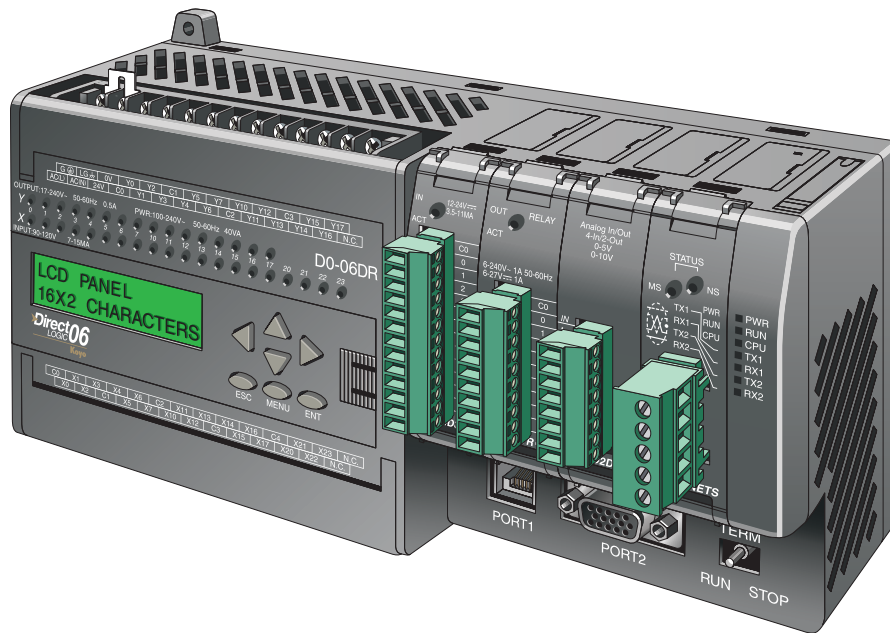


Manual del PLC DL06

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

Volumen 2 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
Etapas (o Stage) (SG)	7-21
Etapas iniciales (ISG)	7-22
El salto o JUMP (JMP)	7-22
No Salto (NJMP)	7-22
Etapas 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 seriales	11-55
Ejemplo 7: Uso de lazo de control PID	11-58

Apéndice A: Funciones auxiliares

Introducción	A-2
Propósito de las funciones auxiliares	A-2
Acceso a funciones AUX con <i>DirectSOFT</i>	A-3
Acceso a funciones AUX con el programador portátil	A-3
AUX 2* — Operaciones de programa ladder RLL	A-4
AUX 21 Verifique Programa	A-4
AUX 22 Cambie Referencia	A-4
AUX 23 Suprimir partes del programa	A-4
AUX 24 Borre el programa	A-4
AUX 3* — Operaciones de memoria V	A-4
AUX 31 Limpia la memoria V completa	A-4
AUX 4* — Configuración de entradas y salidas	A-4
AUX 41 Muestra la configuración de E/S	A-4
AUX 5* — Configuración de la CPU	A-5
AUX 51 Modifique el nombre del programa	A-5
AUX 53 Tiempo de barrido del PLC	A-5
AUX 54 Initialize Scratchpad	A-5
AUX 55 Configure el watchdog Timer	A-5
AUX 56 CPU Network Address	A-5
AUX 57 Configure rangos retentivos	A-6
AUX 58 Pruebe las operaciones	A-6
AUX 59 Bit Override	A-6
AUX 5B Configuración HSIO	A-7
AUX 5D Seleccione el modo de barrido del PLC	A-7
AUX 6* — Configuración del programador D2-HPP	A-8
AUX 61 Muestre el no. de versión	A-8
AUX 62 Zumbador activado o desactivado	A-8
AUX 65 Diagnósticos	A-8
AUX 7* — Operaciones con EEPROM	A-8
Áreas de memoria transferibles	A-8
AUX 71 CPU al EEPROM del programador	A-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

LAS INSTRUCCIONES DE TAMBOR (DRUM)



En este capítulo...

Introducción6-2
Transiciones de pasos6-4
Sumario de la operación tambor (DRUM)6-8
Técnicas de control de tambor6-10
La instrucción DRUM (tambor)6-12

Introducción

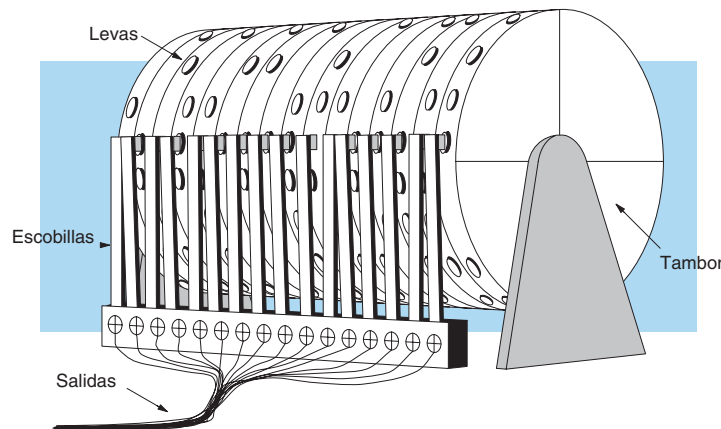
Propósito

La instrucción de tambor (EDRUM) en el PLC DL06 simula electrónicamente un secuenciador electromecánico de tambor con levas. La instrucción ofrece mejoras al principio básico, lo que describiremos primero.

Terminología del tambor

Las instrucciones de tambor se acomodan mejor para procesos repetitivos que se componen de un número finito de pasos. Pueden hacer el trabajo de muchos renglones de lógica ladder en una forma elegante que es muy simple. Por lo tanto, los tambores pueden salvar mucho tiempo de programación y depuración. Vamos a definir alguna terminología asociada con la instrucción de tambor describiendo el tambor mecánico original mostrado abajo. El tambor mecánico tiene generalmente levas en su superficie curva. Las levas se colocan en un cierto padrón, representando un conjunto de acciones deseadas para el control de máquina. Un motor o un solenoide gira el tambor una cantidad precisa en tiempos específicos. Durante la rotación, los contactos inmóviles sienten la presencia de levas (cuando la leva pasa = cierra el contacto, sin la leva = abre el contacto). Esta interacción entonces cierra o abre el contacto eléctrico con las levas, creando salidas eléctricas del tambor. Las salidas son alambradas a artefactos en una máquina para control discreto.

Los tambores tienen generalmente un número finito de posiciones en una rotación, llamado "paso". Cada paso representa alguna etapa del proceso. Durante la energización, el tambor vuelve a cierto paso determinado. El tambor gira de un paso al próximo basado en tiempo o en algún evento externo. Durante condiciones especiales, un operario de la máquina puede incrementar manualmente el paso del tambor con un control de "jog" (empujón) en el mecanismo de accionamiento del tambor. El cierre del contacto cada vez que pasa una leva engendra un perfil de acciones llamado una secuencia, diseñada para controlar una máquina específica. Ya que el tambor es circular, se repite automáticamente la secuencia una vez por



rotación. Las aplicaciones varían ampliamente, y un cierto tambor puede girar una vez por segundo, o tan lentamente como una vez por semana.

Los tambores electrónicos proporcionan las ventajas de tambores mecánicos y otras más. Por ejemplo, tienen una función de paso prefijado que es imposible en los tambores mecánicos: La función de paso prefijado le deja saltar desde el actual paso *directamente* a cualquier otro paso con un comando!

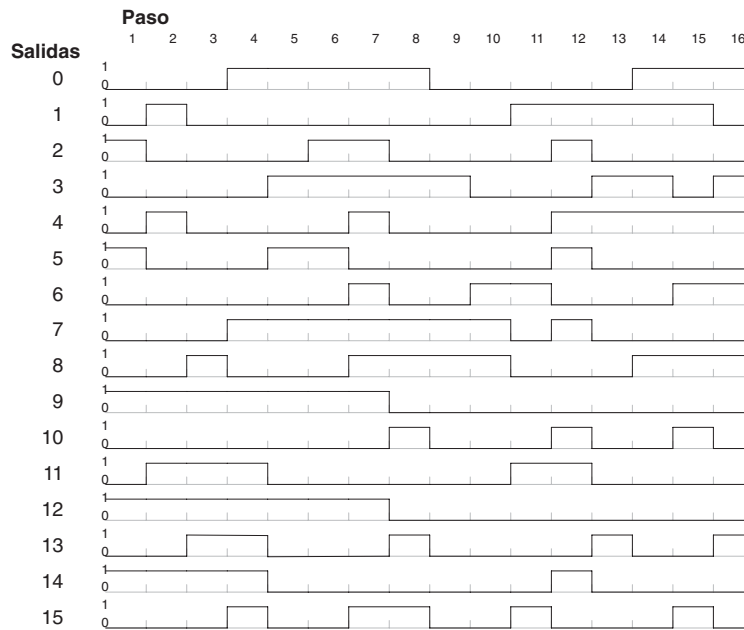
Representación de una tabla del tambor

Para propósitos de configuración, el tambor electrónico se presenta en forma de tabla en *DirectSOFT* y en este manual. Imagínese cortar la superficie de un cilindro hueco de tambor entre dos filas de levas y luego extenderlo en una superficie plana. Ahora usted puede ver el tambor como una tabla, mostrada abajo. Cada fila representa un paso, numerado 1 a 16. Cada columna representa una salida, numerada 0 a 15 (para simular cada bit de una palabra). Los círculos sólidos en la tabla representan levas (En el estado ON) en el tambor mecánico y los círculos abiertos son los sitios sin levas (Estado OFF).

PASO	SALIDAS															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	○	●	○	●	○	○	●	○	○	○	○	○	○	○	○	○
2	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
3	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
4	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
5	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
6	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
7	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
8	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
9	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
10	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
11	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
12	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
13	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
14	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
15	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
16	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

Secuencias de salida

El secuenciador mecánico de tambor deriva su nombre de sucesiones de cambios de control en sus salidas eléctricas. La figura siguiente muestra la secuencia de controles ON/OFF engendrados por el modelo de tambor mostrado mas arriba. ¡Compare los dos, y usted encontrará que ellos son equivalentes! Si usted puede ver su equivalencia, usted está encaminado para comprender la operación de la instrucción tambor o DRUM.



Transiciones de pasos

Tipos de instrucción de tambor

Hay dos tipos de instrucciones de tambor en el PLC DL06:

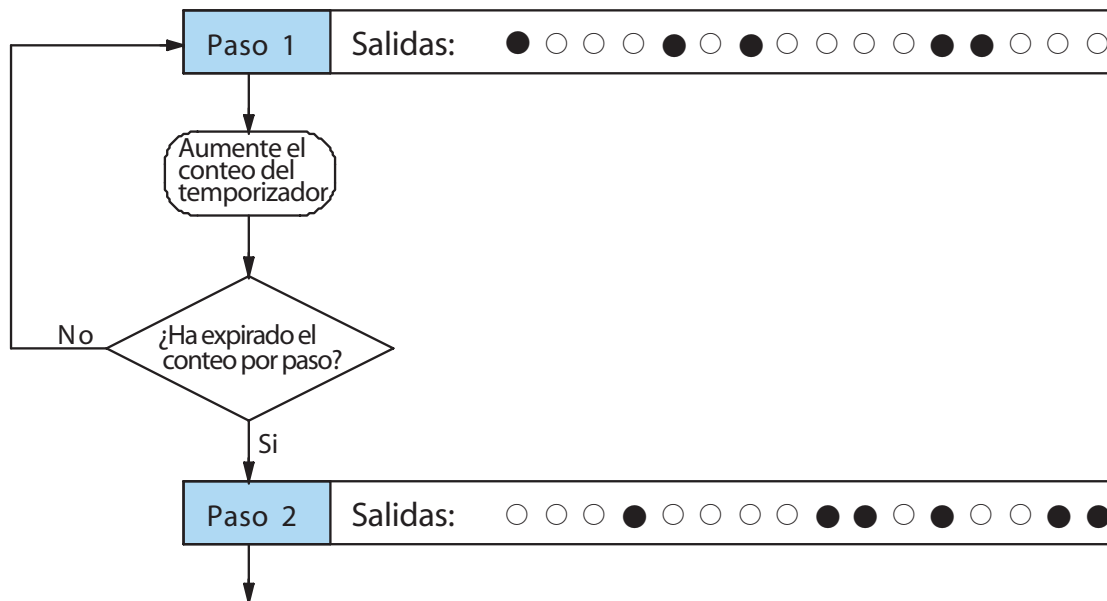
- Tambor temporizado con salidas discretas (TAMBOR)
- Tambor de tiempo y eventos con salidas discretas(EDRUM)

Las dos instrucciones de tambor incluyen transiciones basadas en tiempo y el EDRUM incluye también transiciones basadas en un evento. Cada tambor tiene 16 pasos y cada paso tiene 16 salidas. Vea la figura de abajo. Cada salida puede ser o una X, una Y o una bobina C, para ofrecer flexibilidad en la programación. Asignamos al paso 1 un perfil arbitrario único de salida.

Transiciones basadas en tiempo

El tambor se mueve de un paso a otro basado en tiempo y/o un evento (entrada) externo. Cada paso tiene su propia condición de transición que usted asigna durante la configuración de la instrucción de tambor. La figura de abajo muestra cómo trabajan las transiciones basadas en tiempo.

Use el próximo criterio de transición



Use el criterio de la próxima transición

El tambor permanece en el paso 1 por una duración específica (programable por el usuario). La base de tiempo del reloj es programable, de 0,01 segundos a 99,99 segundos. Esto establece la resolución o la duración de cada "tic del reloj". Cada paso usa la misma base de tiempo, pero tiene su propio conteo único por paso, que usted programa. Cuando los conteos para el paso 1 terminan el tambor se mueve para el paso 2.

El tambor gasta una cantidad específica de tiempo en cada paso, dado por la fórmula:

$$\text{Tiempo en el paso} = 0,01 \text{ segundos} \times \text{base de tiempo} \times \text{Cantidad}$$

Por ejemplo, si usted programa una base de tiempo de 5 segundos y a 12 conteos para el paso 1, entonces el tambor gastará 60 segundos en el paso 1. El tiempo máximo para cualquier paso es dado por la fórmula:

$$\begin{aligned} \text{Tiempo máximo por paso} &= 0,01 \text{ segundos} \times 9999 \times 9999 = 999,800 \text{ segundos} \\ &= 277,7 \text{ horas} = 11,6 \text{ días} \end{aligned}$$

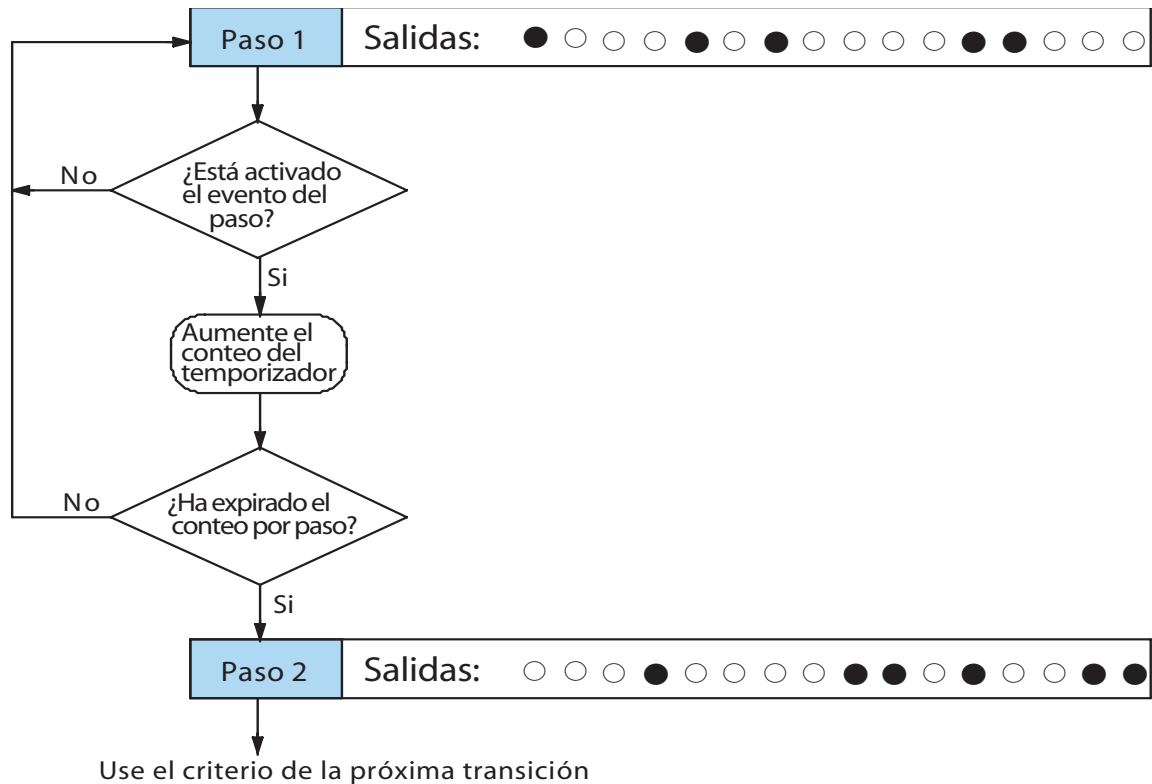


NOTA: Cuando escoja la resolución de base de tiempo por primera vez, una regla empírica apropiada es considerar cerca de 1/10 de la duración del paso más corto en su tambor. Entonces usted será capaz de optimizar la duración de ese paso en incrementos de 10%. Otros pasos con duraciones más largas permiten optimizar por incrementos aún más pequeños. También, note que la instrucción de tambor se ejecuta una vez por barrido de la CPU. Por lo tanto, no tiene sentido especificar una base de tiempo de tambor mucho más rápida que el tiempo de barrido de la CPU.

Transiciones de evento y tiempo

Pueden también ocurrir transiciones basados en eventos externos y/o en tiempo. La figura debajo muestra cómo trabajan las transiciones en estos casos.

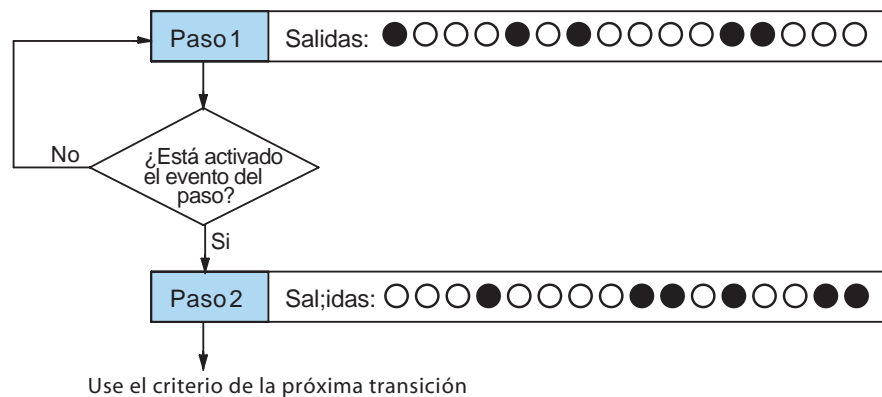
6



Quando el tambor entra al paso 1, hace el perfil de la salida como mostrado. Entonces comienza a interrogar la entrada externa programada para ese paso. Usted puede definir las entradas del evento como tipos X, Y, o C. Suponga que escogemos X0 para la entrada del evento del paso 1. Si X0 está apagado, entonces el tambor permanece en el paso 1. Cuando X0 está ON, los criterios del evento se cumplen y el temporizador comienza a contar el tiempo. El temporizador incrementa el tiempo mientras el evento (X0) es verdadero. Si el evento se hace OFF, el tiempo no vuelve a cero, y se mantiene el tiempo transcurrido. Cuando ha expirado el tiempo del paso 1, entonces el tambor se mueve al paso 2. Las salidas cambian para coincidir inmediatamente con el nuevo perfil del paso 2.

Transiciones de evento solamente

Para transiciones que no necesitan de eventos y tiempo programados para cada paso, Ud. tiene la opción de programar solamente uno de los dos, e incluso mezclar transiciones entre todos los pasos del tambor. Por ejemplo, usted quizás quiera el paso 1 con la transición en un evento particular, el paso 2 con transición por tiempo y el paso 3 a la transición a la vez por evento y por tiempo. Además, usted puede elegir usar sólo parte de los 16 pasos, y sólo parte de las 16 salidas.



6

Asignando contadores

Cada instrucción de tambor usa recursos de cuatro contadores en la CPU. Cuando se programa la instrucción de tambor, usted escoge el primer número de contador. El tambor usa también los próximos tres contadores automáticamente. El bit del contador asociado con el primer contador prende cuando el tambor ha completado su ciclo, y se hace 0 cuando el tambor vuelve a la condición inicial. Estos valores del contador y el bit del contador indican precisamente el progreso de la instrucción de tambor, y pueden ser controlado por su programa ladder.

Suponga que programamos un tambor para tener 8 pasos y escogemos CT10 para el número del contador (recuerde, el número del contador está en octal). El uso de contadores se muestra a la derecha. La columna a la derecha contiene los valores típicos, interpretados abajo.

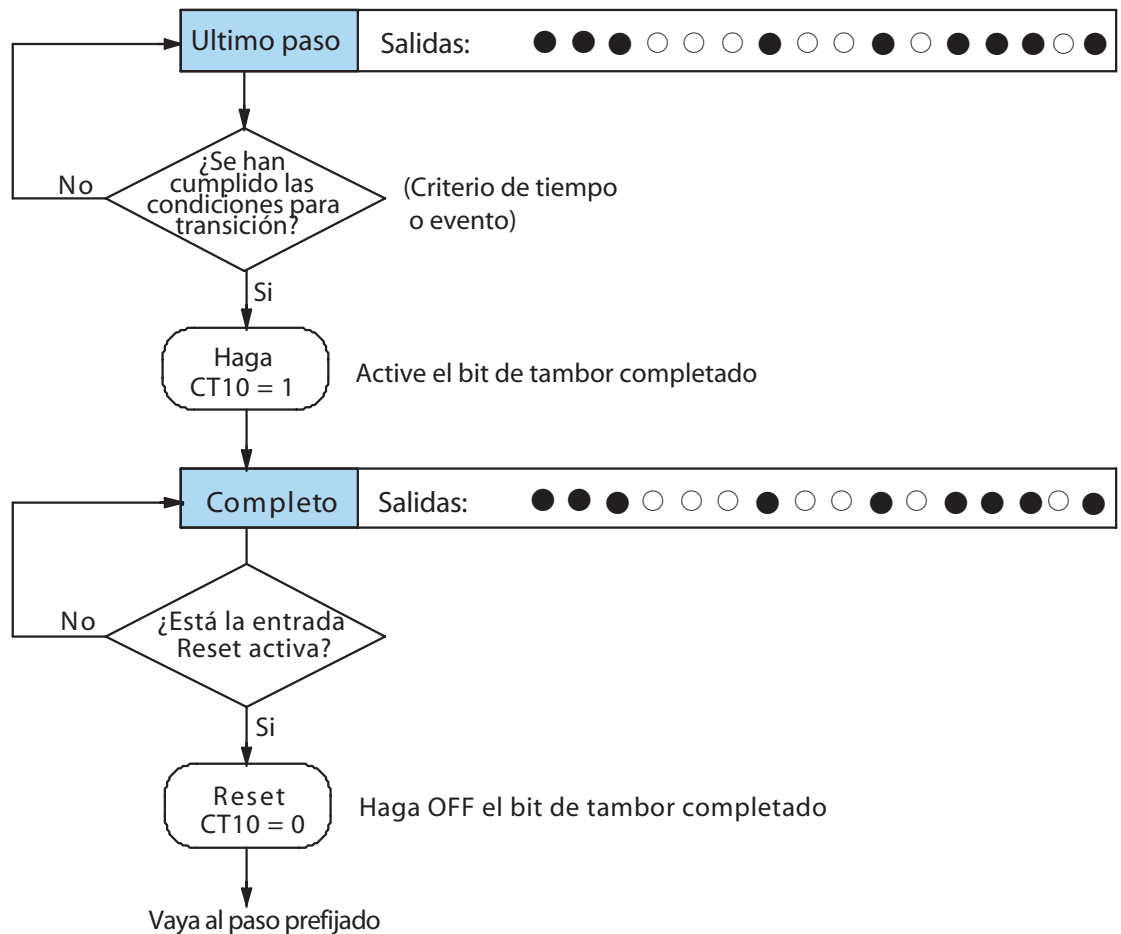
Asignación de contadores			
CT10	Conteos en el paso	V1010	1528
CT11	Valor del temporizador	V1011	0200
CT12	Valor prefijado del paso	V1012	0001
CT13	Paso corriente	V1013	0004

CT10 muestra que estamos en el conteo 1528 en el paso actual, que es el paso 4 (mostrado en CT13).

Si hemos programado el paso 4 para tener 3000 conteos, entonces el paso está solamente completado en solamente un poco más de la mitad. CT11 es el temporizador de conteo, mostrado en unidades de 0,01 segundos. De modo que cada cambio del dígito menos significativo representa 0,01 segundos. El valor 200 significa que hemos estado en la cuenta actual (1528) por 2 segundos (0,01 x 200). Finalmente, CT12 mantiene el valor prefijado del paso que se programó en la instrucción de tambor. Cuando la entrada Reset del tambor es activa, vuelve al paso 1 en este caso. El valor CT12 cambia sólo si el programa ladder le escribe un nuevo valor o la instrucción de tambor se modifica y el programa es iniciado nuevamente. El bit de estado del contador CT10 prende cuando el ciclo de tambor se completa y se apaga cuando el tambor vuelve a 0.

Terminación del último paso

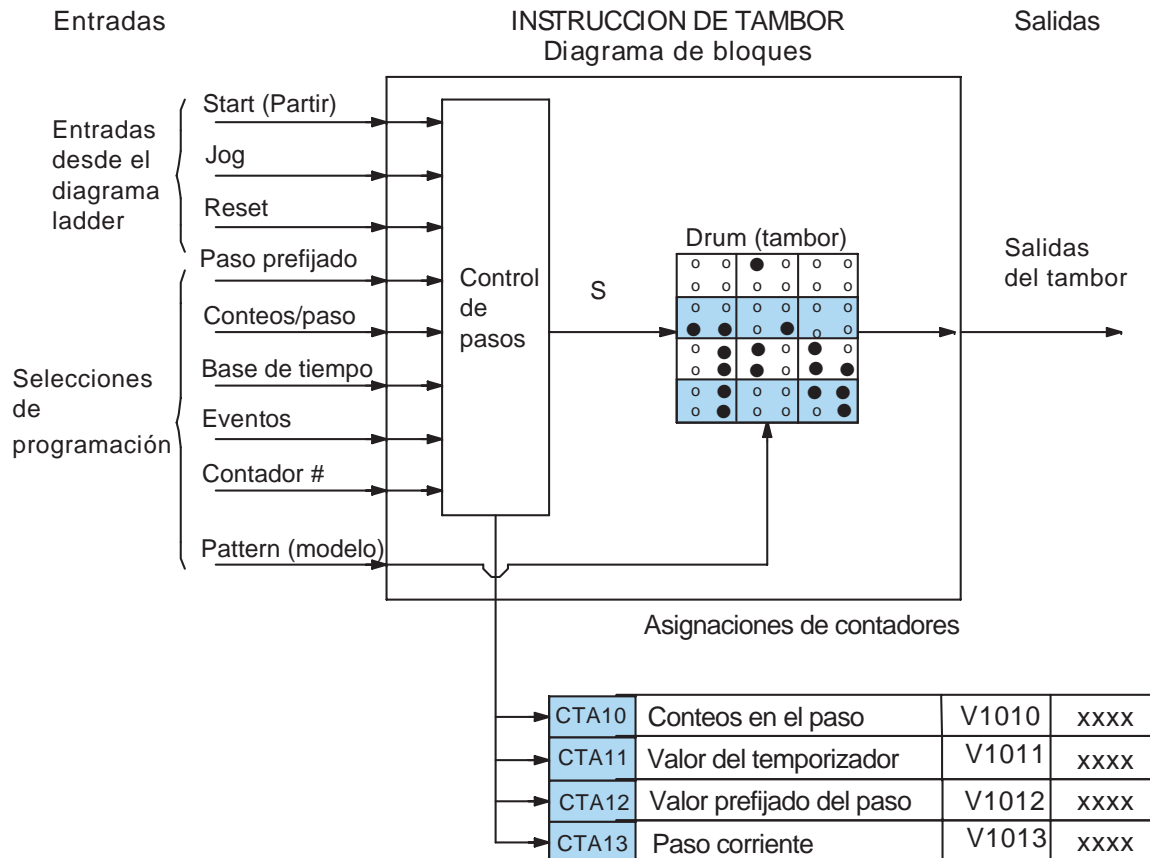
El último paso en una secuencia de tambor puede ser cualquier número de paso, ya que tambores parciales son válidos. Vea la figura siguiente. Cuando se cumplen las condiciones de transición del último paso, el tambor configura verdadero el bit de estado del contador que corresponde al mostrado en el bloque de la instrucción de tambor (tal como CT10). Entonces se mueve a un estado de "tambor completo". Las salidas del tambor permanecen en el modelo definido para el último paso. Al terminar un ciclo de tambor, las entradas de Comienzo y jog (Jog) no tienen efecto en este punto. El tambor deja el estado de "tambor completo" cuando la entrada Reset se hace activa (o en la transición del modo Program para Run). El bit de estado de tambor completo (tal como CT10) vuelve a cero, y luego va directamente al paso apropiado definido como el paso de valor prefijado.



Sumario de la operación del tambor (DRUM)

Esquema de bloque de la instrucción de tambor

La instrucción de tambor utiliza varias entradas y las salidas además del modelo del tambor. Vea la figura abajo.



La instrucción del tambor acepta varias entradas para el control de cada paso, el control principal del tambor. Las entradas y sus funciones son:

- **START**–La entrada "Inicio" es efectiva sólo cuando está apagada la señal RESET. Cuando Start es verdadero, el temporizador del tambor funciona y avanza si está en transición de tiempo, y el tambor busca la entrada del evento durante transiciones de evento. Cuando Start está apagado, el tambor para en su estado actual (RESET debe permanecer desactivado), y las salidas de tambor mantienen su estado corriente.
- **Jog** – La entrada de JOG sólo es efectiva cuando RESET está apagado (Start puede estar activado o no). La entrada de JOG incrementa el tambor al próximo paso en cada transición de OFF para ON (sólo EDRUM permite la entrada de JOG).
- **RESET** – La entrada RESET tiene prioridad sobre la entrada Start. Cuando RESET está activada, el tambor se mueve al paso prefijado. Cuando la señal RESET está apagada, la entrada START funciona normalmente.
- **Paso prefijado (Preset step)**- Este paso que usted define va del número de 1 al 16 (típicamente es el paso 1). El tambor se mueve a este paso cuando la señal RESET está activada y cuando la CPU entra al modo RUN.

- **Conteos por paso**(Counts/step) - El número de tics del temporizador que el tambor gasta en cada paso. Cada paso tiene su propio parámetro de conteo. Sin embargo, es opcional programar conteos por paso.
- **El valor de temporizador**(Timer Value)- el valor actual de los conteos por paso.
- **Counter #** – El número del contador especifica el primero de cuatro contadores consecutivos que el tambor usa para el control de pasos. Usted puede controlar éstos para determinar el progreso del tambor por su ciclo de control. El DL06 tiene 128 contadores (CT0 - CT177 en octal).
- **Events** – Una entrada X, Y, C, S, T, o CT sirve como entradas de transición a pasos. Cada paso tiene su propio evento. Sin embargo, programar el evento es opcional.



ADVERTENCIA: Las salidas de un tambor se habilitan cada vez que la CPU está en el Modo RUN. La Entrada Start no tiene que estar activada y la entrada RESET no deshabilita las salidas. Al entrar al Modo RUN, las salidas de tambor se activan o desactivan automáticamente según el perfil del paso actual del tambor. Este número inicial de paso depende de la configuración de la memoria de contador: no retentivo o retentivo.

Estado al energizar el PLC de las memorias de tambor

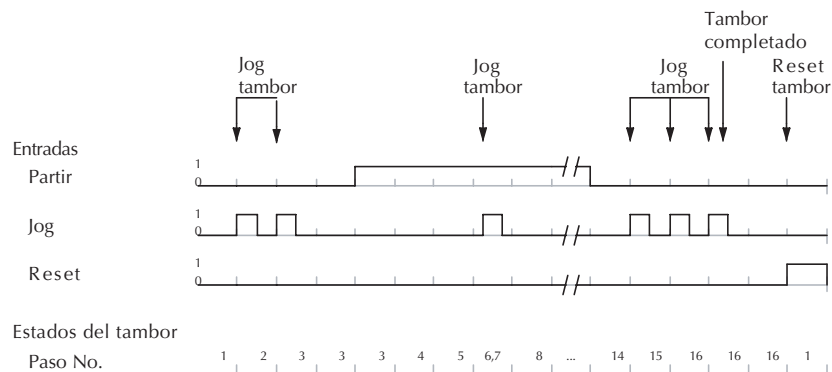
La elección del paso de inicio en la energización y la transición del modo Program a RUN es una consideración importante para su aplicación. Vea por favor la tabla siguiente. Si la memoria del contador se configura como no retentiva, el tambor se inicializa de la misma manera en cada energización o en la transición del modo Programa a RUN. Sin embargo, si la memoria del contador se configura para ser retentiva, el tambor permanecerá en su estado previo.

Contador número	Función	Inicialización en la energización	
		Sin retentividad	Con retentividad
CTA(n)	Conteo del paso corriente	Inicialice = 0	Use el previo (sin cambio)
CTA(n + 1)	Valor del temporizador del contador	Inicialice = 0	Use el previo (sin cambio)
CTA(n + 2)	Paso prefijado	Inicialice = Paso # prefijado	Use el previo (sin cambio)
CTA(n + 3)	No. del paso corriente	Inicialice = Paso # prefijado	Use el previo (sin cambio)

Las aplicaciones con tiempo relativamente rápido de ciclos de tambor necesitarán típicamente tener un RESET en la energización, usando la opción no retentiva. Las aplicaciones con tiempo relativamente largos de ciclo de tambor pueden necesitar reanudar en el punto previo donde paró la operación, usando el modo de memorias retentivas. La opción por defecto es el caso retentivo. Esto significa que si usted inicializa la memoria de scratchpad, la memoria será retentiva.

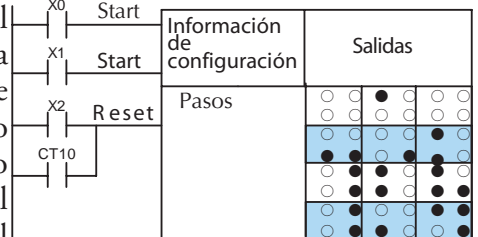
En la figura de abajo, nosotros nos enfocaremos en cómo la entrada Jog trabaja en tambores de evento. A la izquierda del esquema, note que la transición de OFF para ON de la entrada Jog incrementa el paso. El Start puede ser ON u OFF (sin embargo, RESET debe estar apagado). Dos activaciones de Jog llevan el tambor al paso tres. Luego, la entrada de Start prende, y el tambor comienza a funcionar normalmente. Durante el paso 6 ocurre otra señal de Jog. Esto lleva el tambor para el paso 7, poniendo el temporizador a 0. El tambor comienza a funcionar inmediatamente en paso 7, porque Start ya está ON. El tambor avanza al paso 8 normalmente.

Cuando el tambor entra al paso 14, la entrada Start se apaga. Dos señales de Jog más mueven el tambor para el paso 16. Sin embargo, note que se necesita una tercera señal de Jog para mover el tambor hasta el paso 16 a "tambor completado". Finalmente, la señal de entrada RESET fuerza el tambor en el paso prefijado y apaga el bit de tambor completado.



Como hacer para que el tambor vuelva al comienzo

A menudo hay aplicaciones que requieren tambores que comiencen de nuevo automáticamente una vez que se completa un ciclo. Esto se puede hacer fácilmente, usando el bit de tambor completado. En el reset de la figura a la derecha, la configuración de la instrucción de tambor es para CT10, de modo que hacemos la lógica OR con el bit de tambor completado (CT10) en la entrada RESET. Cuando se hace el último paso, el tambor prende CT10 que hace volver el tambor al paso prefijado, también volviendo a 0 el bit CT10. El contacto X2 trabaja también como un RESET manual.



Inicializando salidas del tambor

Las salidas de un tambor se habilitan cada vez que la que la CPU está en el modo RUN. En las transiciones del modo de programa a RUN, el tambor va al paso prefijado y las salidas se activan según el modelo de ese paso. Si su aplicación requiere que todas salidas estén apagadas durante la energización, haga el paso prefijado en el tambor una "etapa de vuelta a 0", con todas salidas OFF.

Usando transiciones de eventos complejas

Cada transición basada en eventos acepta sólo una referencia de contacto para el evento. Sin embargo, esto no limita los eventos a apenas un contacto. Para solucionar el impasse, use un contacto de relevador de control tal como C0 para la transición del evento del paso. En otra parte de la lógica ladder, usted puede usar C0 como una bobina de salida, haciéndolo dependiente de muchos otros "eventos" (contactos).

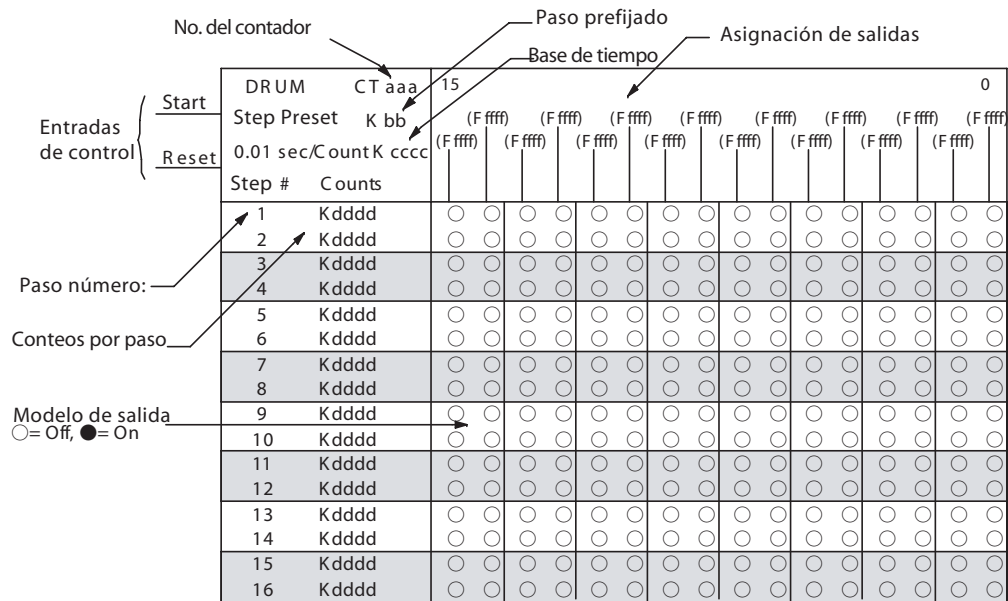
La instrucción DRUM (tambor)

Las instrucciones de tambor del PLC DL06 se pueden programar usando *DirectSOFT* o para la instrucción de EDRUM solamente se puede utilizar un programador (use al menos la versión de firmware V2.21). Esta sección cubre el como entrar las instrucciones usando *DirectSOFT* para todas las instrucciones más las nemotécnicas para la instrucción de EDRUM.

Tambor temporizado con salidas discretas(DRUM)

El tambor temporizado con salidas discretas es la más básica de las instrucciones de tambor. Opera según los principios descritos en las páginas anteriores. Debajo está la instrucción en una forma similar a como es mostrado en *DirectSOFT*.

El tambor tiene 16 pasos y 16 salidas. Las transiciones a otro paso ocurren solamente sobre la



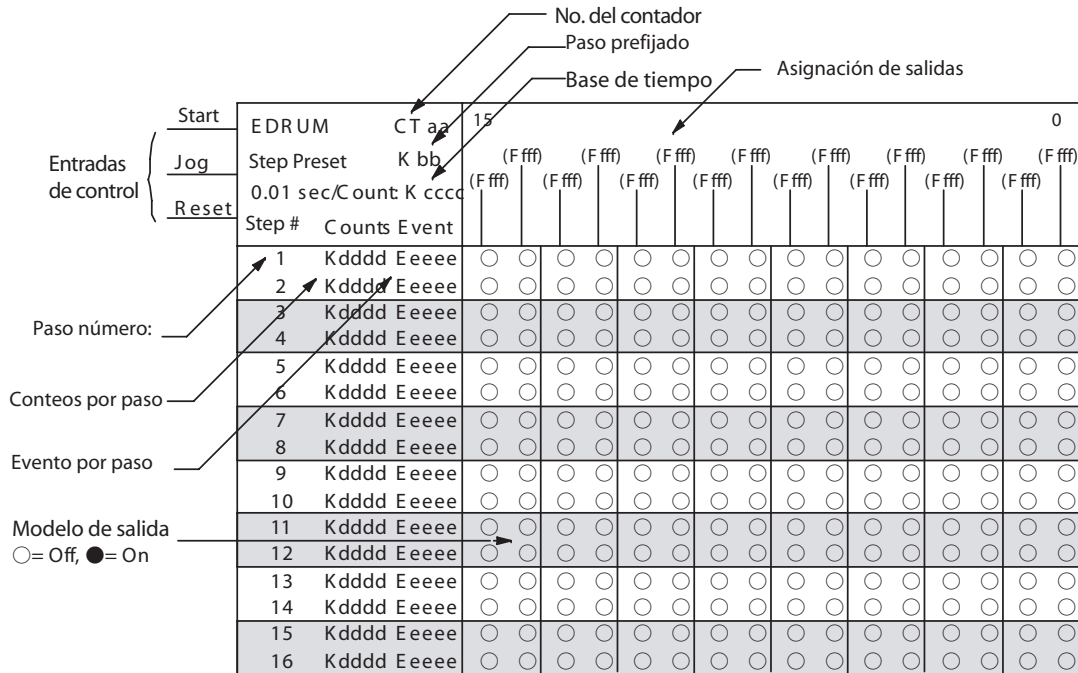
base del tiempo programado, especificado por los conteos por paso. Los pasos no usados se deben programar en "conteos por etapa" = 0 (esto es el valor por defecto). Los puntos de salidas discretos se pueden asignar individualmente como los tipos X, Y o C, o se pueden dejar sin usar. El perfil de salidas se puede modificar gráficamente con *DirectSOFT*.

Cuando la entrada Start es energizada, el temporizador del tambor se habilita. Se detiene cuando el último paso es completado o cuando la entrada RESET se activa . El tambor entra el paso prefijado escogido después de una transición de la CPU del modo Program a RUN y cuando la entrada RESET se activa.

Parámetros de tambor	Campo	Tipos de datos	Rangos
Número del contador	aaa	0 -174	0 -174
Paso prefijado (Preset step)	bb	K	1 - 16
Base de tiempo	cccc	K	0 -- 99,99 segundos
Conteos por paso	dddd	K	0 - 9999
Salidas discretas	Ffff	X, Y, C	Vea el mapa de memorias

El Tambor de eventos (EDRUM)

El tambor de eventos (EDRUM) tiene transiciones de pasos basadas en tiempo y en eventos. Opera según los principios generales de la operación de tambor descritos en el comienzo de este capítulo. Abajo está mostrada la instrucción similar a como aparece en *DirectSOFT*.



El tambor de eventos tiene 16 pasos y 16 salidas discretas . Las transiciones entre pasos ocurren basados en eventos o en tiempo o en las dos condiciones . La entrada Jog también avanza el paso cada transición de OFF para ON. El tiempo se especifica en conteos por paso y los eventos se definen con los contactos discretos. Los pasos y los eventos no usados se deben dejar en blanco. Las salidas discretas se pueden asignar individualmente.

Parámetros de Tambor	Campo	Tipo de datos	Rangos
Número del contador	aa	CT	0 - 174
Paso prefijado	bb	K	1 - 16
Base de tiempo	cccc	K	0 - 99.99 segundos
Conteos por paso	dddd	K	0 - 9999
Eventos	Eeeee	X, Y, C, S, T, CT, SP	Vea la tabla de memorias
Salidas discretas	Ffff	X, Y, C	Vea la tabla de memoria

Cuando la entrada Start se activa, se habilita el temporizador del tambor. Mientras el evento es verdadero para el paso actual, el temporizador funciona durante ese paso. Cuando el conteo del paso es igual a los conteos por paso, el tambor salta al próximo paso. Este proceso para cuando se llega al último paso o cuando la entrada RESET se activa. El tambor salta al paso prefijado escogido en la transición del modo Program a RUN de la CPU y en cualquier momento cuando la entrada RESET se activa.

Capítulo 6: Programación de las instrucciones DRUM

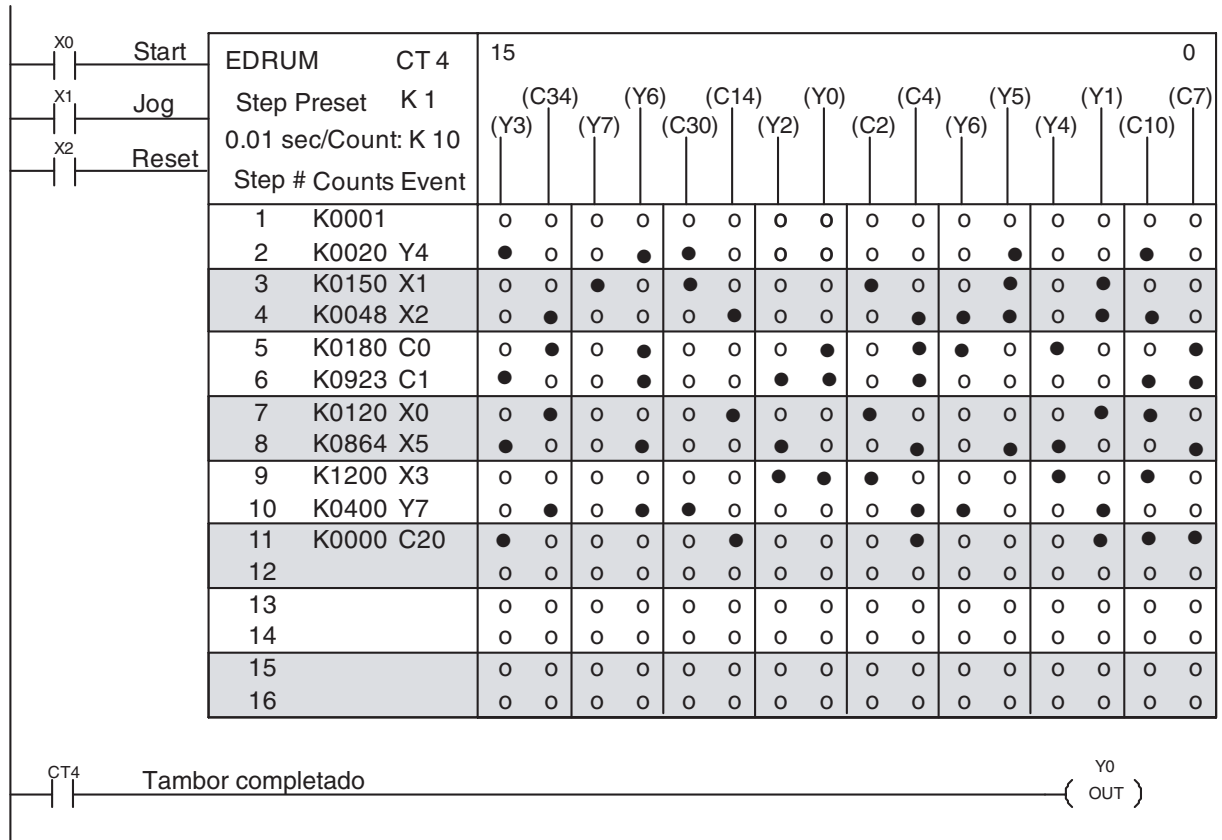
Las instrucciones de tambor usan cuatro contadores en la CPU . El programa ladder puede leer los valores de los contadores para determinar el estado del tambor. El programa ladder puede escribir un nuevo paso prefijado en CTA (n+2) en cualquier momento.

Contador número:	Rangos de (n)	Función	Función del bit del contador
CTA(n)	0 -- 174	Conteos en el paso	CT(n) = Tambor completado
CTA(n+1)	1 -- 175	Valor de tiempo	CT(n+1) = (no usado)
CTA(n+2)	2 -- 176	Paso prefijado	CT(n+2) = (no usado)
CTA(n+3)	3 -- 177	Paso corriente	CT(n+3) = (no usado)

El programa siguiente ladder muestra la instrucción EDRUM en un programa típico, similar a como es mostrado por *DirectSOFT*. Se usan los pasos 1 hasta 11 y las 16 salidas. El paso prefijado es el paso 1. El base de tiempo funciona como $(K10 \times 0,01) = 0,1$ segundo por conteo. Por lo tanto, la duración del paso 1 es $(1 \times 0,1) = 0,1$ segundo. Note que el paso 1 es basado solamente en tiempo (el evento se deja en blanco). Y el perfil de la salida para el paso 1 programa todas salidas OFF, que es una condición típicamente deseable de energización. En el ultimo renglón el bit de Tambor Completado (CT4) prende la salida Y0 al último paso (Paso 11). La entrada RESET también hace OFF a CT4.

6

DirectSOFT



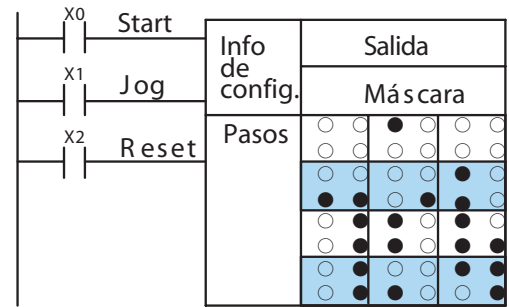
Nemotécnicos de la instrucción DRUM con el programador D2-HPP

La instrucción EDRUM se puede programar usando *DirectSOFT* o un programador D2-HPP. Esta sección cubre la entrada a través del programador.

Primero, entre las instrucciones Store para los renglones ladder que controlan las entradas del tambor. En el ejemplo a la derecha, las entradas Start, Jog y Reset del tambor de tiempo son controladas por X0, X1 y X2 respectivamente. Las secuencias de teclear requeridas se enumeran al lado de la nemotécnica.

Estos preceden la nemotécnica de la instrucción EDRUM. Observe que los renglones para las entradas Start, Jog y Reset no están limitadas a un contacto.

Después de la instrucción STR, entre el EDRUM (usando CT0) según lo mostrado:



Después de entrar el nemotécnico EDRUM como arriba, el programador crea un formulario de entrada para todos los parámetros del tambor. El formulario de entrada consiste en aproximadamente cincuenta o más entradas nemotécnicas por defecto que contienen declaraciones DEF (definir). Las nemotécnicas por defecto "están ya entradas", de modo que aparecen automáticamente. Use las teclas NXT y PREV para moverse para adelante y para atrás a través del formulario. Solamente se necesitan corregir los valores prefijados, así eliminando muchos tecleados. Las entradas requeridas para el tambor básico de tiempo están en la tabla abajo.

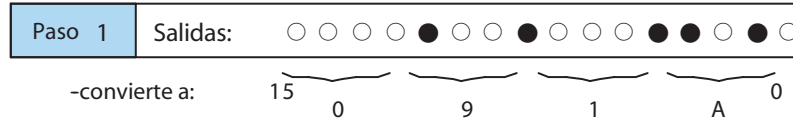


NOTA: Las entradas por defecto para los puntos y los eventos de salida son "DEF 0000", que significa que son no asignados. Si usted necesita ir para atrás y cambiar una salida asignada como no usada otra vez, entre "K0000". La entrada mostrará otra vez como "DEF 0000".

Parámetros de tambor	Entradas múltiples	Nemotécnico	Nemotécnico por defecto	Tipo de datos válidos	Rangos
Entrada Start	--	STR (más renglón de entrada)	--	--	--
Entrada Jog	--	STR (más renglón de entrada)	--	--	--
Entrada Reset	--	STR (más renglón de entrada)	--	--	--
Nemotécnico	--	DRUM CNT aa	--	CT	0 - 174
Paso prefijado	1	bb	DEF K0000	K	1 - 16
Base de tiempo	1	cccc	DEF K0000	K	1 -- 9999
Puntos de salidas	16	ffff	DEF 0000	X, Y, C	Vea mapa de memorias
Conteos por paso	16	dddd	DEF K0000	K	0 -- 9999
Eventos	16	dddd	DEF K0000	X, Y, C, S, T, CT, SP	Vea mapa de memorias
Modelo de salidas	16	gggg	DEF K0000	K	0 -- FFFF

Capítulo 6: Programación de las instrucciones DRUM

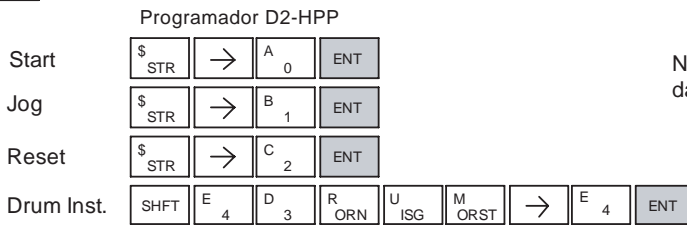
Usando la tabla de entradas del TAMBOR (dos páginas antes), mostramos el método de entrada para la instrucción básica del tambor de tiempos/eventos. Primero, convertimos el modelo de salidas para cada paso al número equivalente hexadecimal, según lo mostrado en el ejemplo siguiente:



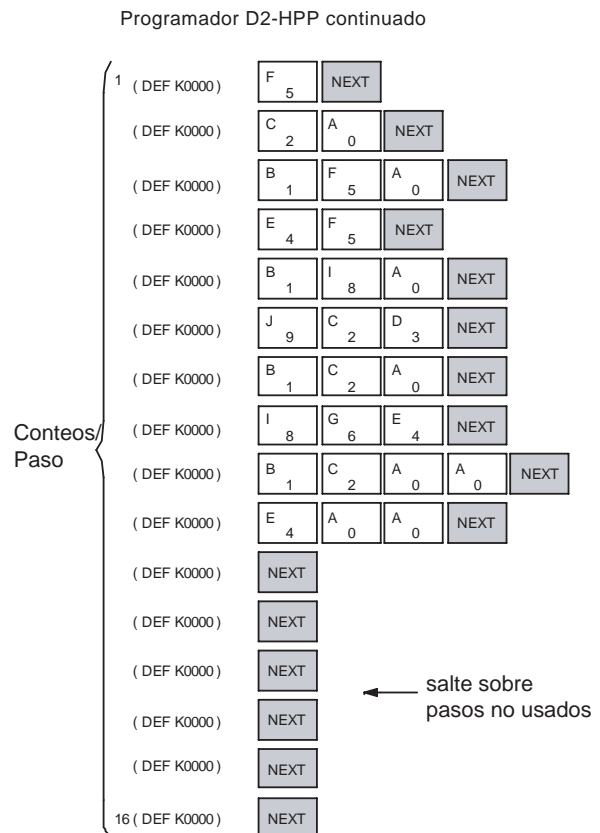
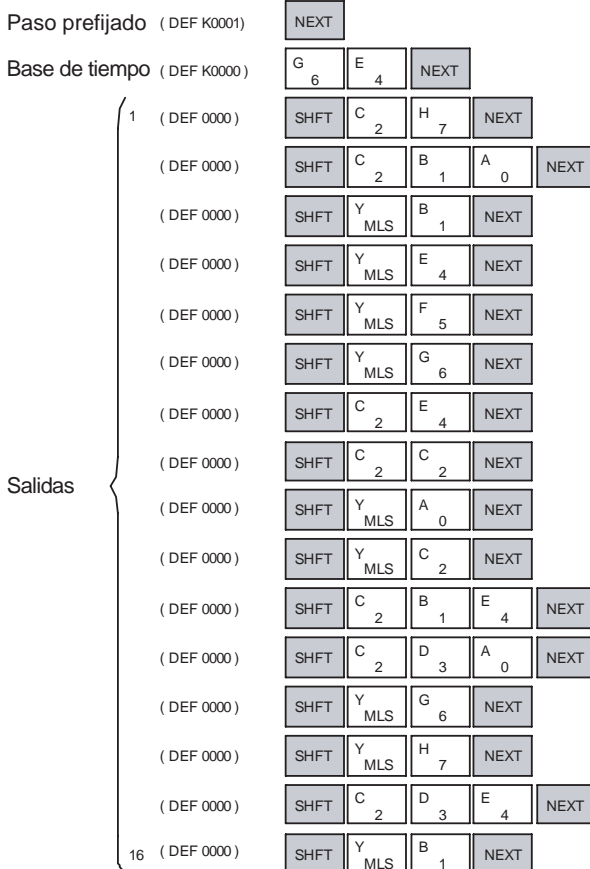
El diagrama siguiente demuestra el método para entrar el ejemplo anterior de EDRUM en el D2-HHP. Las entradas por defecto del formulario están en paréntesis. Después de entrar la instrucción del tambor (en la cuarta fila), los teclados restantes sobrescriben la porción numérica de cada declaración por defecto DEF.



