecuta cuando la instrucción de Goto se habilita. Pueden ser usadas hasta 256 instrucciones de Goto y 256 instrucciones de LBL en el programa.

K aaa
—(GOTO)

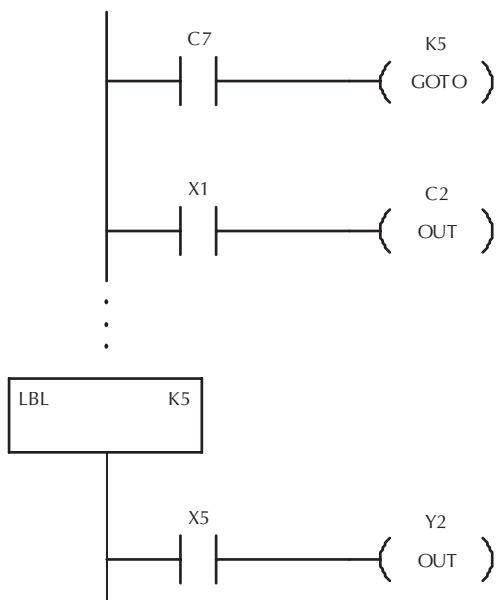
LBL K aaa

Tipo de operando de datos	Rango del DL06
	aaa
Constante K	1-FFFF

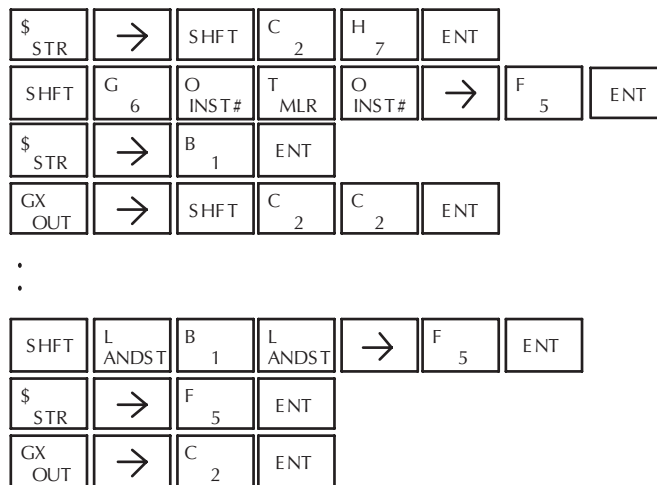
5

En el ejemplo siguiente, cuándo C7 está ON, se saltará toda la lógica del programa entre el GOTO y la instrucción correspondiente de LBL (designado con el mismo valor constante de Kaaa). Las instrucciones a ser saltadas no serán ejecutadas por la CPU.

DirectSOFT



Programador D2-HPP

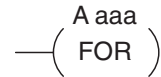


La instrucción For / Next (FOR) (NEXT)

DS5	Usado
HPP	Usado

Las instrucciones FOR y NEXT se usan para ejecutar una sección de la lógica ladder entre la instrucción FOR y NEXT un número de veces especificado.

Cuándo la instrucción FOR es activada, el programa se ejecutará el número de veces especificado en esa sección del programa. Si la instrucción FOR no es energizada no se ejecutan las instrucciones en la sección de la lógica ladder entre el FOR y NEXT.



Las instrucciones FOR/ NEXT no se pueden anidar, es decir, no se puede usar una instrucción dentro de otra. La actualización normal de entradas y salidas y el trabajo de la CPU se suspende al ejecutar el lazo FOR/NEXT.

El barrido del programa puede aumentar significativamente, dependiendo de la cantidad de tiempo que tome para ejecutarse la lógica entre las instrucción FOR/NEXT.



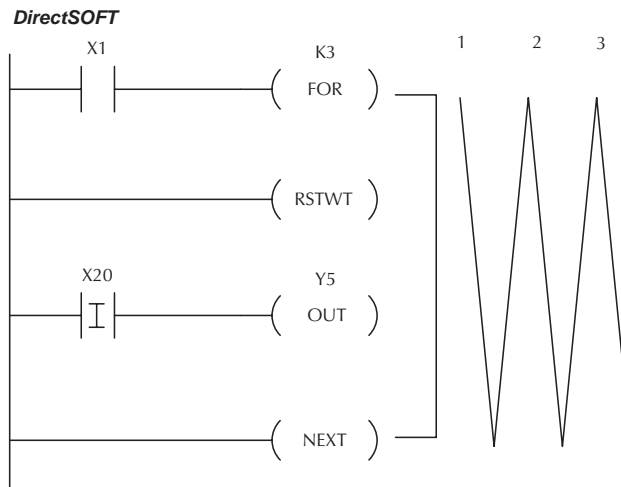
Con la excepción de instrucciones inmediatas de entradas y salidas, las entradas y salidas no se actualizarán hasta que la ejecución del programa se complete para ese barrido.

Dependiendo del plazo de tiempo requerido para completar la ejecución del programa, puede ser necesario usar la instrucción de RSTWT dentro del lazo FOR/NEXT.

Tipo de operando de datos	Rango del DL06
	aaa
Memoria V.....V	Vea el mapa de memoria
ConstanteK	1-9999

En el ejemplo siguiente, cuándo X1 está ON, el programa de aplicación dentro del lazo FOR/NEXT se ejecutará tres veces. Si X1 está apagado el programa dentro del lazo no se ejecutará. Las instrucciones inmediatas pueden o no pueden ser necesarias dependiendo de su aplicación. También, la instrucción RSTWT no es necesaria si el lazo FOR/NEXT no extiende el tiempo de barrido más de lo que esté ajustado el Watch dog timer.

Para más información del Watch dog timer, vea la instrucción RSTWT.



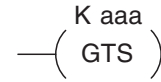
Programador D2-HPP

\$ STR	→	B 1	ENT			
SHFT	F 5	O INST#	R ORN	→	D 3	ENT
SHFT	R ORN	S RST	T MLR	W ANDN	T MLR	ENT
\$ STR	SHFT	I 8	→	C 2	A 0	ENT
GX OUT	→	F 5	ENT			
SHFT	N TMR	E 4	X SET	T MLR	ENT	

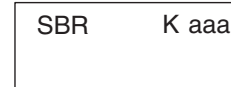
La instrucción Goto Subroutine (GTS) (SBR)

DS5	Usado
HPP	Usado

La instrucción de GOTO Subrutine permite que una sección de la lógica ladder sea colocada fuera del cuerpo principal del programa y ejecutada sólo cuando sea necesario. Puede haber un máximo de 256 instrucciones de GTS y 256 instrucciones de SBR usados en un programa. Las instrucciones de GTS se pueden anidar hasta 8 niveles. Un error E412 ocurrirá si se exceden los límites máximos.



Típicamente esto se usará en una aplicación donde un bloque de lógica del programa puede ser lento de ejecutar y no es necesario ejecutar las instrucciones en cada barrido. El LABEL de la subrutina y toda la lógica asociada se colocan después la declaración END en el programa. Cuando la subrutina es llamada desde el programa principal, la CPU ejecutará la subrutina (SBR) con el mismo número (K) constante que la instrucción de GTS que llamó la subrutina.



El código en una subrutina es ejecutado solamente cuando sea necesario ya que va después de la instrucción END. El código que no es ejecutado no afecta el tiempo de barrido completo del programa.

Tipo de operando de datos	Rango del DL06
	aaa
Constante K	1-FFFF

La instrucción Subroutine Return (RT)

DS5	Usado
HPP	Usado

Cuando se ejecuta una instrucción RT en la subrutina, la CPU volverá al punto en el cuerpo principal del programa de donde se llamó la subrutina. Esta instrucción se usa como terminación de la subrutina, que debe ser la última instrucción en la subrutina y es una instrucción incondicional (no hay ningún contacto de entrada en el renglón).



La instrucción Subroutine Return Conditional (RTC)

DS5	Usado
HPP	Usado

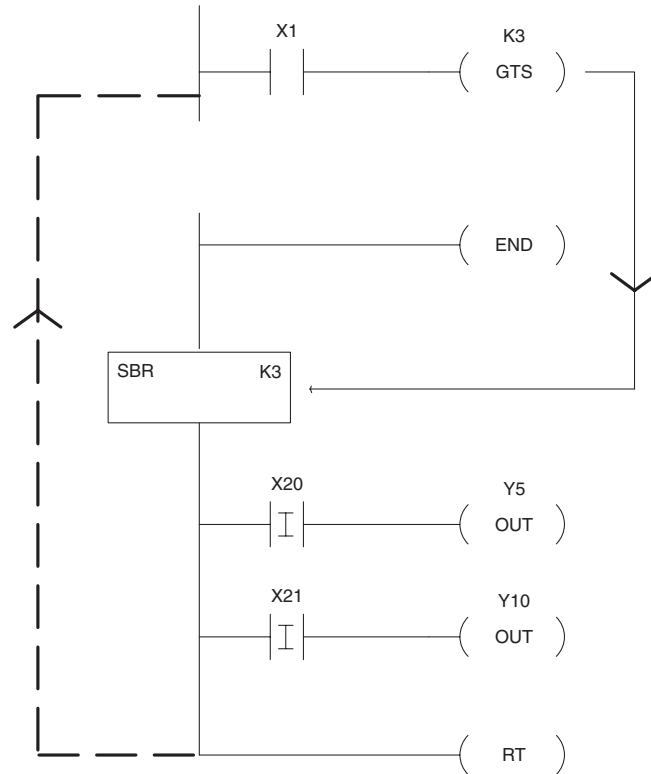
La instrucción RTC es una instrucción opcional usada con un contacto de entrada para implementar un regreso condicional de la subrutina. Se necesita aún la instrucción RT para terminación de la subrutina.



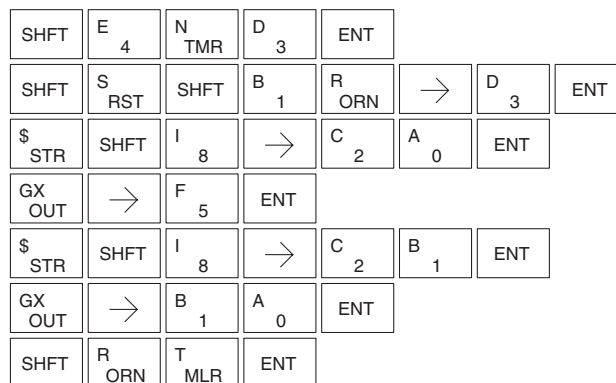
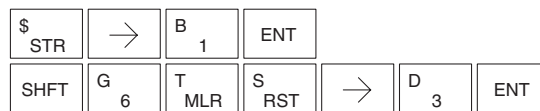
Capítulo 5: Instrucciones de control de programa

En el ejemplo siguiente, cuándo X1 está ON, se llamará la Subrutina K3. La CPU saltará al Label K3 de la subrutina y se ejecutará la lógica ladder en la subrutina. La CPU volverá al cuerpo principal del programa después que se ejecuta la instrucción RT.

DirectSOFT



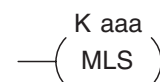
Programador D2-HPP



La instrucción Master Line Set (MLS)

DS5	Usado
HPP	Usado

La instrucción MLS permite que el programa controle las secciones de lógica ladder formando un nuevo riel de energía controlado por el riel principal izquierdo de energía. El riel principal izquierdo es siempre la línea maestra 0. Cuando se usa una instrucción de MLS K1, se crea un riel nuevo de energía en el nivel 1. Las instrucciones MLS y MLR pueden ser usadas para anidar rieles de energía de hasta siete niveles de profundidad.

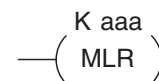


Tipo de operando de datos	Rango del DL06
	aaa
Constante K	1-FFFF

La instrucción Master Line Reset (MLR)

DS5	Usado
HPP	Usado

Las instrucción MLR marca en final del control de la instrucción correspondiente MLS. La referencia MLR es una menos que el de la instrucción correspondiente MLS.



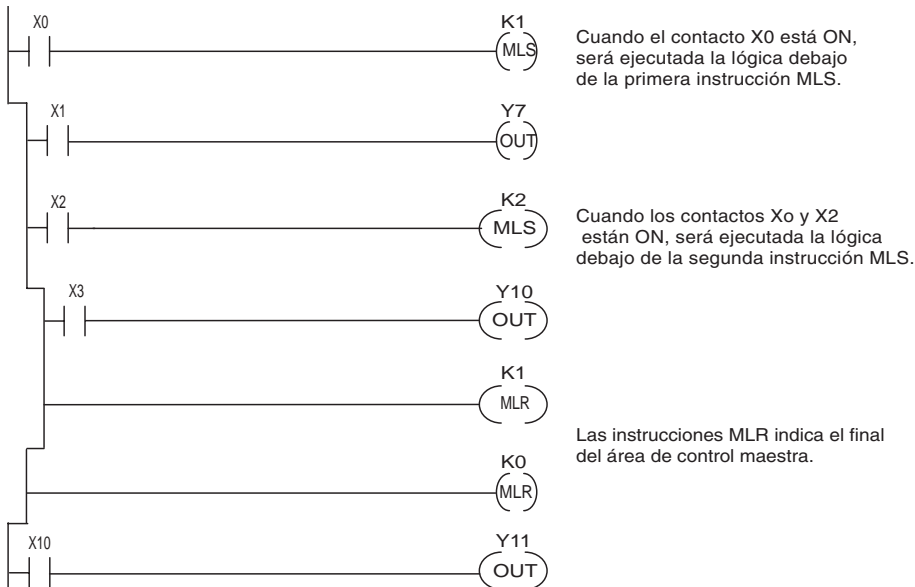
Tipo de operando de datos	Rango del DL06
	aaa
Constante K	1-FFFF

Entendiendo relevadores de control maestros (Master Line)

Las instrucciones MLS y MLR permiten activar o desactivar rápidamente secciones de un programa ladder. Esta característica le proporciona flexibilidad del control al programa.

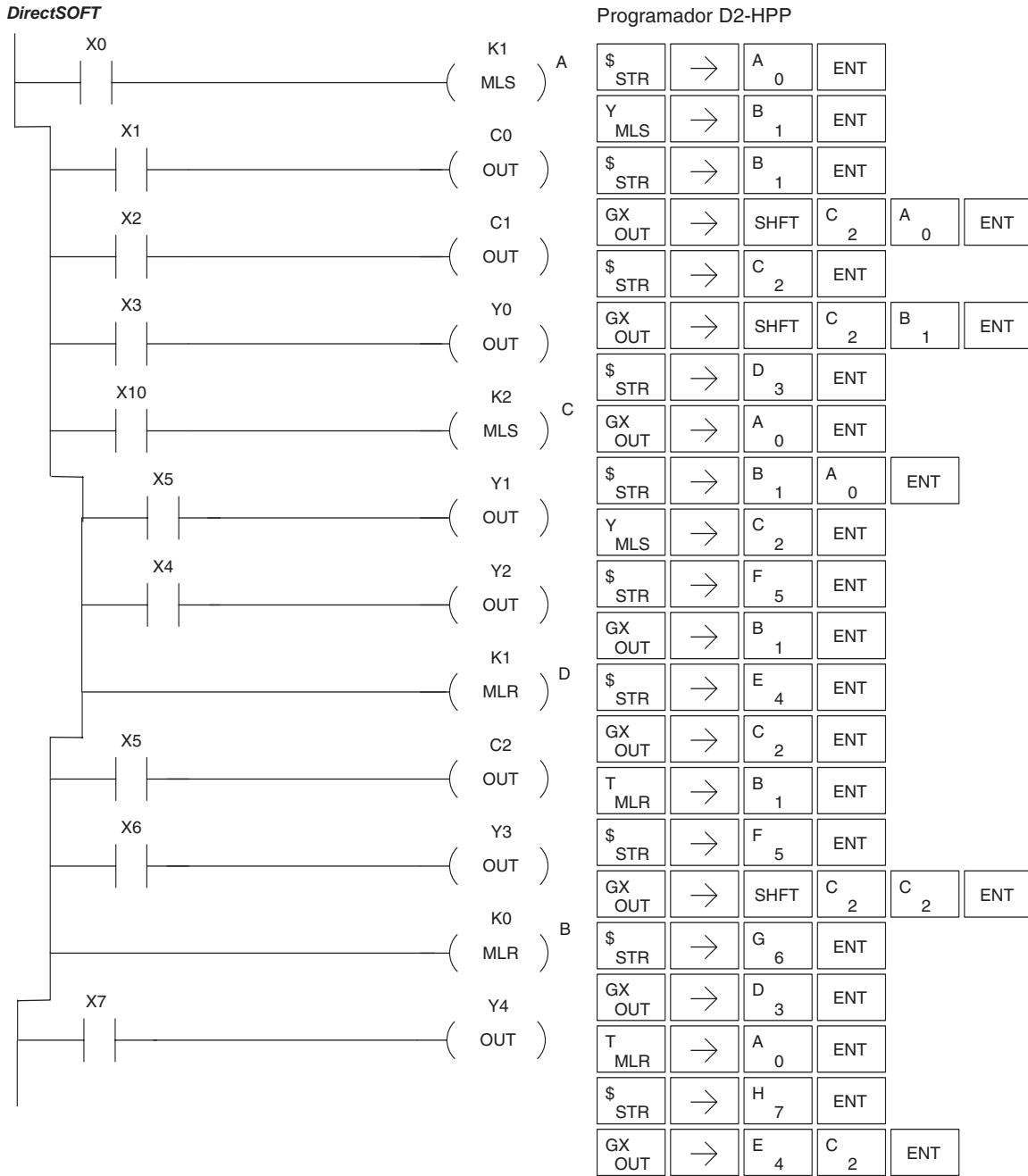
El ejemplo siguiente muestra cómo las instrucciones MLS y MLR operan creando un riel de energía secundario en la lógica de control.

DirectSOFT



Ejemplo de MLS/MLR

En el programa del ejemplo siguiente con MLS/MLR, la lógica funcionará entre el primer MLS K1 (A) y MLR K0 (B) sólo si la entrada X0 está ON. La lógica entre el MLS K2 (C) y MLR K1 (D) funcionará sólo si la entrada X10 y X0 están ON . El último renglón no es controlado por ninguna de las bobinas MLS.



Instrucciones de acción de interrupción

La instrucción Interrupt (INT)

DS5	Usado
HPP	Usado

La instrucción INT permite que sea colocada una sección de lógica ladder debajo del cuerpo principal del programa y ejecutada sólo cuando sea necesario. Los modos HSIO de alta velocidad de entradas y salidas 10, 20, y 40 pueden engendrar una interrupción. Con el modo 40, usted puede escoger una interrupción externa (la entrada X0) o una interrupción basada en tiempo (entre 3-999 ms).



Típicamente, las interrupciones se usan en una aplicación cuando se necesita una respuesta rápida a una entrada o cuando se debe ejecutar una sección de programa más rápido que el barrido normal de la CPU. La instrucción etiqueta de interrupción (Interrupt label) y toda la lógica asociada se deben colocar después de la declaración END en el programa. Cuando ocurre una interrupción, la CPU completará la ejecución de la instrucción que se está procesando en la lógica ladder y luego ejecuta la rutina de interrupción. Después de la ejecución de la rutina de interrupción el programa ladder reanuda del punto en que se interrumpió.

Vea la sección de operación del modo 40 (interrupción) para más detalles en la configuración de interrupción. En el DL06, sólo hay disponible una interrupción de software. La interrupción de software usa el interrupt # 00 (INT 0), que significa que el hardware interrupt #0 y el software interrupt no se pueden usar juntas. Las interrupciones de hardware se marcan en octal para corresponder con la señal de la entrada de hardware (Por ejemplo, X1 iniciará INT 1).

Tipo de operando de datos	Rango del DL06
Constante..... 0	aaa 1-FFFF

La instrucción Interrupt Return (IRT)

DS5	Usado
HPP	Usado

La instrucción IRT se ejecuta normalmente como la última instrucción en la rutina de interrupción. Vuelve la CPU al punto en el programa principal de donde se llamó. IRT es una instrucción incondicional (no se necesita contacto de entrada en el renglón).



La instrucción Interrupt Return Conditional (IRTC)

DS5	Usado
HPP	Usado

IRTC es una instrucción opcional usada con un contacto de entrada para causar un regreso condicional de la rutina de interrupción. En todo caso se requiere IRT para terminar la rutina de interrupción.



La instrucción Enable Interrupts (ENI)

DS5	Usado
HPP	Usado

La instrucción ENI se coloca en el programa principal ladder (antes de la instrucción END) para posibilitar la interrupción. La interrupción permanece habilitada hasta que el programa ejecute una instrucción DISI.



La instrucción Disable Interrupts (DISI)

DS5	Usado
HPP	Usado

Una instrucción DISI en el cuerpo principal del programa de aplicación (antes la instrucción END) incapacitará la interrupción (ya sea externa o por tiempo). La interrupción permanece incapacitada hasta que el programa ejecute una instrucción ENI.

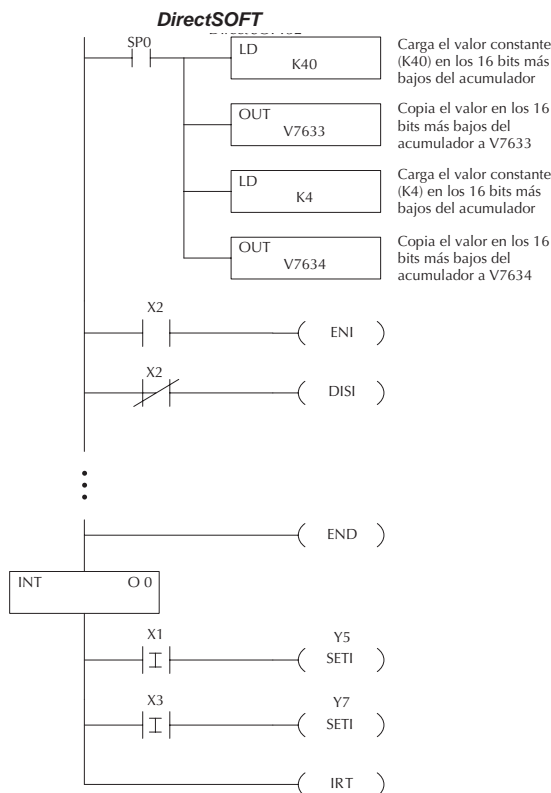


Ejemplo de programa de interrupción externa

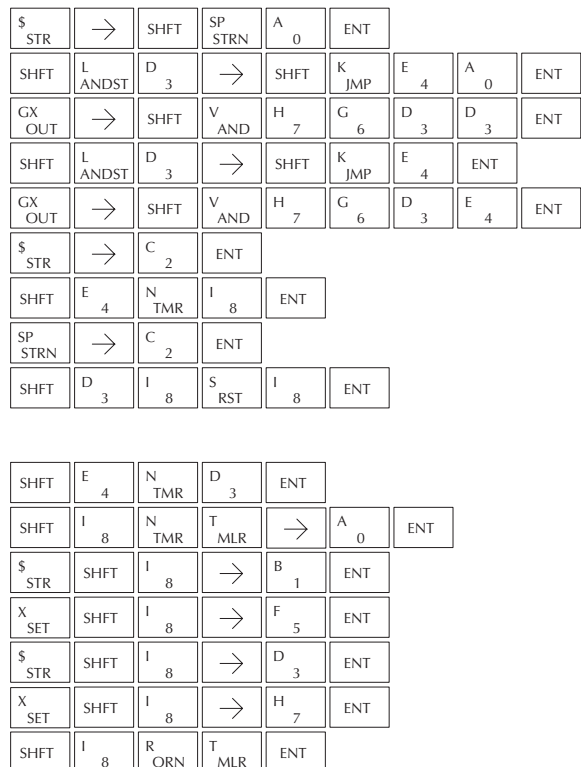
En el ejemplo siguiente, se hace una inicialización en el primer barrido usando el contacto de primer barrido SP0. La característica de interrupción es el modo HSIO 40. Luego se configura X0 como interrupción externa escribiendo al registro de configuración correspondiente, V7634. Vea la operación del modo 40 en el capítulo 3 para más detalles.

Durante la ejecución del programa, se activa la interrupción cuándo X2 está ON. Cuándo X2 está apagado se incapacitará la interrupción. Cuándo ocurre una señal de interrupción (X0) la CPU saltará a la marca de interrupción INT 00.

Se realizará entonces la lógica de aplicación en la rutina de interrupción. La CPU volverá al cuerpo principal del programa después que se ejecuta la instrucción IRT.



Programador D2-HPP



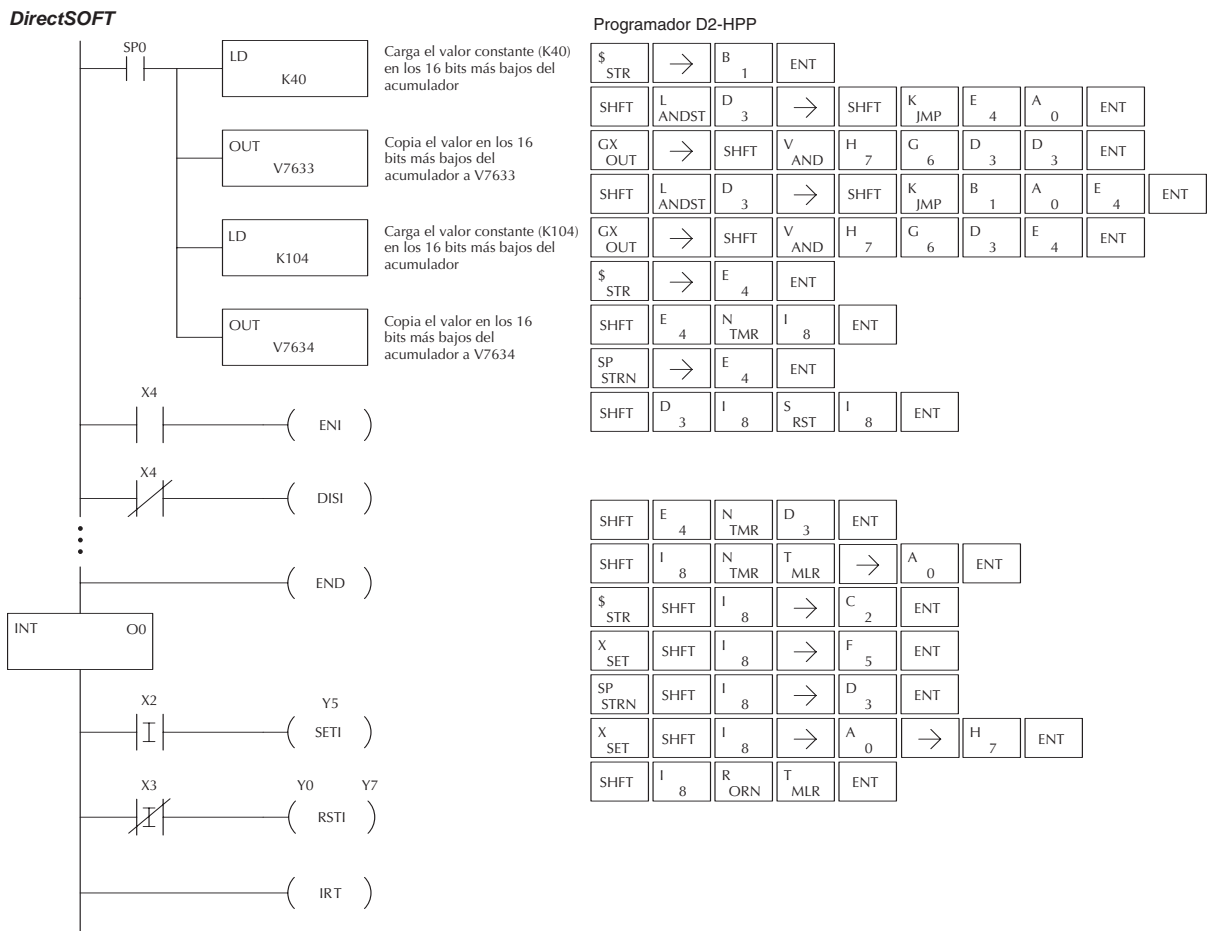
Ejemplo de programa de interrupción por tiempo

En el ejemplo siguiente, se hace una inicialización en el primer barrido, usando el contacto SP0 de primer barrido. La característica de interrupción es el modo HSIO 40. Luego se configura el temporizador de HSIO como una interrupción de 10 ms escribiendo K104 al registro de configuración para X0 (V7634).

Vea la operación del modo 40 en el capítulo 3 para más detalles. Cuando X4 prende, la interrupción se habilitará.

Cuando X4 se apaga, la interrupción se incapacitará. Cada 10 ms la CPU saltará a la marca de interrupción INT O0. Se ejecutará la lógica de aplicación en la rutina de interrupción.

Si X3 no está ON Y0-Y7 será colocado OFF y luego la CPU volverá al cuerpo principal del programa.

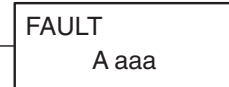


Instrucciones de mensajes

La instrucción Fault (FAULT)

DS5	Usado
HPP	Usado

La instrucción Fault se usa para mostrar un mensaje de FALLA en un programador portátil, o el visor opcional LCD o en el menú PLC>Diagnostics> messages> Fault messages en *DirectSOFT*. El mensaje tiene un máximo de 23 caracteres y puede ser datos de memoria V, datos o constantes numéricas o texto ASCII.



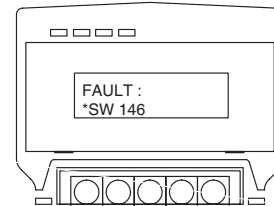
Para mostrar un valor en una dirección de memoria V, especifique la dirección de memoria V en la instrucción. Para mostrar los datos en las instrucciones ACON (constante ASCII) o NCON (constante numérica), especifique el valor de la constante (K) para el área correspondiente de etiqueta de datos (Data Label).

Tipo de operando de datos	Rango del DL06
	aaa
Memoria V..... V	Vea el mapa de memoria
Constante K	1-FFFF

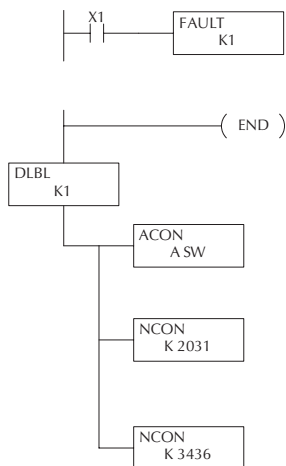
Indicadores	Descripción
SP50	ON cuando la instrucción FAULT es ejecutada

Ejemplo de instrucción Fault

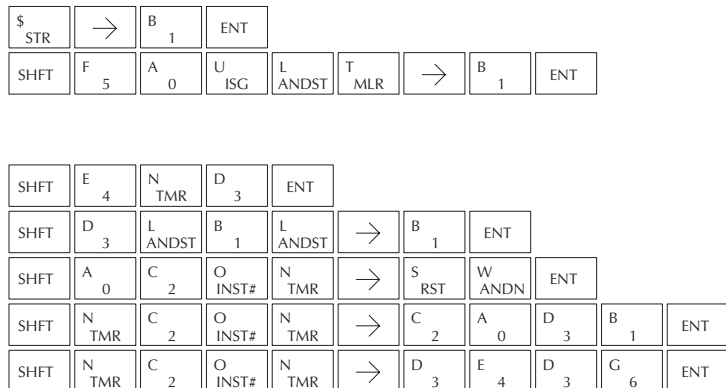
En el ejemplo siguiente cuando X1 está ON, se mostrará el mensaje “Baja presión 5” en el Programador portátil o en el visor LCD. El NCON usa el equivalente hexadecimal ASCII del texto a ser mostrado.



DirectSOFT



Programador D2-HPP



La instrucción Data Label (DLBL)

DS5	Usado
HPP	Usado

La instrucción DLBL marca el comienzo de un área ASCII/numérica de datos. DLBLS se programa después la declaración END.

```
DLBL      K aaa
```

Se puede usar un máximo de 64 instrucciones de DLBL en un programa. Se pueden usar múltiples NCONs y ACONs en un área de DLBL.

Tipo de operando de datos	Rango del DL06
	aaa
Constante K	1-FFFF

5

La instrucción ASCII Constant (ACON)

DS5	Usado
HPP	Usado

La instrucción ACON se usa con la instrucción DLBL para almacenar texto ASCII para uso con otras instrucciones. Se pueden almacenar 2 caracteres ASCII en una instrucción ACON.

```
ACON
      A aaa
```

Si se almacena solamente un carácter en un ACON será insertado un espacio delantero.

Tipo de operando de datos	Rango del DL06
	aaa
ASCII A	0-9 A-Z

La instrucción Numerical Constant (NCON)

DS5	Usado
HPP	Usado

La instrucción NCON se usa con la instrucción DLBL para almacenar el equivalente hexadecimal ASCII de datos numéricos para el uso con otras instrucciones.

```
NCON
      K aaa
```

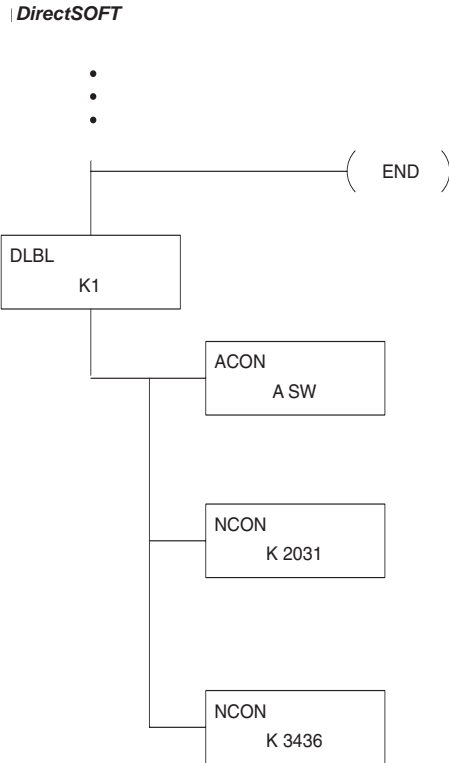
Se pueden almacenar 2 dígitos en una instrucción de NCON.

Tipo de operando de datos	Rango del DL06
	aaa
Constante K	1-FFFF

Ejemplo de Data Label

En el ejemplo siguiente, se usa un ACON y 2 instrucciones de NCON dentro de una instrucción DLBL para construir un mensaje de texto.

Vea la instrucción FAULT para información de cómo exhibir los mensajes. El Manual del visor DV-1000 tiene también información de los mensajes a ser exhibidos.



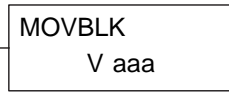
Programador D2-HPP

SHFT	E 4	N TMR	D 3	ENT						
SHFT	D 3	L ANDST	B 1	L ANDST	→	B 1	ENT			
SHFT	A 0	C 2	O INST#	N TMR	→	S RST	W ANDN	ENT		
SHFT	N TMR	C 2	O INST#	N TMR	→	C 2	A 0	D 3	B 1	ENT
SHFT	N TMR	C 2	O INST#	N TMR	→	D 3	E 4	D 3	G 6	ENT

La instrucción Move Block (MOVBLK)

DS5	Usado
HPP	Usado

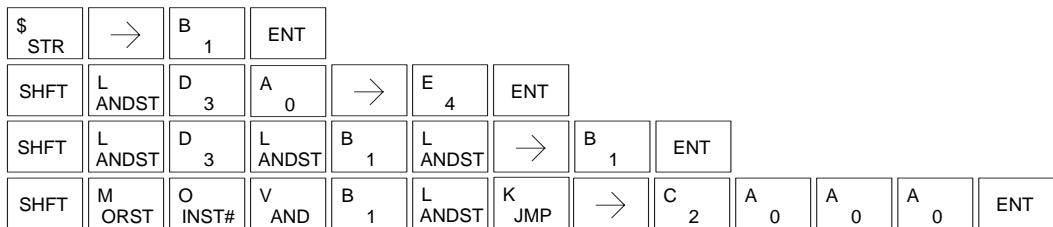
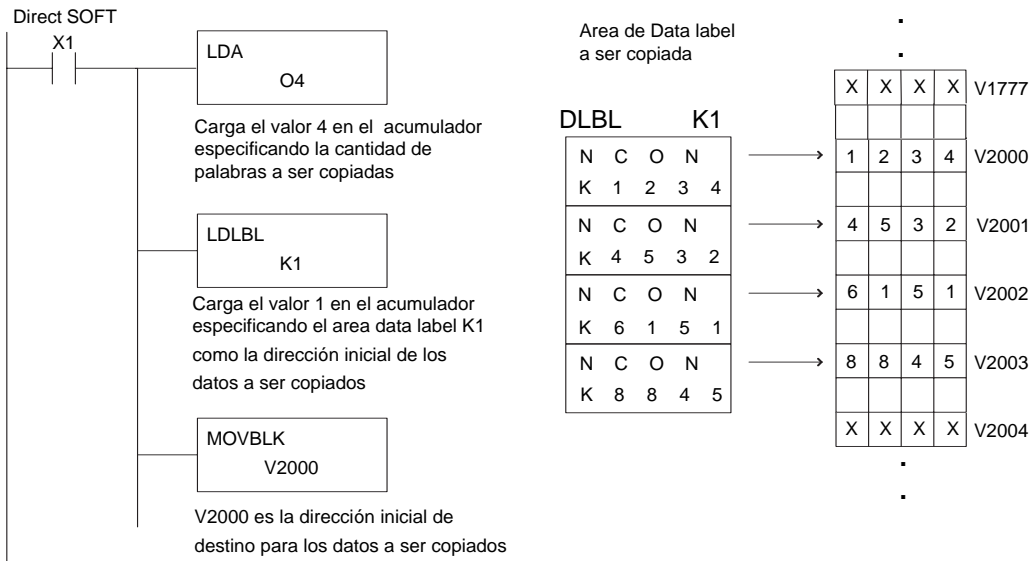
Esta instrucción copia un número especificado de palabras de un área de etiqueta de datos (Data Label) de la memoria del programa (ACON, NCON) a la localización especificada de Memoria. Se describen a continuación los pasos para usar esta instrucción:



- Paso 1: Cargue la cantidad de palabras (octal) que se copiarán al primer nivel del stack del acumulador
- Paso 2: Cargue la etiqueta de datos origen (LDLBL Kaaa) en el acumulador. De aquí es de donde serán copiados los datos.
- Paso 3: Insiera la instrucción MOVBLK que especifica la memoria de destino. Aquí es donde serán copiados los datos.

La instrucción Copy Data From a Data Label Area to Memory

Cuando X1 está encendido, el valor octal (O4) se copia al primer nivel del stack del acumulador usando el instrucción LDA. Este valor especifica la cantidad de palabras que se copiarán. La instrucción LDLBL cargará la dirección de los datos origen (K1) en el acumulador. Aquí es de donde serán copiados los datos. La instrucción MOVBLK especifica la localización inicial de destino y ejecuta el copiado de datos del área de la etiqueta de datos a Memoria.

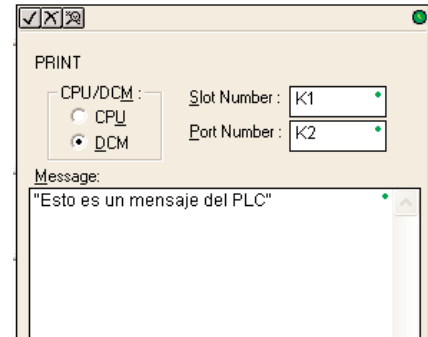


La instrucción Print Message (PRINT)

DS5	Usado
HPP	N/A

La instrucción PRINT imprime un mensaje con texto o con texto y variable o datos empotrados al puerto 2 en la CPU DL06 o el módulo D0-DCM, el cual debe estar configurado adecuadamente con el protocolo Non-sequence.

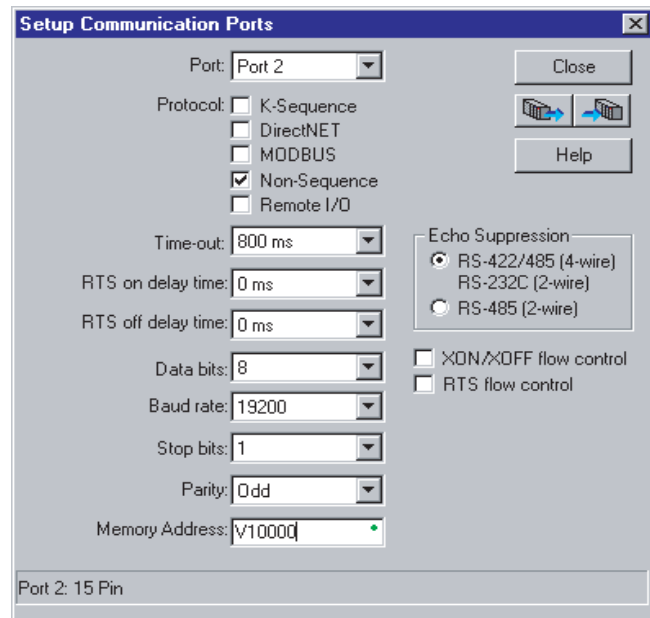
Tipo de operando de datos	Rango del DL06
	aaa
Constante	K 2 para puerto; 1-4 para ranura



5

Usted puede recordar de las especificaciones del PLC DL06 que los puertos son capaces de procesar varios protocolos. El puerto 1 no puede ser configurado como protocolo "Non-sequence". Para configurar el puerto 2 en *DirectSOFT*, escoja el menú " PLC", luego **SETUP**, luego "Setup Sec. Comm Port". Aparece un cuadro de diálogo como la figura de abajo :

- **Port:** En el campo de la lista de puertos disponibles en la parte superior, escoja "Port 2"
- **Protocol:** Haga clic en el cuadro de verificación a la izquierda de "Non- sequence"
- **Time-out:** El período que el puerto esperará después que envíe un mensaje para obtener una respuesta antes de detectar un error.
- **RTS On Delay Time:** tiempo que espera el PLC para mandar datos después que la señal TRS se ha hecho ON.
- **RTS Off Delay Time:** tiempo que espera el PLC DL06 después de mandar datos para hacer OFF la señal TRS.
- **Data Bits:** Seleccione 7 o 8 bits y hágalo igual a los bits de datos especificados para los aparatos conectados.
- **Baud Rate:** Las tasas disponibles de baud incluyen 300, 600, 1200, 2400, 4800, 9600, 19200, y 38400 Baud. Escoja una tasa más alta de baud inicialmente, y baje el valor si experimenta errores de datos o problemas de ruido en la red.



- **Stop Bits:** Escoja 1 o 2 bits de parada que debe ser los mismos que los de la impresora conectada.
- **Parity:** Escoja paridad none, even, o odd para verificación de error. Asegúrese de hacer igual la paridad especificada en la impresora a ser conectada.
- **Echo Suppression:** Seleccione el botón de radio adecuado basado en la configuración usada en el puerto 2 (RS-232C, RS-422 o RS-485).
- **Xon/Xoff Flow control:** Escoja esta selección si Ud. tiene el puerto 2 conectado para el control de flujo con hardware (Xon/Xoff) con las señales RTS y CTS conectada entre los dispositivos.
- **RTS Flow control:** Escoja esta selección si Ud. tiene la señal RTS del puerto 2 cableada a la impresora.
- **Memory address:** Escoja una dirección de memoria para usar como buffer para almacenamiento de datos ASCII.



Luego haga clic en el botón para enviar la configuración del puerto a la CPU y haga clic en CLOSE.

El puerto 2 en el DL06 tiene niveles de voltaje RS232 normales y debe trabajar con la mayoría de las conexiones seriales de las impresoras. .

Elemento de texto – esto se usa para imprimir conjuntos de caracteres. Los conjuntos de caracteres se definen como los caracteres (fuera de 0) entre comillas. Dos números hexadecimales precedidos por el signo de dólar significa un código de 8 bits de caracteres ASCII. También, dos caracteres precedidos por el signo de dólar se interpretan según la tabla siguiente:

#	Código de caracteres	Descripción
1	\$\$	Signo dólar (\$)
2	\$"	Comillas (")
3	\$L o \$l	Line feed (LF)
4	\$N o \$n	Carriage return line feed (CRLF)
5	\$P o \$p	Form feed
6	\$R o \$r	Carriage return (CR)
7	\$T o \$t	Tab

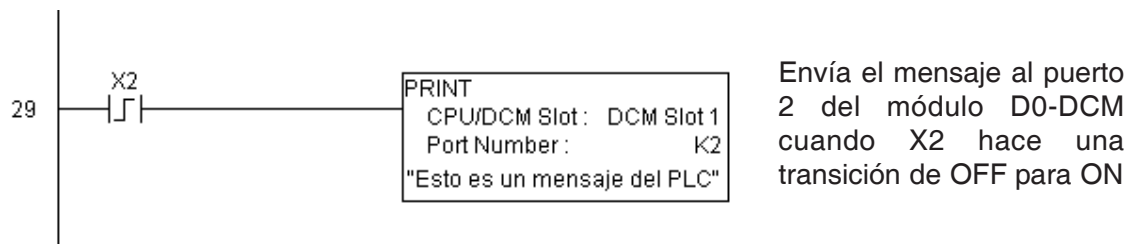
Los ejemplos siguientes muestran varias convenciones de sintaxis y la longitud de las señales de salida a la impresora.

Ejemplo:

- ” ” Longitud 0 sin el carácter
- ”A” Longitud 1 con el carácter A
- ” ” Longitud 1 con espacio en blanco
- ” \$ ” Longitud 1 con comillas
- ” \$ R \$ L ” Longitud 2 con un CR y un LF
- ” \$ 0 D \$ 0 A ” Longitud 2 con un CR y un LF
- ” \$ \$ ” Longitud 1 con la marca \$

Al imprimir una línea ordinaria de texto, usted necesitará incluir "comillas" antes y después del conjunto de texto. El código de error 499 ocurrirá en la CPU cuando la instrucción PRINT contiene texto inválido o está sin comillas. Es importante probar los datos de la instrucción PRINT durante el desarrollo de la aplicación.

El ejemplo siguiente imprime el mensaje al puerto 2. Usamos un contacto PD, que causa que la instrucción de mensaje sea activa por un barrido solamente. Note el \$N al fin del mensaje, que produce un carriage return/line feed en la impresora. Esto prepara la impresora para imprimir la próxima línea y comenzar desde el margen izquierdo.



Elemento de memoria V - esto se usa para imprimir el contenido de memorias V en el formato entero o real. Use el número de memoria V o el número de memoria V con ":" y el tipo de datos. Los tipos de datos se muestran en la tabla abajo. El código del carácter debe ser con letras mayúsculas.



NOTA: Debe colocarse un espacio antes y después de la dirección de memoria V para separarla de la cadena de texto. Si no se hace esto aparecerá el error 499.

#	Código de caracteres	Descripción
1	ninguno	Binario de 16 bits (decimal)
2	: B	4 dígitos BCD
3	: D	Binario de 32 bits (decimal)
4	: D B	8 dígitos BCD

Ejemplos:

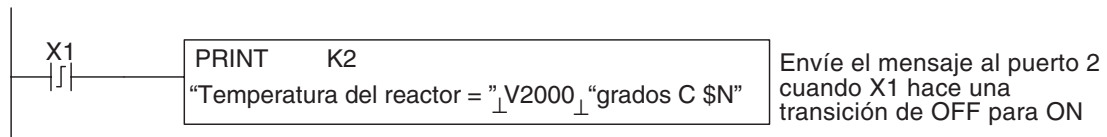
V2000 Imprime datos binarios en V2000 como número decimal

V2000 : B Imprime datos BCD en V2000

V2000 : D Imprime un número binario en V2000 y V2001 para un número decimal

V2000 : D B Imprime datos BCD en V2000 y V2001

Ejemplo: El ejemplo siguiente imprime un texto que contiene un texto y una variable. "Temperatura del reactor" marca los datos, que están en V2000 (como binario). Usted puede usar el calificativo :B después de V2000 si los datos están en el formato BCD, por ejemplo. El texto final agrega las unidades de grados a la línea de texto y el \$N agrega un Carriage return/Line feed (CRLF), un comando de la impresora.



El mensaje será visto como:

Temperatura del reactor = 0156 grados C

Envíe el mensaje al puerto 2 cuando X1 hace una transición de OFF para ON

┘ representa un espacio

V2000 contiene el valor 156 binario

Elemento Texto de memoria V - Esto se usa para imprimir texto almacenado en memoria V. Use el signo % seguido por el número de caracteres después del número de memoria V para representar el texto.

Si usted asigna "0" como el número de caracteres, la función de impresión leerá al conteo desde el carácter de la primera localización. Luego comenzará en la próxima dirección de memoria V y leerá ese número de códigos ASCII para el texto desde la memoria.

Ejemplo:

V2000 % 16 Se imprimen 16 caracteres en V2000 hasta V2007.

V2000 % 0 Serán impresos los caracteres en V2001 a Vxxxx (determinado por el número en V2000).

Elemento de bit

Esto se usa para imprimir el estado del bit designado en la memoria V o un bit de relevador C. El elemento bit puede ser asignado por un punto (.) y el número de bit precedido por el número de memoria V o el número de relevador C. El tipo de salida se describe como mostrado en la tabla de abajo.

#	Formato de datos	Descripción
1	ninguno	Imprime 1 cuando el estado es ON y 0 cuando el estado es OFF
2	:BOOL	Imprime "TRUE" cuando el estado es ON y "FALSE" cuando el estado es OFF
3	:ONOFF	Imprime "ON" cuando el estado es ON y "OFF" cuando el estado es OFF

Ejemplo:

V2000.15 Imprime el estado del bit 15 en V2000, en formato 1/0.

C100 Imprime el estado de C100 en el formato 1/0.

C100 : BOOL Imprime el estado de C100 en formato TRUE/FALSE

C100 : ON/OFF Imprime el estado de C100 en formato ON/OFF

V2000.15 : BOOL Imprime el estado del bit 15 en V2000 en formato TRUE/FALSE

El máximo número de caracteres que puede imprimir es 128. El número de caracteres para cada elemento se lista en la tabla de abajo:

Tipo de elemento	Cantidad máxima de caracteres
Texto, 1 carácter	1
16 bit binarios	6
32 bits binarios	11
4 dígitos BCD	4
8 dígitos BCD	8
Número real (punto flotante)	12
Real con exponente	12
Texto en una memoria V	2
Bit (formato 0/1)	1
Bit (formato TRUE/FALSE)	5
Bit (formato ON/OFF)	3

El nemotécnico del programador D2-HPP es "PRINT" seguido del campo DEF.

Los relevadores especiales SP116 y SP117 indican el estado de los puertos del PLC DL06 **busy** (Ocupado), o **communications error** (error de comunicación)).

Vea el apéndice D relativo a relevadores especiales para una mejor descripción.



NOTA: Usted debe usar el relevador especial apropiado con la instrucción PRINT para asegurarse que el programa ladder no trate de IMPRIMIR a un puerto que está todavía ocupado por una instrucción PRINT previa o una instrucción WX o RX. .

Instrucciones de módulos inteligentes

La instrucción Read from Intelligent Module (RD)

DS32	Usado
HPP	Usado

Esta instrucción lee un bloque de datos (máximo cantidad de 128 bytes) de un módulo inteligente de E/S a la memoria de la CPU. Cargue los parámetros de la función en el primer y segundo nivel del stack del acumulador y al acumulador por tres instrucciones adicionales



Se enumeran abajo los pasos para programar esta instrucción.

- Paso 1: Cargue el número de la base en el primer byte y el número de la ranura (1 a 4) o al segundo byte del segundo nivel del Stack del acumulador.
- Paso 2: Cargue el número de bytes a ser transferidos al primer nivel del Stack del acumulador. Pueden ser transferidos hasta 128 bytes (o 64 palabras de 16 bits) en cada transacción, ya que hay 2 bytes por cada palabra.
- Paso 3: Cargue la dirección de los datos desde donde van a ser leídos. Este parámetro requiere un valor hexadecimal.
- Paso 4: Coloque la instrucción RD especificando la dirección inicial de memoria V (Aaaa) de donde se leerán los datos.

Sugerencia: — Para parámetros que requieren valores hexadecimales, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor en el acumulador.