NOTA: Se necesita por lo menos la versión 2.21 del firmware para hacer modificaciones en el D2-HPP. Se pueden usar las teclas NXT y PREV para saltar a las entradas para salidas o los pasos no usados.



Nota: Se pueden usar las teclas NXT y PREV para saltar datos ya entrados para pasos o salidas no usados..



(Continúa en la próxima página)

Programador D2-HPP continuado

Salidas	1	(DEF 0000)	NEXT	← S					
	(DEF 0000)	SHFT	Y	MLS	E	4	NEXT		
	(DEF 0000)	SHFT	X	SET	B	1	NEXT		
	(DEF 0000)	SHFT	X	SET	C	2	NEXT		
	(DEF 0000)	SHFT	C	2	A	0	NEXT		
	(DEF 0000)	SHFT	C	2	B	1	NEXT		
	(DEF 0000)	SHFT	X	SET	A	0	NEXT		
	(DEF 0000)	SHFT	X	SET	F	5	NEXT		
	(DEF 0000)	SHFT	X	SET	D	3	NEXT		
	(DEF 0000)	SHFT	Y	MLS	H	7	NEXT		
	(DEF 0000)	SHFT	C	2	C	2	A	0	NEXT
	(DEF 0000)	NEXT							
	(DEF 0000)	NEXT							
	(DEF 0000)	NEXT							
	(DEF 0000)	NEXT							
	16	(DEF 0000)	NEXT						

Modelo de salidas

Programador D2-HPP continuado.

Modelo de salidas	1	(DEF K0000)	NEXT	← Modelo del paso 1 = (
	(DEF K0000)	J	9	I	8	B	1	C	2	NEXT
	(DEF K0000)	C	2	I	8	J	9	E	4	NEXT
	(DEF K0000)	E	4	E	4	H	7	G	6	NEXT
	(DEF K0000)	F	5	B	1	G	6	J	9	NEXT
	(DEF K0000)	J	9	D	3	E	4	D	3	NEXT
	(DEF K0000)	E	4	E	4	I	8	G	6	NEXT
	(DEF K0000)	J	9	E	4	F	5	J	9	NEXT
	(DEF K0000)	D	3	I	8	SHFT	A	0	NEXT	
	(DEF K0000)	F	5	I	8	G	6	E	4	NEXT
	(DEF K0000)	I	8	E	4	E	4	H	7	NEXT
	(DEF K0000)	NEXT								
	(DEF K0000)	NEXT								
	(DEF K0000)	NEXT	← Pasos no usados							
	(DEF K0000)	NEXT								
	16	(DEF K0000)	NEXT							

Ultimo renglón

\$	GY	E	NEXT
STR	CNT	4	
SHFT	Y	A	NEXT
	MLS	0	

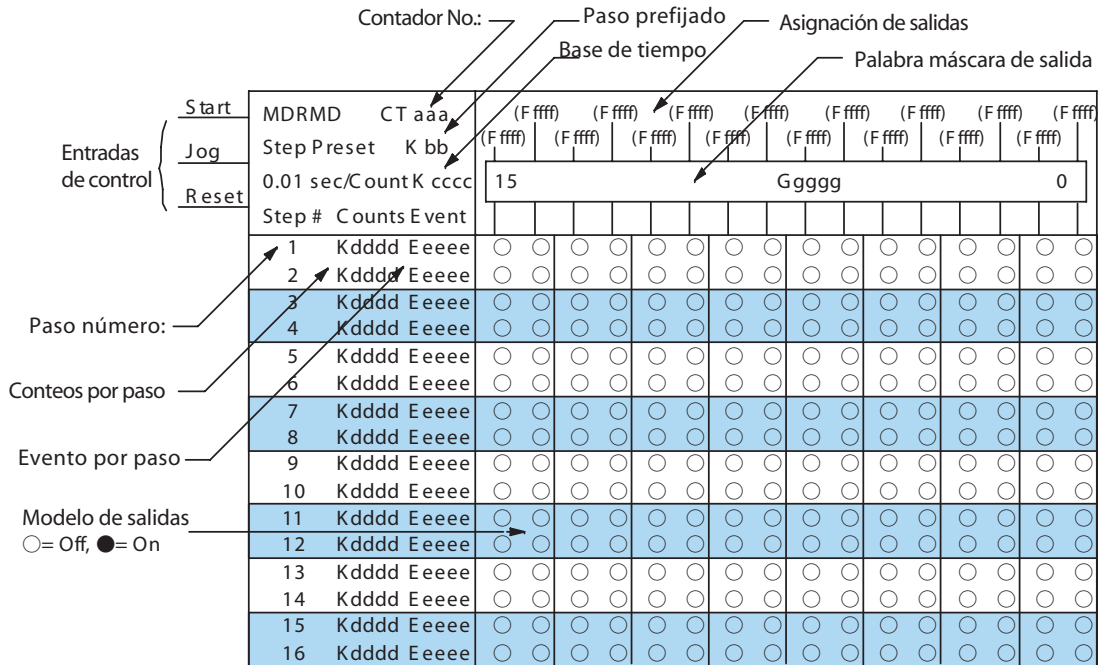
NOTA: Se pueden usar la teclas NXT y PREV para saltar datos ya entrados para pasos o paradas no usadas.

NOTA: Se pueden usar las teclas NXT y PREV para saltar más allá de lo entrado para las salidas o los pasos no usados.

NOTA: Para mayor comodidad de uso al usar la instrucción EDRUM, recomendamos la utilización de DirectSOFT en vez del programador.

Tambor de eventos con máscara en las salidas discretas (MDRMD)

La instrucción MDRMD tiene todas las características de control del tambor básico de eventos más control de las salidas finales para cada paso. Opera de acuerdo a los principios generales de la operación de tambor descrito en el comienzo de este capítulo. Abajo se muestra la instrucción en la forma de tabla similar a como aparece en *DirectSOFT*.



El tambor de eventos con máscara en las salidas tiene 16 pasos y 16 salidas. Las salidas del tambor son operadas AND (recuerde la función AND) bit por bit lógicamente con una palabra de máscara de salida para cada paso. El campo **Ggggg** especifica la dirección del principio de la tabla con las 16 palabras de máscara. La transición de pasos ocurre en base de eventos o tiempo. La entrada **Jog** avanza también el paso en la transición de OFF para ON. El tiempo se especifica en conteos por paso y los eventos se definen con los contactos discretos. Los pasos y los eventos no usados se pueden dejar en blanco (esto es, la entrada por defecto). Cuando se activa la entrada **Start**, se habilita el temporizador del tambor. Si el evento es verdadero para el paso actual, el tiempo funciona durante ese paso. Cuando el conteo del paso es igual a los conteos por paso hay una transición del tambor al próximo paso. Este proceso se detiene cuando el último paso es completado o cuando la entrada **RESET** se activa. El tambor va al paso prefijado escogido en la transición del modo Programa para RUN de la CPU y cuando la entrada **RESET** se activa.

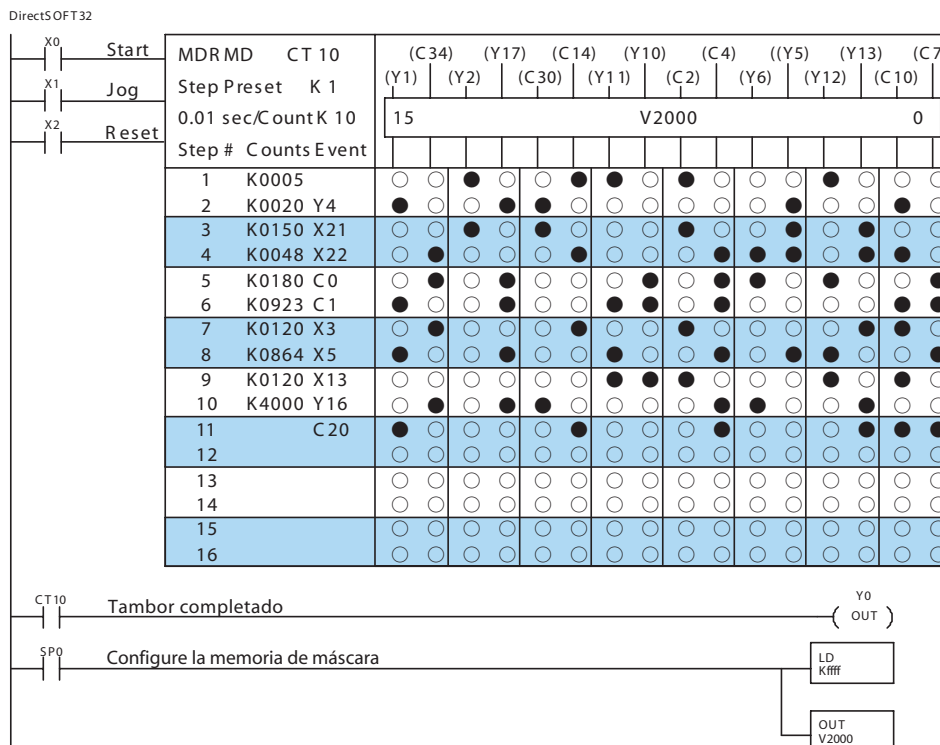
Parámetros de Tambor	Campo	Tipo de datos	Rangos
Número del contador	aaa	CT	0 – 174
Base de tiempo	bb	K	1 – 16
Timer base	cccc	K	0 – 99,99 segundos
Conteos por paso	dddd	K	0 – 9999
Eventos	eeee	X, Y, C, S, T, ST, GX, GY, CT, SP	Vea el mapa de memorias
Salidas discretas	Fffff	X, Y, C, GX, GY	
Máscara de salida	Ggggg	V	

Capítulo 6: Programación de las instrucciones DRUM

Las instrucciones del tambor usan cuatro contadores en la CPU. El programa ladder puede leer los valores del contador para determinar el estado del tambor. El programa ladder puede escribir un nuevo número de paso prefijado a CTA(n+2) en cualquier momento. Los otros contadores son solamente para propósitos de supervisión.

Contador número:	Rangos de (n)	Función	Función del bit de estado
CTA(n)	0 – 174	Conteos por paso	CT(n) = tambor completado
CTA(n+1)	1 – 175	valor de tiempo	CT(n+1) = (no usado)
CTA(n+2)	2 –176	paso prefijado	CT(n+2) = (no usado)
CTA(n+3)	3 –177	Paso corriente	CT(n+1) = (no usado)

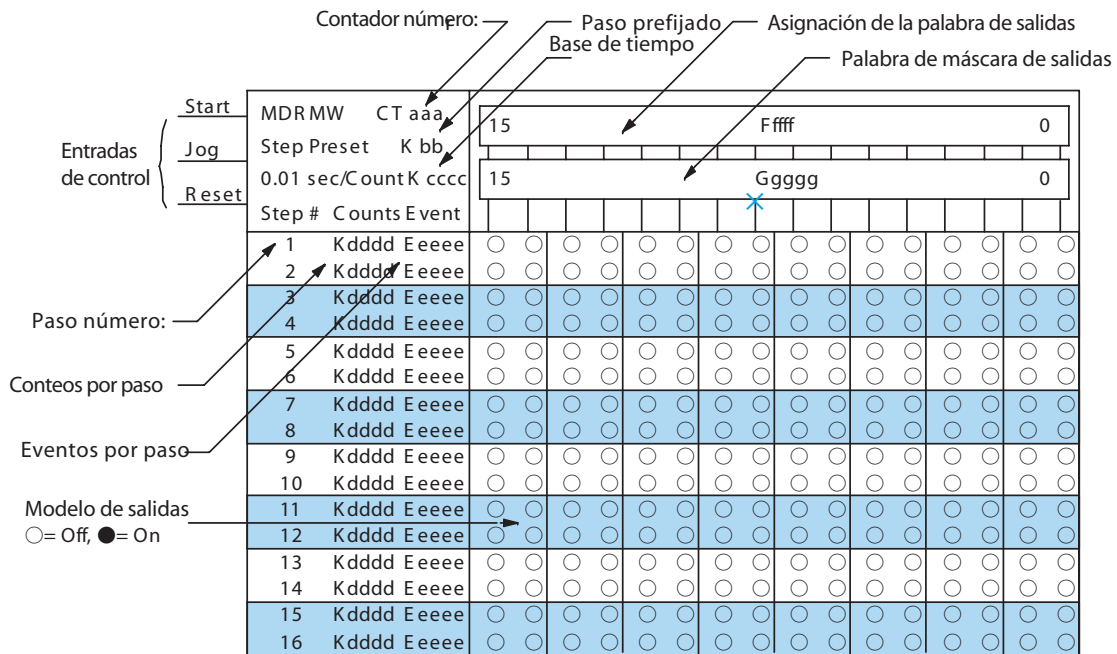
El programa siguiente ladder muestra la instrucción MDRMD en un programa típico similar a como es mostrado por *DirectSOFT*. Se están usando los pasos 1 hasta 11 y 16 salidas. La palabra de la máscara de salidas está en V2000. Las salidas finales del tambor se muestran arriba de la palabra de máscara como bits individuales. Los bits de datos en V2000 son operados AND en forma lógica con el perfil de salida del paso corriente en el tambor (esto es, solo será verdadera la salida si el bit de la palabra es 1). Si usted quiere que todas salidas de tambor estén apagadas después de la energización, escriba ceros en V2000 en el primer barrido. La lógica ladder puede actualizar la máscara de salidas en cualquier momento para habilitar o incapacitar las salidas de tambor. el paso prefijado es el paso 1. La base de tiempo funciona en (K10x0,01)=0,1 segundo por conteo. Por lo tanto, la duración del paso 1 es (5 x 0,1) = 0,5 segundos. Note que el paso 1 es basado de tiempo sólo (el evento se deja blanco). En el penúltimo renglón, el bit de tambor completado (CT10) prende la salida Y0 cuando termina el último paso (Paso 10). La entrada RESET del tambor también vuelve a OFF CT10.



NOTA: El programa ladder debe cargar las constantes en V2000 hasta V2012 para cubrir todas las palabras de máscara para los once pasos usados en este tambor.

El tambor de eventos con máscara de palabra en las salidas (MDRMW)

El Tambor de Eventos con máscara de palabra en las salidas representa las salidas organizadas como bits de una sola palabra, en vez de puntos discretos. Opera según los principios generales de la operación de tambor descritos en el comienzo de este capítulo. Abajo está la instrucción en la forma de tabla, similar a como es mostrado por *DirectSOFT*.



La instrucción MDRMW tiene 16 pasos y 16 salidas. Las salidas del tambor son operadas "AND" en forma lógica bit a bit con una palabra de máscara de salidas para cada paso. El campo **Ggggg** especifica la dirección inicial de las 16 palabras de máscara, creando la salida final (campo de Ffff). Las transiciones de eventos ocurren basadas en eventos o por tiempo. La entrada de Jog avanza también un paso en la transición de la señal de OFF para ON. El tiempo se especifica en conteos por paso, y los eventos se especifican como contactos discretos. Los pasos y eventos no usados se pueden dejar en blanco (esto es la entrada por defecto).

Cuando se activa la entrada START, se activa el temporizador del tambor. Mientras el evento sea verdadero para el paso corriente, el **temporizador** funciona durante ese paso. Cuando el conteo del paso es igual a los conteos por paso el tambor hace la transición al próximo paso. Este proceso para cuando se completa el último paso o cuando se activa la entrada RESET. El tambor entra al paso prefijado después de una transición del modo Program para RUN, y cuando se activa la entrada RESET.

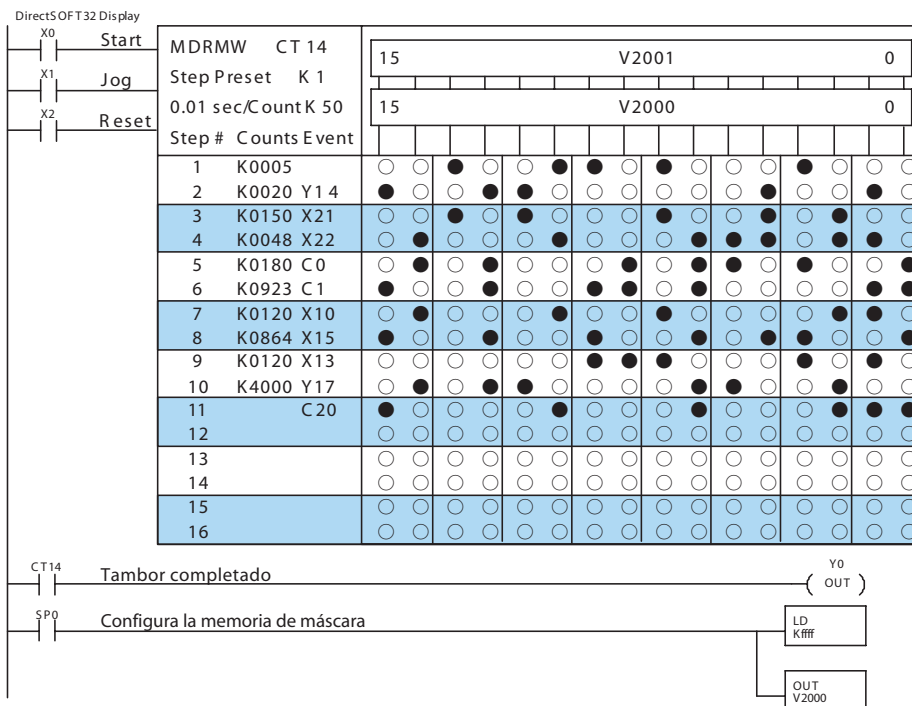
Parámetros de tambor	Campo	Tipo de datos	Rangos
Contador Number	aaa	CT	0 – 174
Paso prefijado	bb	K	1 – 16
Base de tiempo	cccc	K	0 – 99.99 segundos
Conteos por paso	dddd	K	0 – 9999
Eventos	eeee	X, Y, C, S, T, ST, GX, GY, SP	Vea el mapa de memorias
Palabra de salidas	Ffff	V	Vea el mapa de memorias
Máscara de salidas	Ggggg	V	Vea el mapa de memorias

Capítulo 6: Programación de las instrucciones DRUM

Las instrucciones de tambor usan cuatro contadores en la CPU. El programa ladder puede leer los valores de contadores para el estado del tambor. El programa ladder puede escribir un nuevo valor prefijado de paso número a CTA (n+2) en cualquier momento. Sin embargo, los otros contadores son sólo para propósitos de control.

Contador número	Rangos de (n)	Función	Función del bit de estado del contador
CTA(n)	0 – 174	Conteos en el paso	CT(n) = Tambor completado
CTA(n+1)	1 – 175	Valor de tiempo	CT(n+1) = (no usado)
CTA(n+2)	2 –176	Paso prefijado	CT(n+2) = (no usado)
CTA(n+3)	3 –177	Paso corriente	CT(n+1) =(no usado)

El programa ladder siguiente muestra la instrucción de MDRMD en un programa típico similar a como es mostrado por *DirectSOFT*. Son usados los pasos 1 hasta 11 y se usan los puntos de las 16 salidas. La palabra de máscara de las salidas está en V2000. Las salidas finales del tambor se muestran arriba de la palabra de máscara como la palabra V2020. Los bits de datos en V2000 son operados AND en forma lógica con el modelo de las salidas del paso corriente en el tambor, engendrando el contenido de V2020. Si usted quiere que todas salidas de tambor estén apagadas después de la energización, escriba ceros en V2000 en el primer barrido. La lógica ladder puede actualizar la máscara de la salida en cualquier momento para habilitar o incapacitar las salidas de tambor. El paso prefijado es el paso 1. El base de tiempo funciona en $(K50 \times 0,01) = 0,5$ segundos por conteo. Por lo tanto, la duración del paso 1 es $(5 \times 0,5) = 2,5$ segundos. Note que el paso 1 está basado en tiempo solamente (el evento se deja en blanco). En el último renglón, el bit de tambor completado (CT14) prende la salida Y0 al terminar el último paso (paso 10). La señal de tambor RESET también coloca OFF al contador CT14.



NOTA: El programa ladder debe cargar las constantes en V2000 hasta V2012 para definir todas las máscaras para los once pasos usados en este tambor.

PROGRAMACIÓN POR ETAPAS RLL^{PLUS}



En este capítulo...

Introducción a la programación de etapas	7-2
Venciendo el temor de programar por etapas	7-3
Diseñando diagramas de transición de estados	7-3
Usando la instrucción de salto de etapas para transiciones de estados	7-7
Ejemplo de programa de etapas: Controlador de lámpara con flip flop	7-8
Ejemplo de programa de etapas: Abridor de un portón de garaje	7-10
Consideraciones de diseño del programa de etapas	7-15
Conceptos de procesamiento paralelo	7-19
Instrucciones de RLLPLUS	7-21
Preguntas y respuestas acerca de la programación de etapas	7-27

Introducción a la programación de etapas

La programación por etapas (etapa programming) le permite tener una forma de organizar y programar aplicaciones complejas con relativa comodidad, cuando es comparado a programar soluciones puramente con lógica ladder (RLL). La programación por etapas no reemplaza ni anula el uso del tradicional programa ladderbooleano. También se le llama RLL^{plus}. Usted no tendrá que descartar ninguna experiencia ni entrenamiento que ya tenga. La programación por etapas le permite simplemente dividir y organizar un programa en grupos de instrucciones ladder llamado etapas. Esto le permite un desarrollo de un programa más rápido y más intuitivo del que proporciona un tradicional programa ladder.

Venciendo el temor de programar por etapas

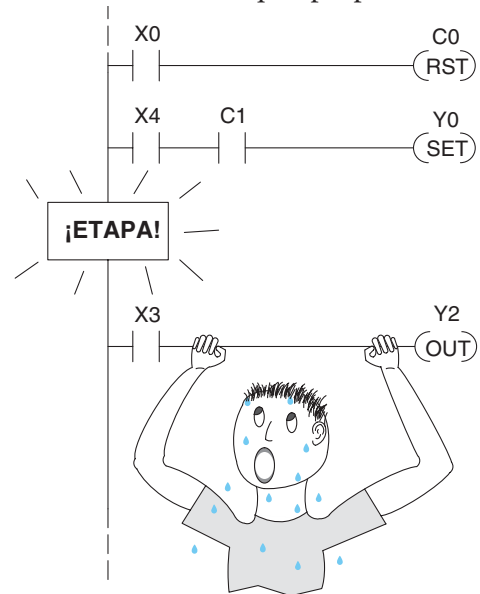
Muchos programadores de PLC en la industria se sienten confortables usando RLL para cada programa de PLC que ellos escriben... pero a menudo permanece escéptico o aún temeroso del aprendizaje de técnicas nuevas tales como la programación por etapas. Aunque RLL es excelente para resolver relaciones de lógica booleana, tienen también desventajas:

- Los programas grandes pueden llegar a ser casi inmanejables, a causa de una falta de estructura.
- En RLL los enclavamientos deben ser hechos a partir de relevadores auto enclavados
- Cuando un proceso se detiene es difícil de encontrar el renglón donde el error ocurrió.

Es fácil de ver que estas ineficacias consumen un tiempo adicional, y el tiempo es dinero. ¡La programación por etapas vence estos obstáculos! ¡Creemos que gastar un momento para estudiar el concepto de etapas es una de las mejores inversiones que un programador de PLC puede hacer para programar con velocidad y con eficiencia!

Por lo tanto lo alentamos a estudiar programación por etapas y agregarlo a su "bloque de herramientas" de técnicas de programar. Este capítulo está diseñado para aprender a programar con etapas. Para mejores resultados:

- Comience en el principio y no se salte sobre ninguna sección.
- Estudie cada concepto de programación de etapas trabajando con cada ejemplo. Los ejemplos se construyen progresivamente uno al otro.
- Lea las Preguntas y Respuestas al final del capítulo para una revisión rápida.

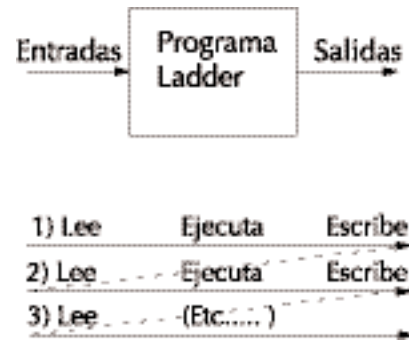


Diseñando diagramas de transición de estados

Introducción a estados de procesos

Los técnicos familiarizados con la ejecución de programas ladder saben que la CPU debe barrer el programa ladder repetidas veces. Sus tres pasos básicos son:

1. Lea las entradas
2. Ejecute el programa de escalera
3. Escriba las salidas



El beneficio es que un cambio en las entradas puede afectar las salidas en apenas unos pocos milisegundos.

La mayoría de los procesos de fabricación se componen de una serie de actividades o condiciones que duran varios segundos, minutos o aún horas. Podríamos llamar éstos "estados de proceso", que pueden estar activos o inactivos en algún tiempo determinado. Un desafío para programas de RLL es que cierto evento de entrada puede durar solamente un breve instante. Típicamente creamos relevadores autoenclavados en RLL para mantener el evento de entrada para mantener un estado de proceso por una duración requerida.

Podemos organizar y poder dividir la lógica en secciones llamadas "etapas" que representan estados del proceso. Pero antes de describir las etapas con detalles, le diremos el secreto para la comprensión de la programación por etapas: diagramas de transición de estado.

Necesidad de diagramas de estado

A veces necesitamos olvidarnos de la naturaleza de PLCs en el sentido del ciclo continuo, y enfocar nuestro pensamiento hacia estados del proceso que necesitamos identificar. El análisis claro, pensado y conciso de una aplicación nos da la mejor oportunidad para escribir programas eficientes sin errores. ¡Los diagramas del estado son apenas una herramienta de ayuda para dibujar un retrato de nuestro proceso! ¡Usted descubrirá que si podemos obtener del retrato correcto, nuestro programa estará correcto también!

Proceso de 2 estados

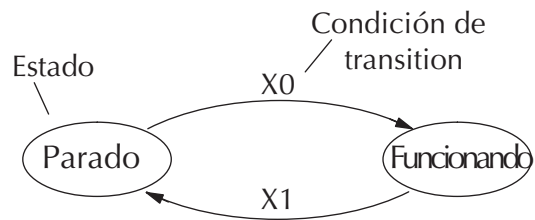
Consideremos el proceso sencillo mostrado a la derecha, que controla un motor industrial. Usaremos un botón momentáneo verde para prender el motor, y un rojo para apagarlo. El operario de la máquina apretará el botón apropiado por aproximadamente un segundo. Los dos estados de nuestro proceso es **Funcionando** y **Parado**.



El próximo paso deberá dibujar un diagrama de transición de estados, como mostrado a la derecha. Muestra los dos estados **Funcionando** y **Parado**, con dos líneas de transición intermedias. Cuando el evento de la entrada X0 es verdadero, pasa de **Parado** a **Funcionando**. Cuando la entrada X1 es verdadera, se pasa de **Funcionando** a **Parado**.

Si ha seguido la explicación, está muy cerca de agarrar el concepto y el poder de resolver el problema de creación de diagramas de transición de estado. La salida del controlador es X0, que es verdadera cada vez que estamos en el estado **Funcionando**.

En un sentido booleano, el estado $X0 = \text{Funcionando}$. Luego, aplicaremos el diagrama de estados primero como RLL y luego como un programa por etapas. Esto lo ayudará a ver la relación entre los dos métodos en la resolución de problemas



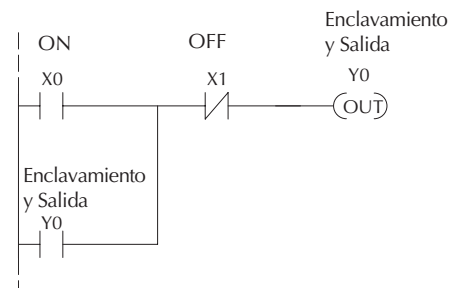
Ecuación de salida: $Y0 = \text{On}$

El diagrama de transición de estados es un retrato de la solución que necesitamos crear. La belleza de esto es: expresa el problema independientemente del idioma de programación que podríamos usar para ejecutarlo. *¡En otras palabras, dibujando el diagrama nosotros hemos resuelto ya el problema del control!*

Primero, traduciremos el diagrama del estado a RLL tradicional. Entonces mostraremos cuán fácil deberá traducir el diagrama en una solución de programación por etapas.

Equivalente RLL

La solución de RLL se muestra a la derecha. Se compone de un relevador de control y salida al mismo tiempo que se auto enclava, $Y0$. Cuando el botón ($X0$) es apretado, la bobina de salida $Y0$ prende y el contacto $Y0$ en la segunda fila se cierra con lo cual mantiene energizada la bobina. De modo que $X0$ puede abrir y $Y0$ permanece activado después que el contacto $X0$ abre. El contactor del motor también se cierra ya que está conectado a $Y0$, de modo que el motor ahora está **Funcionando**.

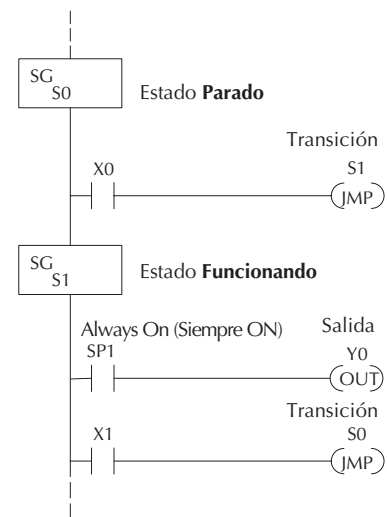


Cuando se activa el botón $X1$, se abre el contacto normalmente cerrado $X1$ que libera la bobina y la salida $Y0$ se apaga cuando la bobina $Y0$ se desactiva. El motor ahora está **Parado**.

Equivalente con etapas

La solución del programa con etapa se muestra a la derecha. Los dos bloques $S0$ y $S1$ de etapas corresponden a los dos estados **Funcionando** y **Parado**. Los renglones debajo de cada bloque de etapas pertenecen a cada etapa respectiva. ¡Esto significa que el PLC sólo tiene que barrer esos renglones cuando la etapa correspondiente es activa!

Por ahora, asumamos que comenzamos en el estado **Parado**, de modo que la etapa $S0$ está activa. Cuando se aprieta el botón $X0$, ocurre una transición de etapas. Se ejecuta la instrucción $\text{JMP } S1$, que apaga simplemente el bit de la etapa $S0$ y prende el bit de la etapa $S1$. ¡Así que en el próximo barrido del PLC, la CPU no ejecutará la etapa $S0$, sino que ejecutará la etapa $S1$!



En el estado **Funcionando** (etapa $S1$), queremos que el motor siempre esté **Funcionando**. El contacto $SP1$ especial del relevador se define como siempre activado, de modo que $Y0$ activa el contactor que prende el motor.

Cuando se aprieta el botón $X1$, ocurre una transición al estado **Parado**.

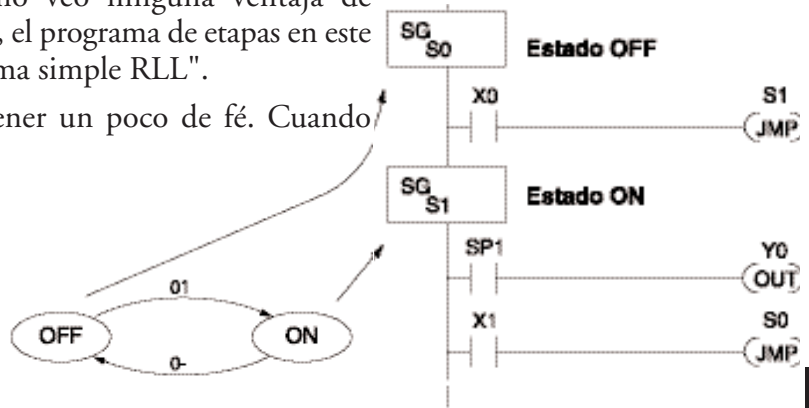
Se ejecuta la instrucción JMP S0, que apaga simplemente el bit de la etapa S1 y prende el bit de la etapa S0. En el próximo barrido del PLC, la CPU no ejecutará la etapa S1, de modo que la salida Y0 se apagará. El estado Parado (S0) estará listo para el próximo ciclo.

Hagamos comparaciones entre RLL y PLL^{plus}

Ud. puede estar pensando "no veo ninguna ventaja de programar en etapas... de hecho, el programa de etapas en este caso es más largo que el programa simple RLL".

Bien, ahora es el tiempo de tener un poco de fé. Cuando crecen en complejidad los problemas de control, la programación por etapas gana rápidamente en la sencillez, el tamaño del programa, etc.

Por ejemplo, considere el diagrama adyacente. Note cuán fácil es establecer una correlación de estados del diagrama de la transición de estados al programa de etapas a la derecha. ¡Ahora, desafiamos a cualquiera a identificar los mismos estados en el programa de RLL en la página previa!



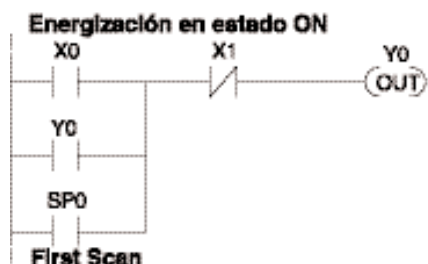
Etapas iniciales

Durante la energización del PLC y en la transición del modo Program para RUN, el PLC siempre comienza con todas etapas (SG) normales desactivadas. Las etapas del programa mostrado hasta ahora no tienen realmente una manera de iniciar el programa (porque no se examinan renglones a menos que la etapa esté activa).

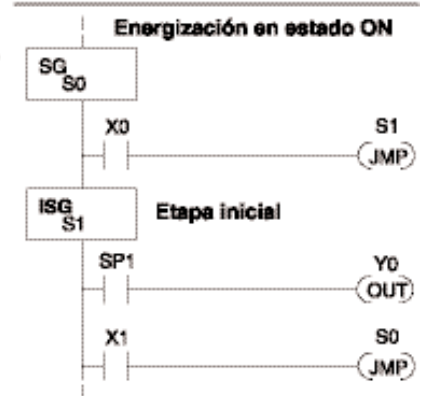
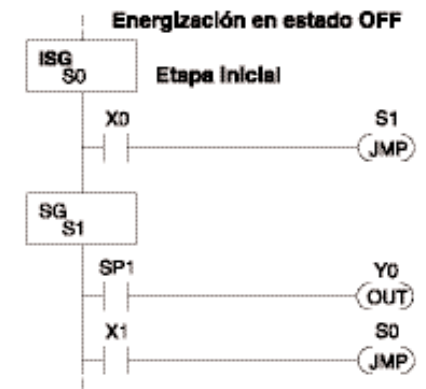
Asumamos que siempre comenzaremos en el Estado Parado, que es como funciona el programa en ladder. La etapa inicial ISG se define para ser activa en la energización.

En el programa modificado a la derecha, hemos cambiado la etapa S0 al tipo ISG. Esto asegura que el PLC examina el contacto X0 después de la energización, porque la etapa S0 es activa. ¡Después de la energización, una etapa inicial (ISG) trabaja como cualquier otra etapa!

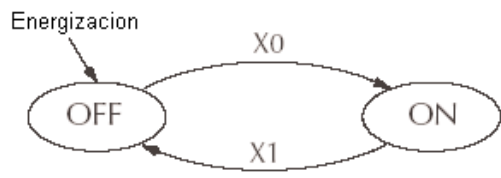
Podemos cambiar ambos programas para que el motor esté



ON o funcionando en la energización. En el programa RLL a la izquierda, debemos agregar el relevador SP0 que se cierra en el primer barrido de la CPU, que enclava Y0. En el ejemplo de etapas a la derecha, hacemos simplemente la etapa S1 una etapa inicial (ISG) en vez de S0.



Podemos marcar el estado deseado de la energización como mostrado en la figura adyacente, que nos ayuda a recordar de usar la etapa inicial ISG apropiada cuándo se crea un programa de etapas. Es permisible tener tantas etapas iniciales ISG como requiera el proceso.



Qué hacen los bits de etapas

Recuerde que una etapa es apenas una sección del programa que es activa o inactiva en un momento dado. Todos los bits de etapas (S0 a 1777) residen en la memoria imagen del PLC como bits de estado individuales. Cada bit de etapa es un booleano 0 o 1 en cualquier momento.

La ejecución del programa siempre lee los renglones de arriba hacia abajo, y de la izquierda a la derecha. El dibujo debajo muestra el efecto del estado del bit de etapa. Los renglones debajo de la instrucción de etapa continúan a ser ejecutados hasta que la próxima instrucción de etapa o el fin del programa pertenezca a la etapa 0. Su operación equivalente se muestra a la derecha. Cuándo la etapa S0 es verdadera, los dos renglones conducen corriente.

- Si el bit de la etapa S0 = 0, los renglones *no son barridos* (ejecutados).
- Si el bit de la etapa S0 = 1, los renglones *son barridos* (ejecutados).

Como se ve el programa por etapas

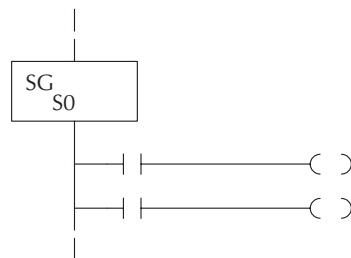
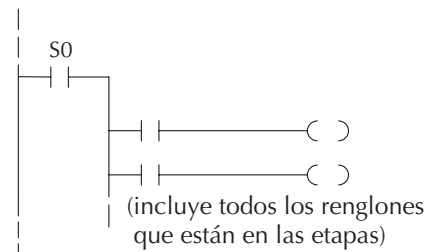


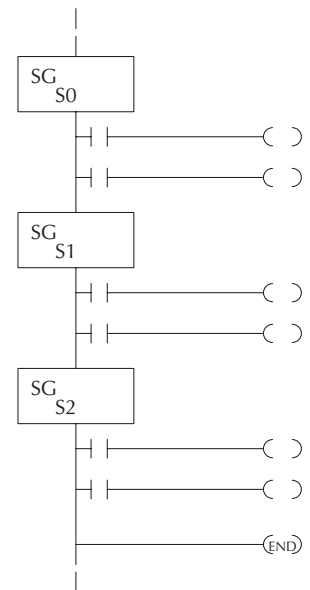
Diagrama ladder equivalente



Características de la instrucción de etapas

Los bloques de etapas en el riel de poder izquierdo dividen los renglones del programa en etapas. Algunas reglas para las etapas son:

- **Ejecución** – Sólo se ejecuta lógica en etapas activas en cualquier barrido.
- **Transiciones** – Las instrucciones de transición de etapas surten efecto en la próxima ocurrencia de las etapas implicadas.
- **Numeración octal** – Las etapas se numeran en octal, como puntos de entradas y salidas, etc. de modo que "S8" no es válido.
- **Cantidad de etapas posibles** – El DL06 ofrece hasta 1024 etapas (S0 a 1777 en octal).
- **No hay etapas duplicadas** – Cada número de etapa es único y puede ser usado solamente una vez.
- **Cualquier orden** – Usted puede saltarse los números y ordenar los números de etapas en cualquier orden.
- **Ultima etapa** – La última etapa en el programa incluye todos los renglones de su bloque de etapa hasta la bobina END.



Usando la instrucción de salto de etapas para transiciones de estados

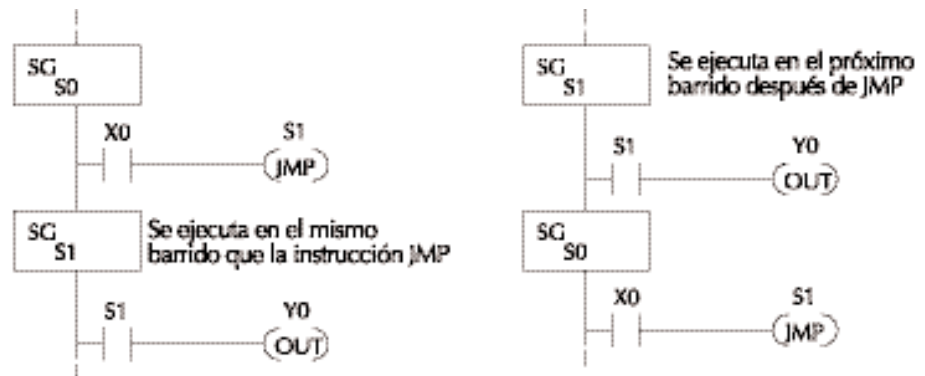
Las instrucciones de salto JMP, SET y RESET

La instrucción de salto de etapas JMP que hemos usado desactiva la etapa en que ocurre la instrucción, activando la etapa definida en la instrucción JMP. Vea la transición de estados mostrada abajo. Cuando el contacto X0 se energiza, ocurre una transición de estados de S0 a S1. Los dos ejemplos de etapa mostrados debajo son equivalentes. La instrucción de salto de etapas JMP es igual a un reset de etapa de la etapa actual, más una instrucción SET para la etapa a que queremos hacer la transición.



Lea con mucho cuidado, por favor - La instrucción de salto JMP es muy mal entendida. El "salto" no ocurre inmediatamente como un GOTO o una instrucción de control de programa GOSUB cuando ejecutada. Así es cómo trabaja:

- La instrucción de salto coloca en OFF el bit de la etapa en que ocurre. ¡Todos los renglones en la etapa terminan la ejecución durante el barrido corriente, *aunque haya otros renglones en la etapa debajo de la instrucción de salto!*
- El estado OFF será vigente en el siguiente barrido, de modo que la etapa que ejecutó la instrucción de salto previamente será inactiva y descartada.
- El bit de etapa llamado en la instrucción de salto JMP se colocará ON inmediatamente, de modo que la etapa se ejecutará en su próxima ocurrencia. En el programa mostrado abajo a la izquierda, la etapa S1 se ejecuta durante el mismo barrido que el JMP S1 ocurre en S0. En el ejemplo a la derecha, la etapa S1 ejecuta en el barrido siguiente después que el JMP S1 se ejecuta, porque la etapa S1 se localiza arriba de la etapa S0.

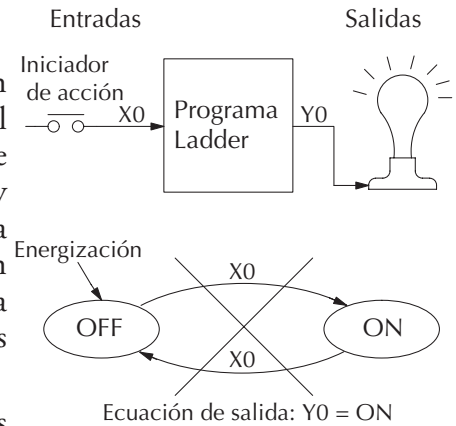


Nota: Asumimos que comenzamos con la etapa S0 activa y la etapa S1 inactiva en ambos ejemplos.

Ejemplo de programa de etapas: Controlador de lámpara con flip flop

Proceso de 4 estados

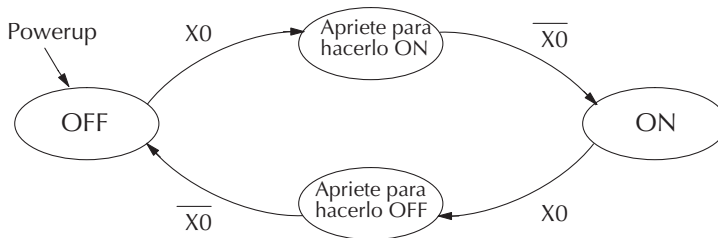
En el proceso mostrado a la derecha, usamos un botón momentáneo normal para controlar una lámpara. El programa enclavará la entrada del interruptor, de modo que empujaremos y liberaremos el botón para prender la luz y luego lo empujaremos y soltaremos para apagarlo (llamada a veces función flip flop). Claro, podríamos comprar un interruptor mecánico con la acción de mantener el estado ya en él. ¡Sin embargo, este ejemplo es educativo! Dibujemos ahora el diagrama de transición de estados.



Un primer enfoque típico sería usar X0 para ambas transiciones (como el ejemplo mostrado a la derecha). Sin embargo, *esto no es correcto* (siga leyendo por favor).

Note que este ejemplo difiere del ejemplo del motor porque ahora tenemos solamente un botón. Cuando nosotros apretamos el botón se tienen dos condiciones de transición. Haremos una transición alrededor del diagrama de estado a una alta velocidad. ¡Si se usan etapas, esta solución prendería la luz cada barrido (obviamente indeseable)!

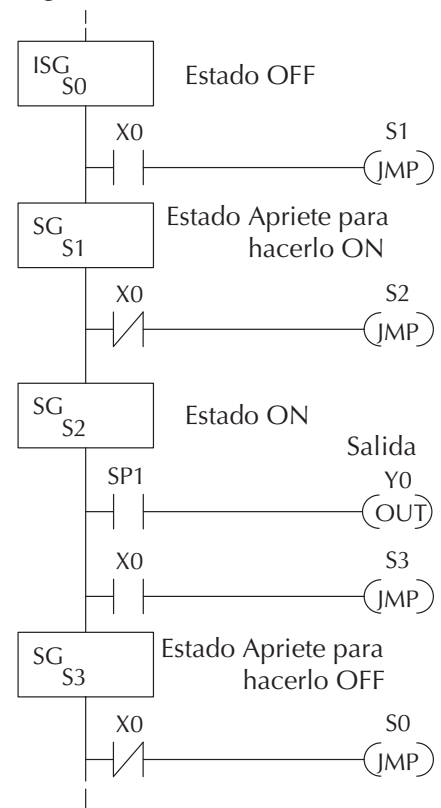
La solución deberá hacer que el apretar y soltar el botón sean eventos separados. Vea al nuevo diagrama de transición de estados de la figura de abajo. En la energización entramos el estado OFF.



Cuándo se aprieta el botón X0, entramos al estado de "apriete para hacerlo ON". Cuándo se suelta, entramos al estado ON. Note que X0 con la barra encima denota X0 negado.

Cuándo se está en el estado ON, otro ciclo de apretar y soltar nos lleva de forma similar al estado OFF. Ahora tenemos dos estados únicos (OFF y ON) usados cuando el botón se suelta, que es lo que se necesitaba para resolver el problema de control.

El programa equivalente de etapas se muestra a la derecha. El estado deseado de la energización es OFF, de modo que haremos S0 una etapa inicial ISG. En el estado ON agregamos el contacto del relevador especial SP1 que es siempre ON. ¡Note que aunque los programas van siendo más complejos, es todavía fácil poner en correlación el diagrama de transición de estado con el programa de etapas!



Cuatro pasos para escribir un programa por etapas

Por ahora, usted ha notado probablemente que seguimos los mismos pasos para resolver cada problema ejemplo. Los pasos le vendrán probablemente automáticamente a su memoria si usted trabaja en todos los ejemplos en este capítulo. Es útil tener una lista de verificación para indicarnos la resolución de problemas. Los pasos siguientes resumen el procedimiento del diseño del programa de etapas:

1. Escriba una descripción de la operación de la aplicación

Describa todas funciones del proceso en sus propias palabras. Comience listando lo que sucede primero, luego lo que viene en segundo lugar, en tercer lugar, etc. Si usted encuentra que hay demasiadas cosas que suceden inmediatamente, trate de dividir el problema en más de un proceso. Recuerde, usted puede tener todavía los procesos comunicándose uno con otro para coordinar su actividad completa.

2. Dibuje un diagrama de bloques

Las entradas representan toda la información que el proceso necesita para hacer decisiones y las salidas se conectan a todos los aparatos controlados por el proceso.

- Haga una lista de entradas y salidas del proceso.
- Asigne números de entradas y salidas (X y Y) a entradas y salidas físicas.

3. Dibuje el diagrama de transición de estados

. El diagrama de transición de estados describe la función central del diagrama de bloques, leyendo entradas y generando salidas.

- Identifique y denomine los estados del proceso.
- Identifique el o los eventos requeridos para cada transición entre estados.
- Asegúrese que el proceso tenga una manera de volver a encender o es cíclico.
- escoja el estado de energización para su proceso.
- Escriba las ecuaciones de salidas.

4. Escriba el programa por etapas

Traduzca el diagrama de transición de estados en un programa de etapas.

- Haga cada estado una etapa. Recuerde de numerar las etapas en octal. Hay hasta 1024 etapas totales disponibles en el DL06, numeradas 0 a 1777 en octal
- Ponga la transición lógica dentro de la etapa que origina cada transición (la etapa que es abandonada por cada flecha).
- Use una etapa inicial ISG para cualquiera estado que deba ser activo en la energización.
- Coloque las salidas o las acciones en las etapas apropiadas.

Usted notará que los pasos 1 a 3 solamente nos preparan para escribir el programa de etapas en el paso 4. Sin embargo, el programa se escribe virtualmente a causa de la preparación anterior. ¡Luego Ud. será capaz de comenzar con una descripción escrita de una aplicación y crear un programa de etapas en una sesión fácil!

Ejemplo de programa de etapas: Abridor de un portón de garaje

Ejemplo del control para abrir un portón de garaje.

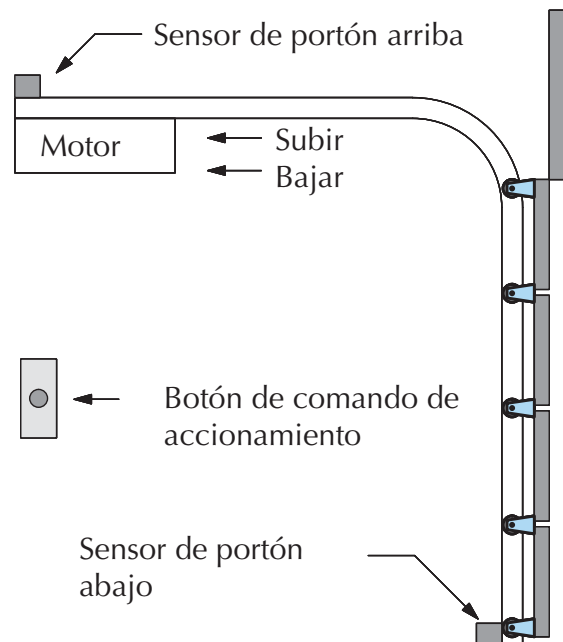
En este programa ejemplo crearemos un controlador de apertura de portón de garaje. ¡La mayoría de los lectores están familiarizados con esta aplicación!

El primer paso que debemos tomar deberá ser describir cómo trabaja un abridor de portón. Comenzaremos describiendo la operación básica, y luego agregaremos más características. Los programas de etapas son muy fáciles de modificar.

El controlador del portón del garaje tiene un motor que levanta o baja el portón al apretar un botón. El dueño del garaje aprieta un botón momentáneo para levantar una vez el portón. Después que el portón está arriba, otro apriete del botón bajará el portón.

Para identificar las entradas y las salidas del sistema, es a veces útil hacer un croquis sus componentes principales, como mostrado en la figura a la derecha del lado del portón. El portón tiene un interruptor de límite arriba y uno abajo. Cada interruptor de límite se cierra sólo cuando el portón llega al fin del movimiento en la dirección correspondiente. Durante el movimiento no se acciona ningún interruptor límite.

El motor tiene dos entradas de comando: subir y bajar. Cuando ninguna entrada es activada, el motor se para. El comando del portón es un botón sencillo, ya sea si es montado en la pared como mostrado, o es un control remoto de radio, o ambos, actuando como un control OR como dos contactos abiertos en paralelo.



Dibuje el diagrama de bloques

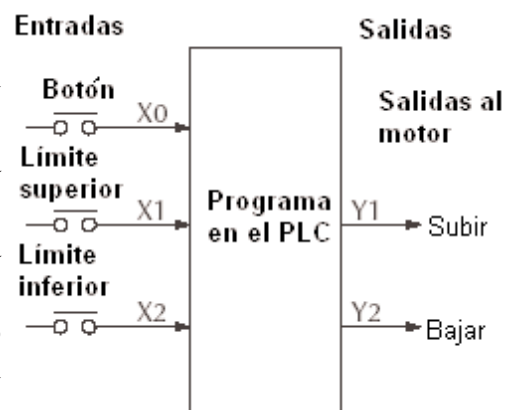
Se muestra a la derecha el diagrama de bloques del controlador.

La entrada X0 corresponde al control del portón (el botón).

La entrada X1 se energiza cuando el portón alcanza la posición superior.

La entrada X2 se energiza cuando el portón llega a la posición inferior.

Cuando el portón esté posicionado entre arriba o abajo, ambos interruptores límite están abiertos. El controlador tiene dos salidas para manejar el motor. Y1 es el comando para levantar el portón (Subir), y Y2 el comando para bajar el portón (Bajar).



Dibuje el diagrama de estados

Ahora estamos listos para dibujar el diagrama de transición de estados. Como el ejemplo previo del controlador de una lámpara, esta aplicación tiene también solamente un interruptor para entrar el comando. Vea la figura de abajo.

- Cuando el portón está completamente abajo (Estado ABAJO), nada sucede hasta que X0 se energice. Al apretar y soltar el botón el portón pasa al estado de SUBIR, donde la salida Y1 se prende y causa que el motor levante el portón.
- Cuando el portón está completamente abajo (Estado ABAJO), nada sucede hasta que X0 se energice. Al apretar y soltar el botón el portón pasa al estado de SUBIR, donde la salida Y1 se prende y causa que el motor levante el portón.
- Nada sucede hasta que ocurra otro ciclo de apretar y soltar el botón X0. Eso hace que el portón pase al estado de BAJAR y se prende la salida Y2 para causa que el motor baje el portón. Se llega al estado ABAJO cuando el interruptor límite X2 se activa.



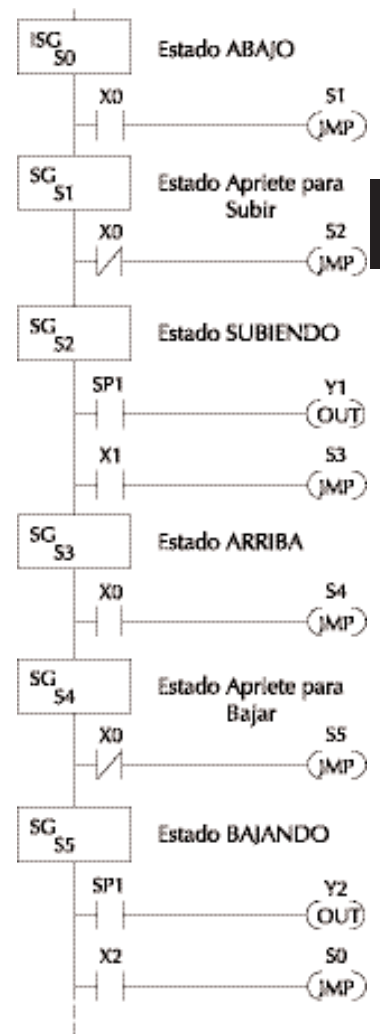
Ecuaciones de salidas: Y1 = SUBIENDO Y2 = BAJANDO

El programa equivalente de etapas se muestra a la derecha. Por ahora, asumiremos que el portón está abajo en la energización, así que el estado deseado en la energización es ABAJO. Hacemos S0 una etapa inicial ISG. La etapa S0 permanece activa hasta que se active el botón de control del portón. Luego hacemos la transición (JMP) a la etapa de transición "Apretar para subir", estado S1.

La acción de apretar y soltar el botón (etapa S1) nos conduce a la etapa de SUBIENDO, S2. Usamos el contacto SP1 siempre ON para energizar el motor con el comando de subir, Y1. Cuando el portón alcanza la posición completamente levantada, el interruptor límite X1 se activa. Esto nos lleva a la etapa ARRIBA, S3, donde esperamos hasta que ocurra otro comando de control de portón.

Estando en la etapa S3, ARRIBA, apretando y soltando del botón nos llevará a la etapa S5, ABAJO, donde activamos Y2 para ordenar al motor bajar el portón. Esto continúa hasta que el portón alcance el interruptor límite inferior, X2 .

Cuándo X2 se cierra, saltamos de la etapa S5 ABAJO a la etapa S0, donde comenzamos.



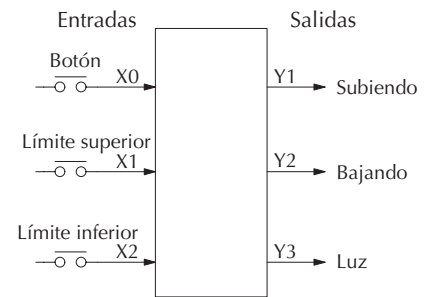
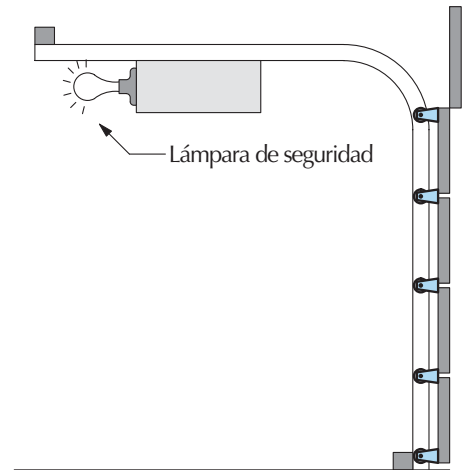
NOTA: La única cosa especial acerca de la etapa inicial ISG es que es automáticamente activa en la energización. Después, es igual que cualquier otra etapa.

Agregue una lámpara para iluminar el garage por un corto tiempo

A continuación agregaremos una lámpara de seguridad al sistema del abridor del portón. Es mejor asegurarse que la principal función esté funcionando como hemos hecho y luego se agrega esta característica secundaria.

La luz de seguridad es normal en los abridores de portón de garage comercialmente disponibles. Se muestra a la derecha junto con el motor. La lámpara se enciende con cualquier movimiento del portón y queda encendida por aproximadamente 3 minutos.

Esta parte del ejercicio le mostrará el uso de estados paralelos en nuestro diagrama del estado. En vez de usar la instrucción JMP, usaremos las instrucciones SET y RESET



7

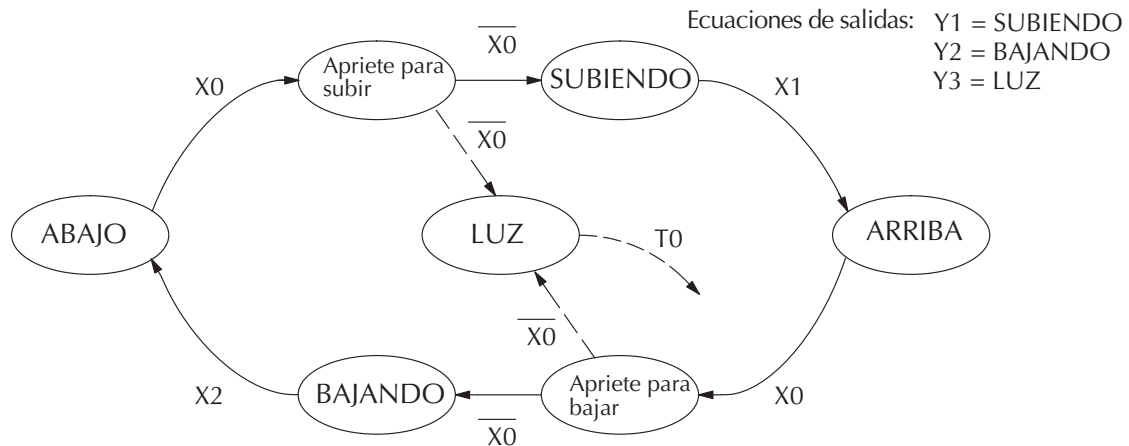
Modifique los diagramas de bloque y de estado

Para controlar la lámpara, le agregamos una salida a nuestro diagrama de bloques del controlador, mostrado a la derecha, Y_3 es la salida de control de la lámpara. En el diagrama de abajo, agregamos un estado adicional llamado "LUZ".

Cuando el dueño del garage aprieta el interruptor de control del portón, el estado de SUBIR o BAJAR es activo y el estado LUZ se hace activo al mismo tiempo. La línea al estado LUZ es segmentada, porque no es el sendero primario.

Podemos pensar en el estado LUZ como un proceso paralelo al estado de SUBIR y BAJAR. El paso al estado LUZ no es una transición (una etapa JMP), sino una instrucción SET.

En la lógica de la etapa LUZ, colocaremos un temporizador de 3 minutos. Cuando el temporizador expira su tiempo, el bit de estado T_0 del temporizador se activa y desactiva la etapa LUZ. ¡El paso para fuera de la etapa LUZ no va a ningún lugar, indica que la etapa LUZ se desactiva, y la lámpara se apaga!



Usando un temporizador dentro de una etapa

El programa modificado terminado se muestra a la derecha. Las áreas con sombra indican las partes agregadas del programa.

En la etapa S1 agregamos la instrucción SET etapa S6. Cuando el contacto X0 abre, pasamos de S1 a dos estados activos nuevos: S2 y S6. En el estado S4, agregamos la misma instrucción. De modo que cada vez que alguien aprieta el botón de control de portón, la lámpara se enciende.

La mayoría de los programadores nuevos de etapa se preocupan donde colocar la etapa LUZ y cómo numerarla. ¡La buena noticia es que no importa!

- Escoja un número nuevo de etapa, y úselo para la etapa nueva y como referencia desde otras etapas.
- La colocación en el programa no es crítica, así que lo colocamos al fin del programa.

Usted tal vez piense que cada etapa tiene que estar directamente debajo de la etapa anterior. Aunque es una buena práctica, no es necesario (eso es bueno, porque hay 2 localizaciones para la instrucción SET S6 lo que hace esto imposible). Los números de las etapas y cómo ellos son usados determinan como se hace la transición.

En la etapa S6, encendemos la lámpara energizando Y3. El contacto del relevador especial SP1 está siempre ON o activado.

El temporizador T0 cuenta el tiempo en un unidades de 0,1 segundo. Para lograr períodos de 3 minutos, calculamos:

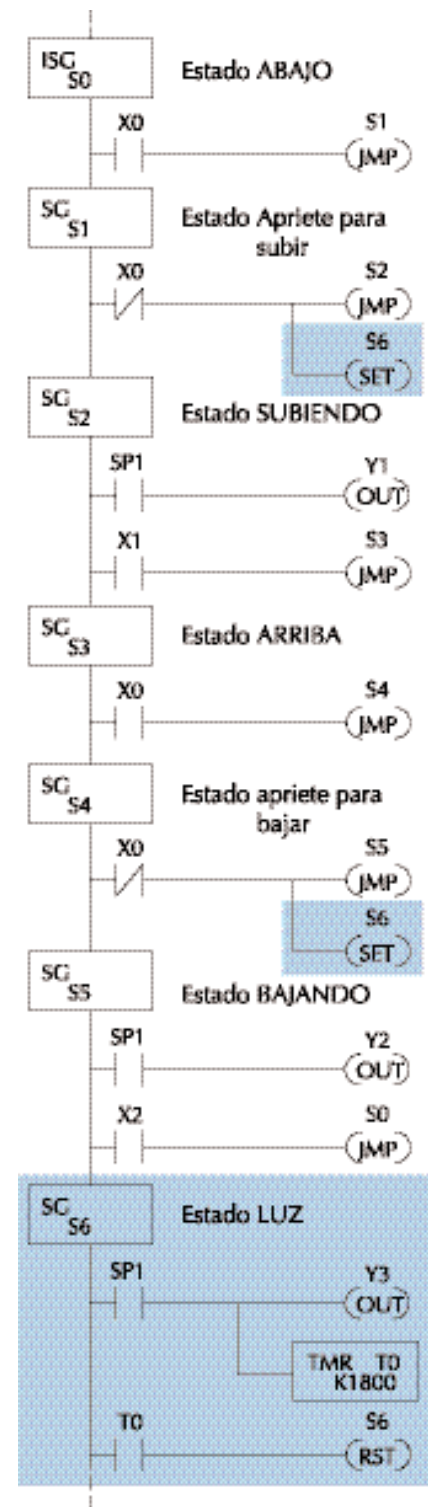
$$K = 3 \text{ min.} \times 60 \text{ s/min}$$

$$\Rightarrow K = 1800 \text{ conteos}$$

$$0,1 \text{ s/conteo}$$

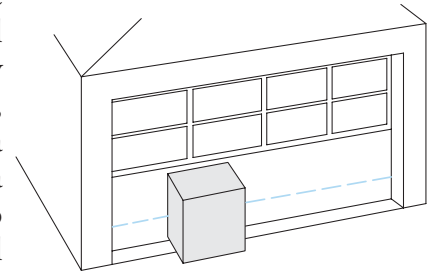
El temporizador está activado cuando la etapa S6 está activa. El bit T0 correspondiente del temporizador se activa cuando el temporizador cumple su tiempo. Así que en tres minutos transcurridos, T0=1 y la instrucción RESET S6 hace que la etapa sea inactiva.

Mientras la etapa S6 es activa y la lámpara está encendida, las transiciones de etapas en el transcurso primario continúan normalmente e independientemente de la etapa 6. Eso es, el portón puede subir o ir hacia abajo pero la lámpara estará encendida por exactamente 3 minutos.

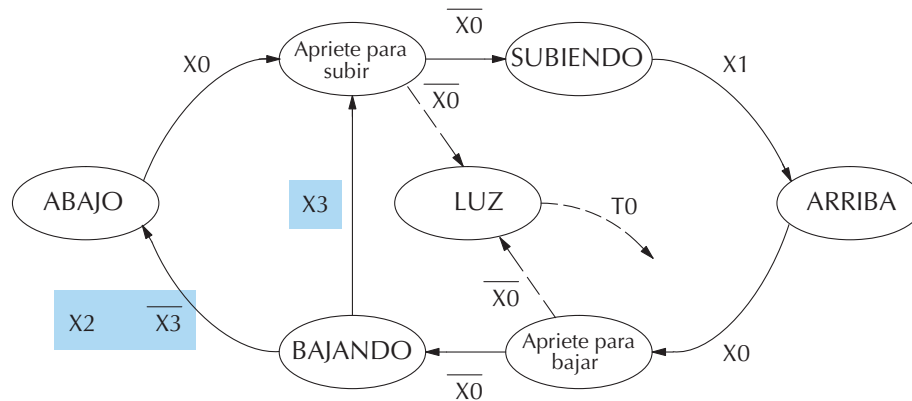
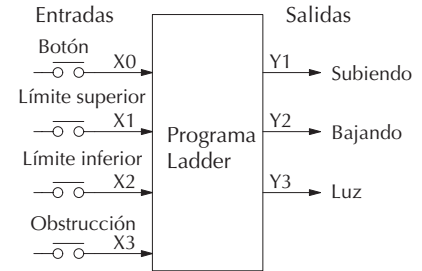


Agregue una parada de emergencia

Algunos abridores de portón de garaje detectan actualmente un objeto bajo el portón. Este evento para el descenso del portón. Generalmente es implementado con una fotocelda, y al pasar un objeto en frente al sensor y el portón bajando, este parará y comenzará a subir. Definiremos la característica de seguridad para trabajar de esta manera, agregando la entrada de la fotocelda al diagrama del bloque como mostrado a la derecha. X3 estará ON si un objeto está en el sendero del portón



Luego agregamos estos estados al diagrama de la transición de estados, mostrado en áreas sombreadas en la figura de abajo. Note el sendero nuevo de la transición a la cabeza del estado BAJAR. Si bajamos el portón y detectamos una obstrucción (X3), entonces saltamos al Estado de Apriete para subir. Hacemos esto en vez de saltar directamente al estado de SUBIR para dar a la salida BAJAR Y2 un barrido para apagarse, antes de la salida Y1 se energice.



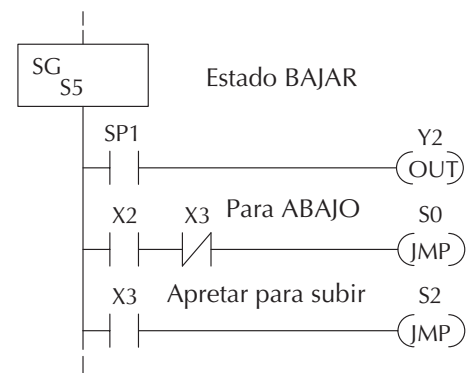
Transiciones exclusivas

Es teóricamente posible que el límite inferior (X2) y la obstrucción (X3) se podrían energizar en el mismo momento. En ese caso, "saltaríamos" simultáneamente a los estados **Apriete para subir** y **Apriete para bajar**, que no tiene sentido.

En vez de eso, le daremos prioridad a la obstrucción cambiando la condición de transición del estado ABAJO [X2 y (AND) no X3].

Esto asegura que el evento obstrucción tenga prioridad. Las modificaciones a la lógica que debemos hacer a la etapa MAS BAJA (S5) se muestra a la derecha. El primer renglón permanece igual. El segundo y tercer renglones implementan las transiciones que se necesitan.

Note el uso del contacto cerrado del relevador X3, que asegura que la etapa ejecute sólo una de las instrucciones JMP.



Consideraciones de diseño del programa de etapas

Organización del programa

Los ejemplos en este capítulo usaron hasta ahora un diagrama auto contenido del estado para representar el proceso principal. Sin embargo, se pueden tener múltiples procesos aplicados en etapas, todo en el mismo programa ladder. Los programadores nuevos de etapas tratan a veces de prender y apagar una etapa cada barrido, basado en la suposición falsa que sólo una etapa puede estar activa a la vez. Para renglones ladder que usted quiere ejecutar en cada barrido, colóquelos en una etapa que esté siempre activada.

La figura siguiente muestra una aplicación típica. Durante la operación de la actividad primaria de fabricación, los estados **Proceso principal**, **Inicialización en la energización**, **Parada de Emergencia y Supervisión de alarmas** e **interfase de operador** funcionan todos al mismo tiempo. En la energización, los cuatro estados mostrados comienzan la operación.

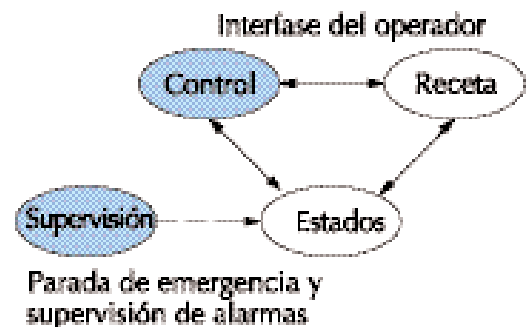


7

En una aplicación típica, las sucesiones separadas de etapas encima operan como sigue:

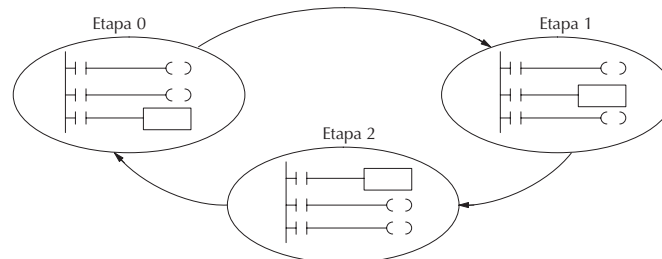
- **Inicialización en la energización** – Esta etapa contiene tareas hechas solamente una vez en la energización. El último renglón hace que se desactive la etapa, de modo que esta etapa es sólo activa por un barrido (o sólo tantos barridos como sean requeridos).
- **Proceso principal** Esta sucesión de etapas controla el corazón del proceso o la máquina. Un paso por la sucesión representa un ciclo de parte de la máquina, o de una serie en el proceso.
- **Parada de emergencia y supervisión de alarmas** –Esta etapa está siempre activa porque cuida de errores que podrían indicar una condición de alarma o requerir una parada de emergencia. Es común para esta etapa para reponer las etapas en el proceso principal o en otra parte, para inicializarlos después de una condición de error. .
- **Interfase de operador** –Esta es otra tarea que siempre debe estar activa y lista para responder a un operador. Permite que un operador se comunique para cambiar los modos, etc. independientemente del paso principal actual del proceso.

Aunque tengamos procesos separados, puede haber coordinación entre ellos. Por ejemplo, en una condición de error, la etapa de estados puede querer cambiar automáticamente la interfase de operador al modo de estados para mostrar la información de error como mostrado a la derecha. La etapa de supervisión podría configurar el bit para los estados y reponer (volver al valor original) el control de etapas y receta.



Cómo trabajan las instrucciones dentro de una etapa

Podemos pensar en estados o etapas simplemente dividiendo el programa ladder como está representado en la figura de abajo. Cada etapa contiene sólo los renglones necesarios para el estado correspondiente del proceso. La lógica para hacer una transición fuera de una etapa es hecha dentro de esa etapa. Es fácil escoger cuáles renglones estarán activos durante la energización del PLC usando un tipo "inicial" de etapa ISG.



La mayoría de las instrucciones trabajan tal como trabajan en ladder. Usted puede pensar en una etapa así como un programa miniatura ladder que está activo o inactivo.

Las bobinas de salidas – Como esperado, las bobinas en las etapas activas prenderán las salidas si la bobina tiene un flujo de corriente. Sin embargo, note lo siguiente:

- Las salidas trabajan como de costumbre, si es que cada salida (tal como "Y3") es usada en sólo una etapa. Las bobinas de salida se apagan automáticamente cuando la etapa es desactivada. Sin embargo, las instrucciones SET y RESET no dejan que la bobina cambie de estado para OFF.
- Una salida se puede referenciar en más de una etapa, pero sólo una de las etapas será activa cada vez.
- Si una bobina de salida es controlada por más de una etapa simultáneamente, la etapa activa más cercana al final del programa determina la posición final de la salida durante cada barrido. Por lo tanto, use la instrucción OROUT en vez de OUT cuando usted quiera que múltiples etapas tengan el control de una salida lógica OR.

Bobinas PD o One Shot – tenga cuidado al usar una bobina PD (Diferencial) en una etapa. Recuerde que la entrada a la bobina debe hacer una transición de OFF para ON. Si la bobina está energizada ya en el primer barrido cuando la etapa se hace activa, la bobina PD no trabajará. Esto es porque no ocurrió la transición de OFF para ON.

Alternativa de bobina PD: Si hay una tarea que usted quiere de hacer sólo una vez (en 1 barrido), se puede colocar en una etapa que haga la transición a la próxima etapa en el mismo barrido.

Contador – Al usar un contador dentro de una etapa, la etapa debe estar activa por un barrido antes de que la entrada al contador haga una transición de OFF para ON. De otro modo, no hay una verdadera transición y el contador no contará.

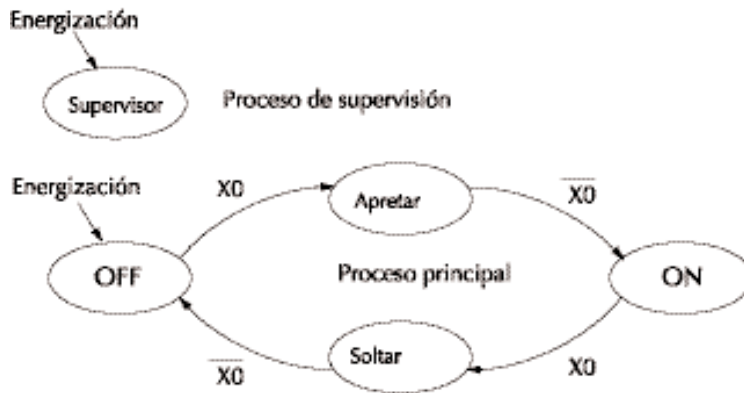
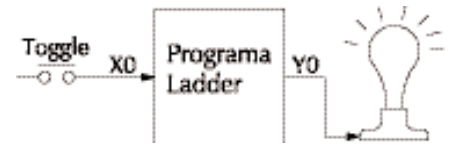
La instrucción contador normal tiene una restricción dentro de un programa de etapas: no puede ser repuesta desde otras etapas que usan la instrucción RST para el bit de contador. Sin embargo, el contador especial de etapa provee una solución (vea próximo párrafo).

Contador de etapas – El Contador de etapa tiene el beneficio que el conteo puede estar reponerse globalmente de otras etapas usando la instrucción RST. Tiene una entrada de conteo, pero no una entrada reset. Esto es la única diferencia de un contador uniforme..

Tambor – Por favor considere que el secuenciador de tambor es su propio proceso y es un método diferente de programación que la programación de etapas. Si necesita usar un tambor con etapas, esté seguro de colocar la instrucción de tambor en una etapa ISG que esté siempre activa.

Usando una etapa como un proceso de supervisión

Usted puede recordar la lámpara en el ejemplo del controlador en este capítulo. Para propósitos de ilustración, suponga que queremos controlar la "productividad" del proceso de la lámpara, contando el número de ciclos que ocurren. Esta aplicación requerirá la adición de un contador sencillo, pero la decisión clave está en donde poner el contador.

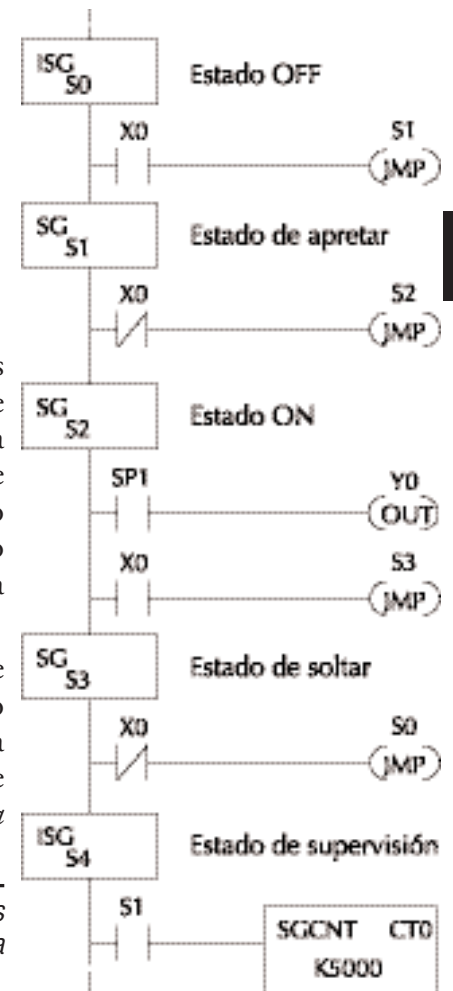


Los estudiantes nuevos de programación de etapas típicamente tratarán de colocar el contador dentro de una de las etapas del proceso que se trata de controlar. El problema con este enfoque es que la etapa esta activa solamente parte del tiempo. Para que el contador cuente, la entrada de conteo hacer la transición de OFF para ON por lo menos un barrido después que su etapa se activa. Asegurar esto requiere una lógica extra que puede ser complicada.

En este caso, sólo necesitamos agregar otra etapa de supervisión como mostrado arriba, para "mirar" el proceso principal. El contador dentro de la etapa de supervisión usa el bit S1 de la etapa del proceso principal como su entrada de conteo. ¡Los bits de etapas usados como un contacto nos deja controlar un proceso!



Note que la etapa de supervisión y la etapa de estado OFF son las etapas iniciales. La etapa de supervisión permanece activa indefinidamente.



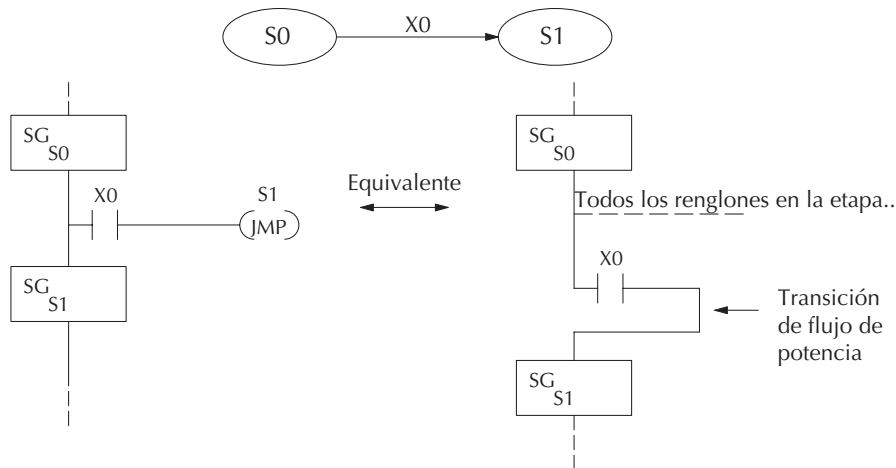
Contador de etapas

El contador en el ejemplo de encima es un contador de etapa especial. Note que no tiene una entrada RESET. El conteo es repuesto (reset) ejecutando la instrucción RESET, llamando al bit de estado del contador (CT0 en este caso). El contador de etapa tiene el beneficio que su conteo puede reponerse globalmente desde otras etapas. La instrucción contador normal no tiene esta capacidad de reset global. Puede usar también una instrucción contador normal dentro de una etapa... sin embargo, la entrada RESET del contador es la única manera de reponerlo.

La técnica de transición del flujo de potencia

Nuestra discusión de transiciones de estado ha mostrado cómo la instrucción de etapa JMP hace la etapa corriente inactiva y la próxima etapa (denominada en el JMP) activa. Como una manera alternativa de entrar esto en DirectSOFT 32, usted puede usar el método del flujo de potencia para transiciones de etapa.

El requisito principal es que la etapa actual esté localizada directamente encima de la próxima (el salto a) etapa en el programa ladder. Este arreglo se muestra en el diagrama abajo, para las etapas S0 y S1, respectivamente.

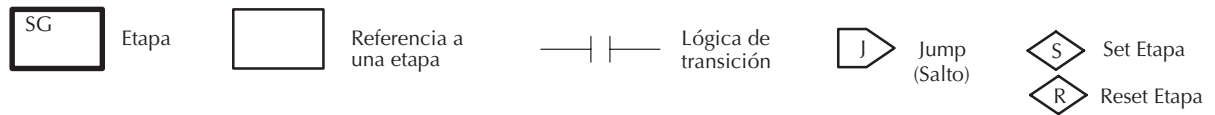


Recuerde que la instrucción JMP puede ocurrir dondequiera en la etapa actual y el resultado es el mismo. Sin embargo, las transiciones de flujo de potencia (mostrado arriba) deben ocurrir en el último renglón en una etapa. Todos los otros renglones en la etapa deben ser anteriores. El método de la transición de flujo de potencia es también factible en el programador portátil, simplemente siguiendo la condición de la transición con la instrucción de etapa para la próxima etapa.

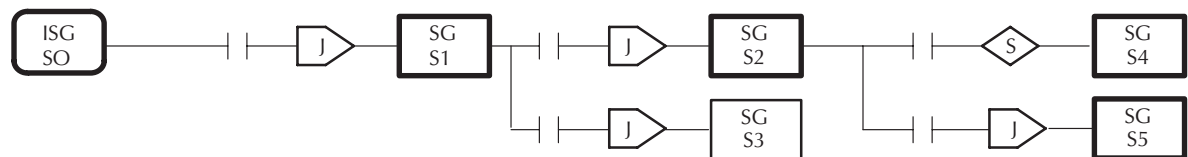
El método de la transición de flujo de potencia elimina una instrucción JMP, su única ventaja. Sin embargo, no es tan fácil de hacer los cambios del programa como usando la etapa JMP. Por lo tanto, recomendamos usar las transiciones JMP para la mayoría de los programadores.

La pantalla Stage View en DirectSOFT

La opción de tener una ventana con el diagrama de etapas (Etapa View) en DirectSOFT le permitirá ver el programa como un diagrama de flujo. La figura de abajo le muestra las convenciones de símbolos usada en los diagramas. Usted puede encontrar que el diagrama de etapas es útil como una herramienta para verificar que su programa de etapas ha reproducido fielmente la lógica del diagrama de transición de estado que usted



piensa ejecutar. El diagrama siguiente es un diagrama típico de etapas de un programa ladder que contiene etapas. Note la dirección de izquierda a derecha del diagrama de flujo.



Conceptos de procesamiento paralelo

Procesos paralelos

Previamente en este capítulo discutimos cómo un estado puede hacer una transición de un estado a otro, llamado una transición exclusiva. En otros casos, podemos necesitar bifurcar simultáneamente a dos o más procesos paralelos, como mostrado abajo. Es aceptable usar todas las instrucciones JMP como mostrado o podríamos usar una instrucción JMP y una instrucción SET de etapa (por lo menos una debe ser un JMP, para dejar S1). Recuerde que todas instrucciones en una etapa se ejecutan, aún cuando se hagan transiciones (el JMP no es un GOTO).



Note que si queremos que las etapas S2 y S4 se activen exactamente en el mismo barrido, ambos debe estar localizados abajo o encima de la etapa S1 en el programa ladder. ¡En general, bifurcar estados paralelos es fácil!

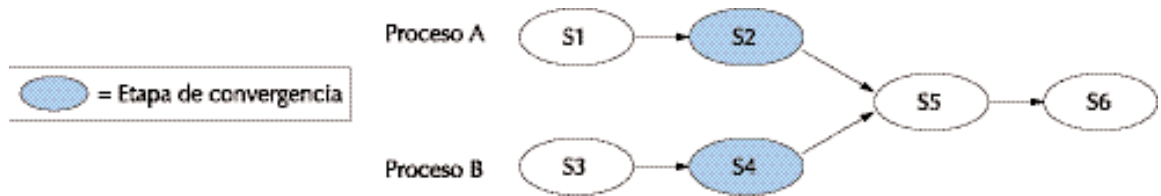


Procesos de convergencia

Ahora consideraremos el caso opuesto de bifurcar en paralelo, que es el proceso de convergencia. Esto significa simplemente que paramos de hacer múltiples etapas y seguimos haciendo uno a la vez. En la figura de abajo, el proceso A y el B convergen cuando etapas S2 y S4 hacen una transición a S5 en algún momento. En este caso, S2 y S4 son las etapas de Convergencia.

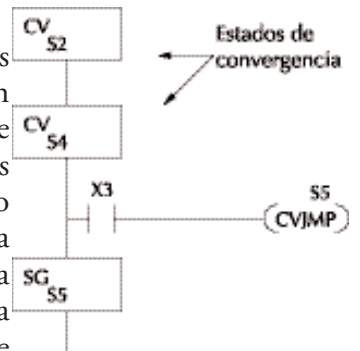
Etapas de convergencia (CV)

Aunque el principio de convergencia es suficientemente sencillo, trae una nueva complicación. Cuando el procesamiento paralelo se completa, los múltiples procesos casi nunca terminan al



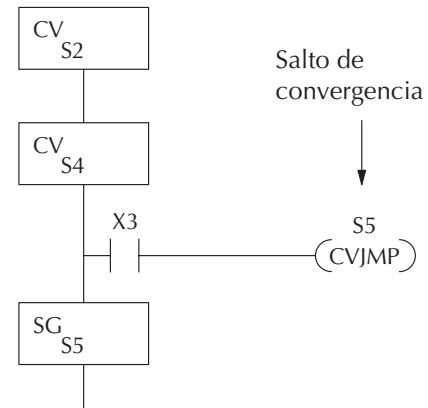
mismo tiempo. ¿En otras palabras, cómo podemos saber si una de las etapas S2 o S4 terminará por último? Esto es un punto importante, porque hay que decidir cómo se hace la transición para la etapa S5.

La solución es coordinar la condición de la transición fuera de las etapas de convergencia Convergencia de etapas. Logramos esto con un tipo de etapa diseñado para este propósito: la etapa de Convergencia (tipo CV). En el ejemplo a la derecha, se requieren las etapas de convergencia S2 y S4 para agruparse juntas como mostrado. ¡NO se permite ninguna lógica entre etapas CV! La condición de transición (X3 en este caso) debe estar localizada dentro de la última etapa de convergencia. La condición de la transición sólo tiene flujo de corriente cuando todas las etapas de convergencia en el grupo están activas.



Salto de convergencia (CVJMP)

Recuerde que la última etapa de convergencia sólo tiene flujo de corriente cuando todas las etapas CV en el grupo están activas. Para complementar la etapa de convergencia, necesitamos una nueva instrucción de salto. La instrucción salto de convergencia (CVJMP) mostrado a la derecha hace la transición a la etapa S5 cuando la entrada X3 está activa (como se puede esperar), pero también desactiva automáticamente todas las etapas de convergencia en el grupo. Esto hace al CVJMP una instrucción muy poderosa. Note que esta instrucción puede sólo ser usada con etapas de convergencia.



Reglas de uso de la etapa de convergencia

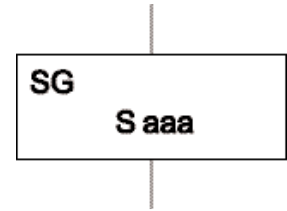
Lo siguiente hace un resumen de los requisitos en el uso de etapas de convergencia, inclusive algunas sugerencias para su aplicación efectiva:

- Una etapa de convergencia debe ser usada como la última etapa de un proceso que corre en paralelo a otro proceso o procesos. Una transición a la etapa de la convergencia significa que cierto proceso se ha acabado y representa un punto de espera hasta todos los otros procesos paralelos también terminen.
- El número máximo de etapas de convergencia que componen un grupo es 16. En otras palabras, hasta máximo de 16 etapas puede convergir en una etapa.
- Las etapas de convergencia del mismo grupo se deben colocar junto en el programa, conectado en el riel de poder sin cualquier otra lógica en el medio.
- Las etapas pueden ocurrir en cualquier orden dentro de un grupo de convergencia de arriba para abajo. No importa cuál etapa es la última a entrar el grupo, porque toda etapa de convergencia debe estar activa antes de la última etapa que tenga el flujo de poder.
- La última etapa de convergencia de un grupo puede tener lógica dentro de la etapa. Sin embargo, esta lógica no se ejecutará hasta que todas etapas de convergencia del grupo sean activadas.
- El salto de convergencia (CVJMP) es el método destinado a ser usado para hacer una transición del grupo de etapas de convergencia a la próxima etapa. El CVJMP repone todas etapas de convergencia del grupo, y activa la etapa denominada en el salto.
- La instrucción CVJMP debe sólo ser usada en una etapa de convergencia, y es inválida en etapas regulares o iniciales.
- Las etapas de convergencia o instrucciones de CVJMP no se pueden usar en rutinas de subprogramas o interrupción.

Instrucciones de RLL^{PLUS}

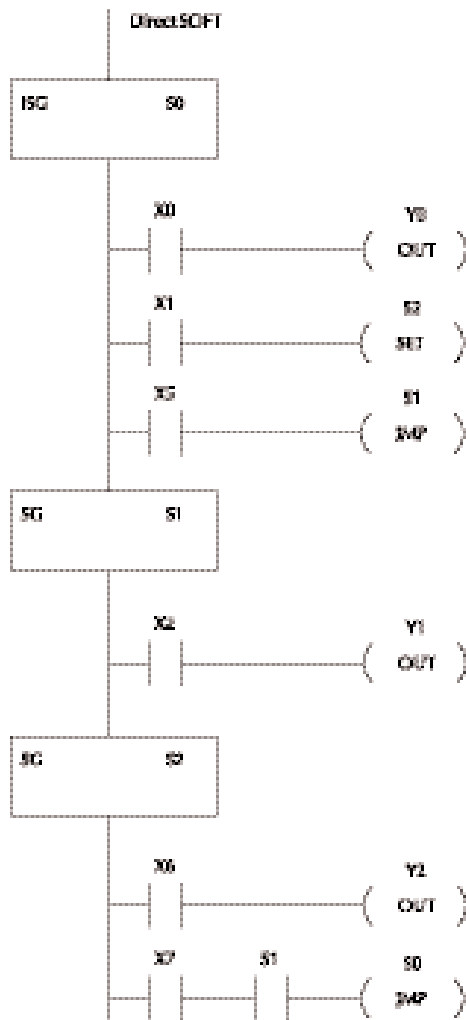
Etapa (o Stage) (SG)

Las instrucciones de etapa se usan para crear programas estructurados de RLL^{PLUS}. Las etapas son los segmentos de programa que pueden ser activados por una lógica de transición, un salto o una etapa que se ejecuta de una etapa activa. Las etapas se desactivan un barrido después que se ejecuta la lógica de transición, la instrucción de etapa, de salto, o un RESET.



Tipo de datos del operando	Rango del DL06
	aaa
Etapa S	0-1777

El ejemplo siguiente es un programa sencillo de RLL^{PLUS}. Este programa utiliza instrucciones etapa inicial ISG, etapa SG y salto JMP para crear un programa estructurado.

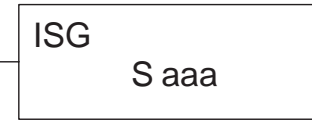


Programador D2-HPP

U	ISG	→	A	0	ENT		
S	STR	→	A	0	ENT		
CO	OUT	→	A	0	ENT		
S	STR	→	B	1	ENT		
X	SET	→	SHIFT	S	RST	C	2
S	STR	→	F	1	ENT		
K	JMP	→	B	1	ENT		
1	SG	→	B	1	ENT		
S	STR	→	C	2	ENT		
CO	OUT	→	B	1	ENT		
1	SG	→	C	2	ENT		
S	STR	→	D	5	ENT		
CO	OUT	→	C	2	ENT		
S	STR	→	H	7	ENT		
V	AND	→	SHIFT	S	RST	B	1
K	JMP	→	A	0	ENT		

Etapa inicial (ISG)

La instrucción Inicial de etapa se usa normalmente como el primer segmento de un programa RLL^{PLUS}. Se permiten múltiples etapas Iniciales en un programa. Ellos serán activos cuando la CPU entra al modo RUN para tener un punto de partida en el programa.

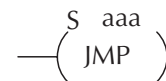


Tipo de datos del operando	Rango del DL06
	aaa
Etapa.....S	0-1777

Las etapas iniciales son activadas también por la lógica de transición, un salto o una etapa fija ejecutados de una etapa activa.

El salto o JUMP (JMP)

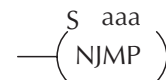
La instrucción de salto permite al programa hacer una transición de una etapa activa que contiene la instrucción del salto a otra etapa (especificada en la instrucción). El salto ocurre cuando la lógica de entrada es verdadera. La etapa activa que contiene el salto se desactivará 1 barrido mas tarde.



Tipo de datos del operando	Rango del DL06
	aaa
Etapa.....S	0-1777

No Salto (NJMP)

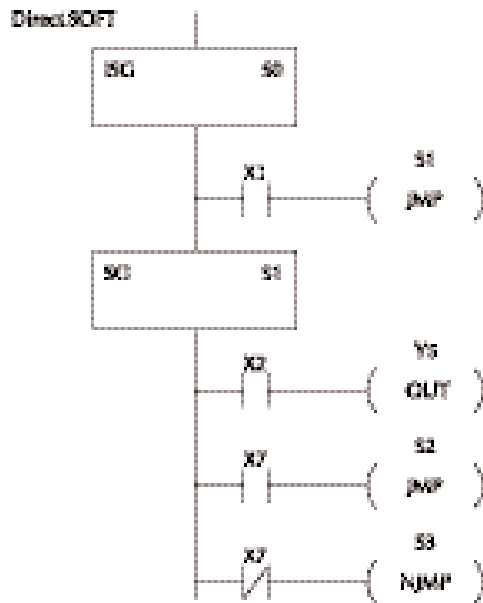
La instrucción de No Salto permite al programa hacer una transición de una etapa activa que contiene la instrucción de salto a otra que se especifica en la instrucción. El salto ocurrirá cuando la lógica de entrada está apagada. La etapa activa que contiene el No Salto se desactivará 1 barrido después que se ejecuta la instrucción de No Salto.



Tipo de datos del operando	DL06 Range
	aaa
Etapa.....S	0-1777

En el ejemplo siguiente, la etapa ISG0 sólo estará activa cuando la ejecución del programa comienza.

Cuándo X1 esté ON, la ejecución del programa saltará desde la etapa Inicial 0 a etapa 1.



Programador DZ-HPP

0	ISC	→	A	0	ENT			
1	STR	→	B	1	ENT			
2	JMP	→	B	1	ENT			
3	SGI	→	B	1	ENT			
4	STR	→	C	2	ENT			
5	OUT	→	F	5	ENT			
6	STR	→	H	7	ENT			
7	JMP	→	C	2	ENT			
8	SHFT	→	D	8	ENT			
	N TMR	SHFT	K	JMP	→	D	8	ENT

Etapa de convergencia (CV) y el salto de convergencia (CVJMP)

La instrucción de etapa de convergencia se usa para agrupar ciertas etapas juntas definiéndolas como etapas de convergencia.

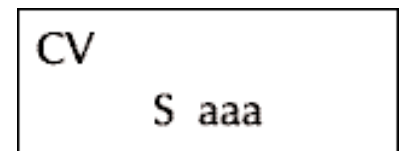
Cuándo todas las etapas de convergencia dentro de un grupo se hacen activas, será ejecutada la instrucción CVJMP (y cualquier lógica adicional en la etapa final de CV).

Todas las etapas CV anteriores debe estar activas antes que la lógica final de etapas CV se puedan ejecutar.

Todas las etapas de convergencia se desactivan un barrido después que se ejecuta la instrucción CVJMP.

Sólo son permitidas instrucciones adicionales de lógica que siguen la última instrucción de etapa y antes de la instrucción CVJMP. Se permiten múltiples instrucciones CVJMP.

Las etapas de convergencia se deben programar en el cuerpo principal del programa de la aplicación. Esto significa que no pueden ser programadas en subrutinas o Subprogramas de interrupción.

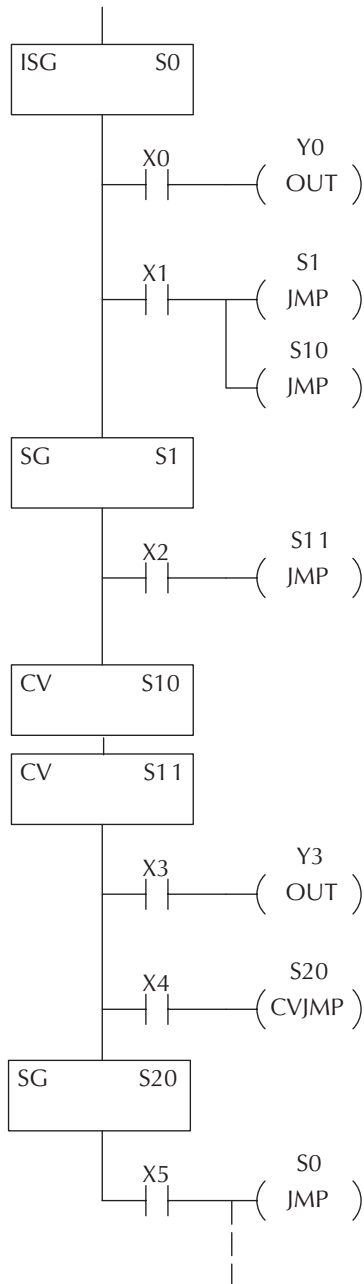


Tipo de datos del operando	Rango del DL06
Etapa..... S	aaa 0-1777

Capítulo 7: Programación por etapas

En el ejemplo siguiente, cuando las etapas de convergencia S10 y S11 están activas al mismo tiempo, se ejecutará la instrucción CVJMP cuando X4 esté ON. La instrucción CVJMP desactivará S10 y S11, y activará S20. Luego, si X5 está ON, la ejecución del programa saltará de vuelta a la etapa inicial, S0.

Direct SOFT



Programador D2-HPP

U ISG	→	A 0	ENT					
\$ STR	→	A 0	ENT					
GX OUT	→	A 0	ENT					
\$ STR	→	B 1	ENT					
K JMP	→	B 1	ENT					
K JMP	→	B 1	A 0	ENT				
² SG	→	B 1	ENT					
\$ STR	→	C 2	ENT					
K JMP	→	B 1	B 1	ENT				
SHFT	C 2	V AND	→	B 1	A 0	ENT		
SHFT	C 2	V AND	→	B 1	B 1	ENT		
\$ STR	→	D 3	ENT					
GX OUT	→	D 3	ENT					
\$ STR	→	E 4	ENT					
SHFT	C 2	V AND	SHFT	K JMP	→	C 2	A 0	ENT
² SG	→	C 2	A 0	ENT				
\$ STR	→	F 5	ENT					
K JMP	→	A 0	ENT					

7

Llamada de bloque (BCALL)

Las instrucciones BCALL se usan para activar un bloque de etapas. Las instrucciones Llamada de Bloque BCALL, Bloque BLK y Fin de Bloque BEND se deben usar juntas. La instrucción BCALL se usa para activar un bloque de etapa. Hay varias cosas que es necesario saber acerca de la instrucción BCALL.

- Usa números C - La instrucción BCALL aparece como una bobina de salida, pero no se refiere verdaderamente a un número de etapa como se podría pensar. En vez de eso, el bloque se identifica con un relevador de control (Caaa). Este relevador de control no se debe usar como una salida en ninguna otra parte del programa.
- Debe permanecer activa - La instrucción BCALL verdaderamente controla todas las etapas entre las instrucciones BLK y BEND aún después que las etapas dentro del bloque han comenzado a ejecutarse. La instrucción BCALL debe permanecer activa o todas las etapas en el bloque se apagarán automáticamente. Si la instrucción BCALL o la etapa que contiene la instrucción BCALL se desactiva, entonces las etapas en el bloque definido se apagarán automáticamente.
- Activa la primera etapa de bloque - Cuando se ejecuta la instrucción BCALL se activa automáticamente la primera etapa que sigue a las instrucciones BLK.



Tipo de datos del operando	Rango del DL06
	aaa
Relevador de control S	0-1777

BLOCK (BLK)

La instrucción de bloque BLK es una etiqueta que marca el comienzo de un bloque de etapas que se pueden activar como un grupo. Una instrucción de etapa debe seguir inmediatamente la instrucción de Bloque BLK. No se permiten instrucciones iniciales de etapa en un bloque. El relevador de control (Caaa) especificado en la instrucción BLK no se debe usar como una salida en ningún otro lugar en el programa.



Tipo de datos del operando	Rango del DL06
	aaa
Relevador de control S	0-1777

El fin de bloque (BEND)

La instrucción Fin de bloque BEND es una etiqueta usada con la instrucción de bloque. Marca el fin de un bloque de etapas. No hay operando con esta instrucción. Sólo se permite una instrucción Fin de Bloque BEND por bloque.

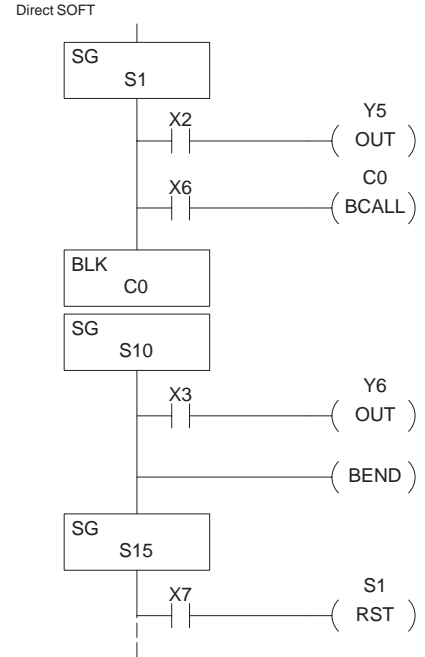


En este ejemplo, la llamada de bloque se ejecuta cuando la etapa 1 es activa y X6 está ON.

La llamada de bloque activa automáticamente la etapa S10, que sigue inmediatamente la instrucción de bloque.

Esto permite que las etapas entre S10 y la instrucción Fin de Bloque operen como programado. Si se apaga la instrucción BCALL, o si la etapa que contiene la instrucción BCALL se apaga, entonces todas las etapas entre las instrucciones BLK y BEND son apagadas automáticamente.

Si examina S15, notará que X7 podría desactivar la etapa S1, que incapacitaría la instrucción BCALL, así desactivando todas las etapas dentro del bloque.



7

Programador D2-HPP

SG	→	S(SG)	1	ENT					
STR	→	X(IN)	2	ENT					
OUT	→	Y(OUT)	5	ENT					
STR	→	X(IN)	6	ENT					
SHFT	B	C	A	L	L	→	C(CR)	0	ENT
SHFT	B	L	K	→	C(CR)	0	ENT		
SG	→	S(SG)	1	0	ENT				
STR	→	X(IN)	3	ENT					
OUT	→	Y(OUT)	6	ENT					
SHFT	B	E	N	D	ENT				
SG	→	S(SG)	1	5	ENT				
STR	→	X(IN)	7	ENT					
RST	→	S(SG)	1	ENT					

Preguntas y respuestas acerca de la programación por etapas

Incluimos la siguientes preguntas comúnmente hechas en la programación de etapas como una ayuda a nuevos estudiantes. Todos los asuntos en las preguntas se cubren en más detalle en este capítulo.

¿ Qué hace la programación por etapas que no se pueda hacer con programas regulares?

Respuesta: Las etapas le permiten identificar todos los estados de su proceso antes que comience a programar. Este enfoque es más organizado, porque usted divide un programa en secciones. Como etapas, éstas secciones de programas son activas sólo cuando se necesitan por el proceso. La mayoría de los procesos se pueden organizar en una sucesión de etapas, conectada por transiciones basadas en eventos.

¿Qué son los bits de etapa?

Respuesta: Un bit de etapa es un solo bit en el registro de imagen de la CPU, representando la posición activa o inactiva de la etapa en el tiempo real. Por ejemplo, el bit de la etapa 0 es referenciado como "S0". Si S0 = 0, entonces los renglones en la etapa 0 son evitados (no ejecutado) en cada barrido de la CPU. Si S0 = 1, entonces los renglones en la etapa 0 son ejecutados en cada barrido de la CPU. Los bits de etapa, cuando se usan como contactos, permiten que una parte del programa controle otra parte detectando estados de etapa activa o inactiva.

¿ Cómo una etapa llega a ser activa?

Respuesta: Hay tres maneras:

- Si la etapa es una etapa inicial (ISG), es automáticamente activo en la energización.
- Otra etapa puede ejecutar una instrucción etapa JMP nombrando esta etapa, que la hace activa en la próxima ocurrencia en el programa.
- Un renglón de programa puede ejecutar una instrucción Fija del Bit de la etapa (tal como Fijo S0).

¿ Cómo una etapa llega a ser inactiva?

Respuesta: Hay tres maneras:

- las etapas (SG) son automáticamente inactivas en la energización.
- Una etapa puede ejecutar una instrucción de etapa JMP, colocando el bit de etapa a 0.
- Un renglón en el programa puede ejecutar la instrucción RESET Bit de etapa (tal como Reset S0).

¿ Como es la técnica de flujo de poder en transiciones de etapa?

Respuesta: El método del flujo de poder de conectar etapas adyacentes (directamente encima o abajo en el programa) es realmente lo mismo que la instrucción de salto de etapa JMP ejecutada en la etapa arriba, nombrando la etapa abajo. Las transiciones de flujo de poder son más difíciles de modificar en *DirectSOFT* y destacamos esto separadamente de las dos preguntas anteriores.

¿ Puedo tener una etapa que es activa solamente por un barrido?

Respuesta: Sí, pero esto no es un uso normal para una etapa. En vez de eso, haga un renglón activo en 1 barrido por inclusive una instrucción del Salto de etapa en el fin del renglón. Luego el código ejecutará en el último barrido antes que la etapa salte a un nuevo.

¿ No es una etapa JMP igual que una instrucción regular GOTO usada en DirectSoft?

Respuesta: No, es muy diferente. Una instrucción GOTO manda la ejecución del programa inmediatamente a la localización del código denominada por el GOTO. Una etapa JMP repone simplemente el bit de etapa de la etapa actual, al preparar el camino bit de etapa denominó en la instrucción de JMP. Los bits de la etapa son 0 o 1, determinando el inactivo/la posición activa de las etapas correspondientes. Una etapa JMP tiene los resultados siguientes:

- Cuando el JMP se ejecuta, el resto del renglones actual de la etapa se ejecuta, aunque ellos residan pasado (abajo) la instrucción de JMP. En el siguiente barrido, eso prepara no es ejecutado, porque es inactivo.
- La etapa denominada en la instrucción de la etapa JMP se ejecutará sobre su próxima ocurrencia. Si localizó el pasado (abajo) la etapa actual, se ejecutará en el mismo barrido. Si localizó antes (arriba) la etapa actual, se ejecutará en el siguiente barrido.

¿Cómo puedo saber cuando se usa la instrucción JMP en vez de la instrucción SET o RESET un bit de etapa?

Respuesta: Estas instrucciones se usan según la topología de diagrama de estado que usted ha derivado:

- Use una instrucción de etapa JMP para una transición de estado... moverse de un estado a otro.
- Use una instrucción SET bit de etapa cuando el estado actual crea una sucesión paralela nueva de estado o etapa, o cuando un estado de supervisión comienza una sucesión del estado bajo su comando.
- Use la instrucción RESET bit cuando el estado actual es el último estado en una sucesión y su tarea es completa o cuando un estado de supervisión finaliza una sucesión del estado bajo su comando.

¿Qué es una etapa inicial, y cuándo se puede usar?

Respuesta: Una etapa (ISG) inicial es automáticamente activa en la energización. Después, trabaja así como cualquier otra etapa. Usted puede tener el múltiplo las etapas iniciales, si requirió. Use una etapa inicial para escalera que siempre debe ser activa, o cuando un punto de partida

¿Puedo tener renglones de programa fuera de las etapas, de modo que estén siempre ON?