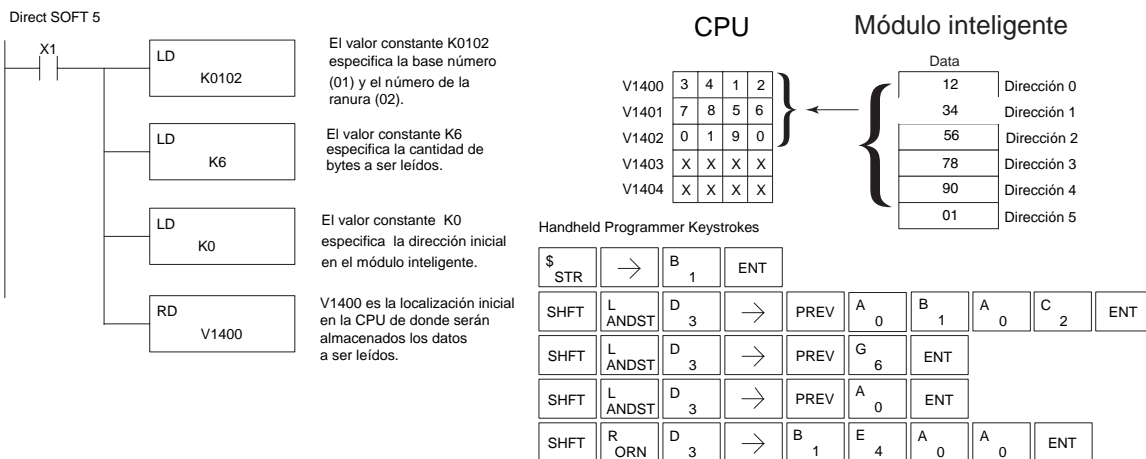
Tipo de operando de datos	Rango del DL06
	aaa
Memoria V V	Vea el mapa de memoria

Indicadores	Descripción
SP54	On cuando las instrucciones RX, WX RD, WT son ejecutada con parámetros errados.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

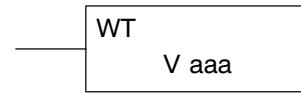
En el ejemplo siguiente cuando X1 está ENCENDIDO, la instrucción RD leerá seis bytes de datos de un módulo inteligente en la base 1, ranura 2 que comienzan en la dirección 0 en el módulo inteligente y copiará la información en las localizaciones de memoria V1400 hasta V1402



La instrucción Write to Intelligent Module (WT)

DS32	Usado
HPP	Usado

Esta instrucción escribe un bloque de datos (máximo de 128 bytes) a un módulo inteligente de E/S desde un bloque de Memoria en la CPU. Los parámetros de la instrucción son cargados en el primer y segundo nivel del stack del acumulador y el acumulador por tres instrucciones adicionales.



Se enumeran abajo los pasos para programar esta instrucción.

- Paso 1: Cargue el número de la base en el primer byte y el número de la ranura (1 a 4) o al segundo byte del segundo nivel del Stack del acumulador.
- Paso 2: Cargue el número de bytes a ser transferidos al primer nivel del Stack del acumulador. Pueden ser transferidos hasta 128 bytes (o 64 palabras de 16 bits) en cada transacción, ya que hay 2 bytes por cada palabra.
- Paso 3: Cargue la dirección de los datos del módulo inteligente donde van a ser recibidos los datos. Este parámetro requiere un valor hexadecimal.
- Paso 4: Coloque la instrucción WT que especifica la dirección inicial de memoria V (Aaaa) en la CPU desde donde se leerán los datos.

Sugerencia: — Para parámetros que requieren valores hexadecimales, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor en el acumulador.

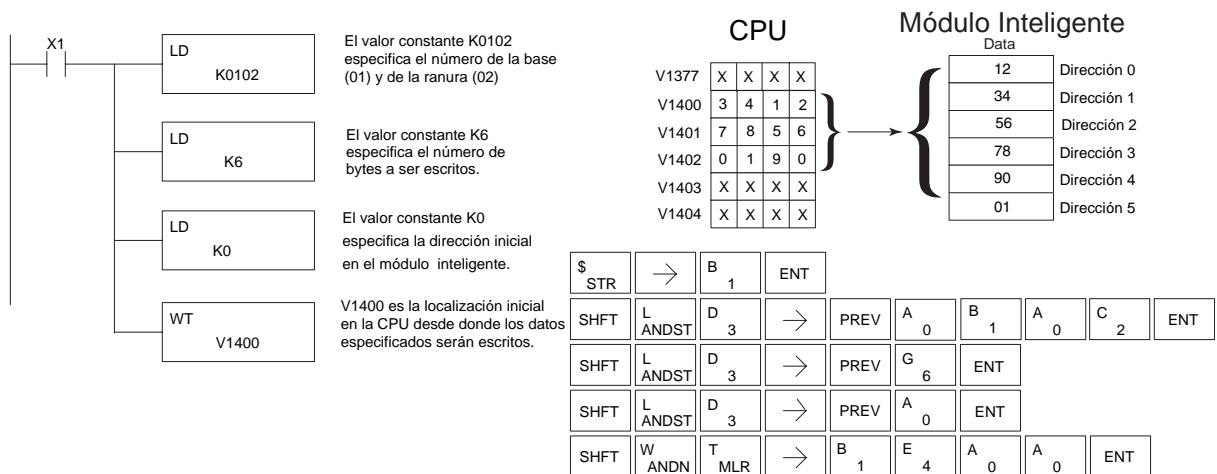
Tipo de operando de datos	Rango del DL06
Memoria V V	aaa Vea el mapa de memoria

Indicadores	Descripción
SP54	On cuando se ejecutan las instrucciones RX, WX RD, WT con parámetros errados.



NOTA: Las indicaciones de estado discretas SP son válidas sólo hasta que se ejecute otra instrucción que use el mismo relevador especial SP.

En el ejemplo siguiente, cuando X1 está encendido, la instrucción WT escribe seis bytes de datos a un módulo inteligente en la base 1, ranura 2 que comienzan en la dirección 0 en el módulo inteligente y copia los datos de las localizaciones de Memoria V1400 hasta V1402

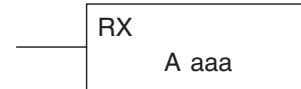


Instrucciones de comunicación en una red

La instrucción Read from Network (RX)

DS32	Usado
HPP	Usado

La instrucción RX es usada por el aparato maestro en una red para leer un bloque de datos de un aparato esclavo en la misma red. Los parámetros de la función son cargados al primer y segundo nivel del Stack del acumulador y al acumulador con tres instrucciones adicionales. Abajo están listados los pasos necesarios para programar la instrucción RX.

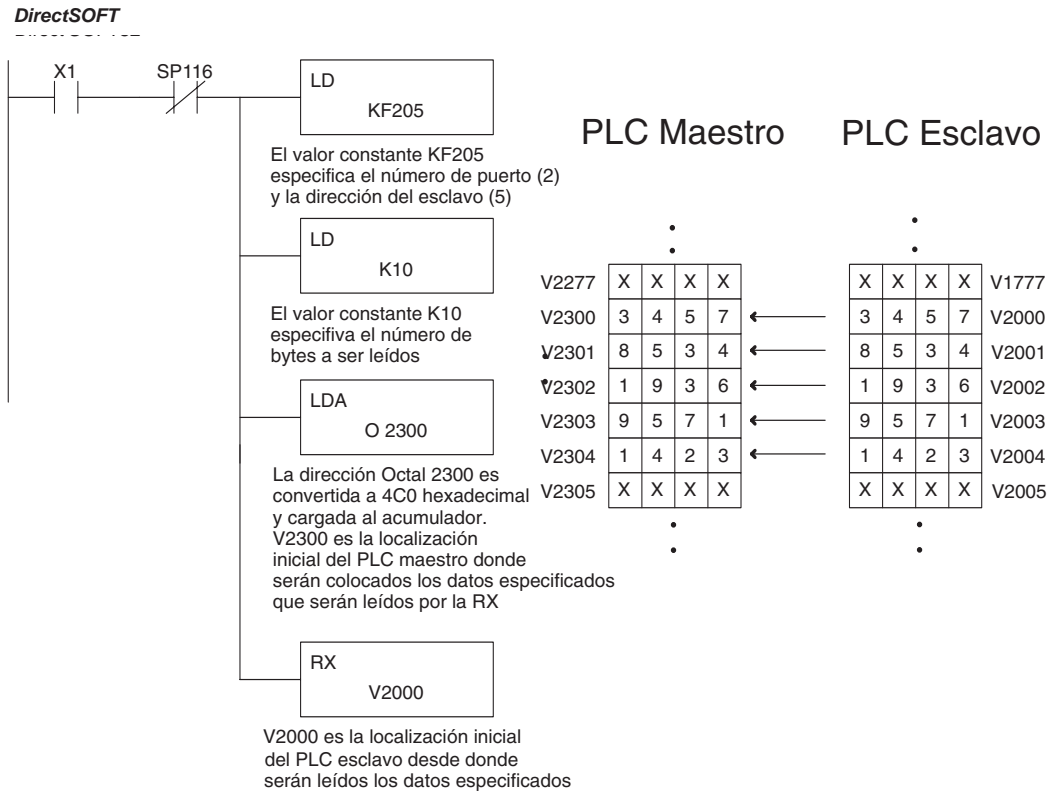


- Paso 1: Cargue la dirección del esclavo (0- 90 BCD) en el primer byte y el puerto del PLC (KF2) o ECOM maestro (0- 7) al segundo byte del segundo nivel del Stack del acumulador.
- Paso 2: Cargue el número de bytes a ser transferidos al primer nivel del Stack del acumulador. Pueden ser transferidos hasta 128 bytes (o 64 palabras de 16 bits) en cada transacción, ya que hay 2 bytes por cada palabra.
- Paso 3: Cargue la dirección de los datos a ser leída en el aparato esclavo al acumulador. Este parámetro requiere un valor hexadecimal.
- Paso 4: Coloque la instrucción RX especificando la dirección inicial de memoria V (Aaaa) en el aparato esclavo de donde se leerán los datos.

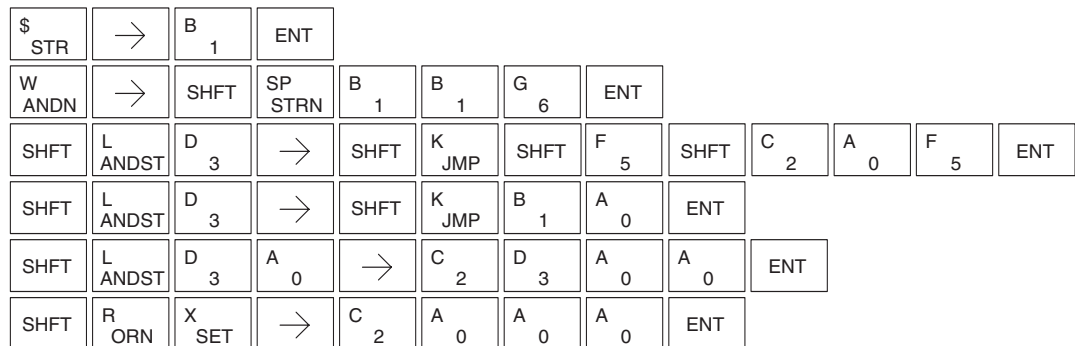
Sugerencia: — Para parámetros que requieren valores hexadecimales, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor en el acumulador.

Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V	V
Puntero	P
Entradas	X
Salidas	Y
Relevadores de control	C
Bits de etapas	S
Bits de temporizadores	T
Bits de contadores	CT
Relevadores especiales	SP
Memoria de programa	\$
	0-7680 (2K de memoria de programa)

En el ejemplo siguiente, cuándo X1 está ON y el relevador SP116 del puerto “busy” (ocupado) (vea relevadores especiales) no está ON, la instrucción RX tendrá acceso al puerto 2, que opera como maestro. Serán leídos diez bytes consecutivos o 5 palabras de datos (V2000 - V2004) de un aparato esclavo en el nodo de dirección 5 y copiado a las memorias V2300-V2304 en la CPU con el puerto maestro.



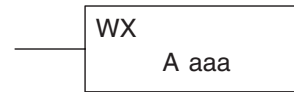
Programador D2-HPP



La instrucción Write a Network (WX)

DS5	Usado
HPP	Usado

La instrucción WX se usa para escribir un bloque de datos desde el aparato maestro a un aparato esclavo en la misma red. Los parámetros de la instrucción WX son cargados al acumulador y al primer y segundo nivel del Stack. Abajo están listados los pasos necesarios para ejecutar un programa para escribir datos en la red.



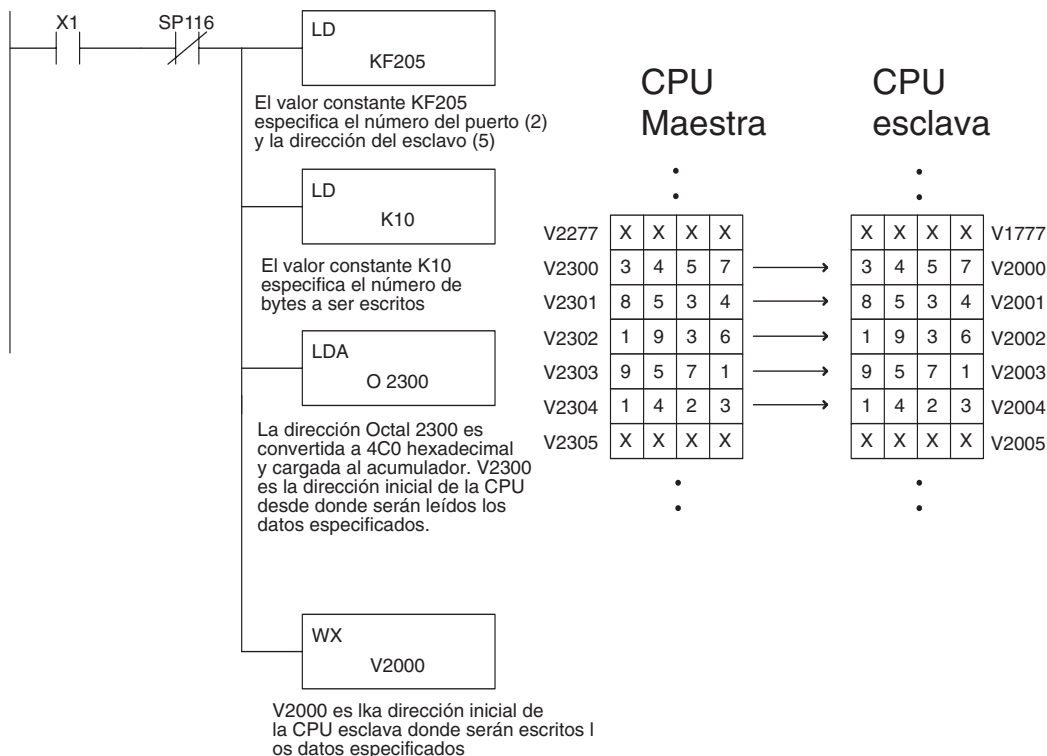
- Paso 1: Cargue la dirección de esclavo (0-90 BCD) en el byte bajo y "F2" o ECOM maestro (0-7) en el byte alto del acumulador (las próximas dos instrucciones empujan esta palabra hacia abajo al segundo nivel del Stack).
- Paso 2: Cargue el número de bytes a ser transferido al acumulador (la próxima instrucción empuja esta palabra al primer nivel del Stack). Pueden ser transferidos hasta 128 bytes (o 64 palabras de 16 bits) en cada transacción.
- Paso 3: Cargue la dirección inicial de la CPU maestra al acumulador. Esta es la dirección de memoria de donde se escribirán los datos. Este parámetro requiere un valor hexadecimal.
- Paso 4: Coloque la instrucción WX especificando la dirección de memoria V (Aaaa) donde los datos serán escritos al esclavo.

Sugerencia: — Para parámetros que requieren valores hexadecimales, se puede usar la instrucción LDA para convertir una dirección octal al equivalente hexadecimal y cargar el valor al acumulador.

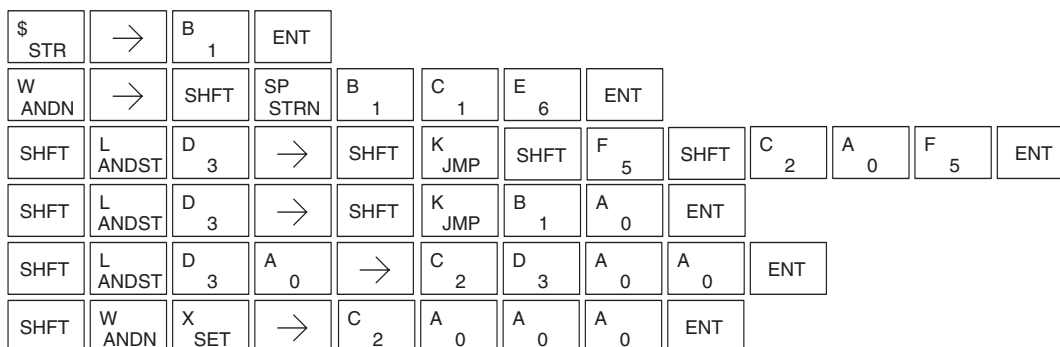
Tipo de operando de datos	Rango del DL06
..... A	aaa
Memoria V V	Vea el mapa de memoria
Puntero P	Vea el mapa de memoria
Entradas X	0-777
Salidas Y	0-777
Relevadores de control C	0-1777
Etapas S	0-1777
Bits de estado de temporizadores T	0-377
Bits de estado de contadores CT	0-177
Relevadores especiales SP	0-777
Memoria de programa \$	0-7680 (2K de memoria de programa)

En el ejemplo siguiente cuando X1 está ON y el relevador SP116 “busy” (ocupado) (vea los relevadores especiales) no está ON, la instrucción WX tendrá acceso al puerto 2 que opera como maestro. Se leen diez bytes o 5 palabras consecutivas de datos de la CPU maestra y son copiados a las memorias V2000-V2004 en el aparato esclavo en la dirección del nodo 5.

DirectSOFT



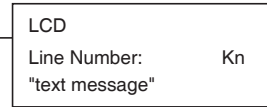
Programador D2-HPP



La instrucción LCD

DS5	Usado
HPP	N/A

La instrucción LCD causa que un mensaje definido de texto de usuario sea mostrado en el panel del visor LCD. El visor es 16 caracteres de ancho y 2 filas de alto de modo que puede ser mostrado un "mensaje de texto" de 32 caracteres. Cada fila se define separadamente; el número máximo de caracteres que cada instrucción aceptará es 16.



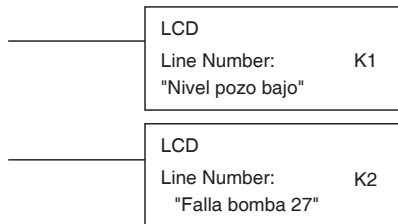
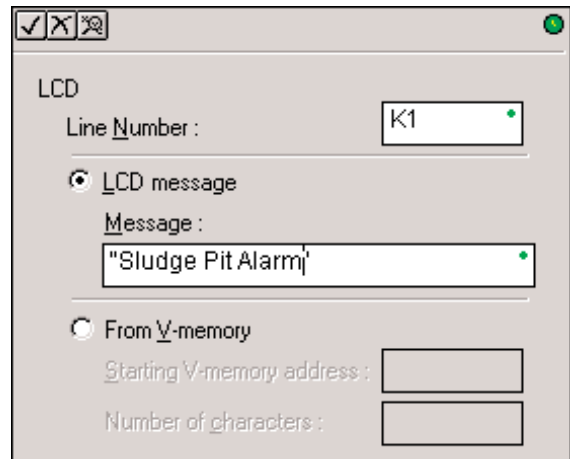
El mensaje de texto se puede entrar directamente en el campo de mensaje del diálogo de la instrucción o se puede localizar dondequiera en la memoria V. Si el texto se localiza en la memoria V, se usa la instrucción LCD para señalar el inicio de la localización de la memoria donde está el texto deseado. También se debe colocar la longitud del texto.

De la carpeta del proyecto de *DirectSOFT32*, utilice el navegador de instrucciones para localizar la instrucción LCD o use F7. Cuando usted selecciona la instrucción LCD y hace clic en OK, aparecerá el diálogo del LCD, según lo mostrado en los ejemplos. La instrucción LCD se inserta en el programa ladder a través de esta caja de diálogo de configuración.

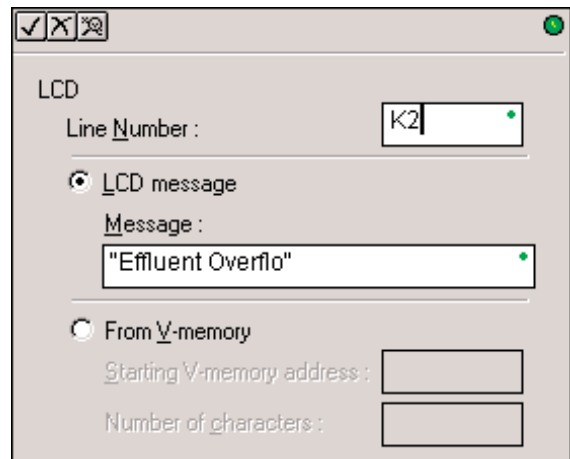
Los textos del visor pueden tener variables incluidas. Se puede incluir la fecha y la hora y valores de memoria V en el texto mostrado. Vea los ejemplos de cada caso.

Colocación directa de texto

Los dos diálogos muestran a la derecha las selecciones necesarias para crear las dos instrucciones ladder de la figura abajo. Deben colocarse comillas en el texto definido. En el primer diálogo, el texto "Nivel pozo bajo" usa quince espacios de caracteres y aparecerá en la línea 1 cuando la instrucción se activa. Note que el número de la línea es K1. Haciendo clic en el botón superior a la izquierda hace que la instrucción sea colocada en el programa.



Identificando el segundo número de la línea como K2, el texto "Falla bomba 27" aparecerá en la segunda línea del visor cuando se activa la segunda instrucción.

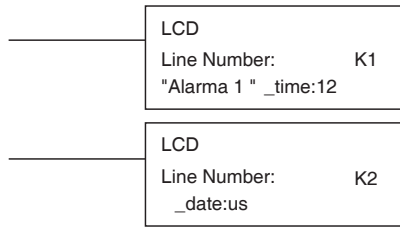


N	i	v	e	l		p	o	z	o		b	a	j	o
F	a	l	l	a		b	o	m	b	a		2	7	

Mostrando fecha y/o hora

Se puede incluir la fecha y/o la hora en el texto desplegado usando las variables listadas en la tabla abajo. Estas variables se pueden incluir en el campo "Message" del diálogo LCD. En el ejemplo, la variable de tiempo (formato de 12 horas) es incluida colocando `_time:12`. Este formato de tiempo usa un máximo de siete espacios de caracteres. El segundo diálogo crea una instrucción que imprime la fecha en la segunda línea del visor, cuando está activado.

Variables y formatos de fecha y hora		
<code>_date:us</code>	Formato de E.E.U.U	MM/DD/AA
<code>_date:e</code>	Formato de Europa	DD/MM/AA
<code>_date:a</code>	Formato de Asia	YY/MM/DD
<code>_time:12</code>	Formato de 12 horas	HH:MMAM/PM
<code>_time:24</code>	Formato de 24 horas	HH:MM:SS

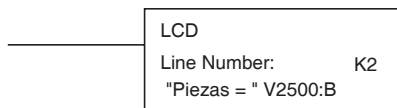


A	l	a	r	m	a		1		1		1	:	2		1	P	M
0	5	-	0	8	-	0	2										

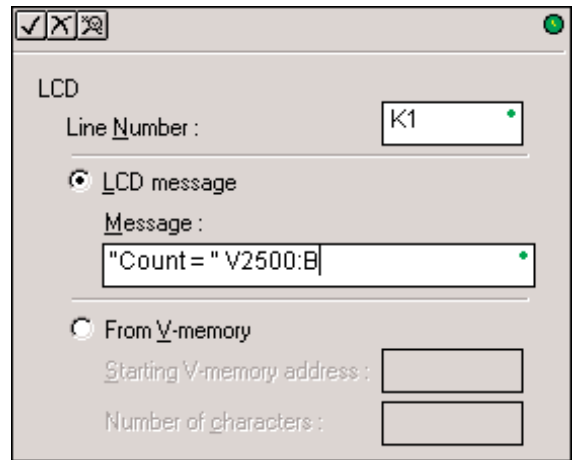
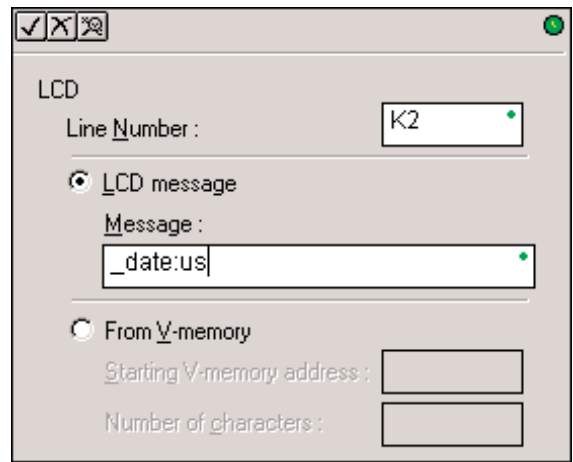
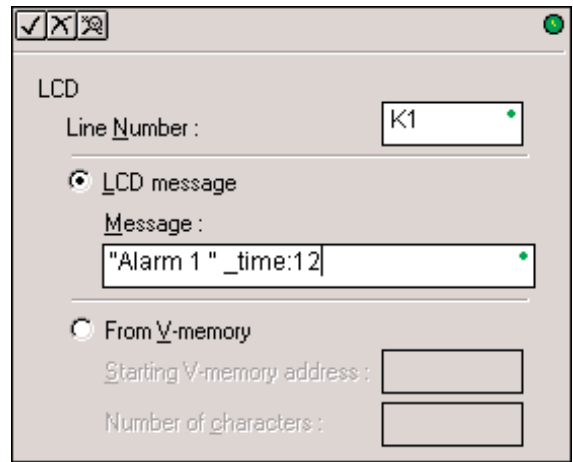
Colocando datos de memorias V

Cualquier dato en memoria puede ser mostrado en cualquiera de seis formatos disponibles de datos. Hay un ejemplo a la derecha.

Hay una lista de formatos de datos y sufijos en la próxima página. Note que formatos diferentes de datos necesitan de una cantidad diferente de caracteres en el visor.



P	i	e	z	a	s	=		0		4		1		2			



Sufijos de formatos de datos para datos incluidos de memoria V

Varios formatos de datos están disponibles para desplegar datos de memoria V en el LCD. Las opciones se muestran en la tabla a continuación. Se usan dos puntos para separar la localización incluida de memoria V del sufijo de formato de datos y calificativo. Vea el ejemplo anterior.

Formato de datos	Sufijo	Ejemplo	Caracteres mostrados															
ningún sufijo (Formato de 16-bits)		V2000 = 0000 0000 0001 0010	1	2	3	4												
		V2000			1	8												
	[:S]	V2000:S	1	8														
	[:C0]	V2000:C0	0	0	1	8												
	[:0]	V2000:0			1	8												
:B (4 dígitos BCD)		V2000 = 0000 0000 0001 0010	1	2	3	4												
	[:B]	V2000:B	0	0	1	2												
	[:BS]	V2000:BS	1	2														
	[:BC0]	V2000:BC0	0	0	1	2												
	[:B0]	V2000:B0			1	2												
:D (32 bits decimales)		V2000 = 0000 0000 0000 0000	Palabra doble															
		V2001 = 0000 0000 0000 0001	1	2	3	4	5	6	7	8	9	10	11					
	[:D]	V2000:D							6	5	5	3	6					
	[:DS]	V2000:DS	6	5	5	3	6											
	[:DC0]	V2000:DC0	0	0	0	0	0	0	6	5	5	3	6					
[:D0]	V2000:D0							6	5	5	3	6						
:DB (8 dígitos BCD)		V2000 = 0000 0000 0000 0000	Palabra doble															
		V2001 = 0000 0000 0000 0011	1	2	3	4	5	6	7	8								
	[:DB]	V2000:DB	0	0	0	3	0	0	0	0								
	[:DBS]	V2000:DBS	3	0	0	0	0											
	[:DBC0]	V2000:DBC0	0	0	0	3	0	0	0	0								
[:DB0]	V2000:DB0				3	0	0	0	0									
:R (Número real- Palabra doble)		V2001/V2000 = 222.11111 (real number)	Palabra doble															
			1	2	3	4	5	6	7	8	9	10	11	12	13			
	[:R]	V2000:R				f	2	2	2	.	1	1	1	1	1			
	[:RS]	V2000:RS	f	2	2	2	.	1	1	1	1	1						
	[:RC0]	V2000:RC0	f	0	0	0	2	2	2	.	1	1	1	1	1			
[:R0]	V2000:R0				f	2	2	2	.	1	1	1	1	1				
:E (Número real- Palabra doble con exponente)		V2001/V2000 = 222.1 (real number)	Palabra doble															
			1	2	3	4	5	6	7	8	9	10	11	12	13			
	[:E]	V2000:E		f	2	.	2	2	1	0	0	E	+	0	2			
	[:ES]	V2000:ES	f	2	.	2	2	1	0	0	E	+	0	2				
	[:EC0]	V2000:EC0	f	2	.	2	2	1	0	0	E	+	0	2				
[:E0]	V2000:E0	f	2	.	2	2	1	0	0	E	+	0	2					

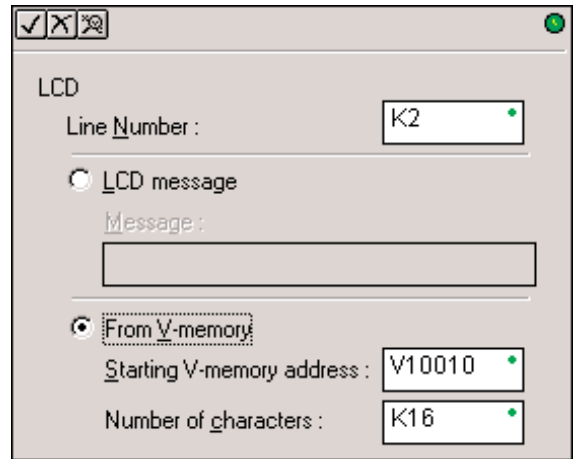
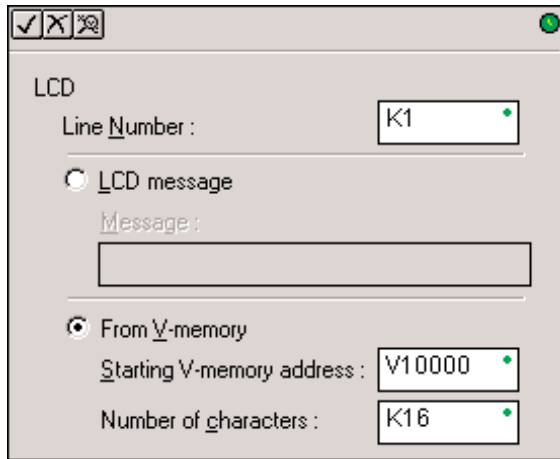
f = indicación más/ menos (más = sin símbolo, menos = -)

Los calificativos “S”, “C0”, y “0” alteran la presentación de ceros y espacios a la izquierda. “S” saca los espacios y justifica el resultado a la izquierda. “C0” reemplaza los espacios delanteros con ceros. “0” es una modificación de “C”0. “0” elimina cualquier cero delantero en la versión de formato “C0” y los convierte a espacios.

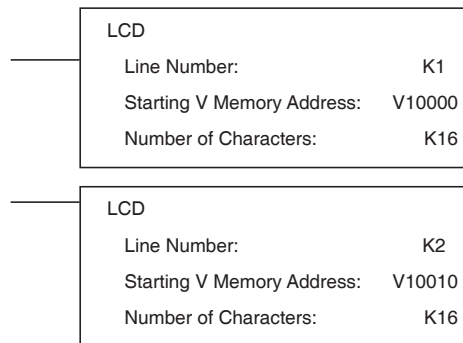
Colocación de texto desde la memoria V

Alternativamente, un texto que ya está en la memoria V se puede mostrar en el visor LCD siguiendo el ejemplo en esta página. El diálogo LCD se usa dos veces, una vez por cada línea en el visor.

El diálogo necesita la dirección del primer carácter y el número de caracteres a ser exhibido. Por ejemplo, los dos diálogos mostrados en la próxima página crearían las dos instrucciones LCD de abajo. Cuando activadas, estas instrucciones harían que sean exhibidos los caracteres ASCII en V10000 -V10017. Los caracteres ASCII y sus direcciones correspondiente de memoria se muestran en la tabla de abajo.



5



V10000	a	T
V10001	q	n
V10002	e	u
V10003	h	
V10004	r	o
V10005	o	n
V10006	C	
V10007		
V10010	l	A
V10011	a	t
V10012	t	
V10013	m	e
V10014	e	p
V10015	a	r
V10016	u	t
V10017	a	r

T	a	n	q	u	e		h	o	r	n	o		C		
A	l	t	a		t	e	m	p	e	r	a	t	u	r	a

Instrucciones MODBUS RTU

La instrucción MODBUS Read from Network (MRX)

DS5	Usado
HPP	N/A

La instrucción MRX es usada por el maestro (master) de la red DL06 para leer un bloque de datos de un aparato conectado como esclavo y para escribir los datos en direcciones de memoria V dentro del maestro. La instrucción permite al usuario especificar si el puerto corresponde a la CPU o al módulo D0-DCM, el código de la Función de MODBUS, la dirección de la estación del esclavo, las direcciones de memoria de inicio del maestro y de esclavo, el número de elementos para transferir, formato de datos de MODBUS y la memoria intermedia (buffer) de Respuesta de Excepción.

5

- **CPU/DCM** : especifica si el maestro lee datos desde el puerto 2 o desde el módulo D0-DCM.
- **Slot Number** (Número de la ranura): debe ser la ranura del PLC donde está instalado el D0-DCM).
- **Port Number** (Número del puerto): debe ser el Puerto DL06 2 (K2).
- **Slave Address** (Dirección de esclavo): especifica una dirección del esclavo (0-247).
- **Function Code** (Código de Función): Los siguientes códigos de función MODBUS pueden ser manejados por la instrucción MRX:

- 01 – Lea un grupo de bobinas
- 02 – Lea un grupo de entradas
- 03 – Lea registros
- 04 – Lea registros de entradas
- 07 – Lea el estado de Excepción

- **Start Slave Memory Address** (Dirección inicial de memoria de esclavo) : especifica la dirección de memoria de esclavo de inicio de los datos a ser leídos. Vea la tabla en la página siguiente.
- **Start Master Memory Address** (Dirección de memoria inicial del maestro) : especifica la dirección de memoria de inicio en el maestro, donde los datos serán colocados. Vea la página siguiente.
- **Number of Elements** (Número de Elementos) : especifica cuántas bobinas, entradas, registros o registro de entradas se leerán. Vea la tabla en la página siguiente.
- **MODBUS Data Format** (Formatos de Datos de MODBUS): especifica el formato de datos MODBUS 584/984 o 484 a ser usado.
- **Exception Response Buffer** (Memoria Intermedia de Respuesta por Excepción): especifica la dirección de memoria del PLC maestro donde se colocará la Respuesta por Excepción. Vea la tabla en la página siguiente.

Rangos de direcciones de esclavo MRX

Código de función	Formato de datos MODBUS	Rango de direcciones de esclavo
01 – Lea bobina	Modo 484	1–999
01 – Lea bobina	Modo 584/984	1–65535
02 – Lea estado de entradas	Modo 484	1001–1999
02 – Lea estado de entradas	Modo 584/984	10001–19999 (5 dígitos) o 100001–165535 (6 dígitos)
03 – Lea Holding register	Modo 484	4001–4999
03 – Lea Holding register	Modo 584/984	40001–49999 (5 dígitos) o 4000001–465535 (6 dígitos)
04 – Lea Input register	Modo 484	3001–3999
04 – Lea Input register	Modo 584/984	30001–39999 (5 dígitos) o 3000001–365535 (6 dígitos)
07 – Lea estado	Modo 484 y 584/984	n/a

5

Rango de direcciones de memoria del maestro MRX	
Tipo de operando de datos	Rango del DL06
Entradas X	0–1777
Salidas Y	0–1777
Relevadores de control C	0–3777
Bits de etapas S	0–1777
Bits de temporizadores T	0–377
Bits de contadores CT	0–377
Relevadores especiales SP	0–777
Memoria V V	all
Entradas globales GX	0–3777
Salidas globales GY	0–3777

Número de elementos	
Tipo del operando de datos	Rango en el DL06
Memoria V V	Todas
Constante K	Bits: 1–2000 Registros: 1–125

Buffer de Exception Response	
Tipo de operando de datos	Rango del DL06
Memoria V V	Todas

Ejemplo de MRX

See page 5-208

La instrucción MODBUS Write a Network (MWX)

La instrucción MWX usa para escribir un bloque de datos de la memoria del maestro de la red del PLC DL06 a las direcciones de memoria de MODBUS dentro de un aparato esclavo en la red. La instrucción permite al usuario especificar si el puerto corresponde a la CPU o al módulo D0-DCM, el código de la función de MODBUS, la dirección de la estación esclavo, las direcciones de memoria inicial del maestro y esclavo, el número de elementos a transferir, el formato de datos de MODBUS y la Memoria Intermediaria (buffer) de Respuesta por Excepción.

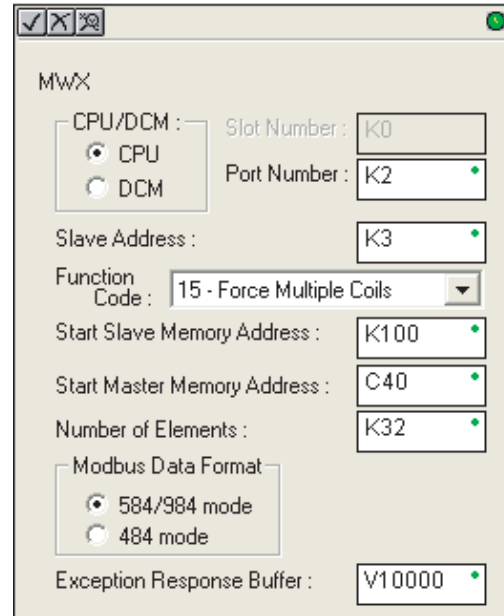
DS5	Usado
HPP	N/A

5

- **CPU/DCM** : especifica si el maestro lee datos desde el puerto 2 o desde el módulo D0-DCM.
- **Slot Number** (Número de la ranura): debe ser la ranura del PLC donde está instalado el módulo D0-DCM).
- **Port number** (número del puerto): Debe ser el Puerto 2 (K2) del PLC DL06 o del D0-DCM
- **Slave Address** (Dirección de esclavo): especifica una dirección de la estación esclavo (0-247).
- **Function Code** (Código de función): Los siguientes códigos de función MODBUS pueden ser manejados por la instrucción MRX:

- 05 – Fuerce una bobina
- 06 – Prefije un registro único
- 15 – Fuerce bobinas múltiples
- 16 – Prefije registros múltiples

- **Start Slave Memory Address** (Dirección de memoria inicial del esclavo) : Especifica la dirección de memoria de esclavo de inicio de los datos a ser escritos. Vea la tabla en la página siguiente.
- **Start Master Memory Address** (Dirección de memoria inicial del maestro) : especifica la dirección de memoria de inicio en el maestro, de donde los datos serán escritos. Vea la tabla en la página siguiente.
- **Number of elements** (Número de elementos) : Especifica cuántas bobinas, entradas, registros o registro de entradas se escribirán. Vea la tabla en la página siguiente.
- **MODBUS Data Format** (Formatos de datos de MODBUS): especifica el formato de datos MODBUS 584/984 o 484 a ser usado.
- **Exception Response Buffer** (Memoria Intermediaria de Respuesta por Excepción): especifica la dirección maestro de memoria donde se colocará la Respuesta por Excepción. Vea la tabla en la página siguiente.



Rangos de direcciones de esclavo MWX

Rangos de direcciones de memoria del maestro MWX	
Tipo de operando de datos	Rango del DL06
Entradas X	0-1777
Salidas Y	0-1777
Relevadores de control C	0-3777
Bits de etapas S	0-1777
Bits de temporizadores T	0-377
Bits de contadores CT	0-377
Relevadores especiales SP	0-777
Memoria V V	Todas
Entradas globales GX	0-3777
Salidas globales GY	0-3777

5

Rangos de direcciones de memoria del maestro MWX

Cantidad de elementos	
Tipo de operando de datos	Rango del DL06
Memoria V V	all
Constante K	Bits: 1-2000 Registros: 1-125

Número de elementos MWX

Número de elementos	
Tipo del operando de datos	Rango en el DL06
Memoria V V	Todas
Constante K	Bits: 1-2000 Registros: 1-125

Buffer de exception response MWX

Cantidad de elementos	
Tipo de operando de datos	Rango del DL06
Memoria V V	Todas

Ejemplo de MRX y de MWX

El puerto 2 del DL06 tiene dos contactos de relevadores especiales asociados con éste. Uno indica "Puerto ocupado" (SP116), y el otro indica "Error de comunicación de puerto" (SP117).

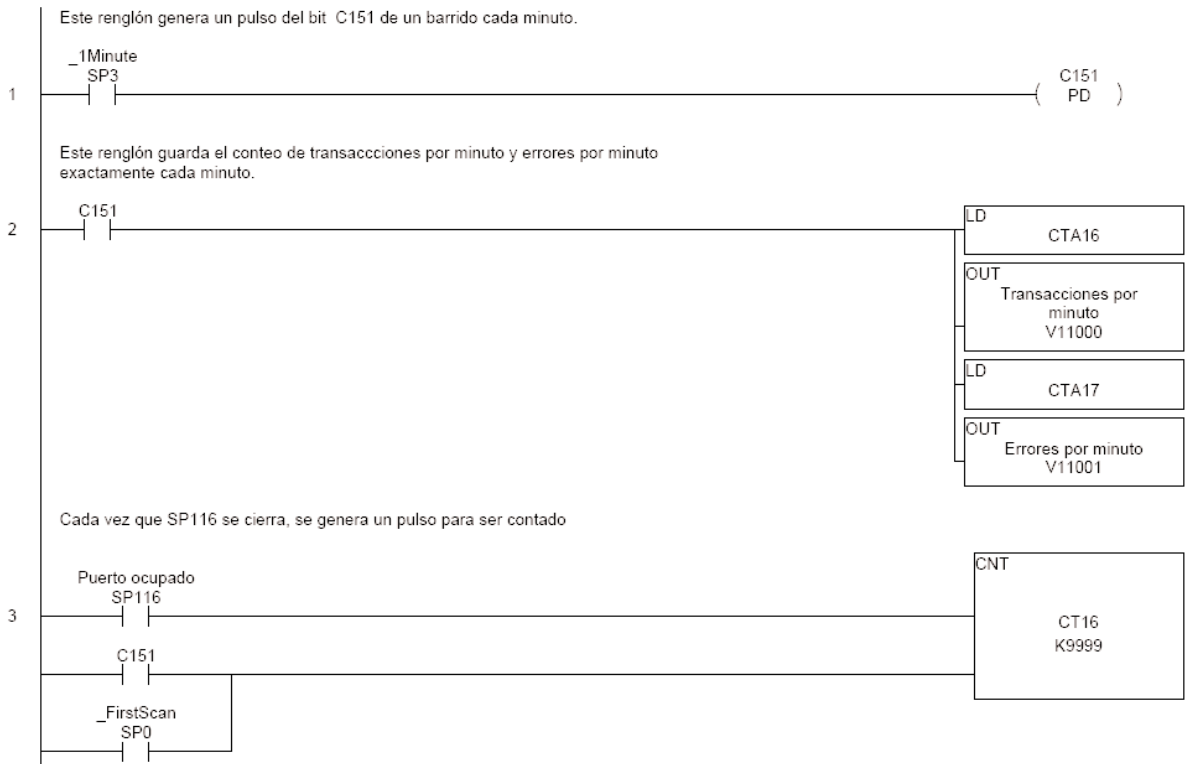
El bit de "puerto ocupado" está ON mientras el PLC se comunica con un esclavo. Cuando el bit está apagado el programa puede iniciar el próximo pedido a la red.

El bit de "Error de comunicación de puerto" se activa cuando el PLC ha detectado un error. El uso de este bit es opcional. Cuando se usa, debe estar adelante de cualquier instrucción de red ya que el bit de error es vuelto a 0 cuando se ejecuta una instrucción MRX o MWX.

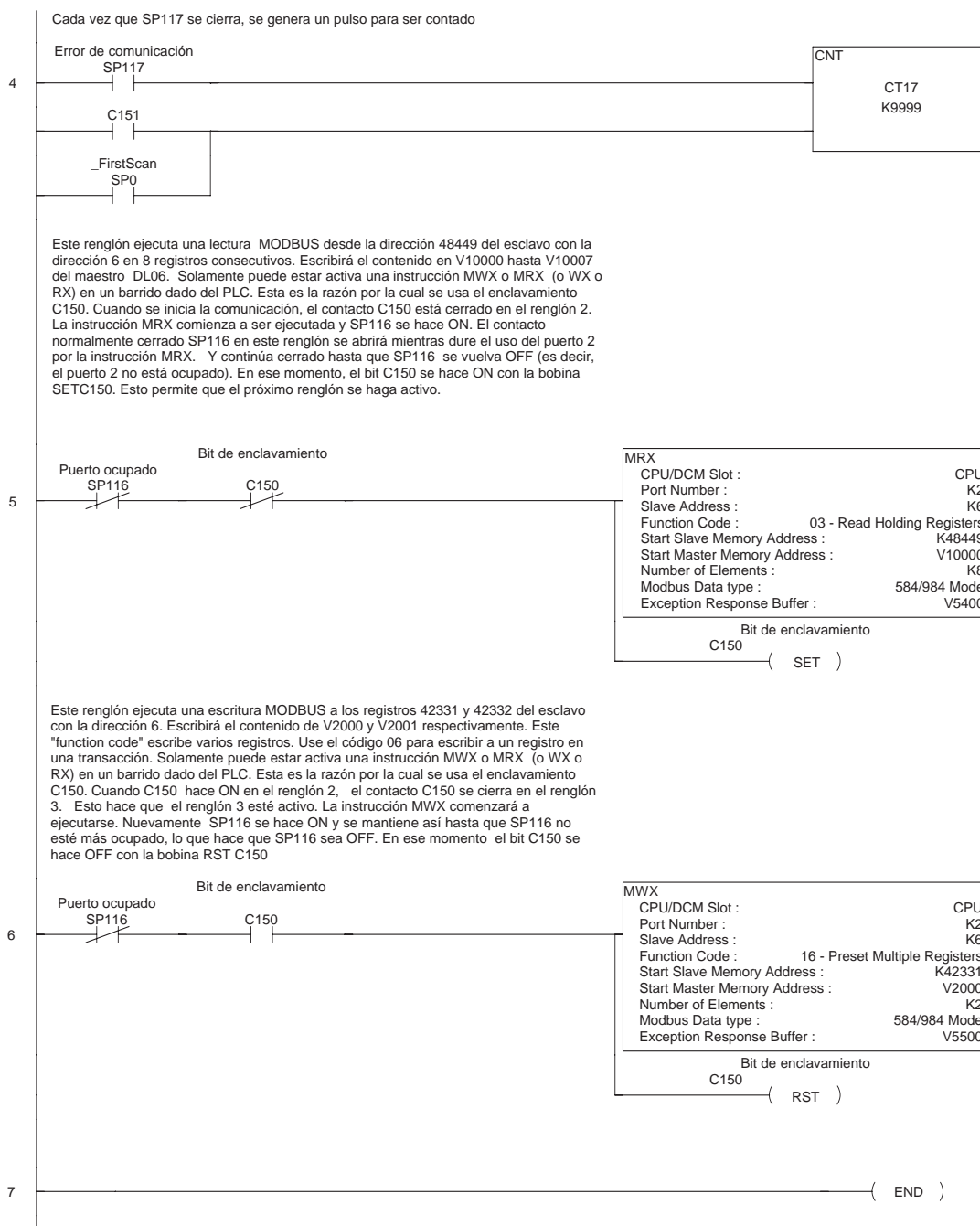
Típicamente las comunicaciones de red durarán más de 1 barrido. El programa debe esperar terminar la comunicación para comenzar la próxima transacción. Es por eso que se coloca un enclavamiento entre la lectura y la escritura con el relevador C150, en el ejemplo de la página 5-209.

El relevador interno C150 es usado para enclavar la lectura y la escritura. Vea por favor el ejemplo mostrado abajo para entender como funciona el enclavamiento.

En algunas aplicaciones hay ruido electromagnético que puede bajar la cantidad de transacciones y aparecen errores de comunicación. es perfectamente aceptable tener un 5% de errores en algunas aplicaciones. Los renglones 1 a 4 supervisan la cantidad de transacciones y errores por minuto. El renglón 2 guarda el conteo de cada minuto, lo que permite calcular el tiempo promedio que demora cada transacción.



Continúa en la próxima página....



En este ejemplo tratamos con una tasa de 9600 kbps. Hubo 1350 transacciones por minuto leídas en V11000; Cuando se aumentó a 38400 kbps, fueron 3535 transacciones por minuto. Por lo tanto, para una velocidad transmisión de 9600 kbps, cada transacción ocurre cada 44.4 milisegundos. Para 38400 kbps, cada, 19.98 ms.

Instrucciones ASCII

El PLC DL06 utiliza varias instrucciones y métodos que permiten leer y escribir texto ASCII a través del puerto 2 de comunicación o el módulo D0-DCM o aún desde el módulo coprocesador F0-CP128. El puerto 2 del DL06 puede ser usado para leer o escribir formatos ASCII pero no pueden ser usados ambos métodos al mismo tiempo en el mismo PLC, es decir, es necesario hacer una lógica para que una operación sea ejecutada en un período y luego la otra ejecutada en el próximo. Vea el apéndice G.

Vea una lista de transformación del código ASCII a decimal y a hexadecimal en el apéndice G.

El PLC DL06 también puede descifrar caracteres ASCII embutidos en uno de los protocolos aceptados (K-sequence, *DirectNet*, Modbus) en el puerto del PLC.

5

Leyendo Texto ASCII

Hay varios métodos que el PLC DL06 puede usar para leer ASCII.

- 1) Instrucción de lectura ASCII IN (AIN) – Esta instrucción se usa para recibir texto ASCII puro con parámetros tales como texto ASCII de longitud fija o variable, caracteres de terminación, opción de intercambio de bytes e instrucciones de control. Puede usar lectores de código de barras, balanzas, etc. para escribir texto ASCII al puerto 2 basados en los parámetros de la instrucción AIN.
- 2) Escriba texto ASCII embutido directamente a la memoria V desde una interfase hombre máquina o un aparato maestro similar usando un protocolo de los aceptados con el puerto 1 o 2 o el módulo D0-DCM. La instrucción AIN no se usa en este caso.
- 3) Si un PLC DL06 es maestro en una red, la instrucción RX (Network Read) puede ser usada para leer datos ASCII embutidos, desde un esclavo, usando protocolos aceptados con el puerto 2. La instrucción RX coloca los datos directamente en la memoria V.

Escribiendo Texto ASCII

Las siguientes instrucciones pueden ser usadas para escribir texto ASCII:

- 1) **Print from V-memory** (PRINTV) – Use esta instrucción para escribir texto ASCII puro, almacenado en la memoria del PLC, por el puerto 2 a un panel o a una impresora serial. La instrucción acepta la dirección inicial de memoria V, la longitud de la cadena, opción de cambio de bytes, etc. Cuando el bit de permiso está activado, la cadena de texto es escrita en el puerto 2.
- 2) **Print a V-memory** (VPRINT) – Use esta instrucción para crear cadenas ASCII en el PLC (Por ejemplo mensajes de alarma). Cuando el bit de permiso de la instrucción está activado, el mensaje es escrito en una localización de memoria pre-definida. Luego la instrucción PRINTV puede ser usada para escribir esta cadena ya definida saliendo por el puerto 2. Pueden ser usadas fechas de tipo americano, europeo o asiático.

Adicionalmente, si un PLC DL06 es maestro en una red, la instrucción WR (Network write) puede ser usada para escribir datos ASCII embutidos a una interfase hombre máquina o a un esclavo directamente desde la memoria V usando uno de los protocolos usados por el PLC con el puerto 2.

Administrando texto ASCII

Las siguientes instrucciones pueden ser útiles para programar las cadenas de texto ASCII en la memoria del PLC DL06:

- **ASCII Find (AFIND)** – Encuentra en que localización de memoria está contenida una porción específica de la cadena ASCII. Se permiten búsquedas para adelante y para atrás.
- **ASCII Extract (AEX)** – Extrae una porción específica de la cadena ASCII de una serie de memorias V y lo coloca en otra serie de memorias. (Típicamente algún valor).
- **Compare V-memory (CMPV)** – Esta instrucción es usada para comparar 2 bloques de memoria y es usada típicamente para detectar un cambio en una cadena ASCII. El tipo de formato debe ser igual en ambos datos (por ejemplo, BCD, ASCII, etc.)
- **Swap Bytes (SWAPB)** – Típicamente usada para intercambiar bytes de una memoria en datos ASCII que han sido escritos directamente a la memoria desde una interface hombre maquina o equipo maestro similar con un protocolo de comunicación. Las instrucciones AIN y AEX tienen la función de intercambio de bytes integrada.
- **ASCII to HEX (ATH)** – Convierte una tabla de valores ASCII a una tabla de valores hexadecimales.
- **HEX to ASCII (HTA)** – Convierte una tabla de valores hexadecimales a una tabla de valores ASCII.

Para poder usar las funciones de lectura y escritura ASCII, debe configurarse el puerto 2 con el protocolo Non-Sequence. Vea más detalles en el apéndice K de este manual.

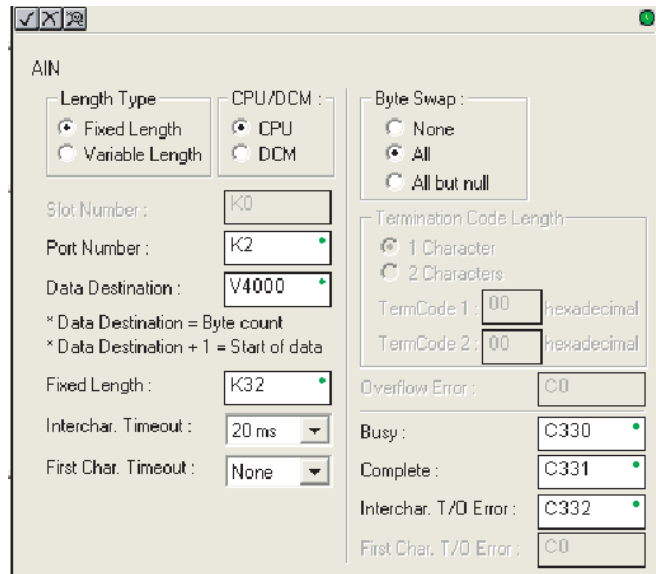
La instrucción ASCII Input (AIN)

DS5	Usado
HPP	N/A

La instrucción AIN permite recibir cadenas ASCII a través del puerto 2 y coloca la cadena en una serie de memorias (Una tabla). Los datos ASCII pueden ser recibidos como un número fijo de bytes o como una cadena variable con un o más caracteres de terminación especificados. Otras características incluyen, preferencia de intercambio de bytes, tiempo de llegada excedido, (Timeout) y bits de alarma libremente definidos para Ocupado (Busy), Completo (Complete) y error de tiempo (Timeout error).

Configuración AIN de longitud fija

- **Length Type:** Seleccione Fixed Length de acuerdo a la longitud de la cadena ASCII que será enviada al puerto del PLC.
- **CPU/DCM :** especifica si el maestro lee datos desde el puerto 2 o desde el módulo D0-DCM.
- **Port Number:** Siempre puerto No. 2 (K2)
- **Data Destination:** Especifique donde en la memoria será colocada la cadena ASCII.
- **Fixed Length:** Especifique la longitud, en bytes, de la cadena ASCII que será recibida por el puerto.
- **Inter-character Timeout:** si el periodo de los caracteres ASCII entrantes excede el tiempo ajustado, será activado. el error de timeout. No será almacenada ninguna información en la localización de memoria de destino. El bit será desactivado cuando el renglón donde está la instrucción está desactivado. Un valor de 0 ms elimina esta función.
- **First Character Timeout:** Si el período transcurrido entre cuando AIN está activado hasta el tiempo que el primer carácter es recibido es mayor que el valor prefijado, será activado el bit de First Carácter Timeout. Este bit será desactivado cuando la instrucción AIN sea desactivada. Un valor de 0 ms elimina esta función.
- **Byte Swap:** Intercambia el byte más alto con el más bajo en cada memoria V de la cadena ASCII de longitud fija. Vea la instrucción SWAPB para más detalles.
- **Busy Bit:** Este está ON cuando la instrucción AIN está recibiendo datos ASCII.
- **Complete Bit:** Se coloca ON cuando los datos ASCII han sido recibidos con la longitud especificada y OFF cuando los bits que hacen la instrucción AIN activa están desactivados.
- **Inter-character Timeout Error bit:** Se coloca ON cuando el Character Timeout ha sido sobrepasado. Vea la explicación de Character Timeout arriba.
- **First Character Timeout Error bit:** Se coloca ON cuando el Character Timeout ha sido sobrepasado. Vea la explicación de Character Timeout arriba.

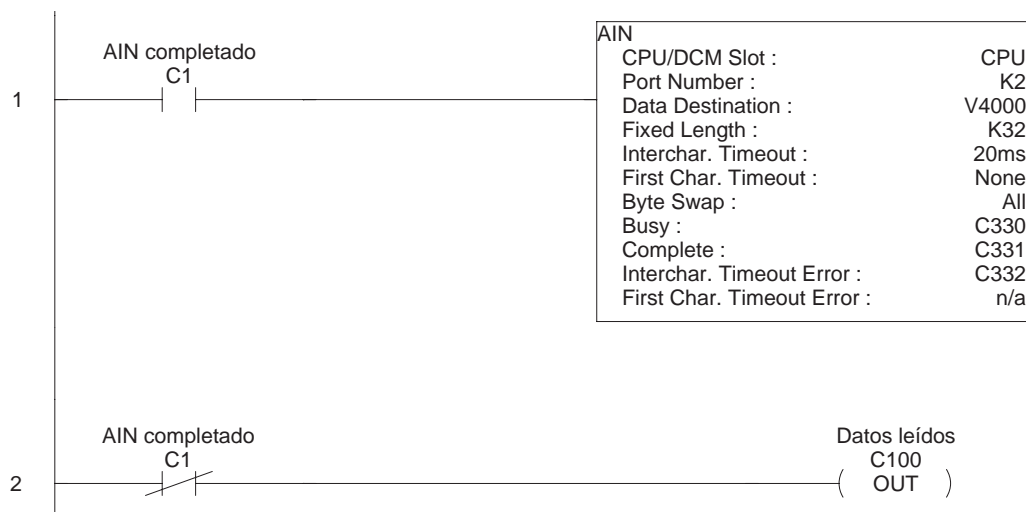


Antes de cada lectura, debe hacerse un reset de la instrucción AIN. Puede hacerse esta acción desactivando y activando el renglón donde está AIN o también con la instrucción ACBR (vea la página 5-228),

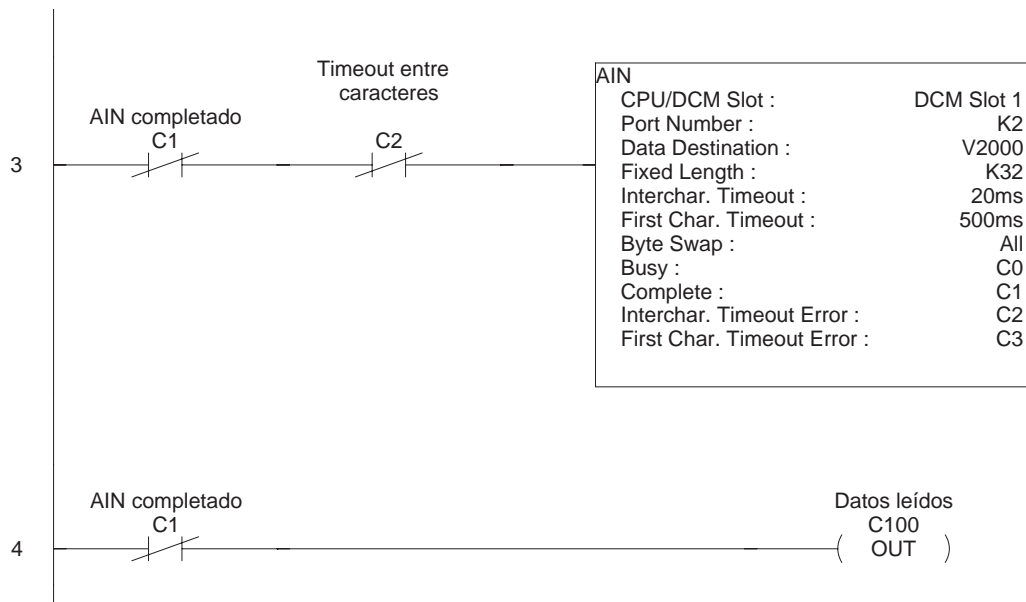
Parámetro	
Destino de los datos	Todas las memorias V
Longitud fija	K1-128
Bits: Busy, Complete, Timeout Error, Overflow	C0-3777

Ejemplos de longitud fija de AIN

Ejemplo de longitud fija cuando el PLC está leyendo el puerto continuamente y el tiempo de operación no es crítico.

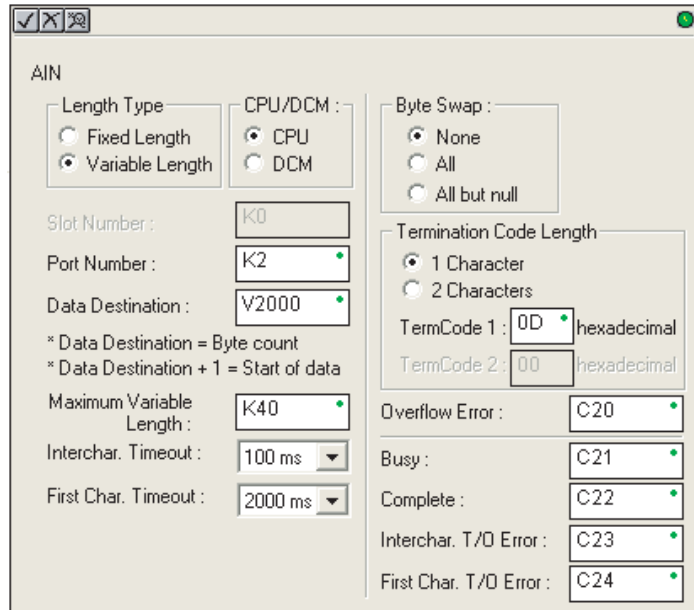


Ejemplo de longitud fija cuando el tiempo de transmisión carácter a carácter es crítico.



Configuración AIN de longitud variable:

- **Length Type:** Seleccione Variable Length de acuerdo a la longitud de la cadena ASCII que será enviada al puerto del PLC.
- **CPU/DCM :** especifica si el maestro lee datos desde el puerto 2 o desde el módulo D0-DCM.
- **Port Number:** Siempre use el puerto No. 2 (K2)
- **Data Destination:** Especifique donde será colocada la cadena ASCII en la memoria.
- **Maximum Variable Length:** Especifique, en bytes, la máxima longitud de la cadena ASCII que será recibida por el puerto.



- **Inter-character Timeout:** si el período de los caracteres ASCII entrantes excede el tiempo ajustado, será activado el error de Timeout. No será almacenada ninguna información en la localización de memoria de destino. El bit será desactivado cuando el renglón donde está la instrucción está desactivado. Un valor de 0 ms elimina esta función.
- **First Character Timeout:** Si el período transcurrido entre cuando AIN está activado hasta el tiempo que el primer carácter es recibido es mayor que el valor prefijado, el bit de First Carácter Timeout será activado. Este bit será desactivado cuando la instrucción AIN sea desactivada. Un valor de 0 ms elimina esta función.
- **Byte Swap:** Intercambia el byte más alto con el más bajo en cada memoria de la cadena ASCII de longitud variable. Vea la instrucción SWAPB para los detalles.
- **Termination Code Length:** consiste de uno o dos caracteres como valor hexadecimal. Vea la tabla ASCII en el apéndice G.
- **Overflow Error Bit:** ES colocado ON cuando los datos ASCII recibidos exceden la longitud variable máxima especificada.
- **Busy Bit:** Este bit está ON cuando la instrucción AIN está recibiendo datos ASCII.
- **Complete Bit:** Se coloca ON cuando los datos ASCII han sido recibidos hasta que se termina de recibir los caracteres de terminación y OFF cuando los bits que hacen real la instrucción AIN están desactivados.
- **Inter-character Timeout Error Bit:** Se coloca ON cuando el Character Timeout ha sido sobrepasado. Vea la explicación de Character Timeout arriba.
- **First Character Timeout Error Bit:** Se coloca ON cuando el Character Timeout ha sido sobrepasado. Vea la explicación de Character Timeout arriba.

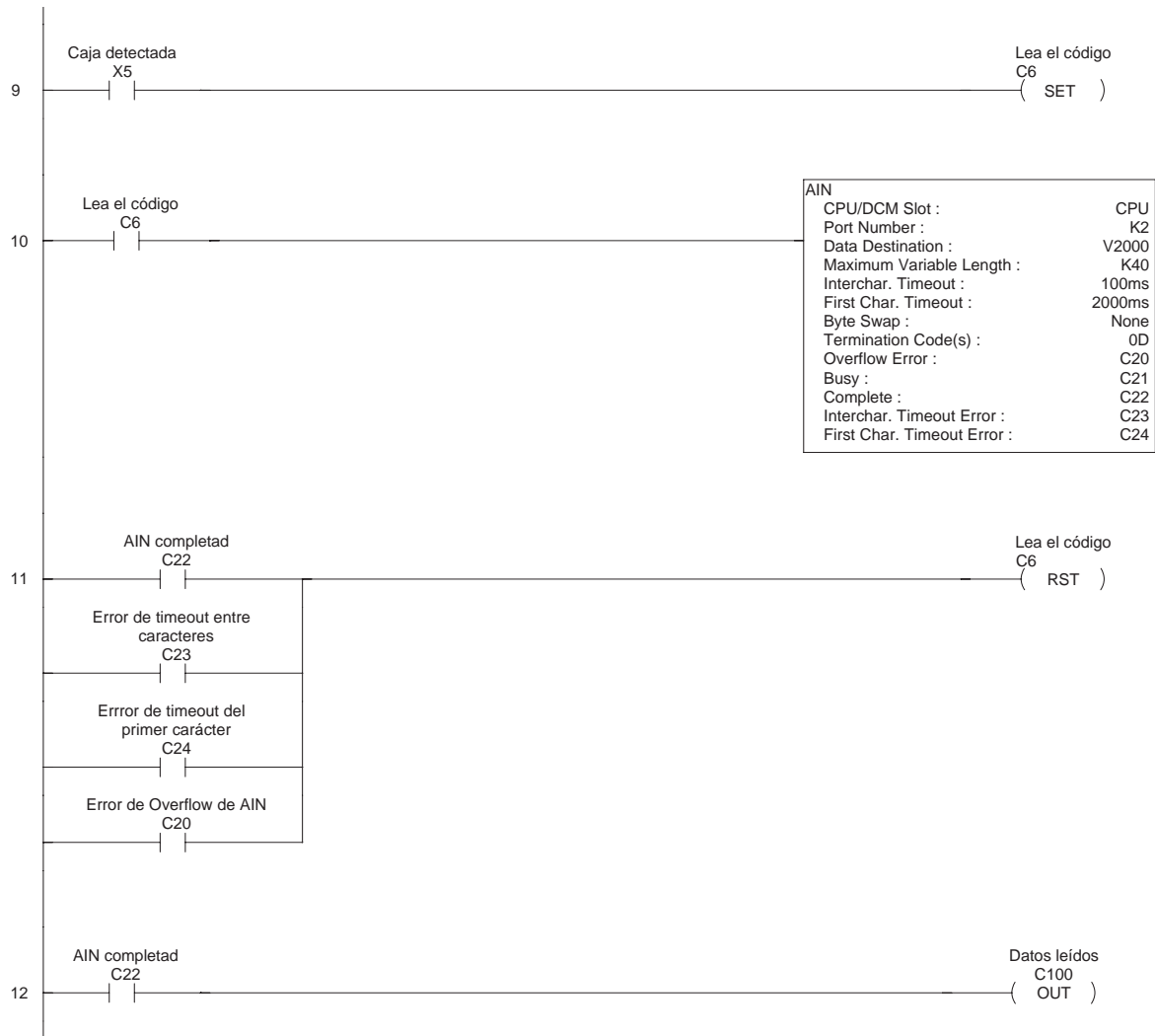
Parámetro	
Destino de los datos	Todas las memorias V
Longitud variable	K1-128
Bits: Busy, Complete, Timeout Error, Overflow	C0-3777

Ejemplo de longitud variable con AIN

Ejemplo de AIN con longitud variable usado para leer códigos de barras en cajas

En este ejemplo se tiene un lector de código de barras que lee el código de una caja de cartón sobre una correa transportadora detectado por un sensor photoelectrico conectado a la entrada X5. Los datos son colocados en formato ASCII en el conjunto de 40 registros consecutivos que comienzan en V2000.

5



La instrucción ASCII Find (AFIND)

DS5	Usado
HPP	N/A

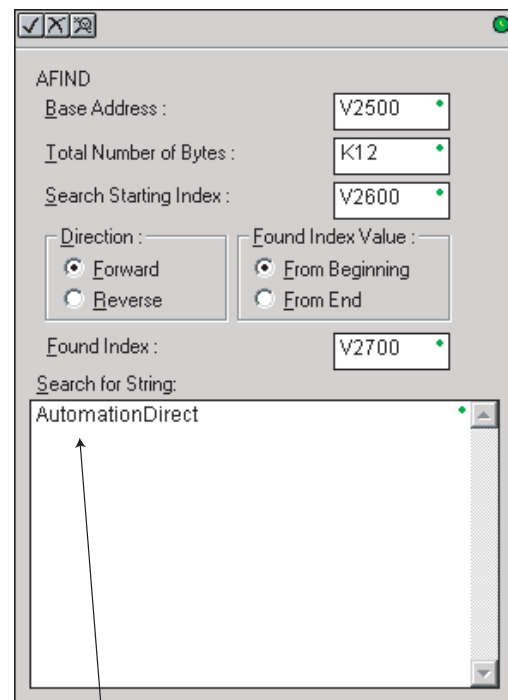
La instrucción AFIND localiza una cadena ASCII específica o una porción de ella en un rango de registros y coloca un número en una memoria especificada, número que corresponde al número del byte donde se encontró la cadena deseada. Este número se llama Found Index.

Otras características incluyen número Search Starting Index (Índice de inicio de búsqueda) para saltar los bytes que no son necesarios antes de iniciar la operación de búsqueda, búsqueda en la dirección Forward o Reverse, es decir, para adelante o para atrás y por último la selección de cual es el lado de donde se comienza a contar el valor Found Index. (From Beginning o From End)

Aquí está la definición de cada uno de los parámetros:

- **Base Address** (Dirección base): Define el comienzo de una tabla de memorias donde está almacenada la cadena ASCII.
- **Total Number of Bytes** (Cantidad de bytes): Define el número total de bytes que serán vistos para encontrar el texto ASCII deseado. El número máximo es 128 bytes.
- **Search Starting Index** (Índice de búsqueda): Define hasta que byte se va a saltar (con respecto a la dirección base) antes de iniciar la búsqueda.
- **Direction - Forward**: es la dirección de inicio para buscar la cadena desde memorias de localización baja para memorias de localización alta. **Reverse** es la dirección de inicio para buscar la cadena desde memorias de localización alta para memorias de localización baja.
- **Found Index Value** (# de índice encontrado): Define si el byte inicial o final de la cadena ASCII será cargado en la memoria Found Index.
- **Found Index**: Define la dirección de memoria donde será almacenado el Found Index number. Si el texto deseado no es encontrado el contenido de esta memoria será FFFF.
- **Search for String**: La cadena a ser buscada, con hasta 128 caracteres.

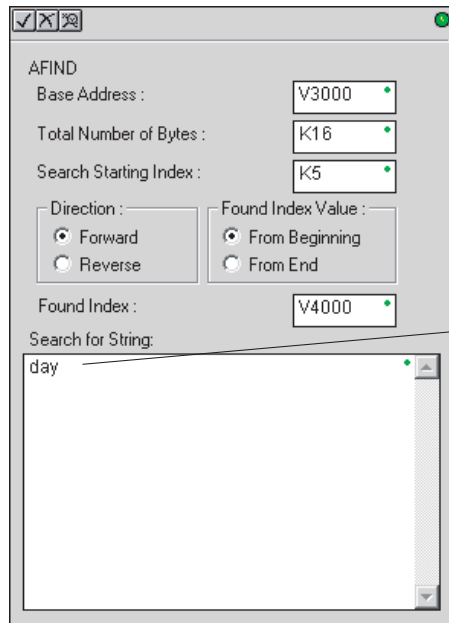
Parámetro	Rango del DL06
Dirección base	Toda la memoria V
Número total de bytes	Toda la memoria V o K1-128
Search Starting Index	Toda la memoria V o K0-127
Found Index	Toda la memoria V



NOTA: No se necesitan comillas en el elemento Search for String. Las comillas son caracteres válidos que AFIND puede buscar.

Ejemplo de búsqueda con AFIND

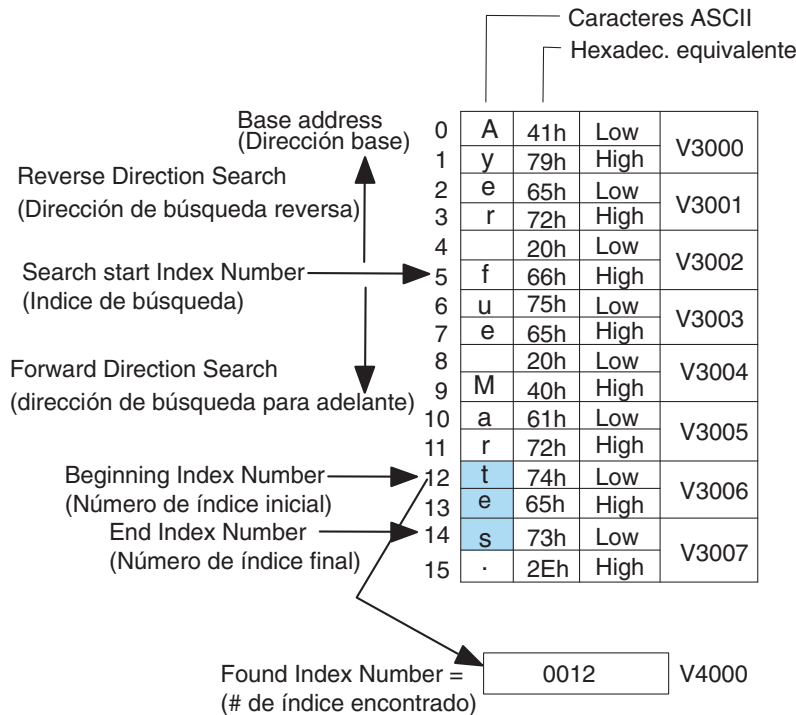
En el ejemplo siguiente, se usa la instrucción AFIND para buscar la porción "tes" en la palabra "Martes" en el texto ASCII "Ayer fue Martes", que ha sido colocado en una tabla de memorias. Note que el valor Search Starting Index (K)3 combinado con un Forward Direction Search es usado para prevenir que se encuentre esta secuencia de texto antes del tercer caracter. El Found Index number será colocado en V4000.



Quizás necesite hacer un "byte swap" en la cadena ASCII buscada, dependiendo de como fueron colocados los datos ASCII.

5

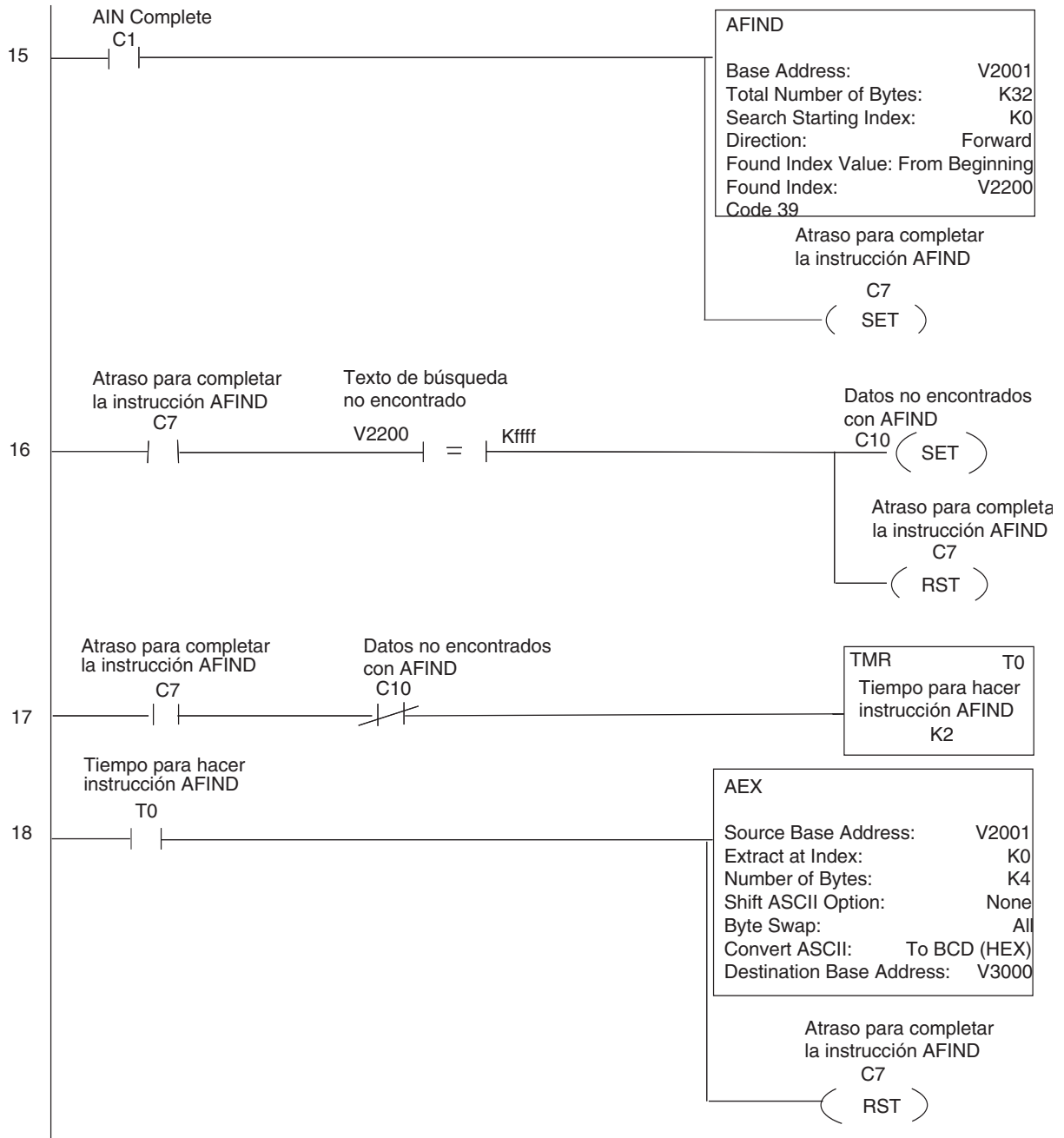
Note que no se colocan comillas alrededor del texto de búsqueda (Search String). Use comillas solamente si son realmente parte del texto de búsqueda (Search for String)



Ejemplo de instrucción AFIND combinado con instrucción AEX

Se puede usar el bit Complete de una instrucción AIN para activar una instrucción AFIND para encontrar un conjunto de texto ASCII. Cuando éste sea encontrado, la instrucción AEX puede usarse para extraer el texto localizado. Vea el ejemplo a continuación.

5



La instrucción ASCII Extract (AEX)

DS5	Usado
HPP	N/A

La instrucción ASCII Extract (AEX) extrae un número especificado de bytes de datos ASCII de una tabla de memoria y la coloca en otra tabla.

Otras características incluyen

Extract at Index que es iniciar la extracción a un número prefijado de bytes para saltar bytes no necesarios antes de comenzar la operación de extracción.

Shift ASCII option, que permite desplazar los datos extraídos para conveniencia

Byte Swap que es intercambio de bytes en una palabra

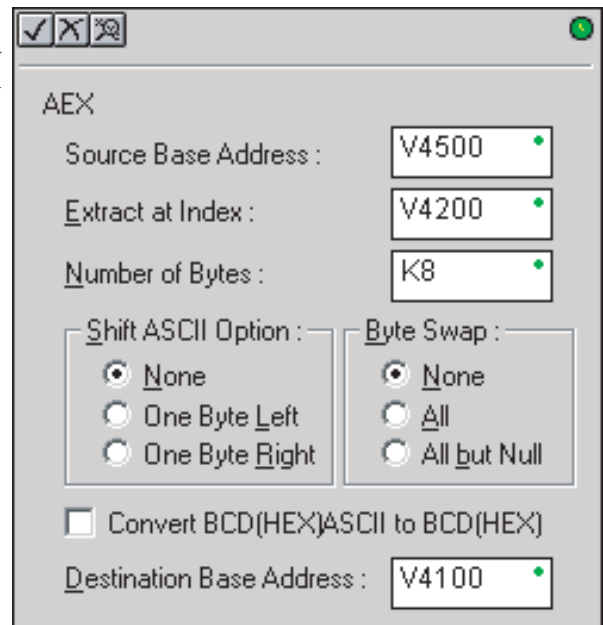
Convertir datos a un número BCD.

Aquí está la definición de cada uno de los parámetros:

- **Source Base Address** (Dirección de la tabla fuente): Define el comienzo de la tabla de memorias donde la cadena ASCII está almacenada.
- **Extract at Index**: Define a que byte va a saltar (siendo el inicio el Source base Address) antes de extraer los datos.
- **Number of Bytes**: define el número de bytes a ser extraído.
- **Shift ASCII Option** : Desplaza todos los datos extraídos un byte a la izquierda o a la derecha para remover caracteres no deseados, si fuera necesario.
- **Byte Swap**: Intercambia el byte más alto con el más bajo en cada palabra de memoria de los datos extraídos. Vea la instrucción SWAPB para más detalles.
- **Convert BCD(Hex) ASCII a BCD (Hex)**: Esta selección permite convertir datos numéricos ASCII a números Hexadecimal.
- **Destination Base Address**: Define la dirección de memoria donde serán almacenados los datos extraídos.

Vea un ejemplo en la página anterior.

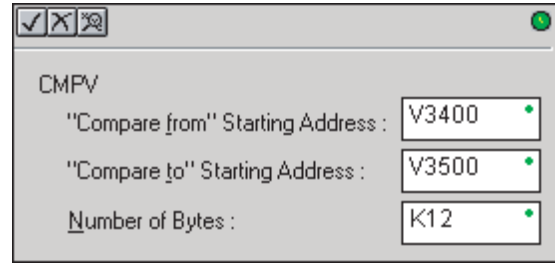
Parámetro	Rango del DL06	
Dirección fuente	Dirección fuente	
Extraiga al "Index"	Extraiga al "Index"	
Número de bytes Cuando "convert BCD(Hex) ASCII" no esté marcado.	Rango constante: K1-128	Memoria V que contiene el valor BCD: 1-128
Number of Bytes Cuando "convert BCD(Hex) ASCII" está marcado	Rango constante: K1-4	Memoria V que contiene el valor BCD: 1-4
Dirección de destino	Toda la memoria V	



La instrucción ASCII Compare (CMPV)

DS5	Usado
HPP	N/A

La instrucción CMPV compara dos tablas de memoria. Esta instrucción compara cualquier tipo de datos (ASCII a ASCII< BCD a BCD, etc.) de una tabla de memorias a otra tabla de memorias por una longitud de bytes definida. Es resultado de la comparación es dado por el relevador especial SP61.



Aquí está la definición de cada uno de los parámetros:

- **“Compare from” Starting Address:** Esta dirección define el comienzo de la tabla de memorias con un texto ASCII a la cual será comparada la segunda tabla de memorias.
- **“Compare to” Starting Address:** Esta dirección define el comienzo de la segunda tabla de memorias donde está la cadena ASCII.
- **Number of Bytes:** Número define la longitud de cada tabla de memoria a ser comparada.

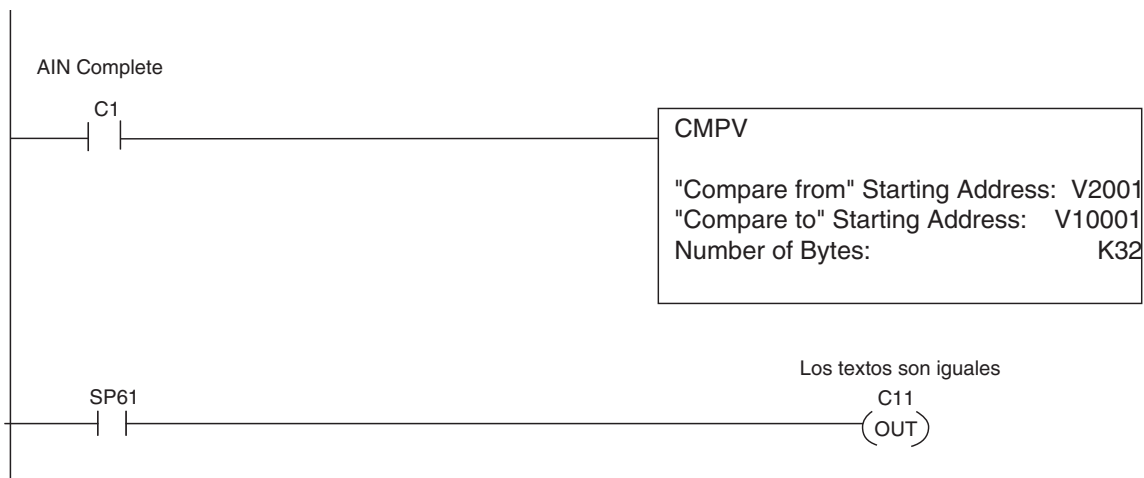
Si SP61 es 1, el resultado de la comparación es igual.

Si SP61 es 0, el resultado de la comparación no es igual.

Parámetro	Rango del DL06
Compare desde la dirección inicial	Toda la memoria V
Compare a la dirección inicial	Toda la memoria V
Número de bytes	K0-127

Ejemplo de CMPV

La instrucción CMPV se ejecuta cuando la instrucción AIN ha sido completada. Si el resultado de la comparación es igual, SP61 se hará ON y C11 será activado.



La instrucción ASCII Print a V-memory (VPRINT)

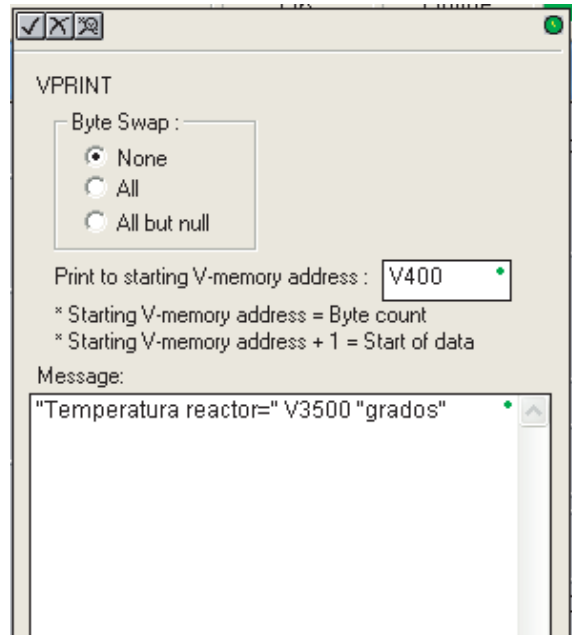
DS5	Usado
HPP	N/A

La instrucción VPRINT escribirá una cadena ASCII definida en uno de los campos de la instrucción en una tabla de memorias.

Esta instrucción puede intercambiar bytes, puede suprimir ceros a la izquierda, convertir espacios a ceros, usar fecha en formato EE.UU., europeo o asiático y horas en formato de 12 horas o de 24 horas.

Aquí está la definición de cada uno de los parámetros:

- **Byte Swap:** Este parámetro intercambia el byte más alto con el más bajo en cada palabra de la memoria donde la cadena ASCII es escrita, si es seleccionado ALL; para no hacer este intercambio seleccione None.
- **Print a Starting V-memory Address:** Este parámetro define el comienzo de la tabla de memorias donde será colocada la cadena ASCII por la instrucción VPRINT. La primera memoria de la tabla contendrá la longitud en bytes de la cadena ASCII. La segunda memoria y las que siguen contendrán los datos ASCII de la cadena impresa en la tabla.
- **Message:** Aquí en "mensaje" se escribe el texto a ser almacenado en la tabla de memorias, entre comillas (""); Acepta contenidos numéricos de memorias o fechas y hora.



5



NOTA: Starting V-memory Address (Dirección de memoria inicial) es la primera memoria V de la serie de memorias especificadas que contendrá la longitud de la secuencia de texto ASCII en bytes. La memoria V + 1 y las memorias subsecuentes contendrán la secuencia de texto ASCII que se imprime a la memoria V.

Parámetro	Rango del DL06
Imprima a la dirección inicial de memoria V	Todo el rango de memoria V

Colocación de fecha y hora con VPRINT– Pueden ser usados los códigos mostrados en la tabla de abajo en la secuencia de texto con VPRINT ASCII para “imprimir a la memoria” la fecha y hora corriente.

#	Character code	Date / Time Stamp Options
1	_date:us	Norma americana (Mes/día/año con 2 dígitos)
2	_date:e	Norma europea (día/mes/año con 2 dígitos)
3	_date:a	Norma asiática (año con 2 dígitos/mes. Día)
4	_time:12	Norma de 12 horas (0-12: minutos AM/PM)
5	_time:24	Norma de 24 horas (0-24: minutos)

Modificadores de números contenidos en memoria – Los siguientes modificadores de números pueden ser usados en un mensaje VPRINT para almacenar el número en formato entero o real. Puede usar el número contenido en una memoria V sin modificador o con el modificador de tipo de datos después de ":". Los tipos de datos son mostrados en la tabla abajo: El código debe ser escrito con mayúsculas.



NOTA: Debe colocar un espacio antes y después de la dirección de memoria V para separarla de la cadena de texto. Si no hace ésto aparece el error 499.

#	Character code	Descripción
1	none	Binario de 16 bits (Número decimal)
2	: B	BCD de 4 dígitos
3	: D	Binario de 32 bits (Número decimal)
4	: D B	BCD de 8 dígitos
5	: R	Número de punto flotante (Número real)
6	: E	Número de punto flotante (Número real con exponente)

Ejemplos:

V2000 imprime datos binarios en V2000 como decimal

V2000 : B imprime datos BCD en V2000

V2000 : D imprime datos binarios en V2000 y V2001 como decimal

V2000 : D B imprime datos en V2000 y V2001

V2000 : R imprime datos de punto flotante en V2000 y V2001 como número real

V2000 : E imprime datos de punto flotante en V2000/V2001 como número real con exponente

Los siguientes modificadores pueden ser agregados a los anteriores para suprimir o convertir ceros a la izquierda o espacios. El código debe ser escrito con mayúsculas.

#	Código de caracteres	Descripción
1	S	Elimina espacios a la izquierda
2	C0	Convierte espacios a la izquierda a ceros
3	0	Elimina ceros a la izquierda

Ejemplo con V2000 = 0018 (Formato binario)

Memoria con modificador	Número de caracteres			
	1	2	3	4
V2000	0	0	1	8
V2000:B	0	0	1	2
V2000:BO	1	2		

Ejemplo con V2000 = sp sp 0018 (Formato binario) donde "sp" significa espacio

Memoria con modificador	Número de caracteres			
	1	2	3	4
V2000	sp	sp	1	8
V2000:B	sp	sp	1	2
V2000:BS	1	2		
V2000:BC0	0	0	1	2

Modificadores de texto contenido en memoria – El siguiente modificador de longitud de texto puede ser usado en un mensaje VPRINT para almacenar el texto a partir de la primera o consecutivas direcciones de memoria. Use el signo "%" seguido del número de caracteres en la cadena que Ud. desea imprimir. Si usa "0" como número de caracteres, la instrucción leerá la cantidad de caracteres desde la primera dirección. Luego comenzará en la próxima dirección y leerá la cantidad leída de caracteres ASCII indicada allí desde la memoria indicada.

Ejemplo:

V2000 % 16 Se transfieren 16 caracteres en V2000 hasta V2007

V2000 % 0 Se transfieren XX caracteres a partir de V2001 (XX es determinado por el número almacenado en V2000).

Modificadores de bits contenidos en memoria– Los siguientes modificadores de estados de bits en una memoria o en un relevador de control pueden ser usados en un mensaje VPRINT para almacenar un cierto formato. Puede usarse la dirección del relevador o la memoria seguida

#	Formato de datos	Descripción
1	Sin modificador	Imprime un "1" para el estado ON; "0" para OFF
2	: BOOL	Imprime "TRUE" para el estado ON; "FALSE" para OFF
3	: ONOFF	Imprime "ON" para el estado ON; "OFF" para OFF

de "." Y el número de bit con el modificador de tipo de datos después de ":". Los tipos de datos son mostrados en la tabla abajo: El código debe ser escrito con mayúsculas.

Ejemplos:

V2000 . 15 imprime el estado del bit 15 en V2000 en el formato 1 o 0.

C100 imprime el estado de C100 en el formato 1 o 0.

C100 : BOOL imprime el estado de C100 en el formato TRUE o FALSE

C100 : ON/OFF imprime el estado de C100 en el formato ON u OFF

V2000.15 : imprime el estado del bit 15 en V2000 en el formato TRUE o FALSE.

El máximo número de caracteres que se puede operar con VPRINT es 128. En la lista a continuación es mostrado el número requerido por cada elemento, sin importar si se usa o no los modificadores :S, :C0 o :0.

Tipo de elemento	Cantidad máxima de caracteres
Texto, 1 carácter	1
Binario de 16 bit	6
Binario de 32 bit	11
BCD de 4 dígitos	4
BCD de 8 dígitos	8
Punto flotante (Número real)	3
Punto flotante (real con exponente)	13
Texto en memoria V	2
Bit (formato 1/0)	1
Bit (formato TRUE/FALSE)	5
Bit (formato ON/OFF)	3

Modificadores de caracteres especiales – Los siguientes modificadores pueden ser usados en un mensaje VPRINT para almacenar caracteres especiales. Caracteres en una cadena ASCII son definidos como los caracteres contenidos entre comillas (") en el campo de mensaje en VPRINT. Dos números hexadecimales que sean precedidos por el signo \$ significa un código de caracteres ASCII de 8 bits. También, dos caracteres precedidos por el signo \$ es interpretado de acuerdo a la siguiente tabla:

#	Código de carácter	Descripción
1	\$\$	Signo dólar (\$)
2	\$"	Comillas (")
3	\$L o \$l	Line feed (LF) usado por ejemplo con impresoras
4	\$N o \$n	Carriage return line feed (CRLF)
5	\$P o \$p	Form feed
6	\$R o \$r	Carriage return (CR)
7	\$T o \$t	Tab

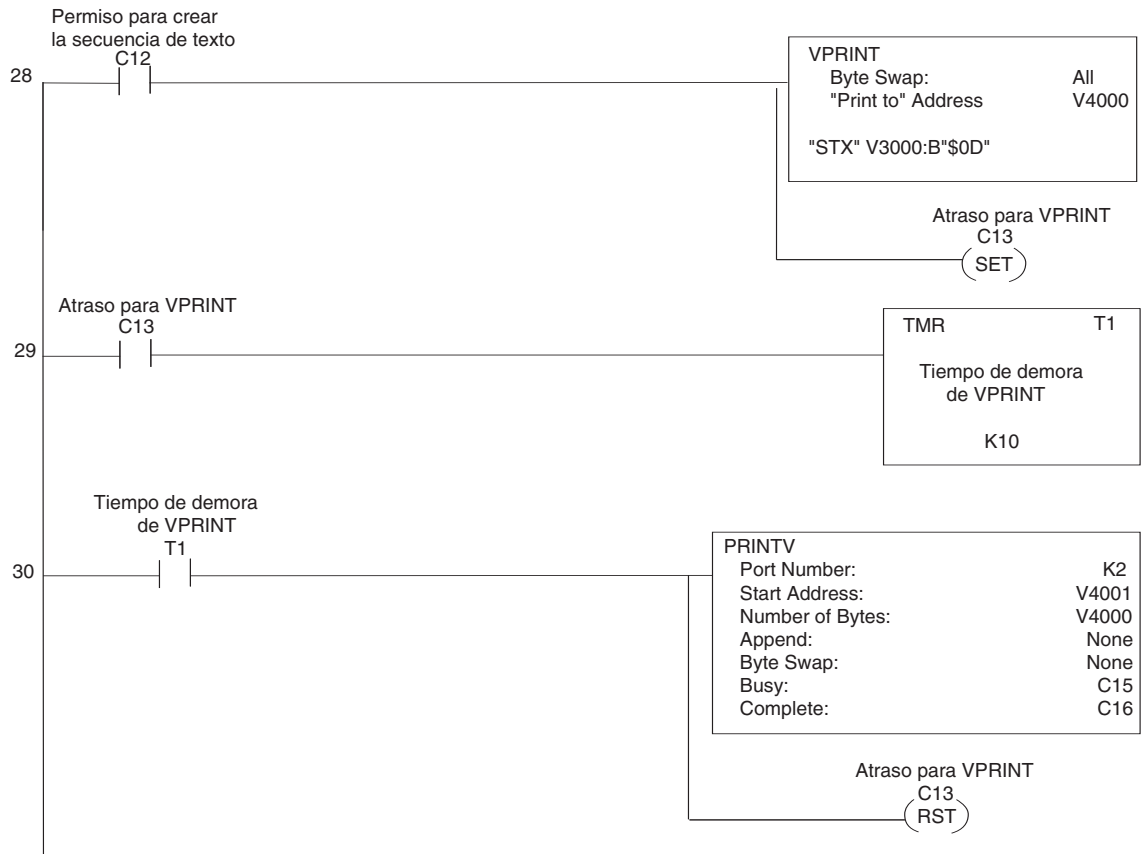
Los siguientes ejemplos muestran varias convenciones de sintaxis y la longitud de la salida a la impresora.

" "	Longitud 0 sin ningún carácter
"A"	Longitud 1 con carácter A
" "	Longitud 1 con espacio en blanco
"\$"	Longitud 1 con comillas
"\$R\$L"	Longitud 2 con un CR y un LF
"\$O D \$O A"	Longitud 2 con un CR y un LF
"\$\$"	Longitud 1 con un carácter \$

Al imprimir una línea de texto, Ud. deberá incluir comillas antes y después de la cadena de texto. Aparecerá el error 499 en la CPU si la instrucción contiene texto inválido o no contiene comillas. Es importante probar los datos de la instrucción VPRINT durante el desarrollo del programa.

Ejemplo de VPRINT combinado con la instrucción PRINT V

Se usa aquí la instrucción VPRINT para crear una cadena de caracteres en la memoria V que se inicia en V4000. Luego se usa la instrucción PRINTV para generar una salida de caracteres ASCII por el puerto 2 del PLC.



5

La instrucción ASCII Print from V-memory (PRINTV)

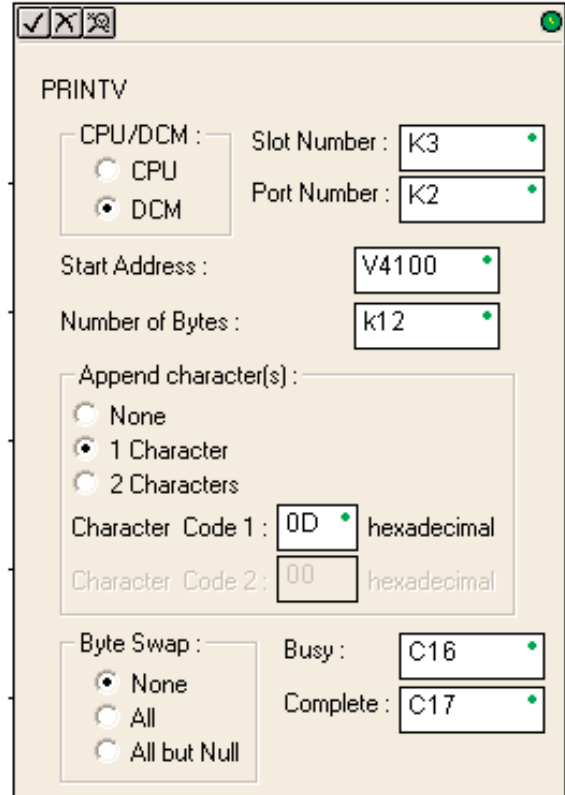
DS5	Usado
HPP	N/A

La instrucción PRINTV enviará un texto ASCII de longitud determinada definida en una tabla de memorias saliendo por el puerto 2 de la CPU o del módulo D0-DCM.

Esta instrucción puede agregar caracteres definidos por el usuario después de un texto de datos para aparatos que requieren caracteres específicos de terminación, puede intercambiar bytes y usar indicaciones definidas por el usuario para los estados Busy y Complete.

Aquí está la definición de cada uno de los parámetros:

- **CPU/DCM** : especifica si el maestro lee los datos desde el puerto 2 o desde el módulo D0-DCM.
- **Port Number**: Para el PLC DL06 debe ser siempre puerto 2 (K2)
- **Start Address** (Dirección inicial) : define el comienzo de una tabla que contiene la cadena ASCII a ser transferida.
- **Number of Bytes** (cantidad de bytes): define la longitud de la cadena a ser transferida.
- **Append Characters**: Define los caracteres ASCII a ser agregados al final de la cadena para aparatos que necesiten caracteres de terminación. Debe ser escrito en hexadecimal.
- **Byte Swap**: Este parámetro intercambia el byte más alto con el más bajo en cada palabra de la memoria donde la cadena ASCII mientras imprime. Vea la instrucción SWAPB para más detalles.
- **Busy Bit**: Este bit, a ser definido por el usuario, cambia a ON mientras la instrucción está imprimiendo datos ASCII. Muestra que la CPU está ocupada.
- **Complete Bit**: Este bit, a ser definido por el usuario, cambia a ON cuando la instrucción terminó de imprimir y es OFF cuando los bits de permiso de ejecución de la instrucción PRINTV están desactivados (Es decir, el renglón es falso).



Vea el ejemplo de la página anterior para ver el uso de esta instrucción.

Parámetro	Rango en el DL06
CPU/DCM	Ranura 1 a 4
Port number (número de puerto)	Puerto 2 (K2)
Start Address (Dirección inicial)	Todas las memorias V
Number of bytes (Cantidad de Bytes)	Toda la memoria o k1-128
Bits: Busy, Complete	C0-3777

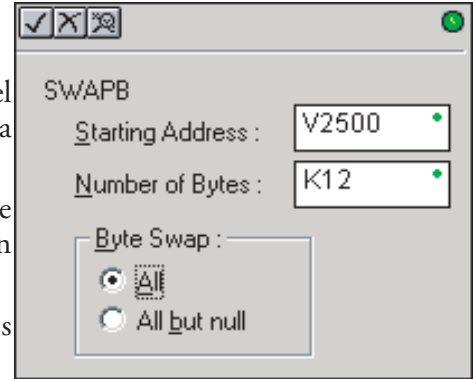
La instrucción ASCII Swap Bytes (SWAPB)

DS5	Usado
HPP	N/A

La instrucción SWAPB intercambia posiciones de bytes (del byte más alto al más bajo y viceversa) en cada memoria de la tabla que contiene una cadena ASCII (o un conjunto de datos no necesariamente ASCII)

Aquí está la definición de cada uno de los parámetros:

- **Starting Address:** Este campo define la dirección del comienzo de la tabla de memorias que usará la instrucción para intercambiar bytes.
- **Number of Bytes:** Este campo define el número de bytes a ser intercambiados a partir de la dirección Starting Address.
- **Byte Swap:** Define si será intercambiados todos los bytes o todos menos el primero.