Respuesta: Es posible, pero no es buena práctica de diseño de software. Coloque el código que siempre debe estar activo en una etapa inicial y no resete esa etapa ni use una instrucción de etapa JMP dentro de el. Puede comenzar otras sucesiones de etapa en el tiempo apropiado colocando ON el correspondiente bit (s) de etapa.

¿ Puedo tener más que una etapa activa a la vez?

Respuesta: Sí, y esto es una ocurrencia normal en muchos programas. Sin embargo, es importante organizar su aplicación en procesos separados, cada un compuesto de etapas. Y un diseño bueno de proceso será en su mayor parte secuencial, con sólo una etapa ON a la vez. Sin embargo, todos los procesos en el programa pueden estar activos simultáneamente.

OPERACIÓN DEL CONTROL PID CON EL DL06



CAPÍTULO 8

En este capítulo...

Funciones de control PID en el PLC DL06	8-2
¿Qué es control PID?	8-4
Introducción al control PID en el PLC DL06	8-6
Operación del control PID	8-9
Diez pasos para obtener un buen control de proceso	8-16
Configuración lazos de control PID en el PLC DL06	8-18
Configurando las alarmas del control PID	8-36
Sintonizando lazos de control PID	8-42
Usando las funciones especiales del control PID	8-52
Filtro de la variable de proceso PV	8-57
Generador de Ramp/Soak	8-60
Ejemplo de Ramp/Soak con DirectSOFT	8-65
Control en cascada	8-67
Control de pulso con PWM	8-70
Control Feedforward	8-72
Ejemplo de programa ladder con control PID	8-74
Consejos para búsqueda de fallas en PID	8-77
Glosario de la terminología del control PID	8-79
Bibliografía	8-81

Funciones de control PID en el PLC DL06

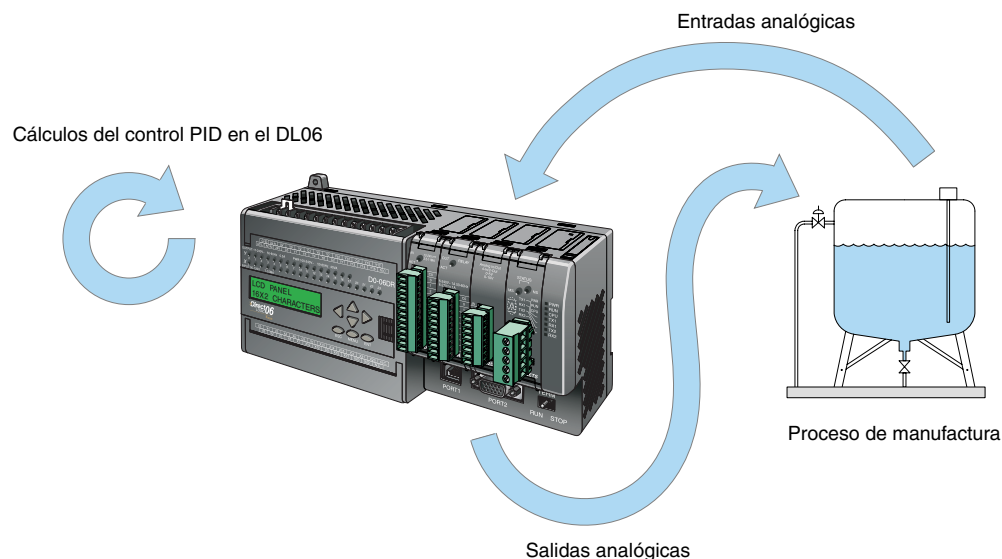
Características principales

El control de lazos cerrados en un proceso con el PLC DL06 ofrece un conjunto sofisticado de funciones para solucionar muchas necesidades de la industria.

Las características principales son

- hasta 8 lazos, con frecuencia programable individual de muestreo
- capacidad de control PID en los modos Manual/Automático/En cascada disponibles
- Dos tipos de modos de transferencia "sin saltos" disponible
- Alarmas de todos los tipos
- Generador Ramp/Soak (Rampa y valor constante) con hasta 16 segmentos
- Sintonía automática del control PID

El PLC DL06 tiene capacidad de controlar el proceso con lazos de control PID, además de la ejecución del programa ladder. Se puede escoger configurar hasta ocho lazos. Todo el cableado de sensores analógicos y actuadores se conectan directamente a los módulos analógicos. Todas las variables de proceso, los valores de ganancia, los niveles del alarma, etc., asociados con cada lazo residen en una tabla de variables llamada *tabla PID* en la CPU. La CPU DL06 lee las entradas de la variable de proceso (PV) durante cada barrido, luego hace los cálculos de control PID durante un período dedicado en cada barrido del PLC, actualizando el valor de la salida de control (CV) a ciertos tiempos predefinidos. El control usa un algoritmo Proporcional Integral Derivativo (PID) para generar la salida de control. Este capítulo describe cómo es la operación del control PID y lo que usted debe hacer para configurar y sintonizar la entradas analógicas.



La mejor herramienta para configurar los lazos en el DL06 es el software *DirectSOFT*, versión 4.0, o posterior. *DirectSOFT* usa cajas de diálogo para ayudarlo a configurar los lazos individuales. Después que completa la configuración, usted puede usar la ventana PID View en *DirectSOFT* para sintonizar cada lazo. Las selecciones de configuración y sintonía que usted hará se almacenan en la memoria RAM del PLC DL06, que es retentiva. Los parámetros de lazos pueden ser salvados también al disco para posterior recuperación.

Características de función PID	Especificaciones
Número de lazos	Seleccionable, 8 lazos
Memoria necesaria	Necesita 32 palabras por lazo, 64 si se usa ramp/soak
Algoritmo PID	Modo de Posición o Velocidad
Polaridad de la CV	Seleccionable entre acción Directa o Inversa
Curvas de error	Seleccionable como error lineal, raíz cuadrada del error o error al cuadrado
Tasa de actualización del cálculo PID	0,05 a 99,99 segundos
Tasa mínima de actualización de lazos	0,05 segundos en 1 a 4 lazos, 0,1 segundos en 5 a 8 lazos
Modos de operación	Automático, manual (control por el operador) o en cascada
Generador Ramp/Soak	Hasta 8 etapas de ramp/soak (16 segmentos) por lazo
Curvas de PV	Seleccionable como lineal o raíz cuadrada (para entrada de flujo)
Límites de señal de referencia	Se puede especificar los valores límites mínimos y máximos de la señal de referencia
Límites de PV	Se puede especificar los valores límites mínimos y máximos del valor PV
Ganancia	En el rango de 0,01 hasta 99,99
Valor RESET (integral)	Rango de 0,1 hasta 999,8 en segundos o minutos
Ganancia derivativa	Rango de 0,01 hasta 99,99 segundos
Límites de derivativa (rate)	Ganancia derivativa de 1 a 20
Transferencia de salto I	Automáticamente inicializa el bias y la señal de referencia cuando el control cambia de manual a Automático.
Transferencia de salto II	Automáticamente hace el bias igual a la salida de control cuando el control cambia de manual a Automático.
STEP Bias	Proporciona ajuste proporcional de tendencia para cambios grandes de señal de referencia
Anti-windup	Para el modo de posición de PID, esta función inhibe la acción de integrador cuando la salida de control alcanza 0% o 100 % (acelera la recuperación del lazo cuando la salida se recupera de la saturación)
Error de banda muerta	Especifica una tolerancia (más y menos) para el término de error (SP-PV), de modo que no haya cambio en el valor de salida de control

Características de alarma	Especificaciones
Banda muerta	Rango de 0,1 hasta 5% de banda muerta de alarma en todas las alarmas
Puntos de alarma PV	Configuración de alarmas PV Low-Low, Low, High y High-High
Desvío de PV	Configuración de alarmas en 2 rangos de desvío desde la referencia
Tasa de cambio	Detecta cuando el PV excede un límite de tasa de cambio

¿Que es control PID?

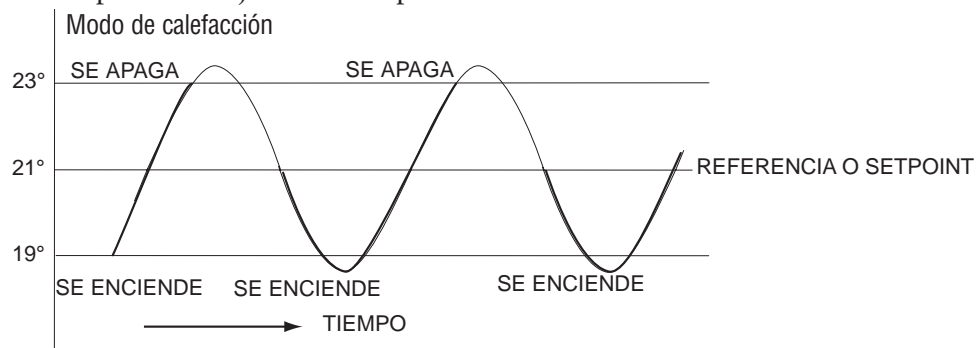
En esta discusión, explicaremos qué es y porqué se usa el control PID en control de procesos en vez de proporcionar control simplemente usando una entrada analógica y una salida discreta.

Hay muchos tipos de controladores o controladores analógicos disponibles, y la selección apropiada dependerá de la aplicación particular. Hay dos tipos de controladores analógicos que se utilizan en la industria:

1. El controlador ON-OFF, designado a veces un controlador de lazo abierto.
2. El controlador PID, a veces llamado un controlador de lazo cerrado.

Sin importar el tipo, los controladores analógicos requieren señales de entrada de sensores electrónicos tales como señales de presión, presión diferencial, nivel, flujo o termopares. Como ejemplo, uno de los controles analógicos más comunes de control está situado en su casa, para controlar la temperatura.

Usted desea que su casa esté a una temperatura cómoda, de modo que usted ajusta la temperatura en un termóstato a una temperatura deseada (referencia o setpoint). Luego selecciona el modo de calentamiento o aire acondicionado. Un dispositivo de detección de la temperatura, normalmente un termistor, está situado dentro del termóstato. Si el termóstato se ajusta para calentamiento y la temperatura de referencia se ajusta a 21° Celsius, la estufa será encendida para suministrar calor en, digamos, 2° Celsius debajo del setpoint. En este caso, se encendería en 19° Celsius. Cuando la temperatura alcanza 23° Celsius, sobre el setpoint, la estufa se apagará. En el ejemplo opuesto, si el termóstato se ajusta para aire acondicionado (para refrescar el ambiente), el termóstato encenderá la unidad de aire acondicionado de una forma opuesta a la de calentamiento. Por ejemplo, si el termóstato se ajusta para enfriar a 24° Celsius, la unidad de aire acondicionado se encenderá cuando la temperatura detectada alcanza 2° Celsius sobre el setpoint o a los 26° Celsius, y se apagará cuando la temperatura alcanza 22° Celsius. Éste sería considerado un controlador ON-OFF. La forma de onda abajo muestra la acción del ciclo de calefacción. Observe que la temperatura sube mas allá del punto de apagado, y también la temperatura baja más en el punto donde debe encenderse.



El controlador ON-OFF se usa en algunos usos de control industriales, pero no es práctico en la mayoría de procesos de control industriales.

El controlador de proceso más común que se usa en industria es el controlador PID.

El controlador PID controla un lazo de realimentación continuo que mantiene la salida de proceso (variable de control) a un valor deseado tomando la acción correctiva siempre que haya un desvío o error del valor deseado (setpoint) de la variable de proceso (PV) por ejemplo, flujo,

temperatura, voltaje, etc. Ocurre un "error" cuando un operador cambia manualmente el setpoint o cuando un evento (una válvula se abrió o se cerró, etc.) o una perturbación (agua fría, viento, etc.) cambia la carga en el proceso, causando un cambio en la variable de proceso.

El controlador PID recibe señales desde los sensores y calcula una acción correctiva al actuador con un algoritmo basado en un valor proporcional al error (proporcional), a la suma de todos los errores anteriores (integral) y a la tasa de cambio del error (derivativo).

Podemos asimilar el controlador de PID en forma simple con un ejemplo. Tomemos el control automático de velocidad en un automóvil como ejemplo. Digamos que estamos viajando en una carretera en un automóvil equipado con este control. El conductor decide usarlo y llega a la velocidad deseada, digamos 100 kilómetros por hora. Una vez que se alcance esta velocidad, aprieta el botón de control que deja la velocidad ajustada a 100 km/h, que es la referencia o setpoint. Ahora, digamos que el automóvil viaja a 100 km/h constante hasta que comienza a subir a una colina. Cuando el automóvil sube, tiende a bajar la velocidad. El sensor de velocidad detecta esto y hace que la válvula aumente el combustible al motor. El vehículo aceleró para mantener 100 km/h sin que cause que el automóvil salte y luego llega a la parte superior de la colina a la velocidad del sistema. Cuando el coche se nivela después de alcanzar la parte superior, acelerará. El sensor de velocidad detecta esto y le indica a la válvula reguladora para proporcionar menos combustible al motor, y así, el motor deja de entregar toda la fuerza permitiendo que el coche mantenga la velocidad de 100 km/h. ¿Cómo se aplica este ejemplo al control PID? Veamos esto en función de los parámetros P, I y D:

- **Proporcional** - Se refiere comúnmente como **ganancia proporcional**. El término proporcional es la acción correctiva que es proporcional al "error", es decir, a la diferencia entre la variable manipulada y la referencia, y el valor de corrección es proporcional a la **ganancia proporcional** multiplicado por el error. En términos matemáticos:

Acción proporcional = Error x Ganancia Proporcional

Siendo el Error = Referencia (SP) - Variable de proceso (PV)

Aplicando esto al control de velocidad, la velocidad fue ajustada a 100 Km/h, que es el valor de referencia (setpoint). El sensor de velocidad detecta la velocidad real del coche y envía esta señal al controlador como la variable de proceso (PV). Cuando el coche está en una carretera a nivel, la velocidad se mantiene en 100 Km/h, y no hay ningún error, puesto que el error sería $SP - PV = 0$. Cuando el coche va subiendo la colina, el sensor de velocidad detectó una desaceleración del coche, $SP - PV = \text{error}$ que es mayor que cero (Por ejemplo, $100 - 95$). La ganancia proporcional haría la salida del controlador de velocidad traer el coche de nuevo al setpoint de 100 Km/h. Ésta sería la salida controlada.

- **Integral** - Este término se refiere a menudo como acción de **reset**. Proporciona una compensación adicional a la salida de control, que causa un cambio en proporción con la suma de los valores de error durante el tiempo. Es decir el es la integral de los valores del error durante el tiempo.
- **Derivativo** - Este término se refiere como **rate** o tasa. La acción derivativa agrega una compensación a la salida de control, que causa un cambio proporcional a la tasa de cambio del error. Su trabajo es anticipar el crecimiento probable del error y generar una contribución a la salida por adelantado.

Introducción al control PID con el PLC DL06

El DL06 es capaz de controlar una variable de proceso tal como mencionado anteriormente, por ejemplo el control de una variable de proceso en un nivel dado (setpoint), tal como temperatura de agua, incluso si hay perturbaciones (tal como ingreso de agua fría) en el proceso.

El PLC DL06 tiene capacidad de aceptar directamente señales de sensores electrónicos, tales como termopares, presión, nivel, etc. Estas señales se pueden usar en sistemas de control con algoritmos matemáticos.

Además, el DL06 tiene ya algoritmos incorporados de control PID de acuerdo a teorías de control clásicas que se pueden poner en ejecución. La función básica del control de proceso de lazo cerrado PID es mantener un valor del proceso cerca del valor de referencia deseado. En general, el proceso se desvía de la referencia deseada como resultado de cambios de material de la carga o perturbaciones y de la interacción con otros procesos. Durante este control, se mide continuamente la condición corriente de las características de proceso (nivel, temperatura, control de velocidad motor, etc.) como variable de proceso (PV) y se compara con el valor de referencia deseado (setpoint SP). Cuando ocurre un desvío, se genera un error que es la diferencia entre la variable de proceso (valor corriente) y el valor de referencia (valor deseado). Una vez que se detecte un error, la función del lazo de control es modificar la salida analógica del lazo de control para forzar el error a cero.

El control PID del PLC DL06 maneja lazos cerrados usando el algoritmo PID. La salida de control es calculada desde el valor de error como sigue:

$$M(t) = K_c \left[e(t) + 1/T_i \int_0^t e(x) dx + T_d d/dt e(t) \right] + M_o$$

Siendo:

K_c = ganancia proporcional

T_i = Reset o tiempo de la integral

T_d = tiempo derivativo o rate

SP = Setpoint o valor de referencia

PV(t) = variable de proceso en el tiempo "t"

$e(t) = SP - PV(t)$ = desvío de la PV desde la referencia (SP) en el tiempo "t" o error del PV.

$M(t)$ = la salida analógica de control en el tiempo "t"

El módulo de entrada analógico recibe la variable de proceso en forma analógica y también un valor de referencia o setpoint dado por el operador; la CPU calcula el error. El error se usa en el algoritmo para proporcionar la acción correctiva en la salida analógica de control. La función de la acción de control se basa en un control de la salida analógica, que es **proporcional** al valor instantáneo del error. La acción **integral** de control (acción de reset) proporciona una compensación adicional a la salida de control, que causa un cambio proporcional al valor del error acumulado durante el tiempo. La acción **derivativa** de control (cambio de tasa) agrega una compensación a la salida del control, que causa un cambio en proporción con la tasa de cambio del error.

Estos tres modos se usan para proporcionar la acción deseada de control en las formas (P) proporcional, Proporcional-Integral (PI), o Proporcional-Integral-Derivativo (PID) de control.

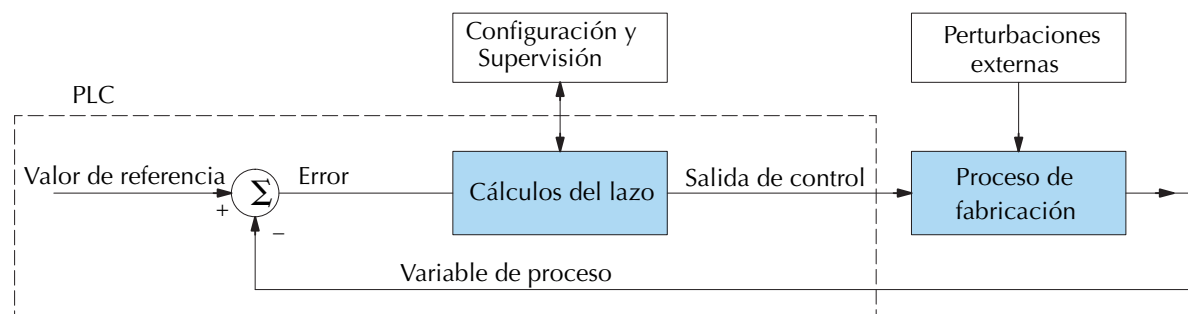
Cada proceso es diferente y es por eso que se necesitan valores diferentes de los parámetros de ganancia, integral y derivativo. Este proceso de encontrar estos valores para cada proceso se llama sintonización o “tuning” en inglés.

Se usan módulos de entradas analógicas estándares de los PLCs DL05/06 para interconectar a transductores de campo para obtener la PV. Estos transductores proporcionan normalmente una señal de 4-20 mA o un voltaje analógico en varios rangos para el lazo de control.

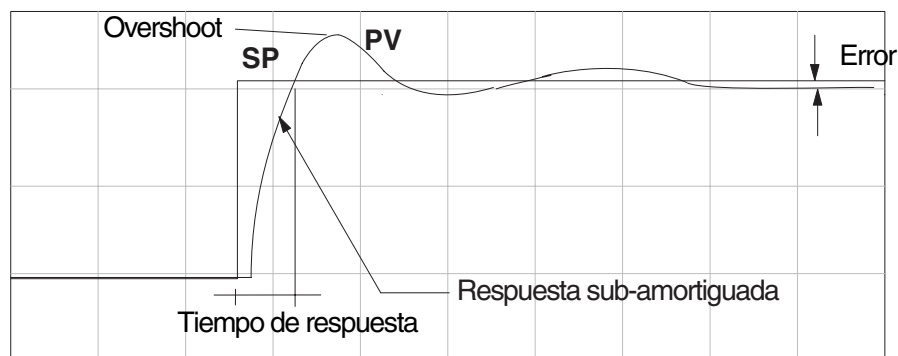
Para el control de la temperatura, se puede conectar directamente un termopar o una resistencia de medición de temperatura RTD con el módulo apropiado. El algoritmo del control PID, que reside en la memoria de la CPU, recibe la información del programa de usuario, es decir, los parámetros de control y de los valores de referencia o setpoints. Una vez que la CPU haga el cálculo de PID, el resultado se puede usar para controlar directamente un actuador conectado con un módulo de salida de corriente 4-20 mA, por ejemplo, para controlar una válvula.

Con *DirectSOFT*, se pueden programar esquemas de control con lógica adicional, incluso control ON-OFF o PWM de tiempo proporcional (por ejemplo, calentadores para el control de temperatura) y actuador de posición (por ejemplo, motor reversible en una válvula). Este capítulo explicará cómo configurar el lazo de control PID en el PLC DL06, cómo poner el software en ejecución y cómo sintonizar un lazo de control.

El diagrama de bloque siguiente muestra las partes más importantes de un lazo de control de PID. La trayectoria del PLC al proceso de fabricación y de nuevo al PLC es un control de lazo cerrado.



La figura siguiente muestra gráficamente una respuesta típica a este controlador y se muestran algunas definiciones explicadas en la próxima página.



Definiciones de procesos de control

Proceso de fabricación – El conjunto de acciones que agrega valor a las materias primas. El proceso puede implicar cambios físicos y/o cambios químicos al material. Los cambios hacen el material más útil para un propósito particular, utilizado en última instancia en un producto final.

Variable de proceso PV– Una medida de una cierta característica física de las materias primas. Se hacen las medidas usando un cierto tipo de sensor. Por ejemplo, si el proceso de fabricación utiliza un horno, usted deseará muy probablemente controlar temperatura. La temperatura es una variable de proceso.

Valor de referencia SP – La cantidad teóricamente perfecta de la variable de proceso o la cantidad deseada que rinde el mejor producto. El operador sabe este valor y lo ajusta manualmente o lo programa en el PLC para uso automatizado más tarde.

Salida de control CV– El resultado del cálculo del lazo, que se convierte en un comando para el proceso (tal como el nivel de calentamiento en un horno).

Error – La diferencia algebraica entre el variable de proceso y la referencia. Éste es el error del lazo de control, y es igual a cero cuando la variable de proceso es igual al valor de la referencia (valor deseado). Un lazo de control funcionando bien puede mantener una magnitud pequeña del error.

Variable manipulada – La variable usada para afectar la variable controlada. Por ejemplo, la cantidad de combustible usada en un horno puede ser la variable manipulada, para controlar la variable temperatura.

Perturbaciones externas – Las fuentes de error imprevisibles que el sistema de control trata de cancelar compensando sus efectos. Por ejemplo, si la entrada del combustible en un horno es constante un horno funcionará más caliente durante el tiempo caliente que durante el tiempo frío. Un sistema de control automático del horno debe compensar este efecto para mantener una temperatura constante del horno durante cualquier estación. Así, el clima (que no es muy confiable), es una fuente de perturbaciones a este proceso.

Elemento de control final – El aparato usado para controlar la variable manipulada. Un ejemplo es la válvula que controla el flujo de un combustible.

Tiempo de atraso (Lag time) – El tiempo en que responde un proceso a un cambio en la variable manipulada. Ésto también es conocido como la capacitancia del sistema. Cuando Ud. está en una ducha y Ud aumenta el agua caliente, el tiempo que lleva para que el agua se caliente es el tiempo de atraso.

Tiempo muerto (Dead time) - El tiempo que toma el controlador para que sea reconocido un cambio en el proceso.

Tiempo de respuesta – El tiempo que toma en producir un cambio en la variable de proceso cuando se ha dado un cambio escalón en la referencia en el proceso.

Overshoot – Es la onda que sucede en la variable de proceso al haber una transición de valor de la referencia desde un valor más bajo a uno más alto; el valor de la variable de proceso que excede al valor de referencia en la transición de un valor mas bajo a uno más alto de la referencia es el valor de overshoot, en por ciento. Vea la figura de la página anterior.

Undershoot – Es la onda de la PV que sucede al haber una transición de valor de la referencia desde un valor más alto a uno más bajo.

Configuración del lazo – Las selecciones iniciadas por un operador que configuran y optimizan el funcionamiento de un lazo de control.

Supervisión del lazo – Las funciones que permiten que un operador observe el estado y el funcionamiento de un lazo de control. Esto se utiliza conjuntamente con la configuración del lazo para optimizar el funcionamiento de éste (para reducir al mínimo el error).

Operación del control PID

El algoritmo Proporcional-Integral-Derivativo (PID) se usa extensamente en control de proceso. El método de control PID se adapta bien a soluciones electrónicas, ya sea con componentes analógicos o digitales (CPU). El PLC DL06 resuelve las ecuaciones de control PID en forma digital solucionando el algoritmo en software. Los módulos de E/S sirven para convertir solamente señales electrónicas en forma digital (o viceversa).

El DL06 usa dos modos de control de PID: "posición" y "velocidad". Estos términos se refieren generalmente a situaciones de control de movimiento, pero aquí las utilizamos en un sentido diferente:

- Algoritmo de *posición* de PID – Se calcula la salida de control de modo que corresponda al desplazamiento (posición) del PV desde el SP (término de error).
- Algoritmo de *velocidad* de PID – Se calcula la salida de control para representar la tasa de cambio (velocidad) para que el PV llegue a ser igual al SP.

Modo "Position" del algoritmo PID

En relación a la ecuación de la salida de control en la página 8-6, la CPU DL06 aproxima la salida $M(t)$ usando una forma discreta de posición del algoritmo PID.

Sea:

- T_s = Período de muestreo
- K_c = Ganancia proporcional
- $K_i = K_c * (T_s/T_i)$ = Coeficiente de la parte integral
- $K_r = K_c * (T_d/T_s)$ = Coeficiente de la parte derivativa
- T_i = Reset o tiempo de la parte integral
- T_d = Rate o tiempo de la parte derivativa
- SP = Referencia (Setpoint)
- PV_n = Variable de proceso en el muestreo n ésimo
- $e_n = SP - PV_n$ = Error en el muestreo n ésimo
- M_o = Valor en el cual se ha iniciado la salida del controlador PID

Entonces :

M_n = salida de control en el muestreo n ésimo

$$M_n = K_c * e_n + K_i \sum_{i=1}^n e_i + K_r (e_n - e_{n-1}) + M_o$$

Esta forma de la ecuación de PID se refiere como el modo de posición puesto que se calcula la posición corriente del actuador. El modo de velocidad de la ecuación de PID calcula la velocidad del cambio en la posición del actuador. La CPU modifica levemente la ecuación estándar para usar la derivada de la variable de proceso en vez del error, como sigue:

$$M_n = K_c * e_n + K_i \sum_{i=1}^n e_i + K_r (PV_n - PV_{n-1}) + M_o$$

Estos dos modos son equivalentes a menos que se cambie el setpoint. En la ecuación original, un cambio grande de una entrada escalón en el setpoint causará un cambio correspondientemente grande en el error dando por resultado un salto al proceso debido a la acción derivativa. Este salto no está presente en el segundo modo de la ecuación.

El algoritmo del PLC DL06 también combina la integral y la salida inicial en un solo valor llamado el bias (M_x). Esto nos lleva al siguiente sistema de ecuaciones:

$$M_{x_0} = M_0$$

$$M_x = K_i * e_n + M_{x_{n-1}}$$

$$M_n = K_c * e_n - K_r(PV_n - PV_{n-1}) + M_{x_n}$$

El DL06 por defecto mantendrá la salida normalizada M en el rango de 0,0 a 1,0. Esto es hecho forzando M al valor más cercano de 0.0 o 1.0 siempre que la salida calculada caiga fuera de este rango. El DL06 también le permite especificar los valores límites de salida mínimo y máximo (dentro del rango 0 a 4095 si usa una resolución de 12 bits unipolar).



NOTA: Las ecuaciones y los algoritmos, en este capítulo, son solamente para referencia. El análisis de estas ecuaciones se puede encontrar en la mayoría de buenos libros de texto acerca de control de proceso.

Protección de Reset Windup (saturación de la integración)

Puede ocurrir este efecto si se configura acción de reset (valor integral) en el control PID y siendo el cálculo de bias M_x :

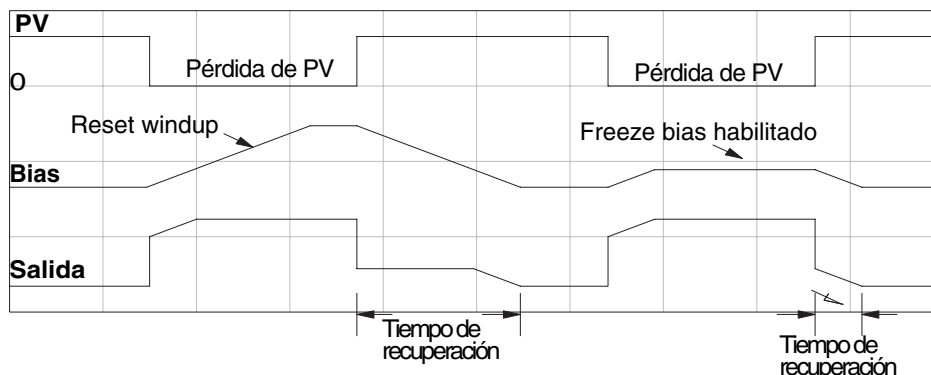
$$M_x = K_i * e_n + M_{x_{n-1}} \text{ (La integración del error)}$$

Por ejemplo, asumamos que la salida del control PID está controlando una válvula y la variable de proceso PV permanece en un cierto valor mayor que la referencia (setpoint). El error negativo (e_n) causará que el bias (M_x) disminuya constantemente hasta que la salida M vaya a 0, lo que cierra la válvula. Sin embargo, ya que el valor del error sigue siendo negativo, el bias continuará disminuyendo siendo siempre más negativo.

Cuando la variable de proceso finalmente vuelve a estar debajo del valor de referencia (SP), la válvula permanecerá cerrada hasta que el error sea positivo por cierto tiempo tal que haga el bias llegue a ser positivo otra vez. Esto causará que la variable del proceso produzca un *undershoot*.

Una forma de solucionar el problema es forzar simplemente el bias en un valor normalizado entre 0,0 y 1,0. La CPU DL06 hace ésto. Sin embargo, si ésta es la única cosa que se hace, entonces la salida no se moverá de 0.0 (así abriendo la válvula) hasta que el PV se haya convertido a un valor menor que el valor de referencia SP . Esto también causará que la variable del proceso produzca un *undershoot*.

La CPU DL06 es programada para solucionar el problema de *overshoot* congelando el valor bias (freeze bias), o ajustando el término bias.



Freeze Bias (Congelar el bias)

Si se selecciona la opción “Freeze Bias” cuando se configura el lazo de PID (discutido más adelante) entonces la CPU simplemente para de cambiar el bias (M_x) siempre que la salida normalizada calculada (M) tenga un valor fuera del intervalo 0 a 1,0.

$$M_x = K_i * e_n + M_{x_{n-1}}$$

$$M = K_c * e_n - K_r(PV_n - PV_{n-1}) + M_x$$

$$\begin{aligned} M_n &= 0 && \text{si } M < 0 \\ M_n &= M && \text{si } 0 \leq M \leq 1 \\ M_n &= 1 && \text{si } M > 1 \end{aligned}$$

$$\begin{aligned} M_{x_n} &= M_x && \text{si } 0 \leq M \leq 1 \text{ o entonces} \\ M_{x_n} &= M_{x_{n-1}} \end{aligned}$$

Así en este ejemplo, probablemente el bias no irá hasta cero de modo que, cuando el PV comienza a bajar de valor, el lazo comienza a abrir la válvula más pronto que sería si se hubiera permitido que el bias fuera a cero. Esta acción tiene el efecto de reducir la cantidad de *overshoot*.

Ajustando el Bias

La acción normal de la CPU es ajustar el valor de bias cuando la salida sale de rango según lo mostrado abajo.

$$M_x = K_i * e_n + M_{x_{n-1}}$$

$$M = K_c * e_n - K_r(PV_n - PV_{n-1}) + M_x$$

$$\begin{aligned} M_n &= 0 && \text{si } M < 0 \\ M_n &= M && \text{si } 0 \leq M \leq 1 \\ M_n &= 1 && \text{si } M > 1 \end{aligned}$$

$$\begin{aligned} M_{x_n} &= M_x && \text{si } 0 \leq M \leq 1 \text{ o entonces} \\ M_{x_n} &= M_n - K_c * e_n - K_r(PV_n - PV_{n-1}) \end{aligned}$$

Al ajustar el bias, la válvula comenzará a abrirse tan pronto como el valor de la variable de proceso PV comience bajar. Si el lazo se sintoniza correctamente, se puede eliminar el *overshoot* totalmente. Si la salida salió del rango debido a un cambio del setpoint, entonces el lazo oscilará probablemente porque debemos esperar que se estabilice el término bias otra vez.

La opción de si usar la acción del lazo por defecto o congelar el bias es dependiente de la aplicación. Si se esperan cambios escalón grandes a la referencia (setpoint), es probablemente mejor seleccionar la opción del freeze bias (vea la página 8-36).

Eliminación de la acción proporcional, integral o derivativa

No es siempre necesario hacer funcionar un lazo de control PID con las tres compensaciones. La mayoría de los lazos requieren solamente los Factores PI o solamente el factor P. Pueden ser eliminadas partes del algoritmo PID eligiendo los valores apropiados para la ganancia (K_c), el reset (T_i) y el rate (T_d) que resultan en control P, PI, PD, I e incluso un ID y un D.

Eliminando la acción Integral El efecto de la acción integral en la salida puede ser eliminado colocando $T_i = 9999$. Cuando se hace esto, el usuario puede entonces controlar manualmente el bias (M_x) para eliminar esta compensación.

Eliminando la acción Derivativa El efecto de la acción derivativa en la salida puede ser eliminado colocando $T_d = 0$ (la mayoría de los lazos no requieren un parámetro D; puede hacer el lazo inestable).

Eliminando la acción Proporcional Aunque no se hace normalmente, el efecto de la ganancia proporcional en la salida puede ser eliminado colocando el valor $K_c = 0$. Ya que K_c es también un multiplicador del coeficiente integral (K_i) y del coeficiente derivativo (K_r), la CPU hace el cálculo de estos valores condicional al valor del K_c como sigue:

$$K_i = K_c * (T_s / T_i) \quad \text{si } K_c \neq 0$$

$$K_i = T_s / T_i \quad \text{si } K_c = 0 \text{ (I o ID solamente)}$$

$$K_r = K_c * (T_d / T_s) \quad \text{si } K_c \neq 0$$

$$K_r = T_d / T_s \quad \text{si } K_c = 0 \text{ (ID o D solamente)}$$

Modo de velocidad del algoritmo PID

El modo estándar de posición del algoritmo PID calcula la posición real del actuador. Un modo alternativo del algoritmo PID calcula la velocidad de cambio en la posición del actuador. Se obtiene esta forma restando la ecuación en el tiempo "n" de la ecuación en el tiempo "n-1".

La ecuación del modo velocidad es dada por:

$$\Delta M_n = M - M_{n-1}$$

Transferencia sin saltos (Bumpless)

Imaginemos una condición donde el lazo de control está en el modo Manual, con una salida de 35%. Luego el operador pasa al modo Automático donde la referencia está en 65%. Esto hace que el error sea de 30%. Esta condición puede causar un cambio brusco de salida o un “salto” a 100% de la salida.

El controlador de lazos del PLC DL06 preve cambios de modo sin saltos en la salida de control. Se logra una transferencia sin saltos del modo manual al modo automático previniendo que la salida de control cambie inmediatamente después del cambio de modo.

Cuando un lazo se cambia de modo manual al modo automático, la referencia (setpoint) y el bias se inicializan como sigue:

Algoritmo de posición PID

$$SP = PV$$

$$Mx = M \text{ (La salida de control se hace igual al bias)}$$

Algoritmo de velocidad PID

$$SP = PV$$

La función de transferencia bumpless del PLC DL06 está disponible en dos tipos: Bumpless I y Bumpless II (Vea la página 8-26). Este modo es seleccionado cuando se configura el lazo.

Alarmas

El DL06 permite que el usuario especifique las condiciones de alarmas que deben ser supervisadas para cada lazo. Las condiciones de alarmas son informadas a la CPU al configurar las alarmas en *DirectSOFT*, usando el diálogo de alarmas en el diálogo **Setup PID**.

Las funciones de alarmas para cada lazo son:

- **Límites** – Especifica hasta cuatro puntos de alarmas de la variable de proceso PV

High-High	PV está encima del límite programado <i>High-High Alarm</i> .
High	PV está encima del límite programado <i>High Alarm</i> .
Low	PV está debajo del límite programado <i>Low Alarm</i> .
Low-Low	PV está debajo del límite programado <i>Low-Low</i> .
- **Alarmas de desvío** – Se puede también especificar una alarma de desvío High-High y Low-Low (desvío anaranjado). Cuando el PV es más alto que el valor de referencia (setpoint) que el límite programado amarillo o anaranjado, se activa el bit correspondiente de alarma en una de las palabras de la tabla PID.
- **Tasa de cambio** – Esta alarma se activa cuando el PV cambia más rápidamente que lo especificado en el límite *rate of change*.
- **Histéresis de alarma del PV** – Las alarmas de límite y de desvío del PV se programan usando valores de umbral. Cuando el valor absoluto o el desvío excede el umbral, el estado de la alarmas se torna verdadero. Las señales del mundo real de PV tienen cierto ruido en ellos, que pueden causar una cierta fluctuación en el valor del PV en la CPU. Cuando el valor del PV cruza un umbral de alarma, sus fluctuaciones harán que la alarma sea intermitente y moleste a operadores de proceso. La solución es utilizar la función de histéresis de alarma del PV.

Modos de funcionamiento del control PID

El controlador PID del PLC DL06 funciona en uno de tres modos, o *Manual, Automático o cascada*.

Manual - En el modo manual, la salida de control es determinada por el operador, no el controlador PID. Mientras esté en modo manual, el controlador PID supervisa todas las alarmas incluyendo High-high (Alta-Alta), High (alta), Low (baja), Low-Low (Baja-Baja), desvíos amarillo, anaranjado y Tasa de Cambio. Vea las definiciones mas adelante en este capítulo,

Automático - En el modo automático, el controlador PID calcula la salida de control basada en los parámetros programados almacenados en la memoria. Se supervisan todas las alarmas mientras esté en el modo Automático.

Cascada - El modo en cascada es una opción del modo Automático. Si se usa la función de cascada, el lazo de control funcionará como en modo automático a excepción del hecho de que un lazo conectado en cascada tiene un valor de referencia (setpoint) que es la salida de control de otro lazo.

Algoritmos de condiciones especiales

Lazo de acción Reversa

Aunque el algoritmo de PID se utiliza en un controlador de acción directa (o forward), hay veces en es necesaria una salida reversa de control. El controlador PID del PLC DL06 permite que un lazo funcione con acción reversa. Con un lazo de acción reversa, la salida se mueve en la dirección opuesta del error. Por ejemplo, si el valor de referencia SP es más grande que el valor de la variable de proceso PV, un controlador reverso disminuye la salida para aumentar el PV.

$$M_x = -K_i * e_n + M_{x_{n-1}}$$
$$M = -K_c * e_n + K_r(PV_n - PV_{n-1}) + M_{x_n}$$

La ganancia de un proceso determina, en parte, cómo se debe controlar éste. El proceso que tiene una ganancia positiva, es el que llamamos "acción directa". Esto significa que cuando la salida de control aumenta, la variable de proceso también aumenta. Por supuesto, un proceso verdadero es generalmente una función compleja de transferencia que incluye atrasos de tiempo. Aquí, sólo estamos interesados en la dirección de cambio de la variable de proceso como respuesta a un cambio de la salida del control.

La mayoría de los lazos de proceso serán de actuación directa, tal como un lazo de temperatura. Un aumento en el calor aplicado aumenta la PV (temperatura). De acuerdo a esto, los lazos de acción directa se llaman a veces "*lazos de calentamiento*".

Un lazo de "acción inversa" es uno en que el proceso tiene una ganancia negativa. Un aumento en la salida de control tiene como resultado una disminución en la variable PV. Esto se encuentra comúnmente en controles de refrigeración, donde un aumento en la entrada del sistema de enfriamiento causa una disminución en la PV (temperatura). Por consiguiente, los lazos de acción inversa se llaman a veces "*lazos de enfriamiento*".

¡Es importantísimo saber si cierto lazo es de acción directa o inversa! A menos que usted controle la temperatura, no hay una respuesta obvia. En un lazo de control de flujo, un circuito de posición de válvula se puede configurar y puede ser alambrado como acción directa tan fácilmente como acción inversa.

Una manera fácil de averiguar el tipo de acción es hacer funcionar el lazo en modo manual, donde usted debe generar manualmente los valores de la salida de control. Observe si la variable PV sube o baja en respuesta a un aumento en la salida de control.

Raíz cuadrada de la variable de proceso

Se selecciona la raíz cuadrada siempre que la variable de proceso PV sea de un dispositivo tal como una placa orificio que requiera este cálculo.

Control con error al cuadrado

Siempre que se seleccione el control con error al cuadrado, se calcula el error como:

$$e_n = (SP - PV_n) * ABS(SP - PV_n)$$

Un lazo que usa el error al cuadrado responde menos rápidamente que un lazo usando apenas el error normal, sin embargo, él responderá más rápidamente con un error grande. Mientras más pequeño sea el error, es menos rápida la respuesta del lazo. El control con error al cuadrado sería típicamente usado en una aplicación de control de pH.

Control de banda muerta del error

Con control de banda muerta del error, el controlador PID no toma ninguna acción de control si el PV está dentro del área especificada de la banda muerta alrededor del valor de referencia (setpoint). La banda muerta del error es igual sobre y debajo del valor de referencia (setpoint).

Cuando la variable de proceso PV esté fuera de la banda muerta de error alrededor del valor de referencia (setpoint), se usa el valor de error en el cálculo del lazo.

$$e_n = 0 \quad \text{SP - banda muerta_SP} < PV < \text{SP - banda muerta_Above_SP}$$

$$e_n = P - PV_n \quad \text{o de otra forma}$$

El error será elevado al cuadrado primero si se selecciona error al cuadrado y banda muerta de error.

Limitación de la ganancia derivativa

Cuando el coeficiente del término derivativo, K_d , es un valor grande, el ruido introducido en la variable de proceso PV puede dar lugar a un valor errático de salida del lazo de control. Este problema es corregido especificando un límite a la ganancia derivativa que limita el coeficiente K_d . El límite de la ganancia derivativa es un filtro de primer orden aplicado al cálculo derivado del término, Y_n , según lo mostrado abajo.

$$Y_n = Y_{n-1} + \frac{T_s}{T_s + \left(\frac{T_d}{K_d}\right)} * (PV_n - Y_{n-1})$$

Algoritmo de posición

$$M_x = K_i * e_n + M_{x_{n-1}}$$

$$M = K_c * e_n - K_r * (Y_n - Y_{n-1}) + M_x$$

Algoritmo de Velocidad

$$\Delta M = K_c * (e_n - e_{n-1}) + K_i * e_n - K_r * (Y_n - 2 * Y_{n-1} + Y_{n-2})$$

Diez pasos para obtener un buen control de proceso

Los controladores PID en el PLC DL06 proporcionan funciones sofisticadas de control de proceso. En los sistemas automatizados de control puede ser difícil depurar un programa, porque un síntoma dado puede tener muchas causas posibles. Recomendamos un enfoque cuidadoso, paso a paso, para hacer que un nuevo control funcione correctamente:

Paso 1: Sepa la receta

El más importante conocimiento es cómo hacer su producto. Este conocimiento es la base para diseñar un sistema efectivo de control. Una "receta" buena del proceso hará lo siguiente:

- Identifique todas variables pertinentes de proceso, tales como temperatura, presión, o flujos, etc. que necesitan un control preciso.
- Grafique los valores deseados de señal de referencia para cada variable de proceso mientras dure de un ciclo de proceso.

Paso 2: Planee la estrategia de control de lazos cerrados

Esto significa simplemente escoger el método que la máquina usará para mantener el control sobre las variables de proceso para seguir su señal de referencia. Esto implica muchos asuntos y escenarios, tales como eficiencia energética, costos del equipo, la posibilidad de atender a la máquina durante la producción, y otros. Usted debe determinar también cómo generar el valor de referencia durante el proceso y si un operario de la máquina puede cambiar este valor (SP) u otros valores para la operación correcta del lazo de control.

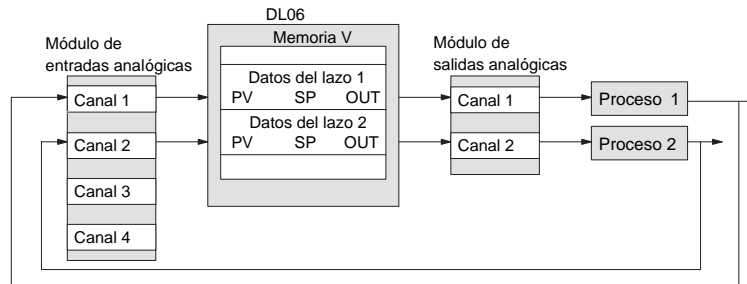
Paso 3: Dimensione correctamente los componentes del lazo

Asumiendo que la estrategia del control es apropiada, también es crucial *dimensionar adecuadamente los actuadores y los sensores*. Escoja un actuador (calentador, bomba, etc.) que sea del tamaño de la carga. Un actuador demasiado grande tendrá un efecto exagerado en su proceso después de un cambio de SP. Sin embargo, un actuador pequeño permitirá que el PV se atrase o se desvíe del SP después de una perturbación o cambio del SP del proceso. Escoja un sensor de PV que cubra el rango de interés (y el control) para el proceso. Decida la resolución del control que usted necesita para el PV (tal como dentro de 2 grados C), y cerciórese que el sensor proporciona el valor de entrada del lazo con por lo menos 5 veces esa resolución (en el nivel de LSB). Sin embargo, un sensor súper sensitivo puede causar oscilaciones de control. El PLC DL06 permite medir valores de señal unipolar y bipolar de 12 bits, de 15 bits y de 16 bits. Esta selección afecta el SP, la PV, la salida de control (CV) y la suma del Integrador.

Paso 4: Escoja los módulos de entradas y salidas

Después de decidir el número de lazos, las variables de proceso PV a medir, y los valores de referencia SP, usted puede escoger los módulos apropiados de entradas y salidas. Vea la figura en la próxima página. En muchos casos, usted será capaz de compartir módulos de entrada o salida, o usar un módulo analógico combinación de entradas y salidas, entre varios lazos del control. El ejemplo mostrado envía la PV y las señales de la salida de control de dos lazos por el mismo conjunto de módulos. Por ejemplo, puede escoger módulos analógicos de entrada con 4 canales por módulo que aceptan señales 0 - 20mA o 4 - 20mA.

Hay también módulos con señal de 0-5 Volt CC o 0-10 VCC, unipolares. O módulos que miden la temperatura directamente de termopares u de detectores de temperatura por resistencia, como los PT100. Vea el catálogo de ventas para mayores informaciones sobre estos módulos, o baje el manual desde nuestro sitio de Internet www.automationdirect.com. Este manual es el artículo D0-OPTIONS-M-SP.



Paso 5: Cableado e instalación

Después que ha hecho la selección y la adquisición de todos componentes del lazo y módulos de entradas y salidas, usted puede realizar el cableado y la instalación. Vea las recomendaciones de cableado en el manual del PLC. Los errores de cableado más comunes cuando se instalan controles de lazos PID son:

- Invertir la polaridad del sensor o conexiones del actuador.
- Las conexiones erradas de la señal entre componentes del lazo.

Paso 6: Los parámetros del lazo

Después que ha hecho el cableado e instalación, escoja los parámetros de configuración del lazo. El método más fácil para programar las tablas de lazo es usar *DirectSOFT*. Este software proporciona cajas de diálogo de configuración PID que simplifican la tarea. NOTA: es importante entender el significado de todos parámetros del lazo mencionados en este capítulo antes de escoger los valores a ser entrados.

Paso 7: Verifique el desempeño del lazo en circuito abierto

Con el cableado de los sensores y actuadores hechos y con los parámetros de lazo entrados, debemos verificar detenidamente y manualmente el sistema nuevo de control (use el modo Manual).

- Verifique que el valor de PV del sensor es correcto.
- Si es seguro hacer así, aumente gradualmente la salida de control encima de 0%, y vea si el PV responde (y se mueve en la dirección correcta!)

Paso 8: Sintonía del lazo

Si la prueba del lazo en circuito abierto muestra que la PV que lee es correcta y que la salida de control tiene el efecto apropiado en el proceso, usted puede proseguir el procedimiento de la sintonía de lazo con circuito cerrado. En este paso, usted afina el lazo de modo que la PV sigue automáticamente al SP.

Paso 9: Haga funcionar el ciclo del proceso

Si la prueba de lazo cerrado muestra que la PV seguirá pequeños cambios en el SP, considere hacer funcionar un ciclo verdadero de proceso. Usted necesitará haber completado la programación que generará la referencia (SP) deseada en tiempo real. En este paso, usted puede querer hacer funcionar una pequeña prueba del producto por la máquina, mirando el cambio de SP según la receta.

ADVERTENCIA: Esté seguro que la Parada de Emergencia esté accesible y que puede cortar la energía sin problemas, en caso de que el proceso se vaya fuera de control. Pueden haber daños al equipo y/o heridas graves al personal si se pierde el control de algunos procesos.



Paso 10: Salve los parámetros

Cuando las pruebas del lazo y sesiones de sintonía estén completas, asegúrese de salvar todos los parámetros de configuración de lazo al disco o incluso en el programa ladder.

Configuración de lazos PID en el PLC DL06

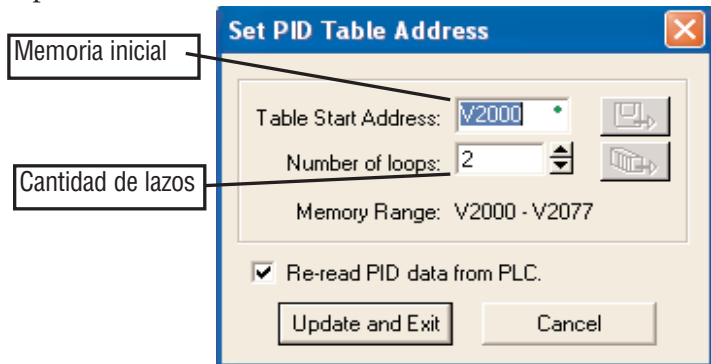
Ud debe conocer lo siguiente antes de comenzar:

Asegúrese de que sus módulos analógicos han sido instalados y estén funcionando antes de comenzar la configuración de lazo PID.

No existe una "instrucción de PID" en la programación ladder en los PLCs DL06, como en otros PLCs. En vez de eso, la CPU lee los parámetros de direcciones reservadas de memoria.

La forma más simple de configurar los parámetros es

con *DirectSOFT*. Para hacer esto, el PLC DL06 debe estar energizado y conectado a la computadora con *DirectSOFT*. Los parámetros pueden ser ingresados solamente cuando el PLC está en el modo Program. Coloque la dirección inicial en el diálogo **Set PID Table Address**, que se obtiene al hacer clic en el menú PLC, luego Setup y por último PID, cuando se abre el PID setup en



DirectSOFT. Este diálogo se ve en la figura adyacente. También, al entrar la cantidad de lazos PID (1 a 8), aparecerá el rango de memoria a ser usado por los lazos PID. Al hacer clic en el botón **Update and Exit**, aparecerán otros diálogos que permiten entrar más parámetros en cada uno de los lazos. Vea más detalles en la página 8-25.

Si trabaja con la lógica ladder, viene a continuación la explicación de las memorias del PLC:

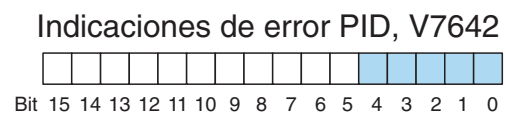
Se debe programar un valor en V7640 para señalar el inicio de la tabla del lazo, como mostrado en la tabla de abajo. Luego se requiere programar V7641 con el número de lazos que quiera que la CPU calcule. V7642 contiene indicaciones de error que se harán verdaderas si V7640 o V7641 se programan impropriamente.

Si el número de lazos es "0", la función del controlador de lazos PID se deshabilita durante el barrido de un programa ladder. El controlador PID le permitirá el uso de lazos en orden creciente, comenzando con 1. Por ejemplo, usted no puede usar el lazo 1 y 4 y saltarse 2 y 3.

Bit	Descripción de error (0 es el estado sin error; 1 es con error) en V7642
0	La dirección inicial (en V7640) está fuera del rango más bajo de memoria del PLC
1	La dirección inicial (en V7640) está fuera del rango más alto de memoria del PLC
2	El número de lazos escogidos (en V7641) es más que 8
3	La tabla de lazos sobrepasa el límite de memoria V de usuario V17777.

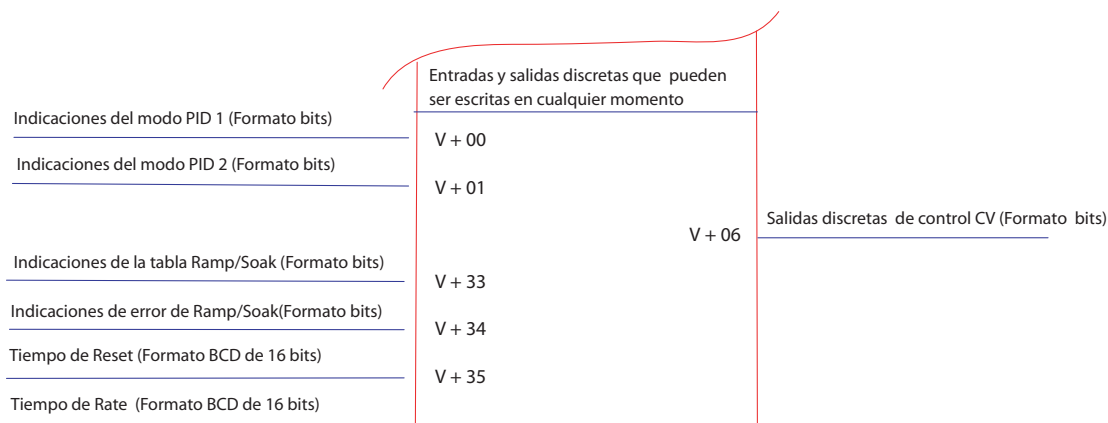
Indicaciones de error de los lazos PID

La CPU informa cualquier error de programación de los parámetros de configuración en V7640 y V7641. Hace esto poniendo los bits apropiados en V7642 en la transición del modo de Programa a RUN.