5

Parámetro	Rango del DL06
Starting Address	Cualquier memoria V
Number of Bytes	Todas las memorias V o K1-128

Indicadores	Descripción
SP53	On si la CPU no puede ejecutar la instrucción.
SP71	On cuando un valor usado por la instrucción es inválido.

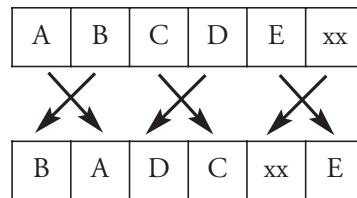
Preferencias de Byte Swap

No Byte Swapping (sin intercambio de bits)
(AIN, AEX, PRINTV, VPRINT)



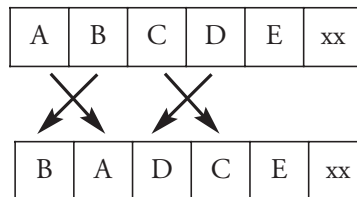
Byte	
Alto	Bajo
V2477	0005h
V2500	B A
V2501	D C
V2502	xx E

Byte Swap All (Intercambie todo)



Byte	
Alto	Bajo
V2477	0005h
V2500	A B
V2501	C D
V2502	E xx

Byte Swap All but Null (Intercambie todo excepto el carácter NULL)



Byte	
High	Low
V2477	0005h
V2500	B A
V2501	D C
V2502	xx E

Ejemplo de SWAPB

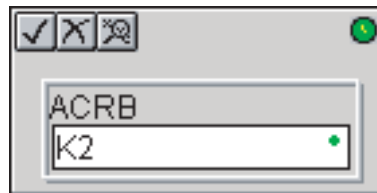
El bit Complete de AIN se usa para activar la instrucción SWAPB. Use la instrucción STRPD para que la instrucción SWAPB sea ejecutada en un barrido solamente.



La instrucción ASCII Clear Buffer (ACRB)

DS5	Usado	Esta instrucción limpia el buffer (memoria intermedia de almacenaje temporario) de recibimiento del puerto 2 del PLC DL06 de los caracteres recibidos.
HPP	N/A	

Ejemplo de ACRB



El bit AIN Complete o los bits de diagnóstico de AIN se utilizan para limpiar el buffer o la memoria de almacenaje intermedia de caracteres ASCII.



Esta página ha sido dejada en blanco intencionalmente.

Instrucciones (IBox) o Cuadros Inteligentes

Las instrucciones designadas comúnmente Iboxes, enumeradas en esta sección son nuevas instrucciones disponibles al usar *DirectSOFT5* para programar su PLC DL06 (el PLC DL06 requiere la versión v2.10 de firmware o mas nuevo para usar las nuevas funciones de *DirectSOFT5*). Para más información sobre *DirectSOFT5* y para bajar una versión gratuita, visite por favor nuestro sitio de Internet en: www.automationdirect.com.

IBoxes de ayuda de señales analógicas

Instrucción	Ibox #	Página
Analog Input / Output Combo Module PunteroSetup (ANLGCMB)	IB-462	5-232
Analog Input Module PunteroSetup (ANLGIN)	IB-460	5-234
Analog Output Module PunteroSetup (ANLGOUT)	IB-461	5-236
Analog Scale 12 Bit BCD to BCD (ANSCL)	IB-423	5-238
Analog Scale 12 Bit Binary to Binary (ANSCLB)	IB-403	5-239
Filter Over Time - BCD (FILTER)	IB-422	5-240
Filter Over Time - Binary (FILTERB)	IB-402	5-242
Hi/Low Alarm - BCD (HILOAL)	IB-421	5-244
Hi/Low Alarm - Binary (HILOALB)	IB-401	5-246

IBoxes de ayuda de señales discretas

Instrucción	Ibox #	Página
Off Delay Timer (OFFDTMR)	IB-302	5-248
On Delay Timer (ONDTMR)	IB-301	5-250
One Shot (ONESHOT)	IB-303	5-252
Push On / Push Off Circuit (PONOFF)	IB-300	5-253

IBoxes de memorias

Instrucción	Ibox #	Página
Move Single Word (MOVEW)	IB-200	5-254
Move Double Word (MOVED)	IB-201	5-255

IBoxes de aritmética

Instrucción	Ibox #	Página
BCD to Real with Implied Decimal Point (BCDTOR)	IB-560	5-256
Double BCD to Real with Implied Decimal Point (BCDTORD)	IB-562	5-257
Math - BCD (MATHBCD)	IB-521	5-258
Math - Binary (MATHBIN)	IB-501	5-260
Math - Real (MATHR)	IB-541	5-262
Real to BCD with Implied Decimal Point and Rounding (RTOBCD)	IB-561	5-263
Real to Double BCD with Implied Decimal Point and Rounding (RTOBCDD)	IB-563	5-264
Square BCD (SQUARE)	IB-523	5-265
Square Binary (SQUAREB)	IB-503	5-266
Square Real(SQUARER)	IB-543	5-267
Sum BCD Numbers (SUMBCD)	IB-522	5-268
Sum Binary Numbers (SUMBIN)	IB-502	5-269
Sum Real Numbers (SUMR)	IB-542	5-270

IBoxes de Comunicación		
Instrucción	Ibox #	Página
ECOM100 Configuration (ECOM100)	IB-710	5-272
ECOM100 Disable DHCP (ECDHCPD)	IB-736	5-274
ECOM100 Enable DHCP (ECDHCPE)	IB-735	5-276
ECOM100 Query DHCP Setting (ECDHCPQ)	IB-734	5-278
ECOM100 Send E-mail (ECEMAIL)	IB-711	5-280
ECOM100 Restore Default E-mail Setup (ECEMRDS)	IB-713	5-281
ECOM100 E-mail Setup (ECEMSUP)	IB-712	5-286
ECOM100 IP Setup (ECIPSUP)	IB-717	5-290
ECOM100 Read Descripción (ECRDDES)	IB-726	5-292
ECOM100 Read Gateway Address (ECRDGWA)	IB-730	5-294
ECOM100 Read IP Address (ECRDIP)	IB-722	5-296
ECOM100 Read Module ID (ECRDMID)	IB-720	5-298
ECOM100 Read Module Name (ECRDNAM)	IB-724	5-300
ECOM100 Read Subnet Mask (ECRDSNM)	IB-732	5-302
ECOM100 Write Descripción (ECWRDES)	IB-727	5-304
ECOM100 Write Gateway Address (ECWRGWA)	IB-731	5-302
ECOM100 Write IP Address (ECWRIP)	IB-723	5-304
ECOM100 Write Module ID (ECWRMID)	IB-721	5-310
ECOM100 Write Name (ECWRNAM)	IB-725	5-312
ECOM100 Write Subnet Mask (ECWRSNM)	IB-733	5-314
ECOM100 RX Network Read (ECRX)	IB-740	5-316
ECOM100 WX Network Write (ECWX)	IB-741	5-319
NETCFG Network Configuration (NETCFG)	IB-700	5-322
Network RX Read (NETRX)	IB-701	5-324
Network WX Write (NETWX)	IB-702	5-327

IBoxes de CTRLIO		
Instrucción	Ibox #	Página
CTRLIO Configuration (CTRLIO)	IB-1000	5-330
CTRLIO Add Entry to End of Preset Table (CTRADPT)	IB-1005	5-332
CTRLIO Clear Preset Table (CTRCLRT)	IB-1007	5-335
CTRLIO Edit Preset Table Entry (CTREDPT)	IB-1003	5-338
CTRLIO Edit Preset Table Entry and Reload (CTREDRL)	IB-1002	5-342
CTRLIO Initialize Preset Table (CTRINPT)	IB-1004	5-346
CTRLIO Initialize Preset Table (CTRINTR)	IB-1010	5-350
CTRLIO Load Profile (CTRLDPR)	IB-1001	5-354
CTRLIO Read Error (CTRRDER)	IB-1014	5-357
CTRLIO Run to Limit Mode (CTRRTLML)	IB-1011	5-359
CTRLIO Run to Position Mode (CTRRTPM)	IB-1012	5-362
CTRLIO Velocity Mode (CTRVELO)	IB-1013	5-365
CTRLIO Write File to ROM (CTRWFTR)	IB-1006	5-368

Configurador del módulo de entradas y salidas análogas (ANLGCMB) (IB-462)

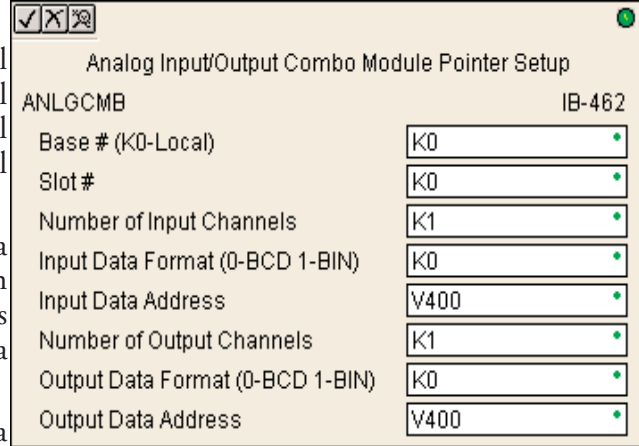
DS5	Usado
HPP	N/A

Esta instrucción genera la lógica para configurar el método del puntero para un módulo análogo combinación de entradas y salidas en el primer barrido del PLC después de una transición de modo program a RUN.

La instrucción ANLGCMB determina el formato de datos y las direcciones del puntero basadas en el tipo de CPU, el número de la base y de la ranura del módulo.

La dirección de datos de entrada es la localización inicial de memoria V en donde serán almacenados los valores de los datos de entrada análoga y crea una localización para cada canal de entrada.

La dirección de datos de salidas es la localización inicial de memoria V de usuario en donde los valores de los datos de salidas análoga serán puestos por código ladder o un dispositivo externo, creando una localización para cada canal de salida.



Puesto que la lógica de IBox se ejecuta solamente en el primer barrido, la instrucción no puede tener ninguna lógica de entrada.

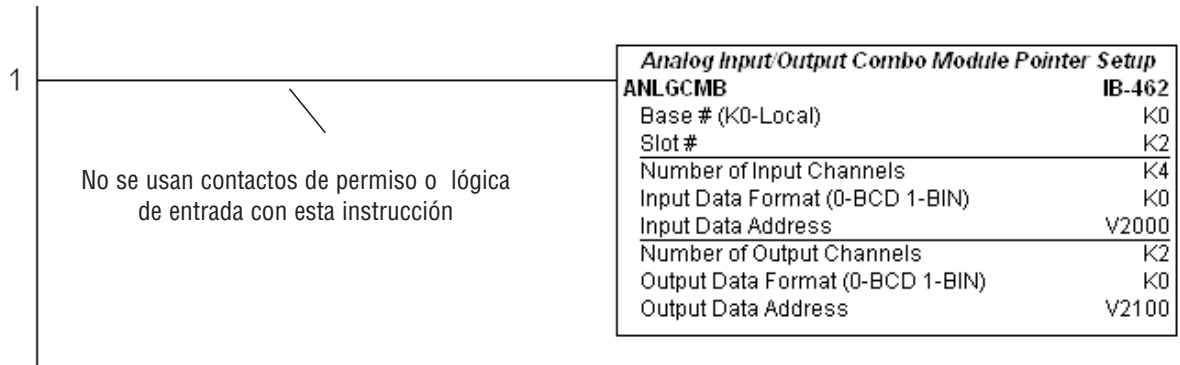
Parámetros ANLGCMB

- **Base # (K0-Local):** Debe ser 0 para PLC DL06.
- **Slot # (Ranura):** Especifica qué ranura de opción del PLC es ocupada por el módulo análogo (1-4)
- **Number of Input Channels:** Especifica el número de canales de entradas análogas a ser explorados.
- **Input Data Format (0-BCD 1-BIN):** Especifica el formato de datos de entradas análogas (BCD o binario) - el formato binario puede ser usado para exhibir datos en paneles de interface de operador.
- **Input Data Address:** Especifica la localización de memoria V inicial que será utilizada para almacenar los datos de entradas análogas.
- **Number of Output Channels:** Especifica el número de canales de salidas análogas que serán usados.
- **Output Data Format (0-BCD 1-BIN):** Especifica el formato de los datos de salidas análogas (BCD o binario).
- **Output Data Address:** Especifica la localización de memoria V inicial que será usada como origen de los datos de salidas análogas.

Parámetro	Rango del DL06
Base # (K0-Local) K	K0 (Solamente base local)
Slot # K	K1-4
Number of Input Channels K	K1-8
Input Data Format (0-BCD 1-BIN) K	BCD: K0; Binario: K1
Input Data Address V	Vea el mapa de memoria V del DL06 - Data Words
Number of Output Channels K	K1-8
Output Data Format (0-BCD 1-BIN) K	BCD: K0; Binario: K1
Output Data Address V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ANLGCMB

En el ejemplo siguiente, se usa la instrucción ANLGCMB para configurar el método del puntero para un módulo de combinación de E/S análogas que esté instalado en la ranura de opción 2. Se activan cuatro canales de entradas y los datos de salidas análogas serán escritos a V2000 - V2003 en formato BCD. Se activan dos canales de salidas y los valores análogos serán leídos en V2100 - V2101 en formato BCD.



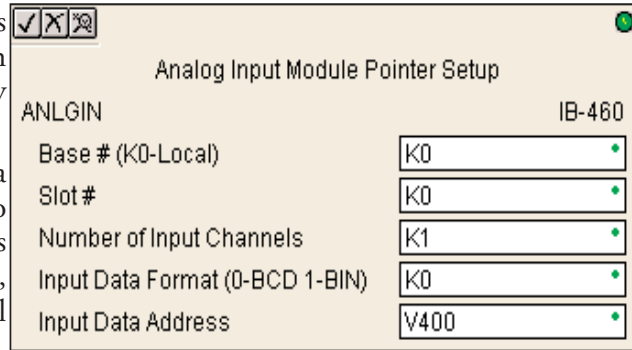
Configurador del módulo de entrada analoga (ANLGIN) (IB-460)

DS5	Usado
HPP	N/A

La configuración del módulo de entradas análogas genera la lógica para configurar el método del puntero para un módulo de entradas análogas en el primer barrido del PLC después de una transición de modo program a RUN.

Este IBox determina el formato de datos y las direcciones del puntero basadas en el tipo de CPU, el número de la base y de la ranura.

La dirección de los datos de entrada es la localización de memoria V de usuario inicial en donde serán almacenados los valores de datos de entradas análogas, creando una localización para cada canal de entrada.



Puesto que esta lógica se ejecuta solamente en el primer barrido, este IBox no puede tener ninguna lógica de entrada.

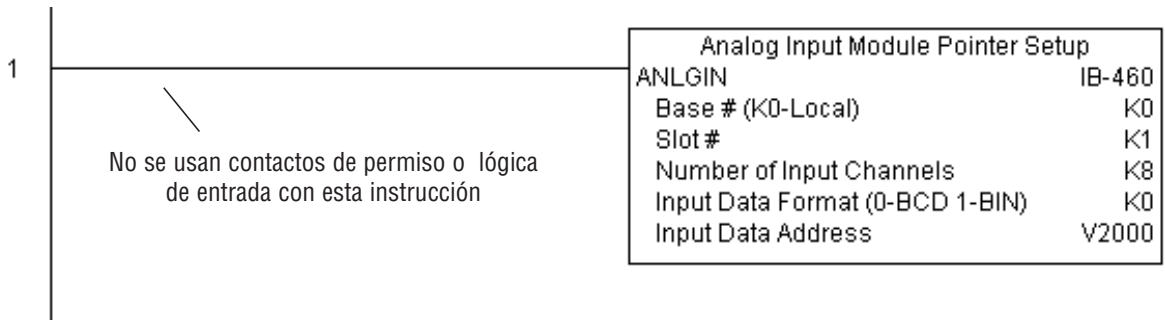
Parámetros ANLGIN

- **Base # (K0-Local):** Debe ser 0 para PLC DL06.
- **Slot #:** Especifica qué ranura de opción del PLC es ocupada por el módulo analoga (1-4)
- **Number of Input Channels:** Especifica el número de canales de entradas análogas a ser explorados.
- **Input Data Format (0-BCD 1-BIN):** Especifica el formato de datos de entradas análogas (BCD o binario) - el formato binario puede ser usado para exhibir datos en paneles de interface de operador.
- **Input Data Address:** Especifica la localización de memoria V inicial que será utilizada para almacenar los datos de entradas análogas.

Parámetro	Rango del DL06
Base # (K0-Local) K	K0 (Solamente base local)
Slot # K	K1-4
Number of Input Channels K	K1-8
Input Data Format (0-BCD 1-BIN) K	BCD: K0; Binario: K1
Input Data Address V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ANLGIN

En el ejemplo siguiente, se usa la instrucción ANLGIN para configurar el método del puntero para un módulo de entradas análogas que esté instalado en la ranura de opción 1. Se activan ocho canales de entradas y los datos análogos serán escritos a V2000 - V2007 en formato BCD.



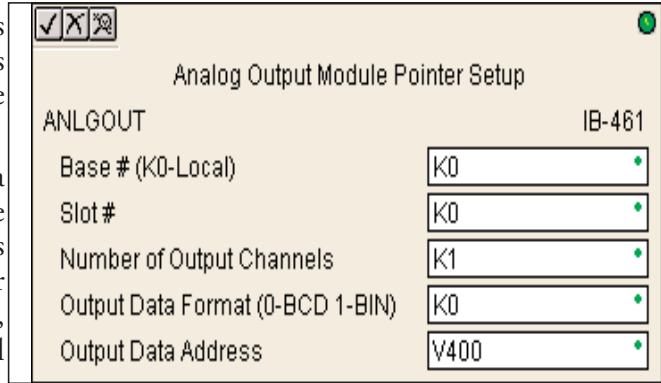
Configurador del módulo de salidas análogas (ANLGOUT) (IB-461)

DS5	Usado
HPP	N/A

La instrucción configurador del módulo de salidas análogas genera la lógica para configurar el método del puntero para un módulo de salidas análogas en el primer barrido del PLC después de una transición de modo program a RUN.

Este IBox determina el formato de datos y las direcciones del indicador basadas en el tipo de CPU, el número de la base y de la ranura.

La dirección de los datos de salidas es la localización inicial de memoria V de usuario en donde los valores de los datos de salidas análogas serán puestos por código ladder o un dispositivo externo, siendo una localización para cada canal de salida.



Puesto que esta lógica se ejecuta solamente en el primer barrido, este IBox no puede tener ninguna lógica de entrada.

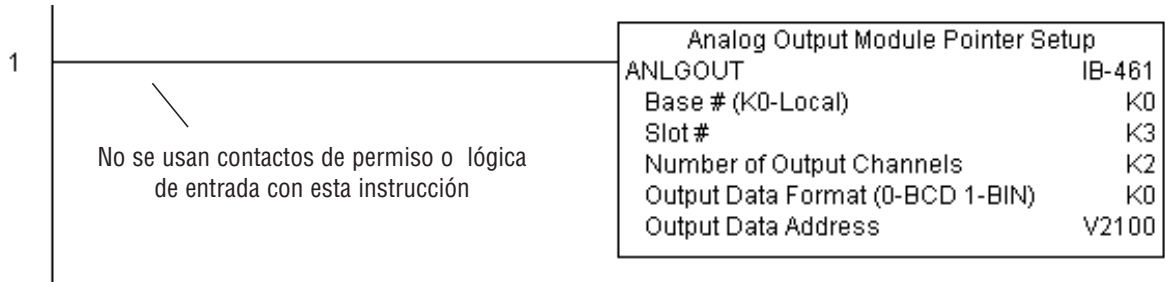
Parámetros ANLGOUT

- **Base # (K0-Local):** Debe ser 0 para PLC DL06.
- **Slot #:** Especifica qué ranura de opción del PLC es ocupada por el módulo análogo (1-4)
- **Number of Output Channels:** Especifica el número de canales de salidas análogas que serán usados.
- **Output Data For:** Especifica el formato de los datos de salidas análogas (BCD o binario).
- **Output Data Address:** Especifica la localización de memoria V inicial que será usada como origen de los datos de salidas análogas.

Parámetro	Rango del DL06
Base # (K0-Local) K	K0 (local base only)
Slot # K	K1-4
Number of Output Channels K	K1-8
Output Data Format (0-BCD 1-BIN). K	BCD: K0; Binary: K1
Output Data Address V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ANLGOUT

En el ejemplo siguiente, se utiliza la instrucción ANLGOUT para configurar el método del puntero para un módulo de salidas análogas que esté instalado en la ranura de opción 3. Son activados dos canales de salidas y los datos análogos serán leídos en V2100 - V2101 en formato BCD.

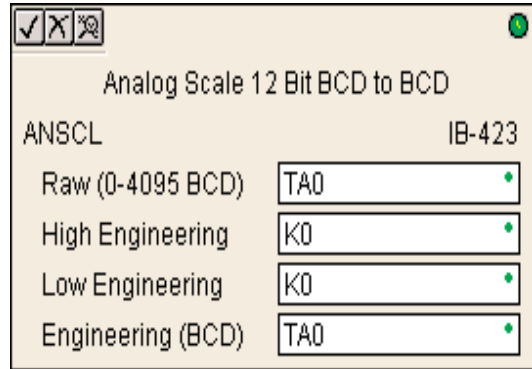


Escala de un valor análogo de 12 bits BCD a BCD (ANSCL) (IB-423)

DS5	Usado
HPP	N/A

Esta instrucción escala un valor análogo BCD de 12 bits (BCD 0-4095) en unidades de ingeniería BCD. Usted especifica el valor de la unidad de ingeniería más alto (cuando el valor sin escala es 4095), y el valor de ingeniería más bajo (cuando el valor sin escala es 0), y la dirección de memoria V de salida que usted desea poner el valor de unidad que dirige escalado. Las unidades de ingeniería se generan como BCD y pueden estar en el el rango completo de 0 a 9999 (Vea la instrucción ANSCLB - si sus unidades sin escala están en formato binario).

Observe que esta instrucción IBox trabaja solamente con valores unipolares sin escala positivos. No trabaja con valores bipolares ni con valores crudos de magnitud más signo.



Parámetros ANSCL

- **Raw (0-4095 BCD):** Especifica la localización de la memoria V donde está el valor unipolar sin escala de rango 0-4095.
- **High Engineering:** Especifica el alto valor de ingeniería cuando la entrada es 4095.
- **Low Engineering:** Especifica el alto bajo de ingeniería cuando la entrada es 0.
- **Engineering (BCD):** Especifica la memoria V en donde será colocado el valor a escala BCD.

Parámetro	Rango del DL06
Raw (0-4095 BCD) V,P	Vea el mapa de memoria V del DL06 - Data Words
High Engineering K	K0-9999
Low Engineering K	K0-9999
Engineering (BCD) V,P	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ANSCL

En el ejemplo siguiente, se utiliza la instrucción ANSCL para colocar a escala un valor crudo (BCD 0-4095) que esté en V2000. El rango de la escala de ingeniería se define como 0-100 (valor bajo de ingeniería - alto valor de ingeniería). El valor a escala será colocado en V2100 en formato BCD.

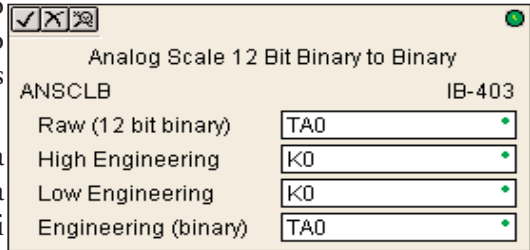


Escala de un valor análogo de 12 bits binario a binario (ANSCLB) (IB-403)

DS5	Usado
HPP	N/A

Esta instrucción escala un valor análogo binario de 12 bits (0-4095 decimal) en unidades de ingeniería binarias. Usted especifica el valor de la unidad de ingeniería más alto (cuando el valor sin escala es 4095), y el valor de ingeniería más bajo (cuando el valor sin escala es 0), y la dirección de memoria V de salida que usted desea poner el valor de unidad que dirige escalado. Las unidades de ingeniería se generan como binarias y pueden estar en el el rango completo de 0 a 9999 (Vea la instrucciónANSCL - si sus unidades sin escala están en formato BCD).

Observe que esta instrucción IBox trabaja solamente con valores unipolares sin escala positivos. No trabaja con valores bipolares ni con valores crudos de magnitud más signo.



5

Parámetros ANSCLB

- **Raw (12 bit binary):** Especifica la localización de la memoria V donde está el valor unipolar sin escala de rango (12 bit binario = 0-4095 decimal)
- **High Engineering:** Especifica el alto valor de ingeniería cuando la entrada es 4095.
- **Low Engineering:** Especifica el alto bajo de ingeniería cuando la entrada es 0.
- **Engineering (binary):** Especifica la memoria V en donde será colocado el valor a escala binario o decimal.

Parámetro	Rango del DL06
Raw (12 bit binary) V,P	Vea el mapa de memoria V del DL06 - Data Words
High Engineering K	K0-65535
Low Engineering K	K0-65535
Engineering (binary) V,P	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ANSCLB

En el ejemplo siguiente, la instrucción ANSCLB es usada para colocar a escala un valor crudo (0-4095 binario) que esté en V2000. El rango del escalamiento de ingeniería se define como 0-1000 (valor bajo de ingeniería - alto valor de ingeniería). El valor escalado será colocado en V2100 en formato binario.



Filtro - BCD (FILTER) (IB-422)

DS5	Usado
HPP	N/A

La instrucción FILTER realizará un filtro de primer orden en los datos en bruto sobre un intervalo definido de tiempo. La ecuación es:

Nuevo valor = Valor antiguo + [(Valor en bruto - Valor antiguo) / FDC]
 donde,

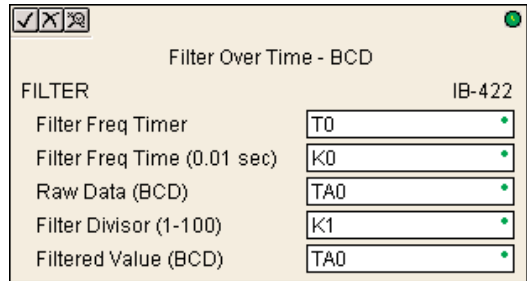
Nuevo valor: Nuevo valor filtrado

Valor antiguo: Valor filtrado antiguo

FDC: Constante divisor del filtro

Valor en bruto: Datos en bruto

La Constante divisor del filtro FDC es un número entero en el rango K1 a K100, tal que crea un amortiguamiento sobre el valor en bruto y si es igual a K1 entonces no será hecho ningún filtrado.



La frecuencia en la cual se realiza el cálculo se especifica por tiempo en centésimos de un segundo (0,01 segundo) como el parámetro del constante tiempo del filtro. Observe que hay una instrucción de temporizador embutida en el IBox y no debe ser usado en cualquier otro lugar en su programa. El control del renglón determina si el cálculo será ejecutado. Si es falso, el valor del filtro no es actualizado. En el primer barrido de donde pasa de modo program al modo RUN, el valor del filtro se inicializa a 0 para dar al cálculo un punto de partida consistente.

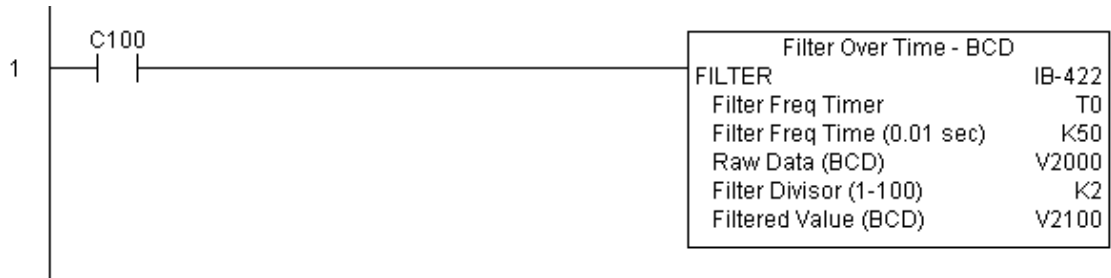
Parámetros de la instrucción FILTER

- **Filter Frequency Timer:** Especifica el número del temporizador (T) que es usado por la instrucción Filter
- **Filter Frequency Time (0.01sec):** Especifica la frecuencia en la cual se realiza el cálculo
- **Raw Data (BCD):** Especifica la localización de memoria V del valor sin filtro en bruto BCD
- **Filter Divisor (1-100):** Esta constante es usada para controlar el efecto de filtrado. Un valor más grande aumentará el efecto de alisamiento del filtro. Un valor de 1 resulta sin filtrado.
- **Filtered Value (BCD):** Especifica la localización de memoria C en donde será colocado el valor filtrado en BCD

Parámetro	Rango del DL06
Filter Frequency Timer T	T0-377
Filter Frequency Time (0.01 sec) K	K0-9999
Raw Data (BCD) V	Vea el mapa de memoria V del DL06 - Data Words
Filter Divisor (1-100) K	K1-100
Filtered Value (BCD) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de FILTER

En el ejemplo siguiente, es usada la instrucción FILTER para filtrar un valor en BCD que esté en V2000. El temporizador (T0) se coloca a 0.5 s, la frecuencia en la cual el cálculo del filtro será realizada. La constante del filtro se coloca en 2. Un valor más grande aumentará el efecto de alisamiento del filtro. Un valor de 1 resulta sin filtrado. El valor filtrado será colocado en V2100.



Filtro Binario (FILTERB) (IB-402)

La instrucción filtro binario (decimal) realizará un filtro de primer orden en los datos en bruto sobre un intervalo definido de tiempo. La ecuación es

DS5	Usado
HPP	N/A

Nuevo valor = Valor antiguo+ [(Valor en bruto- Valor antiguo) / FDC]
 donde,

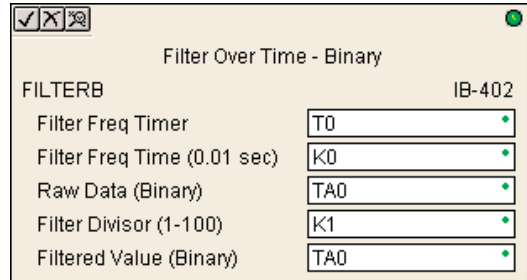
Nuevo valor: Nuevo valor filtrado

Valor antiguo: Valor filtrado antiguo

FDC: Constantee divisor del filtro

Valor en bruto Datos en bruto

La Constantee divisor del filtro FDC es un número entero en el rango K1 a K100, tal que crea un amortiguamiento sobre el valor en bruto y si es igual a K1 entonces no sería hecho ningun filtrado.



La frecuencia en la cual se realiza el cálculo se especifica por tiempo en centésimos de un segundo (0,01 segundo) como el parámetro del constante tiempo del filtro. Observe que hay una instrucción de temporizador embutida en el IBox y no debe ser usado en cualquier otro lugar en su programa. El control del renglón determina si el cálculo será ejecutado. Si es falso, el valor del filtro no es actualizado. En el primer barrido deonde pasa de modo program al modo RUN, el valor del filtro se inicializa a 0 para dar al cálculo un punto de partida consistente.

Parámetros de la instrucción FILTERB

- **Filter Frequency Timer:** Especifica el número del temporizador (T) que es usado por la instrucción FilterB
- **Filter Frequency Time (0.01sec):** Especifica la frecuencia en la cual se realiza el cálculo
- **Raw Data (Binario):** Especifica la localización de memoria V del valor sin filtro en bruto binario (decimal)
- **Filter Divisor (1-100):** Esta contante es usada para controlar el efecto de filtrado. Un valor más grande aumentará el efecto que alisamiento del filtro.Un valor de 1 resulta sin filtrado.
- **Filtered Value (Binario):** Especifica la localización de memoria C en donde será colocado el valor filtrado en binario

Parámetro	Rango del DL06
Filter Frequency Timer T	T0-377
Filter Frequency Time (0.01 sec) K	K0-9999
Raw Data (Binary) V	Vea el mapa de memoria V del DL06 - Data Words
Filter Divisor (1-100) K	K1-100
Filtered Value (Binary) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de FILTERB

En el ejemplo siguiente, es usada la instrucción FILTERB para filtrar un valor en binario que está en V2000. El temporizador (T1) se coloca a 0,5 s, la frecuencia en la cual el cálculo del filtro será realizada. La constante del filtro se coloca en 3.0. Un valor más grande aumentará el efecto de alisamiento del filtro. Un valor de 1 resulta sin filtrado. El valor filtrado será colocado en V2100.

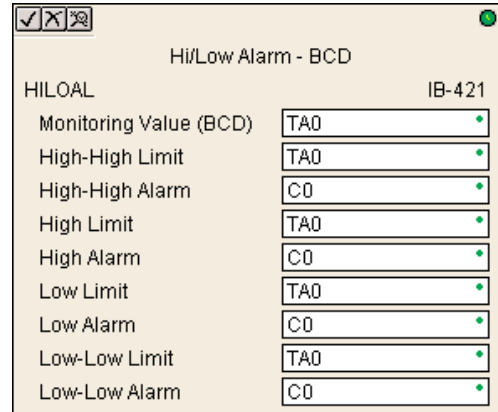


Alarma Hi/Low - BCD (HILOAL) (IB-421)

DS5	Usado
HPP	N/A

Esta instrucción supervisa el valor BCD de una posición de memoria V y configura cuatro estados posibles de alarmas, Alta-Alta, Alta, Baja, y Baja-Baja siempre que la instrucción IBox sea verdadera. Usted define los niveles de alarmas como valores constantes BCD (K0-k9999) o como valor de memoria V en BCD.

Usted debe asegurarse de que los niveles de alarma sean válidos, esto es, $HH \geq H > L \geq LL$. Note que cuando la condición de alarma Alto-Alta o Baja-Baja es verdadera, la alarma alta y baja también estará activada, respectivamente. Esto significa que usted puede usar el mismo nivel y la misma alarma para las alarmas Alta-Alta y Alta en caso de que usted necesite solamente una "alarma Alta". También observe que las condiciones de límite son inclusivas. Es decir, si el límite bajo es K50, y el límite Baja-Baja es K10, y si el valor de supervisión iguala 10, después la alarma Baja y la alarma Baja-Baja quiere ambas estén ENCENDIDAS. Si no hay flujo de energía al IBox, entonces todos los bits de alarmas serán desactivados sin importar el valor del parámetro de supervisión.



Parámetros de HILOAL

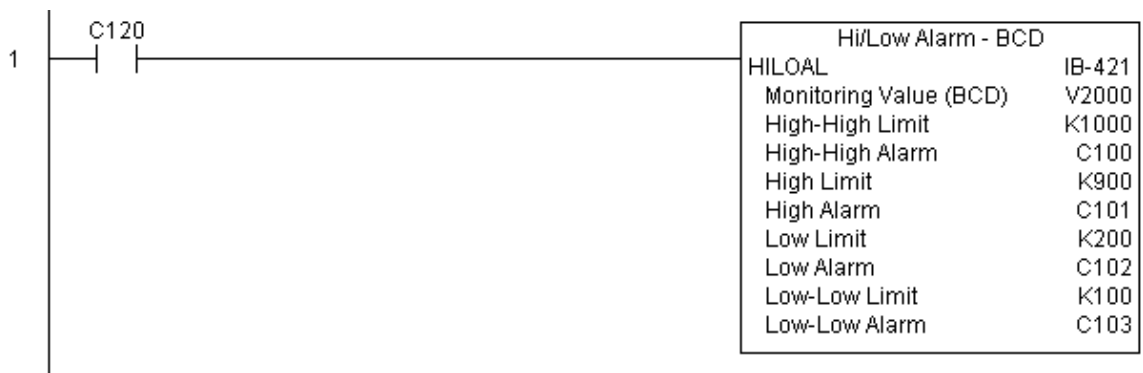
- **Monitoring Value (BCD):** Especifica la memoria V del valor BCD que se supervisará.
- **High-High Limit:** Constante o memoria V que especifica el nivel de alarma Alta-Alta.
- **High-High Alarm:** Bit de alarma Alta Alta activada cuando se alcanza el nivel High-High limit
- **High Limit:** Memoria V o constante que especifica el nivel de alarma Alta
- **High Alarm:** Bit de alarma Alta activada cuando se alcanza el nivel High limit
- **Low Limit:** Memoria V o constante que especifica el nivel de alarma Baja
- **Low Alarm:** Bit de alarma Baja activada cuando se alcanza el nivel Low limit
- **Low-Low Limit:** Memoria V o constante que especifica el nivel de alarma Low Low limit
- **Low-Low Alarm:** Bit de alarma Baja activada cuando se alcanza el nivel

Parámetro	Rango del DL06
Monitoring Value (BCD) V	Vea el mapa de memoria V del DL06 - Data Words
High-High Limit V, K	K0-9999; or Vea el mapa de memoria V del DL06 - Data Words
High-High Alarm X, Y, C, GX,GY, B	Vea el mapa de memoria DL06
High Limit V, K	K0-9999; or Vea el mapa de memoria V del DL06 - Data Words
High Alarm X, Y, C, GX,GY, B	Vea el mapa de memoria DL06
Low Limit V, K	K0-9999; or Vea el mapa de memoria V del DL06 - Data Words
Low Alarm X, Y, C, GX,GY,B	Vea el mapa de memoria DL06
Low-Low Limit V, K	K0-9999; or Vea el mapa de memoria V del DL06 - Data Words
Low-Low Alarm X, Y, C, GX,GY, B	Vea el mapa de memoria DL06

Ejemplo de HILOAL

En el ejemplo siguiente, la instrucción de HILOAL es usada para supervisar un valor BCD que está en V2000. Si el valor en V2000 excede el valor de K900, se activará C101. Si el valor continúa aumentando hasta el nivel High-high, se activará el bit C100. Ambos bits estarían encendido en este caso. Los niveles y alarmas alta y alta-alta se pueden definir al mismo valor si se desea usar una alarma "Alta".

Si el valor en V2000 es igual o cae debajo del nivel K200, se desactivará C102. Si el valor continúa disminuyendo debajo del límite Bajo-Bajo K100, se desactivará C103. Ambos bits estarían encendido en este caso. Las alarmas baja y baja-baja se pueden definir al mismo valor si se desea usar una alarma "Baja".

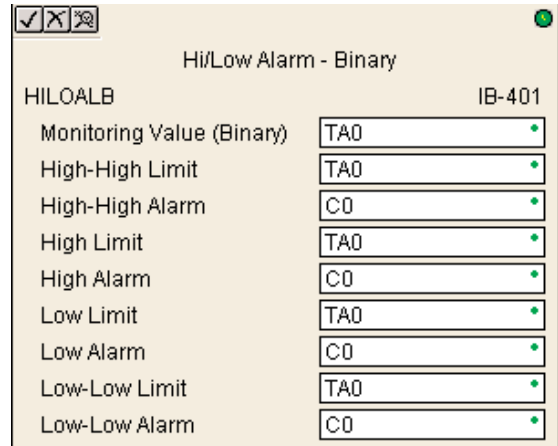


Alarm Hi/Low- Binaria (HILOALB) (IB-401)

DS5	Usado
HPP	N/A

Esta instrucción supervisa el valor binario de una posición de memoria V y configura cuatro estados posibles de alarmas, Alta-Alta, Alta, Baja, y Baja-Baja siempre que la instrucción IBox sea verdadera. Usted define los niveles de alarmas como valores constantes binarios (K0-K65535) o como valor de memoria V en binario.

Usted debe asegurarse de que los niveles de alarma sean válidos, esto es, $HH \geq H > L > LL$. Note que cuando la condición de alarma Alto-Alta o Baja-Baja es verdadera, la alarma alta y baja también estará activada, respectivamente. Esto significa que usted puede usar el mismo nivel y la misma alarma para las alarmas Alta-Alta y Alta en caso de que usted necesite solamente una "alarma Alta". También observe que las condiciones de límite son inclusivas. Es decir, si el límite bajo es K50, y el límite Baja-Baja es K10, y si el valor de supervisión iguala 10, después la alarma Baja y la alarma Baja-Baja quiere ambas estén ENCENDIDAS. Si no hay flujo de energía al IBox, entonces todos los bits de alarmas serán desactivados sin importar el valor del parámetro de supervisión.



Parámetros de HILOALB

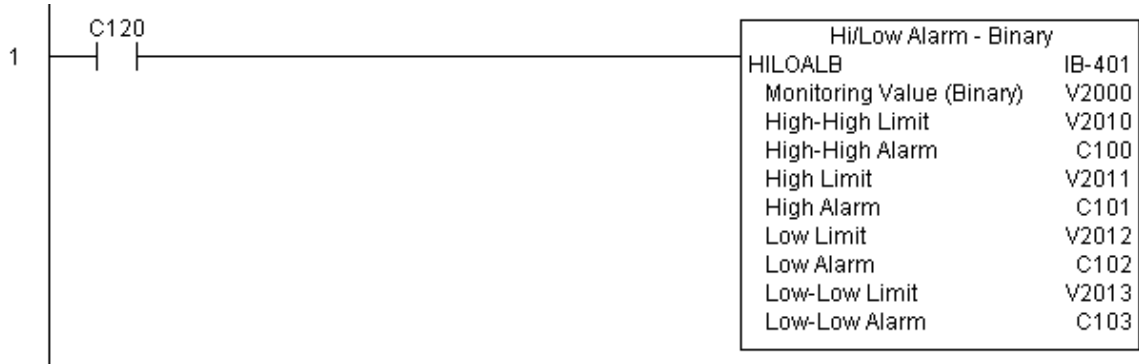
- **Monitoring Value (Binario):** Especifica la memoria V del valor binario que se supervisará.
- **High-High Limit:** Constante o memoria V que especifica el nivel de alarma Alta-Alta.
- **High-High Alarm:** Bit de alarma Alta Alta activada cuando se alcanza el nivel High-High limit
- **High Limit:** Memoria V o constante que especifica el nivel de alarma Alta
- **High Alarm:** Bit de alarma Alta activada cuando se alcanza el nivel High limit
- **Low Limit:** Memoria V o constante que especifica el nivel de alarma Baja
- **Low Alarm:** Bit de alarma Baja activada cuando se alcanza el nivel Low limit
- **Low-Low Limit:** Memoria V o constante que especifica el nivel de alarma Low Low limit
- **Low-Low Alarm:** Bit de alarma Baja activada cuando se alcanza el nivel

Parámetro	Rango del DL06
Monitoring Value (Binario) V	Vea el mapa de memoria V del DL06 - Data Words
High-High Limit V, K	K0-65535; o Vea el mapa de memoria V del DL06 - Data Words
High-High Alarm X, Y, C, GX,GY, B	Vea el mapa de memoria DL06
High Limit V, K	K0-65535; o Vea el mapa de memoria V del DL06 - Data Words
High Alarm X, Y, C, GX,GY, B	Vea el mapa de memoria DL06
Low Limit V, K	K0-65535; o Vea el mapa de memoria V del DL06 - Data Words
Low Alarm X, Y, C, GX,GY,B	Vea el mapa de memoria DL06
Low-Low Limit V, K	K0-65535; o Vea el mapa de memoria V del DL06 - Data Words
Low-Low Alarm X, Y, C, GX,GY, B	Vea el mapa de memoria DL06

Ejemplo de HILOALB

En el ejemplo siguiente, la instrucción HILOALB es usada para supervisar un valor binario que esté en V2000. Si el valor en V2000 es igual o mayor que el límite alto del valor binario en V2011, el bit C101 se activa. Si el valor continúa aumentando hasta ser igual o ser mas grande que el valor límite Alto-Alto en V2010, se activará el bit C100. Ambos bits estarían encendidos en este caso. Los límites y las alarmas Alta y Alta-Alta se pueden configurar al mismo valor o memoria V si se desea usar un límite o alarma "Alta".

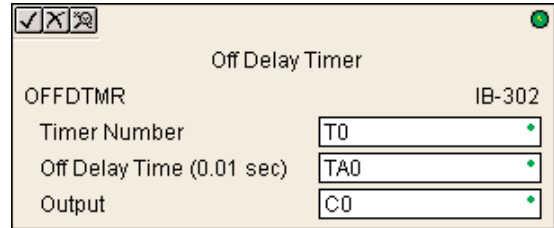
Si el valor en V2000 satisface o baja debajo del límite bajo del valor binario en V2012, se activará el bit C102. Si el valor continúa disminuyendo debajo del límite Bajo-Bajo en V2013, se activará el bit C103. Ambos bits estarían encendidos en este caso. Los límites y las alarmas Baja y Baja-Baja se pueden definir con la misma memoriaV o el mismo valor si se desea usar un límite o alarma Baja.



Temporizador Off Delay (OFFDTMR) (IB-302)

El temporizador Off delay retrasa "el apagado" del parámetro de salida (Output) especificado en la instrucción (en centésimo de segundo) basado en el flujo de energía en el IBox. Una vez que el IBox reciba energía, el bit de salida se encenderá inmediatamente. Cuando el flujo de energía al IBox hace falso, la salida seguirá ENCENDIDA por la cantidad de tiempo especificada (en centésimo de segundo). Una vez que el tiempo ha expirado, la salida se apagará. Si el flujo de energía al IBox se hace verdadero ANTES DE QUE el tiempo de retardo se haya cumplido, el temporizador se rearma y la salida seguirá encendida - así que usted no debe continuamente tener NINGUN flujo de energía al IBox POR LO MENOS el tiempo de retardo especificado antes de que la salida se apague.

DS5	Usado
HPP	N/A



Este IBox utiliza un temporizador (TMRF), que no debe ser usado en cualquier otro lugar en el programa ladder.

Parámetros de OFFDTMR

- **Timer Number:** Especifica el número del temporizador (TMRF) que es usado por la instrucción OFFDTMR
- **Off Delay Time (0,01sec):** Especifica cuánto tiempo la salida seguirá encendida si el flujo de energía al Ibox se torna falso
- **Output:** Especifica la salida que será apagada con un retardo de tiempo.

Parámetro	Rango del DL06
Timer Number T	T0-377
Off Delay Time K,V	K0-9999; Vea el mapa de memoria V del DL06 - Data Words
Output X, Y, C, GX,GY, B	Vea el mapa de memoria DL06

Ejemplo de OFFDTMR

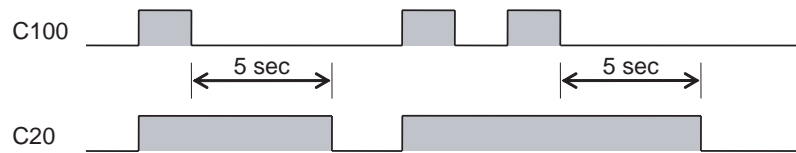
En el ejemplo siguiente, se usa la instrucción OFFDTMR para retrasar la salida C20. El temporizador 2 (t2) define el retardo en 5 segundos.

Cuando se cierra el contacto C100, C20 se activa y permanecerá encendido mientras C100 está encendido. Cuando se abre el contacto C100, C20 permanecerá activado por el tiempo especificado (5s), y luego se apaga.



5

Ejemplo de diagrama de tiempos

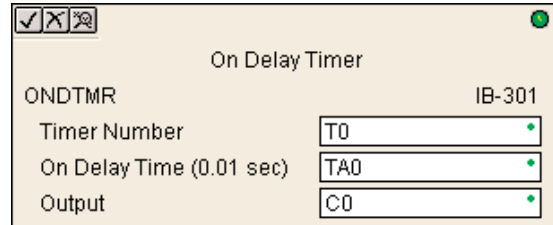


Temporizador On Delay (ONDTMR) (IB-301)

DS5	Usado
HPP	N/A

El temporizador On delay retrasa el tiempo en que el parámetro de salida se activa por la cantidad de tiempo especificada (en centésimo de segundo) basada en el flujo de energía en el IBox. Una vez que el IBox pierde la energía, la salida se desactiva inmediatamente. Si el flujo de energía se apaga ANTES DE QUE transcurra el tiempo de retardo, entonces el contador de tiempo SE REARMA y la salida se apaga, así que usted debe tener flujo de energía continuo al IBox por lo menos el tiempo de retardo especificado antes de que la salida se active.

Este IBox utiliza un temporizador (TMRF), que no puede ser usado en cualquier otro lugar en el programa.



5

Parámetros de ONDTMR

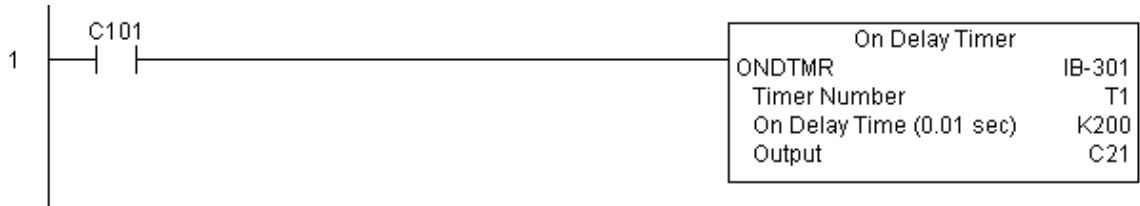
- **Timer Number:** Especifica el número del temporizador (TMRF) que es usado por la instrucción ONDTMR
- **On Delay Time (0,01sec):** Especifica cuánto tiempo la salida se encenderá si el flujo de energía al Ibox es verdadero.
- **Output:** Especifica la salida que será encendida con un retardo de tiempo.

Parámetro	Rango del DL06
Timer Number T	T0-377
On Delay Time K,V	K0-9999; Vea el mapa de memoria V del DL06 - Data Words
Output X, Y, C, GX,GY, B	Vea el mapa de memoria DL06

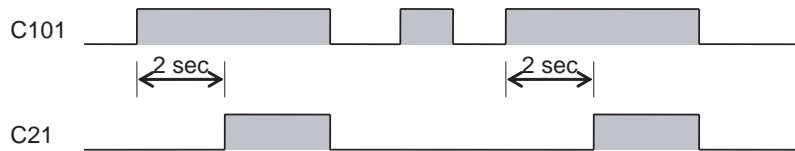
Ejemplo de ONDTMR

En el ejemplo siguiente, la instrucción ONDTMR es usada para retardar el "encendido" de la salida C21. El temporizador 1 (T1) define como de 2 segundos el período de "atraso".

Cuando se cierra el contacto C101, se cierra el contacto C21 con un atraso de 2 segundos. Cuando se abre el contacto C101, el contacto C21 se abre inmediatamente.



Ejemplo de diagrama de tiempos



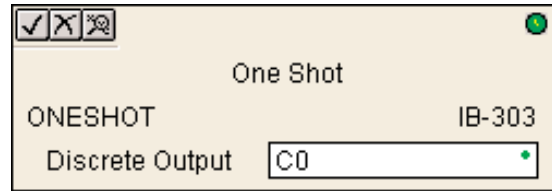
One Shot (ONESHOT) (IB-303)

DS5	Usado
HPP	N/A

La instrucción One Shot encenderá el bit de salida definido en el parámetro durante un barrido en la transición desde apagado a encendido del flujo de energía en el IBox. Este IBox es simplemente un nombre diferente para la bobina PD (diferencial positivo).