El algoritmo interno del control PID se puede interpretar como una instrucción con varias entradas y salidas. Los parámetros de configuración se pueden escribir usando otro método tal como usando programa ladder, de modo que la memoria de indicaciones de error puede ser útil en esos casos. Vea los diagramas siguientes para entender el concepto. En las páginas siguientes se explica cada una de las variables:

Memoria inicial de la tabla PID es V + 0	
	Entradas y salidas analógicas que pueden ser escritas en cualquier momento
Referencia SP (Formato binario de 16 bits)	V + 02
Variable de proceso PV (Formato binario de 16 bits)	V + 03
Bias (Formato binario de 16 bits)	V + 04
	V + 05
	Salida de control CV (Formato binario de 16 bits)
Período de muestreo (Formato BCD de 16 bits)	V + 07
Ganancia proporcional (Formato BCD de 16 bits)	V + 10
Tiempo de Reset (Formato BCD de 16 bits)	V + 11
Tiempo de Rate (Formato BCD de 16 bits)	V + 12
Ajuste de banda muerta (Formato binario de 16 bits)	V + 23
Límite inferior de SP (Formato binario de 16 bits)	V + 26
Límite superior de SP (Formato binario de 16 bits)	V + 27
Puntero del SP remoto (Formato binario de 16 bits)	V + 32
	Entradas y salidas analógicas que pueden ser escritas solamente en program mode
Alarma L-L de PV (Formato binario de 16 bits)	V + 13
Alarma L de PV (Formato binario de 16 bits)	V + 14
Alarma H de PV (Formato binario de 16 bits)	V + 15
Alarma H-H de PV (Formato binario de 16 bits)	V + 16
Alarma amarilla de desvío (Formato binario de 16 bits)	V + 17
Alarma roja de desvío (Formato binario de 16 bits)	V + 20
Alarma de tasa de cambio (Formato binario de 16 bits)	V + 21
Ajuste de histéresis (Formato binario de 16 bits)	V + 22
Ajuste de limite derivativo (Formato BCD de 16 bits)	V + 25
Límite inferior de CV (Formato binario de 16 bits)	V + 30
Límite superior de CV (Formato binario de 16 bits)	V + 31
Dirección de tabla R/S (Formato binario de 16 bits)	v + 34



8

Observe entonces que el algoritmo PID tiene asociadas direcciones de memoria que pueden ser conectadas a los parámetros del algoritmo PID.

Dirección	Parametro de configuración	Tipo de datos	Rangos	Lee/Escribe
V7640	Puntero de la tabla de parámetros de lazos	Octal	V1200 V7340 V10000-V17740	escribe
V7641	Cantidad de lazos	BCD	0 – 8	escribe
V7642	Indicación de error	Binario	0 o 1	lee

Si usted usa el diálogo de configuración de lazos de *DirectSOFT*, la verificación automática de rangos impide posibles errores de configuración. Sin embargo, los parámetros de configuración se pueden escribir usando otros métodos tales como usando programa ladder, de modo que la memoria de indicaciones de error puede ser útil en esos casos. La tabla siguiente lista los errores informados en V7642.

Bit	Descripción de error (0 es el estado sin error; 1 es con error)
0	La dirección inicial (en V7640) está fuera del rango más bajo de memoria del PLC
1	La dirección inicial (en V7640) está fuera del rango más alto de memoria del PLC
2	El número de lazos escogidos (en V7641) es más que 8
3	La tabla de lazos sobrepasa la frontera en V7577. Use una dirección más cerca a V1200.

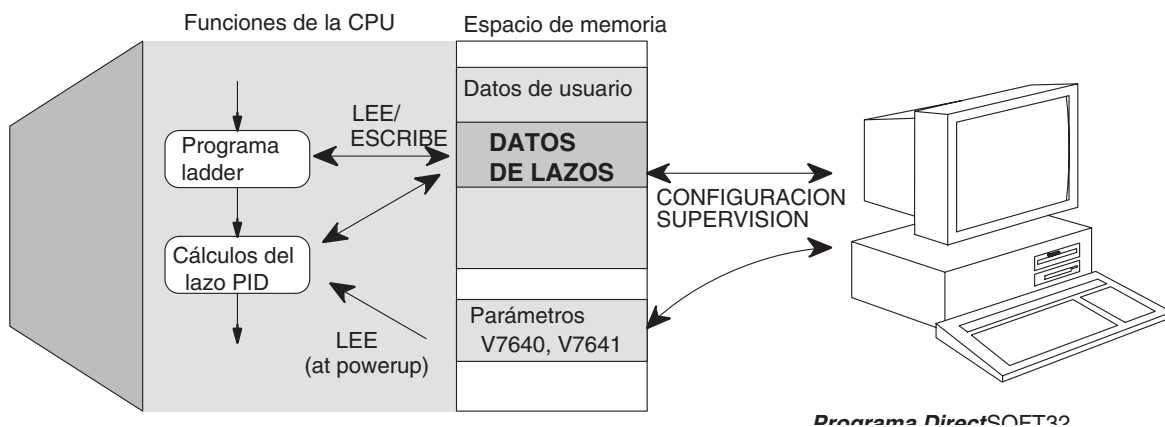
En una transición del modo Program a Run, la CPU lee los parámetros. En ese momento, la CPU sabe cual es la localización de la tabla PID y cuántos lazos de control se han configurado. Luego, durante el barrido del programa ladder, el algoritmo PID usa los valores del lazo para ejecutar los cálculos que determinan la salida de control, para generar alarmas, etc.

Hay algunos parámetros de la tabla que pueden ser leídos o escritos en cada cálculo de la salida del lazo de control PID.

Estableciendo el tamaño de la tabla de lazos y localización

En una transición del modo Program a Run, la CPU lee los parámetros. En ese momento, la CPU sabe cual es la localización de la tabla PID y cuántos lazos de control se han configurado. Luego, durante el barrido del programa ladder, el algoritmo PID usa los valores del lazo para ejecutar cálculos, generar alarmas, etc. Hay algunos parámetros de la tabla que pueden ser leídos o escritos en cada cálculo del lazo.

La tabla de parámetros de lazos contiene los datos para tantos lazos como se haya definido en V7641.



Cada configuración de un lazo ocupa 32 palabras (0 a 37 octal) en la tabla de lazos.

Por ejemplo, supongamos que usted tiene una aplicación con 4 lazos, y usted escoge V2000 como dirección inicial. Los parámetros de lazos ocuparán V2000 - V2037 para el lazo 1, V2040 - V2077 para el lazo 2, etcétera. El lazo 4 ocupa V2140 - V2177.

Determine el bloque de memoria V a ser usado por cada lazo PID. Además de ser el inicio del bloque de memorias V consecutivas del lazo PID, la primera dirección será en inicio de los parámetros del lazo 1. recuerde que hay 32 palabras (0 a 37 octal) usadas en cada lazo. Una vez que se haya determinado la dirección inicial, entoces se puede configurar y almacenar los parámetros del lazo PID ya sea directamente en su programa ladder o usando la utilidad de configuración de PID (Setup PID) en *DirectSOFT*.

Memoria V	Datos del usuario
↑ V2000	LAZO #1
↓ V2037	32 palabras
↑ V2040	LAZO #2
↓ V2077	32 palabras
•	LAZO #3
•	32 palabras
•	LAZO #4
	32 palabras

Definiciones de palabras de la tabla de lazos

Los parámetros asociados con cada lazo se listan en la tabla de abajo. El desvío de la dirección está en octal, para ayudarlo a localizar parámetros específicos en una tabla de lazos. Por ejemplo, si una tabla comienza en V2000, entonces la dirección del término RESET(integral) es **Addr + 11**, o V2011. No use el número de la palabra (en la primera columna) para calcular las direcciones.

Palabra #	Dirección y desvío	Descripción	Formato	Lee "durante la operación"
1	Dirección + 0	Configuración 1 del lazo PID	bits	Si
2	Dirección + 1	Configuración 2 del lazo PID	bits	Si
3	Dirección + 2	Valor de señal de referencia (SP)	palabra/binario	Si
4	Dirección + 3	Variable de proceso (PV)	palabra/binario	Si
5	Dirección + 4	Valor de bias (integral)	palabra/binario	Si
6	Dirección + 5	Valor de salida de control	palabra/binario	Si
7	Dirección + 6	Estado de Modo y de alarmas	bits	-
8	Dirección + 7	Valor del período de muestreo	palabra/BCD	Si
9	Dirección + 10	Valor de ganancia	palabra/BCD	Si
10	Dirección + 11	Valor del tiempo de RESET	palabra/BCD	Si
11	Dirección + 12	Valor de tasa (derivativo)	palabra/BCD	Si
12	Dirección + 13	Alarma Low-Low de PV	palabra/binario	No*
13	Dirección + 14	Alarma Low de PV	palabra/binario	No*
14	Dirección + 15	Alarma High de PV	palabra/binario	No*
15	Dirección + 16	Alarma High-High de PV	palabra/binario	No*
16	Dirección+ 17	Alarma de desvío de PV (amarilla)	palabra/binario	No*
17	Dirección + 20	Alarma de desvío de PV (roja)	palabra/binario	No*
18	Dirección + 21	Alarma de tasa de cambio de PV	palabra/binario	No*
19	Dirección + 22	Valor de alarma de histéresis	palabra/binario	No*
20	Dirección+ 23	Valor de error de banda muerta, PV	palabra/binario	Si
21	Dirección+ 24	Constante del filtro pasa bajo de PV	palabra/BCD	Si
22	Dirección + 25	Factor límite de ganancia derivativa	palabra/BCD	No**
23	Dirección + 26	Límite inferior del valor SP	palabra/binario	Si
24	Dirección + 27	Límite superior del valor SP	palabra/binario	Si
25	Dirección + 30	Límite inferior de salida de control	palabra/binario	No**
26	Dirección + 31	Límite superior de salida de control	palabra/binario	No**
27	Dirección + 32	Puntero del valor remoto de SP	palabra/hex	Si
28	Dirección + 33	Indicación de ramp/Soak	bit	Si
29	Dirección + 34	Dirección inicial de tabla Ramp/Soak	palabra/hex	No**
30	Dirección + 35	Indicaciones de error de Ramp/Soak	bits	No**
31	Dirección + 36	Transferencia automática	palabra/hex	Si
32	Dirección + 37	# de Canal, transferencia automática de la salida de control	palabra/hex	Si

* El algoritmo lee datos solamente al algoritmo cuando el bit de alarma hace la transición de 0 a 1

** El algoritmo lee datos solamente en un cambio de modo del PLC

Descripciones de bits de la palabra configuración de modo PID 1 (Addr + 00)

Las definiciones individuales de bits de la palabra 1 (Addr+00) están listados en la tabla siguiente. Está disponible información adicional acerca del uso de esta palabra posteriormente en este capítulo.

Bit	Descripción	Leer/ escribir	Bit=0	Bit=1
0	Pedido de Operación en modo Manual	Escribe	–	Transición de OFF para ON
1	Pedido de Operación de modo Automático	Escribe	–	Transición de OFF para ON
2	Pedido de Operación modo en Cascada	Escribe	–	Transición de OFF para ON
3	Selección de Transferencia Sin saltos	Escribe	Mode I	Modo II
4	Selección de modo de acción Directo o Invertida	Escribe	Directa	Invertida
5	Selección de algoritmo de Posición o Velocidad	Escribe	Posición	Velocidad
6	Selección de PV lineal o raíz cuadrada	Escribe	Lineal	Raíz cuadrada
7	Selección de error lineal o cuadrado	Escribe	Lineal	Cuadrado
8	Selección de habilitar error de banda muerta	Escribe	Deshabilite	Habilite
9	Selección de límite de ganancia derivativa	Escribe	No	Si
10	Selección de "Congelar tendencia"	Escribe	No	Si
11	Selección de operación Ramp/Soak	Escribe	No	Si
12	Selección de supervisión de alarma de PV	Escribe	No	Si
13	Selección de alarma de desvío de PV	Escribe	No	Si
14	Selección de alarma de tasa de cambio del PV	Escribe	No	Si
15	PID trabaja independientemente de la CPU	Escribe	En modo RUN	Independiente del modo de la CPU

Descripciones de bits de la palabra de configuración PID 2 (Addr + 01)

Las definiciones individuales de bits de la palabra 2 (Addr+01) están listados en la tabla siguiente. Hay información adicional disponible acerca del uso de esta palabra posteriormente en este capítulo.

Bit	Descripción	Leer/ escribir	Bit=0	Bit=1
0	Selección de rango de PV y CV (Vea notas 1 y 2)	Escribe	Unipolar	Bipolar
1	Selección de formato de datos (vea notas 1 y 2)	Escribe	12 bits	15 bits
2	Filtro analógico de entrada y transferencia automática	Escribe	OFF	ON
3	Selección de Habilitación de límites de SP	Escribe	Habilite	Deshabilite
4	Selección de unidades de Reset (Integral gain)	Escribe	Segundos	Minutos
5	Selección de algoritmo de sintonía PID automática	Escribe	Lazo cerrado	Lazo abierto
6	Selección de sintonía automática	Escribe	PID	Sólo en PI (rate = 0)
7	Iniciar sintonía automática	Lea/Escribe	Sintonía hecha	Fuerce inicio
8	Tiempo de barrido del PID (uso interno)	Lea	–	–
9	Selección de formato de 16 bits (vea notas 1 y 2)	Escribe	No 16 bits	Use 16 bits
10	Selección de datos separados para entradas y salidas	Escribe	Mismo formato	Diferente
11	Selección de rango CV uni-0 bi-polar	Escribe	Unipolar	Bipolar
12	Selección de formato de datos CV (Vea notas 2,3)	Escribe	12 bits	15 bits
13	Selección de formato de salida de 16 bits (vea 2,3)	Escribe	No 16 bits	Use 16 bits
14-15	Reservado para uso futuro	–	–	–



Nota 1: Si el valor en el bit 9 es 0, entonces los valores en los bits 0 y 1 son leídos. Si el valor en el bit 9 es 1, entonces los valores en los bits 0 y 1 no son leídos, y el bit 9 define el formato de datos (el rango es automáticamente unipolar).



Nota 2: Si el valor en el bit 10 es 0, entonces los valores en los bits 0, 1, y 9 definen los rangos de entradas y salidas y formatos de datos (los valores en los bits 11, 12 y 13 no son leídos). Si el valor en el bit 10 es 1, entonces los valores en los bits 0, 1, y 9 definen sólo el rango de entrada y formato de datos, y los bits 11, 12, y 13 son leídos y definen el rango de salidas y formato de datos.



Nota 3: Si el bit 10 tiene un valor de 1 y el bit 13 tiene un valor de 0, entonces el bit 11 y 12 son leídos y definen el rango de salidas y formato de datos. Si el bit 10 y el bit 13 cada uno tiene un valor de 1, entonces los bits 11 y 12 no son leídos, y el bit 13 define el formato de datos (el rango de salida es automáticamente unipolar).

Palabra de supervisión de modo/alarmas (Addr + 06)

En la tabla siguiente se describen las definiciones individuales de bits de la palabra de supervisión de modo/alarmas (Addr+06). Hay más detalles en la sección del modo de PID y la sección de alarmas.

Bit	Descripción del bit de Modo / Alarma	Lee/escibe	Bit=0	Bit=1
0	Indicación de modo Manual	Lee	–	Manual
1	Indicación de modo Automatico	Lee	–	Automatico
2	Indicación de modo Cascada	Lee	–	Cascada
3	Alarma Low-Low de la PV	Lee	OFF	ON
4	Alarma Low de la PV	Lee	OFF	ON
5	Alarma High de la PV	Lee	OFF	ON
6	Alarma High-High de la PV	Lee	OFF	ON
7	Alarma de desvío YELLOW de PV	Lee	OFF	ON
8	Alarma de desvío RED de PV	Lee	OFF	ON
9	Alarma de tasa de cambio de PV	Lee	OFF	ON
10	Alarma de error de valor de programación	Lee	–	Error
11	Cálculo de overflow/underflow	Lee	–	Error
12	Lazo está en sintonía automática	Lee	OFF	ON
13	Iniciación de error de sintonía automática	Lee	–	Error
14–15	Reservado para uso futuro	–	–	–

Tabla de indicaciones de Ramp/Soak (Rampa y valor constante) (Addr + 33)

En la tabla siguiente se describen las definiciones individuales de los bits de indicación de la tabla Ramp/Soak (Addr+33). Se describen detalles adicionales en la sección de la Operación Ramp/Soak.

Bit	Descripción de bit de indicación de Ramp/Soak	Lee/escibe	Bit=0	Bit=1
0	Iniciar el perfil Ramp/Soak	Escribe	–	Parte en la transición 0 - 1
1	Mantenga los valores del perfil Ramp/Soak	Escribe	–	Mantiene en transición 0 -1
2	Reanuda el perfil Ramp/Soak	Escribe	–	Reanuda en transición 0-1
3	Jog del perfil Ramp/Soak	Escribe	–	Jog en transición 0-1
4	Perfil Ramp/Soak completo	Lee	–	Completado
5	Desvío de la variable PV en Ramp/Soak	Lee	OFF	ON
6	Valores de Perfil de Ramp/Soak mantenidos	Lee	OFF	ON
7	Reservado	Lee	–	–
8–15	Etapa corriente en el perfil Ramp/Soak	Lee	Decodifique como byte (hexadecimal)	

Los bits 8-15 deben ser leídos como un byte para indicar el número actual del segmento del generador de la Ramp/Soak en el perfil. Este byte tendrá los valores 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, y 10, que representa los segmentos 1 a 16 respectivamente. Si el byte = 0, entonces la tabla Ramp/Soak no está activada.

Localización de la tabla Ramp/Soak (Rampa y valor constante) (Addr + 34)

Cada lazo que usted configura tiene la opción de usar el generador de Ramp/Soak dedicado a ese lazo. Esta característica genera los valores de referencia que siguen un perfil en el tiempo. Para usar la característica Ramp/Soak, usted debe programar una tabla separada de 32 palabras con valores apropiados. Una caja de diálogo *DirectSOFT* hace que esta tabla sea muy fácil de hacer.

En la tabla de lazos, el puntero de la tabla Ramp/Soak en Addr + 34 debe indicar el comienzo de los datos de la Ramp/Soak para ese lazo. Este puede estar en cualquier lugar en la memoria de usuario, y no tiene que estar contiguo a la tabla de Parámetros del lazo, como se muestra a la izquierda. Cada tabla de Ramp/Soak requiere 32 palabras, sin importar el número de segmentos programados.

Los parámetros de tabla de Ramp/Soak se definen en la tabla de abajo. Hay más detalles adicionales en la sección de la Operación de la Ramp/Soak, en la página 8-60.



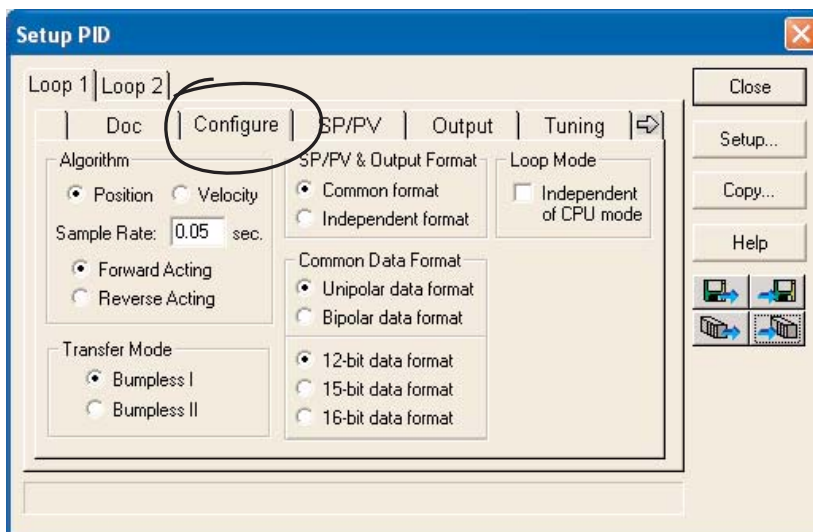
Indicaciones de error de programación en tabla Ramp/Soak (Addr + 35)

Las definiciones individuales del bit de indicaciones de Error de Programación de Tabla de Ramp/Soak (Rampa y valor constante) en la palabra (Addr+35) son listadas en la tabla siguiente. Detalles adicionales son mostrados en la sección de modo de lazos de PID y en la sección de alarmas de la variable de proceso PV luego en este capítulo.

Bit	Descripción de indicación de error Ramp/Soak	Lee/escribe	Bit=0	Bit=1
0	Dirección inicial del rango inferior de memorias V	Lee	-	Error
1	Dirección inicial del rango superior de memorias V	Lee	-	Error
2-3	Reservado para uso futuro	-	-	-
4	Dirección inicial del rango de parámetros de sistema	Lee	-	Error
5-15	Reservado para uso futuro	-	-	-

Configure el lazo de control PID

Cuando se haya definido la tabla PID en la memoria V, se continúa con la configuración del lazo PID con ayuda del diálogo de *DirectSOFT* mostrado en la figura adyacente. Ud debe llenar los datos requeridos para hacer funcional el lazo PID. Seleccione la lengüeta **Configure** y entonces aparecerá el siguiente diálogo:



Seleccione el tipo de algoritmo. Puede escoger entre *Position* o *Velocity*. El valor por defecto es *Position*. Esta es la opción para la mayoría de las aplicaciones que incluye calentamiento y enfriamiento así como también control de nivel. Un modo de *Velocity* consiste en una variable tal como un totalizador de flujo en un lazo de control de flujo.

8

Entre el período de muestreo

Las tareas principales de la CPU se pueden considerar dentro de categorías como las mostradas en la figura adyacente.

La lista representa las tareas cuando la CPU está en modo RUN, en cada barrido de la CPU. Note que los cálculos de los lazos PID ocurren después de las tareas lógicas del programa ladder

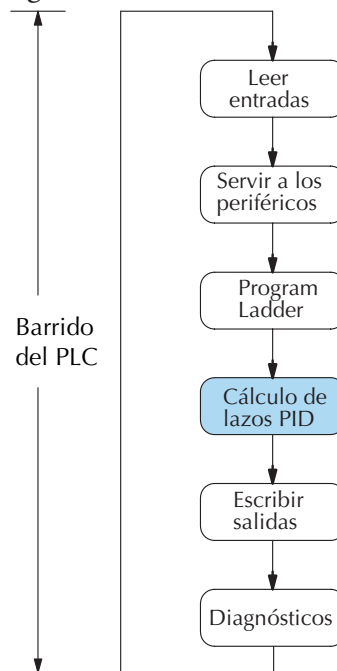
El período de muestreo (sample rate) de un lazo de control es simplemente el período de cálculo del lazo PID. Cada cálculo genera un valor nuevo de la salida de control.

Con el PLC DL06, usted puede definir el período de muestreo de un lazo entre 50 ms a 99,99 segundos. La mayoría de los lazos no requieren un cálculo de PID en cada barrido del PLC. Algunos lazos necesitan ser calculados sólo una vez en 1000 barridos.

Usted escoge el período deseado de muestreo para cada lazo, y la CPU automáticamente ejecuta los cálculos de PID en el barrido apropiado.

No hay un período perfecto de muestreo a usar para un lazo particular de control. Un buen período de muestreo es una configuración que satisface simultáneamente varias condiciones:

- La frecuencia deseada de muestreo es proporcional al tiempo de respuesta de la PV a un cambio en la salida de control. Generalmente, un proceso con una masa grande tendrá una frecuencia lenta de muestreo y una masa pequeña necesita una frecuencia más rápida de muestreo.



- Frecuencias más rápidas de muestreo proporcionan una salida más suave de control y desempeño de PV más exacto, pero usa más tiempo de procesamiento de CPU. El muestreo mucho más rápido que el necesario sólo sirve para malgastar la capacidad de procesamiento de la CPU.
- Frecuencias más lentas de muestreo proporcionan una salida más áspera del control y un desempeño menos exacto de PV, pero usa menos tiempo de procesamiento de CPU.
- Una frecuencia de muestreo que es demasiado lenta causará inestabilidad del sistema, particularmente cuando ocurre un cambio en la referencia (SP) o hay una perturbación.

Como punto de partida, determine un período de muestreo para su lazo que sea rápido suficiente para evitar la inestabilidad del control (que es extremadamente importante).

Selección Forward/Reverse

Es importante saber en qué dirección responderá la salida de control al error (SP-PV), o *directo* o *reverso*. Un lazo de control forward(directo) significa que siempre que la salida de control aumente, la variable de proceso también aumentará. La salida de control de la mayoría de los lazos PID es de acción directa, tal como un lazo de control de calefacción. Un aumento en el calor aplicado aumentará el valor de la variable de proceso PV (temperatura).

Un lazo de control reverso es uno donde un aumento en la salida de control resulta en una disminución del valor de la variable de proceso PV. Un ejemplo común de esto sería un sistema de refrigeración, donde un aumento en la señal de entrada al sistema de enfriamiento causa una disminución de la variable de proceso PV (temperatura).

Transferencia de Modos

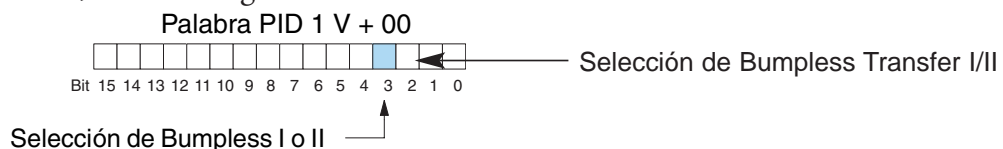
Se puede seleccionar una transferencia Bumpless I o Bumpless II para proporcionar una transición suave de la salida de control desde el modo Manual al modo Auto. Bumpless I hará la referencia SP igual a la variable de proceso PV cuando la salida de control se cambia del manual a automática. Si ésto no se desea, escoja Bumpless II.

Los tipos de transferencia Bumpless I e II se enumeran en la tabla de abajo.

Tipo de transferencia	Bit 3 de transferencia	Algoritmo PID	Acción de transferencia Manual a Auto	Transferencia Auto a Cascada
Bumpless Transfer I Transferencia sin saltos I	0	Posición	Fuerza Bias = Salida de Control Fuerza SP = PV	Fuerza la salida del lazo mayor = PV del lazo menor
		Velocity	Fuerza SP = PV	Fuerza la salida del lazo mayor = PV del lazo menor
Bumpless Transfer II Transferencia sin saltos II	1	Posición	Fuerza Bias = Salida de Control	ninguna
		Velocity	ninguna	ninguna

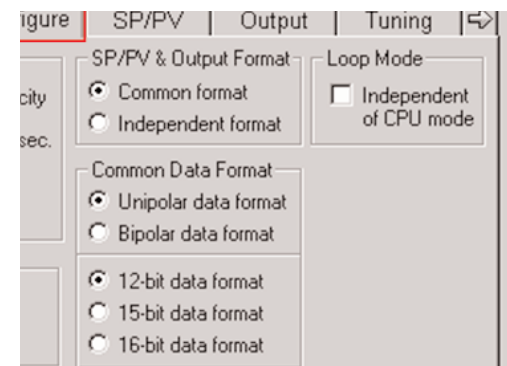
Observe que la operación también depende de qué algoritmo PID está usando, posición o velocidad de la ecuación de PID. Observe que debe usar el tipo I de la transferencia de Bumpless al usar la forma de velocidad del algoritmo de PID.

El tipo de transferencia puede también ser seleccionado en un programa ladder con el bit 3 de la palabra PID 1, V+00 configurada como se muestra:



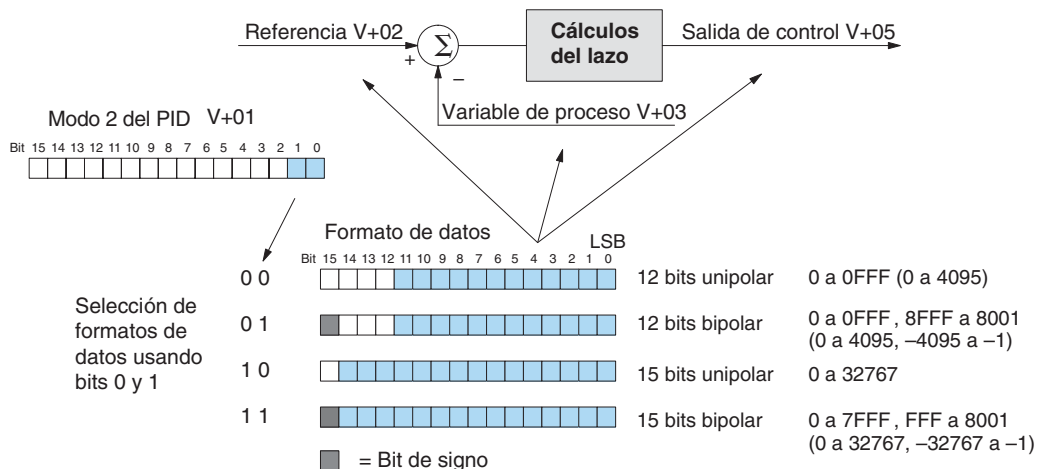
Formato de SP/PV y salida CV

Esta parte le permite seleccionar entre *Common format* o *Independent format*. Common format es por defecto y comunmente usado. Con este formato, las entradas y salidas al controlador PID tendrán el mismo formato de datos, o sea, la misma cantidad de bits, y puede ser uni- o bipolar. Si se selecciona formato independiente, las selecciones estarán grisáceas. En ese caso éstos se seleccionan en los diálogos SP/PV y Output.



Formato de datos común

Seleccione *Unipolar data format* (Sólo datos positivos en 12 bits (0 a 4095) o 15 bits (0 a 32767) o *Bipolar data* que es positivo o negativo (-4095 hasta 4095 o -32767 a 32767) y requiere un bit de signo. La selección bipolar no está disponible cuando se selecciona formato de 16 bits.



El formato de datos determina la interface numérica entre el lazo PID, el sensor de la variable de proceso y el aparato de la salida de control.

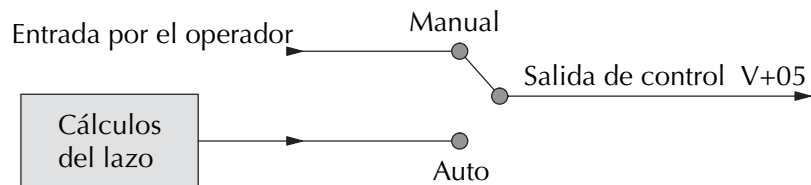
Modos de operación del controlador PID

Esta es una característica especial que permite al controlador ejecutar control de lazo cerrado mientras la CPU está en el modo Program. Debe tenerse cuidado para seleccionar el modo *Independent of CPU mode* en el diálogo.

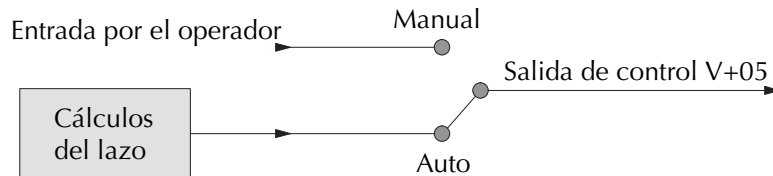
Antes de continuar con la configuración del controlador PID, es necesario definir los tres modos de control.

El PLC DL06 permite usar modos de control estándares: Manual, Automático y Cascada. Los orígenes de las 3 variables básicas, referencia SP, variable de proceso PV y salida de control CV son diferentes para cada modo.

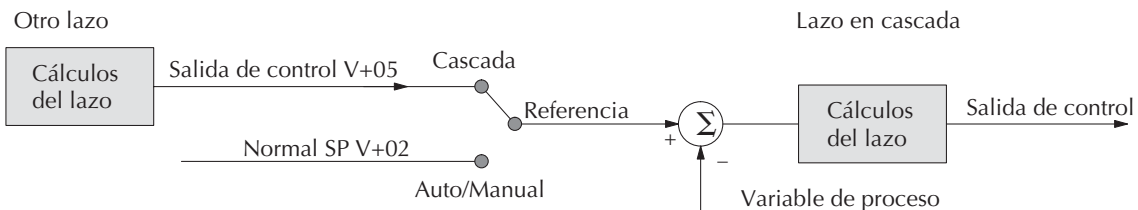
En modo **manual**, el lazo de control no ejecuta el cálculo de la salida (sin embargo, las funciones de alarma están en operación). La CPU deja de escribir valores a la memoria V + 05 (salida de control) para el lazo en cuestión. *Se espera que un operador u otro origen inteligente* controle la salida observando la variable de proceso PV y escriba datos a la salida de control V + 05 como sea necesario para mantener el proceso en control. La figura a continuación muestra el diagrama de flujo de la operación en modo manual.



En el modo **Automático**, el lazo opera normalmente y genera los valores nuevos de la salida de control. Calcula la ecuación de PID y escribe el resultado del cálculo en la dirección **addr + 05** en cada período de muestreo de aquel lazo. Se muestra abajo el diagrama equivalente.



En el modo en **cascada**, el lazo opera como si fuera el modo Automático, con una diferencia importante. El origen de la referencia SP cambia de su dirección normal **addr + 02**, usando el valor de la salida de control de otro lazo. En los modos Manual o Automático, el cálculo del lazo usa los datos en **addr + 02**. En el modo en cascada, el cálculo del lazo lee la salida de control de otra tabla de parámetros de lazos.



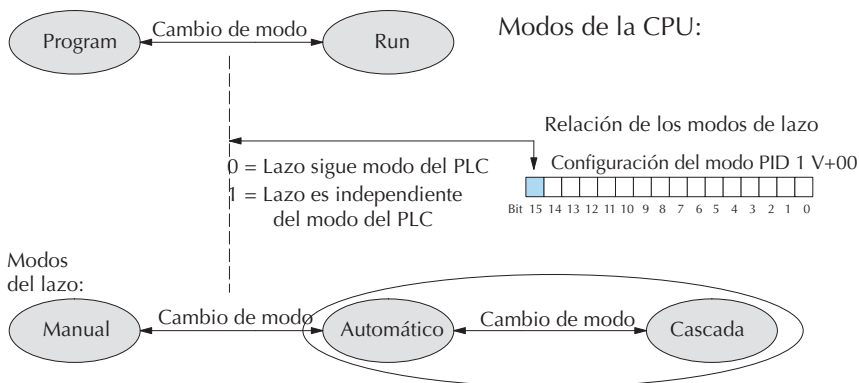
Un lazo se puede cambiar de un modo a otro, pero **no puede ir del modo Manual directamente a Cascada, ni viceversa**. Este cambio del modo no es posible porque un lazo estaría cambiando dos fuentes de datos al mismo tiempo, y podría causar una pérdida de control.



Una vez que la CPU esté funcionando en el modo RUN, la operación normal del controlador PID es leer los datos del lazo y realizar cálculos en cada ejecución del muestreo. Cuando la CPU se coloca en el modo Program, el programa ladder para la operación y todos los lazos de PID se ponen automáticamente en el modo Manual. Los parámetros de PID pueden entonces ser cambiados si es deseado. De la misma forma, colocando la CPU en el modo RUN, los lazos de PID se vuelven al modo operacional que había previamente, es decir, o Manual, Automático o Cascada. Con esta selección usted afecta automáticamente los modos cambiando el modo de la CPU.

Si el bit 15 se coloca como uno, entonces los lazos funcionarán independiente del modo de la CPU. Es como tener dos procesadores independientes en la CPU... uno está funcionando con el programa ladder y el otro está funcionando con los lazos de proceso.

Al tener la capacidad de hacer funcionar lazos independientemente del programa ladder, es factible realizar un cambio de la lógica mientras que el proceso todavía está funcionando.



Esto es especialmente beneficioso para los procesos continuos de gran escala que son difíciles o costosos de interrumpir. *Funcionamiento independiente de la CPU* es la característica usada para esto.

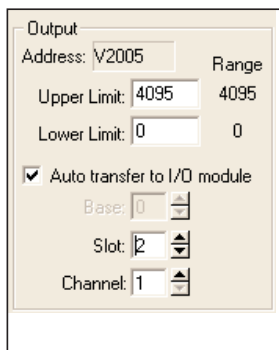
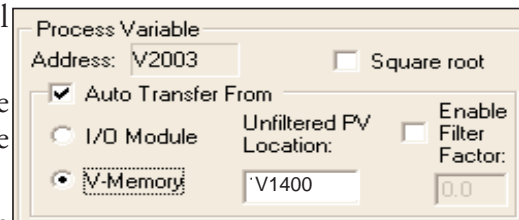
Si usted necesita hacer funcionar el PID mientras detenga el programa ladder, en modo Program, seleccione el modo independiente de la CPU en el diálogo de configuración o modifique su programa para colocar ON y OFF en el bit 15 de la palabra 1 de PID (V+00) en su programa. Si el bit es OFF, el lazo seguirá el modo de la CPU, después cuando la CPU se coloca en el modo Program, todos los lazos será forzados al modo Manual.

Cuando se usa el modo independiente de la CPU, usted debe también configurar el PV para leer directamente desde un módulo de entrada analógica. Esto se puede hacer fácilmente en el diálogo de configuración de PID, SP/PV.

El diálogo SP/PV tiene un bloque titulado *Process Variable*. Hay un bloque dentro de este bloque llamado *Auto Transfer From* (desde una entrada analógica) con la información grisácea. Haciendo clic en la caja a la izquierda de **Auto Transfer From** va a destacar la información. Seleccione *I/O Module* y luego entre el número de la ranura en que reside el módulo de entrada. Luego, seleccione el canal de entrada analógica a ser usado.

La segunda opción es *V-Memory*. Cuando se selecciona esto, se debe especificar la dirección de memoria V de donde es transferido el PV.

Con cualquier método de transferencia automática que se use, se recomienda hacer clic en *Enable Filter Factor* (un filtro pasa-bajo) y especificar el coeficiente de filtro.

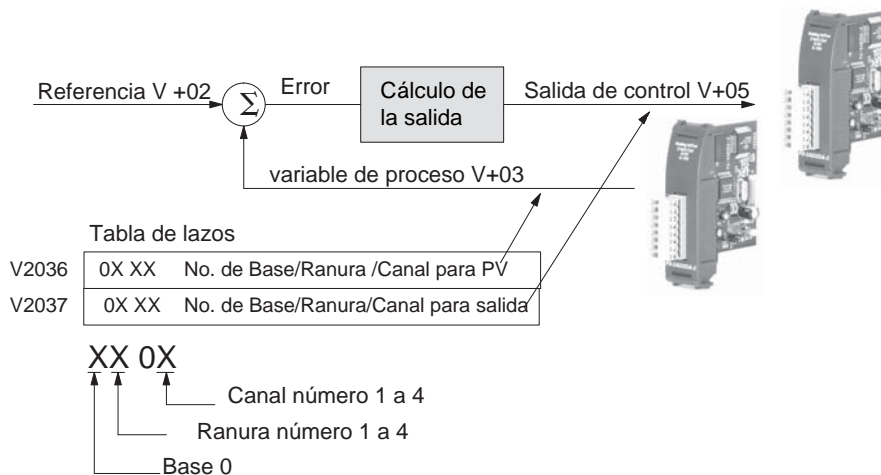


Usted debe también seleccionar la salida analógica para la salida de control a ser transferida. Esto se hace en el diálogo *Output* mostrado en la figura adyacente. El bloque de la información en este diálogo es "grisáceo" hasta que se haga clic en *Auto transfer to I/O module*. Una vez que esté comprobado, entre el número de la ranura donde está residiendo el módulo de salida analógica y después entre el número del canal de la salida analógica.



NOTA: Para hacer cambios en algunos parámetros del lazo PID, éste debe estar en modo manual y el PLC debe ir a modo Program. Si usted ha seleccionado funcionamiento independiente del modo de la CPU, entonces debe tomar ciertas medidas para permitir realizar cambios de parámetros. Usted puede hacer temporalmente que los lazos sigan el modo de la CPU cambiando el bit 15 a 0. Luego usted podrá poner el lazo en modo manual usando **DirectSOFT**. Después de que usted cambie los parámetros del lazo, coloque nuevamente el bit 15 a un valor de 1 para reestablecer el funcionamiento independiente de la CPU del PID.

Usted puede configurar opcionalmente cada lazo para tener acceso a sus valores analógicos (PV y salida de control) poniendo valores apropiados en los registros asociados de la tabla del lazo en su programa ladder. La figura siguiente muestra los parámetros de la tabla del lazo en V+36 y V+37 y el rol de la transferencia automática para tener acceso a los valores analógicos directamente.



Cuando estos parámetros se programan directamente en ladder, un valor de "0102" en el registro V + 36 le ordena al controlador PID a leer los datos de la PV del canal 2 de la entrada analógica. Un valor de "0000" en el registro le dice al controlador que no tenga acceso al valor analógico correspondiente directamente. En ese caso, se debe ejecutar lógica ladder para transferir el valor entre la entrada analógica y la tabla PID.



NOTA: Cuando se utiliza la transferencia automática de E/S, no se tiene acceso a los datos analógicos para todos los canales en el módulo analógico por ningún otro método, es decir, puntero o múltiplex.

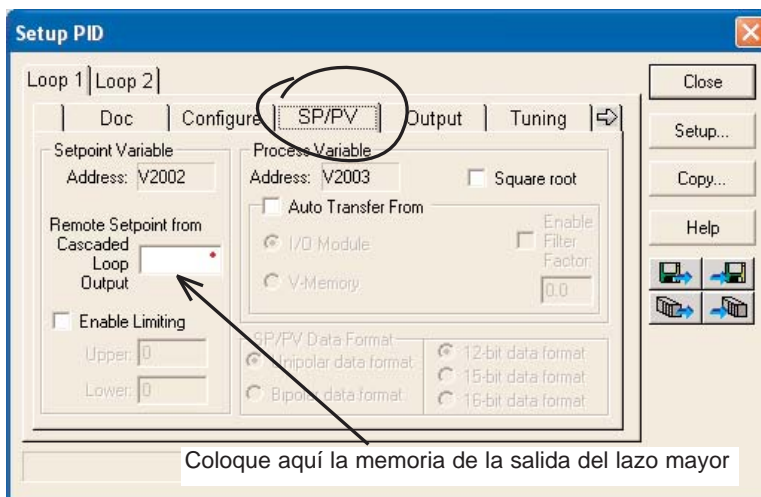
Direcciones de memoria de SP/PV

Hay disponible un diálogo de SP/PV para configurar cómo será usado el valor de referencia (setpoint SP) y la variable de proceso (PV) en el lazo. Si este lazo es el lazo menor de un par conectado en cascada, entre la dirección en la salida de control en el valor de referencia SP, en el campo *Remote SP from Cascaded Loop Output* (Vea la figura que sigue).

Es a veces deseable limitar el rango de los valores de la referencia (setpoint) permitidos. Para activar esta limitación, haga clic en la caja al lado de *Enable Limiting*.

Esto activará los campos superiores (*Upper*) y más bajos (*Lower*) para que los valores sean entrados.

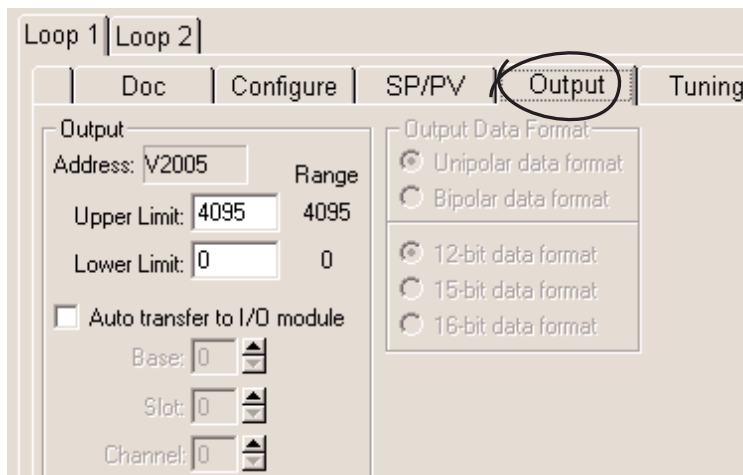
Coloque límites en el valor de referencia (SP) para evitar que un operador entre un valor fuera de un rango seguro. La caja *Square root* box se usa solamente para ciertos lazos de PID, tales como control de flujo. *Auto transfer from I/O module* puede ser usada si se ha aceptado usar **Independent of CPU** en el diálogo **Configure**. Si se selecciona *Auto transfer from I/O module* se puede usar un filtro pasa bajo de primer orden aceptando la caja *Enable Filter*. Se recomienda usar este filtro durante la sintonización de lazo cerrado (Closed loop) Si se escogió formato independiente previamente, haga las selecciones del formato de datos aquí.



NOTA: El diálogo SP/PV se puede dejar como aparece para la operación básica de PID.

Límites de la salida de control

Otro diálogo que estará disponible en la configuración PID será el diálogo OUPUT. La dirección de la salida de control, V+05, (determinada por la dirección del principio de la tabla PID) será mostrado allí. Entre los límites del rango de salida, el límite superior (*Upper Limit*) y el límite inferior (*Lower Limit*), que satisfagan el proceso y que estén de acuerdo con el formato de datos se ha seleccionado. Para una operación básica de PID usando un módulo de salida de 12 bits, configure el límite superior a 4095 y deje el sistema con un límite inferior de 0. Haga clic en la caja *Auto transfer to I/O module* si fuera necesario enviar la salida de control a cierto módulo de salida analógica. Si se usa esta función, se deben usar todos los canales del módulo para las salidas de control PID. Si el se ha elegido formato independiente previamente, el formato de datos de salida necesitará ser configurado aquí, es decir, seleccione el formato unipolar o bipolar y el formato de bits (12 o más). Esta área no está disponible si se ha elegido el formato común *Common format* (vea la página 8-26).



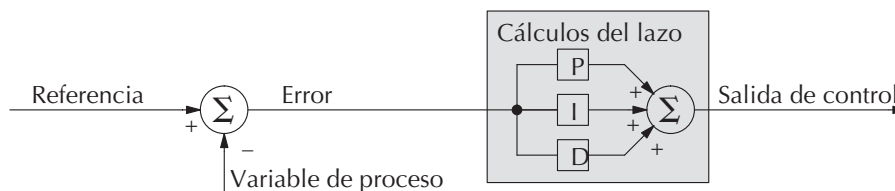
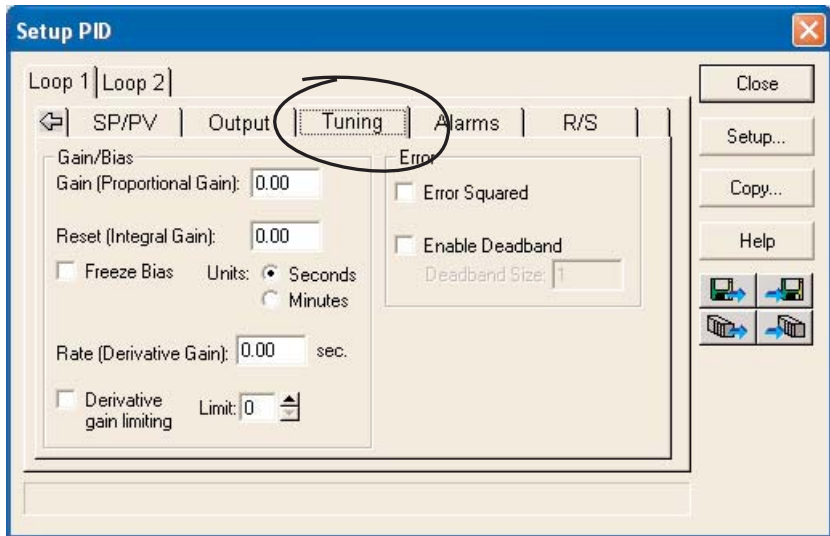
ADVERTENCIA: Si el límite superior se coloca como cero, la salida nunca será diferente a cero. En efecto, no habrá salida de control. Por defecto, el valor superior es cero!

Entrando los parámetros de control PID

Otro diálogo es **Tuning** (sintonizar) que sirve para colocar los parámetros de PID mostrados tales como: Ganancia (Proportional Gain), Reset (Integral) y Rate (Derivative).

Recuerde los modos de posición y de velocidad del algoritmo PID que fueron presentados anteriormente. Las ecuaciones muestran básicamente tres componentes del cálculo de PID: Ganancia proporcional (P),

Integral (I) y Derivativa (D). El diagrama de flujo siguiente muestra el cálculo de PID en el cual la salida de control es la suma de la parte proporcional, integral y derivativa. Con cada cálculo del lazo, cada término recibe el mismo valor de la señal de error.



Las ganancias P, I y D son números de 4 dígitos BCD con valores a partir de 0000 a 9999. Contienen una coma implicada en el centro, así que los valores son realmente 00,00 a 99,99. Algunos valores de ganancia tienen unidades - la ganancia proporcional no tiene ninguna unidad, la ganancia integral se puede seleccionar en segundos o en minutos, y la ganancia derivativa es en segundos.

Gain (Proportional Gain) – Éste es el factor más básico de los tres. Los valores van desde 0000 a 9999, pero se usan internamente como xx.xx. Una entrada de "0000" elimina el factor proporcional de la ecuación de PID. Esto acomoda usos que necesitan lazos integrales solamente.

Reset (Integral Gain) – Los valores están en el rango 0001 a 9998, pero se usan internamente como xx.xx. Una entrada de "0000" o de "9999" causa que el factor integral sea "infinito", eliminando el término integral de la ecuación de PID. Esto acomoda usos que necesitan lazos proporcionales solamente. Las unidades de la ganancia integral pueden ser segundos o minutos, según lo mostrado en el diálogo antedicho.

Rate (Derivative Gain) – Los valores que pueden colocarse están en el rango 0001 a 9999, solamente que se usa internamente como xx.xx. Una entrada de "0000" permite el retiro del factor derivativo de la ecuación de PID (una práctica común). Esto acomoda los usos que requieren solamente lazos proporcionales e integrales. La mayoría de los lazos de control funcionarán como un lazo PI.

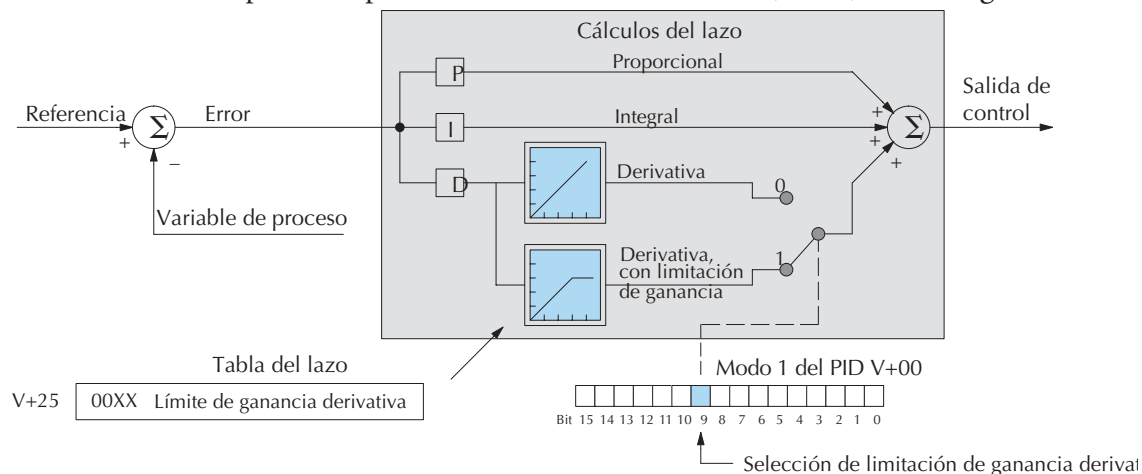


NOTA: Usted puede dejar espacios en blanco en el diálogo de **Tuning** y solamente entrar los parámetros en la ventana PID View de **DirectSOFT**.

Limitación de la ganancia derivativa

La ganancia derivativa (rate) tiene una función limitadora opcional. Se suministra esta función porque la compensación derivativa reacciona muy mal al ruido en la señal de la PV o a otras causas de fluctuaciones repentinas de la PV. Esta función se muestra en el diagrama de abajo.

El límite de esta ganancia puede ser particularmente útil durante la sintonización del lazo. La mayoría de los lazos pueden tolerar solamente un pequeño aumento sin entrar en oscilaciones incontrolables. Si se acepta esta opción, se debe definir un *Limit* (límite) en el rango de 0 a 20.



NOTA: Al configurar un lazo, es mejor usar el término de error estándar hasta después que se sintonice el lazo. Una vez sintonizado, usted podrá decir si estas funciones mejoran el control. Se puede seleccionar error al cuadrado y banda muerta más tarde en la configuración de PID. También, los valores se pueden definir más tarde en la ventana PID View de **DirectSOFT**.

Selección del modo de error

El término de error es interno al controlador PID, y se genera cada vez en cada cálculo de PID. Aunque sus datos no son directamente accesibles, usted puede calcularlos fácilmente restando: **Error = (SP - PV)**. Si está activada la extracción de raíz cuadrada de PV, entonces **Error = (SP - sqrt(PV))**.

De todos modos, el tamaño del error y el signo algebraico determina el próximo cambio de la salida del control para cada cálculo de PID.

Error Squared – Cuando está seleccionada, la función de error al cuadrado ajusta simplemente el término del error (pero preserva el signo algebraico original), que se utiliza en el cálculo. Esto afecta la salida de control disminuyendo su respuesta a valores más pequeños de error, pero manteniendo su respuesta a errores más grandes.

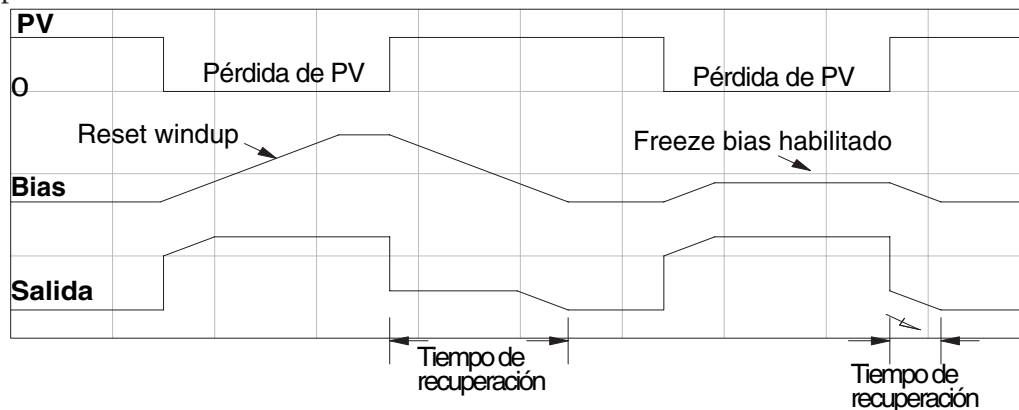
Estas son algunas situaciones en las cuales el error al cuadrado puede ser útil:

- Señal ruidosa del PV – usando un error al cuadrado puede reducir el efecto del ruido eléctrico de baja frecuencia en el PV, que hará el sistema de control “nervioso”. Un error al cuadrado mantiene la respuesta a errores más grandes.
- Proceso no lineal - algunos procesos (tales como control químico del pH) requieren controladores no lineales para mejores resultados. Otro uso es el control de un tanque pulmón, donde la señal de salida del control debe ser lisa.

El error de **banda muerta** – Cuando es seleccionado, la función de error de banda muerta toma un rango de valores de error cerca del valor cero, y simplemente substituye un cero como el valor del error. Si el error es más grande que el rango de banda muerta entonces se usa normalmente el valor de error.