Parámetros de ONESHOT

- **Discrete Output:** Especifica la salida que estará encendida por un barrido

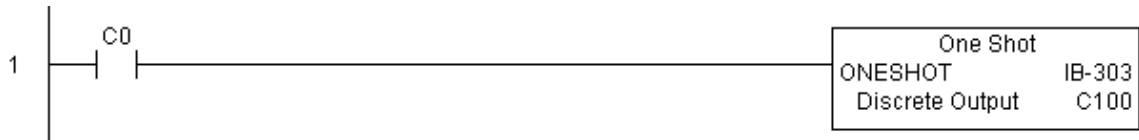


5

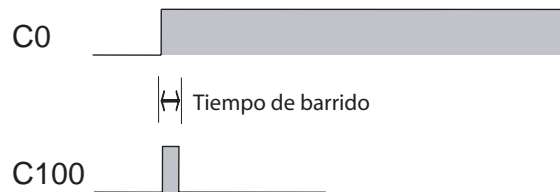
Parámetro	Rango del DL06
Discrete Output X, Y, C	Vea el mapa de memoria DL06

Ejemplo de ONESHOT

En el ejemplo siguiente, la instrucción ONESHOT es usada para activar C100 por un barrido del PLC cuando el contacto C0 haga la transición desde abierto para cerrado. La lógica de entrada debe producir una transición desde abierto para cerrado para ejecutar la instrucción One shot.



Ejemplo de diagrama de tiempos



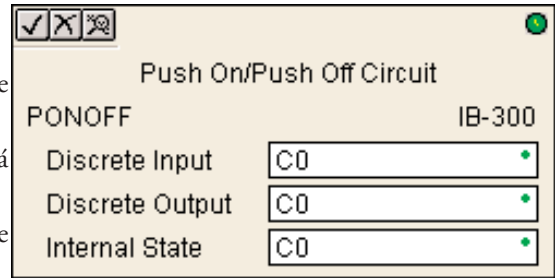
Circuito Push On / Push Off (PONOFF) (IB-300)

DS5	Usado
HPP	N/A

Esta instrucción conmuta un estado de la salida siempre que el flujo de la energía de la entrada haga una transición de apagado a encendido. Requiere un parámetro adicional de bit para trabajar con la información del estado. Este bit adicional no debe ser usado en cualquier otro lugar en el programa. Esto también se conoce como "circuito de flip-flop".

Parámetros de PONOFF

- **Discrete Input:** Especifica la entrada que conmutará la salida especificada
- **Discrete Output:** Especifica la salida que será "conmutada."
- **Internal State:** Especifica un bit de trabajo que será usado por la instrucción



5

Parámetro	Rango del DL06
Discrete Input X,Y,C,S,T,CT,GX,GY,SP,B,PB	Vea el mapa de memoria DL06
Discrete Output X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Internal State X, Y, C	Vea el mapa de memoria DL06

Ejemplo de PONOFF

En el ejemplo siguiente, la instrucción PONOFF es usada para controlar los estados de la salida C20 con una sola entrada C10. Cuando el contacto C10 se cierra una vez, el bit C20 se activa. Cuando el contacto C10 se cierra otra vez, el bit C20 se apaga. El bit C100 es un usado internamente por la instrucción.



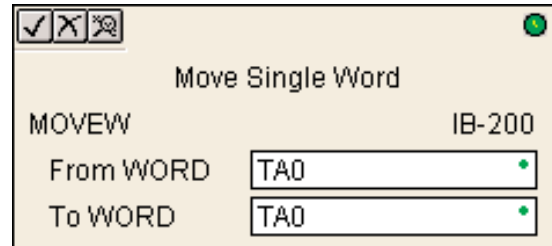
Mover una palabra (MOVEW) (IB-200)

DS5	Usado
HPP	N/A

Esta instrucción copia el contenido de una palabra a otra posición de memoria directamente o indirectamente con un puntero, ya sea como constante HEXADECIMAL, desde una posición de memoria, o indirectamente a través de un puntero.

Parámetros de MOVEW

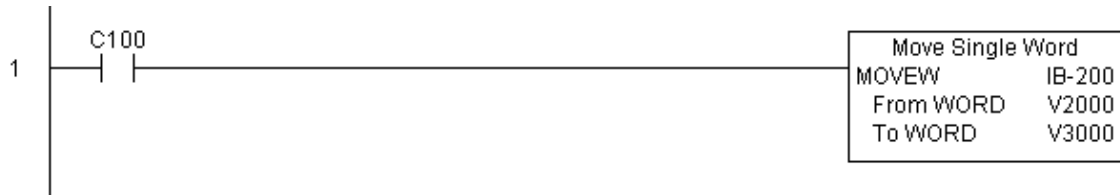
- **From WORD:** Especifica la palabra cuyo contenido será copiado a otra localización
- **To WORD:** Especifica la localización adonde será copiado el contenido de "From WORD"



Parámetro	Rango del DL06
From WORD V,P,K	K0-FFFF; Vea el mapa de memoria V del DL06 - Data Words
To WORD..... V,P	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de MOVEW

En el ejemplo siguiente, la instrucción MOVEW es usada para copiar los 16 bits de datos desde V2000 a V3000 cuando C100 se cierra.



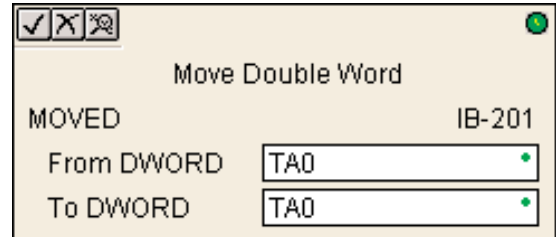
Mover una palabra doble (MOVED) (IB-201)

DS5	Usado
HPP	N/A

Esta instrucción copia el contenido de una palabra doble a a dos posiciones de memoria consecutivas directamente o indirectamente con un puntero, ya sea como constante de palabra doble HEXADECIMAL, desde una posición de memoria doble , o indirectamente a través de un puntero.

Parámetros de MOVED

- **From WORD:** Especifica la palabra doble cuyo contenido será copiado a otra localización
- **To WORD:** Especifica la localización adonde será copiado el contenido de "From WORD"



5

Parámetro	Rango del DL06
From DWORD V,P,K	K0-FFFFFFF; Vea el mapa de memoria V del DL06 - Data Words
To DWORD V,P	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de MOVED

En el ejemplo siguiente, la instrucción MOVED instrucción es usada para copiar los 32 bits de datos desde V2000 y V2001 a V3000 y V3001 cuando C100 se cierra.

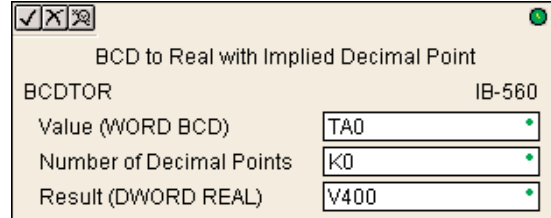


BCD a Real (BCDTOR) (IB-560)

DS5	Usado
HPP	N/A

Esta instrucción convierte un valor dado en una palabra BCD de 4 dígitos a un número Real, con una coma definida por la cantidad de decimales(K0-K4).

Por ejemplo, BCDTOR K1234 con un número implicado de coma igual a K1, resultaría R123.4



Parámetros de BCDTOR

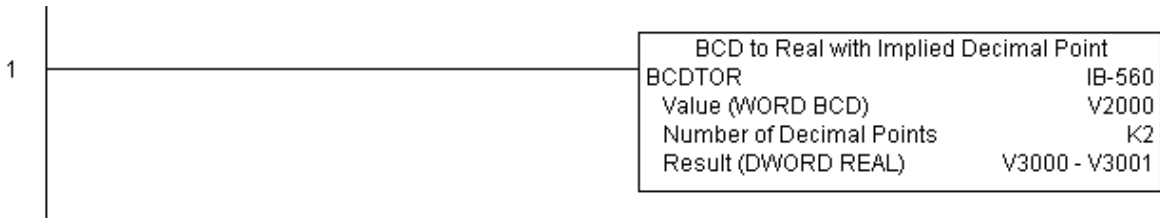
- **Value (WORD BCD):** Especifica la palabra o la constante que será convertida a un número real
- **Number of Decimal Points:** Especifica la cantidad de decimales en el resultado real
- **Result (DWORD REAL):** Especifica la localización en donde será colocado el número real

Parámetro	Rango del DL06
Value (WORD BCD) V,P,K	K0-9999; Vea el mapa de memoria V del DL06 - Data Words
Number of Decimal Points K	K0-4
Result (DWORD REAL) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de BCDTOR

En el ejemplo siguiente, la instrucción BCDTOR es usada para convertir los datos de 16 bits en V2000 desde un formato de datos de 4 dígitos BCD a un formato de datos real de 32bits (de coma flotante) y almacenado en V3000 y V3001.

K2 en la cantidad de decimales indica que tendrá dos dígitos a la derecha de la coma.

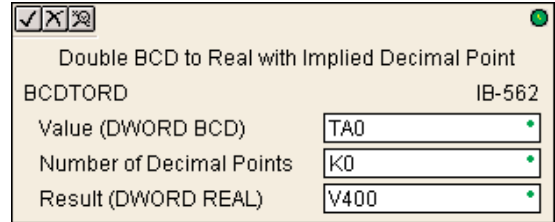


BCD doble a Real (BCDTORD) (IB-562)

Esta instrucción convierte un valor dado en una palabra doble BCD de 8 dígitos a un número Real, con una coma definida por la cantidad de decimales(K0-K8).

DS5	Usado
HPP	N/A

Por ejemplo, BCDTOR K12345678 con un número implicado de coma igual a K5, resultaría R123.45678



Parámetros de BCDTOR

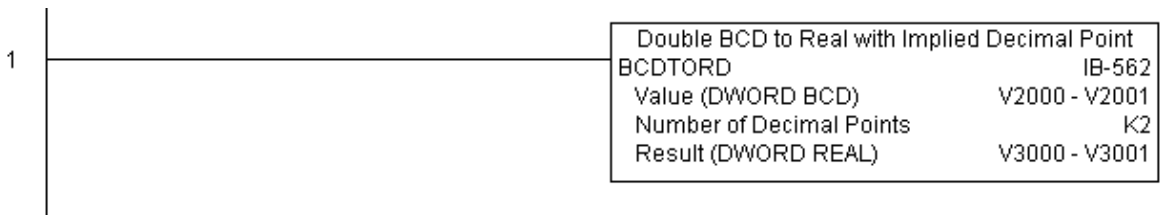
- **Value (WORD BCD):** Especifica la palabra doble o la constante que será convertida a un número real
- **Number of Decimal Points:** Especifica la cantidad de decimales en el resultado real
- **Result (DWORD REAL):** Especifica la localización en donde será colocado el número real

Parámetro	Rango del DL06
Value (DWORD BCD) V,P,K	K0-99999999; Vea el mapa de memoria V del DL06 - Data Words
Number of Decimal Points K	K0-8
Result (DWORD REAL) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de BCDTOR

En el ejemplo siguiente, la instrucción BCDTOR es usada para convertir los datos de 32 bits en V2000 desde un formato de datos de 8 dígitos BCD a un formato de datos real de 32bits (de coma flotante) y almacenado en V3000 y V3001.

K2 en la cantidad de decimales indica que tendrá dos dígitos a la derecha de la coma.

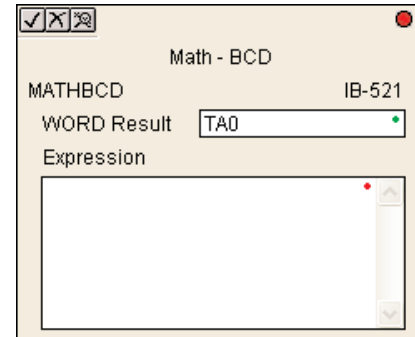


Math - BCD (MATHBCD) (IB-521)

DS5	Usado
HPP	N/A

La instrucción Math-BCD le permite crear expresiones matemáticas complejas tal como usted haría en los programa Visual Basic, EXCEL o C++ para hacer cálculos complejos, con paréntesis de hasta 4 niveles de profundidad.

Además de + - * /, usted puede hacer el modulo (% o resto), AND de bits(&),OR (|), XOR (^), y algunas funciones conBCD - convertir a BCD (BCD), a convertir a binario (BIN), complemento del BCD (BCDCPL), convertir desde código Gray (GRAY), invertir bits (INV) y BCD/HEX a exhibición de siete segmentos (SEG).



Ejemplo: $((V2000 + V2001) / (V2003 - K100)) * GRAY(V3000 \& K001F)$

Cada referencia de memoria V DEBE se usada como valor de formato BCD de una palabra. Los resultados intermedios pueden ir hasta valores de 32 bits, pero el resultado debe corresponder a una palabra BCD de 16 bits, para que el cálculo sea válido. Un ejemplo típico de esto es hacer escala usando multiplicación y luego división, $(V2000 * K1000)/K4095$. El resultado de la multiplicación excederá probablemente 9999 pero estará dentro de 32 bits. La operación de dividir dividirá 4095 en el acumulador de 32 bits, generando un resultado que quepa siempre en 16 bits.

Usted puede referirse a valores binarios de memoria V usando la función de conversión BCD en una posición de memoriaV pero NO una expresión. Ésto es, BCD(V2000) es correcto y convertirá V2000 de binario a BCD, pero la operación BCD(V2000 + V3000) sumará V2000 como BCD a V3000 como BCD, y luego interpretará el resultado como binario y lo convierte a BCD - NO SERÁ CORRECTO.

También, el resultado final es un número BCD de 16 bits y de modo que usted podría hacer la instrucción BIN en la operación completa para almacenar el resultado como binario.

Parámetros de MATHBCD

- **WORD Result:** Especifica la localización en donde el resultado en BCD de la expresión matemática será colocado (el resultado debe caber en una localización de memoria V de 16 bits)
- **Expression:** Especifica la expresión matemática que se ejecutará y el resultado se almacena en la memoria especificada WORD Result. Cada localización de memoria V usada en la expresión debe estar en formato BCD.

Parámetro	Rango del DL06
WORD Result V	Vea el mapa de memoria V del DL06 - Data Words
Expression	Text

Ejemplo de MATHBCD

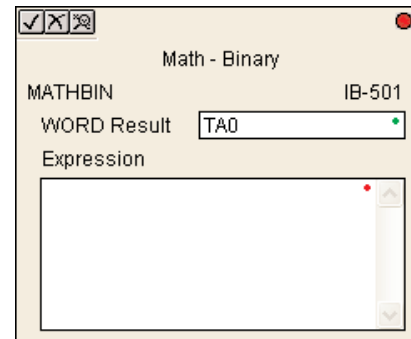
En el ejemplo siguiente, se usa la instrucción MATHBCD para calcular la expresión que multiplica el valor BCD en V1200 por 1000, después se divide por 4095 y carga el valor que resulta en V2000.



Math - Binaria (MATHBIN) (IB-501)

DS5	Usado
HPP	N/A

La instrucción Math-Binaria le permite crear expresiones matemáticas complejas como usted haría en los programa Visual Basic, EXCEL o C++ para hacer cálculos complejos, con paréntesis de hasta 4 niveles de profundidad. Además de + - * /, usted puede hacer el modulo (% o resto), AND de bits(&),OR (|), XOR (^), y algunas funciones conBCD - convertir a BCD (BCD), convertir a binario (BIN), decodificar bits(DECO), codificar bits (ENCO), invertir bits (INV) y HEX a exhibición de siete segmentos (SEG) y sumar bits (SUM).



5

Ejemplo: $((V2000 + V2001) / (V2003 - K100)) * SUM(V3000 \& K001F)$

Cada referencia de memoria V DEBE se usada como valor de formato binario de una palabra. Los resultados intermedios pueden ir hasta valores de 32 bits, pero el resultado debe corresponder a una palabra binaria de 16 bits, para que el cálculo sea válido. Un ejemplo típico de esto es hacer escala usando multiplicación y luego división, $(V2000 * K1000)/K4095$. El resultado de la multiplicación excederá probablemente 65535 pero estará dentro de 32 bits. La operación de dividir dividirá 4095 en el acumulador de 32 bits, generando un resultado que quepa siempre en 16 bits.

Usted puede referirse a valores BCD de memoria V usando la función de conversión BIN en una posición de memoriaV pero NO una expresión. Ésto es, BIN(V2000) es correcto y convertirá V2000 de BCD a binario, pero la operación BIN(V2000 + V3000) sumará V2000 como binario a V3000 como binario y luego interpretará el resultado como BCD y lo convierte a binario - NO SERÁ CORRECTO.

También, el resultado final es un número binario de 16 bits y de modo que usted podría aplicar la instrucción BCD en la operación completa para almacenar el resultado como BCD.

Parámetros de MATHBCD

- **WORD Result:** Especifica la localización en donde el resultado en binario de la expresión matemática será colocado (el resultado debe caber en una localización de memoria V de 16 bits)
- **Expression:** Especifica la expresión matemática que se ejecutará y el resultado se almacena en la memoria especificada WORD Result. Cada localización de memoria V usada en la expresión debe estar en formato binario.

Parámetro	Rango del DL06
WORD Result V	Vea el mapa de memoria V del DL06 - Data Words
Expression	Text

Ejemplo de MATHBIN

En el ejemplo siguiente, se usa la instrucción MATHBIN para calcular la expresión que multiplica el valor binario en V1200 por 1000, después se divide por 4095 y carga el valor que resulta en V2000.



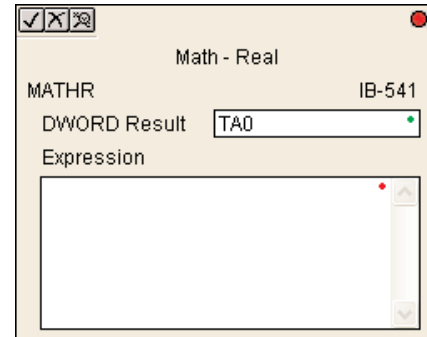
Math - Real (MATHR) (IB-541)

DS5	Usado
HPP	N/A

Esta instrucción le permite crear expresiones matemáticas complejas como usted haría en los programa Visual Basic, EXCEL o C++ para hacer cálculos complejos, con paréntesis de hasta 4 niveles de profundidad. Además de + - * /, usted puede hacer una instrucción And (&) Or (|) Xor (^) y muchas funciones Reales - Arco Coseno (ACOSR), ArcoSeno (ASINR), Arco Tangente (ATANR), Coseno (COSR), Convertir Radianes a Grados (DEGR), Invertir Bits (INV), Convertir grrados a Radianes (RADR), HEX a Seven Segment Display (SEG), Seno (SINR), Raíz cuadrada (SQRTR), Tangente (TANR).

Example: ((V2000 + V2002) / (V2004 - R2.5)) * SINR(RADR(V3000 / R10.0))

Cada referencia de memoria V DEBE caber en un valor ajustado a formato REAL de palabra doble.



Parámetros de MATHR

- **DWORD Result:** Especifica la localización en donde será colocado el resultado Real de la expresión matemática (el resultado debe caber en una localización ajustada a formato real de palabra doble)
- **Expression:** Especifica la expresión matemática que se ejecutará y el resultado se almacena en la localización especificada DWORD Result. Cada localización de memoria V en la expresión debe estar en formato Real.

Parámetro	Rango del DL06
DWORD Result V	Vea el mapa de memoria V del DL06 - Data Words
Expression	Text

Ejemplo de MATHR

En el ejemplo siguiente, se usa la instrucción MATHR para calcular la expresión que multiplica el valor REAL (de coma flotante) en V1200 por 10,5 y después se divide por 2,7 y se coloca el valor resultante de 32 bits en V2000 y V2001.



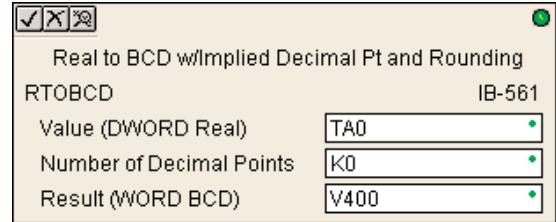
Real a BCD con redondeo (RTOBCD) (IB-561)

DS5	Usado
HPP	N/A

Esta instrucción convierte el valor absoluto de un número Real dado a un número en BCD de 4 dígitos, con la cantidad de decimales definidas en la instrucción (K0-K4) y además realiza el redondeo.

Por ejemplo, RTOBCD R56.74 con una cantidad de decimales igual a K1, resultaría BCD 567. Si el número implicado de comas fuera 0, entonces la función resultaría BCD 57 (note que redondeó para arriba).

Si el número Real es negativo, el resultado será igual al valor positivo absoluto.



Parámetros de RTOBCD

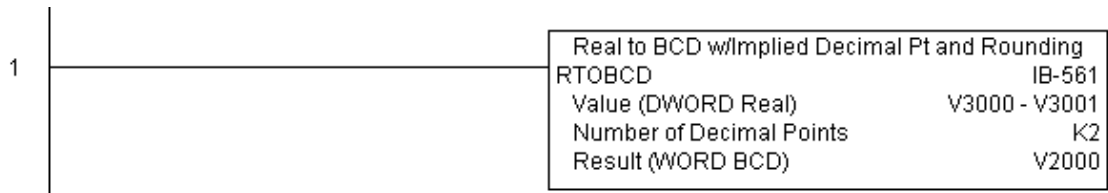
- **Value (DWORD Real):** Especifica la localización o el número Real que será convertido y redondeado a un número BCD con decimales implicados.
- **Number of Decimal Points:** Especifica el número de decimales implicados en el resultado Result WORD
- **Result (WORD BCD):** Especifica la localización en donde será almacenado el valor BCD que ha sido redondeado y transformado

Parámetro	Rango del DL06
Value (DWORD Real) V,P,R	R ; Vea el mapa de memoria V del DL06 - Data Words
Number of Decimal Points K	K0-4
Result (WORD BCD) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de RTOBCD

En el ejemplo siguiente, se usa la instrucción RTOBCD para convertir el formato de datos Real de 32 bits (de coma flotante) en V3000 y V3001 al formato de datos de 4 dígitos en BCD y almacenados en V2000.

K2 en el parámetro **Number of Decimal Points** (cantidad de decimales implicados) en los datos tendrá dos decimales implicados.



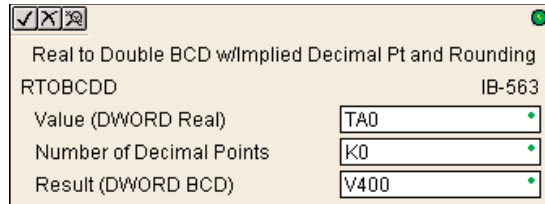
Real a BCD doble con redondeo (RTOBCDD) (IB-563)

Esta instrucción convierte el valor absoluto de un número Real dado a un número en BCD de 8 dígitos, con la cantidad de decimales definidas en la instrucción (K0-K8) y además realiza el redondeo.

DS5	Usado
HPP	N/A

Por ejemplo, RTOBCDD R38156.74 con una cantidad de decimales igual a K1, resultaría BCD 381567. Si el número implicado de comas fuera 0, entonces la función resultaría BCD 38157 (nota que redondeó para arriba).

Si el número Real es negativo, el resultado será igual al valor positivo absoluto.



Parámetros de RTOBCDD

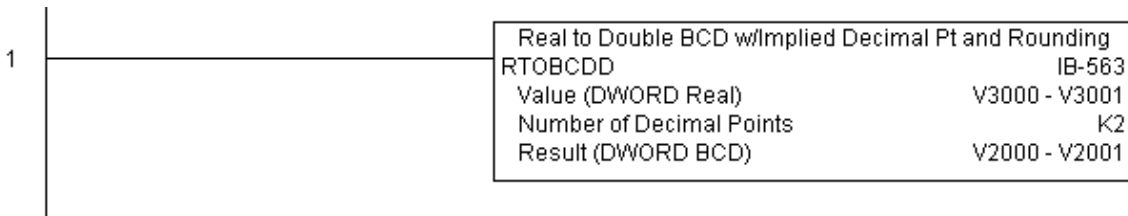
- **Value (DWORD Real):** Especifica la localización o el número Real que será convertido y redondeado a un número BCD con decimales implicados.
- **Number of Decimal Points:** Especifica el número de decimales implicados en el resultado Result WORD
- **Result (WORD BCD):** Especifica la localización en donde será almacenado el valor BCD que ha sido redondeado y transformado

Parámetro	Rango del DL06
Value (DWORD Real) V,P,R	R ; Vea el mapa de memoria V del DL06 - Data Words
Number of Decimal Points K	K0-8
Result (DWORD BCD) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de RTOBCDD

En el ejemplo siguiente, se usa la instrucción RTOBCDD para convertir el formato de datos Real de 32 bits (de coma flotante) en V3000 y V3001 al formato de datos de 8 dígitos en BCD y almacenados en V2000 y V2001.

K2 en el Number of Decimal Points (cantidad de decimales implicados) en los datos tendrá dos decimales implicados.



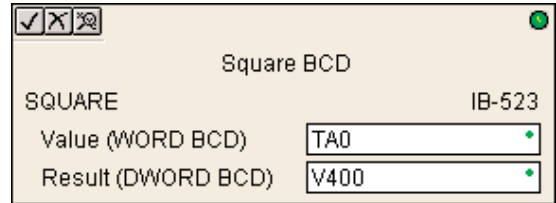
BCD al cuadrado (SQUARE) (IB-523)

DS5	Usado
HPP	N/A

Esta instrucción eleva al cuadrado el número de 4 dígitos dado en formato BCD y lo escribe como un resultado de 8 dígitos en formato BCD (palabra doble).

Parámetros de SQUARE

- **Value (WORD BCD):** Especifica la palabra o la constante en BCD que será elevada al cuadrado
- **Result (DWORD BCD):** Especifica la localización en donde será almacenado el valor al cuadrado de DWORD BCD

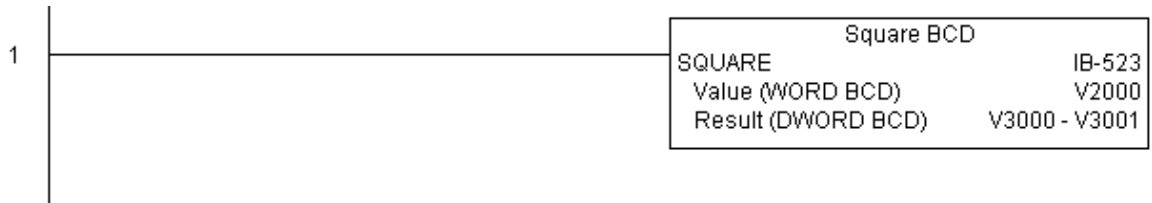


Parámetro	Rango del DL06
Value (WORD BCD) V,P,K	K0-9999 ; Vea el mapa de memoria V del DL06 - Data Words
Result (DWORD BCD) V	Vea el mapa de memoria V del DL06 - Data Words

5

Ejemplo de SQUARE

En el ejemplo siguiente, se usa la instrucción SQUARE para elevar al cuadrado el valor BCD de 4 dígitos en V2000 y para almacenar el resultado de 8 dígitos en V3000 y V3001



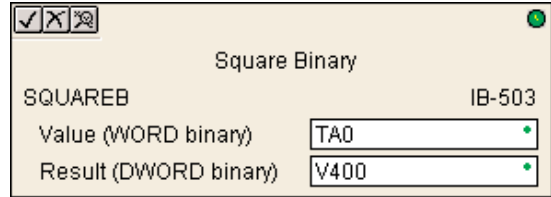
Binario al cuadrado (SQUAREB) (IB-503)

Esta instrucción eleva al cuadrado el número de 4 dígitos dado en formato binario y lo escribe como un resultado de 8 dígitos en formato binario.

DS5	Usado
HPP	N/A

Parámetros de SQUAREB

- **Value (WORD binary):** Especifica la palabra o la constante en FORMATO BINARIO que será elevada al cuadrado
- **Result (DWORD binary):** Especifica la localización en donde será almacenado el valor al cuadrado de la palabra doble en formato binario.

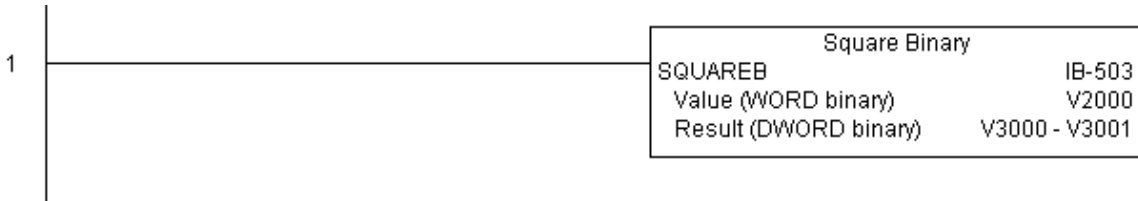


5

Parámetro	Rango del DL06
Value (WORD Binary) V,P,K	K0-65535; Vea el mapa de memoria V del DL06 - Data Words
Result (DWORD Binary) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de SQUAREB

En el ejemplo siguiente, se usa la instrucción SQUAREB para elevar al cuadrado el valor de formato binario de 4 dígitos en V2000 y para almacenar el resultado de 8 dígitos en V3000 y V3001



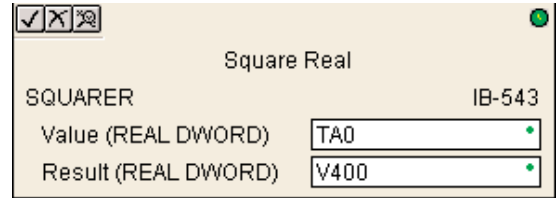
Real al cuadrado (SQUARER) (IB-543)

DS5	Usado
HPP	N/A

Esta instrucción eleva al cuadrado un número dado en formato Real y lo escribe como un resultado de 8 dígitos en formato Real.

Parámetros de SQUARER

- **Value (REAL DWORD)**: Especifica la palabra o la constante en formato Real que será elevada al cuadrado
- **Result (REAL DWORD)**: Especifica la localización en donde será almacenado el valor al cuadrado.

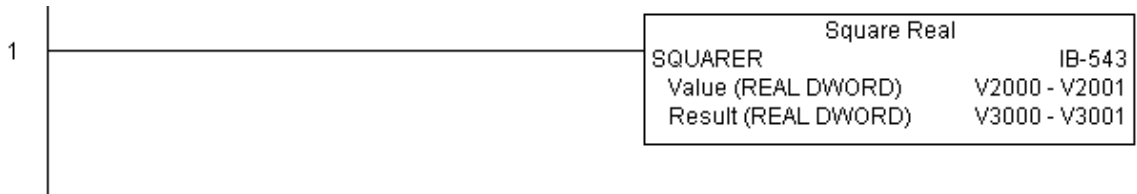


5

Parámetro	Rango del DL06
Value (REAL DWORD) V,P,R	R ; Vea el mapa de memoria V del DL06 - Data Words
Result (REAL DWORD) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de SQUARER

En el ejemplo siguiente, se usa la instrucción SQUARER para elevar al cuadrado el valor de coma flotante o valor REAL en V2000 y V2001 y para almacenar el resultado REAL en V3000 y V3001.

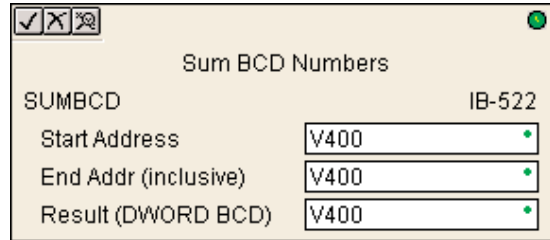


Suma de números BCD (SUMBCD) (IB-522)

DS5	Usado
HPP	N/A

Esta instrucción suma una lista de números de 4 dígitos consecutivos en formato BCD en un resultado de 8 dígitos BCD en una palabra doble.

Usted debe especificar las direcciones iniciales y finales del grupo de memoria V (incluyendo estas direcciones). Cuando está activada, esta instrucción sumará todos los números en el grupo (de modo que usted puede desear usar un contacto diferencial positivo como condición de la instrucción).



SUMBCD podría ser usado como la primera parte para calcular un promedio.

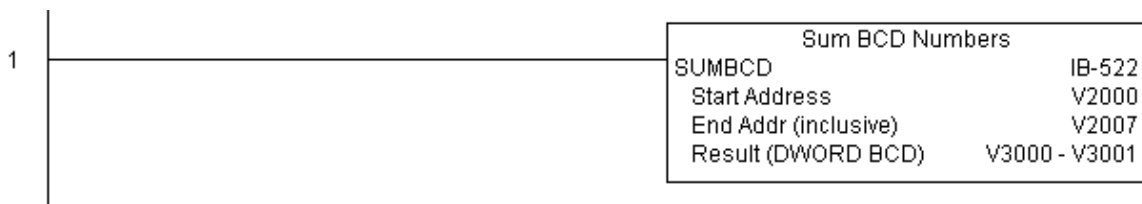
Parámetros de SUMBCD

- **Start Address:** Especifica la dirección inicial de un conjunto de valores consecutivos de localización de memoria V que se sumarán (BCD)
- **End Addr (inclusive):** Especifica la dirección final de un conjunto de valores consecutivos de localización de memoria V que se sumarán (BCD)
- **Result (DWORD BCD):** Especifica la localización de memoria V en donde será colocado el resultado de la suma del conjunto de valores consecutivos BCD

Parámetro	Rango del DL06
Start Address V	Vea el mapa de memoria V del DL06 - Data Words
End Address (inclusive) V	Vea el mapa de memoria V del DL06 - Data Words
Result (DWORD BCD) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de SUMBCD

En el ejemplo siguiente, se usa la instrucción de SUMBCD para sumar todos los valores BCD en las palabras V2000 hasta V2007 y para almacenar el resultado, que es un valor de 8 dígitos BCD, en V3000 y V3001.



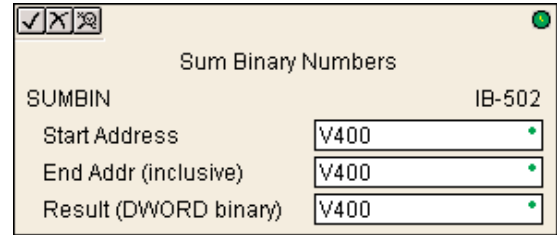
Suma de números Binarios (SUMBIN) (IB-502)

DS5	Usado
HPP	N/A

Esta instrucción suma una lista de números de 4 dígitos consecutivos en formato binario en un resultado de 8 dígitos binario, en una palabra doble.

Usted debe especificar las direcciones iniciales y finales del grupo de memoria V (incluyendo estas direcciones). Cuando está activada, esta instrucción sumará todos los números en el grupo (de modo que usted puede desear usar un contacto diferencial positivo como condición de la instrucción).

SUMBIN podría ser usado como la primera parte para calcular un promedio.



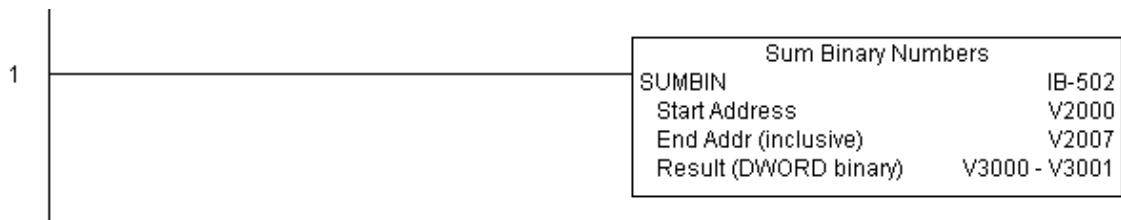
Parámetros de SUMBIN

- **Start Address:** Especifica la dirección inicial de un conjunto de valores consecutivos de localización de memoria V que se sumarán (Binarios)
- **End Addr (inclusive):** Especifica la dirección final de un conjunto de valores consecutivos de localización de memoria V que se sumarán (Binarios)
- **Result (DWORD Binary):** Especifica la localización de memoria V en donde será colocado el resultado de la suma del conjunto de valores consecutivos Binarios

Parámetro	Rango del DL06
Start Address V	Vea el mapa de memoria V del DL06 - Data Words
End Address (inclusive) V	Vea el mapa de memoria V del DL06 - Data Words
Result (DWORD Binary) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de SUMBIN

En el ejemplo siguiente, se usa la instrucción SUMBIN para sumar todos los valores binarios en las palabras V2000 hasta V2007 y para almacenar el resultado, que es un valor de 8 dígitos binarios, en V3000 y V3001.



Suma de números Reales (SUMR) (IB-542)

DS5	Usado
HPP	N/A

Esta instrucción suma una lista de números consecutivos en formato REAL en un resultado en una palabra doble de formato real o coma flotante.

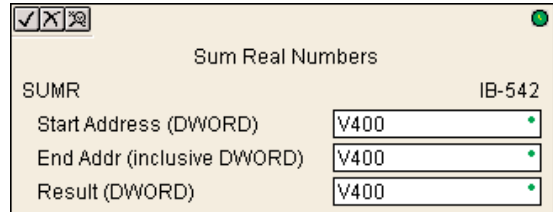
Usted debe especificar las direcciones iniciales y finales del grupo de memoriaV (incluyendo estas direcciones).

Recuerde que los números reales ocupan 2 palabras de memoria V cada uno, de modo que el número de valores reales sumados es igual a la mitad del número de las memorias.

Observe que la dirección final puede ser CUALQUIER palabra de la dirección final de 2 palabras, por ejemplo, si usted está sumando los 4 números reales almacenados en V2000 hasta V2007 (V2000, V2002, V2004, y V2006), usted puede especificar V2006 o V2007 como la dirección final y usted obtendrá el mismo resultado.

Cuando está activada, esta instrucción sumará todos los números en el grupo (de modo que usted puede desear usar un contacto diferencial positivo como condición de la instrucción).

SUMR podría ser usado como la primera parte para calcular un promedio.



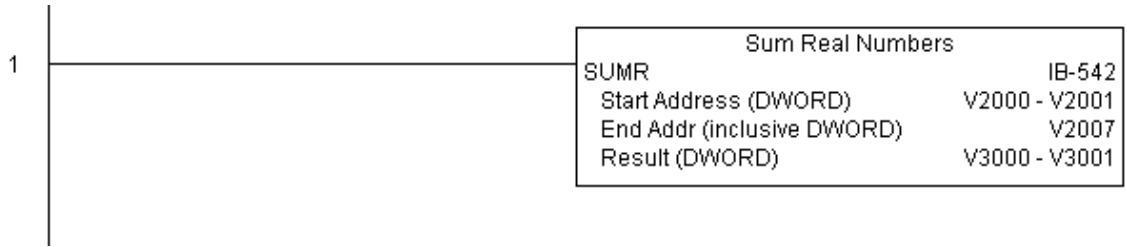
Parámetros de SUMR

- **Start Address(DWORD):** Especifica la dirección inicial de un conjunto de valores consecutivos de localización de memoria V que se sumarán (Real)
- **End Addr (inclusive DWORD):** Especifica la dirección final de un conjunto de valores consecutivos de localización de memoria V que se sumarán (Real)
- **Result (DWORD):** Especifica la localización de memoria V en donde será colocado el resultado de la suma del conjunto de valores consecutivos en formato de coma flotante.

Parámetro	Rango del DL06
Start Address (inclusive DWORD) V	Vea el mapa de memoria V del DL06 - Data Words
End Address (inclusive DWORD) V	Vea el mapa de memoria V del DL06 - Data Words
Result (DWORD) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de SUMR

En el ejemplo siguiente, se usa la instrucción SUMR para sumar todos los valores reales en las palabras V2000 hasta V2007 y para almacenar el resultado, que es un valor de coma flotante, en V3000 y V3001.

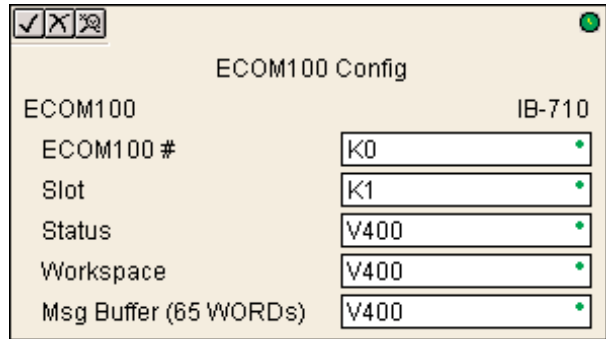


Configuración de ECOM100 (ECOM100) (IB-710)

DS5	Usado
HPP	N/A

La instrucción configuración ECOM100 define toda la información común para un módulo específico ECOM100 que sea utilizado por otro IBox ECOM100; por ejemplo, ECRX - leer la red con ECOM100, ECEMAIL - ECOM100 envía e-mail, ECIPSUP - configuración del IP ECOM100, etc.

Usted DEBE tener el IBox de configuración ECOM100 en la parte superior de su programa ladder o de etapas con cualquier otra configuración IBox. El parámetro Message buffer (almacenador intermediario de mensajes) especifica la dirección inicial de un almacenador intermediario (Message buffer) de 65 Palabras. Ésto es, 101 direcciones octales (por ejemplo. V1400 hasta V1500).



Si usted tiene más de un ECOM100 en su PLC, usted debe tener una configuración IBox ECOM100 diferente para CADA módulo ECOM100 en su sistema que use una instrucción IBox ECOM.

Los parámetros de **espacio de trabajo (Workspace)** y de estado (**Status**) y el almacenador intermediario de mensaje (Message buffer) son registros internos, privados, usados por la instrucción configuración de ECOM100 y DEBEN SER ÚNICOS en esta instrucción y NO SE DEBEN utilizar en cualquier otro lugar en su programa.

Para que funcione LA MAYORÍA DE IBoxes ECOM100, usted debe accionar el DIP switch 7 ON en la placa de circuito ECOM100. Usted puede mantener el DIP switch 7 apagado si usted está usando SOLAMENTE IBoxes ECOM100 ECRX o ECWX.

Parámetros de ECOM100

- **ECOM100#:** Este es un número lógico de identificación asociado a este módulo específico ECOM100 en la ranura especificada. El resto de IBoxes que necesitan referirse a este módulo ECOM100 debe referirse a este número lógico
- **Slot:** Especifica la ranura opcional que ocupa el módulo
- **Status:** Especifica una localización de memoria V que será usada por la instrucción
- **Workspace:** Especifica una localización de memoria V que será usada por la instrucción
- **Msg Buffer:** Especifica la dirección inicial de un almacenador intermediario de 65 palabras que será usado por el módulo para configuración

Parámetro	Rango del DL06
ECOM100# K	K0-255
Slot K	K1-4
Status V	Vea el mapa de memoria V del DL06 - Data Words
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Msg Buffer (65 words usado) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECOM100

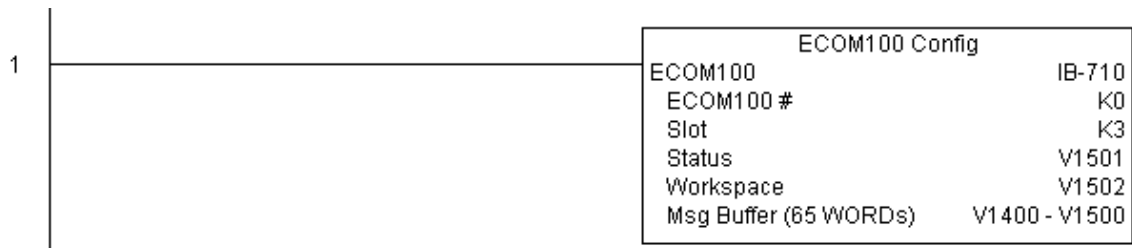
Esta instrucción coordina toda la interacción con otras instrucciones IBox relacionadas con ECOM100 (ECxxxx). Usted debe tener una instrucción de configuración IBox para cada módulo ECOM100 que esté instalado en su sistema. Estas instrucciones IBox de configuración de ECOM100 deben estar en la parte superior de su programa y se deben ejecutar en cada barrido.

Este IBox define que o módulo ECOM100# K0 está en la ranura 3. Cualquier instrucción IBox relacionada a ECOM100 que necesite referirse a este módulo específico (tal como ECEMAIL, ECRX...) debe tener la referencia K0 para el parámetro ECOM100 #.

El registro de estado (Status register) está diseñado para divulgar cualquier información de error o que ha sido completada la operación a otros IBoxes relacionados con ECOM100. Este registro de memoria V no debe ser usado en ningún otro lugar en el programa entero.

El registro del espacio de trabajo (Workspace register) es usado para mantener la información del estado sobre el ECOM100, junto con compartir adecuadamente y enclavarse con otros IBoxes realcionados con ECOM100 en el programa. Este registro de memoria V no debe ser usado en ningún otro lugar en el programa entero.

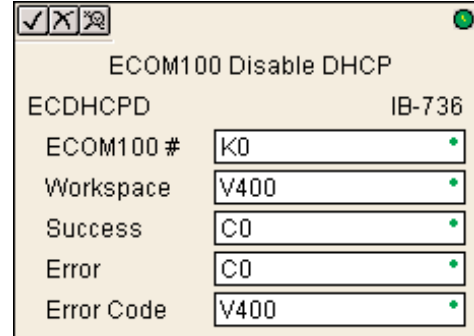
El almacenador intermediario de mensajes (Message Buffer) 65 palabras (130 bytes) es un grupo común de memoria que es usado por otros IBoxes relacionados con ECOM100 (tal como ECEMAIL). De esta manera, usted puede tener una cierta cantidad de IBoxes ECEMAIL, pero necesita solamente 1 almacenador intermediario común para generar y enviar cada email. Estos registro de memoria V no deben ser usados en ningún otro lugar en el programa entero.



Desabilitar DHCP en ECOM100 (ECDHCPD) (IB-736)

DS5	Usado
HPP	N/A

Esta instrucción configura el ECOM100 para utilizar las definiciones internas de TCP/IP en una transición de On para Off al control de este IBox. Para configurar las definiciones de TCP/IP del ECOM100 manualmente, puede usar la utilidad NetEdit3, o usted puede hacerla por el programa del PLC usando la instrucción Ibox de configuración de IP ECOM100 (ECIPSUP), o los IBoxes individuales: ECWRIP (Escribir dirección IP de ECOM), ECWRGWA (Escribir la dirección de gateway) y ECWRSNM (Escribir Subnet Mask).



El parámetro del espacio de trabajo es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en su programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000).

La configuración se almacena en la memoria Flash-ROM en el ECOM100 y la ejecución de este IBox inhabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escribe la memoria Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente este IBox UNA VEZ, en el primer barrido. Puesto que requiere una transición de Off para On , use un SPO NORMALMENTE CERRADO (NO se ejecuta en el primer barrido) para conducir el flujo de energía al IBox.

Para este ECOM100 IBox funcione, usted debe mover el DIP switch 7 en el circuito del módulo ECOM100.

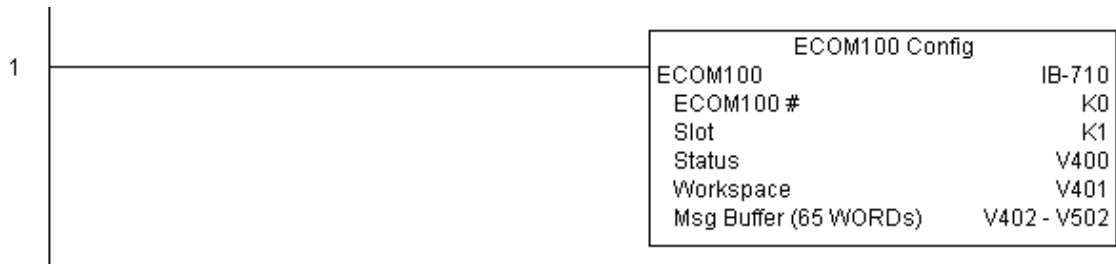
Parámetros de ECDHCPD

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que será usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la requisición se termine con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECDHCPD

Renglón 1: La instrucción ECDHCPD es responsable por la coordinación y enclavamiento de todos los Iboxes relacionados con ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100 de número K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita mover el módulo en la base a una ranura diferente, solamente necesita cambiar la información en este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx usando este módulo específico ECOM100. V401 es usado para coordinación y enclavamiento de la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario(buffer) de 130 bytes disponible para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100



Renglón 2: En el segundo barrido, deshabilita DHCP en el ECOM100. DHCP es el mismo protocolo usado por PCs para usar un servidor de DHCP para asignar automáticamente la dirección de IP (IP ADDRESS), la dirección de entrada y el subnet mask de los módulos ECOM100. Típicamente DHCP es deshabilitado asignando un IP ADDRESS HARD CODED con NetEdit3 o usando uno de los IBoxes de configuración de PI ECOM100, pero este IBox permite que usted deshabilite DHCP en el ECOM100 usando su programa ladder. El ECDHCPD es accionado en una transición de OFF para ON, no por una flujo de energía constante (similar a una entrada de un contador). El comando de deshabilitar DHCP será enviado al ECOM100 siempre que el flujo de energía en el IBox vaya de OFF a ON. Si se completa esta acción con éxito, se activa el bit C100. Si hay una falla, se activa el bit C101. Si falla, usted puede observar V2000 para conocer el código de error específico.

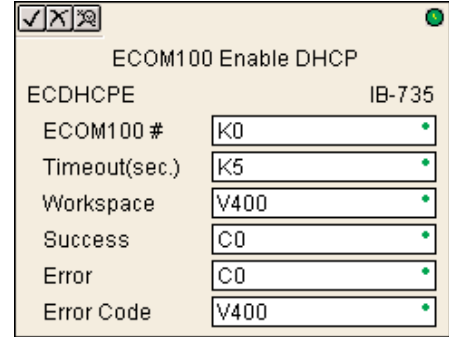


Habilitar DHCP en ECOM100 (ECDHCPE) (IB-735)

Esta instrucción le dirá el ECOM100 que obtenga su configuración de TCP/IP desde un servidor de DHCP en una transición de OFG para ON en el flujo de energía del IBox.

DS5	Usado
HPP	N/A

El IBox será colocado en estado exitoso una vez que el ECOM100 haya recibido las configuraciones de TCP/IP desde el servidor de DHCP. Puesto que es posible que el servidor de DHCP no sea asequible, se suministra un parámetro de atraso de tiempo (Timeout) de modo que el IBox pueda terminar, pero con un error (código de error = el decimal 1004).



Vea también el IBox 717 - configuración de IP del ECOM100 (ECIPSUP) - para configurar directamente TODOS LOS parámetros de TCP/IP en una sola instrucción - IP ADDRESS, subnet mask, y dirección de entrada.

The parámetro **Workspace** es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Uno de los parámetros de éxito(success) o de error (Error) se activará una vez que el comando se haya completado. Si hay un error, el parámetro del código de error divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000).

La configuración "Habilitar DHCP" se almacena en memoria Flash-ROM en el ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos 0,5 segundo hasta que escribe la Flash-ROM. Por lo tanto, SE RECOMIENDA que se ejecute solamente este IBox UNA VEZ, en el primer barrido. Puesto que requiere una transición de OFF para ON para ejecutarse, use un SP0 NORMALMENTE CERRADO para controlar el flujo de energía al IBox.

Para que este IBox funcione, se debe colocar el DIP switch 7 a ON en el circuito del módulo ECOM100.

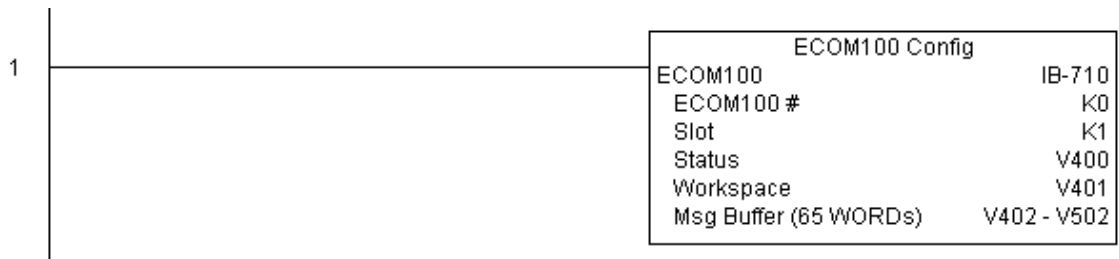
Parámetros de ECDHCPE

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Timeout(sec):** Especifica un período de atraso de modo que la instrucción se complete
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la petición no se ha completado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error

Parámetro	Rango del DL06
ECOM100# K	K0-255
Timeout (sec) K	K5-127
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECDHCPE

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rólulot en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



5

Renglón 2: En el segundo barrido, se habilita DHCP en el ECOM100. DHCP es el mismo protocolo usado en PCs para usar un servidor de DHCP para asignar automáticamente el IP ADDRESS , la dirección de la entrada, y el subnet mask del módulo ECOM100. Esto se hace típicamente usando NetEdit3, pero este IBox permite que usted habilite DHCP en el ECOM100 usando el programa ladder. La instrucción ECDHCPE es accionada por una transición de OFF para ON, y entonces no es controlado por flujo de energía (Por ejemplo, tal como una entrada de contador). Los comandos para habilitar DHCP serán enviados al ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO. La instrucción ECDHCPE no hace más que activar el bit para activar DHCP en el ECOM100, y luego interroga el ECOM100 una vez cada segundo para ver si el ECOM100 ha encontrado un servidor de DHCP y tiene un IP ADDRESS válido. Por lo tanto, es necesario un parámetro de **timeout** en caso de que el ECOM100 no pueda encontrar un servidor de DHCP. Si ocurre un **timeout**, es decir, no encuentra el servidor después de un tiempo, el bit del error se activará y el código de error será el decimal 1005. El bit de éxito(Success) se activará solamente si el ECOM100 encuentra un servidor de DHCP y le asigna un IP ADDRESS válido. Si se encuentra el servidor, el bit C100 se activa. Si hay una falla, se activa el bit C101. Si hay un error, usted puede inspeccionar V2000 para ver el código de error específico.