Freeze Bias

La expresión *reset windup* (saturación de integración) se refiere a una característica indeseable del comportamiento del integrador del controlador PID que ocurre naturalmente bajo ciertas condiciones. Vea la figura de abajo. Supongamos que la señal de la PV se pierde, y el valor de la PV va a cero. Aunque ésto es una falla seria de control del lazo, se hace peor por *reset windup*. Note que el término *bias* sigue integrando el error normalmente durante la desconexión de la PV, hasta que se alcanza el límite superior. Cuando vuelve la señal PV, el valor *bias* se satura (se produce el *windup*) y toma un gran tiempo volver a lo normal. La salida del lazo por lo tanto tiene una recuperación demorada. Hasta la recuperación, el nivel de la salida es malo y causa otros problemas.



En el segundo evento de pérdida de la señal del PV en la figura, se activa *freeze bias*. Esto hace que el valor de *bias* se mantenga cuando la salida de control sale de límites. Se evita así el problema de saturación y el tiempo de recuperación de la salida es mucho menor.

Para la mayoría de las aplicaciones, la función *freeze bias* funcionará con el lazo según lo descrito arriba. Se sugiere activar esta función seleccionándola en el diálogo. El bit 10 de la palabra V+00 se puede también configurar en lógica ladder.

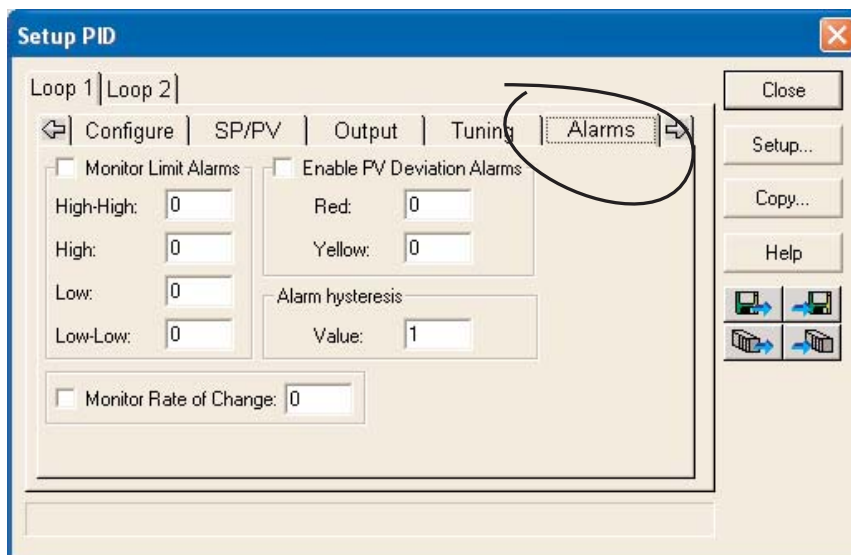


NOTA: *Freeze bias* hace que el *bias* no cambie de valor cuando la salida de control alcanza el extremo del rango de la salida. Si usted ha definido límites en la salida de control diferente que el del rango (es decir, 0-4095 para un lazo unipolar/12bit), el *bias* aún tratará de usar el extremo del rango para el punto donde para y por lo tanto la función no trabajará.

Configurando las alarmas del control PID

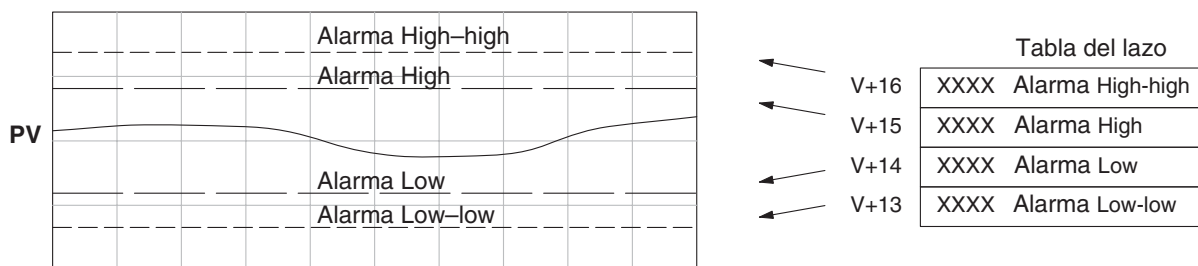
Aunque la configuración de las alarmas es opcional, usted no quisiera hacer funcionar un proceso sin supervisión. El funcionamiento de un lazo de control de proceso puede ser medido generalmente por cómo la variable de proceso sigue la referencia (setpoint). La mayoría de los lazos de control de proceso en la industria funcionan continuamente y pueden perder el control de la PV debido a una condición de error. Las alarmas de proceso son vitales para descubrir al comienzo una condición de error del lazo y pueden alertar al personal de planta para controlar manualmente un lazo o para tomar otras medidas hasta que se ha reparado la condición de error.

Los umbrales (o puntos de activación) de las alarmas son completamente programables, y cada tipo de alarma puede ser activado y supervisado independientemente. El diagrama siguiente muestra el diálogo del alarma en PID setup lo cual simplifica la configuración de las alarmas.



Supervisión de Alarmas de límite

Aceptando la caja *Monitor Limit alarms* permitirá que todas las alarmas de límite de la PV sean supervisados cuando los límites sean definidos. Los valores absolutos de las alarmas se organizan como dos alarmas superiores y dos inferiores. El estado de alarmas es falso mientras el valor del PV se encuentre en la región entre las alarmas superiores e inferiores, según lo mostrado abajo. Las alarmas más cerca la zona ideal se nombran *High Alarm* y *Low Alarm*. Si el lazo pierde control, el PV cruzará uno de estos umbrales primero. Por lo tanto, usted puede programar los valores de umbral apropiados de alarmas en las localizaciones de la tabla del lazo



mostradas en la figura a la derecha.

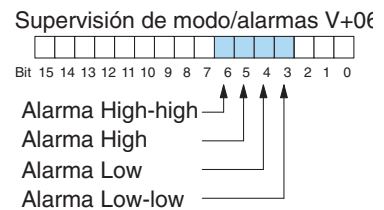
El formato de datos es igual que el PV y el SP (12-bit o 15-bit). Los valores de umbral para estas alarmas se deben definir para dar a un operador una detección temprana si el proceso pierde control.



NOTA: El diálogo de alarmas se puede dejar como aparece, sin entradas de alarmas. Las alarmas se pueden definir posteriormente en la ventana de PID View de **DirectSOFT**.

Si el proceso sigue fuera de control por un cierto tiempo, la PV cruzará más tarde uno de los umbrales externos de alarmas, nombrados *High-high alarm* y *Low-low alarm*. Se programan sus valores de umbral usando los registros de la tabla del lazo enumerados en la figura anterior. Una alarma *High-high* o *Low-low* indica que existe una condición seria y que necesita la atención inmediata del operador.

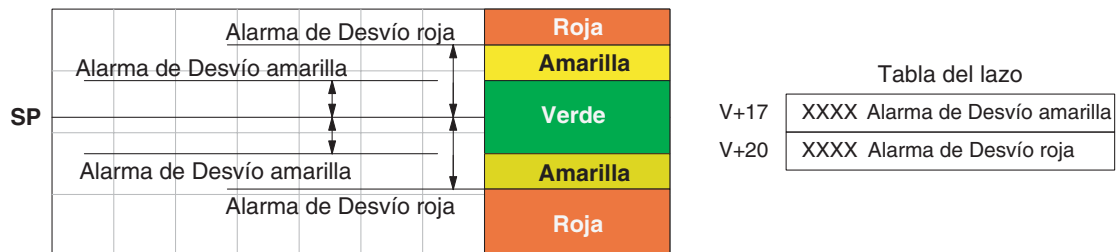
Las alarmas de valor absoluto de la PV se informan en los cuatro Bits en la palabra de estado V+06 en la tabla del lazo, según lo mostrado a la derecha. Recomendamos usar lógica ladder para supervisar estos bits. Las instrucciones de las instrucciones bit-of-word hacen ésto muy fácil. Además, usted puede supervisar alarmas de PID usando *DirectSOFT* con la ventana PID View.



Alarmas de desvío de la PV

La alarma de desvío de la variable de proceso PV supervisa cuanto se ha desviado la PV con respecto al valor de referencia (SP). La alarma de desvío tiene dos umbrales programables, y cada umbral se aplica igualmente sobre y debajo del valor actual del SP.

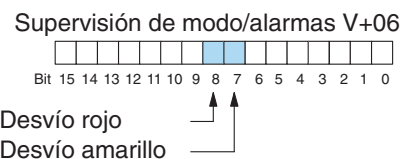
En la figura de abajo, la alarma de desvío más pequeña se llama "desvío amarillo" (Yellow Deviation), indicando una condición de precaución. La alarma de desvío más grande se llama



"desvío rojo" (Red Deviation), indicando una condición de error fuerte. Los valores de umbral usan las localizaciones V+17 y V+20 de la tabla de parámetros del lazo según lo mostrado.

Los umbrales definen las zonas, que fluctúan con el valor de SP. La zona verde que rodea el valor del SP representa una condición normal (sin ninguna alarma). Las zonas amarilla están fuera de la zona verde, y las zonas rojas están más allá de éstas.

Las alarmas de desvío de la PV se informan en los dos bits en la palabra de estado V+06 en la tabla del lazo, según lo mostrado a la derecha. Recomendamos usar lógica ladder para supervisar estos bits. Las instrucciones de bit-of-word hacen esto muy fácil. Además, usted puede supervisar alarmas de PID usando *DirectSOFT* y con el diálogo PID View abierto.



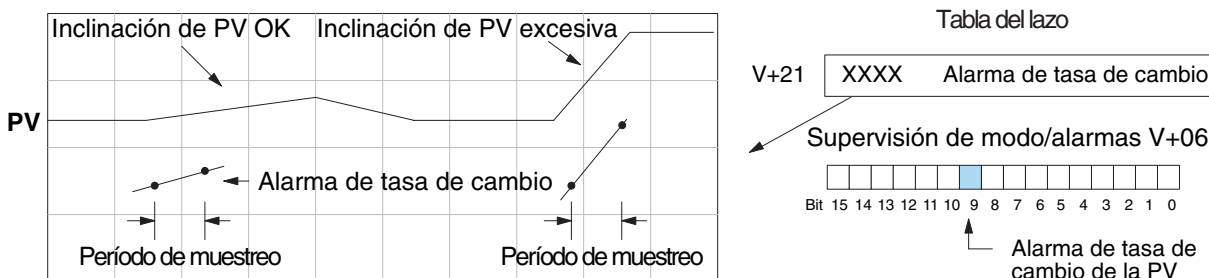
La alarma de desvío de la PV se puede activar o desactivar independientemente de las otras alarmas de la PV, usando el bit 13 de la palabra V+00 de la tabla PID.

Recuerde que la función de histéresis de alarmas trabaja conjuntamente con las alarmas de desvío y de valor absoluto, y se discute al final de esta sección.

Alarma de tasa de cambio de la PV

Una muy buena manera de conseguir una detección temprana de una falla de proceso es supervisar esta alarma. La mayoría de los procesos de hornada de la PV tiene masas grandes y valores de PV que cambian lentamente. Un PV relativamente que cambia rápido resultará por ejemplo de una pérdida de señal de la PV por un cable cortado o la salida, un error del valor del SP, u otra causa. Si el operador responde a esta alarma rápidamente y con eficiencia, el valor absoluto de la PV no alcanzará el punto donde el material en proceso podría ser arruinado.

El controlador PID del PLC DL06 proporciona una alarma de tasa de cambio de PV programable, según lo mostrado abajo. La tasa de cambio se especifica en cambio de unidades de la PV por período de muestreo del lazo. Este valor se programa en la localización V+21 de la tabla PID.



Como ejemplo, supongamos que la PV es la temperatura para su proceso, y usted desea una alarma siempre que los cambios de temperatura sean más rápidos que 15 grados/minuto. Se debe saber las unidades de la PV por grado y la tasa de muestreo del lazo. Entonces, supongamos que el valor de la variable de proceso PV (en la localización V+03) representa 10 unidades por grado, y el período de muestreo del lazo es de 2 segundos (30 muestreos por minuto). Use la fórmula de abajo para convertir las unidades de ingeniería al período de unidades/muestreo:

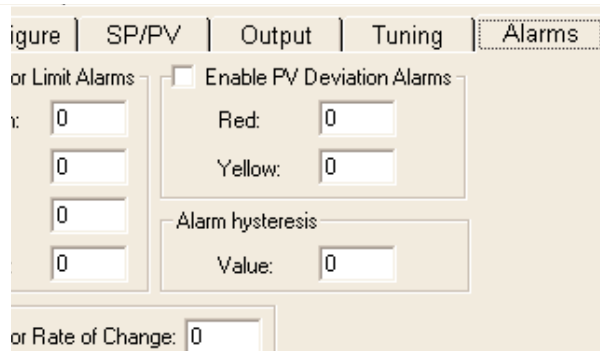
$$\text{Alarma de tasa de cambio de PV} = \frac{15 \text{ grados}}{1 \text{ minuto}} \times \frac{10 \text{ conteos / grado}}{30 \text{ muestreos / min.}} = \frac{150}{30} = 5 \text{ conteos/período de muestreo}$$

Usted programaría el valor 5 en la tabla del lazo para la alarma tasa de cambio, desde el resultado del cálculo. Esta alarma se puede activar y desactivar independientemente de las otras alarmas de la PV, usando el bit 14 de la palabra V+00.

La función de histéresis de alarmas (discutida más adelante) no afecta la alarma Tasa de cambio.

Histéresis de alarmas de la PV

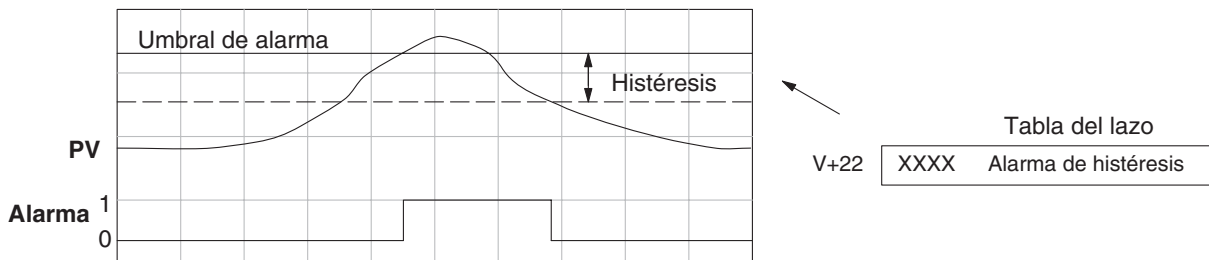
Se programan las alarmas de valor absoluto y de desvío de la PV usando valores de umbral. Cuando el valor absoluto o el desvío excede el umbral, el estado de la alarma se hace verdadero. Las señales del mundo real de la PV tienen cierto ruido en ellos, que pueden causar una cierta fluctuación en el valor de la PV en la CPU. Cuando el valor de la PV cruza un umbral de alarma, sus fluctuaciones hacen que la alarma sea intermitente y moleste a los operadores de proceso.



La solución es usar la función de histéresis de alarmas de la PV.

La cantidad de histéresis de alarmas de la PV es programable a partir de 1 a 200 (binario/decimal). Al usar la alarma de desvío de la PV, la cantidad programada de histéresis debe ser menor que la cantidad programada del desvío. La figura de abajo muestra cómo se aplica la histéresis cuando el valor de la PV pasa de un umbral y luego vuelve atrás.

La cantidad de la histéresis se aplica después de que se cruce el umbral, y hacia la zona segura. De esta manera, la alarma se activa inmediatamente sobre el valor de umbral programado. Atrasa el momento de apagado hasta que el valor de la PV ha retornado cruzando el umbral por la cantidad de histéresis definida.



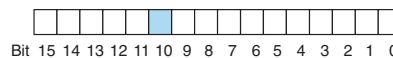
8

Error de programación de Alarmas

Los valores de umbral de alarmas de la PV deben tener ciertas relaciones que deben ser válidas. Los requisitos se enumeran abajo. Si no son satisfechos, se activará el bit de error de programación de alarma, según lo indicado a la derecha.

- Requisitos de valores de alarmas absolutas:
Low-low < Low < High < High-high
- Requisitos de alarmas de desvío:
Yellow < Red

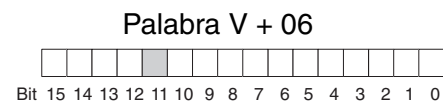
Supervisión de modo/alarmas V+06



↑ Error de programación de alarmas

Error de Cálculo Overflow/Underflow

Este error ocurre siempre que la salida llega a un límite superior o inferior y el PV no alcanza la referencia (setpoint). Un ejemplo típico puede ser cuando se atasca una válvula, la salida está en el límite, pero la PV no llega a la referencia (setpoint). En *DirectSOFT*, esto se indica con un color rojo y aparece una indicación con el código OVR.



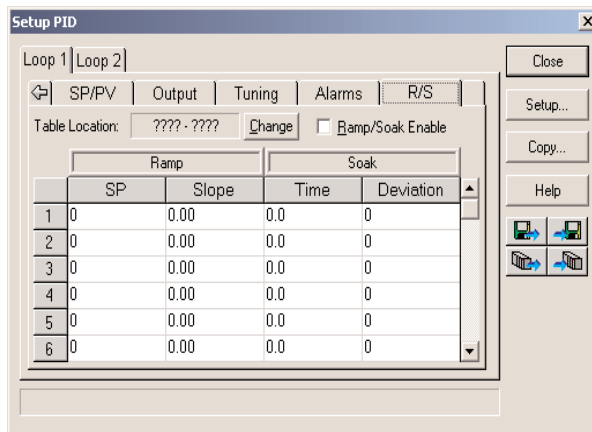
↑ Error de Cálculo Overflow/Underflow



NOTA: Los bits de error en esta página se pueden supervisar solamente en Data View de *DirectSOFT*; sin embargo, overflow/underflow se puede alarmar en PID View. El panel de operador G-more (vea el sitio de Internet de automationdirect) se puede configurar también para leer estos bits de error usando el objeto **PID faceplate**.

Ramp/Soak

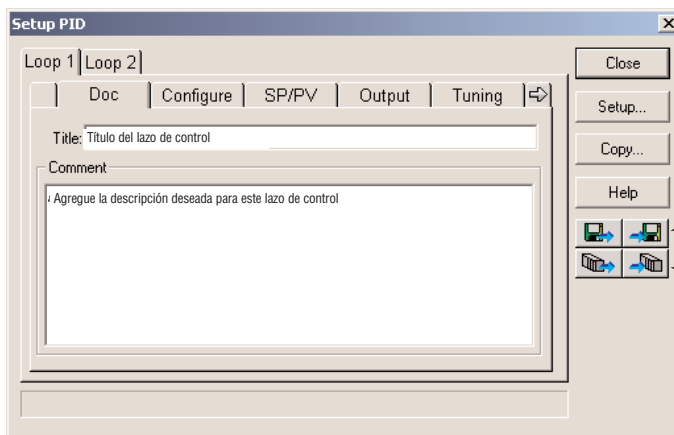
R/S (Ramp/Soak) es el último diálogo disponible en la configuración del controlador PID. El PID básico no requiere ninguna entrada para hacer funcionar el lazo de PID. Ramp/Soak será discutido en otra sección.



Complete la configuración del control del lazo PID

Una vez que usted haya completado la información necesaria para la configuración de los lazos PID, la configuración se debe salvar. Los íconos en el diálogo de configuración PID le permiten salvar la configuración al PLC y al disco. Los íconos para salvar tienen la flecha apuntando al PLC y al disco. Los íconos de lectura tienen las flechas saliendo del PLC y del disco.

Hay disponible otra lengüeta; la lengüeta **Doc** en el diálogo de configuración PID. Usted puede entrar un nombre y una descripción para el lazo. Esto es útil si hay más de un lazo de PID.



NOTA: Es buena práctica salvar el proyecto después de configurar el controlador PID seleccionando **File** desde el menú, luego **Save project > to disk**. Además de salvar su proyecto, se salvan todos los parámetros de PID. Si Ud solamente hace **Write to disk**, solamente se salva el diagrama ladder.

Sintonizando lazos de control PID

Una vez que usted haya instalado un lazo de control PID en su sistema, éste debe ser sintonizado para que trabaje adecuadamente. El objetivo de sintonizar es ajustar las ganancias del algoritmo de control de modo que el lazo tenga un funcionamiento óptimo en condiciones dinámicas. La calidad del funcionamiento de un lazo puede ser juzgada generalmente por cómo sigue la variable de proceso PV a la referencia SP a un cambio escalón. Hay que tener presente que es fundamental entender el proceso para obtener un lazo de control bien diseñado. Los sensores deben estar en localizaciones apropiadas y las válvulas se deben dimensionar correctamente con ajustes apropiados. *El control PID no tiene valores típicos.* No hay un control de proceso que sea idéntico a otro.

Sintonización Manual versus Automática

Usted puede ingresar valores de ganancias en el control PID (sintonización manual) o usted puede confiar en el PLC, que procesará algunos datos para calcular automáticamente los valores de ganancia (sintonización automática). La mayoría de los ingenieros de proceso experimentados tendrán un método preferido; el DL06 acomodará cualquier preferencia. El uso de la sintonía automática puede eliminar mucho del ensayo y error del método manual, especialmente si usted no tiene mucha experiencia en sintonías. Sin embargo, la ejecución del procedimiento de sintonía automática obtendrá solamente ganancias aproximadas a los valores óptimos, pero una sintonía manual adicional puede hacer que las ganancias lleguen sus valores óptimos.

ADVERTENCIA: Solamente personal autorizado y completamente familiar con todos los aspectos del proceso debe realizar los cambios que afectan las constantes del lazo. Usar el procedimiento de sintonía automática afectará el proceso, incluyendo cambios grandes en el valor de la salida de control. Asegúrese de considerar el impacto de cualquier cambio para reducir al mínimo el riesgo de lesión al personal o daño al equipo. La sintonía automática en el DL06 NO debe ser utilizada como reemplazo para el conocimiento del proceso.



Prueba del lazo abierto

Ya sea que se use el método de sintonía manual o automático, es muy importante verificar características básicas de un nuevo proceso instalado antes de sintonizarlo. Con el lazo en modo manual, verifique los siguientes puntos para cada nuevo lazo, con Data View o con PID View.

- **Setpoint** – Verifique que el origen de la referencia (SP) pueda generar un valor. Coloque el PLC en modo RUN y deje el lazo en modo manual, después supervise la localización V+02 de la tabla del lazo para ver el valor SP (si está usando el generador de ramp/soak, pruébelo ahora).
- **Process Variable** – Verifique que el valor de la PV sea una medida exacta, y los datos de la PV que lleguen en la localización V+03 de la tabla del lazo están correctos. Si la señal del PV es muy ruidosa, considere filtrar la entrada a través de hardware (filtro pasa bajo RC) o use el filtro por software, tal como el mostrado en este capítulo.
- **Control Output** – Si es seguro hacer así, cambie manualmente la salida a una cantidad pequeña (quizás el 10%) y observe el efecto en la variable de proceso. Verifique que el proceso sea de acción directa o reversa, y compruebe el ajuste para saber si hay la salida de control (invertida o no invertida). Asegúrese de que los límites superiores e inferiores de la salida de control no sean iguales uno al otro.
- **Sample Rate** – Cuando haga el funcionamiento de lazo abierto, éste es un buen momento para encontrar la tasa ideal de muestreo. Sin embargo, si usted va a usar sintonía automática, este procedimiento calculará automáticamente la tasa de muestreo además de las ganancias de PID.

Recomendamos usar *DirectSOFT* para ejecutar la sintonía manual o automática.

El diálogo **PID View** permite iniciar comandos con estos métodos, y los resultados se pueden visualizar en el diagrama de graficos disponible en esa ventana de diálogo.

Familiarícese con los diversos comandos de la parte de sintonía automática del diálogo de **PID View** mirando la figura siguiente.

- La parte a la izquierda muestra los diversos lazos de control.
- La parte derecha está dividida en tres partes.
- La parte a la derecha superior presenta el gráfico de tendencias, con ajuste de las escalas de unidades y escala de tiempo, para el valor de la variable de proceso PV, la referencia SP, y la salida de control CV. También se muestra el valor de bias a lo largo del tiempo.
- La parte media muestra en valores numéricos cada uno de los datos mas interesantes de cada lazo de control. Allí pueden ajustarse los valores de ganancia, reset y rate durante una sintonización. Se pueden definir las alarmas, las que quedan almacenadas en la memoria, después de colocar los valores límites en los campos correspondientes.
- La parte inferior se usa para la sintonía automática.

The screenshot displays the **PID View** window. On the left, there are two control loops: 'A-Nivel 5- Ovale' and 'A-Nivel 10- Ovale'. The 'A-Nivel 5- Ovale' loop shows SP:3600, PV:3601, BIAS:1999, and OUT:1947. The 'A-Nivel 10- Ovale' loop shows SP:0, PV:0, BIAS:0, and OUT:0. The main area contains a trend graph with three traces: SP (Setpoint), PV (Process Variable), and Output. The SP trace is a step function that jumps from 3000 to 4000. The PV trace shows a smooth curve following the step. The Output trace shows a step function that jumps from 2955 to 4330. Below the graph, there are numerical values for SP (3600), PV (3601), BIAS (1999), and Out (1947). At the bottom, there are settings for the PID controller, including Gain (52.13), Reset (23.90), Rate (0.00), and Autotune options. A 'Start Tuning' button is visible, and a message below it says 'Autotune complete, check PID parameters'.

Sintonía automática

Inicie la Sintonía Automática

Procedimiento de sintonía manual (o no automática)

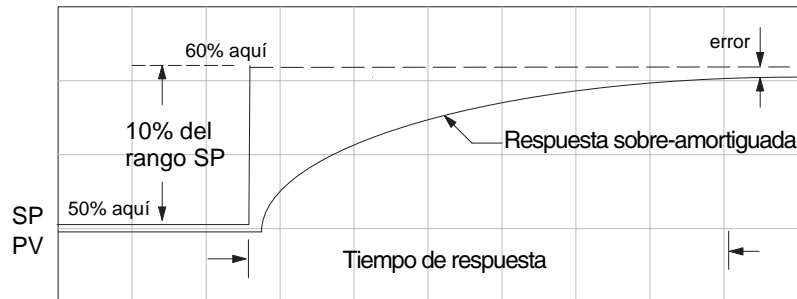
No es conveniente tratar de obtener los mejores valores para P, I y D en el PID por ensayo y error sin un método. Lo que sigue es un procedimiento típico para sintonizar un lazo de control de temperatura que usted puede usar para sintonizar su lazo de control.

Supervise los valores de SP, de la PV y de la CV con un instrumento de gráficos o use la función de PID View en *DirectSOFT* (vea la página 8-53).



NOTA: Recomendamos usar el diálogo PID View con selección manual para la escala vertical del área de SP/PV y las áreas de salida de Bias/Control. La función automática de escala cambia la escala vertical en los parámetros de proceso y agrega confusión al proceso de sintonización del lazo.

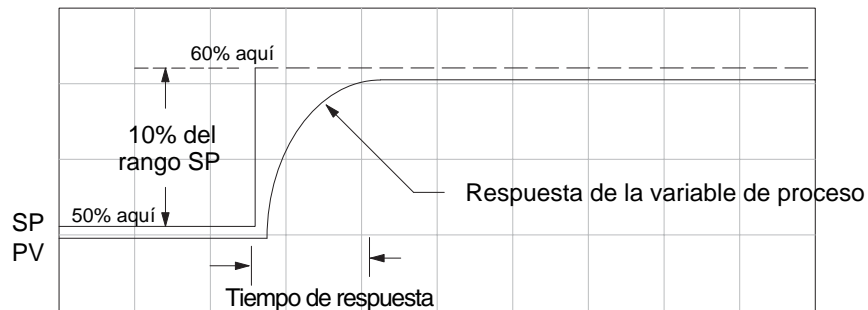
- Ajuste las ganancias de modo que la ganancia P = 0.5 o 1.0 (1.0 es un buen valor basado en experiencia), integral = 9999 (esto elimina básicamente el reset) y el factor derivativo = 0000. Esto deshabilita las compensaciones integral y derivativa y proporciona un cierto valor proporcional adecuado para que no oscile el sistema.
- Ajuste el valor bias en PID View y colóquelo en cero.
- Coloque la referencia SP a un valor igual al de operación o en este ejemplo al 50% del rango total.
- Ahora, seleccione modo **Auto**. Si el lazo no permanece en modo Auto, verifique las instrucciones de localización de fallas en el final de este capítulo. Permita que el PV se estabilice alrededor del punto escogido (aquí es el 50% del rango). El PV y el SP deben estar estables y muy cercanos.



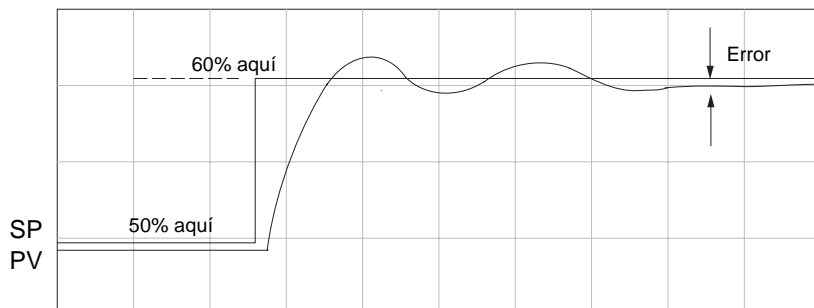
- Cambie el SP al 60% del rango con un escalón de 10% sobre lo seleccionado.

La respuesta puede demorar un momento, pero usted verá que no debería haber ninguna oscilación. Esta respuesta no es deseable puesto que toma un tiempo largo para corregir el error y también hay una diferencia entre el SP y el PV.

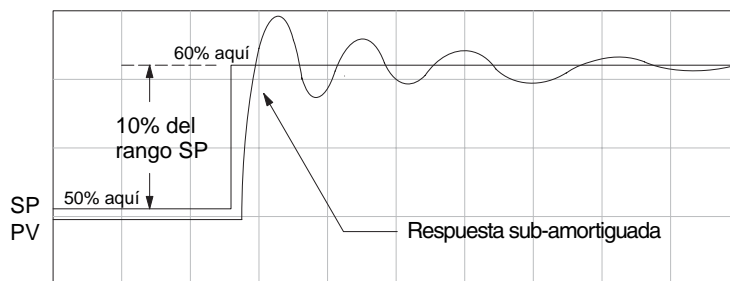
- Aumente la ganancia proporcional, por ejemplo a 2.0. La salida de control será mayor y el tiempo de respuesta será más rápido. La tendencia debe ser similar a la figura de abajo.



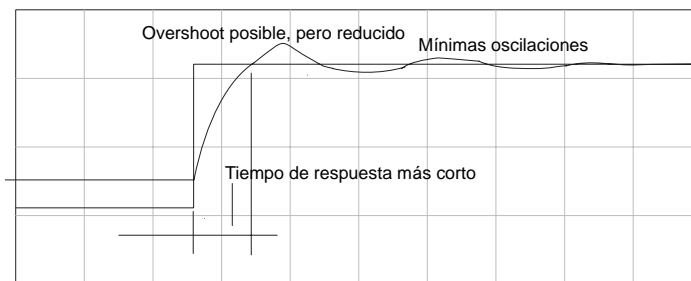
- Aumente la ganancia proporcional en incrementos pequeños, tales como 4, 6, 7, etc. hasta que la respuesta de la salida de control comience a oscilar. Digamos que sea 10,0. Ésta es la ganancia proporcional que se debe guardar temporalmente como referencia para lo que sigue.



- Ahora, devuelva la ganancia proporcional de tal modo de conseguir una respuesta estable, por ejemplo, 9.7. El error, SP-PV, va a ser pequeño, pero no cero.
- Luego agregue un poco de ganancia integral (Reset) de modo que el error tienda a ser a cero. Comience usando 80 segundos (ajuste en minutos en caso de necesidad). El error debe disminuir.
- Siga ajustando la ganancia integral a un valor más bajo, tal como 50, para una respuesta diferente. Si no hay una respuesta significativa, continúe disminuyendo el valor hasta que la respuesta llegue a ser inestable. Vea la figura de abajo.



- Para la discusión, digamos que un valor del reset de 35 hizo que la salida de control inestable. Vuelva el valor del reset a un valor estable, tal como 38. **Tenga cuidado con este ajuste puesto que la oscilación puede destruir el proceso.**
- La respuesta de la salida de control debe ser óptima ahora, sin compensación derivativa. Los valores del ejemplo son: ganancia proporcional = 9.7 e integral = 38 segundos. Observe que se ha reducido al mínimo el error.



El método descrito es el método más común usado para sintonizar un lazo de PID. Casi nunca se usa el factor derivativo en un lazo del control de temperatura. Este método se puede también usar para otros lazos de control, pero se puede necesitar el rate para una salida estable de control.

Pruebe el control PID para un PV alto de por ejemplo 80% y nuevamente para un PV bajo de 20%, y corrija los valores en caso de necesidad. Pequeños ajustes de los parámetros pueden hacer que la salida de control haga el control más exacto o más inestable. Es a veces aceptable hacer un pequeño *overshoot* para hacer que el control reaccione más rápido.

El Rate (ganancia derivativa) puede ser provechoso para lazos de control que no estén controlando temperatura. Para estos lazos, agregue un valor de 0,5 para el Rate y vea si esto mejora la salida de control. Si no hay una respuesta significativa, aumente el valor en incrementos de 0,5 hasta que haya una mejora de la salida. Recuerde que el Rate reacciona con una tasa de cambio del error.

Procedimientos alternativos de sintonización no automática

Los procedimientos siguientes se han extraído de varias publicaciones sobre control de proceso. Estos procedimientos son alternativas al procedimiento mostrado en las páginas anteriores.

Método Zeigler-Nichols – “Decaimiento de un cuarto de la amplitud”

1. Deje sin efecto el reset y rate; coloque la ganancia P a un valor bastante grande.
2. Haga un cambio pequeño del valor de referencia (setpoint) y observe cómo la variable controlada varía.
3. Ajuste la ganancia P hasta que la oscilación es de amplitud constante; este valor de ganancia es la ganancia G_u .
4. Mida el período de la oscilación en minutos. Éste es el período P_u .
5. Calcule los valores de ganancia en el controlador PID como sigue:

$$\text{Cuando es solamente P : } G = G_u/2$$

$$\begin{aligned} \text{Cuando es P \& I : } & G = G_u/2.2 \\ & T_i = 1.2/P_u \quad (\text{repeticiones/minuto}) \end{aligned}$$

$$\begin{aligned} \text{Cuando es P-I-D: } & G = G_u/1.6 \\ & T_i = 2.0/P_u \quad (\text{repeticiones/minuto}) \\ & T_d = P_u/8.0 \quad (\text{minutos}) \end{aligned}$$

Método Pessen

1. Siga el procedimiento descrito arriba (Zeigler-Nichols) para determinar la ganancia G_u y el período P_u .
2. Aplique las fórmulas de más abajo.

Para que no haya *overshoot* durante el funcionamiento:

$$\begin{aligned} G &= G_u/5.0 \\ T_i &= 3/P_u \quad (\text{repeticiones/minuto}) \\ T_d &= P_u/2 \quad (\text{minutos}) \end{aligned}$$

Aceptando un pequeño *overshoot*, pero mejorando la respuesta a las perturbaciones:

$$\begin{aligned} G &= G_u/3 \\ T_i &= 3/P_u \quad (\text{repeticiones/minuto}) \\ T_d &= P_u/3 \quad (\text{minutos}) \end{aligned}$$

El método automático de sintonía, explicado a seguir en las próximas páginas, usa este tipo de consideraciones para calcular un valor estimado de ganancia proporcional, del valor de reset y del rate.

Procedimiento de sintonía automática

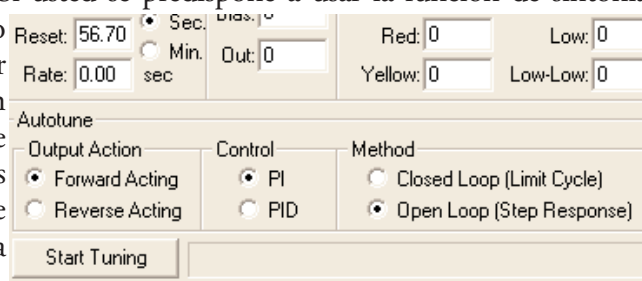
La función de sintonización automática del controlador PID del DL06 funcionará solamente con el comando del ingeniero de control de proceso. Por lo tanto, la sintonización automática no funciona continuamente durante la operación (éste sería control *adaptivo*). Siempre que haya un cambio substancial en la dinámica del lazo, tal como la masa del proceso, tamaño del actuador, etc., el proceso debe ser repetido para calcular las nuevas ganancias requeridas para un control óptimo.



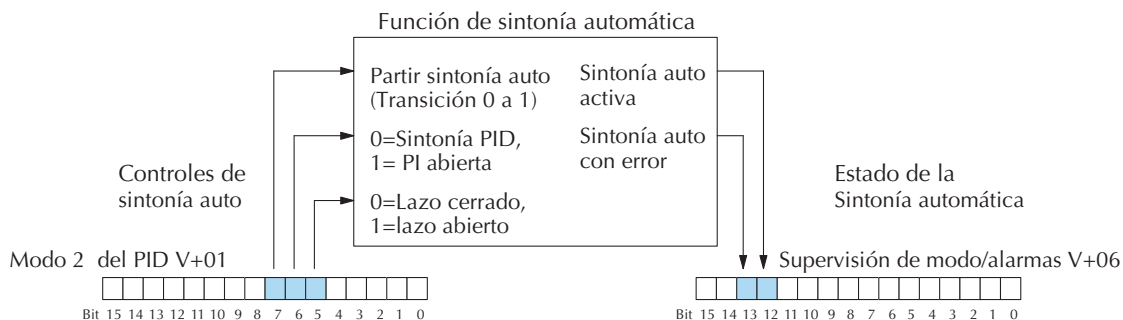
ADVERTENCIA: Solamente personal autorizado completamente familiar con todos los aspectos del proceso debe realizar cambios que afectan las constantes del lazo. Al usar procedimientos de sintonía del lazo se afectará el proceso, incluyendo cambios grandes en el valor de la salida de control. Asegúrese de considerar el impacto de cualquier cambio para reducir al mínimo el riesgo de lesión al personal o daño al equipo. La sintonización automática en el DL06 no reemplaza el conocimiento del proceso.

Una vez que los componentes físicos del lazo estén conectados con el PLC, será iniciada la sintonización automática y puede ser usado para establecer valores iniciales de parámetros del lazo PID. Este procedimiento es el mejor valor que la CPU puede estimar por cálculo después de algunas pruebas de ensayo.

Recomendamos usar *DirectSOFT* para ejecutar la sintonía automática. El controlador PID ofrece métodos de lazo cerrado y de lazo abierto. Si usted se predispone a usar la función de sintonía automática, le recomendamos usar el método de lazo abierto primero. Esto le permitirá usar el método a lazo cerrado cuando el lazo esté en operación (modo Automático) y cuando no se puede cerrar (modo manual). Las secciones siguientes describen cómo usar la función de sintonía automática, y qué ocurre en la sintonía de lazo abierto y cerrado.

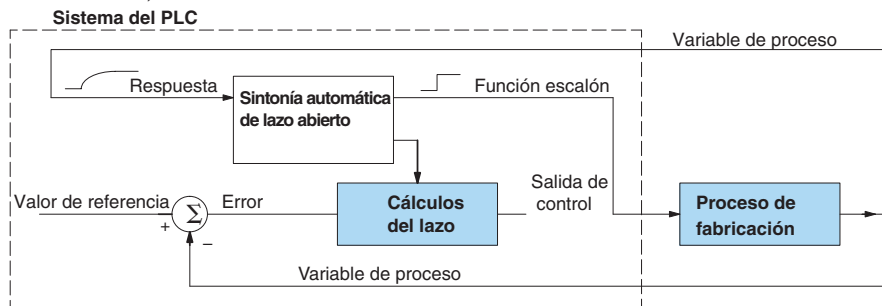


Los controles para la función de sintonía automática utilizan tres bits en la palabra V+01 del modo 2 de PID, según lo mostrado abajo. *DirectSOFT* manipulará estos bits automáticamente cuando usted usa la función de sintonía automática dentro de *DirectSOFT*. O, usted puede tener acceso a estos bits en la lógica ladder directamente para permitir control de otra fuente tal como una interfase de operador dedicada. Los bits de control individual permiten que usted comience el procedimiento de sintonía automática, selecciona PID o PI y sintonía de lazo cerrado o abierto. Si usted selecciona la sintonía PI, el procedimiento de sintonía automática elimina la componente derivativa. La palabra V+06 de estado de alarmas divulga el estado de la sintonía automática según lo mostrado. El bit 12 estará en (1) durante el ciclo de sintonía automática, volviendo automáticamente a (0) cuando está completado.



Sintonía automática en lazo abierto

Durante una sintonización automática de lazo abierto, el controlador PID funciona según lo mostrado en el diagrama de abajo. Antes de comenzar este procedimiento, ponga el control PID en modo manual y asegúrese que la PV y los valores de salida del control están en el centro de sus rangos (lejos de finales de escala).



NOTA: En teoría, el valor del SP no importa en este caso, porque el lazo no es cerrado. Sin embargo, el requisito de firmware es que el valor del SP debe ser más del 5% del rango del PV desde el valor corriente de la PV antes de comenzar el ciclo automático de sintonía (para el DL06, una PV de 12 bits debe estar 205 unidades o más debajo del SP para los lazos de acción directa, o 205 unidades o más sobre el SP para los lazos de acción reversa)

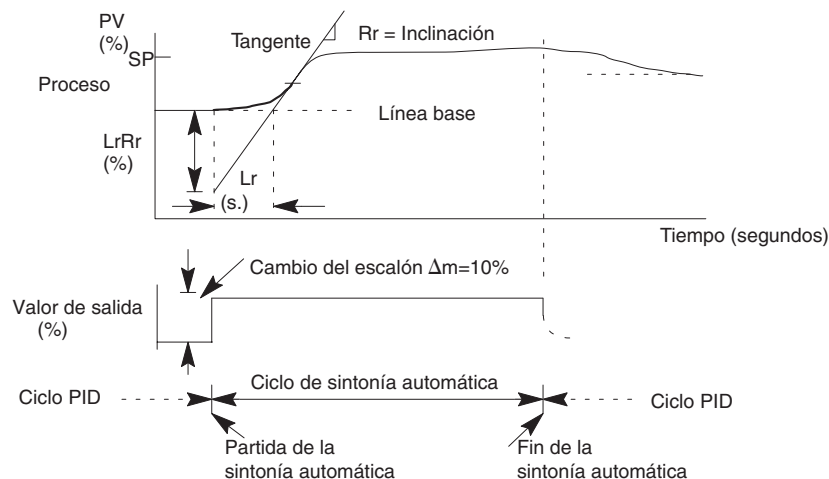


8

Cuando se sintoniza en modo automático, el controlador PID produce un cambio escalón en la salida y observa la respuesta de la PV. Con la respuesta de la PV, la función calcula las ganancias y el tiempo de muestreo. Pone automáticamente los resultados en los registros correspondientes en la tabla del lazo PID.

El esquema siguiente muestra lo que ocurre en el ciclo de lazo abierto de sintonía automática. La función de sintonía automática toma el control de la salida de control e induce un cambio escalón de 10% del rango. Si el cambio de PV que el controlador de lazo observa es menos de 2%, entonces el cambio en la salida es aumentado a 20% del rango.

- * Cuando comienza la sintonía, cambie la salida en un escalón de $\Delta m = 10\%$
- * Durante la sintonía automática, la salida de control alcanzó el límite positivo de la escala total. La sintonía automática paró y se activó el bit de error de sintonía automática en el bit de palabra de alarma.
- * Cuando el cambio de PV está abajo de 2%, la salida de control se cambia a 20%.



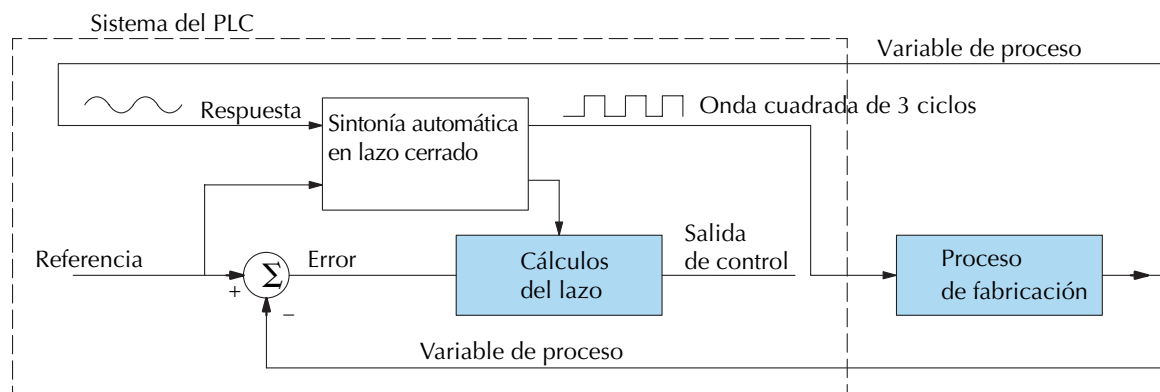
Cuando se completan las observaciones de sintonía del lazo, el controlador de lazos calcula R_r (rampa máxima en % /sec.) y L_r (tiempo muerto en segundos). La función de sintonía automática calcula las ganancias según las ecuaciones de Ziegler Nichols, mostradas abajo:

Sintonización PID	Sintonización PI
$P=1.2 * \Delta m / L_r R_r$	$P=0.9 * \Delta m / L_r R_r$
$I=2.0 * L_r$	$I=3.33 * L_r$
$D=0.5 * L_r$	$D=0$
Período de muestreo = $0.056 * L_r$	Período de muestreo = $0.12 * L_r$
Δm = cambio a un escalón (10% = 0.1, 20% = 0.2)	

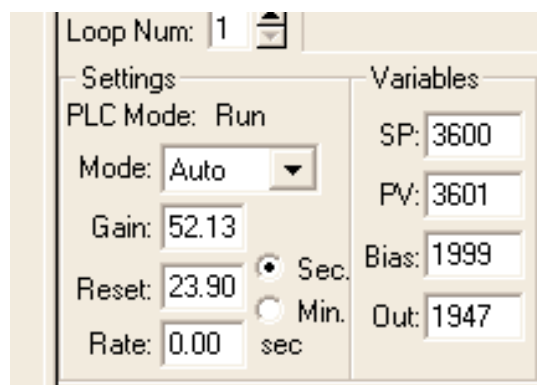
La duración de cada ciclo de sintonía automática dependerá de la masa del proceso. Una PV que cambia en forma lenta tendrá como resultado un tiempo mayor del ciclo de sintonía automática. Cuando la sintonía automática está completa, la ganancia proporcional, integral y derivativa se actualizan automáticamente en las direcciones de tabla del lazo **addr + 10**, **addr + 11**, y **addr + 12** respectivamente. El período de muestreo en el **addr + 07** es actualizado también automáticamente. Usted puede probar la validez de los valores del procedimiento de sintonía automática midiendo la respuesta del PV del lazo cerrado a un cambio escalón en la salida de control. Las instrucciones de cómo hacer esto están en la sección del procedimiento manual de sintonía (localizado antes de esta sección).

Sintonía automática en lazo cerrado

Durante un ciclo de sintonía automática de lazo cerrado, el controlador de lazos opera de acuerdo al diagrama de abajo.

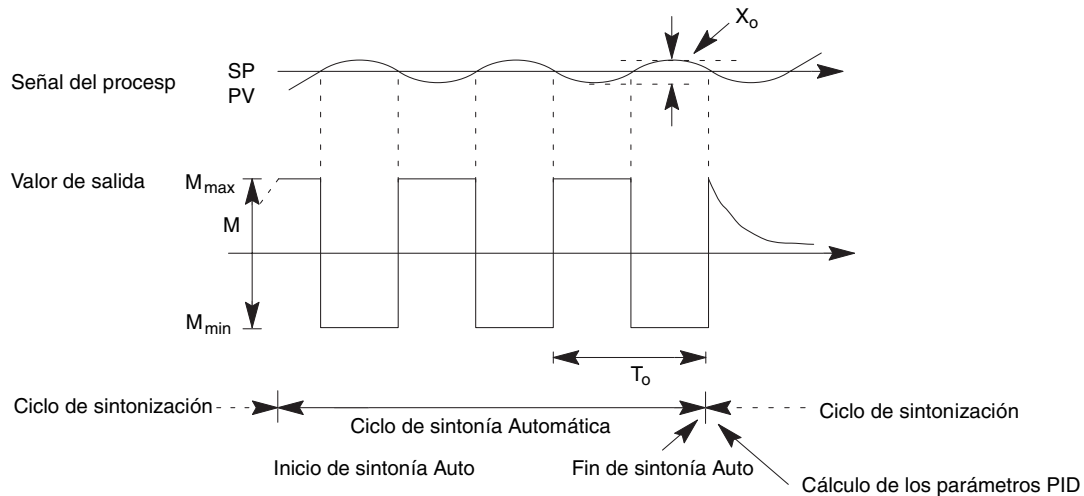


Cuando se ejecuta sintonía automática, el controlador de lazos impone una onda cuadrada en la salida de control. Cada transición de salida ocurre cuando el valor de PV cruza el valor de SP. Con la respuesta de la PV, la función de sintonía automática calcula las ganancias y el tiempo de muestreo ideales. Coloca automáticamente los resultados en las memorias correspondientes en la tabla del lazo de control PID.



Capítulo 8: Operación de control PID

El diagrama de tiempo siguiente muestra los eventos que ocurren durante la sintonización automática de lazo cerrado. La función de sintonía automática examina la dirección de desvío del PV desde el SP. La función entonces toma el control de la salida de control y produce un cambio escalón de cierta amplitud en la dirección opuesta. Cada vez que el signo del error (SP - PV) cambia, la salida cambia a la misma amplitud en la dirección opuesta. Esto sucede durante tres ciclos completos.



*Mmax = Limite superior de la salida de control.

*Mmin = Limite inferior de la salida de control.

* Este ejemplo es de acción directa.

Cuando es de acción reversa, la salida será invertida. Cuando las observaciones de la sintonía se han completado, se calcula T_o (período de un ciclo) y X_o (amplitud de la oscilación de la variable de proceso PV peak to peak). Luego usa estos valores para calcular K_{pc} (límite sensible) y T_{pc} (límite del período). Con estos valores, la función de sintonía automática del controlador PID calcula las ganancias del lazo PID según las ecuaciones de Zeigler-Nichols mostradas abajo:

M = Amplitud de la salida de control = Mmax - Mmin	
Kpc = $4M / (\pi * X_o)$ Tpc = T_o	
Sintonía PID	Sintonía PI
P = $0.45 * K_{pc}$	P = $0.30 * K_{pc}$
I = $0.60 * T_{pc}$	I = $1.00 * T_{pc}$
D = $0.10 * T_{pc}$	D = 0
Período de muestreo = $0.014 * T_{pc}$	Período de muestreo = $0.03 * T_{pc}$

Error de sintonía automática

Si el bit de error de sintonía (bit 13 de la palabra V+06) está encendido, verifique que los valores de la PV y del SP estén dentro del 5% de la diferencia del rango total, según los requisitos de la función. El bit también se activará si el método de lazo cerrado está en uso y la salida se va a los límites del rango.



NOTA: Si la PV fluctúa rápidamente, usted necesita probablemente usar el filtro analógico incorporado (Vea la página 8-57) o cree un filtro en lógica ladder (vea el ejemplo en la página 8-58).

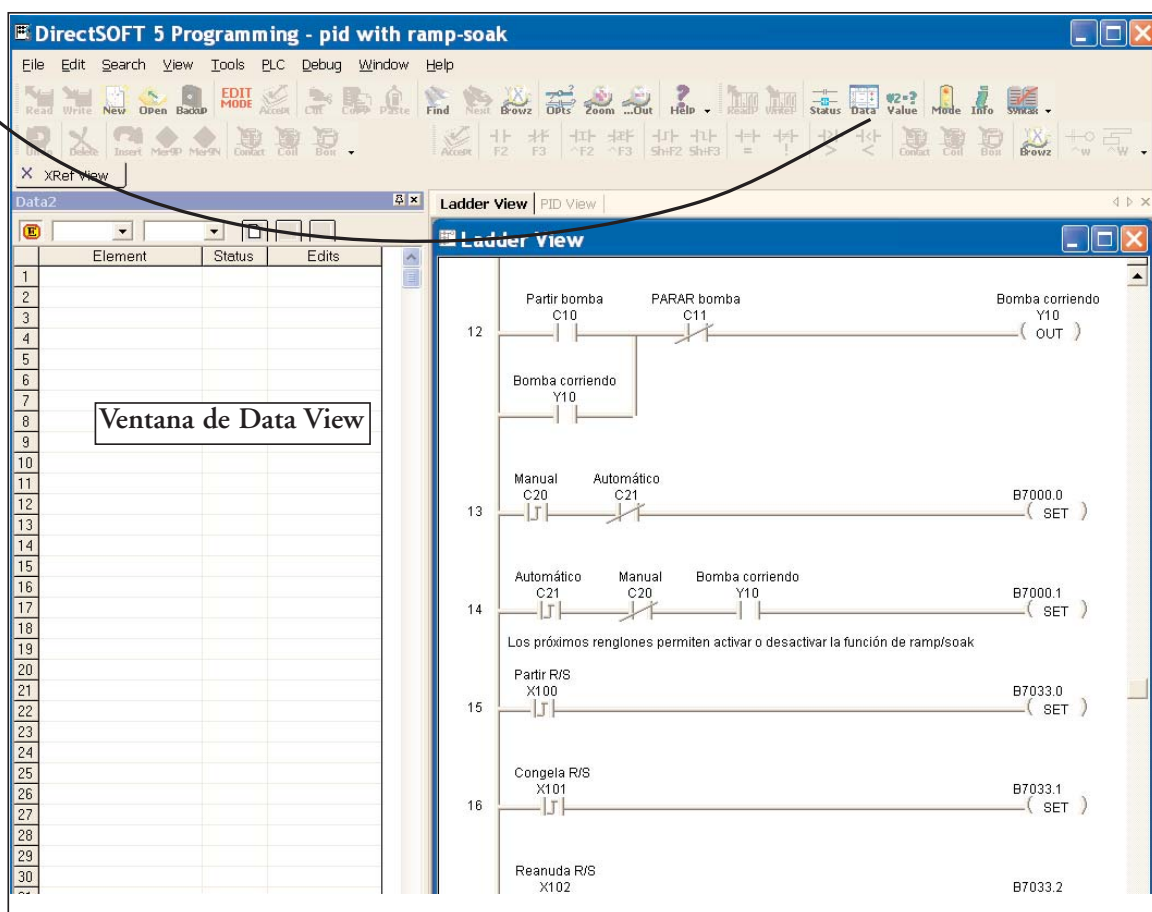
Use Data View de *DirectSOFT* junto con PID View

La ventana Data View es una herramienta muy útil que se puede usar para ayudar a sintonizar el lazo de PID. Usted puede comparar las variables en la ventana PID View con los valores corrientes en la memoria V con Data View.

Abriendo una ventana nueva de Data View

Se puede abrir Data View de tres maneras;

- i) con el menú use **Debug > Data View > New**,
- ii) el atajo del teclado **Ctrl + Shift + F3** o
- iii) el botón **Data** en la barra de herramientas. La ventana Data View es asignada el nombre **Data1** por defecto. Este nombre se puede cambiar en la ventana corriente usando el diálogo de opciones (Options). El diagrama siguiente es un ejemplo de una nueva ventana abierta Data View. La ventana se abrirá al lado de la ventana Ladder View, por defecto.