Configuración de ECOM100 Query DHCP (ECDHCPQ) (IB-734)

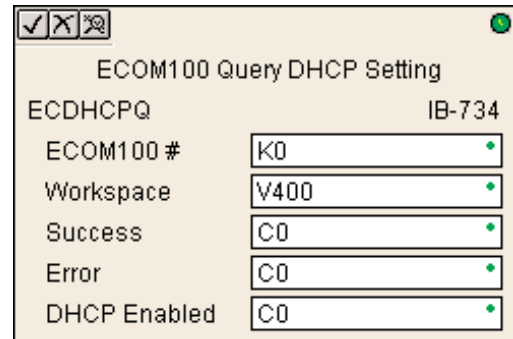
DS5	Usado
HPP	N/A

Esta instrucción determinará si DHCP está habilitado en el módulo ECOM100 en una transición desde OFF para ON a este IBox. El bit del parámetro DHCP Enabled estará ENCENDIDO si DHCP está habilitado o APAGADO si está deshabilitado.

El parámetro del espacio de trabajo (Workspace) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta instrucción y NO DEBE ser usado en ningún otro lugar en el programa.

El parámetro del bit de éxito(Success) o de error (Error) se activará cuando el comando se haya completado.

Para este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición en el circuito de ECOM100.



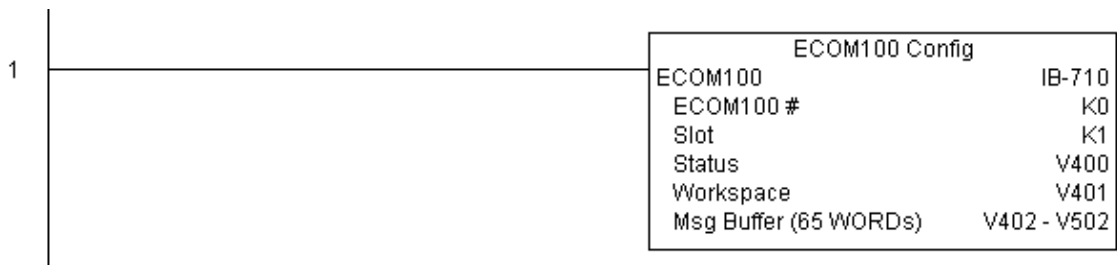
Parámetros de ECDHCPQ

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** especifica a bit that will turn on once the instrucción is completed successfully
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito y completed
- **DHCP Enabled:** Especifica un bit que se activa si el DHCP de módulo ECOM100 se habilita o sigue estando apagado si está deshabilitado - después de que una petición a la instrucción, asegúrese verificar el estado del bit de Success o de Error junto con estado del bit DHCP Enabled para confirmar una petición acertada al módulo

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
DHCP Enabled X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de ECDHCPQ

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido del PLC, lee si DHCP está habilitado o deshabilitado en el ECOM100 y lo almacena en C5. DHCP es el mismo protocolo usado por PCs para usar un servidor de DHCP para asignar automáticamente el IP ADDRESS de módulos ECOM100, la dirección del Gateway, y el subnet mask. La instrucción ECDHCPQ es activada por una transición de OFF para ON, (similar a una entrada de un contador). El comando de leer (Query) si DHCP está habilitado o no será enviado al ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO. Si es hecho con éxito, el bit C100 se activa. Si hay una falla, se activa el bit C101.



Enviar E-mail con ECOM100 (ECEMAIL) (IB-711)

DS5	Usado
HPP	N/A

Esta instrucción, se comportará como cliente de correo electrónico (E-mail) y enviará una petición de SMPT a su servidor de SMTP para enviar un mensaje de e-mail a las direcciones de e-mail en el campo **To:** y también a éstos enumerados en cc:, lista definida explícitamente en la instrucción ECCEMAIL, cuando haya una transición de OFF para ON. Enviará la petición de SMTP basada en el número de ECOM100 especificado, que corresponde a una configuración única específica ECOM100 en la parte superior de su programa.

El campo **Body** (Contenido) contiene lo mismo que las instrucciones PRINT y VPRINT para texto y variables embutidas, permitiendo que usted embuta datos en tiempo real en su email (por ejemplo. "V2000 =" V2000:B).

El parámetro **Workspace** (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta instrucción y NO DEBE ser usado en cualquier otro lugar en su programa. O el parámetro de éxito o de error se activará una vez que la petición se haya completado. Si hay un error, el parámetro del código de error divulgará un código de error ECOM100 (menos de 100), un error del protocolo de SMPT (entre 100 y 999), o un error de lógica del PLC (mayor de 1000).

Puesto que el ECOM100 es solamente un cliente de e-mail y requiere el acceso a un servidor de SMTP, usted DEBE tener los parámetros de SMTP configurados correctamente en el ECOM100 vía el Home Page y/o la instrucción de configuración del e-mail (ECEMSUP) de ECOM100's. Para obtener al Home Page del módulo ECOM100, use su browser preferido de Internet y navegue hasta el IP ADDRESS del módulo ECOM100, por ejemplo. <http://192.168.12.86>

Hay una limitación de hasta aproximadamente 100 caracteres en los datos de mensaje para la instrucción entera, incluyendo los campos **Subject:** y **Body:**. Para ahorrar espacio, el módulo ECOM100 le permite tener una lista explícita de direcciones de e-mail en el campo de copia a carbón (cc:) de modo que usted pueda configurar éstos en el módulo ECOM100, y mantenga el campo **To:** pequeño (o aún vacío), para dejar mas espacio para los campos **Subject:** y **Body:**.

Para que este IBox de ECOM100 funcione, usted debe mover el DIP Switch 7 a ON en el circuito del módulo ECOM100.

Parámetros de ECEMAIL

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que deben referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la petición no se ha completado con éxito

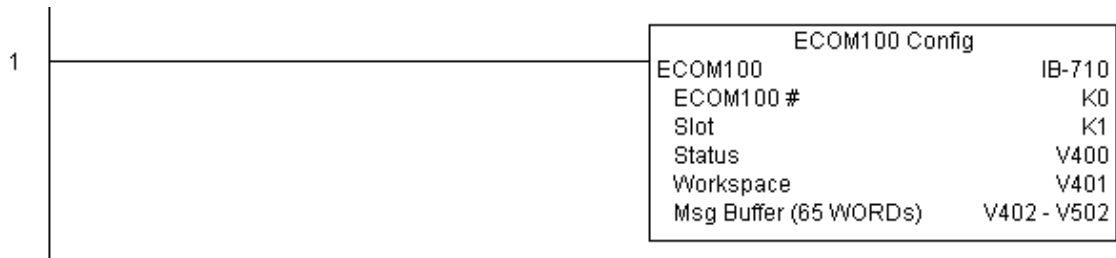
The screenshot shows a configuration window titled "ECOM100 Send EMail" with the identifier "IB-711". It features several dropdown menus and text input fields. The fields are: "ECOM100 #" with value "K0", "Workspace" with value "V400", "Success" with value "C0", "Error" with value "C0", "Error Code" with value "V400", "To", "Subject", and "Body". The "Body" field is a large, empty text area at the bottom of the window.

- **Error Code:** Especifica la localización en donde será escrito el código de error
- **To:** Especifica la dirección de E-mail donde será enviado el mensaje
- **Subject:** Asunto del mensaje de E-mail
- **Body:** Mensaje con datos que son iguales a los que pueden tener las instrucciones PRINT y VPRINT para texto y variables embutidas, permitiendo que usted embuta datos en tiempo real en el mensaje del e-mail

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria DL06
To:.....	Text
Subject:.....	Text
Body:.....	See PRINT and VPRINT instrucciones

Ejemplo de ECEMAIL

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100 K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



(Este ejemplo continúa en la próxima página)

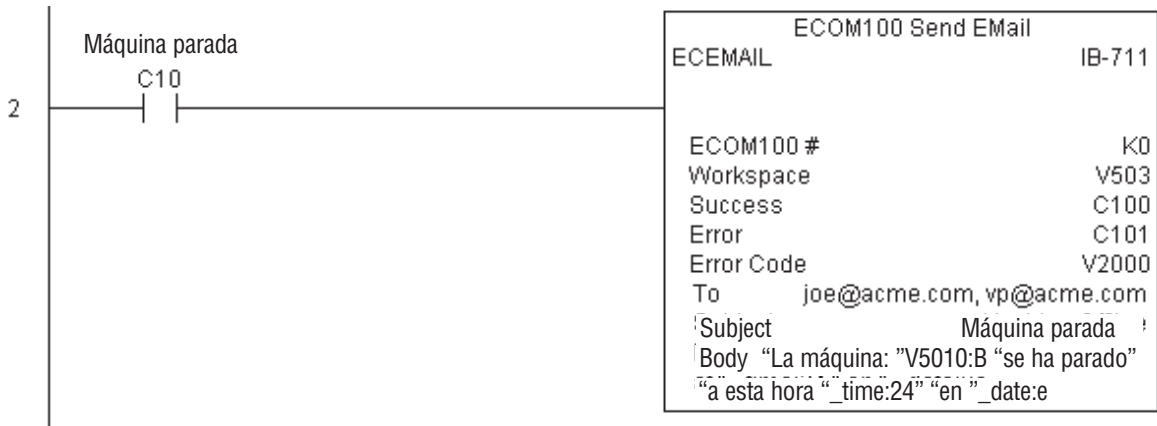
Ejemplo de ECEMAIL (continuado)

Renglón 2: Cuando se para una máquina, envíe un e-mail a Joe (joe@acme.com) en el departamento de mantención y al gerente de producción (vp@acme.com) informando que máquina está parada junto con la fecha y hora cuando la máquina se paró.

El ECEMAIL es accionado con una transición de OFF para ON, no necesita de un flujo de energía constante (similar a una entrada de un contador cuando cuenta una vez). Será enviado un email siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO. Esto ayuda a prevenir que se envíen correos electrónicos continuamente.

Si se envía el email, se activa el bit C100. Si hay una falla, se activa el bit C101. Si falla, usted puede observar V2000 para ver el código de error de SMTP u otros códigos de error posibles.

5

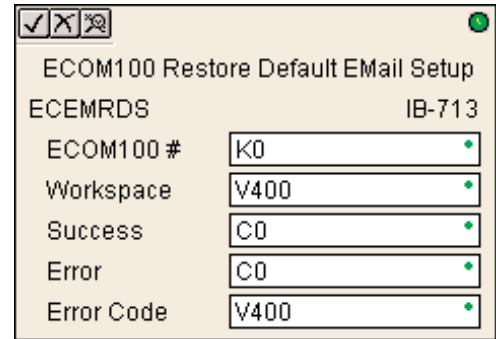


ECOM100 Restore Default E-mail Setup (ECEMRDS) (IB-713)

DS5	Usado
HPP	N/A

Esta instrucción de restaurar la configuración del e-mail por defecto ECOM100, en una transición de OFF para ON, restaurará los datos originales de configuración del e-mail almacenados en el ECOM100 de nuevo a la copia de trabajo basada en ECOM100 # especificado, que corresponde a una configuración única específica de ECOM100 (ECOM100) en la parte superior de su programa.

Cuando el ECOM100 es energizado, copia los datos de la configuración del e-mail almacenados en la ROM a la copia de trabajo en RAM. Ud. puede entonces modificar esta copia de trabajo del programa usando el IBox de configuración del e-mail ECOM100 (ECEMSUP). Después de modificar la copia de trabajo, usted puede restaurar más adelante los datos originales de la configuración con el programa usando este IBox.



El parámetro del espacio de trabajo es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta instrucción y NO DEBE ser usado en ningún otro lugar en su programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000).

Para este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

Parámetros de ECEMRDS

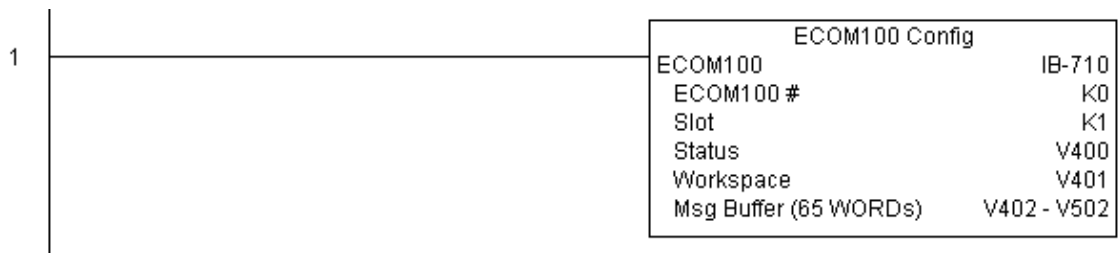
- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECEMRDS

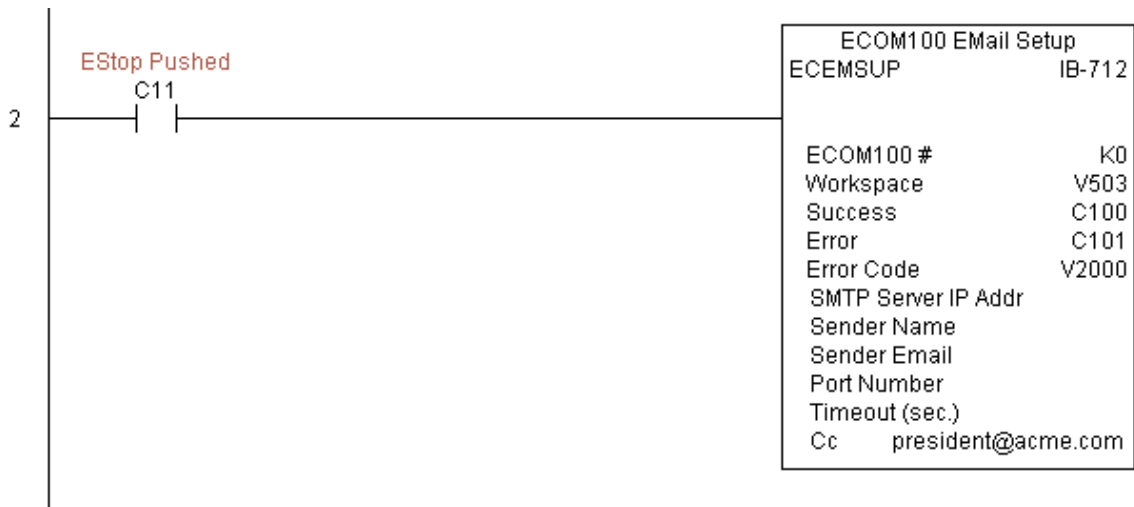
Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100 K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.

5



Renglón 2: Siempre que se empuje una parada de emergencia, asegúrese que el presidente de la compañía reciba copias de todos los e-mails que son enviados.

La instrucción IBox de configuración de una e-mail de ECOM100 le permite definir o cambiar los datos de configuración de SMTP para uso con e-mail almacenados en el módulo ECOM100.

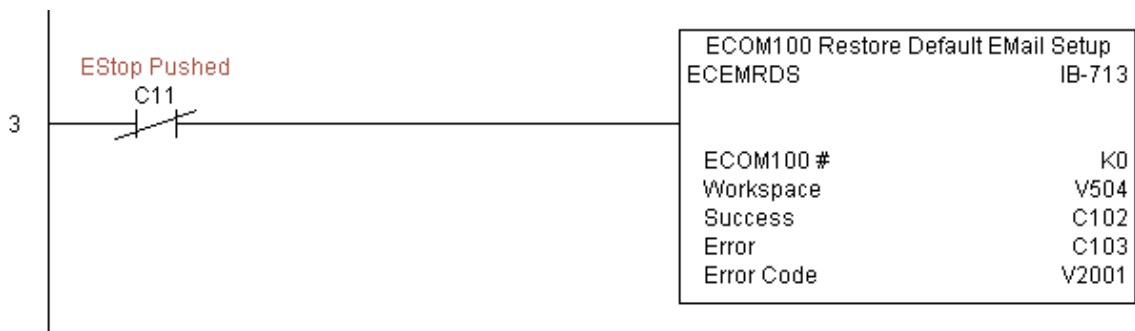


Ejemplo de ECEMRDS (continuado)

Renglón 3: Una vez que se rearme la parada de emergencia, retire al presidente de la lista cc: restaurando la configuración de email por defecto en el módulo ECOM100.

La instrucción ECEMRDS se acciona en la transición de OFF para ON , (similar a la entrada de conteo de un contador). La configuración del e- mail almacenada en la memoria ROM del módulo ECOM100 será copiada sobre la "copia de trabajo" siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO (la copia de trabajo puede ser cambiada usando la instrucción IBox ECEMSUP).

Si funciona correctamente, se activa el bit C102. Si hay una falla, se activa C103. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.



Configuración de E-mail con ECOM100 (ECEMSUP) (IB-712)

DS5	Usado
HPP	N/A

Esta instrucción, en una transición de OFF para ON, modificará la copia de trabajo de la configuración de e-mail actual en el ECOM100 basado en el número especificado de ECOM100, que corresponde a una configuración única específica ECOM100 (ECOM100) en la parte superior de su programa.

Usted puede escoger y elegir cualquiera o todos los campos que se modificarán usando esta instrucción. Observe que estos cambios son acumulativos: Si usted se ejecuta IBoxes múltiples de configuración de e-mail de ECOM100, después todos los cambios se realizan en la orden que se ejecutan. También observe que usted puede restaurar la configuración del e-mail original ECOM100 que es almacenado en el ECOM100 a la copia de trabajo usando el IBox de restaurar la configuración del email por defecto ECOM100 (ECEMRDS).

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

El parámetro del bit de éxito (Success) o de error se activará una vez que el comando se haya completado. Si hay un error, el parámetro del código de error divulgará un código de error ECOM100 (menos de 100), o un error de la lógica del PLC (mayor de 1000).

Hay un límite de aproximadamente 100 caracteres/bytes de datos de configuración para la instrucción entera. Si fuera necesario, usted podría dividir la configuración entera con IBoxes múltiples ECEMSUP sobre una base de campo por campo, por ejemplo, hace el campo copia a carbón (cc:) en un IBox ECEMSUP y los parámetros restantes de configuración en otro.

Para que este ECOM100 IBox funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

Parámetros de ECEMSUP

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **SMTP Server IP Addr:** Parámetro opcional que especifica el IP ADDRESS del servidor del SMTP en la red de módulos ECOM100
- **Sender Name:** Parámetro opcional que especifica el nombre del remitente el cual aparecerá en el campo "From:" a los que reciben el E-mail
- **Sender EMail:** Parámetro opcional que especifica la dirección del e-mail del remitente de la cual aparecerá en el campo "From:" a los que reciben el E-mail

Parámetros de ECEMSUP (continuado)

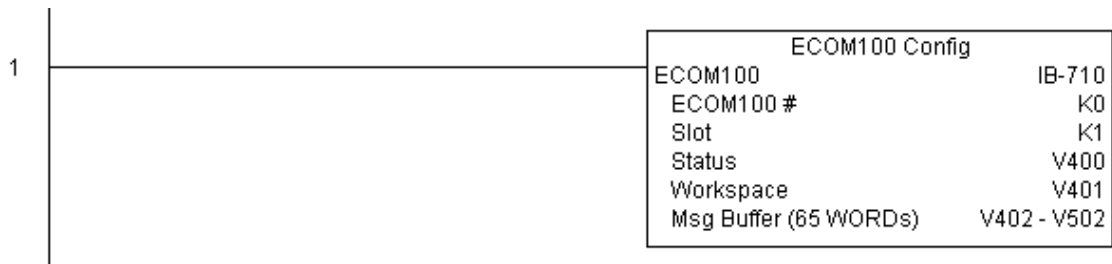
- **Port Number:** Parámetro opcional que especifica the TCP/IP Port Number to send SMTP requests; usually this does not to be configured (see your network administrator for information on this setting)
- **Timeout (sec):** Parámetro opcional que especifica la cantidad de segundos que debe esperar para que el servidor SMTP envíe el e-mail a todos los que reciben éste.
- **Cc:** Parámetro opcional que especifica una lista de direcciones de e-mail “carbon copy” que será enviadas

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECEMSUP

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinatar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.

5



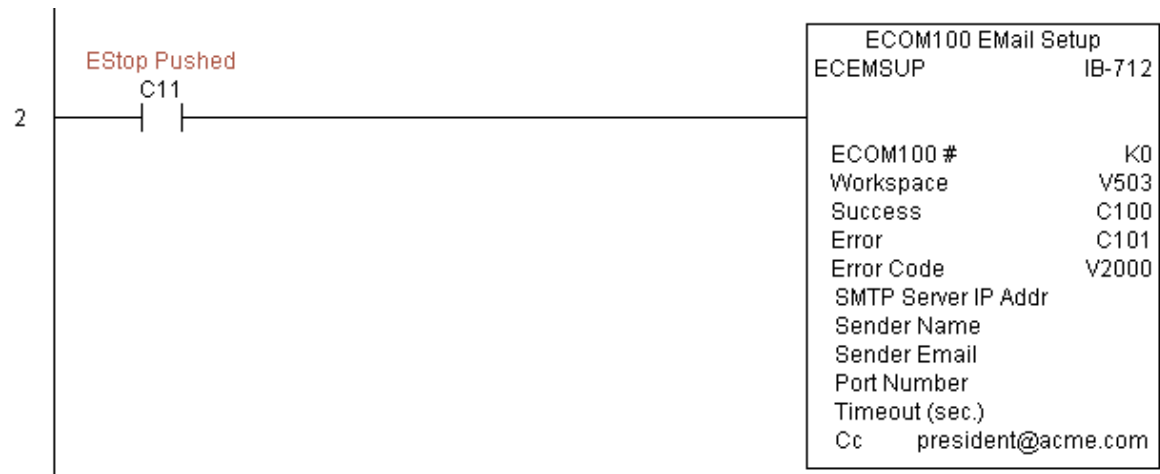
(Este ejemplo continúa en la próxima página)

Ejemplo de ECEMSUP (continuado)

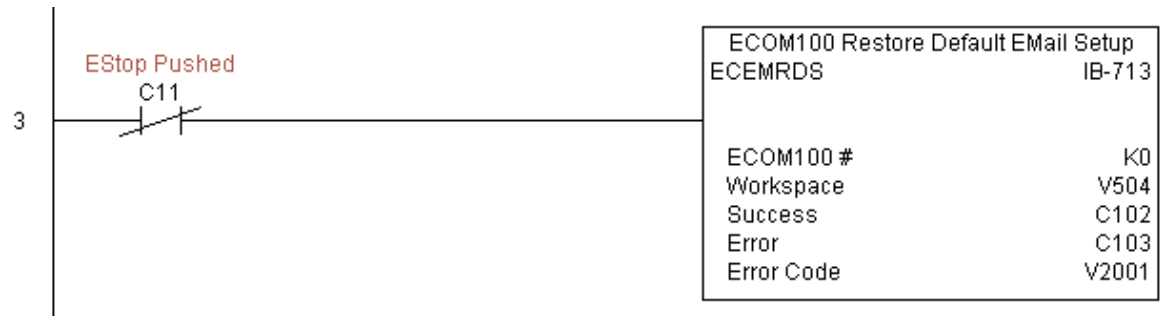
Renglón 2: Siempre que se empuje un botón de parada de emergencia, asegúrese de que el presidente de la compañía consiga las copias de todos los email. La instrucción IBox de configuración de email con ECOM100 le permite definir o cambiar los parámetros de smtp de un e-mail almacenados en el ECOM100. La instrucción ECEMSUP es activada en la transición de OFF para ON (similar a una entrada de un contador). En la energización del PLC, la configuración del e-mail almacenada en la memoria ROM del ECOM100 se copia a una memoria RAM "copia de trabajo". Usted puede cambiar esta copia de trabajo usando el IBox ECEMSUP. Para restaurar la configuración original en la memoria ROM, use la instrucción IBox ECEMRDS.

Si funciona correctamente, se activa el bit C102. Si hay una falla, se activa el bit C103. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.

5



Renglón 3: Una vez que se remueva la condición de parada de emergencia, saque el presidente de la lista en cc: restaurando la configuración de e-mail por defecto en el módulo ECOM100.



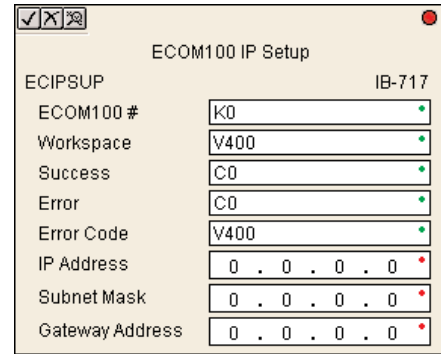
Configuración de PI de ECOM100 (ECIPSUP) (IB-717)

DS5	Usado
HPP	N/A

Esta instrucción configurará los tres parámetros de TCP/IP en el ECOM100: IP ADDRESS, subnet mask, y dirección de Gateway, en una transición desde OFF para ON para activar el IBox. El ECOM100 es especificado por el número ECOM100 #, que corresponde a un IBox de configuración única (ECOM100) en la parte superior de su programa.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000).



Estos datos de configuración se almacenan en memoria Flash-ROM en el ECOM100 e deshabilitarán el módulo ECOM100 por lo menos un medio segundo hasta que se escribe la memoria Flash-ROM. Por lo tanto, SE RECOMIENDA ALTAMENTE que usted ejecute solamente este IBox UNA VEZ en el primer barrido. Ya que se requiere una transición de OFF para ON, use un SP0 NORMALMENTE CERRADO para ejecutar al IBox.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

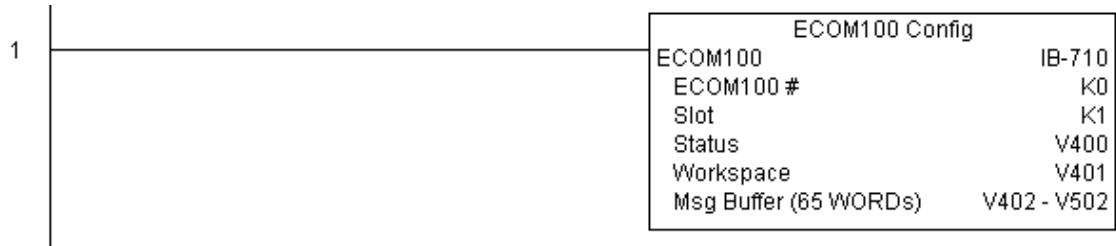
Parámetros de ECIPSUP

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito y completed
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **IP Address:** Especifica la dirección de IP del módulo
- **Subnet Mask:** Especifica la Subnet Mask para el módulo en cuestión
- **Gateway Address:** Especifica la dirección de Gateway para el módulo en cuestión

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words
IP Address IP Address	0.0.0.1. to 255.255.255.254
Subnet Mask Address IP Address Mask	0.0.0.1. to 255.255.255.254
Gateway Address IP Address	0.0.0.1. to 255.255.255.254

Ejemplo de ECIPSUP

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



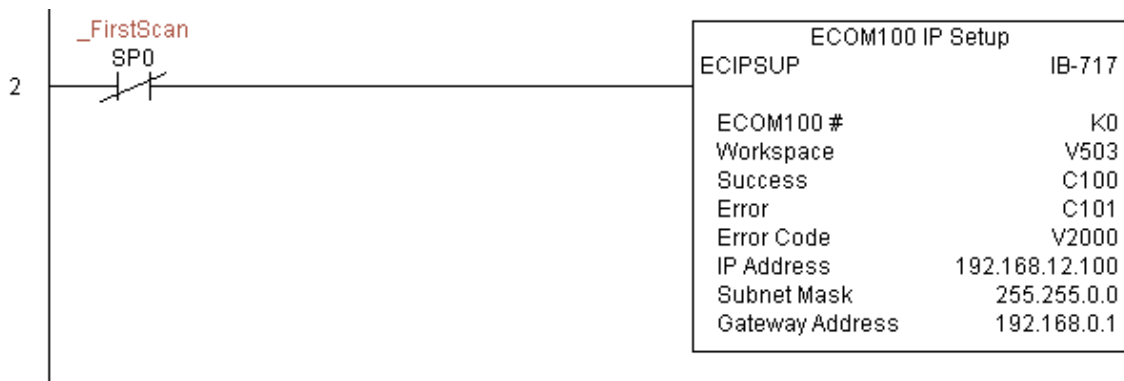
5

Renglón 2: En el segundo barrido, configure todos los parámetros de TCP/IP en el ECOM100:

IP Address: 192.168. 12.100
 Subnet Mask: 255.255. 0. 0
 Dirección de Gateway: 192.168. 0. 1

La instrucción ECIPSUP es accionada en una transición de OFF para ON, (similar a una entrada de un contador). El comando de escribir los parámetros de la configuración de TCP/IP será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C102. Si hay una falla, se activa C103. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.



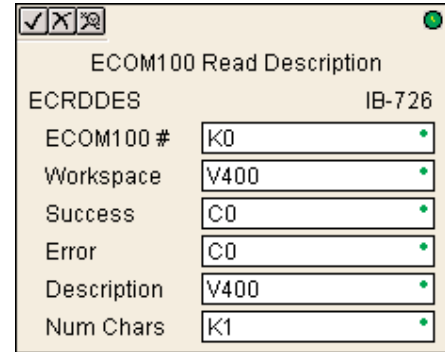
Lea la descripción del ECOM100 (ECRDDES) (IB-726)

DS5	Usado
HPP	N/A

Esta instrucción leerá el campo de descripción del módulo ECOM100 hasta el número de caracteres especificados en una transición de OFF para ON al IBox.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa. MUST BE UNIQUE in this one instrucción and MUST NOT be usado anywhere else in your program.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000).ete.



Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición en el circuito de ECOM100. ECOM100 circuit board.

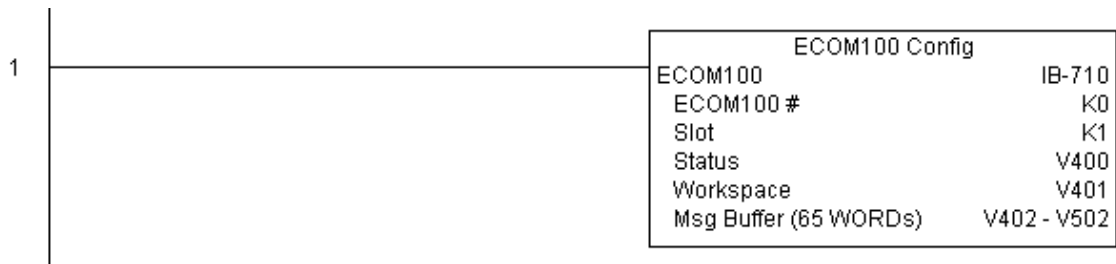
Parámetros de ECRDDES

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico cified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Descripción:** Especifica la dirección de memoria V para el almacenamiento donde el nombre del módulo será colocado
- **Num Char:** Especifica la cantidad de caracteres (bytes) a ser leídos desde el campo Descripción del módulo ECOM100

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Descripción V	Vea el mapa de memoria V del DL06 - Data Words
Num Chars..... K	K1-128

Ejemplo de ECRDDES

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



5

Renglón 2: En el segundo barrido , lee la descripción del módulo del módulo ECOM100 y la almacena en las memorias V3000 hasta V3007 (16 caracteres). Este texto se puede exhibir en una interface de operador, por ejemplo.

La instrucción ECRDDES es activada en una transición de OFF para ON, (similar a una entrada de un contador). El comando de leer la descripción del módulo será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa C101.



Lea la dirección Gateway del ECOM100 (ECRDGWA) (IB-730)

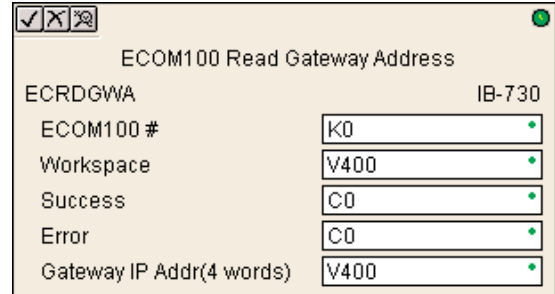
DS5	Usado
HPP	N/A

Esta instrucción leerá las 4 partes de la dirección IP del Gateway y las almacenará en 4 posiciones de memoria V consecutivas en formato decimal, en una transición desde OFF para ON para activar este IBox.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000).ete.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.



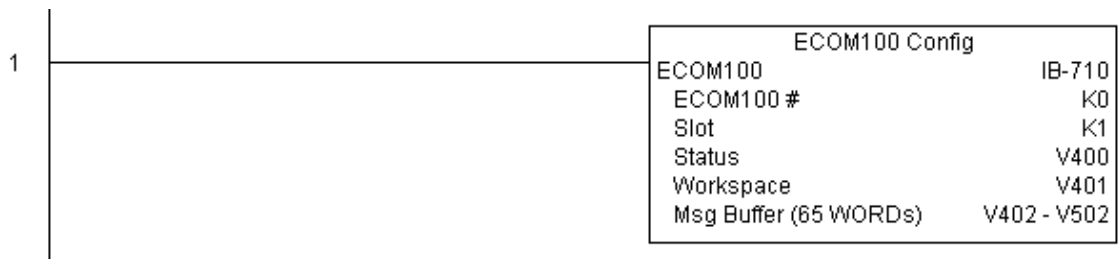
Parámetros de ECRDGWA

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico cified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Gateway IP Addr:** Especifica la dirección inicial donde la dirección del Gateway del módulo ECOM100 será colocada en 4 localizaciones consecutivas de memoria V

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Gateway IP Address (4 Words)..... V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECRDGWA

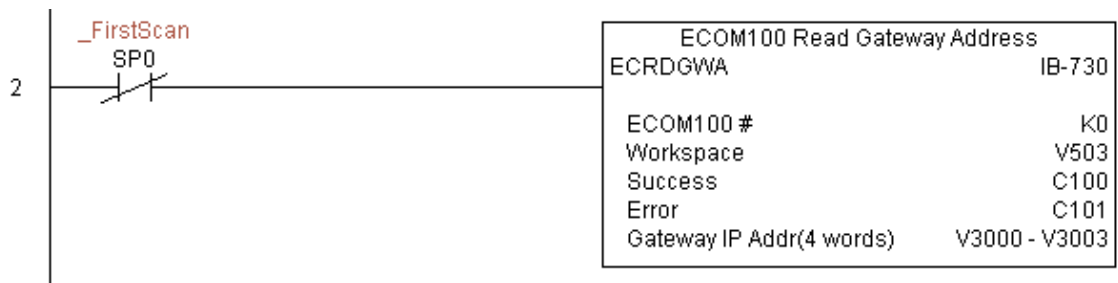
Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido , lee la dirección Gateway del módulo ECOM100 y la almacena en V3000 hasta V3003 (4 númerosdecimales). Esta información se puede exhibir en una interface de operador, por ejemplo.

La instrucción ECRDGWA es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de leer la dirección de Gateway del módulo será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa C101.



ECOM100 Read IP Address (ECDIP) (IB-722)

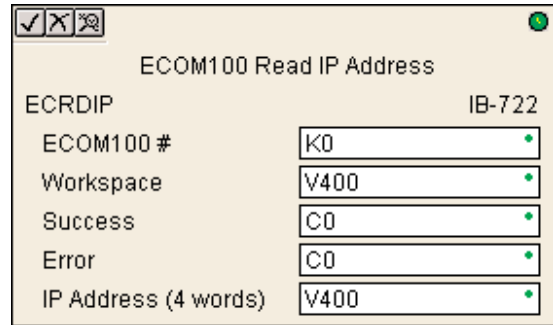
DS5	Usado
HPP	N/A

ECOM100 Read IP Address will read the 4 parts of the IP address and store them in 4 consecutive V Memory locations in decimal format, on a transition from OFF to ON to the IBox.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición en el circuito de ECOM100. ECOM100 circuit board.



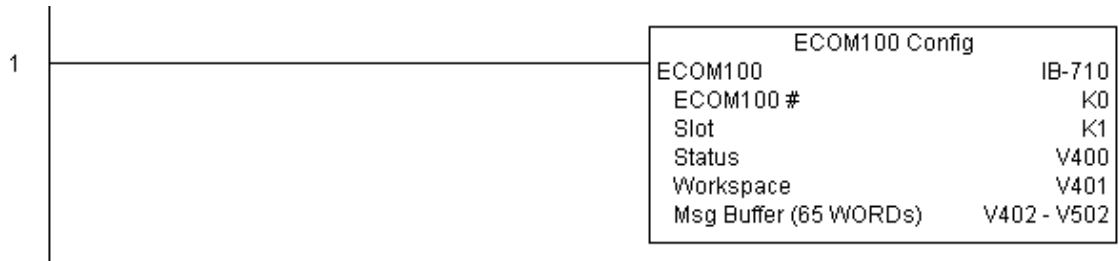
Parámetros de ECRDIP

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico cified slot.
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **IP Address:** Especifica la dirección inicial donde será colocada la dirección de IP del módulo ECOM100, en 4 localizaciones consecutivas de memoria V

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
IP Address (4 Words) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECRDIP

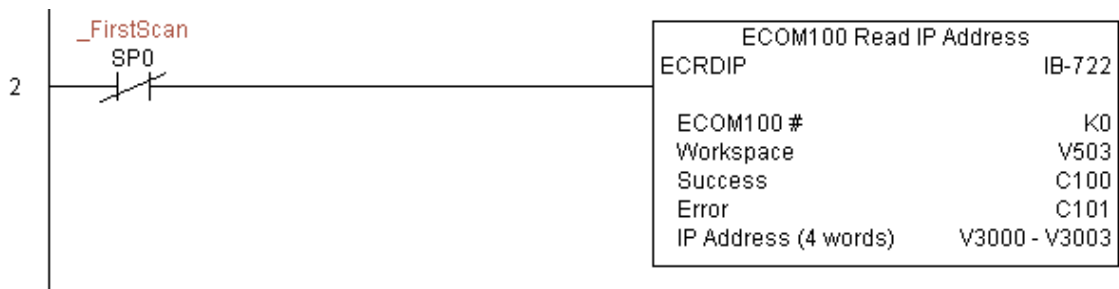
Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido, lee la dirección IP del módulo ECOM100 y la almacena en V3000 hasta V3003 (4 números decimales). Esta información se puede exhibir en una interface de operador, por ejemplo.

La instrucción ECRDIP es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de leer la dirección IP del módulo será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bit C101.



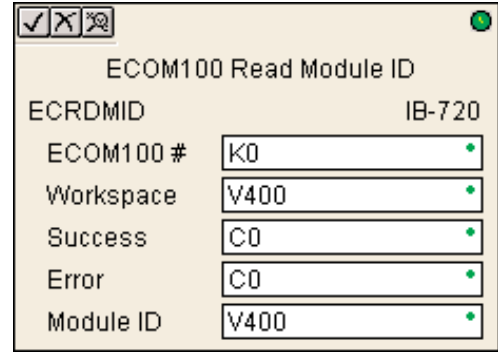
ECOM100 Read Module ID (ECRDMID) (IB-720)

DS5	Usado
HPP	N/A

ECOM100 Read Module ID will read the binary (decimal) WORD sized Module ID on a transition from OFF to ON to the IBox.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.



Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

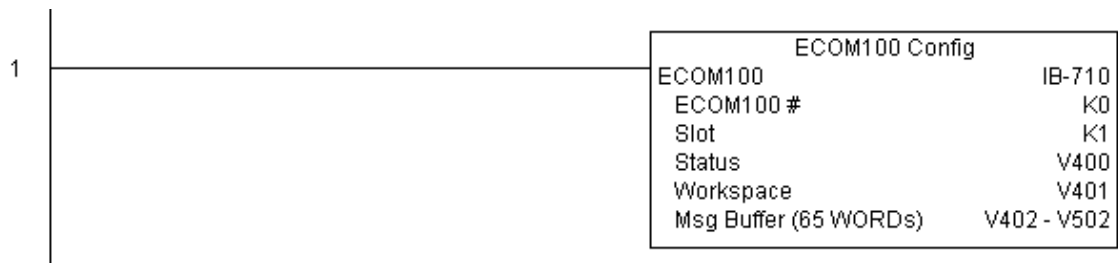
Parámetros de ECRDMID

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Module ID:** Especifica la localización donde será colocada la identificación **Module ID** (decimal) del módulo ECOM100

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Module ID. V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECRDMID

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido, lee la identificación ID del módulo ECOM100 y la almacena en V2000. Esta información se puede exhibir en una interface de operador, por ejemplo.

La instrucción ECRDMID es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de leer la identificación ID del módulo será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bir C101.



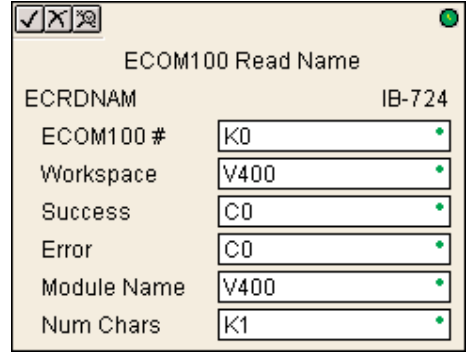
Leer el nombre del módulo ECOM100 (ECRDNAM) (IB-724)

DS5	Usado
HPP	N/A

Esta instrucción leerá el nombre del módulo hasta el número de caracteres especificados en una transición de APAGADO a ENCENDIDO al IBox.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.



Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

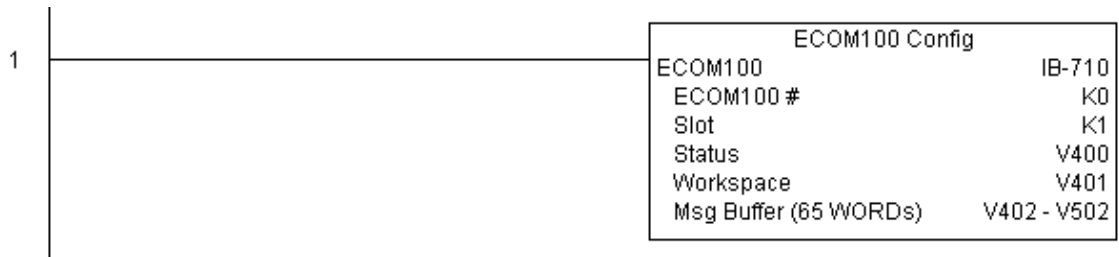
Parámetros de ECRDNAM

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Module Name:** Especifica la localización inicial del almacenador intermediario en donde será almacenado el nombre del módulo ECOM100
- **Num Chars:** Especifica la cantidad de caracteres (bytes) a ser leídos desde el campo Module Name del módulo ECOM100

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Module Name V	Vea el mapa de memoria V del DL06 - Data Words
Num Chars K	K1-128

Ejemplo de ECRDNAM

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



5

Renglón 2: En el segundo barrido, lee el Module Name (Nombre del módulo) de ECOM100 y la almacena en V3000 hasta V3003 (8 caracteres). Esta información se puede exhibir en una interface de operador, por ejemplo.

La instrucción ECRDNAM es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de leer el nombre del módulo será enviado desde el módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bit C101.



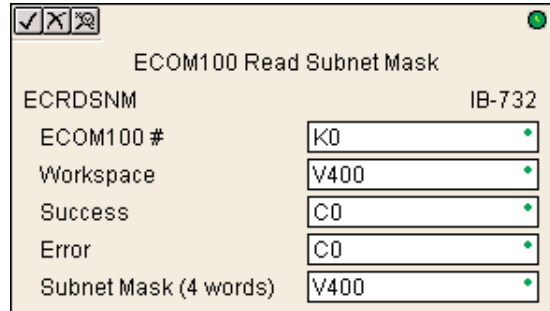
Lea Subnet Mask ECOM100 (ECRDSNM) (IB-732)

DS5	Usado
HPP	N/A

Esta instrucción permite leer las 4 partes del Subnet Mask y las almacena en 4 localizaciones de memoria V consecutivas en formato decimal, en una transición desde OFF para ON al IBox.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.



Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

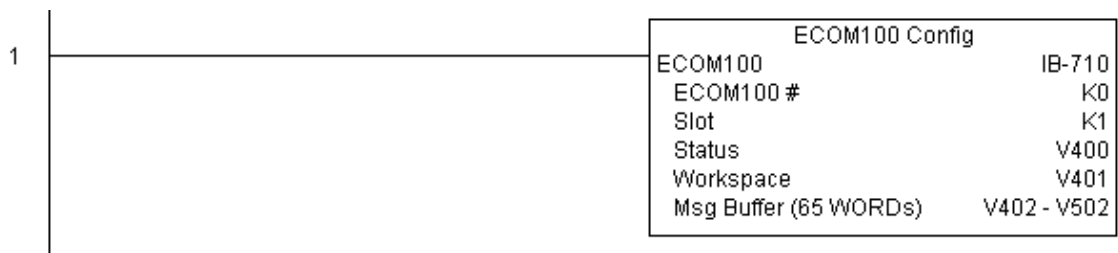
Parámetros de ECRDSNM

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito y completed
- **Subnet Mask:** Especifica la dirección inicial donde será almacenada la Subnet Mask del módulo ECOM100 en 4 localizaciones de memoria V consecutivas

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Subnet Mask (4 Words) V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de ECRDSNM

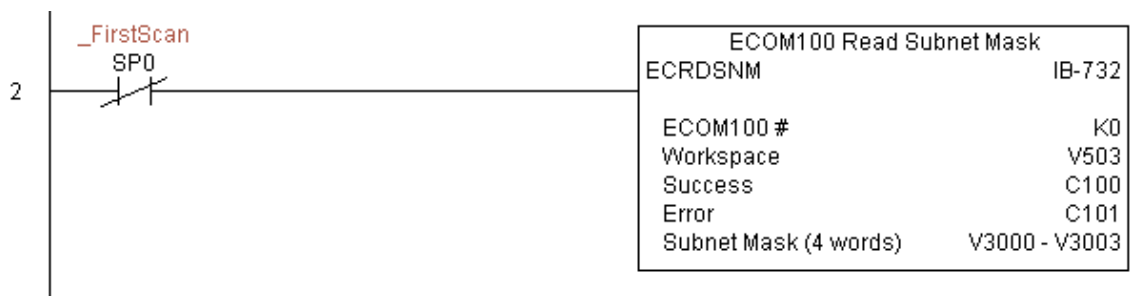
Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido, lee el Subnet Mask desde el módulo ECOM100 y la almacena en V3000 hasta V3003 (4 números decimales). Esta información se puede exhibir en una interface de operador, por ejemplo.

La instrucción ECRDSNM es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de leer el Subnet Mask será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa C101.



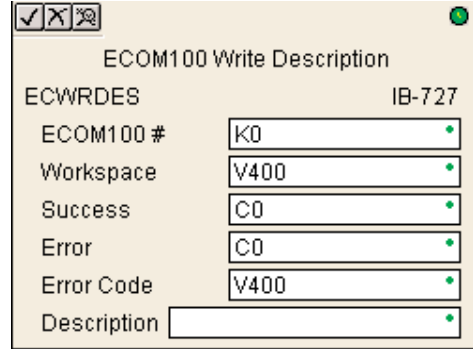
ECOM100 Write Descripción (ECWRDES) (IB-727)

Esta instrucción permite escribir una descripción al módulo ECOM100 en una transición de APAGADO a ENCENDIDO al IBox.

DS5	Usado
HPP	N/A

Si usted utiliza un signo dólar (\$) o una comilla ("), use la secuencia de escape de PRINT/VPRINT de **dos** signos dólar (\$\$) para un signo solamente o una comilla con un signo dólar (\$) para un carácter de comilla.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.



Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.

La descripción se almacena en la memoria Flash-ROM en el módulo ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escribe la memoria Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente este IBox UNA VEZ en el primer barrido. Ya que se requiere la ejecución de una transición de OFF para ON, use un contacto SP0 NORMALMENTE CERRADO para conducir el flujo de energía al IBox.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

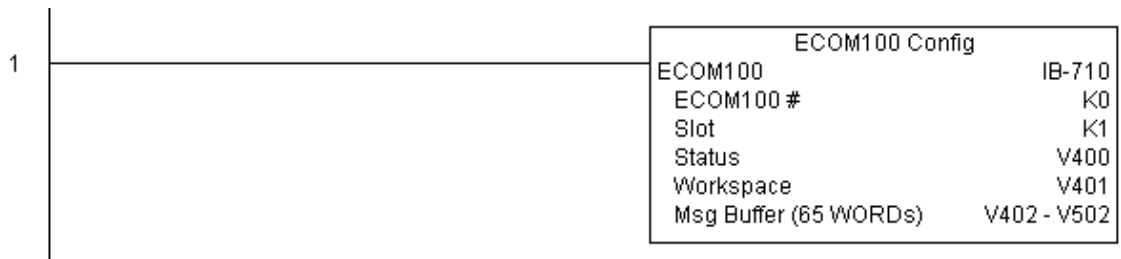
Parámetros de ECWRDES

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico cified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **Descripción:** Especifica la descripción (Descripción) que será escrita al módulo

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words
Descripción	Text

Ejemplo de ECWRDES

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



5

Renglón 2: En el segundo barrido del PLC, esta instrucción define la descripción del módulo (Module Descripción) ECOM100. Típicamente, this is done using NetEdit, but this IBox allows you to configure the module descripción in the ECOM100 using your ladder program.

La instrucción EWRDES es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de escribir la descripción del módulo será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bit C101. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.



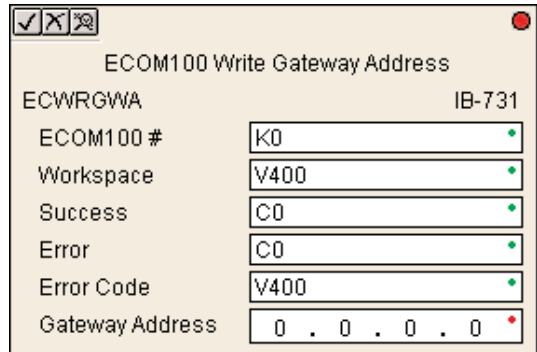
Escriba la dirección de Gateway del ECOM100 (ECWRGWA) (IB-731)

DS5	Usado
HPP	N/A

Esta instrucción permite escribir una dirección dada de Gateway al módulo ECOM100 en una transición de APAGADO a ENCENDIDO al IBox. Vea también la instrucción IBox ECIPSUP IB-717 para configurar TODOS LOS parámetros de TCP/IP en una sola instrucción - IP ADDRESS, subnet mask, y dirección de Gateway.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.