La ventana Data View puede ser usada como se muestra en la figura siguiente para localizar fallas en la lógica de PID, y puede ser muy útil al sintonizar el lazo de control PID.

Abra la ventana PID View

La ventana de diálogo de Data View puede ser usada para observar datos de memoria.



Element	Status	Edits
1		
2	V7013	100 100
3	V7014	400 400
4	V7015	3900 3900
5	V7016	4000
6	V7017	100 100
7	V7020	160 160
8		
9	V6030	25447.645 18000
10	Volumen [L] real	
11	V7003	3600
12	PV en mm	
13	V7002	3600 3600
14	Referencia nivel	
15	V7005	1978 288
16	CV	
17	V7007	71
18	V7023	50 50
19	V7010	5213 6000
20	V6014	1120 0
21	Consumo L/m	
22	V2001	1000 1000
23	Consumo agua	
24	V6000	1119.91540
25	Fujo bomba L/m	
26	V6020	-0.001409:0

PID View es abierta seleccionándola con el menú View, **View > PID View**, como se muestra en la figura de abajo. PID View también puede ser abierta haciendo clic en el ícono PID View de la barra de íconos. Solamente funciona cuando hay por lo menos un lazo de control definido.

La PID View se abrirá y aparecerá sobre la Ladder View, que se puede llamar haciendo clic en la lengüeta. Al usar la Data View y PID View juntas, cada ventana se puede ajustar según lo mostrado en la figura de abajo.

The PID View window displays a graph with the following parameters:

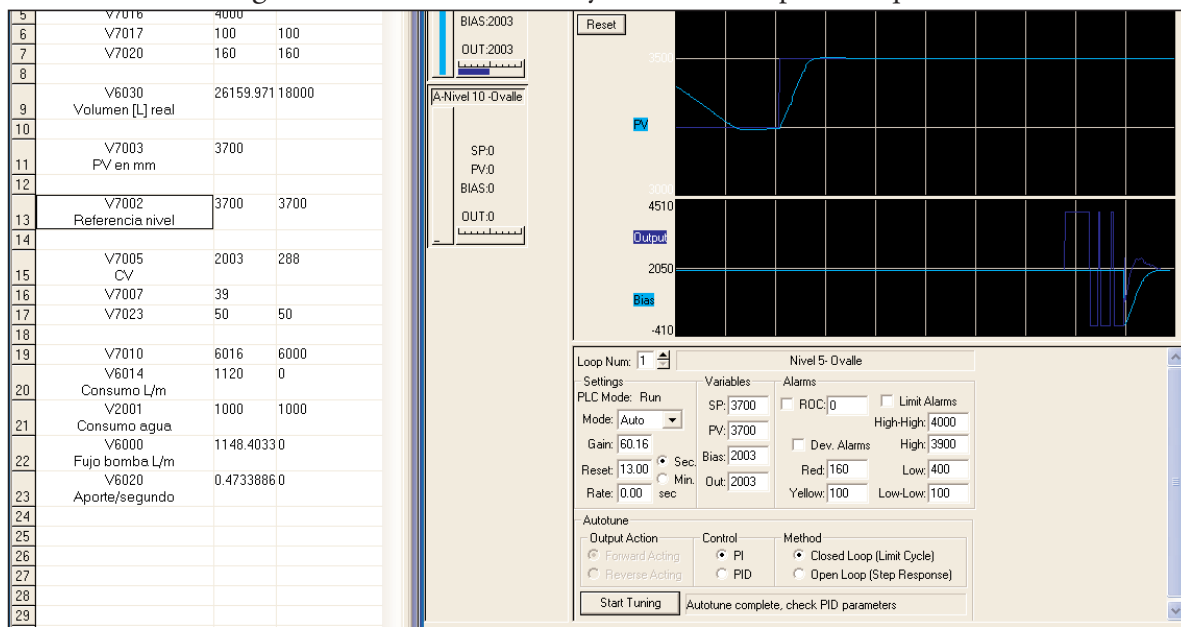
- SP: 3500
- PV: 3500
- BIAS: 1978
- OUT: 1978

The graph shows a step change in the setpoint (SP) from 3500 to 4000, and the process variable (PV) and output (OUT) respond accordingly. The output (OUT) is shown in blue, and the process variable (PV) is shown in red.

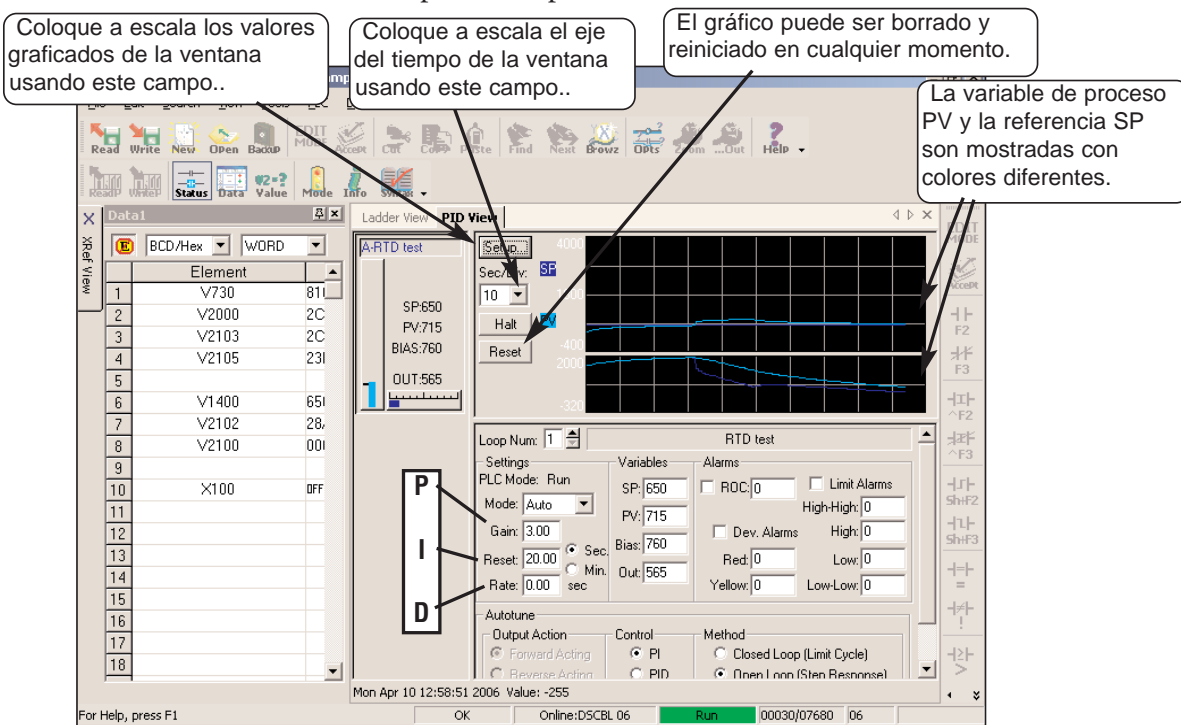
The PID View window also includes a settings panel with the following parameters:

- Loop Num: 1
- Settings:
 - PLC Mode: Run
 - Mode: Auto
 - Gain: 52.13
 - Reset: 23.90
 - Rate: 0.00
- Variables:
 - SP: 3500
 - PV: 3500
 - Bias: 1978
 - Out: 1978
- Alarms:
 - RDC: 0
 - Limit Alarms: High-High: 4000, High: 3900, Low: 400, Low-Low: 100
 - Dev. Alarms: Red: 160, Yellow: 100

Las dos ventanas ahora están listas para ser usadas en la sintonización. Usted podrá ver adonde se han configurado los valores de PID y ver datos del proceso que está controlando.



Con ambas ventanas colocadas de este modo, usted puede ver que valores de PID se han configurado y puede ver el proceso que está controlando. En el diagrama de abajo, usted puede ver la referencia SP, la variable de proceso PV y los valores corrientes de salida, junto con otras direcciones de memoria de la tabla PID. Vea las definiciones de palabras de la tabla PID (página 8-20) para detalles para cada palabra en la tabla. Esto es también un buen modo de encontrar el formato de datos de referencia para cada palabra en la tabla.



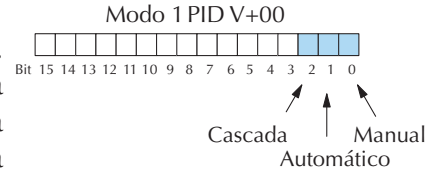
Usando las funciones especiales del control PID

Es una buena idea entender las funciones especiales del DL06 para control PID y cómo usarlas. Usted puede desear incorporar algunas de estas características en su control PID.

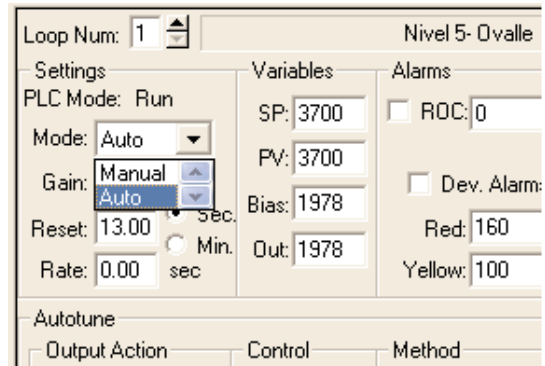
Cómo cambiar modos del lazo desde un programa ladder

Los primeros tres bits de la petición de la palabra V+00 piden al controlador PID cambiarse al modo deseado del lazo correspondiente. Estos bits son peticiones o solicitudes de cambio de modo, no comandos (ciertas condiciones pueden prohibir un cambio de modo particular - vea la página siguiente).

El estado normal de estos bits de petición de modo es "000". Para solicitar un cambio de modo, usted debe usar la instrucción SET para colocar el bit en "1", usando una instrucción One-shot. El controlador del lazo PID cambia automáticamente los bits de nuevo a "000" después de que lea la petición de cambio de modo. Los métodos de solicitar cambios de modo son:



- Con **PID View** de *DirectSOFT*— éste es el método más fácil. Haga clic en la derecha del campo de modos (**mode**); esto hace aparecer una lista de los tres modos. Seleccione la que le corresponda.

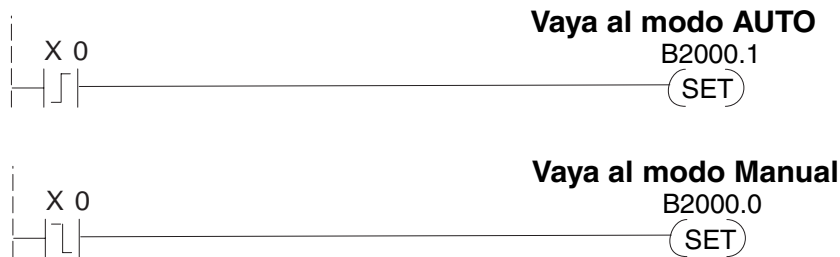


Esto hará cambiar el bit correspondiente en el PLC y aparecerá el modo seleccionado en el campo Mode al finalizar la operación.

- Con **lógica ladder**— El programa puede solicitar cualquier modo del controlador PID cuando el PLC está en modo RUN. Esto será necesario en la mayoría de las aplicaciones.

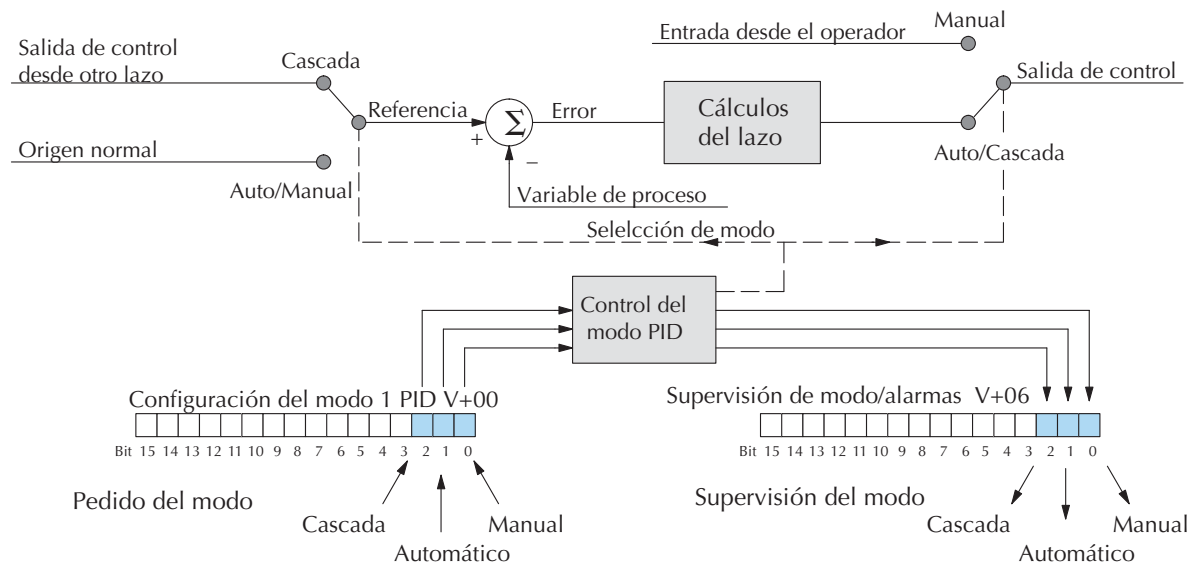
Use el programa mostrado en la figura de abajo con una instrucción SET en el bit de modo (no use una bobina OUT). En una transición 0-1 de X0, B2000.1 se hace 1 para solicitar al PLC que el lazo correspondiente vaya al modo automático. El PLC coloca en cero este bit una vez que ha sido aceptado el cambio de modo.

En una transición 1-0 de X0, B2000.0 se hace 1 para solicitar al PLC que el lazo correspondiente vaya al modo manual. El PLC coloca en cero este bit una vez que ha sido aceptado el cambio de modo.



- **Panel de operador** — Interconecte el panel de operador a la lógica ladder usando métodos estándares, y luego use la lógica de la figura para activar el bit correspondiente.

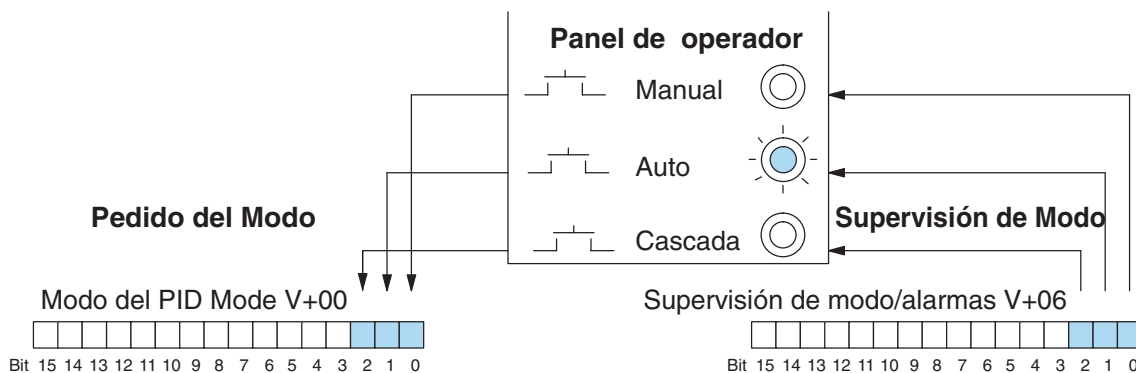
Ya que solamente los cambios de modo pueden ser *solicitados*, el controlador PID decidirá cuando permitir cambios de modo y suministrar el estado del modo del lazo. Se informa el modo corriente en los bits 0, 1, y 2 de la palabra V+06 en la tabla del lazo. Se muestran las funciones de petición y supervisión en la figura de abajo También muestra los dos orígenes de la referencia (SP) dependiendo del modo y los dos posibles orígenes de la salida de control.



Control de los modos de PID con panel de operador

Siendo que los modos Manual, Auto y Cascada son los modos más fundamentales e importantes del lazo de PID, usted puede desear controlar el modo desde un panel de operador. La mayoría de las aplicaciones necesitarán solamente selecciones manuales y auto (cascada se utiliza en usos especiales). Recuerde que los controles del modo son realmente bits de petición de modo, y el modo real del lazo se indica en un bit de la palabra V + 06.

La figura siguiente muestra un panel de operador usando botones momentáneos para solicitar cambios de modo de PID. Los indicadores de modo del panel no se conectan a los conmutadores, sino se conectan a las localizaciones correspondientes de datos.



Efecto de los modos del PLC sobre los modos del controlador PID

Si usted ha seleccionado la opción de que el controlador PID siga el modo del PLC, los modos del PLC (Program, RUN) obran recíprocamente con el controlador como grupo. Lo que sigue resume esta interacción:

- Cuando el PLC está en modo Program, todos los lazos se ponen en modo Manual y no ocurre ningún cálculo del valor de la salida de control. Sin embargo, observe que los módulos de salida (incluyendo salidas analógicas) se apagan en el modo Program del PLC. De modo que el control Manual o es posible cuando el PLC está en modo Program.
- La única vez que la CPU permite un cambio de modo es durante la operación del modo RUN del PLC. Como tal, la CPU registra los modos de los 8 lazos como el modo de operación deseado. Si ocurre una falla de energía durante modo RUN del PLC, la CPU vuelve todos los controladores PID a su modo anterior (que podría ser Manual, Auto, o Cascada).
- En la transición del modo Program a RUN, la CPU fuerza cada controlador PID a volver a su modo anterior registrado durante el último modo RUN del PLC.
- Usted puede agregar y configurar nuevos controladores PID solamente cuando el PLC está en modo Program. Los nuevos controladores PID comienzan automáticamente en modo Manual.

Condiciones de cambio de Modo del controlador PID

En condiciones normales el modo es determinado por la petición a V+00, bits 0, 1 y 2. Sin embargo, existen algunas condiciones que prevendrán un cambio al modo solicitado:

- Un lazo de control que no es independiente del modo del PLC no puede cambiar modos cuando el PLC está en modo Program.
- El lazo mayor de un par conectado en cascada no puede ir de Manual a Auto hasta que el lazo menor esté en modo de cascada.

En otras situaciones, el controlador PID cambiará automáticamente el modo del lazo para mantener una operación segura:

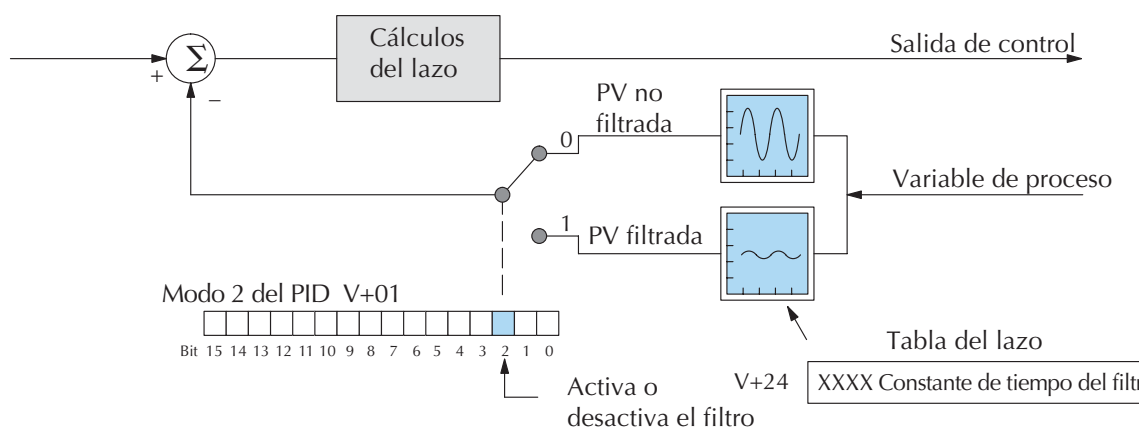
- Un lazo que desarrolla una condición de error automáticamente se va a modo Manual.
- Si el lazo menor de un par conectado en Cascada sale del modo de Cascada por cualquier razón, su lazo importante va automáticamente al modo manual.

Filtro de la variable de proceso PV

Una señal ruidosa de PV puede hacer la sintonía difícil y puede causar que la salida de control sea más alta que lo necesario, cuando la salida trata de responder a picos y valles del PV. Hay varios métodos equivalentes de filtrar la entrada de PV de hacer el lazo más estable. El primer método se logra usando el filtro incorporado del PLC DL06. El segundo método logra un resultado similar usando lógica ladder.

El filtro analógico incorporado en el PLC DL06

El PLC DL06 tiene un filtro pasabajo seleccionable de primer orden en la entrada de PV que puede ser particularmente útil durante la sintonía automática, usando el método de lazo cerrado. **Recomendamos usar un filtro durante la sintonía automática.** Usted puede eliminar el filtro después que la sintonía automática se ha completado, o continúe a usarlo si la señal de la entrada de PV es ruidosa.



El Bit 2 de la palabra Modo de PID 2 habilita / deshabilita el control del filtro para la PV (0=deshabilita del filtro pasa bajo, 1=habilita). La frecuencia de corte del filtro pasa bajo es determinada usando la memoria addr + 24 en la tabla de parámetros de lazos, la constante del filtro. El formato de datos del valor de la constante del filtro es BCD, con una coma decimal implicada 00X,X, como sigue:

- La constante del filtro tiene un rango válido de 000,1 a 001,0.
- *DirectSOFT* convierte los valores encima del rango válido a 001,0 y los valores debajo de este rango a 000,1.
- Un valor de 000,0 o 001,1 a 999,9 esencialmente deshabilita el filtro.
- Valores cerca de 001,0 tienen como resultado una frecuencia de corte más alta, mientras que valores más cerca a 000,1 tienen como resultado frecuencias de cortes bajas.

El algoritmo que se usa en el filtro incorporado es:

$$y_i = k (x_i - y_{i-1}) + y_{i-1}$$

y_i es la salida actual del filtro;

x_i es la entrada actual del filtro

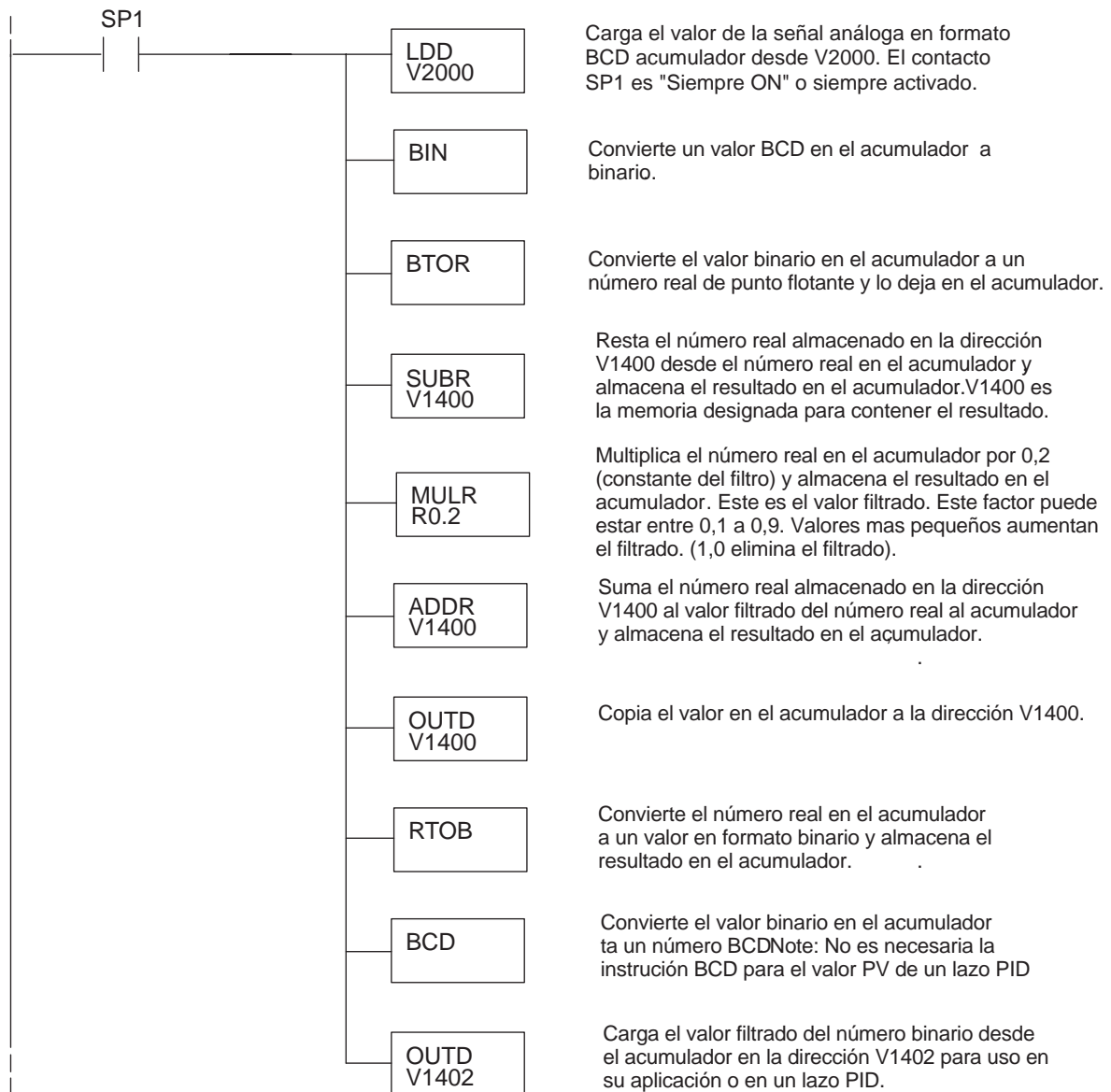
y_{i-1} es la salida previa del filtro

K es el Factor de Filtro de Entrada PV

Creando un filtro analógico en lógica ladder

Se puede construir un algoritmo similar en su programa ladder. Se pueden filtrar eficazmente sus entradas analógicas usando cualquier método. El ejemplo de programación siguiente describe la lógica ladder que usted necesitará. Asegúrese de cambiar las direcciones de memoria del ejemplo a las que Ud. usará en su caso.

El filtrado puede inducir un error de 1 parte en 1000 en su salida debido al "redondeo". Si su proceso no puede tolerar un error de 1 en 1000, no use el filtrado. Debido al error de redondeo, usted no debe usar cero o el valor mayor del rango como puntos de alarmas. Además, cuanto más pequeña es la constante del filtro mayor es el efecto de amortiguamiento, pero más lento es el tiempo de reacción. Asegúrese que una respuesta más lenta es aceptable al controlar su proceso.



Usando la instrucción inteligente IBox del filtro en *DirectSOFT5*

Para los que estén usando *DirectSOFT 5*, se puede usar la instrucción inteligente Ib-402, el filtro en binario (decimal). Este IBox realizará un filtro de primer orden sobre las informaciones en bruto sobre un intervalo definido de tiempo. La ecuación es,

$$\text{Nuevo} = \text{Viejo} + [(\text{Bruto} - \text{Viejo}) / \text{FDC}]$$

donde

Nuevo = Nuevo valor filtrado

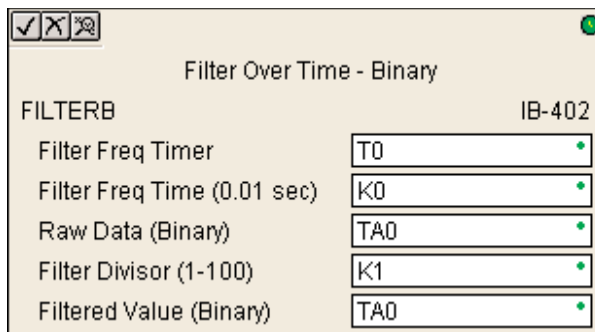
Viejo = Valor filtrado antiguo

FDC = Constante de división del filtro

Bruto = Informaciones en bruto

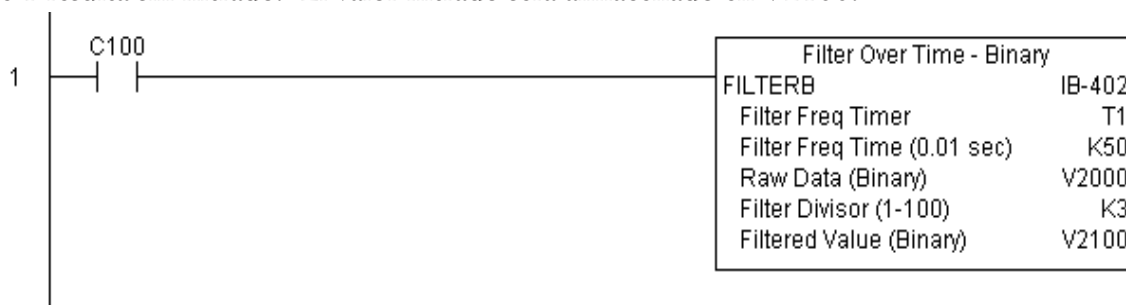
La constante de división del filtro es un número entero en el rango K1 a K100, tal que si igual a K1, no se realiza ningún filtrado. Al aumentar este valor, el filtrado aumenta.

El período en el cual se realiza el cálculo se especifica por tiempo en centésimos de segundo (0,01 segundo) como el parámetro **Filter Freq Time**. Observe que esta instrucción tiene un temporizador que está embutido en el IBox y no se debe usar en ningún otro lugar en su programa. La instrucción puede ser activada o desactivada en el renglón. Si está desactivada, el valor del filtro no es actualizado. En el primer barrido del programa, el valor del filtro se inicializa a 0 para dar al cálculo un punto de partida consistente.



Ejemplo de FilterB

Lo que sigue es un ejemplo de cómo se usa el IBox FilterB en un programa ladder. Se usa la instrucción de filtrar un valor binario que esté en V2000. El temporizador (T1) se configura en 0,5 segundos, el período en el cual será realizado el cálculo del filtro. La constante del filtro se coloca como 3.0. Un valor más grande aumentará el efecto de alisamiento del filtro. Un valor de 1 resulta sin filtrado. El valor filtrado será almacenado en V2100.



Vea el capítulo 5 del manual D0-06USER-MSP, página 242, para una información más detallada.

Por último, también se puede usar un filtro de “promedio rodante”. Vea el documento AN-MISC-023 en el sitio de Internet de AUTOMATION DIRECT en la parte de apoyo técnico.

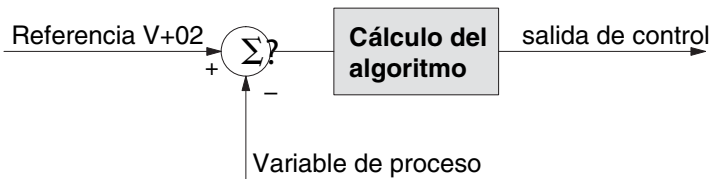
Generador de Ramp/Soak (Rampa y valor constante)

Introducción a Ramp/soak

En la descripción de la operación básica de control de lazos se dijo que la referencia (SP) para un lazo se puede generar de varias maneras, dependiendo del modo que opera el lazo y las preferencias en la programación. En la figura de abajo, el generador de Ramp/Soak (Rampa y valor constante) es una de las maneras en que se puede generar el SP. Es la responsabilidad de su programa ladder de asegurar que ocurre sólo un valor de SP en un determinado momento, en `addr + 02`.

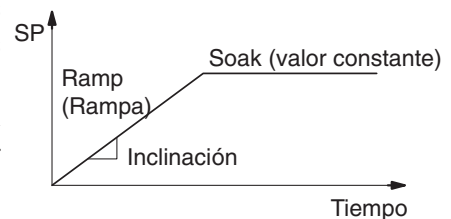
Origen de referencias:

- Por el operador
- Generador Ramp/soak
- Programa ladder
- Otra salida de un lazo (cascada)



Si la referencia SP del proceso cambia muy raramente o puede tolerar cambios escalón, probablemente no necesitará usar el generador de Ramp/Soak. Sin embargo, algunos procesos requieren cambios precisamente controlados del valor de SP. El generador de Ramp/Soak puede reducir la cantidad de programación requerida para estas aplicaciones.

Los términos "Ramp" y "Soak" tienen significados especiales en el área del control de proceso y se refiere a valores deseados de referencia (SP) en aplicaciones de control de temperatura. En la figura a la derecha, la referencia (SP) aumenta durante el segmento de rampa. Permanece estabilizada en un valor durante el segmento soak.



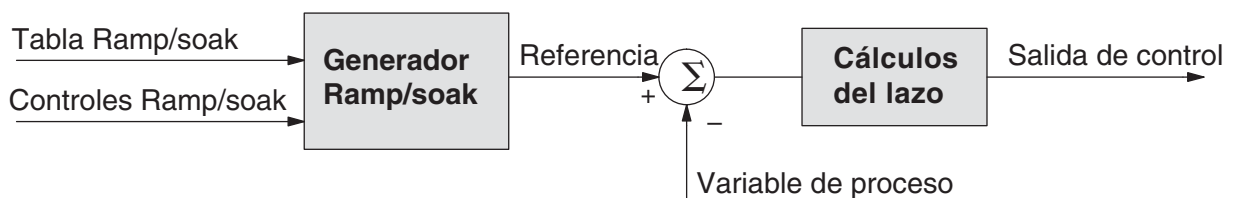
Pueden ser generados perfiles de SP especificando una serie de segmentos Ramp/Soak en una tabla especial.

Los segmentos de rampas se especifican en unidades de SP por segundo. El tiempo de **Soak** es programado en minutos.

Es instructivo ver el generador de Ramp/Soak como una función dedicada para generar los valores de SP, como mostrado abajo. Tiene dos categorías de entradas, que determinan la SP generada.

La tabla Ramp/Soak se debe programar por adelantado, conteniendo los valores que definirán el perfil de la Ramp/Soak. El lazo lee de la tabla durante cada cálculo de PID, como sea necesario.

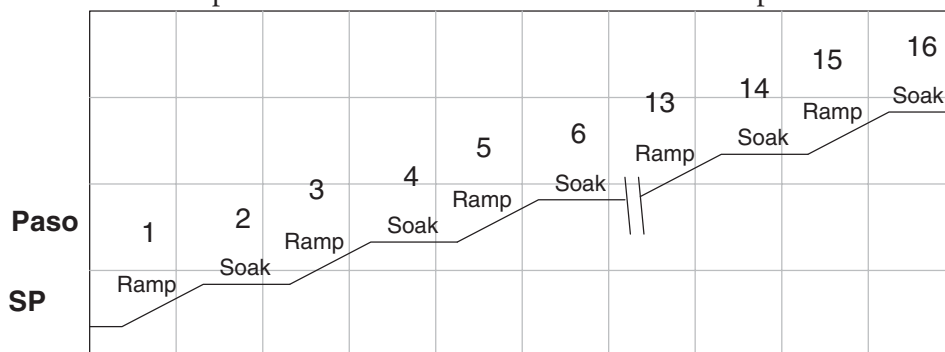
Los controles de la referencia Ramp/Soak son bits en una palabra especial de la tabla de lazos que controla el comienzo y la parada en tiempo real de la funcionalidad del generador de Ramp/Soak. El programa ladder puede supervisar el estado del perfil de la Ramp/Soak (rampa corriente/número del segmento).



Ahora que hemos descrito la operación general del generador de Ramp/Soak listamos sus características específicas:

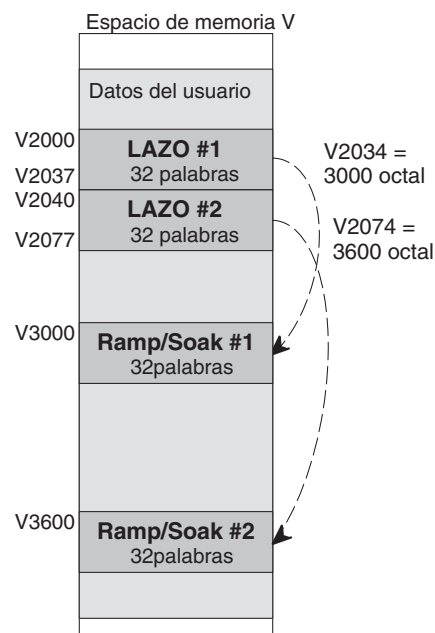
- Cada lazo tiene su propio generador Ramp/Soak (el uso es opcional).
- Usted puede especificar hasta ocho pasos de Ramp/Soak (16 segmentos).
- El generador de ramp/Soak puede funcionar en cualquier momento si el PLC está en el modo RUN. Su operación es independiente del modo del lazo (Manual o Automático).
- Los controles en tiempo real de Ramp/Soak incluyen Start (iniciar), Hold (mantención del valor), Resume (Reanude), y Jog.
- La supervisión de Ramp/Soak incluye Perfil Completado, Desvío (SP menos PV), y número corriente del paso de Ramp/Soak.

La figura siguiente muestra un perfil compuesto de pares de segmentos Ramp/Soak. Los segmentos se numeran individualmente como pasos de 1 a 16. La inclinación de cada una de las rampas puede ser positiva o negativa. El generador de Ramp/Soak (Rampa y valor constante) sabe automáticamente si debe aumentar o disminuir el SP basado en los valores relativos de los valores extremos de la rampa. Estos valores vienen de la tabla de Ramp/Soak .



La Tabla Ramp/Soak

Los parámetros que definen el perfil de Ramp/Soak para un lazo están en la tabla de Ramp/Soak. Cada lazo puede tener su propia tabla Ramp/Soak y es opcional. Recuerde que la tabla de parámetros de lazos consiste de un bloque de 32 palabras de memoria para cada lazo y ocupan una área contigua de memoria. Sin embargo, la tabla Ramp/Soak para un lazo se localiza individualmente porque es opcional para cada lazo. Un puntero de dirección especifica la dirección inicial de la tabla Ramp/Soak, en la dirección **addr + 34** en la tabla de lazos. En el ejemplo a la derecha, las tablas de parámetros de lazos para el Lazo #1 y #2 ocupan 32 palabras en un bloque contiguo, como mostrado. Cada tabla tiene un puntero a la tabla Ramp/Soak, independientemente localizada en otra parte en la memoria. Por supuesto, usted puede localizar todas las tablas en un grupo, obviamente sin superponerlas.



Capítulo 8: Operación de control PID

Los parámetros en la tabla Ramp/Soak (Rampa y valor constante) deben ser definidos por el usuario. La manera más conveniente es usando *DirectSOFT*, que tiene una tabla especial para ser llenada con los valores programados. Se necesitan cuatro parámetros para definir un par de segmentos Ramp/Soak, como mostrado abajo.

- **Valor de fin de rampa** – especifica el valor del destino SP para el fin de la rampa. Use el mismo formato de datos para este número del que usted usa para SP (Formato binario). Puede estar encima o debajo del valor de inicio de SP, de modo que la inclinación podría estar ser hacia arriba o hacia abajo (No es necesario saber el valor de SP que comienza la rampa #1).
- **La inclinación de la rampa** – especifica el aumento de la referencia SP en unidades por segundo. Es un número BCD de 00,00 a 99,99 (usa 2 puntos de decimal).
- **Duración del Soak** – Especifica el tiempo de segmento constante en minutos, en el rango desde 000,1 to 999,9 minutos en BCD (punto decimal implicado).
- **Desvío de PV en Soak** – (opcional) especifica un desvío admisible de la variable de proceso PV encima y debajo del valor de referencia SP durante el período Soak. Es generado un bit del estado de la alarma del desvío de PV por el generador de Ramp/Soak.

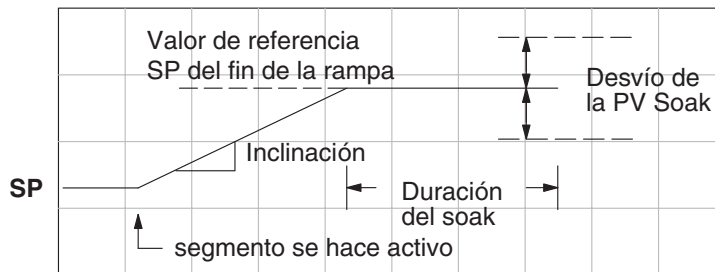


Tabla Ramp/Soak	
V+00	XXXX Valor final SP de rampa
V+01	XXXX Inclinación de rampa
V+02	XXXX Duración del Soak
V+03	XXXX Desvío de PV Soak

El segmento de la rampa se activa cuando el segmento Soak previo termina. Si la rampa es el primer segmento, se activa cuando el generador de Ramp/Soak comienza y asume automáticamente la referencia SP corriente como la referencia SP inicial.

Avance	Paso	Descripción	Avance	Paso	Descripción
+ 00	1	Valor de fin de rampa	+ 20	9	Valor de fin de rampa
+ 01	1	Inclinación de rampa	+ 21	9	Inclinación de rampa
+ 02	2	Duración del Soak	+ 22	10	Duración del Soak
+ 03	2	Desvío de PV en Soak	+ 23	10	Desvío de PV en Soak
+ 04	3	Valor de fin de rampa	+ 24	11	Valor de fin de rampa
+ 05	3	Inclinación de rampa	+ 25	11	Inclinación de rampa
+ 06	4	Duración del Soak	+ 26	12	Duración del Soak
+ 07	4	Desvío de PV en Soak	+ 27	12	Desvío de PV en Soak
+ 10	5	Valor de fin de rampa	+ 30	13	Valor de fin de rampa
+ 11	5	Inclinación de rampa	+ 31	13	Inclinación de rampa
+ 12	6	Duración del Soak	+ 32	14	Duración del Soak
+ 13	6	Desvío de PV en Soak	+ 33	14	Desvío de PV en Soak
+ 14	7	Valor de fin de rampa	+ 34	15	Valor de fin de rampa
+ 15	7	Inclinación de rampa	+ 35	15	Inclinación de rampa
+ 16	8	Duración del Soak	+ 36	16	Duración del Soak
+ 17	8	Desvío de PV en Soak	+ 37	16	Desvío de PV en Soak

Muchas aplicaciones no requieren todos los 16 pasos de Ramp/ Soak. Use 0's en la tabla para pasos no usados. El generador finaliza el perfil cuando encuentra una rampa = 0

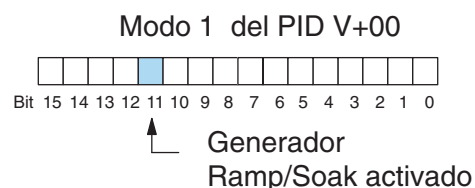
Los indicadores de bit de la tabla Ramp/Soak

Las definiciones de los bits de la palabra de la Tabla Ramp/Soak, Addr+33, están en la lista en la tabla siguiente:

Bit	Descripción del bit	Lee/escrive	Bit=0	Bit=1
0	Partir perfil Ramp / Soak	escribe	–	Parte en transición 0-1
1	Congelar perfil Ramp / Soak	escribe	–	Congela en transición 0-1
2	Reanudar perfil Ramp / soak	escribe	–	Reanuda en transición 0-1
3	Jog del perfil Ramp / Soak	escribe	–	Jog en transición 0-1
4	Perfil Ramp / Soak completado	Lee	–	Perfil completado
5	Desvío de la entrada PV en Ramp / Soak	Lee	Off	On
6	Perfil Ramp / Soak congelado	Lee	Off	On
7	Reservado	Lee	Off	On
8–15	Paso corriente en perfil R/S	Lee	Decodifique como byte (hex)	

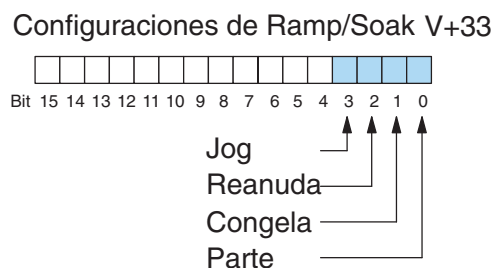
Activando el generador de Ramp/Soak

El principal control de activación para permitir la generación del valor de referencia SP con Ramp/Soak se hace con el bit 11 de la palabra Modo de PID 1 addr + 00, como mostrado a la derecha. Los otros controles de la Ramp/Soak en el addr + 33 mostrado en la tabla encima no funcionarán a menos que este bit sea 1 durante el proceso de Ramp/ Soak.



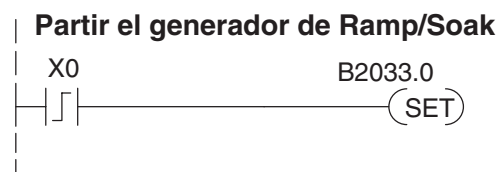
Controles de la Ramp/Soak

Los cuatro controles principales para el generador de Ramp/Soak están en los bits 0 a 3 de la palabra de configuración de la Ramp/Soak en la tabla de parámetros de lazos. *DirectSOFT* controla estos bits directamente desde el diálogo de configuración de la Ramp/Soak. Sin embargo, usted debe usar la lógica ladder para controlar estos bits durante la ejecución del programa. Recomendamos usar la instrucción bit de palabra (Bit-of Word).



La lógica ladder debe colocar el bit de control en "1" para ordenar la función correspondiente. Cuando el controlador de lazo lee el valor de la Ramp/Soak, apaga automáticamente el bit. Por lo tanto, no es necesario ejecutar una instrucción de Reset del bit, cuando la CPU está en el modo RUN.

El programa ejemplo a la derecha muestra cómo un interruptor X0 externo puede prender y el contacto de PD usa la transición de 0 para 1 para poner el bit apropiado de control para comenzar el perfil Ramp/Soak. Esto usa la instrucción Bit-of-Word.



El estado normal de los bits de control para la Ramp/Soak son todos ceros. La lógica ladder debe poner sólo un bit de control a la vez.

- **Start** – una transición 0 a 1 comenzará a ejecutar el perfil. La CPU debe estar en el modo RUN y el lazo puede estar en el modo Manual o Automático. Si el perfil no es interrumpido por una orden de Hold o Jog, termina normalmente.
- **Hold** – una transición 0 a 1 parará el perfil de Ramp/Soak (Rampa y valor constante) en su estado actual, y el valor de SP se congelará.
- **Resume** – Una transición 0 a 1 causa que el generador de Ramp/Soak reanude la operación si está en el estado Hold. Los valores de SP comienzan desde el valor previo.
- **Jog** – Una transición 0 a 1 causará que el generador de Ramp/Soak (Rampa y valor constante) pare la ejecución del segmento (paso) actual, y vaya al próximo segmento.

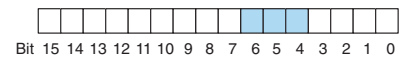
Supervisión del perfil de Ramp/Soak

Usted puede supervisar el estado del perfil de la Ramp/Soak usando otros bits en la palabra de configuración de Ramp/Soak **addr + 33**, como mostrado a la derecha.

- El Perfil R/ S Completado – Es igual a 1 cuando se termina el ultimo paso programado.
- Desvío de PV – Es igual a 1 cuando el error (SP-PV) excede el desvío especificado en la tabla de R/S.
- El Perfil esta en Hold – Es igual a 1 cuando el perfil estaba activo pero ahora está en Hold.

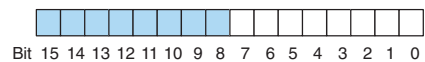
El número del paso actual está disponible en los 8 bits superiores de la palabra Configuración de Ramp/Soak **addr + 33**. Los bits representan un número de 2 dígitos hexadecimales, en el rango de 1 a 10. La lógica ladder puede supervisar éstos bits para sincronizar otras partes del programa con el perfil de la Ramp/Soak. Cargue esta palabra al acumulador y haga un right shift de 8 bits y se obtiene el número del paso.

Valores para Ramp/Soak V+33



Perfil R/S congelado
Desvío de PV en Soak
Perfil R/S completado

Valores para Ramp/Soak V+33



Paso corriente del perfil,
hexadecimal de 2 dígitos

Valor = 01 a 10 hexadecimal
o 1 a 16 decimal

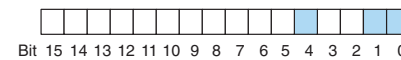
Errores de programación de Ramp/Soak

La dirección inicial para la tabla Ramp/Soak debe ser una dirección válida. Si la dirección apunta fuera del rango de memoria de usuario V, uno de los bits a la derecha prenderá cuando comienza el generador de Ramp/Soak. Recomendamos usar *DirectSOFT* para configurar la tabla Ramp/Soak. Verifica las direcciones automáticamente para usted.

Probando el perfil de Ramp/Soak

Es muy recomendable probar el perfil de Ramp/Soak antes de usarlo para controlar el proceso. Esto es fácil de hacer, porque el generador de Ramp/Soak correrá aún cuando el lazo está en modo Manual. Usando el PID View de *DirectSOFT* será un ahorro de tiempo porque dibujará el perfil en la pantalla para usted. Asegúrese de colocar la base de tiempo del grafico lo suficiente lento para mostrar por lo menos un par de rampas/ segmentos en el grafico.

Error de tabla Ramp/Soak V+35



Dirección inicial en rango
de memoria V reservada

Dirección inicial fuera del
rango superior de memoria V

Dirección inicial fuera del
rango inferior de memoria V

Ejemplo de Ramp/Soak con *DirectSOFT*

El ejemplo siguiente le muestra como configurar la función Ramp/Soak.

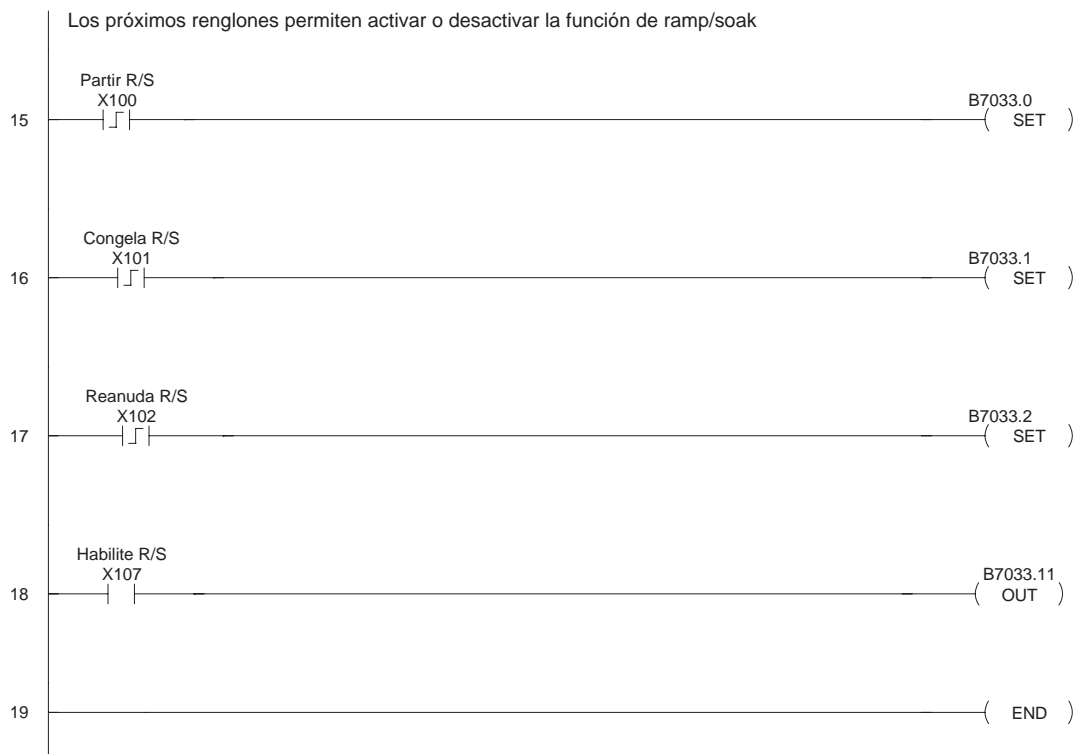
Configure el perfil de ramp/soak en el diálogo PID Setup

El primer paso es usar Setup PID en *DirectSOFT* para configurar el perfil de la referencia del proceso. Abra la ventana Setup PID y seleccione la lengüeta R/S, y luego entre los datos de rampa y soak.

	Ramp		Soak	
	SP	Slope	Time	Deviation
1	3000	1.00	2.0	200
2	3200	2.00	1.0	100
3	3400	3.00	1.0	100
4	3000	1.50	2.0	100
5	0	0.00	0.0	0
6	0	0.00	0.0	0

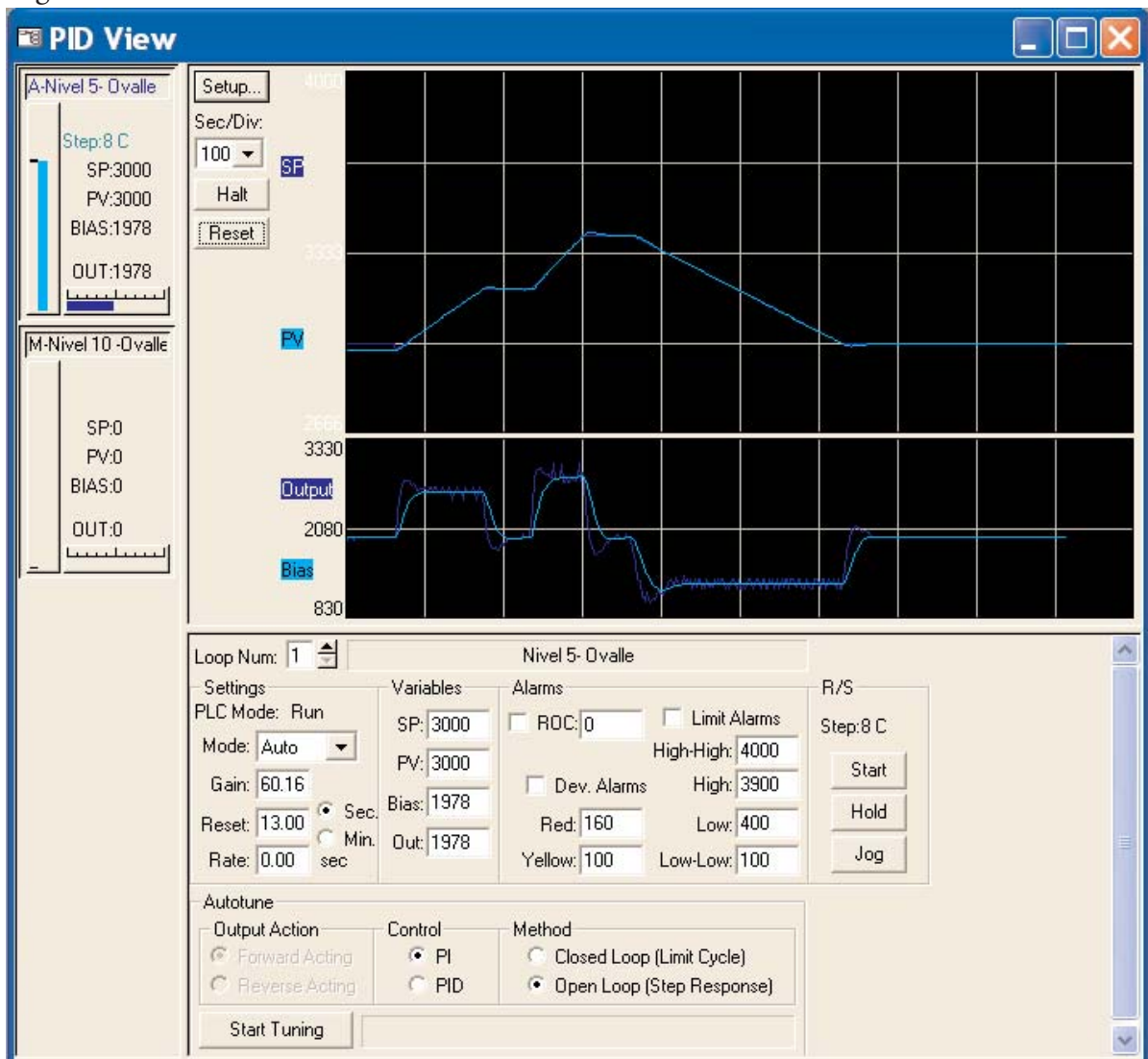
Programando el control de Ramp/Soak en lógica ladder

Vea la tabla de descripción de los bits de Ramp/Soak en la página 8-62 para agregar los renglones de control a su programa similar a los renglones de la lógica de abajo. La tabla de PID comienza en V7000.