La dirección del Gateway se almacena en memoria Flash-ROM en el ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escribe la Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente este IBox UNA VEZ, en el primer barrido. Ya que se requiere la ejecución de una transición de OFF para ON, use un contacto SP0 NORMALMENTE CERRADO para conducir el flujo de energía al IBox.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

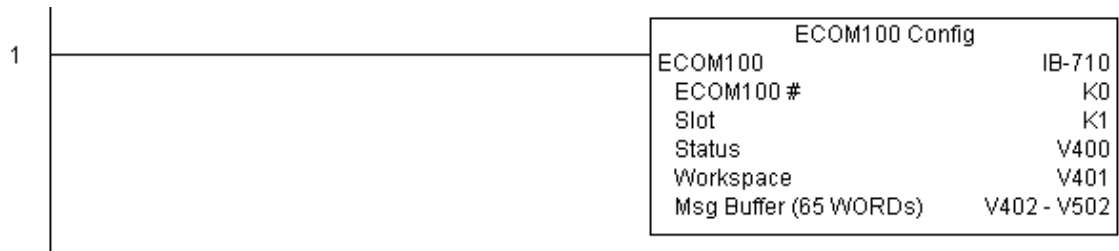
Parámetros de ECWRGWA

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico cified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **Gateway Address:** Especifica la dirección de Gateway que será escrita el módulo

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words
Gateway Address	0.0.0.1. to 255.255.255.254

Ejemplo de ECWRGWA

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



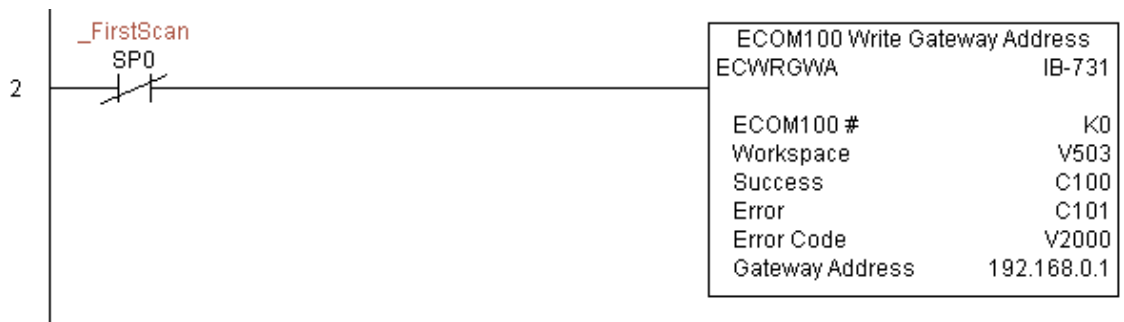
5

Renglón 2: En el segundo barrido del PLC, esta instrucción asigna la dirección de Gateway del módulo ECOM100 a 192.168.0.1.

La instrucción ECWRGWA es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de escribir la dirección de Gateway será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bit C101. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.

Para configurar todos los parámetros de TCP/IP al módulo ECOM100 en una instrucción, vea el IBox ECIPSUP.



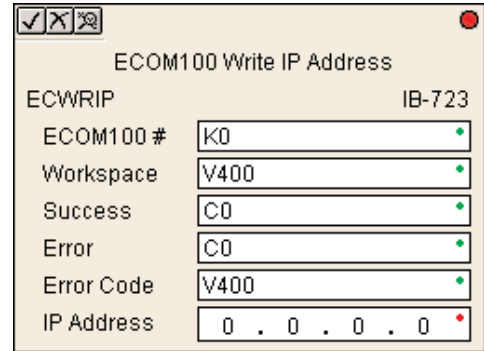
ECOM100 Write IP Address (ECWRIP) (IB-723)

DS5	Usado
HPP	N/A

Esta instrucción escribe una dirección IP dada al módulo ECOM100 en una transición de APAGADO a ENCENDIDO al IBox. Vea también la instrucción IBox ECIPSUP IB-717 para configurar TODOS LOS parámetros de TCP/IP en una sola instrucción - IP ADDRESS, subnet mask, y dirección de Gateway.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.



La dirección de IP se almacena en memoria Flash-ROM en el ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escribe la Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente este IBox UNA VEZ, en el primer barrido. Ya que se requiere la ejecución de una transición de OFF para ON, use un contacto SP0 NORMALMENTE CERRADO para conducir el flujo de energía al IBox.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

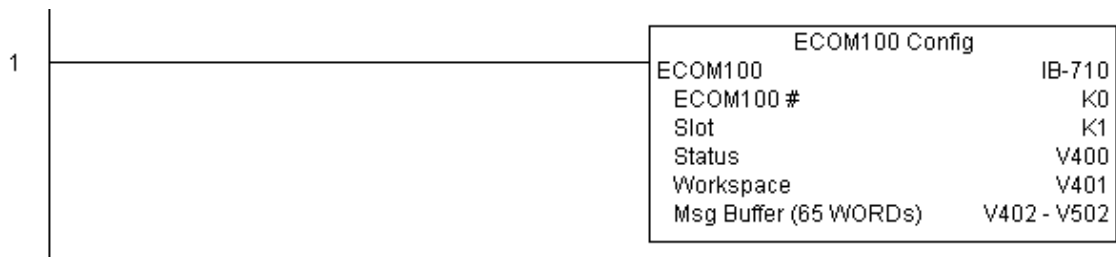
Parámetros de ECWRIP

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **IP Address:** Especifica la dirección IP que será escrita al módulo

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words
IP Address	0.0.0.1. to 255.255.255.254

Ejemplo de ECWRIP

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido del PLC, esta instrucción asigna la dirección IP Address del módulo ECOM100 a 192.168.12.100.

La instrucción ECWRIP es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de escribir la dirección de IP será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bit C101. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.

Para configurar todos los parámetros de TCP/IP de ECOM100 en una instrucción, vea el IBox ECIPSUP.



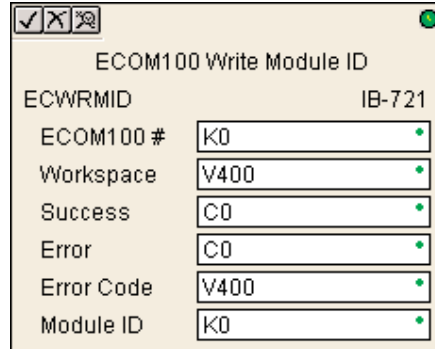
ECOM100 Write Module ID (ECWRMID) (IB-721)

DS5	Usado
HPP	N/A

Esta instrucción escribe una identificación dada (Module ID) al módulo en una transición de APAGADO a ENCENDIDO al IBox.

Si la identificación del módulo (ID) es configurada en hardware usando los DIP switches, esta instrucción IBox will fail and return error code 1005 (decimal).

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.



Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.

La información **Module ID** se almacena en memoria Flash-ROM en el ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escriba la memoria Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente UNA VEZ este IBox, en el primer barrido. Ya que se requiere la ejecución de una transición de OFF para ON, use un contacto SP0 NORMALMENTE CERRADO para conducir el flujo de energía al IBox.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

Parámetros de ECWRMID

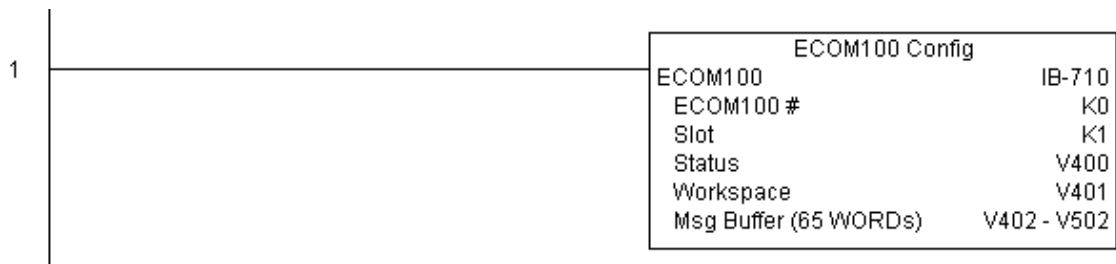
- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico cified slot. All other ECxxxx IBoxes that need to reference this ECOM100 module must reference this logical number
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **Module ID:** Especifica la identificación Module ID que será escrita al módulo

Parámetro	Rango del DL06
ECOM100#	K
Workspace	V
Success	X,Y,C,GX,GY,B
Error	X,Y,C,GX,GY,B
Error Code	V
Module ID	

Parámetro	Rango del DL06
ECOM100#	K0-255
Workspace	Vea el mapa de memoria V del DL06 - Data Words
Success	Vea el mapa de memoria DL06
Error	Vea el mapa de memoria DL06
Error Code	Vea el mapa de memoria V del DL06 - Data Words
Module ID	K0-65535

Ejemplo de ECWRMID

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido del PLC, esta instrucción asigna la identificación Module ID del módulo ECOM100. Típicamente esto es hecho usando NetEdit3, pero este IBox le permite configurar the module ID del ECOM100 usando el programa ladder.

La instrucción EWRMID es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de escribir el module ID será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, se activa el bit C101. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.



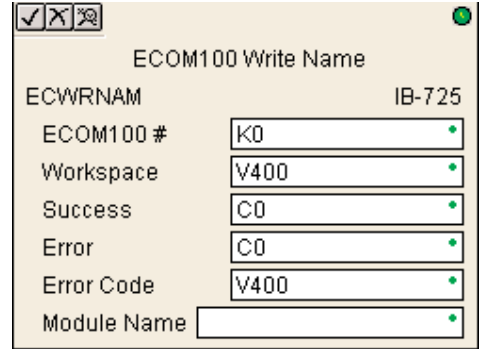
Escribir el nombre del ECOM100 (ECWRNAM) (IB-725)

DS5	Usado
HPP	N/A

Esta instrucción escribe un nombre dado al módulo ECOM100 en una transición de APAGADO a ENCENDIDO al IBox.

Si usted utiliza un signo dólar (\$) o una comilla ("), use la secuencia de escape de PRINT/VPRINT de dos signos dólar (\$\$) para un signo solamente o una comilla con un signo dólar (\$) para un carácter de comilla.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.



Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.

El nombre se almacena en la memoria Flash-ROM en el ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escribe la Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente este IBox UNA VEZ en el primer barrido. Ya que se requiere la ejecución de una transición de OFF para ON, use un contacto SP0 NORMALMENTE CERRADO para conducir el flujo de energía al IBox.

Para que este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

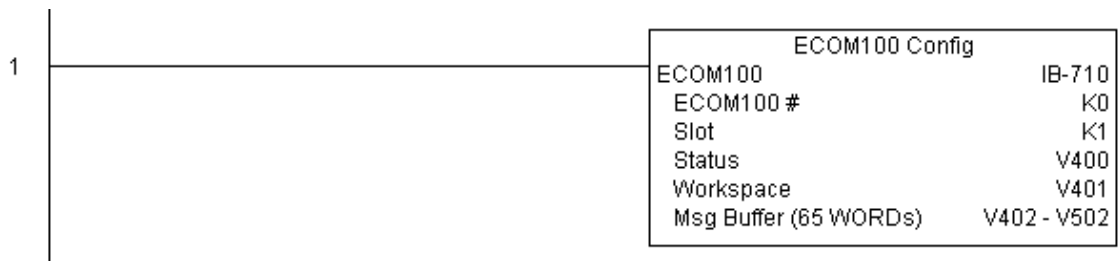
Parámetros de ECWRNAM

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **Module Name:** Especifica el nombre que será escrita al módulo

Parámetro	Rango del DL06
ECOM100# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error Code V	Vea el mapa de memoria V del DL06 - Data Words
Module Name	Text

Ejemplo de ECWRNAM

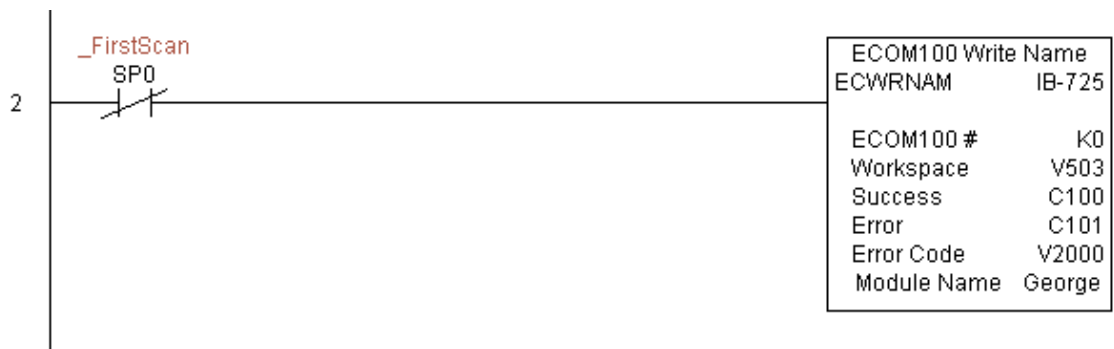
Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



Renglón 2: En el segundo barrido del PLC, esta instrucción asigna el nombre del módulo ECOM100. Típicamente esto es hecho usando NetEdit3, pero este IBox le permite configurar the nombre del ECOM100 usando el programa ladder.

La instrucción EWRNAM es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de escribir el nombre será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.



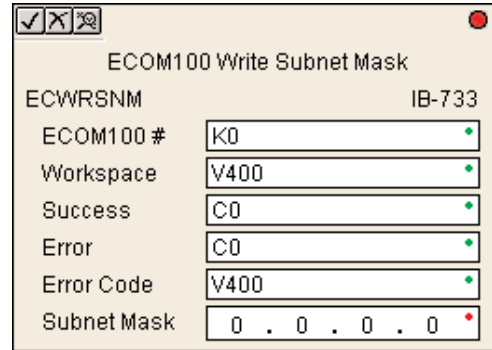
Escribir Subnet Mask a ECOM100 (ECWRSNM) (IB-733)

DS5	Usado
HPP	N/A

Esta instrucción escribe una Subnet Mask dada al módulo ECOM100 en una transición de APAGADO a ENCENDIDO al IBox. Vea también la instrucción IBox ECIPSUP IB-717 para configurar TODOS LOS parámetros de TCP/IP en una sola instrucción - IP ADDRESS, subnet mask, y dirección de Gateway.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Los bits de los parámetros Success o Error se activan una vez que el comando sea completado. Si hay un error, el parámetro Error Code (código de error) divulgará un código de error ECOM100 (menos de 100), o un error de lógica del PLC (mayor de 1000), etc.



La información de Subnet Mask se almacena en la memoria Flash-ROM en el ECOM100 y la ejecución de este IBox deshabilitará el módulo ECOM100 por lo menos un medio segundo hasta que escribe la Flash-ROM. Por lo tanto, SE RECOMIENDA que usted ejecute solamente este IBox UNA VEZ en el primer barrido. Ya que se requiere la ejecución de una transición de OFF para ON, use un contacto SP0 NORMALMENTE CERRADO para conducir el flujo de energía al IBox.

Para este IBox ECOM100 funcione, usted debe mover el DIP switch 7 a la posición ON en el circuito de ECOM100.

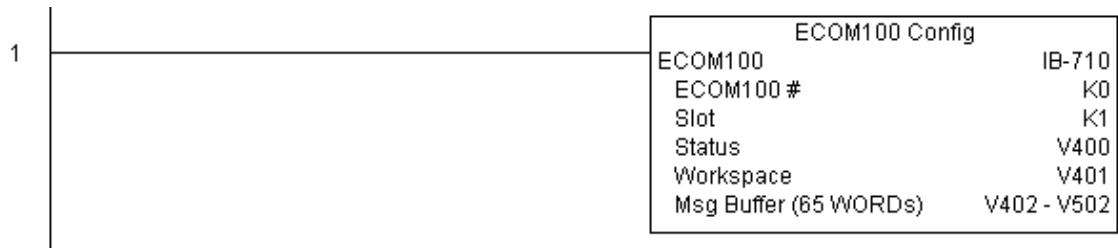
Parámetros de ECWRSNM

- **ECOM100#:** éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito
- **Error Code:** Especifica la localización en donde será escrito el código de error
- **Subnet Mask:** Especifica la Subnet Mask que será escrita al módulo

Parámetro	Rango del DL06
ECOM100#	K0-255
Workspace	Vea el mapa de memoria V del DL06 - Data Words
Success	Vea el mapa de memoria DL06
Error	Vea el mapa de memoria DL06
Error Code	Vea el mapa de memoria V del DL06 - Data Words
Subnet Mask	Masked IP Address

Ejemplo de ECWRSNM

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



5

Renglón 2: En el segundo barrido del PLC, esta instrucción asigna el Subnet Mask del módulo ECOM100 a 255.255.0.0

La instrucción ECWRSNM es activada por una transición de OFF para ON, (similar a la entrada de un contador). El comando de escribir el Subnet Mask será enviado al módulo ECOM100 siempre que el flujo de energía en el IBox vaya de APAGADO a ENCENDIDO.

Si funciona correctamente, se activa el bit C100. Si hay una falla, usted puede mirar V2001 para ver el código de error específico.

Para configurar todos los parámetros de TCP/IP de ECOM100 en una instrucción, vea el IBox ECIPSUPP.



Lee datos RX con ECOM100 (ECRX) (IB-740)

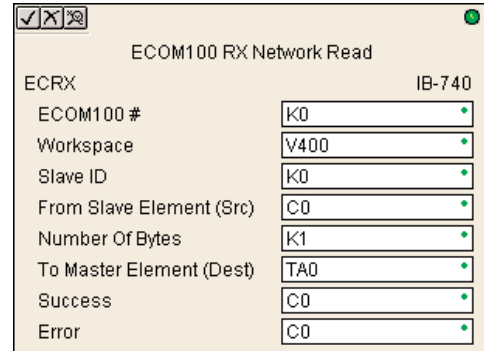
DS5	Usado
HPP	N/A

Esta instrucción es la instrucción RX con enclavamiento incorporado con otros IBoxes de ECOM100 RX (ECRX) y ECOM100 WX (ECWX) en el programa para simplificar el establecimiento de una red de comunicaciones. Realizará la instrucción RX en la red especificada de módulos ECOM100, que corresponde a un IBox de configuración específica en la parte superior del programa.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Siempre que este IBox tenga energía, leerá datos de elementos del esclavo especificado en el almacenador intermediario dado de la memoria V de destino, dándole oportunidad de que otros IBoxes de ECOM100 RX y ECOM100 WX en éste ECOM100 # sean ejecutados.

Por ejemplo, si usted desea leer y escribir datos continuamente a partir de 5 esclavos diferentes, usted puede tener todas estas instrucciones de ECRX y de ECWX en UN RENGLON controlado con SP1 (siempre encendido). Se ejecutarán secuencialmente, automáticamente.



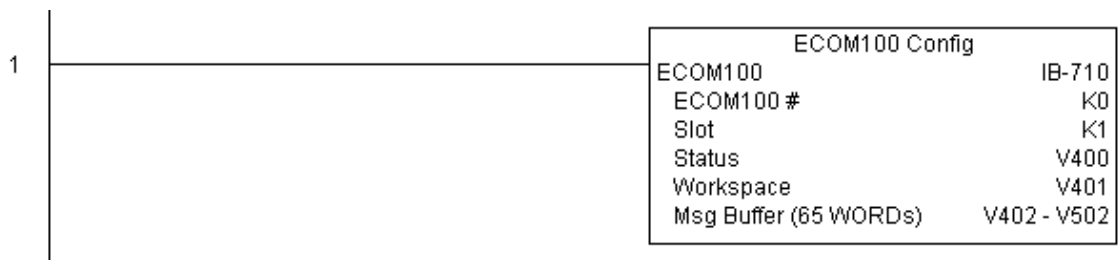
Parámetros de ECRX

- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Slave ID:** Especifica el PLC esclavo que será apuntado por la instrucción de ECRX
- **From Slave Element (Src):** Especifica la dirección de esclavo de donde se leerán los datos
- **Number of Bytes:** Especifica el número de bytes a leer en el PLC esclavo
- **To Master Element (Dest):** Especifica la localización en donde serán colocados los datos del esclavo en el PLC maestro con el ECOM100
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
ECOM100#	K
ECOM100#	K0-255
Workspace	V
Workspace	Vea el mapa de memoria V del DL06 - Data Words
Slave ID	K
Slave ID	K0-90
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	Vea el mapa de memoria DL06
Number of Bytes	K
Number of Bytes	K1-128
To Master Element (Dest)	V
To Master Element (Dest)	Vea el mapa de memoria V del DL06 - Data Words
Success	X,Y,C,GX,GY,B
Success	Vea el mapa de memoria DL06
Error	X,Y,C,GX,GY,B
Error	Vea el mapa de memoria DL06

Ejemplo de ECRX

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todos los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.



(Este ejemplo continúa en la próxima página)

Ejemplo de ECRX (continuado)

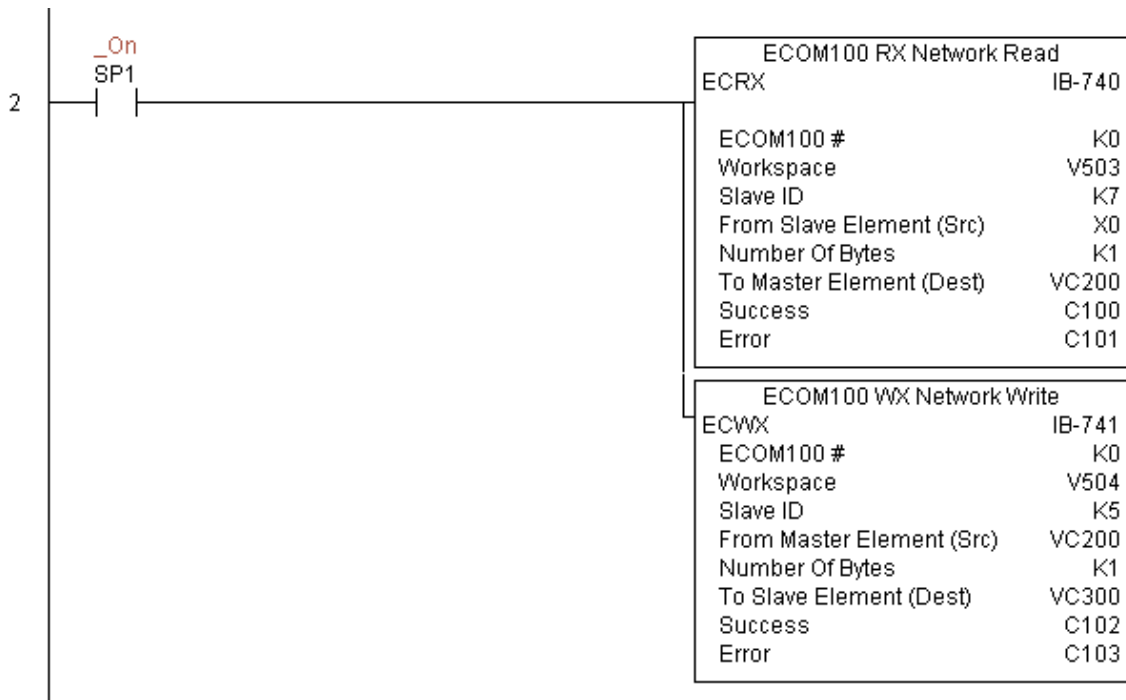
Renglón 2: Usando ECOM100 # K0, lea X0-X7 del esclavo K7 y escribalo al esclavo K5 tan rápidamente como sea posible. Almacénelos en este PLC local en C200-C207, y escribalo a C300-C307 en el esclavo K5.

Las instrucciones ECRX y ECWX trabajan con los IBoxes para simplificar todo el establecimiento de una red haciendo los enclavamientos y distribuyendo los recursos apropiados. También suministran un reportaje de errores muy simplificada. Usted no tiene que preocuparse de ningun relevador especial SP116, por ejemplo, Busy u "ocupado" o los "bits de error", o en qué número de ranura está colocado un módulo, o tener contadores o shift registers o cualquiera otro enclavamiento para la administración de recursos.

En este ejemplo, SP1 (siempre ON) está conduciendo los IBoxes ECRX y ECWX en el mismo renglón. En el barrido que termina RX, la instrucción WX comenzará en ese mismo barrido. Tan pronto como la instrucción WX termine, cualquier operación pendiente debajo de ella en el programa conseguirá una chance the ejecutarse. Si no hay IBoxes de ECOM100 pendientes debajo del ECWX, el ECRX comenzaría su petición otra vez en el pxóximo barrido.

Usando el ECRX y el ECWX para todos las lecturas y escrituras en la red es el método más rápido que el PLC puede hacer para establecimiento de una red. Para los puertos seriales locales, los módulos de DCM, o los módulos originales de ECOM, use las instrucciones IBoxes NETCFG y NETRX/NETWX.

5



Escribe datos WX con ECOM100 (ECWX) (IB-741)

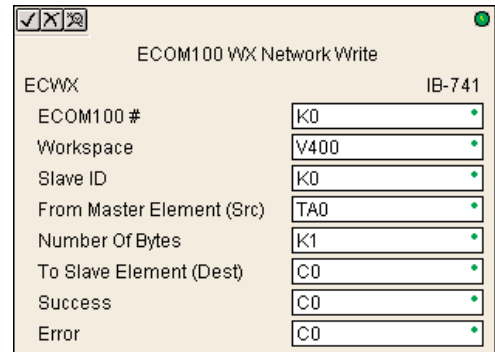
DS5	Usado
HPP	N/A

Esta instrucción es la instrucción WX con enclavamiento incorporado con otros IBoxes de ECOM100 RX (ECRX) y ECOM100 WX (ECWX) en el programa para simplificar el establecimiento de una red de comunicaciones. Realizará la instrucción WX en la red especificada de módulos ECOM100, que corresponde a un IBox de configuración específica en la parte superior del programa

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Siempre que este IBox tenga energía, escribirá datos de elementos desde el maestro en el almacenador intermediario dado de memoria V al esclavo especificado comenzando con el elemento dado del esclavo, dándole oportunidad de que otros IBoxes de ECOM100 RX y ECOM100 WX en ése ECOM100 # sean ejecutados.

Por ejemplo, si usted desea leer y escribir datos continuamente a partir de 5 esclavos diferentes, usted puede tener todas estas instrucciones de ECRX y de ECWX en UN RENGLON controlado con SP1 (siempre encendido). Se ejecutarán secuencialmente, automáticamente.



Parámetros de ECWX

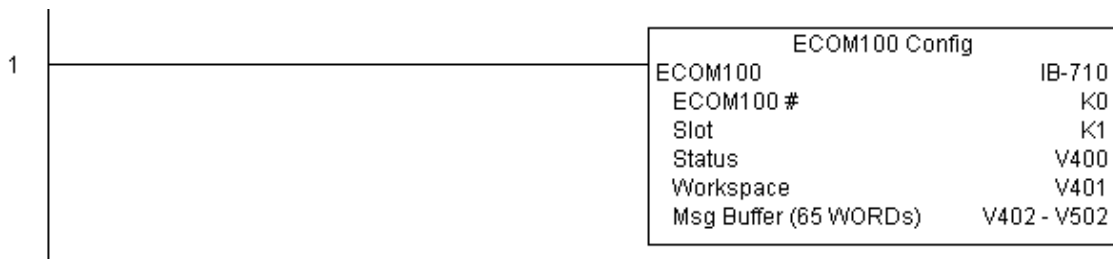
- **ECOM100#:** Éste es un número lógico asociado a este módulo específico ECOM100 en la ranura especificada. El resto de los IBoxes ECxxxx que necesitan referirse a este módulo ECOM100 deben referirse a este número lógico
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Slave ID:** Especifica el PLC esclavo que será apuntado por la instrucción de ECWX
- **From Master Element (Src):** Especifica una localización de memoria V en el PLC maestro con ECOM100 de donde será el origen de los datos
- **Number of Bytes:** Especifica la cantidad de bytes a ser escritos al PLC esclavo con ECOM(100)
- **To Slave Element (Dest):** Especifica la dirección del esclavo donde serán escritos los datos
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06	
ECOM100#	K	K0-255
Workspace	V	Vea el mapa de memoria V del DL06 - Data Words
Slave ID	K	K0-90
From Master Element (Src)	V	Vea el mapa de memoria V del DL06 - Data Words
Number of Bytes	K	K1-128
To Slave Element (Dest) ..	X,Y,C,S,T,CT,GX,GY,V	Vea el mapa de memoria DL06
Success	X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error	X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de ECWX

Renglón 1: Esta instrucción es responsable por la coordinación y enclavamiento de todos los tipos de IBoxes ECOM100 para un módulo específico ECOM100. Marque el ECOM100 con un rótulo en la ranura 1 como ECOM100K0. El resto de los IBoxes ECxxxx se refieren a este módulo como K0. Si usted necesita cambiar el módulo en la base a una ranura diferente, se necesita solamente cambiar este IBox. V400 es usado como registro global de estado del resultado para otros IBoxes ECxxxx que usan este módulo específico ECOM100. V401 es usado para coordinar y enclavar la lógica en todo los otros IBoxes ECxxxx usando este módulo específico ECOM100. V402-V502 es un campo común almacenador intermediario de 130 bytes disponibles para uso por otros IBoxes ECxxxx usando este módulo específico ECOM100.

5



(Este ejemplo continúa en la próxima página)

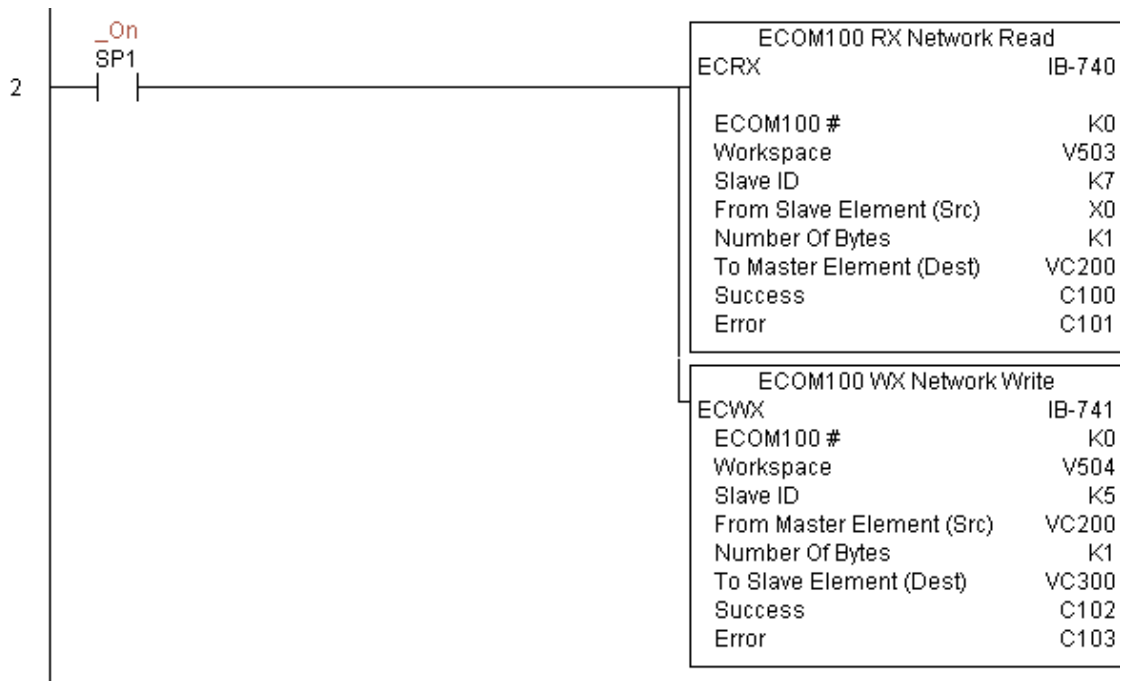
Ejemplo de ECWX (continuado)

Renglón 2: Usando ECOM100 # K0, lea X0-X7 del esclavo K7 y escribalo al esclavo K5 tan rápidamente como sea posible. Almacene los datos en este PLC local en C200-V207, y escribalos a C300-C307 en el esclavo K5.

Las instrucciones ECRX y ECWX trabajan con el IBox de configuración del ECOM100 para simplificar todo el establecimiento de una red administrando todo el enclavamiento y recursos apropiados. También suministran un reportaje de error muy simplificada. Usted no tiene que preocuparse de ningun SP "Busy" o "bits de error", o en qué número de ranura está un módulo, o tener algunos contadores o shift registers u otros enclavamientos para la administración de recursos.

En este ejemplo, el contacto SP1 (Siempre ON) está controlando las instrucciones IBoxes ECRX y ECWX en el mismo renglón. En el barrido cuando se termina la ejecución de la instruccion RX, la instrucción comenzará en el mismo barrido. Tan pronto como termine WX, cualquier operación pendiente debajo de ella en el programa tendrá la oprtunidad de poder ser ejecutada. Si no hay IBoxes ECOM100 pendientes debajo del ECWX, entonces en el próximo barrido la instrucción ECRX comenzaría su petición otra vez.

Usando el ECRX y el ECWX para toda sus lecturas y escrituras en la red de ECOM100 es lo más rápido que el PLC puede hacer para el establecimiento de una red. Para los puertos seriales locales, los módulos de DCM, o los módulos originales de ECOM, use los IBoxes NETCFG y NETRX/NETWX.



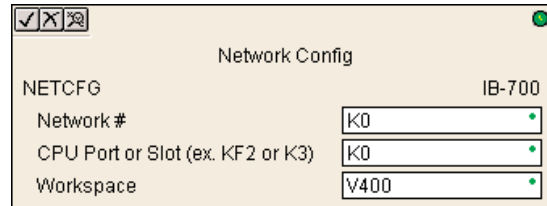
Configuración de una read NETCFG (NETCFG) (IB-700)

DS5	Usado
HPP	N/A

Esta instrucción define toda la información común necesaria para realizar establecimiento de una red con instrucciones RX/WX usando las instrucciones IBox NETRX y NETWX utilizando un puerto serial local de la CPU, el módulo D0-DCM o H0-ECOM.

Usted debe tener la instrucción de configuración de la red en la parte superior de su programa ladder o de etapas con cualquier otros IBoxes de configuración.

Si usted utiliza más que un puerto serial local, D0-DCM o H0-ECOM en su PLC para el establecimiento de una red RX/WX, usted debe tener una instrucción diferente de configuración de red para CADA red de RX/WX en su sistema que utilice alguna instrucción IBox del tipo NETRX/NETWX.



El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

El segundo parámetro "puerto o ranura de la CPU" es el mismo valor que en el byte mas significativo de la primera instrucción del LD si usted estuviera haciendo el programa de RX o de WX usted mismo. Este valor es específico de la CPU y del puerto, pero los valores posibles incluyen KF2 para el puerto serial local 2 de la CPU de 06, K3 para un DCM o un ECOM en la ranura 3 de un local 205 bajo, o de K37 para un DCM en una base 3, ranura 7 de 405 extensiones.

Parámetros de NETCFG

- **Network#:** Especifica un único número para cada red de ECOM(100) o de DCM a ser usado
- **CPU Port or Slot:** Especifica el número de acceso de la CPU o el número de la ranura DCM/ECOM(100) usado
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción

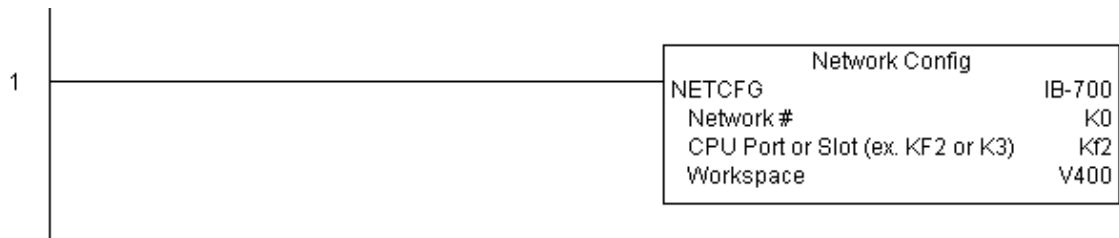
Parámetro	Rango del DL06
Network# K	K0-255
CPU Port or Slot K	K0-FF
Workspace V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de NETCFG

Esta instrucción coordina toda la interacción con otras instrucciones IBox (NETRX/NETWX) en al red. Usted debe tener un IBox de configuración de red para cada red de puerto serial, del módulo de DCM, o la red original del módulo de ECOM en su sistema. Las instrucciones IBox de configuración deben estar en la parte superior de su programa y debe ser ejecutadas en cada barrido.

Este IBox define la red # K0 estando ubicada en el puerto serial local #2 (KF2) de la CPU. Para los puertos seriales locales de la CPU o los módulos de DCM/ECOM, use el mismo valor que usted utilizaría en el byte más significativo de la primera instrucción del LD en un renglón normal de RX/WX de referirse al puerto o al módulo. Cualquiera de los IBoxes NETRX o NETWX que necesite referirse a esta red específica incorporaría K0 para el parámetro de número de red.

El registro del espacio de trabajo es usado para mantener la información del estado sobre el puerto o el módulo, junto con compartir apropiado y enclavamientos con el otros IBoxes del tipo NETRX y NETWX en el programa. Este registro de memoria de V no debe ser usado en cualquier otro lugar en el programa entero.



Leer la red RX (NETRX) (IB-701)

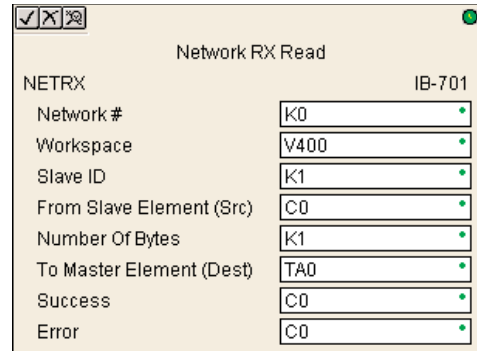
DS5	Usado
HPP	N/A

La instrucción RX lee datos en la red con enclavamiento incorporado con el resto de los IBoxes RX (NETRX) y WX (NETWX) en el programa ladder, para simplificar el establecimiento de una red de comunicación. Realizará el RX en el número de la red especificada, que corresponde a una configuración de red única específica (NETCFG) en la parte superior del programa ladder.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Siempre que este IBox tenga energía, leerá datos del elemento del esclavo especificado en el almacenador intermediario dado de la memoria V de destino, dándole la oportunidad de ejecutar otros IBoxes RX y WX en ese número de red.

Por ejemplo, si usted desea leer y escribir datos continuamente a partir de 5 esclavos diferentes, usted puede tener todas estas instrucciones de NETRX y NETWX en UN RENGLON controlado con SP1 (siempre encendido). Se ejecutarán secuencialmente, automáticamente.



Parámetros de NETRX

- **Network#:** Especifica el número de red (puertos de la CPU, DCM, ECOM) definido por la instrucción NETCFG
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Slave ID:** Especifica el PLC esclavo que será interrogado por la instrucción
- **From Slave Element (Src):** Especifica la dirección del PLC esclavo de donde serán leídos los datos
- **Number of Bytes:** Especifica la cantidad de bytes a ser leídos desde el PLC esclavo
- **To Master Element (Dest):** Especifica la localización de memoria en el PLC maestro donde serán colocados los datos del esclavo
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

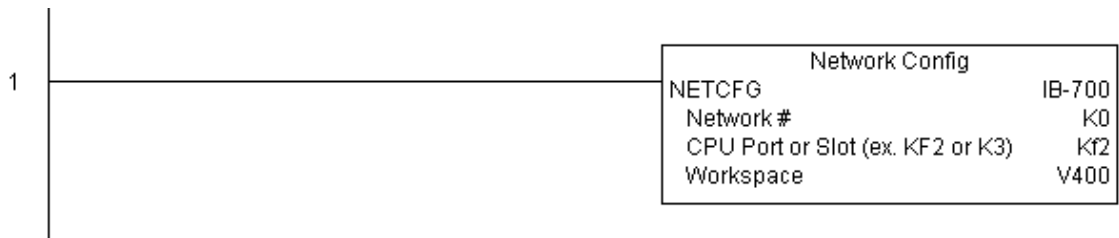
Parámetro	Rango del DL06
Network# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Slave ID K	K0-90
From Slave Element (Src) X,Y,C,S,T,CT,GX,GY,V	Vea el mapa de memoria DL06
Number of Bytes K	K1-128
To Master Element (Dest) V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de NETRX

Esta instrucción coordina toda la interacción con otras instrucciones IBox (NETRX/NETWX) en al red. Usted debe tener un IBox de configuración de red para cada red de puerto serial, del módulo de DCM, o la red original del módulo de ECOM en su sistema. Las instrucciones IBox de configuración deben estar en la parte superior de su programa y debe ser ejecutadas en cada barrido.

Este IBox define la red # K0 estando ubicada en el puerto serial local #2 (KF2) de la CPU. Para los puertos seriales locales de la CPU o los módulos de DCM/ECOM, use el mismo valor que usted utilizaría en el byte más significativo de la primera instrucción del LD en un renglón normal de RX/WX de referirse al puerto o al módulo. Cualquiera de los IBoxes NETRX o NETWX que necesite referirse a esta red específica incorporaría K0 para el parámetro de número de red.

El registro del espacio de trabajo es usado para mantener la información del estado sobre el puerto o el módulo, junto con compartir apropiado y enclavamientos con el otros IBoxes del tipo NETRX y NETWX en el programa. Este registro de memoria de V no debe ser usado en cualquier otro lugar en el programa entero.



(Este ejemplo continúa en la próxima página)

Ejemplo de NETRX (continuado)

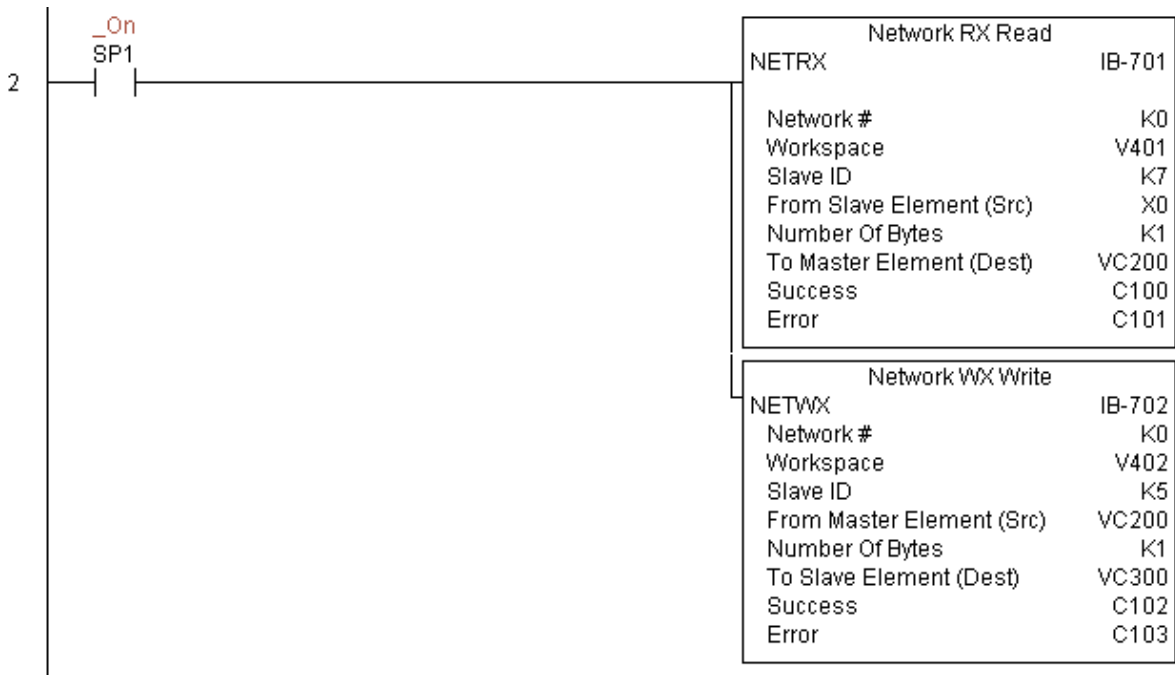
Renglón 2: Using Network# K0, read X0-X7 from Slave K7 and write them to slave K5 as fast as possible. Store them in this local PLC in C200-C207, and write them to C300-C307 in slave K5.

Los IBoxes NETRX y NETWX trabajan con los IBox de configuración de la red para simplificar todo el establecimiento de una red administrando los enclavamientos y recursos apropiados. También suministran un reportaje de error muy simplificado. Usted no necesita preocuparse de ningún SP "Busy bits" o "bits de error", o qué número de acceso o en que número de ranura está instalado un módulo, o tener contadores o shift register u otros enclavamientos para la administración de recursos.

En este ejemplo, SP1 (siempre ON) está controlando los IBoxes NETRX y NETWX en el mismo renglón. En el barrido que la red leída termina, la red escribe comenzará que igual explora. Tan pronto como la se complete de ejecutar la instrucción RX, cualquier operación pendiente debajo de ella en el programa va a poder ser ejecutada. Si no hay IBoxes NETRX o NETWX pendientes debajo de este IBox, entonces en el próximo barrido el NETRX comenzaría su operación nuevamente.

Usando los IBoxes NETRX y el NETWX para todos los puertos seriales, para D0-DCM, o una red original de ECOM es la forma más rápida que el PLC puede hacer establecimiento de una red. Para los módulos ECOM100, use los IBoxes ECOM100 y ECRX/ECWX.

5



Escribir a la red WX (NETWX) (IB-702)

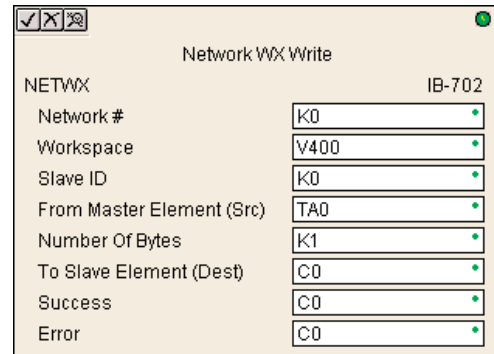
DS5	Usado
HPP	N/A

La instrucción WX escribe datos en la red con enclavamiento incorporado con el resto de los IBoxes RX (NETRX) y WX (NETWX) en el programa ladder, para simplificar el establecimiento de una red de comunicación. Realizará el RX en el número de la red especificada, que corresponde a una configuración de red única específica (NETCFG) en la parte superior del programa ladder.

El parámetro Workspace (espacio de trabajo) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.

Siempre que este IBox tenga energía, escribirá datos desde una memoria del PLC maestro especificado a la memoria V de destino en un esclavo dado, dándole la oportunidad de ejecutar otros IBoxes RX y WX en ese número de red.

Por ejemplo, si usted desea leer y escribir datos continuamente a partir de 5 esclavos diferentes, usted puede tener todas estas instrucciones de NETRX y NETWX en UN RENGLON controlado con SP1 (siempre encendido). Se ejecutarán secuencialmente, automáticamente.



Parámetros de NETWX

- **Network#:** Especifica el número de red (puertos de la CPU, DCM, ECOM) definido por la instrucción NETCFG
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Slave ID:** Especifica el PLC esclavo que será escrito por la instrucción
- **From Master Element (Src):** Especifica la localización del PLC maestro en donde los datos serán originados
- **Number of Bytes:** Especifica la cantidad de bytes a ser escritos al PLC esclavo
- **To Slave Element (Dest):** Especifica la dirección del esclavo en donde serán escritos los datos
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

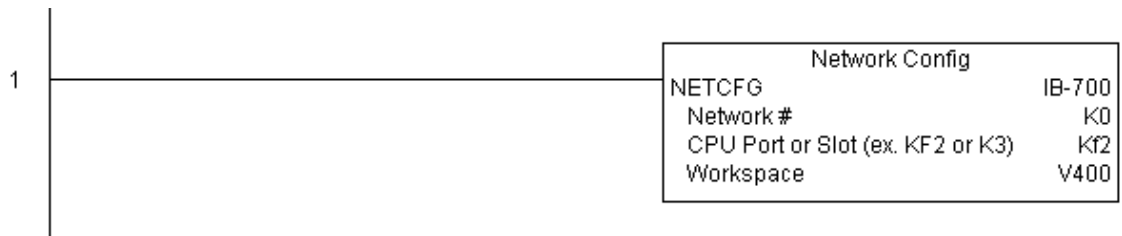
Parámetro	Rango del DL06
Network# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Slave ID K	K0-90
From Master Element (Src) V	Vea el mapa de memoria V del DL06 - Data Words
Number of Bytes K	K1-128
To Slave Element (Dest) . . X,Y,C,S,T,CT,GX,GY,V	Vea el mapa de memoria DL06
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de NETWX

Esta instrucción coordina toda la interacción con otras instrucciones IBox (NETRX/NETWX) en al red. Usted debe tener un IBox de configuración de red para cada red de puerto serial, del módulo de DCM, o la red original del módulo de ECOM en su sistema. Las instrucciones IBox de configuración deben estar en la parte superior de su programa y debe ser ejecutadas en cada barrido.

Este IBox define la red # K0 estando ubicada en el puerto serial local #2 (KF2) de la CPU. Para los puertos seriales locales de la CPU o los módulos de DCM/ECOM, use el mismo valor que usted utilizaría en el byte más significativo de la primera instrucción del LD en un renglón normal de RX/WX de referirse al puerto o al módulo. Cualquiera de los IBoxes NETRX o NETWX que necesite referirse a esta red específica incorporaría K0 para el parámetro de número de red.

El registro del espacio de trabajo es usado para mantener la información del estado sobre el puerto o el módulo, junto con compartir apropiado y enclavamientos con el otros IBoxes del tipo NETRX y NETWX en el programa. Este registro de memoria de V no debe ser usado en cualquier otro lugar en el programa entero.



(Este ejemplo continúa en la próxima página)

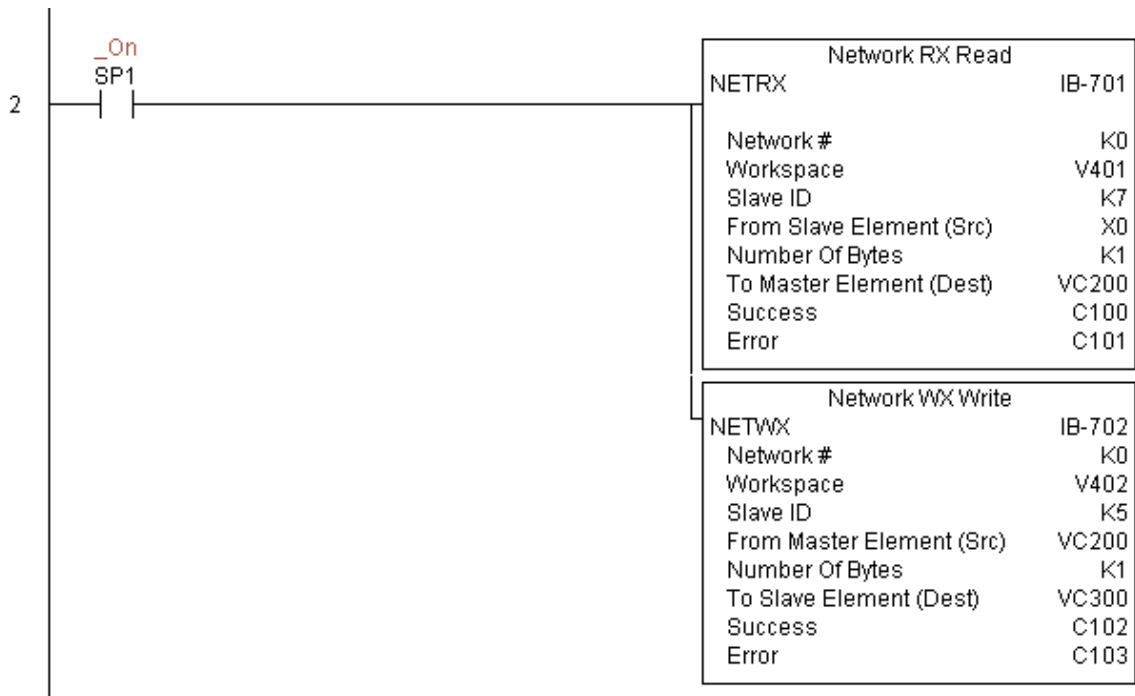
Ejemplo de NETWX (continuado)

Renglón 2: Con la red # K0, lea X0-X7 del esclavo K7 y escríbalo al esclavo K5 tan rápidamente como sea posible. Almacénelos en este PLC local en C200-C207, y escríbalos a C300-C307 en el esclavo K5.

Las instrucciones NETRX y NETWX trabajan con el IBox de configuración de la red para simplificar todo el establecimiento de una red controlados los enclavamientos y recursos apropiados. También suministran un reportaje de error muy simplificado.. Usted no necesita preocuparse de ningún SP "Busy bits" o "bits de error", o qué número de acceso o en que número de ranura está instalado un módulo, o tener contadores o shift register u otros enclavamientos para la administración de recursos.

En este ejemplo, SP1 (siempre ON) está controlando los IBoxes NETRX y NETWX en el mismo renglón. En el mismo barrido en que termina la instrucción RX, comenzará la instrucción WX. Tan pronto como WX termine, cualquier operación pendiente debajo de ella en el programa podrá ser ejecutada. Si no hay IBoxes NETRX o NETWX pendientes debajo de este IBox, entonces la instrucción NETRX comenzaría su petición en el próximo barrido.

Usando los IBoxes NETRX y el NETWX para todos los puertos seriales, para D0-DCM, o una red original de ECOM es la forma más rápida que el PLC puede hacer establecimiento de una red. Para los módulos ECOM100, use los IBoxes ECOM100 y ECRX/ECWX.



Configuración de CTRIO (CTRIO) (IB-1000)

DS5	Usado
HPP	N/A

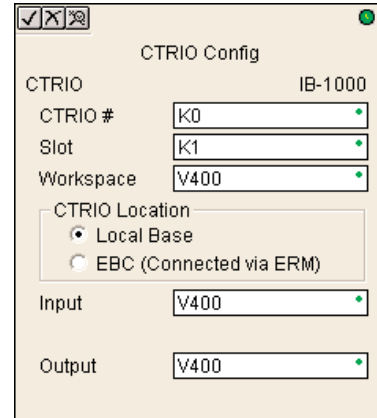
Esta instrucción define toda la información común para un módulo específico de CTRIO que sea usado por las otras instrucciones IBox de CTRIO (por ejemplo, CTRLDPR - cargar perfil de CTRIO, CTREDRL - CTRIO corregir y recargar una tabla de valores predefinidos, CTRRTL - Modo Run to limit de CTRIO ...).

Los parámetros de entradas-salidas para esta instrucción se pueden copiar directamente de la configuración del banco de trabajo de CTRIO para este módulo de CTRIO.

Usted debe tener los IBoxes de configuración de CTRIO en la parte superior de su programa ladder o de etapas junto con cualquier otros IBoxes de configuración.

Si usted tiene más de un CTRIO en su PLC, usted debe tener un IBox de CTRIO diferente para CADA módulo de CTRIO en su sistema que use alguna instrucción de IBox CTRIO . Cada IBox de configuración de CTRIO debe tener un ÚNICO valor de número de CTRIO. Ésto es cómo los IBoxes CTRIO se distinguen entre los diferente módulos de CTRIO en su sistema.

El parámetro del espacio de trabajo(Workspace) es un registro interno, privado usado por este IBox y DEBE SER ÚNICO en esta una instrucción y NO DEBE ser usado en cualquier otro lugar en el programa.



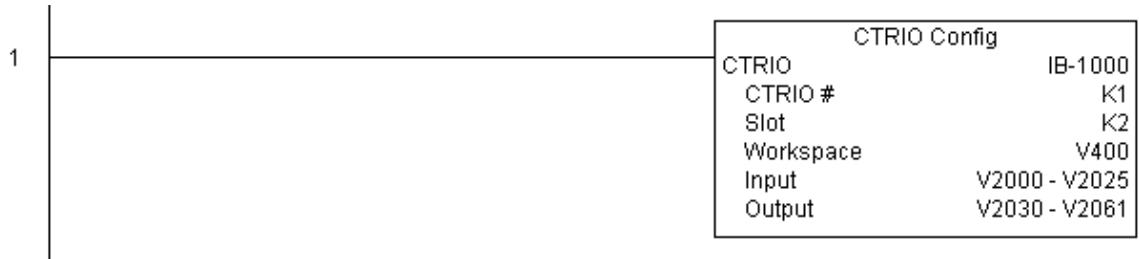
Parámetros de CTRIO

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario
- **Slot:** Especifica which PLC option slot the CTRIO module occupies
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **CTRIO Location:** Especifica donde se localiza el módulo (base local solamente para DL06)
- **Input:** Esto necesita ser configurado al mismo registro de memoria V que se especifica en CTRIO Workbench que dirección que comienza para las entradas, para este CTRIO único.
- **Output:** Esto necesita ser configurado al mismo registro de memoria V que se especifica en CTRIO Workbench como 'Starting V address for outputs' para este CTRIO único.

Parámetro	Rango del DL06
CTRIO# K	K0-255
Slot K	K1-4
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Input V	Vea el mapa de memoria V del DL06 - Data Words
Output V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de CTRIO

Renglón 1: Este ejemplo configura el módulo H0-CTRIO en la ranura 2 de la base del PLC DL06. Cada CTRIO en el sistema debe usar un I-box CTRIO diferente antes de que otros I Boxes CTRxxxx puedan ser usados. El módulo H0-CTRIO ha sido configurado para usa V2000 hasta V2025 en sus datos de entrada, y V2030 hasta V2061 for para sus datos de salidas.



Cree una tabla de valores predefinidos en CTRIO (CTRADPT) (IB-1005)

DS5	Usado
HPP	N/A

Esta instrucción añadirá una entrada al final de una tabla de valores predefinidos en un recurso específico de salida de CTRIO, en una transición de OFF para ON. Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el IBox de CTRIO leer código de error (CTRRDER) para obtener para obtener información adicional del error.

Posibles tipos de entradas:

- K0: Set
- K1: Reset
- K2: Pulse On (usa Pulse Time)
- K3: Pulse Off (usa Pulse Time)
- K4: Toggle
- K5: Reset Count

CTRIO Add Entry to End of Preset Table
CTRADPT IB-1005

CTRIO #	K0
Output #	K0
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Observe que el parámetro Pulse Time no tiene importancia para algunos tipos de entradas.

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.

Parámetros de CTADPT

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)
- **Output#:** Especifica una salida del módulo H0-CTRIO a ser usada por la instrucción
- **Entry Type:** Especifica un tipo de entrada a ser agregado al final de una tabla de valores predefinidos
- **Pulse Time:** Especifica un tiempo de un pulso para los tipos de entradas Pulse On y Pulse Off
- **Preset Count:** Especifica un valor inicial de conteo en que comenzará después de un Reset
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Entry Type V,K	K0-5; Vea el mapa de memoria V del DL06 - Data Words
Pulse Time V,K	K0-65535; Vea el mapa de memoria V del DL06 - Data Words
Preset Count V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRADPT

Renglón 1: Este ejemplo considera instalar el módulo H0-CTRIO en la ranura 2 del DL06. Cada H0-CTRIO en el sistema necesitará un IBox separada de CTRIO antes de que algún IBox de CTRxxxx pueda ser usado. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para los datos de entradas y V2030 hasta V2061 para sus datos de salidas.

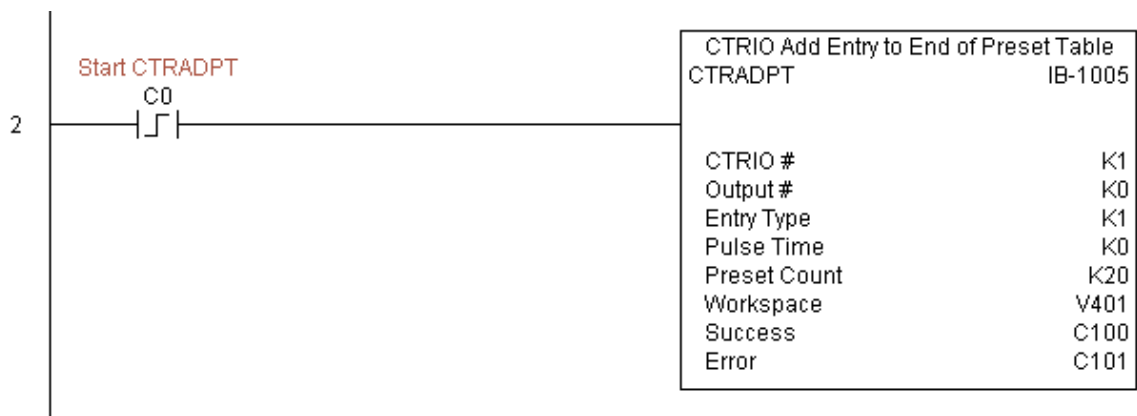


5

Renglón 2: Este renglón es un método de muestreo para permitir el comando de CTRADPT. Se usa un bit C para permitir que el programador controle un comando con Data View para propósitos de prueba.

Al activar C0 causará que la instrucción CTRADPT agregue una nueva tabla de valores predefinidos en la salida número 0 en el H0-CTRIO en la ranura 2. El nuevo valor predefinido serán un comando de RESET (tipo de entrada K1=Reset), y el tiempo de pulso es dejado en cero pues el tipo del reset no utiliza esto, y la cuenta en la cual él se reseteará será 20.

El procedimiento de funcionamiento para este ejemplo es cargar el archivo de CTRADPT_ex1.cwb a su H0-CTRIO, luego introducir el código mostrado aquí, cambiar al modo RUN, permitir la salida número 0 activando el bit C2 en Data View, gire le encoder conectado al módulo H0-CTRIO a un valor superior a 10 conteos y la salida y el LED de salida 0 se encenderá y permanecerá así en todos los conteos que estén arriba de 10. Ahora resetee el contador con el bit C1, habilite C0 para ejecutar la instrucción CTRADPT para dar un reset a la salida 0 cuando el conteo llegue a 20, encienda el bit C2 para activar la salida 0, y luego mueva el encoder a un valor mas grande que 10+ (La salida 0 debe encenderse) y luego continúe hasta que pase de 20+ (la salida 0 debe apagarse).



Ejemplo de CTRADPT (continuado)

Renglón 3: Este renglón le permite al programador que resetee el contador desde *DirectSOFT*.



Renglón 4: Este renglón le permite al programador que habilite la salida 0 desde **DirectSOFT**.



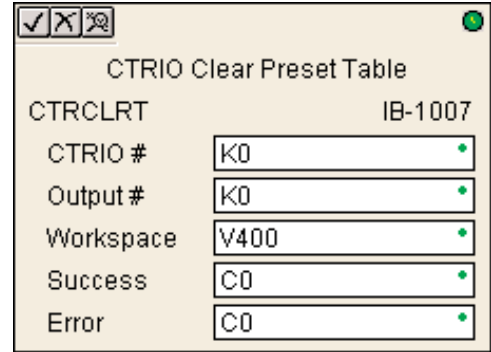
Limpia la tabla de valores predefinidos de CTRIO (CTRCLRT) (IB-1007)

DS5	Usado
HPP	N/A

Esta instrucción coloca en cero una tabla de valores predefinidos en la memoria RAM del PLC, en una transición de OFF para ON. Este IBox tomará más de un barrido del PLC para ejecutarse.

El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el código de error leído del IBox de CTRIO (CTRRDER) para obtener información adicional del error.

El registro Workspace (espacio de trabajo) es usado internamente y no debe ser usado en ningún otro lugar en el programa entero.



Parámetros de CTRCLRT

- **CTRIO#:** Especifica un módulo H0-CTRIO con un número definido por el usuario (Vea CTRIO Config)TRIO Config)
- **Output#:** Especifica una salida de un módulo H0-CTRIO a ser usado por la instrucción
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRCLRT

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado algún otro IBox CTRxxxx. El módulo H0-CTRIO se ha configurado para utilizar V2000 hasta V2025 para sus datos de entradas y V2030 con V2061 para sus datos de salidas.



Renglón 2: Este renglón es un método para permitir el comando de CTRCLRT. Se usa un bit C para permitir que el programador controle el comando desde Data View para propósitos de prueba.

Activando C0 causará que la instrucción de CTRCLRT limpie la tabla de valores predefinidos en la salida 0 en el H0-CTRIO en la ranura 2 .

El procedimiento de funcionamiento para este ejemplo es cargar el archivo de CTRCLRT_ex1.cwb a su módulo H0-CTRIO, luego introducir el código mostrado aquí, cambio al modo RUN, habilitar la salida 0 activando el bit C2 en Data View, gire el encoder conectado al módulo H0-CTRIO para obtener un valor mas grade que 10 y el LED de la salida 0 se encenderá y permanecerá encendida hasta que el conteo llegue a sobre 20, y en ese momento se apagará. Luego resetee el contador con el bit C1, habilite el bit C0 para ejecutar la instrucción CTRCLRT para hacer cero la tabla de valores predefinidos, active el bit C2 para habilitar la salida 0, y luego gire el encoder a un valor de conteo sobre 10+ (La salida 0 NO DEBE encenderse).



Ejemplo de CTRCLRT (continuado)

Renglón 3: Este renglón le permite al programador que resetee el contador desde *DirectSOFT*.



Renglón 4: Este renglón le permite al programador que habilite la salida 0 desde *DirectSOFT*.



Corregir una tabla de valores predefinidos de CTRIO (CTREDPT) (IB-1003)

Esta instrucción corrige, en una transición de APAGADO a ENCENDIDO a este IBox, una sola entrada de una tabla de valores predefinidos en un recurso específico de la salida de CTRIO. Este IBox es bueno si usted está corrigiendo más de una entrada en un archivo a la vez. Si usted desea hacer solamente una corrección y después recargar la tabla inmediatamente, vea el IBox CTRIO CTREDRL. Este IBox tomará más de un barrido del PLC para ejecutarse.

DS5	Usado
HPP	N/A

El bit de éxito o de error se activará cuando el comando sea completado. Si el bit de error está encendido, usted puede usar el IBox código de error leído de CTRIO (CTRRDER) para conseguir una información más detallada del error.

Tipo de entrada:

K0: Set

K1: Reset

K2: Pulse On (usa Pulse Time)

K3: Pulse Off (usa Pulse Time)

K4: Toggle

K5: Reset Count

Observe que el parámetro Pulse Time es ignorado por algunos tipos de entradas.

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.

CTRIO Edit Preset Table Entry	
IB-1003	
CTRIO #	K0
Output #	K0
Table #	V400
Entry # (0-based)	V400
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Parámetros de CTREDPT

- **CTRIO#:** Especifica un módulo H0-CTRIO con un número definido por el usuario (Vea CTRIO Config)
- **Output#:** Especifica una salida del módulo H0-CTRIO a ser usado by the instrucción
- **Table#:** Especifica un número de tabla en la cual será modificada un dato
- **Entry#:** Especifica la localización del dato en la tabla de valores predefinidos a ser modificada
- **Entry Type:** Especifica un tipo de entrada a ser agregado durante la modificación
- **Pulse Time:** Especifica un pulse time para tipos de entradas Pulse On y Pulse Off
- **Preset Count:** Especifica un valor inicial de conteo para comenzar después de un Reset
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Table# V,K	K0-255; Vea el mapa de memoria V del DL06 - Data Words
Entry# V,K	K0-255; Vea el mapa de memoria V del DL06 - Data Words
Entry Type V,K	K0-5; Vea el mapa de memoria V del DL06 - Data Words
Pulse Time V,K	K0-65535; Vea el mapa de memoria V del DL06 - Data Words
Preset Count V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTREDPT

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas, y V2030 hasta V2061 para sus datos de salidas.



(Este ejemplo continúa en la próxima página)

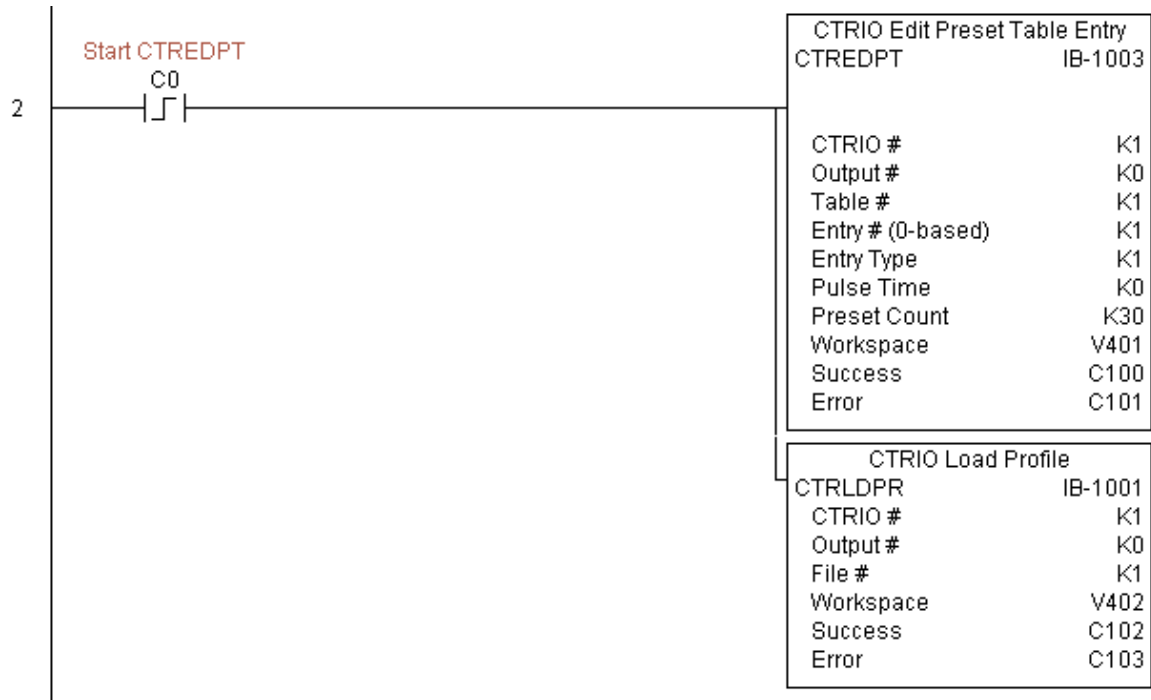
Ejemplo de CTREDPT (continuado)

Renglón 2: Este renglón es un método para permitir el comando de CTREDPT. Se usa un bit Co para permitir que el programador controle el comando desde Data View para propósitos de prueba.

Al activar el bit C0 causará que la instrucción CTREDPT cambie el segundo valor predefinido a un reset de un conteo en 20 a un reset en una conteo en 30 para la salida 0 en el H0-CTRIO en la ranura 2.

El procedimiento de funcionamiento para este ejemplo es cargar el archivo de CTREDPT_ex1.cwb al H0-CTRIO, luego colocar el código mostrado aquí, cambie el modo a RUN, permitir habilitar la salida 0 activando el bit C2 en Data View, luego gire el encoder conectado al módulo H0-CTRIO para obtener una valor sobre 10 y el LED de la salida 0 se encenderá y permanecerá encendido hasta cuando se llegue a un valor sobre 20, cuando se apagará. Luego haga un reset del contador con el bit C1, habilite el bit C0 para ejecutar la instrucción CTREDPT command para cambiar el segundo valor predefinido, active el bit C2 para habiliar la salida 0, y luego gire el encoder a un valor sobre 10+ (La salida 0 debería activarse) y luego continúe mas arriba de un conteo de 30 (La salida 0 debería apagarse).

Observe que debemos también cargar el perfil después de cambiar el o los valores predefinidos, esta es la razón por la cual la instrucción CTRLDPR sigue la instrucción CTREDPT en este ejemplo.



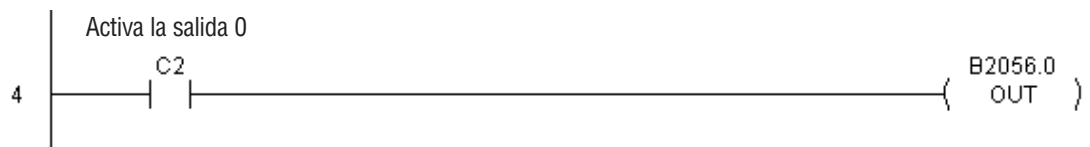
(Este ejemplo continúa en la próxima página)

Ejemplo de CTREDPT(continuado)

Renglón 3: Este renglón le permite al programador que resetee el contador desde *DirectSOFT*.



Renglón 4: Este renglón le permite al programador que habilite la salida 0 desde *DirectSOFT*.



Modificar una tabla de valores predefinidos de CTRIO (CTREDRL) (IB-1002)

DS5	Usado
HPP	N/A

Esta instrucción ejecutará la doble operación de modificar datos en una tabla de valores predefinidos y recargar a una salida de un módulo H0-CTRIO en una instrucción, en una transición de APAGADO a ENCENDIDO a este IBox. Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el IBox de CTRIO leer código de error (CTRRDER) para obtener información adicional del error.

Tipo de entrada:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Observe que el parámetro Pulse Time no tiene importancia para algunos tipos de entradas.

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.

CTRIO Edit Preset Table Entry and Reload	
CTREDRL IB-1002	
CTRIO #	K0
Output #	K0
Table #	V400
Entry # (0-based)	V400
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Parámetros de CTREDRL

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)
- **Output#:** Especifica una salida del módulo H0-CTRIO a ser usada por la instrucción
- **Table#:** Especifica un Table number of which an Entry is to be edited
- **Entry#:** Especifica un Entry location in the Preset Table to be edited
- **Entry Type:** Especifica el tipo de entrada a ser agregado durante la modificación
- **Pulse Time:** Especifica un tiempo del pulso para los tipos de entradas Pulse On y Pulse Off
- **Preset Count:** Especifica un valor inicial de conteo al comenzar después de un Reset
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Table# V,K	K0-255; Vea el mapa de memoria V del DL06 - Data Words
Entry# V,K	K0-255; Vea el mapa de memoria V del DL06 - Data Words
Entry Type V,K	K0-5; Vea el mapa de memoria V del DL06 - Data Words
Pulse Time V,K	K0-65535; Vea el mapa de memoria V del DL06 - Data Words
Preset Count V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTREDRL

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



(Este ejemplo continúa en la próxima página)

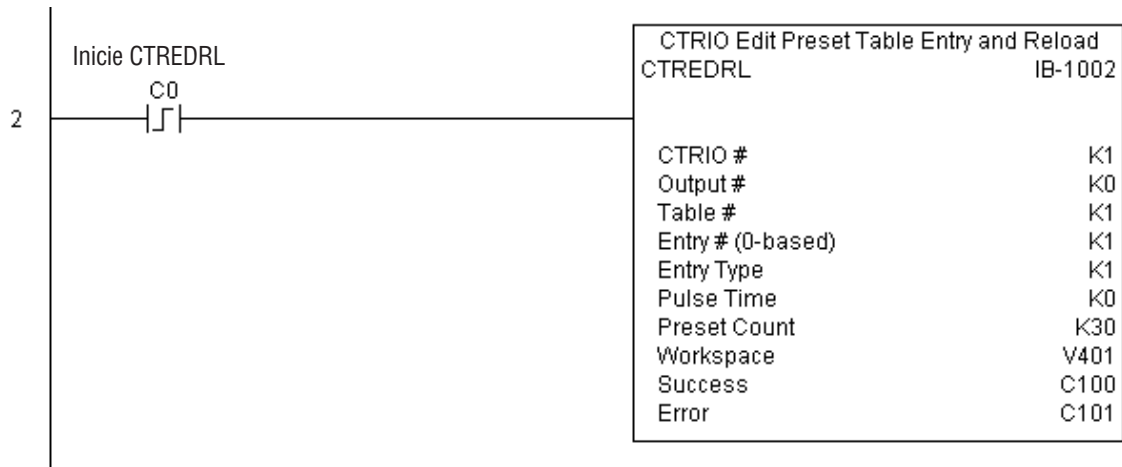
Ejemplo de CTREDRL (continuado)

Renglón 2: Este renglón es un método para permitir usar la instrucción CTREDRL. Se usa un bit C para permitir que el programador controle la instrucción desde Data View para propósitos de prueba.

Al activar el bit C0 causará que la instrucción CTREDRL cambie el segundo valor predefinido a un reset de un conteo en 20 a un reset en una conteo en 30 para la salida 0 en el H0-CTRIO en la ranura 2.

El procedimiento de funcionamiento para este ejemplo es cargar el archivo de CTREDRL_ex1.cwb al H0-CTRIO, luego colocar el código mostrado aquí, cambie el modo a RUN, luego habilite la salida 0 activando el bit C2 en Data View, luego gire el encoder conectado al módulo H0-CTRIO para obtener una valor sobre 10 y el LED de la salida 0 se encenderá y permanecerá encendido hasta cuando se llegue a un valor sobre 20, cuando se apagará. Luego haga un reset del contador con el bit C1, habilite el bit C0 para ejecutar la instrucción CTREDRL para cambiar el segundo valor predefinido a 30, y luego gire el encoder a un valor sobre 10+ (La salida 0 debería activarse) y luego continúe mas arriba de un conteo de 30 (La salida 0 debería apagarse).

Observe que no es necesario recargar el archivo separadamente. sin embargo, la instrucción puede cambiar un valor por vez.



(Este ejemplo continúa en la próxima página)

Inicializar una tabla de valores predefinidos de CTRIO (CTRINPT) (IB-1004)

DS5	Usado
HPP	N/A

Esta instrucción creará una Tabla de Valores Predefinidos en la memoria, pero como como archivo, en un recurso de salidas específica de CTRIO, en una transición de APAGADO a ENCENDIDO a este IBox. Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el IBox de CTRIO leer código de error (CTRRDER) para obtener para obtener información adicional del error.

Tipo de entrada:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Observe que el parámetro **Pulse Time** no tiene importancia para algunos tipos de entradas.

El registro **Workspace** (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.

CTRIO Initialize Preset Table	
CTRINPT	IB-1004
CTRIO #	K0
Output #	K0
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Parámetros de CTRINPT

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)
- **Preset Count:** Especifica un valor inicial de conteo al comenzar después de un Reset
- **Output#:** Especifica una salida del módulo H0-CTRIO a ser usada por la instrucción
- **Entry Type:** Especifica el tipo de entrada a ser agregado durante la modificación
- **Pulse Time:** Especifica un tiempo del pulso para los tipos de entradas Pulse On y Pulse Off
- **Preset Count:** Especifica un valor inicial de conteo al comenzar después de un Reset
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Entry Type V,K	K0-5; Vea el mapa de memoria V del DL06 - Data Words
Pulse Time V,K	K0-65535; Vea el mapa de memoria V del DL06 - Data Words
Preset Count V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRINPT

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.

5



(Este ejemplo continúa en la próxima página)

Ejemplo de CTRINPT(continuado)

Renglón 2: Este renglón es un método para permitir usar la instrucción CTRINPT. Se usa un bit C para permitir que el programador controle la instrucción desde Data View para propósitos de prueba.

Al activar el bit C0 causará que la instrucción CTRINPT cree una tabla de valores predefinidos pero no como archivo y la usará con la salida 0. En este caso, el valor predefinido será un SET cuando el conteo llegue a 15 para la salida 0.

El procedimiento de funcionamiento para este ejemplo es cargar el archivo de CTRINPT_ex1.cwb al módulo H0-CTRIO, luego colocar el código mostrado aquí, cambie el modo a RUN, luego habilite la salida 0 activando el bit C2 en Data View, luego gire el encoder conectado al módulo H0-CTRIO para obtener un valor sobre 15 y el LED de la salida 0 no se encenderá. Luego haga un reset del contador con el bit C1, habilite el bit C0 para ejecutar la instrucción CTRINPT para crear un único valor predefinido para hacer un set an contar 15, y luego gire el encoder a un valor sobre 15+ (La salida 0 debería activarse).

Observe que no es necesario recargar el archivo separadamente. Sin embargo, la instrucción puede cambiar un valor por vez.



(Este ejemplo continúa en la próxima página)

Ejemplo de CTRINPT (continuado)

Renglón 3: Este renglón le permite al programador que resetee el contador desde *DirectSOFT*.



Renglón 4: Este renglón le permite al programador que habilite la salida 0 desde *DirectSOFT*.



Inicializar una tabla de valores predefinidos en CTRIO (CTRINTR) (IB-1010)

Esta instrucción creará una sola entrada en la tabla de valores predefinidos en memoria pero no como archivo, en una transición de APAGADO a ENCENDIDO a este IBox.

DS5	Usado
HPP	N/A

Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el IBox de CTRIO leer código de error (CTRRDER) para obtener para obtener información adicional del error.

Entry Type:

K0: Set

K1: Reset

K2: Pulse On (uses Pulse Time)

K3: Pulse Off (uses Pulse Time)

K4: Toggle

K5: Reset Count

Observe que el parámetro **Pulse Time** no tiene importancia para algunos tipos de entradas..

El registro **Workspace** (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.

CTRIO Initialize Preset Table on Reset	
CTRINTR	IB-1010
CTRIO #	K0
Output #	K0
Entry Type	V400
Pulse Time	V400
Preset Count	V400
Workspace	V400
Success	C0
Error	C0

Parámetros de CTRINTR

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)
- **Output#:** Especifica una salida del módulo H0-CTRIO a ser usada por la instrucción
- **Entry Type:** Especifica el tipo de entrada a ser agregado durante la modificación
- **Pulse Time:** Especifica un tiempo del pulso para los tipos de entradas Pulse On y Pulse Off
- **Preset Count:** Especifica un valor inicial de conteo al comenzar después de un Reset
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Entry Type V,K	K0-5; Vea el mapa de memoria V del DL06 - Data Words
Pulse Time V,K	K0-65535; Vea el mapa de memoria V del DL06 - Data Words
Preset Count V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRINTR

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



(Este ejemplo continúa en la próxima página)

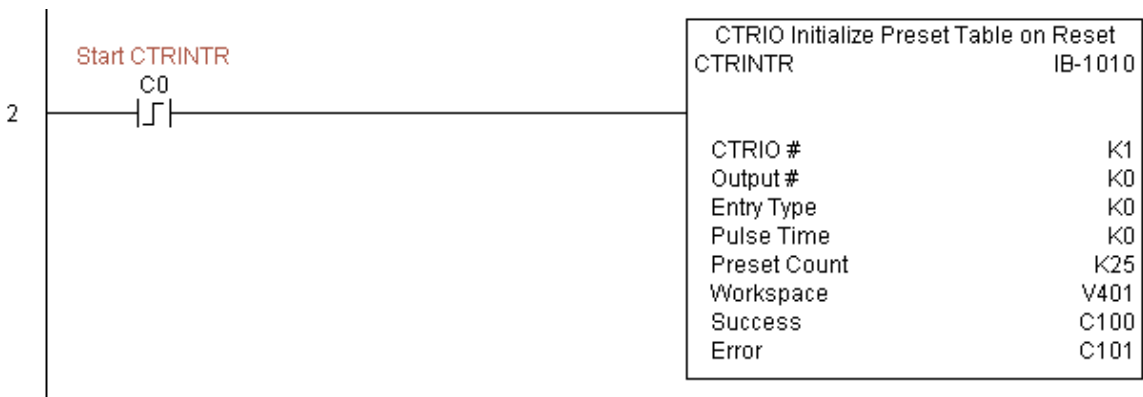
Ejemplo de CTRINTR(continuado)

Renglón 2: Este renglón es un método para permitir usar la instrucción CTRINTR. Se usa un bit C para permitir que el programador controle la instrucción desde Data View para propósitos de prueba.

Al activar el bit C0 causará que la instrucción CTRINTR cree una tabla única de valores predefinidos pero no como archivo y la usará con la salida 0. En este caso, el valor predefinido será un RESET cuando el conteo llegue a 25 para la salida 0.

El procedimiento de funcionamiento para este ejemplo es cargar el archivo de CTRINTR_ex1.cwb al módulo H0-CTRIO, luego coloque el código mostrado aquí, cambie el modo a RUN, luego habilite la salida 0 activando el bit C2 en Data View, luego gire el encoder conectado al módulo H0-CTRIO para obtener una valor sobre 1o y el LED de la salida 0 se encenderá. Luego active el bit C0 para ejecutar la instrucción CTRINTR y luego gire el encoder a un valor sobre 25+ (La salida 0 debería activarse).

Observe que no es necesario recargar el archivo separadamente. Sin embargo, la instrucción puede cambiar un valor por vez.



(Este ejemplo continúa en la próxima página)

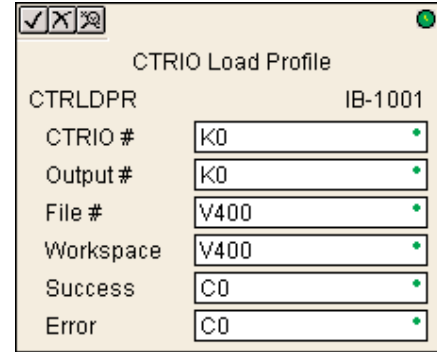
Cargar un perfil en CTRLDPR (IB-1001)

Esta instrucción carga un archivo de perfil de un módulo H0-CTRIO an un CTRIO Output resource en una transición desde OFF para ON a este IBox.

DS5	Usado
HPP	N/A

Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el IBox de CTRIO leer código de error (CTRRDER) para obtener para obtener información adicional del error

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.



5

Parámetros de CTRLDPR

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario(Vea CTRIO Config)
- **Output#:** Especifica una salida de CTRIO a ser usada por la instrucción
- **File#:** Especifica un número de archivo de CTRIO a ser cargado
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
File# V,K	K0-255; Vea el mapa de memoria V del DL06 - Data Words
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRLDPR

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



5

Renglón 2: Este IBox de carga el perfil de CTRIO cargará el archivo 1 en la memoria de trabajo de la salida 0 en el módulo H0-CTRIO 1. Este programa de ejemplo requiere que Ud. cargue CTRLDPR_IBox.cwb en su módulo H0-CTRIO.



(Este ejemplo continúa en la próxima página)

Ejemplo de CTRLDPR(continuado)

Renglón 3: Si el archivo es cargado con éxito, active el bit C1.



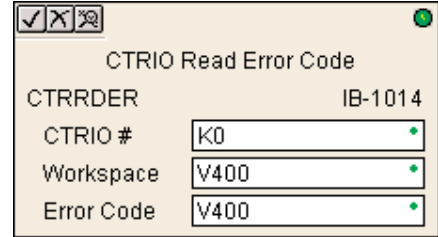
Lea error en CTRIO (CTRRDER) (IB-1014)

DS5	Usado
HPP	N/A

Esta instrucción obtendrá el valor de código decimal de error del módulo CTRIO (enumerado abajo) y lo pondrá en el registro dado del código de error, en una transición de APAGADO a ENCENDIDO al IBox.

Ya que el código de error en el CTRIO se mantiene solamente hasta que se da otro comando de CTRIO, usted debe utilizar esta instrucción inmediatamente después del IBox de CTRIO que entrega un error con el parámetro del bit de error.

El registro Workspace (espacio de trabajo) es para uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.



Códigos de error:

- 0: No hay error
- 100: El código de comando especificado está desconocido o sin apoyo
- 101: Número de archivo no encontrado en el sistema de archivos
- 102: El tipo del archivo es incorrecto para la función de salida especificada
- 103: El tipo del perfil es desconocido
- 104: La entrada especificada no se configura como límite en esta salida
- 105: El borde especificado de la entrada del límite está fuera de rango
- 106: La función de entrada especificada no está configurada o es inválida
- 107: El número especificado de la función de entrada está fuera de rango
- 108: La función Especificada de valor predefinida es inválida
- 109: La tabla de valores prefedidados está llena
- 110: La entrada especificada de la tabla está fuera de rango
- 111: El número especificado del registro está fuera de rango
- 112: El registro especificado es una entrada o salida que no está configurada
- 2001: Código de error de lectura de error - no puede tener acceso a CTRIO a través de ERM

Parámetros de CTRRDER

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario(Vea CTRIO Config)
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Error Code:** Especifica la localización en donde será escrito el código de error

Parámetro	Rango del DL06
CTRIO# K	K0-255
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Error Code V	Vea el mapa de memoria V del DL06 - Data Words

Ejemplo de CTRRDER

Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



Renglón 2: Este IBox que lee el código de error desde CTRIO leerá información detallada del error del módulo CTRIO número 1. Este programa ejemplo requiere que usted cargue CTRRDER_IBox.cwb en el módulo H0-CTRIO.



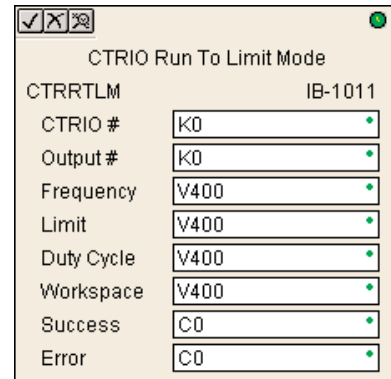
Modo Run to Limit del CTRIO (CTRRTLM) (IB-1011)

DS5	Usado
HPP	N/A

Esta instrucción carga el comando de RUN to Limit y los parámetros dados en un recurso específico de la salida en una transición de APAGADO a ENCENDIDO al IBox. Las entradas de CTRIO se deben configurar como Limit(s) para que esta función trabaje.

Valores Límites Hexadecimales Válidos:

- K00 - Borde de subida de Ch1/C
- K10 - Borde de caída de Ch1/C
- K20 - Ambos bordes de Ch1/C
- K01 - Borde de subida de Ch1/D
- K11 - Borde de caída de Ch1/D
- K21 - Ambos bordes de Ch1/D
- K02 - Borde de subida de Ch2/C
- K12 - Borde de caída de Ch2/C
- K22 - Ambos bordes de Ch2/C
- K03 - Borde de subida de h2/D
- K13 - Borde de caída de Ch2/D
- K23 - Ambos bordes de Ch2/D



Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede usar el IBox de leer el código de error de CTRIO (CTRRDER) para obtener información ós detallada del error.

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa. be usado anywhere else in your program.

Parámetros de CTRRTLM

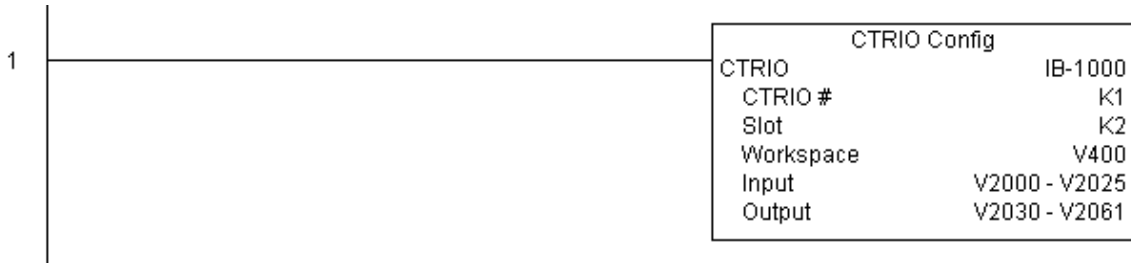
- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario(Vea CTRIO Config)
- **Output#:** Especifica la salida del móduloH0-CTRIO a ser usada por la instrucción
- **Frequency:** Especifica la frecuencia de pulsos de la salida (20Hz - 20KHz)
- **Limit:** Las entradas del H0-CTRIO se deben configurar como Limites para que esta función funcione
- **Duty Cycle:** Especifica el % de tiempo On en el ciclo completo. Esto es un número hexadecimal. El valor por defecto de 0 es el 50%, también entrando 50 resultará en 50%. 50% se define como mitad del tiempo ON y mitad del tiempo OFF
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Frequency V,K	K20-20000; Vea el mapa de memoria V del DL06 - Data Words
Limit V,K	K0-FF; Vea el mapa de memoria V del DL06 - Data Words
Duty Cycle V,K	K0-99; Vea el mapa de memoria V del DL06 - Data Words
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRRTLM

5

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



Renglón 2: Este IBox de CTRIO instala en la salida 0 en el módulo H0-CTRIO número 1 el hacer salir pulsos en una frecuencia de 1000 Hertz hasta que se llegue al Límite 0. Este programa ejemplo requiere que usted cargue CTRRTLM_IBox.cwb en su módulo Ho-CTRIO.



Ejemplo de CTRRTLM (continuado)

Renglón 3: Si los parámetros del modo Run to Limit es ACEPTABLE, active los bits de dirección y de habilitar la salida.



Modo Run to Position del CTRIO (CTRRTPM) (IB-1012)

Esta instrucción carga el comando Run to Position para colocar el comando y los parámetros dados en un recurso específico de la salida en una transición de APAGADO a ENCENDIDO a este IBox.

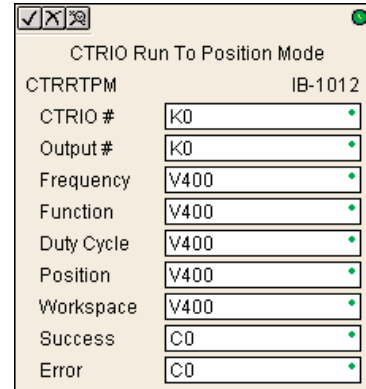
DS5	Usado
HPP	N/A

Los valores válidos de la función son:

- 00: Menor que Ch1/Fn1
- 10: Más grande que Ch1/Fn1
- 01: Menor que Ch1/Fn2
- 11: Más grande que Ch1/Fn2
- 02: Menor que Ch2/Fn1
- 12: Más grande que Ch2/Fn1
- 03: Menor que Ch2/Fn2
- 13: Más grande que Ch2/Fn2

Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede usar el IBox de leer el código de error de CTRIO (CTRRDER) para obtener información más detallada del error.

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa. be usado anywhere else in your program.



Parámetros de CTRRTPM

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)TRIO Config Ibox)
- **Output#:** Especifica la salida del módulo H0-CTRIO a ser usada por la instrucción
- **Frequency:** Especifica la frecuencia de pulsos de la salida (20Hz - 20KHz)
- **Duty Cycle:** Especifica el % de tiempo On en el ciclo completo. Esto es un número hexadecimal. El valor por defecto de 0 es el 50%, también entrando 50 resultará en 50%. 50% se define como mitad del tiempo ON y mitad del tiempo OFF
- **Position:** Especifica el valor de conteo, según lo medido en la entrada del codificador, en el momento en el cual el tren de pulsos de salida será apagado
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Frequency V,K	K20-20000; Vea el mapa de memoria V del DL06 - Data Words
Duty Cycle V,K	K0-99; Vea el mapa de memoria DL06
Position V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRRTPM

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



Ejemplo de CTRRTPM (continuado)

Renglón 2: Este IBox instala la salida número 0 en el CTRIO 1 para hacer salir pulsos en una frecuencia de 1000 Hertz, usa la comparación ' mayor que Ch1/Fn1', hasta que se alcanza la posición 1500 en la entrada. Este programa ejemplo requiere que usted cargue CTRRTPM_IBox.cwb en su módulo H0-CTRIO.



Renglón 3: Si los parámetros del modo Run to Position sons ACEPTABLES, activa los bits de dirección y de habilitar la salida.¶



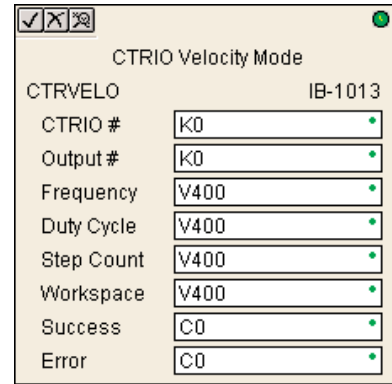
Modo de Velocidad de CTRIO (CTRVELO) (IB-1013)

DS5	Usado
HPP	N/A

Esta instrucción carga el comando de velocidad para colocar el comando y los parámetros dados en un recurso específico de la salida en una transición de APAGADO a ENCENDIDO a este IBox.

Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando el comando se haya completado. Si el bit de error está encendido, usted puede utilizar el IBox de CTRIO leer código de error (CTRRDER) para obtener para obtener información adicional del error.

El registro Workspace (espacio de trabajo) es para el uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.



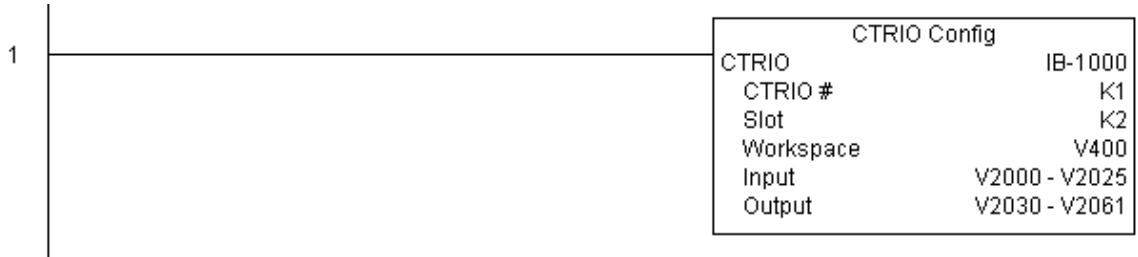
Parámetros de CTRVELO

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)TRIO Config Ibox)
- **Output#:** Especifica la salida del módulo H0-CTRIO a ser usada por la instrucción
- **Frequency:** Especifica la frecuencia de pulsos de la salida (20Hz - 20KHz)
- **Duty Cycle:** Especifica el % de tiempo On en el ciclo completo. Esto es un número hexadecimal. El valor por defecto de 0 es el 50%, también entrando 50 resultará en 50%. 50% se define como mitad del tiempo ON y mitad del tiempo OFF
- **Step Count:** Especifica la posición de la blanco como número 32-bit hexadecimal, un valor de Kffffff se causará que el perfil funcione continuamente mientras la salida sea habilitada
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO# K	K0-255
Output# K	K0-3
Frequency V,K	K20-20000; Vea el mapa de memoria V del DL06 - Data Words
Duty Cycle V,K	K0-99; Vea el mapa de memoria DL06
Step Count V,K	K0-2147434528; Vea el mapa de memoria DL06
Workspace V	Vea el mapa de memoria V del DL06 - Data Words
Success X,Y,C,GX,GY,B	Vea el mapa de memoria DL06
Error X,Y,C,GX,GY,B	Vea el mapa de memoria DL06

Ejemplo de CTRVELO

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la base del PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que pueda ser usado cualquier IBox de CTRxxxx. El módulo H0-CTRIO se ha configurado para usar V2000 hasta V2025 para sus datos de entradas y V2030 hasta V2061 para sus datos de salidas.



Renglón 2: Este IBox de modo de velocidad CTRIO configura la salida 0 en el módulo H0-CTRIO número 1 para generar 10,000 pulsos a una frecuencia de 1000 Hz. Este ejemplo de programa requiere que Ud. cargue CTRVELO_IBox.cwb en el módulo H0-CTRIO.



(Este ejemplo continúa en la próxima página)

Ejemplo de CTRVELO (continuado)

Renglón 3: Si los parámetros del Modo Velocidad están correctos, active el bit de dirección y habilite la salida.



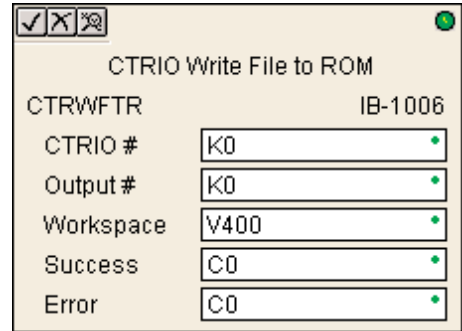
Escriba archivo a ROM en CTRIO (CTRWFTTR) (IB-1006)

DS5	Usado
HPP	N/A

Esta instrucción escribe los cambios runtime realizados a una tabla de valores predefinidos de H0-CTRIO a una memoria Flash-ROM en una transición de APAGADO a ENCENDIDO a este IBox. Este IBox tomará más de un barrido del PLC para ejecutarse. El bit de éxito o de error se activará cuando la instrucción se haya completado.

Si el bit de error está encendido, usted puede utilizar el IBox de lectura del código de error del módulo H0-CTRIO (CTRRDER) para conseguir una información más detallada del error.

El registro Workspace (espacio de trabajo) es para uso interno por esta instrucción de IBox y NO DEBE ser usado en cualquier otro lugar en su programa.



5

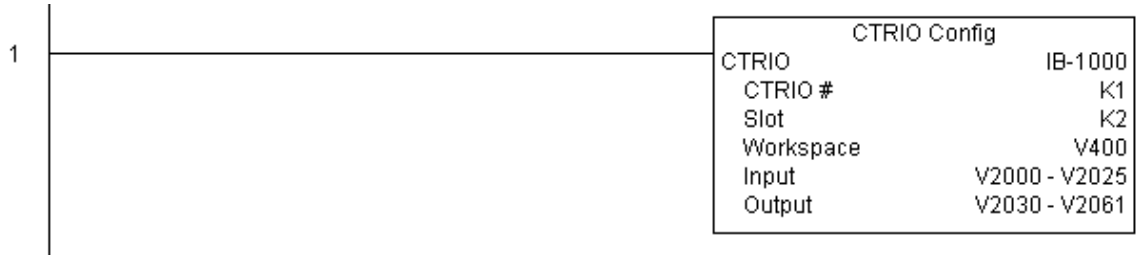
Parámetros de CTRWFTR

- **CTRIO#:** Especifica un módulo CTRIO con un número definido por el usuario (Vea CTRIO Config)
- **Output#:** Especifica una salida del módulo H0-CTRIO a ser usado por la instrucción
- **Workspace:** Especifica una localización de memoria V que es usada por la instrucción
- **Success:** Especifica un bit que se activa cuando la petición se completa con éxito
- **Error:** Especifica un bit que se activa cuando la requisición no se ha terminado con éxito

Parámetro	Rango del DL06
CTRIO#	K
CTRIO#	K0-255
Output#	K
Output#	K0-3
Workspace	V
Workspace	Vea el mapa de memoria V del DL06 - Data Words
Success	X,Y,C,GX,GY,B
Success	Vea el mapa de memoria DL06
Error	X,Y,C,GX,GY,B
Error	Vea el mapa de memoria DL06

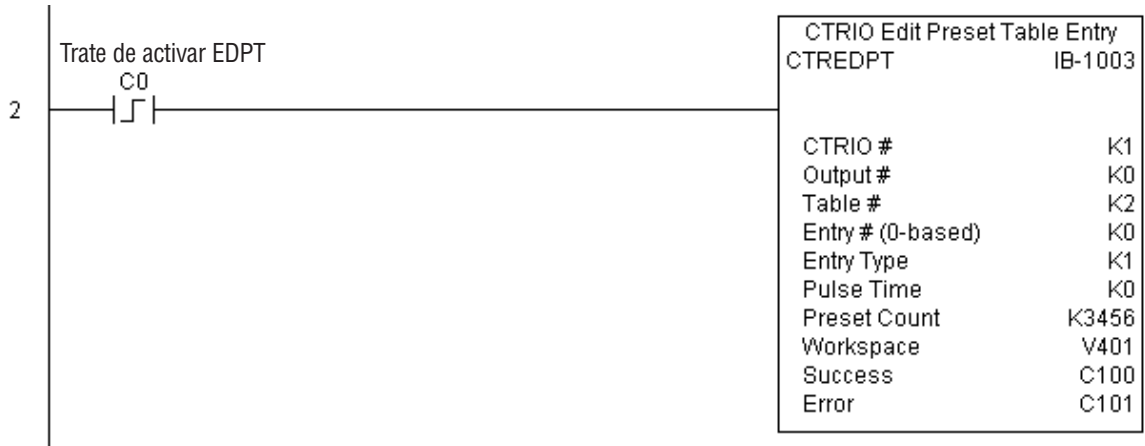
Ejemplo de CTRWFTR

Renglón 1: Este ejemplo instala el módulo H0-CTRIO en la ranura 2 de la basedel PLC. Cada H0-CTRIO en el sistema necesitará un IBox de CTRIO separado antes de que cualquier IBox de CTRxxxx pueda ser usado para él. El H0-CTRIO se ha configurado para utilizar V2000 hasta V2025 para sus datos de entradas, y V2030 hasta V2061 para sus datos de salidas.



5

Renglón 2: Este IBox de CTRIO cambiará la entrada 0 en la tabla #2 para ser un RESET en el conteo 3456. Este programa de ejemplo requiere que usted cargue CTRWFTR_IBox.cwb en su módulo H0-CTRIO.



(Este ejemplo continúa en la próxima página)

Ejemplo de CTRWFTR (continuado)

Renglón 3: Si el archivo se modifica con éxito, use un IBox de escribir archivo a ROM para almacenar la tabla corregida de nuevo a la ROM del módulo H0-CTRIO, de tal modo de hacer los cambios retentivos.