Pruebe el perfil de ejemplo de ramp/soak

Pruebe el comportamiento de la función Ramp/Soak usando PID View, va a resultar algo como sigue.



Al activar X100 comienza el perfil de ramp /soak,

Note como la referencia sube en la primera rampa, desde el valor 3000 hasta 3200. Esta rampa tien una unidad por segundom, de modo que son 200 segundos.

En 3200 se queda por dos minutos, lo que es 120 segundos

Luego continúa hasta 3400 con una rampa de 100 segundos.

En 3400 se queda por un minuto, lo que es 60 segundos

Luego baja hasta 3000 con una rampa negativa de 1.5 unidades poe segundo.

Vea la curva obtenida. Note que la curva es la de la referencia. La variable de proceso PV trata de seguir esa curva, pero no siempre se obtiene lo mismo.

Control en cascada

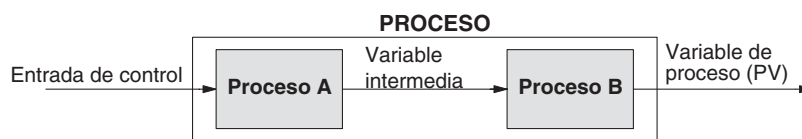
Introducción a lazos en cascada

Los lazos en cascada son una técnica avanzada de control que es superior al control individual de lazo en ciertas situaciones. Como el nombre implica, en cascada significa que un lazo es conectado a otro lazo. Además de los modos Manual (lazo abierto) y Automático (lazo cerrado), el PLC DL06 también puede suministrar el modo en cascada.



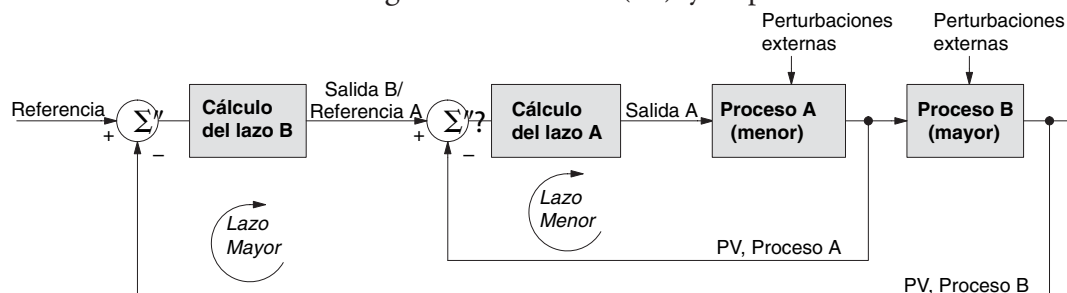
NOTA: Los lazos en cascada son una técnica avanzada del control de procesos. Por lo tanto recomendamos su uso sólo para ingenieros experimentados de control de procesos.

Cuándo un proceso de fabricación es complejo y contiene un atraso de tiempo en la entrada de control para procesar la variable de salida, aún un lazo perfectamente sintonizado alrededor de proceso puede hacer el control lento e inexacto. Puede ser que los actuadores operen en una propiedad física que afecta eventualmente la variable de proceso, medido por una propiedad física diferente. Identificar la variable intermedia nos permite dividir el proceso en dos partes, como es mostrado en la figura siguiente.



¡Un lazo en cascada es un sistema en que simplemente agregamos otro lazo en el proceso para controlar la variable intermedia más precisamente! Esto también separa la fuente del atraso del control en dos partes.

El esquema debajo muestra un sistema del control de cascada, mostrando que es simplemente un lazo anidado dentro de otro. El lazo interno se llama **lazo menor** y el lazo externo se llama **lazo mayor**. Para obtener estabilidad, el lazo menor debe ser el lazo que responde más rápido en los dos. Hay que agregar el sensor adicional para medir la variable intermedia (PV para el proceso A). Note que la referencia (SP) para el lazo menor se genera automáticamente usando la salida del lazo mayor. Una vez que el control en cascada es programado y depurado sólo necesitamos tratar con la variable original de referencia (SP) y de proceso en el nivel de sistema.



Los lazos en cascada se comportan como un lazo, pero con el desempeño mejorado sobre la solución anterior de lazo simple.

Uno de los beneficios del control en cascada puede ser vistos examinando su respuesta a perturbaciones externas. Recuerde que el lazo secundario es de actuación más rápida que el lazo mayor. Por lo tanto, si una perturbación afecta el proceso A en el lazo menor, el cálculo de PID en el lazo A puede corregir el error resultante antes que el lazo mayor vea el efecto.

Los lazos en cascada en el PLC DL06

En el uso del término de "lazos en cascada", debemos hacer una distinción importante. Sólo el lazo menor estará verdaderamente en el modo en cascada. En la operación normal, el lazo mayor debe estar en modo automático. Si usted tiene más de dos lazos en cascada JUNTOS, el lazo más exterior (mayor) debe estar en el modo automático durante la operación normal y todos los lazos interiores en el modo en Cascada.

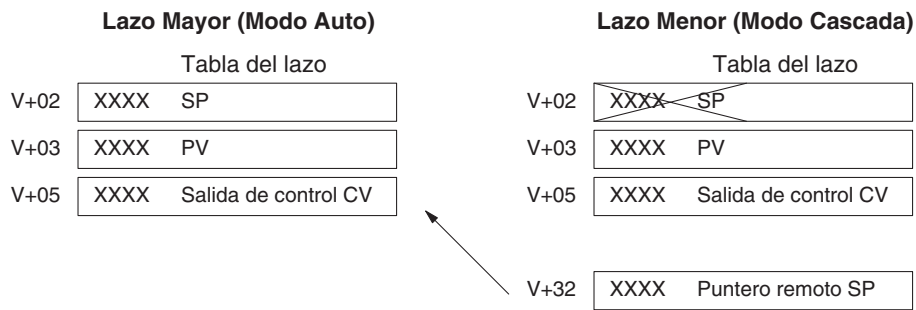


NOTA: *Técnicamente, ambos lazos mayor y menor están "en cascada" en la terminología estricta de control de procesos. Desgraciadamente, no podemos retener esta convención cuando se controlan los modos del lazo. Recuerde que todo lazo menor estará en el modo en Cascada, y sólo el lazo más exterior (mayor) estará en el modo automático.*

Usted puede colocar en cascada tantos lazos como sea necesario en el DL06 y puede tener múltiples grupos de lazos en cascada. Para una operación apropiada de los lazos en cascada se debe usar el mismo rango de datos (12/15 bits) y ser unipolares/bipolares en el lazo mayor y menor.

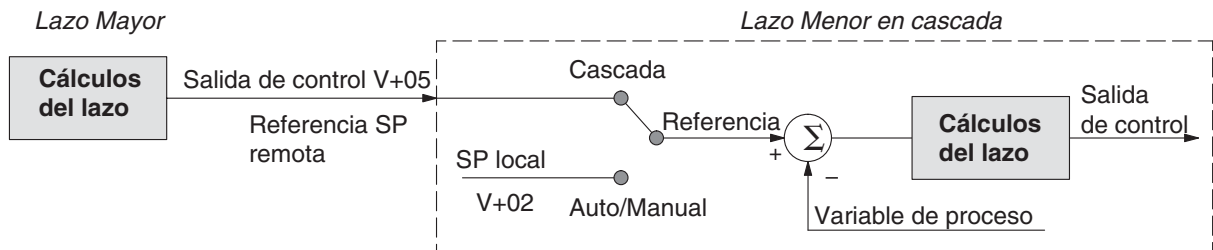
Para preparar un lazo para la operación del modo en cascada como un lazo menor, usted debe programar el puntero remoto de referencia (SP) en la dirección de la tabla de parámetros de lazos **addr + 32**, como mostrado abajo. El puntero debe ser la dirección **addr + 05** (la salida de control) del lazo mayor. En el modo en cascada, el lazo menor ignorará el valor de SP local (**addr+02**) y leerá entonces la salida de control del lazo mayor como la referencia SP.

8



Cuando se usa PID View de *DirectSOFT* para mirar el valor de SP del lazo menor, *DirectSOFT* lee automáticamente la salida de control del lazo mayor y lo lleva a la referencia SP del lazo menor. La dirección normal de SP del lazo menor, **addr + 02**, permanece igual.

Ahora, usamos la configuración de parámetros de lazos encima y dibujamos el diagrama equivalente, mostrado abajo.



Recuerde que el lazo mayor va al modo Manual automáticamente si su lazo menor es sacado del modo en cascada.

Sintonizando lazos en cascada

Al sintonizar lazos en cascada necesitaremos desconectar la relación en cascada y sintonizar los lazos individualmente, usando uno de los procedimientos de sintonía de lazo descritos previamente.

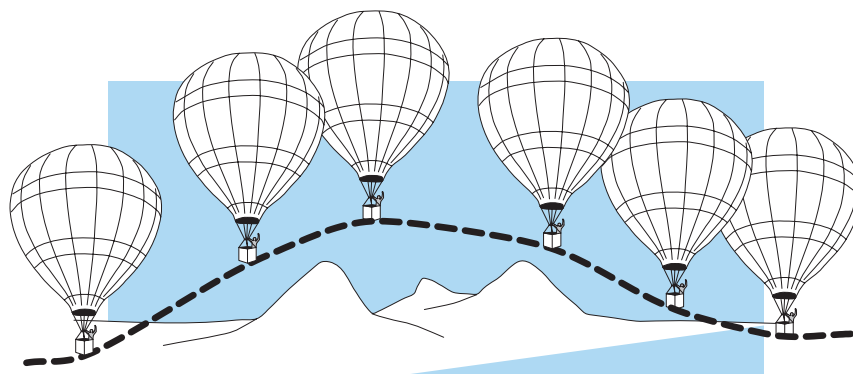
1. Si usted no usa la sintonía automática, entonces encuentre el período de muestreo para el lazo menor, usando el método discutido previamente en este capítulo. Luego coloque el período de muestreo del lazo mayor más lento que el lazo menor por un factor de 10. Use este valor como punto de partida.
2. Sintonice el lazo menor primero. Deje el lazo mayor en modo Manual, y necesitará generar los cambios de SP para el lazo menor manualmente como descrito en el procedimiento de la sintonía del lazo.
3. Verifique que el lazo menor dé una respuesta críticamente amortiguada con un 10% de cambio de SP en el Modo automático. Esto termina la sintonía el lazo menor.
4. En este paso necesitará colocar el lazo Menor en el modo en Cascada y el lazo Mayor en el modo automático. Sintonizaremos el lazo mayor con el lazo menor como un componente en serie al proceso completo. Por lo tanto, no vuelva y sintonice el lazo menor otra vez mientras hace la sintonía del lazo mayor.
5. Afine el lazo mayor, siguiendo el procedimiento normal de la sintonía del lazo en esta sección. La respuesta del lazo mayor PV es la respuesta completa de los lazos en cascada juntos.

Control de pulsos con amplitud modulada (PWM)

El controlador de lazos de PID en el PLC DL06 genera una señal similar a una señal analógica de la salida de control en un rango numérico. El valor de la salida de control es adecuado para manejar un módulo analógico de salida que es conectado al proceso. En el campo de control de proceso, esto se llama control continuo o analógico, porque la salida está en algún nivel continuamente.

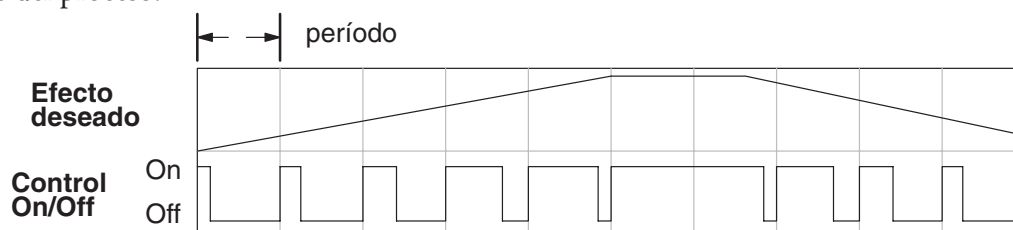
Aunque el control continuo puede ser suave y robusto, el costo de los componentes de control (tales como actuadores, controladores de alta corriente) pueden ser caros. Una forma simplificada de control se llama control de pulsos de amplitud modulada (o PWM). Este método usa actuadores discretos ON-OFF que están activados o desactivados (sin valores intermedios). Los componentes para sistemas discretos de control son de costo más bajo que los componentes de trabajo continuo de control.

En esta sección, le mostraremos cómo convertir la salida de control de un lazo a control de pulsos de amplitud modulada (PWM) para aplicaciones que necesitan de este tipo de control. Veamos como prentiendo y apagando alternadamente una carga se puede controlar un proceso. El esquema de abajo muestra un globo de aire caliente que sube siguiendo un trayecto a través de algunas montañas.



El trayecto deseado es la *referencia* (SP). El piloto del globo prende y apaga el quemador alternadamente, lo que es la *salida de control* CV. La gran masa de aire en el globo promedia efectivamente el efecto del quemador, convirtiendo los chorros de calor en un efecto continuo: cambia lentamente la temperatura del globo y últimamente la altitud, que es la *variable de proceso* PV.

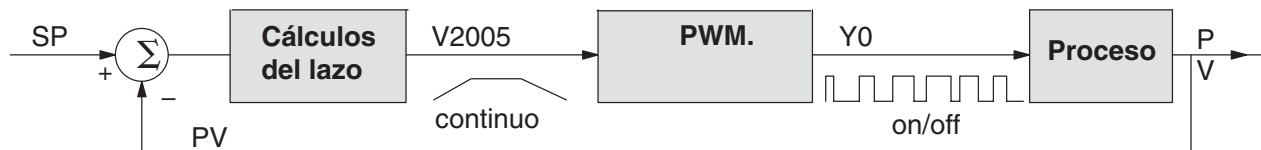
El control de pulsos de amplitud modulada se aproxima al control continuo en virtud de su ciclo, la razón del tiempo con el quemador funcionando y no funcionando. La figura siguiente muestra un ejemplo de cómo el ciclo se aproxima a un nivel continuo cuando es promediado por una masa grande del proceso.



Si hiciéramos un gráfico de los tiempos del quemador activado o no en el globo de aire caliente, veríamos probablemente una relación muy semejante a su efecto en la temperatura del globo y altitud.

Ejemplo de programa de control PWM

El segmento siguiente ladder proporciona un control de tiempo ON/OFF en una salida. Convierte la salida continua en V2005 a un control discreto usando la bobina de salida Y0.



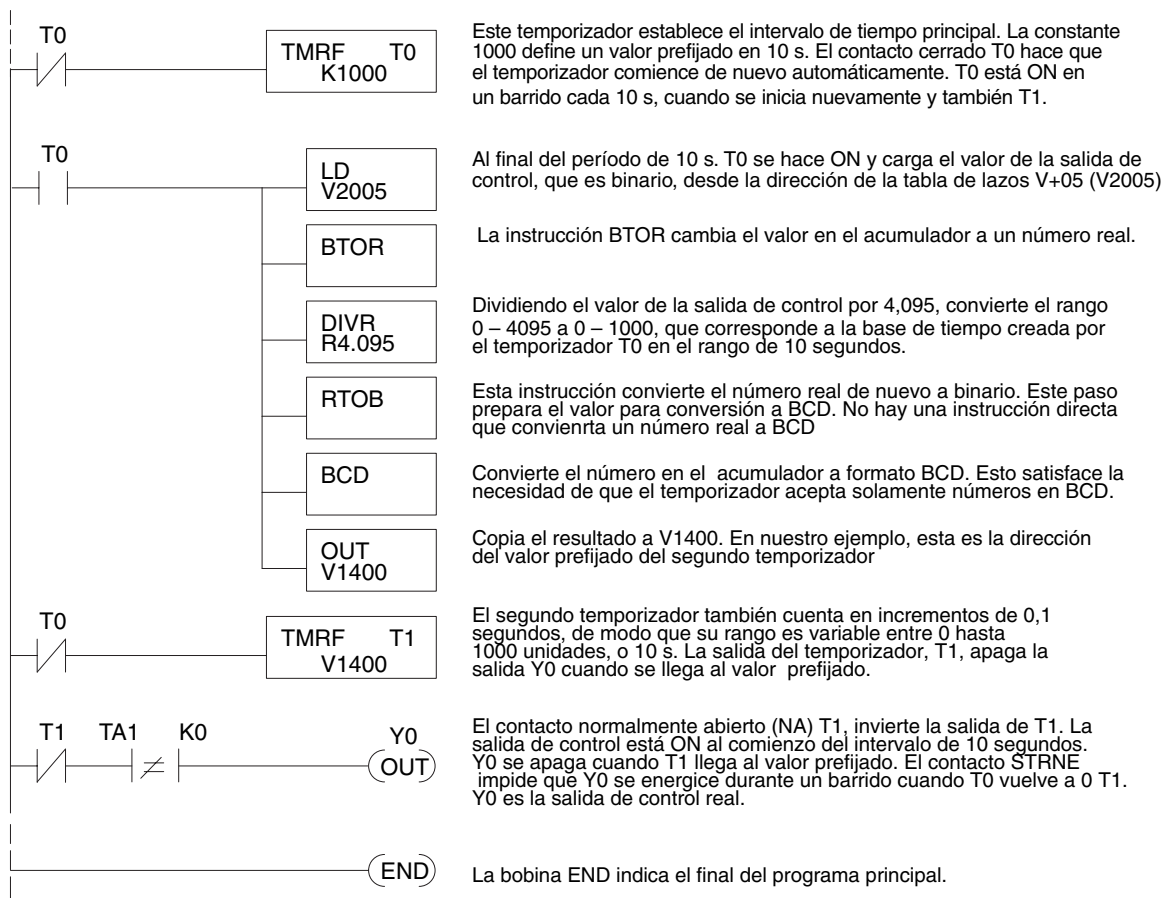
El programa ejemplo usa dos temporizadores para generar el control ON/OFF. Hace las suposiciones siguientes, que se pueden modificar para adaptarlas a su aplicación:

- La tabla del lazo PID comienza en V2000, de modo que la salida de control está en V2005.
- El formato de datos de la salida de control es de 12 bits, unipolar (0 - FFF) o 0-4095.
- La salida de control ON/OFF es Y0.

El programa PWM debe hacer similar la resolución de la salida (1 en 1000) a la resolución de la base de tiempo de T0 (también 1 en 1000).

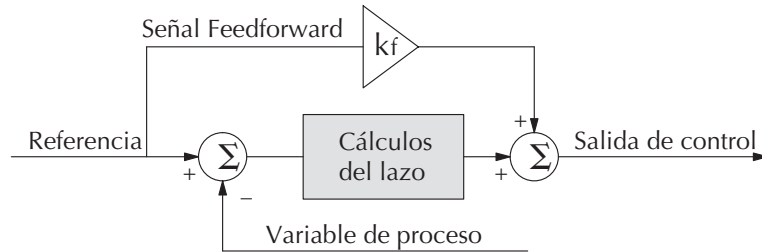


NOTA: Algunos procesos cambian demasiado rápidamente para este control de tiempo. Considere la velocidad de su proceso cuando usted escoge este método de control. Use el control continuo para los procesos que cambian demasiado rápido para control de tipo PWM.



El Control feedforward

El control feedforward es una mejora al control de lazo cerrado. Es muy útil para disminuir los efectos de una perturbación *cuantificable y previsible* en un lazo o del cambio repentino en la señal de referencia SP. El uso de esta característica es una opción disponible en el PLC DL06. Sin embargo, es mejor aplicar y afinar un lazo sin feedforward y sólo agregarlo si se necesita aún un mejor desempeño del control del lazo. El término "feedforward" se refiere a la técnica de control mostrada en el esquema adyacente. El valor entrante de referencia se suma con la salida alrededor de la ecuación de PID, como se muestra en el diagrama de la figura de abajo.



En una sección anterior en el término de la tendencia (bias), dijimos que "el valor del término de la tendencia establece una región de trabajo" o un punto de operación para la salida de control. *Cuándo el error fluctúa alrededor de cero, la salida fluctúa alrededor del valor de tendencia (Bias).* Ahora, cuando hay un cambio en la referencia, se genera un error y la salida debe cambiar a un nuevo punto de operación. Esto sucede también si una perturbación introduce un desvío nuevo de la variable de proceso en el lazo. El lazo no "sabe ir" al nuevo punto de operación... la corrección de tendencia (o bias) debe incrementar/decrementar hasta que el error desaparezca y entonces la tendencia (bias) ha encontrado el nuevo punto de operación.

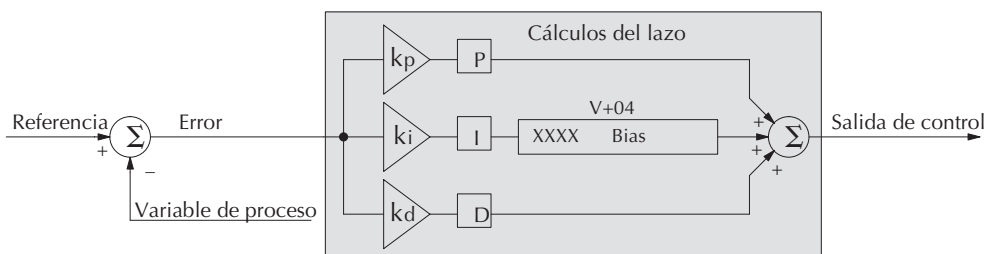
Suponga que somos capaces de saber que un cambio repentino de referencia (SP) está a punto de ocurrir (común en algunas aplicaciones). Podemos evitar gran parte del error resultante en primer lugar, si podemos cambiar rápidamente la salida al nuevo punto de operación. Si sabemos (de pruebas anteriores) a qué nuevo punto de operación (el valor de la tendencia) se llegará después del cambio de la referencia (SP), podemos cambiar artificialmente la salida directamente (que es feedforward).

Los beneficios de usar feedforward son:

- El error de SP-PV se reduce durante cambios previsibles de referencia (SP) o perturbaciones.
- El uso apropiado de feedforward permitirá que reduzcamos la ganancia integral. Reduciendo la ganancia integral nos da un sistema estable de control.

Feedforward es muy fácil de usar en el controlador de lazos DL06, como mostrado abajo. El término de tendencia está disponible al usuario en una dirección especial de lectura/escritura, en la dirección de la tabla de Parámetros PID **addr + 04**.

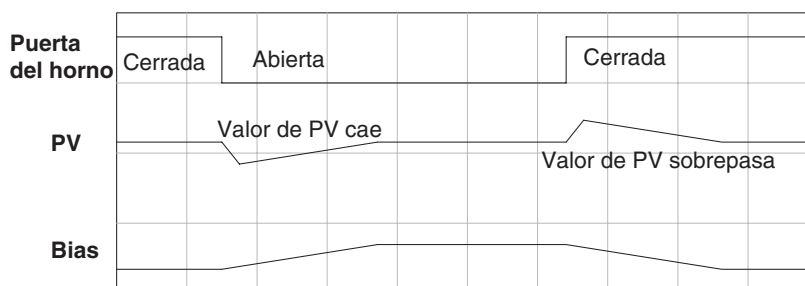
Dirección de la tabla de parámetros V+04.



Para cambiar la tendencia (el punto de operación), la lógica ladder sólo tiene que escribir el valor deseado a **addr + 04**. El cálculo del lazo de PID lee primero el valor de la tendencia (bias) desde el **addr + 04** y modifica el valor basado en el cálculo actual de integración. Luego escribe el resultado de vuelta a la dirección **addr + 04**. Esta configuración crea un tipo "transparente" del valor de la tendencia. Todo lo que tiene que hacer para aplicar control feedforward es escribir el valor correcto a la tendencia en el tiempo correcto (el ejemplo de abajo muestra cómo hacerlo).



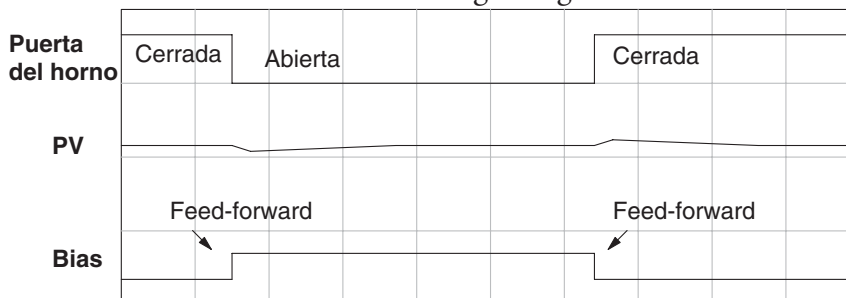
NOTA: *Cuándo escriba a la tendencia, se debe tener cuidado para diseñar la lógica ladder para escribir el valor sólo una vez, actualmente cuando deba ocurrir el punto de operación con la nueva tendencia. Si la lógica ladder escribe el valor de la tendencia en cada barrido, el integrador del lazo se incapacita efectivamente.*



Ejemplo de feedforward

¿Cómo sabemos cuando escribir al término de la tendencia, y qué valor debe escribirse? Suponga que tenemos un lazo de control de temperatura de un horno, y hemos sintonizado ya el lazo para óptimo desempeño. Vea la figura de abajo. Note que cuando el operario abre la puerta del horno, la temperatura baja un poco mientras la tendencia del lazo ajusta a la pérdida de calor. Luego, cuando la puerta cierra, la temperatura sobrepasa la referencia SP hasta que el lazo se ajuste otra vez. El control de Feedforward puede ayudar a disminuir este efecto.

Primero, registramos la cantidad de cambio de tendencia que se genera por el controlador cuando la puerta abre o cierra. Luego, escribimos un programa ladder para detectar el estado de un interruptor límite de la puerta del horno. Cuándo la puerta se abre, el programa ladder lee el valor actual de la tendencia desde **addr + 04**, agrega la cantidad deseada de cambio, y lo escribe de vuelta a la memoria **addr + 04**. Cuándo la puerta cierra, duplicamos el procedimiento, pero ahora restamos la cantidad deseada de cambio. La figura siguiente muestra los resultados.



Los cambios en la tendencia son el resultado de escrituras de valor feedforward al valor de la tendencia. Podemos ver que las variaciones de PV se reducen. La misma técnica puede ser aplicada para cambios en la referencia (SP).

Ejemplo de programa ladder con control PID

Programa ladder asociado a un lazo de control PID

Después de que el o los lazos de control PID, han sido configurados con *DirectSOFT*, se necesitará modificar el programa ladder para incluir los renglones necesarios para configurar el módulo analógico de E/S que se usará como interface al proceso.

El programa siguiente de ejemplo muestra cómo se usa y configura un módulo de medición de temperatura con RTD, F0-04RTD, y un módulo analógico de combinación, F0-4AD2DA-2 pasa un lazo de control PID. Este ejemplo asume que la tabla de PID para el lazo 1 tiene una dirección inicial en V2100.

Por lo tanto las principales variables en este lazo son:

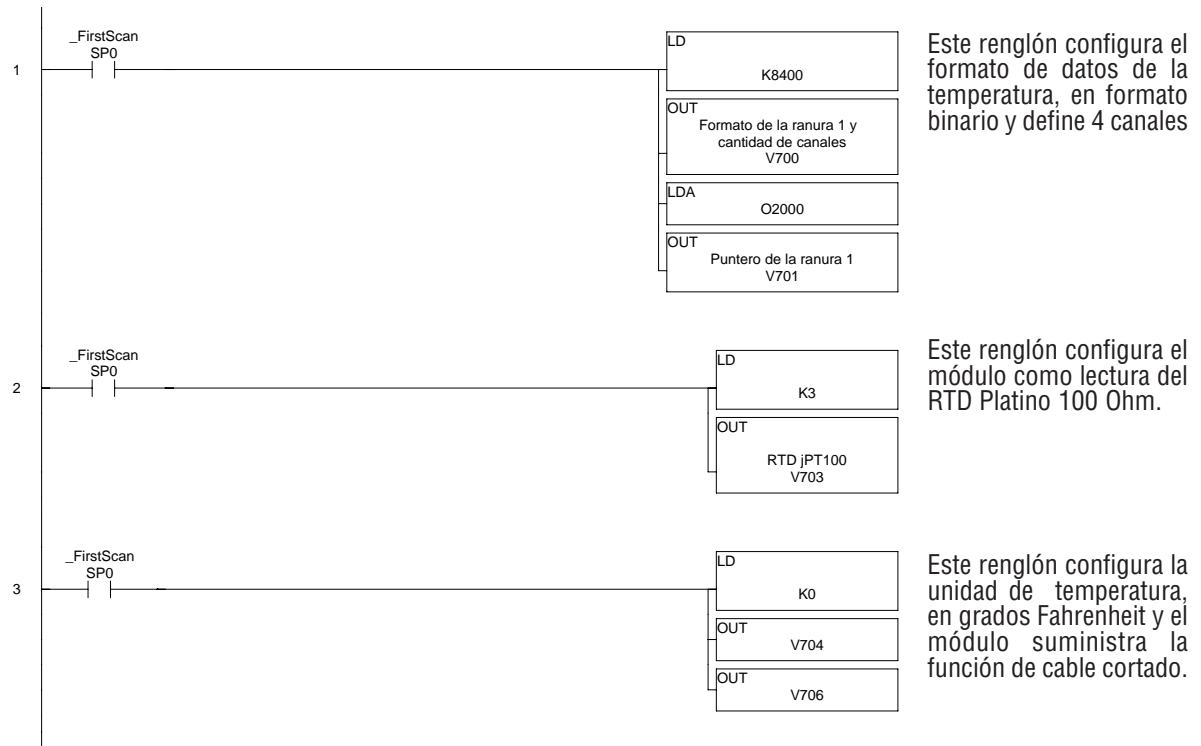
[V2102] = Referencia SP

[V2103] = Variable de proceso PV

[V2105] = Salida de control CV

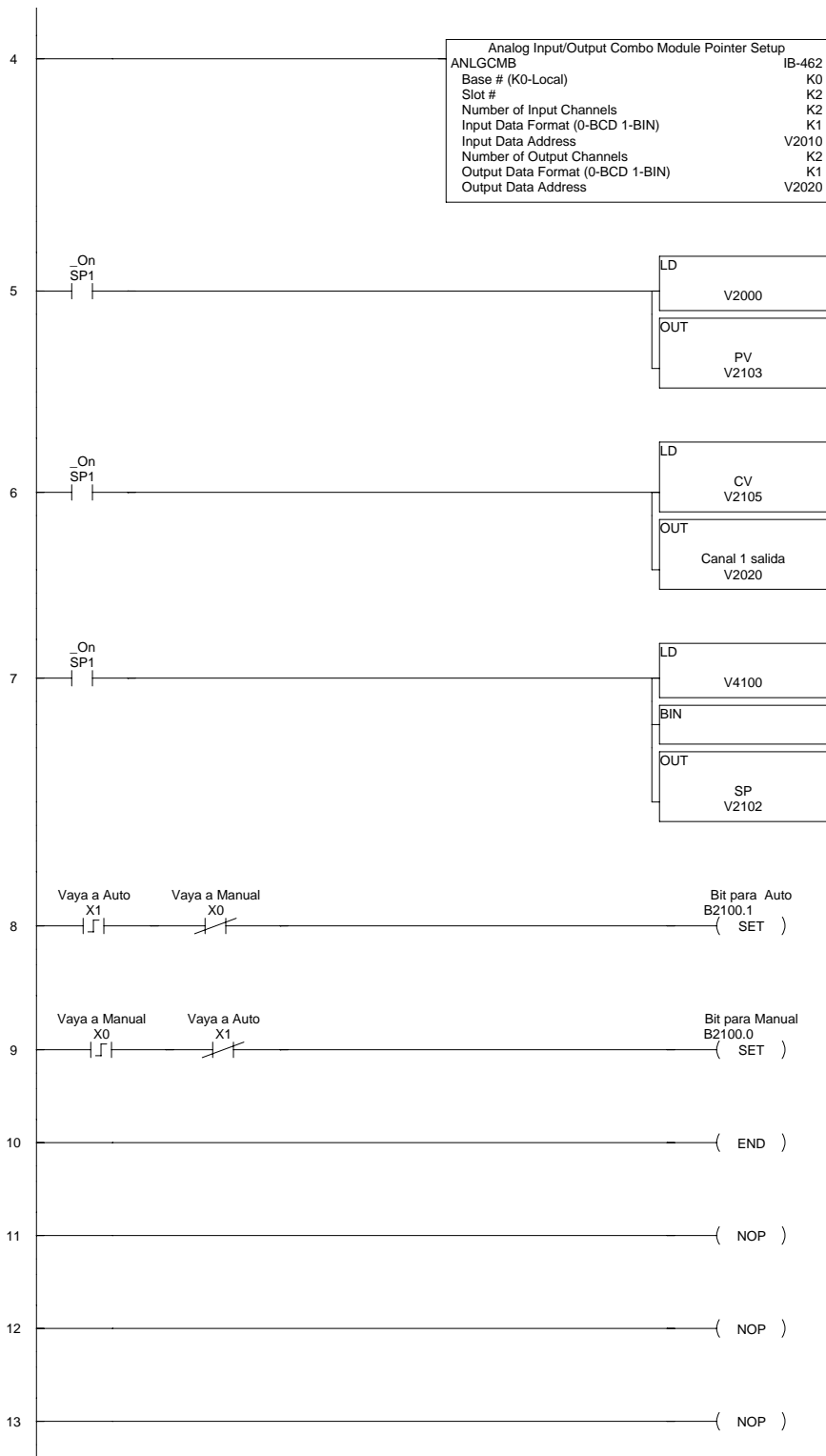
Todos los módulos analógicos de entradas y salidas usados con el PLC DL06 son configurados de una manera similar, con *DirectSOFT*. *DirectSOFT5* ofrece Iboxes para configuración de módulos analógicos, que hacen esta acción aún mas simple.

8



Programa continuado en la próxima página

Programa continuado desde la página anterior



Este renglón configura el módulo de 2 entradas y 2 salidas en la ranura 2.

Se colocan las entradas en las memorias V2010 y V2011 y las salidas en V2020 y V2021

Este renglón copia en contenido de la memoria V2000, la variable de proceso o temperatura, en V2103, en la tabla de PID

Este renglón copia en contenido de la memoria V2105, la salida de control, en V2020

Este renglón copia en contenido de la memoria V4100, la referencia de temperatura, en V2102

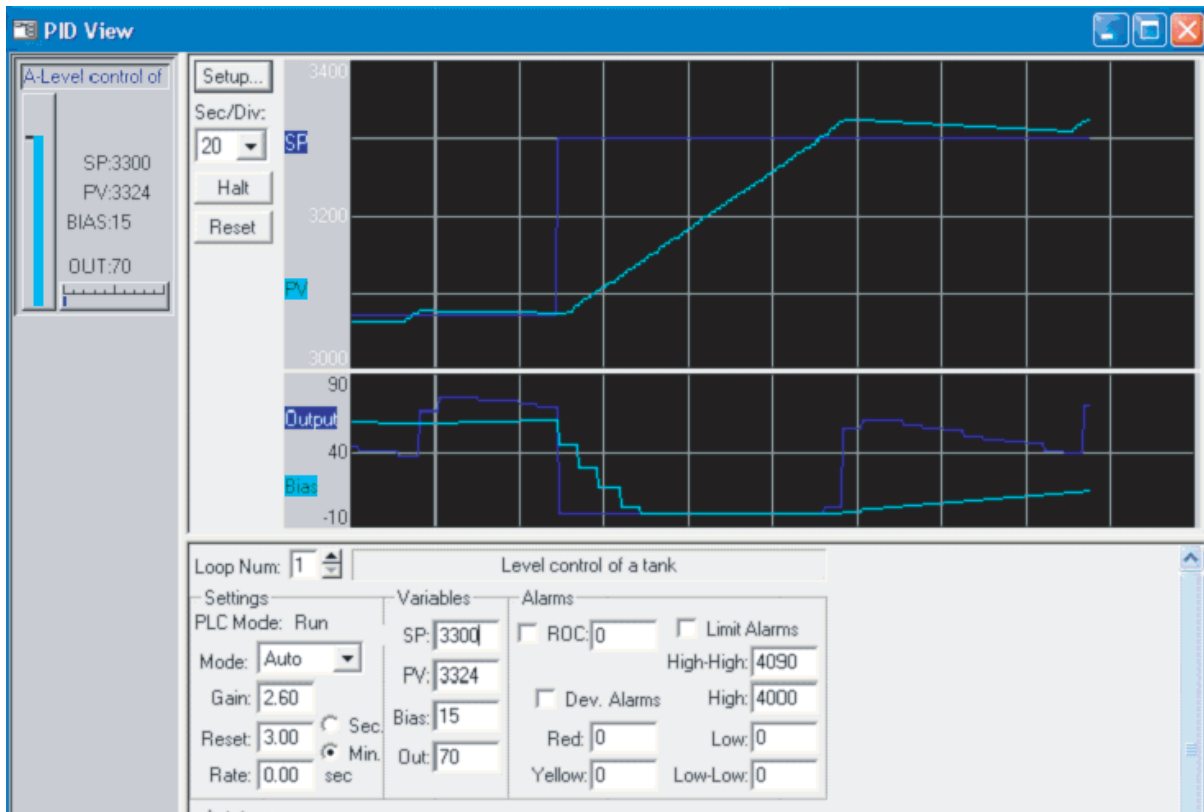
Este renglón selecciona el modo del lazo como Auto

Este renglón selecciona el modo del lazo como Manual

Capítulo 8: Operación de control PID

Observe que los módulos usados en el ejemplo de control PID fueron configurados en formato binario (o decimal en *DirectSOFT*). Este formato podría haber sido configurado como BCD. En este caso, los datos BCD deberían haber sido convertidos a formato binario antes de almacenar estos datos en las memorias de la tabla PID correspondientes a la referencia (SP) y la variable de proceso (PV) y la salida de control debería ser convertida desde formato binario a BCD antes de almacenar el valor en la memoria del módulo de salida analógica.

La ventana PID View se verá de la siguiente forma después que se haya sintonizado el lazo de control. Note como un escalón de referencia es seguido por la variable de proceso.



Consejos para búsqueda de fallas en PID

Pregunta: El lazo no entra en modo Automático.

Respuesta: Verifique lo siguiente para posibles causas:

- Hay una alarma de PV o un error de programación de alarma de PV.
- El lazo es el lazo mayor de un par en cascada y el lazo menor no está en el modo de Cascada.

Pregunta: La salida de control CV permanece en cero constantemente cuando el lazo está en el modo automático.

Respuesta: Verifique lo siguiente para posibles causas:

- El límite superior de la Salida de Control en la dirección de la tabla de lazos **addr + 31** es cero.
- El lazo se va a saturación, porque el error nunca va a cero y cambia el signo del valor (algebraico).

Pregunta: El valor de la Salida de Control no es cero, pero no es correcto.

Respuesta: Verifique lo siguiente para posibles causas:

- Los valores de ganancia se han entrado impropriadamente. Recuerde, las ganancias se entran en la tabla del lazo en BCD, mientras el SP y PV están en binario. Si usted usa *DirectSOFT*, muestra la referencia SP, la variable de proceso PV, Bias (tendencia) y salida de control en decimal (BCD), convirtiéndolo a binario antes de actualizar la tabla de lazos.

Pregunta: El generador de Ramp/Soak no opera cuando se activa el bit Start.

Respuesta: Verifique lo siguiente para posibles causas:

- El bit de habilitación de la Ramp/Soak está apagado. Verifique el estado del bit 11 de la dirección de la tabla de parámetros de lazos **addr + 00**. Debe ser configurado igual a 1.
- El bit HOLD u otros bits en el control de la Ramp/Soak está ON.
- Los valores de inicio del SP y el primer valor de fin de rampa SP son los mismos, así que el primer segmento de rampa no tiene inclinación y no tiene consecuentemente duración. El generador de Ramp/Soak se mueve rápidamente al segmento SOAK, dando la ilusión que la primera rampa no trabaja.
- El lazo está en el modo en Cascada, y trata de obtener el SP remotamente.
- El valor superior límite de SP en la dirección de tabla de lazos **addr + 27** es muy bajo.
- Verifique su programa ladder para verificar que no está escribiendo al SP (la dirección en la tabla del lazo **addr+02**). Una manera rápida de hacer esto será colocar temporalmente una instrucción END al comienzo del programa y luego vaya al modo RUN y comience manualmente el generador Ramp/Soak.

Pregunta: El valor de PV en la tabla es constante, aunque el módulo analógico recibe una señal de PV.

Respuesta: Su programa ladder debe leer el valor analógico del módulo correctamente y lo escribe en la dirección de la tabla de lazos **addr + 03**. Verifique que el módulo analógico genera el valor y el código ladder trabaja.

Pregunta: La ganancia derivativa no parece tener ningún afecto en la salida.

Respuesta. El límite derivativo probablemente está habilitado y entonces no hay ningún efecto en la salida (vea la sección de limitación de ganancia derivativa)

Pregunta: La referencia (SP) aparece estar cambiando por sí misma.

Respuesta: Verifique lo siguiente para posibles causas:

- El generador de Ramp/Soak está habilitado y está generando una referencia (SP).
- Si este síntoma ocurre en cambios de modo de Manual a Automático, el lazo pone automáticamente el SP = PV (característica de transferencia sin saltos [Bumpless mode]).
- Verifique su programa ladder para verificar que no está escribiendo al valor de referencia SP (en la dirección de la tabla de lazos **addr+02**). Una manera rápida de hacer esto es colocar temporalmente una instrucción END al comienzo de su programa y luego vaya al modo RUN.

Pregunta: El SP y PV que he entrado en *DirectSOFT* trabaja bien pero estos valores no trabajan adecuadamente cuando el programa ladder escribe los datos.

Respuesta: El PID View en *DirectSOFT* entra SP, PV y valores de Tendencia en valor decimal y los muestra en decimal para su conveniencia. Por ejemplo, cuando el formato de datos es unipolar, 12 bits, el rango de valores es de 0 a 4095. Sin embargo, la tabla de lazos requiere éstos en hexadecimal, de modo que *DirectSOFT* los convierte para usted. Los valores en el rango de la tabla van de 0 a FFF, para el formato de 12 bits unipolar.

Pregunta: El lazo parece inestable e imposible de sintonizar, con ninguna ganancia trabaja.

Respuesta: Verifique lo siguiente para posibles causas:

- El período de muestreo del lazo esta ajustado muy grande. Vea la sección de este capítulo para escoger el tiempo de actualización del lazo.
- Las ganancias son muy altas. Empiece reduciendo la ganancia derivativa a cero. Luego reduzca la ganancia integral y la ganancia proporcional si fuera necesario.
- Hay mucho atraso en su proceso. Esto significa que el PV reacciona muy lento para controlar los cambios de la salida. Puede haber demasiada "distancia" entre el actuador y el sensor de PV, o el actuador puede ser débil en la transferencia de energía al proceso.
- Puede haber una perturbación del proceso que está exigiendo demás al lazo de control. Asegúrese que la variable de proceso PV es relativamente constante cuando la referencia SP está estable.

Glosario de la terminología de lazos de control PID

Modo automático: un modo operacional de un lazo, en el cual hace cálculos de PID y pone al día la salida del control del lazo.

Congelamiento de Bias: Método de preservar el valor bias (punto de funcionamiento) para una salida del control, inhibiendo el integrador cuando la salida se va fuera del rango. La ventaja es una recuperación más rápida del lazo.

Término de Bias o tendencia: En la forma de la posición de la ecuación de PID, es la suma del integrador y del valor inicial de la salida del control.

Bumpless Transfer (Transferencia sin saltos): Método de cambiar el modo de la operación de un lazo evitando un cambio repentino en la salida de control. Esta consecuencia es evitada artificialmente haciendo el SP y el PV iguales, o el término bias y la salida de control en el momento del cambio de modo.

Lazos de Cascada: Un lazo conectado en cascada recibe su valor de referencia de la salida de control de otro lazo. Los lazos conectados en cascada tienen una relación mayor/menor, y trabajan juntos para controlar una variable de proceso PV.

Modo en cascada: Un modo operacional de un lazo, en el cual recibe su SP desde la salida de otro lazo.

Control continuo: Control de un proceso hecho entregando una señal (analógica) como salida de control.

Lazo de acción directa: Lazo en el cual el PV aumenta en respuesta a un aumento de la salida de control. Es decir, el proceso tiene una ganancia positiva.

Error: La diferencia en valor entre el SP y el PV, es decir, $\text{Error} = \text{SP} - \text{PV}$.

Banda muerta de error: una característica opcional que hace el lazo insensible a los errores cuando son pequeños. Usted puede especificar el tamaño de la banda muerta.

Error al cuadrado: Una característica opcional que multiplica el error por sí mismo, pero conserva la muestra algebraica original. Reduce el efecto de errores pequeños, mientras que magnifica el efecto de errores grandes.

Feedforward Método de optimizar la respuesta del control de un lazo cuando un cambio en la compensación de la referencia o de la perturbación es conocido y tiene un efecto cuantificable en el término bias.

Salida de control : El resultado numérico de una ecuación PID que es enviada por el lazo con la intención de anular el error actual.

Ganancia derivativa : Constante que determina la magnitud del término derivativo PID en respuesta al error actual.

Ganancia integral (Reset): Constante que determina la magnitud del término integral PID en respuesta al error actual.

Lazo mayor : En el control en cascada, es el lazo que genera una referencia para el lazo conectado en cascada.

Modo Manual: un modo operacional de un lazo, el cual los cálculos de PID se paran. El operador debe controlar manualmente el lazo escribiendo al valor de la salida del control directamente.

Lazo menor: en control de cascada, el lazo de menor importancia es el lazo subordinado que recibe su SP del lazo principal.

Control PWM : Método simple de controlar un proceso, encendiendo y apagando la energía en el sistema. La masa del proceso tiene el efecto de suavizar y hacer un promedio al PV. Un programa simple ladder puede convertir el lazo continuo del PLC DL06 a control PWM.

Lazo de control PID: Método matemático de control a circuito cerrado envolviendo la suma de tres términos basados en valores proporcionales, integrales, y derivativos del error. Los tres términos tienen ganancias independientes, permitiendo que una optimice (afine o sintonice) el lazo para un sistema físico particular.

Algoritmo de posición: Se calcula la salida de control de modo que responda al desvío del PV desde la referencia SP (error).

Proceso: El procedimiento de fabricación que agrega valor a las materias primas. El control de proceso se refiere particularmente a inducir cambios químicos al material en proceso.

Variable de proceso (PV): Una medida cuantitativa de la variable (PV) de una característica física del material en el proceso, que afecta la calidad del producto final y es importante supervisar y controlar.

Ganancia proporcional: Constante que determina la magnitud del término proporcional PID en respuesta al error actual.

Alarma absoluta de PV: Alarma programable que compara el valor del PV a valores límites de alarma.

Alarma de desvío de PV: Alarma programable que compara la diferencia entre los valores de SP y PV a un valor de umbral de desvío.

Perfil Ramp / Soak: un sistema de valores del SP llamados un perfil, que se genera en tiempo real sobre cada cálculo del lazo. El perfil consiste en una serie de rampa y empapa los pares del segmento, simplificando grandemente la tarea de programar el PLC para generar tales secuencias del SP.

Tasa: diferenciador también llamado, el término de la tarifa responde a los cambios en el término del error.

Referencia remota : la dirección en donde un lazo lee su valor de referencia (setpoint) cuando se configura mientras que el lazo de menor importancia en una topología conectada en cascada del lazo.

Reset: también llamado integral, el término de reajuste que suma cada error muestreado al anterior, manteniendo un total de funcionamiento llamado bias.

Reset Windup: Una condición creada cuando el lazo no puede encontrar equilibrio y el error persistente hace la suma integral crecer excesivamente (saturación). Reset wind up causa una recuperación adicional retrasada cuando se remedia la falla original del lazo.

Lazo de acción reversa o inversa: Lazo en el cual el PV aumenta en respuesta a una disminución de la salida de control. Es decir el proceso tiene una ganancia negativa.

Tiempo de muestreo : El tiempo entre los cálculos del lazo PID. El método de control de proceso de la CPU se llama un controlador de muestreo, porque muestrea el SP y el PV solamente periódicamente..

Valor de referencia (SP) El valor deseado para la variable de proceso. El valor de referencia (SP) es el comando de la entrada al controlador del lazo durante la operación del lazo cerrado.

Soak Deviation: Es una medida de la diferencia entre la SP y la PV durante un segmento de Soak del perfil Ramp/Soak, cuando el generador R/S está activo.

Respuesta escalón: El comportamiento de la variable de proceso en respuesta a un cambio escalón en la SP (en la operación del lazo cerrado), o un cambio escalón en la salida de control (en la operación de lazo abierto).

Transferencia: El cambio a partir de un modo operacional del lazo a otro (entre manual, automático, o en cascada). La palabra "transferencia" se refiere a la transferencia de control de la salida de control o de la SP, dependiendo del cambio de modo particular.

Algoritmo de velocidad: La salida de control se calcula para representar el índice del cambio (velocidad) para que la PV llegue a ser igual a la SP.

Bibliografía

Fundamentals of Process Control Theory, 3a. Edición Autor: Paul W. Murrill Publicado por: Instrument Society of America ISBN 1-55617-297-4	Application Concepts of Process Control Autor: Paul W. Murrill Publicado por: Instrument Society of America ISBN 1-55617-080-7
PID Controllers: Theory, Design, and Tuning, 2a. Edición Autor: K. Astrom y T Hagglund Publicado por: Instrument Society of America ISBN 1-55617-516-7	Fundamentals of Temperature, Pressure, and Flow Measurements, 3a. Edición Autor: Robert P. Benedict Publicado por: John Wiley and Sons ISBN 0-471-89383-8
Process/Industrial Instruments & Controls Handbook, 4a. Edición Autor (Editor-in-Chief): Douglas M. Considine Publicado por: McGraw-Hill, Inc ISBN 0-07-012445-0	pH Measurement and Control Autor: Gregory K. McMillan Publicado por: Instrument Society of America ISBN 1-55617-483-7
Instrument Engineer's Handbook, Volumen 2: Process Control, 3a. Edición Autor (Editor-in-Chief): Bela G. Liptak Publicado por: Chilton . . . ISBN 0-8019-8242-1	Instrument Engineer's Handbook, Volumen 1: Process Measurement, 3a. Edición Autor (Editor-in-Chief): Bela G. Liptak Publicado por: Chilton . . . ISBN 0-8019-8197-2

MANTENCIÓN Y BÚSQUEDA DE PROBLEMAS



En este capítulo...

Mantencción del sistema de hardware	9-2
Diagnósticos	9-2
Indicadores LED de la CPU	9-6
Problemas de comunicaciones	9-7
Búsqueda de problemas de puntos de E/S	9-8
Búsqueda de problemas de ruido	9-10
Partida de máquinas y búsqueda de problemas del programa . . .	9-11
Contactando apoyo técnico en AUTOMATIONDIRECT	9-20

Mantenimiento del sistema de hardware

Mantenimiento normal.

No se requiere ningún mantenimiento regular o preventivo para este producto (no hay baterías internas); sin embargo, es buena práctica una verificación rutinaria (cada un o dos meses) del PLC y sistema de control y debe incluir los puntos siguientes:

- **Temperatura del aire** - supervisar la temperatura del aire en el gabinete de control, de modo que no se sobrepase la temperatura de funcionamiento máxima de ningún componente.
- **Filtro de aire** - si el gabinete de control tiene un filtro de aire, límpielo o reemplacelo periódicamente según lo requerido.
- **Fusibles o interruptores** - verifique que todos los fusibles e interruptores sean en orden.
- **Limpiando la unidad** - compruebe que todas las salidas de aire están sin obstrucción. Si es necesario limpiar la unidad, desconecte la alimentación y limpie cuidadosamente la caja usando un paño húmedo. No deje que entre agua a la caja a través de las salidas de aire y no use detergentes fuertes porque esto puede descolorar la caja.

Diagnósticos

El PLC DL06 realiza muchas rutinas de diagnóstico predefinidas con cada barrido de la CPU. El diagnóstico puede detectar varios errores o fallas en el PLC. Hay dos clases de error: *fatales y no fatales*.

Errores fatales.

Los errores fatales son errores que pueden hacer el sistema funcionar incorrectamente, quizás introduciendo un problema de seguridad. La CPU cambiará automáticamente al modo de programa si está en modo RUN (recuerde, en modo de programa todas las salidas se apagan). Si se detecta un error fatal mientras la CPU esté en modo de programa, la CPU no le permitirá hacer la transición al modo RUN hasta que se haya corregido el error.

Algunos ejemplos de errores fatales son:

- Falla en la fuente de alimentación
- Error de paridad o malfuncionamiento de la CPU
- Errores de programación particulares

Errores no fatales

Los errores no fatales son errores que necesitan su atención, pero no causan una operación incorrecta. No causan ni bloquean ninguna transición del modo de la CPU. El programa en uso puede utilizar contactos de un relevador especial para detectar errores no fatales e incluso lleva el sistema a una parada ordenada o cambia la CPU al modo Program si se desea. Un ejemplo de un error no fatal es:

- **Errores de programación particulares** - los dispositivos de programación le notificarán de un error si ocurre uno mientras está conectado.
- **DirectSOFT** proporciona el número del error y un mensaje de error.
- El programador D2-HPP exhibe números de error y descripciones cortas del error.

El apéndice B tiene una lista completa de los mensajes de error en orden por número de error. Muchos mensajes de error apuntan a direcciones complementarias de memoria V que contienen la información relacionada. Los relevadores especiales (contactos SP) también proporcionan indicaciones de error (vea el apéndice D).

Direcciones de memoria de códigos de error

La tabla siguiente lista las direcciones de memoria específicas que corresponden a ciertos tipos de mensajes de error.

Clase de error	Categoría del error	Memoria V de diagnóstico
Definido por el usuario	Código de error usado con la instrucción FAULT	V7751
Error del sistema	Código de error fatal	V7755
	Código de error mayor	V7756
	Código de error menor	V7757
Error gramatical	Dirección donde ocurrió el error de sintaxis	V7763
	Código de error encontrado durante la verificación de syntaxis	V7764
CPU Scan	Número de barridos desde la última transición de modo Program a RUN	V7765
	Tiempo de barrido corriente (ms)	V7775
	Tiempo de barrido mínimo (ms)	V7776
	Tiempo de barrido máximo (ms)	V7777

Relevadores especiales (SP) que corresponden a códigos de error

La tabla de relevadores especiales también incluye indicadores de estado que pueden indicar errores. Para una descripción más detallada de cada uno de estos relevadores especiales vea el apéndice D.

Relevadores de estado de la CPU	
SP11	Modo funcionamiento forzado
SP12	Modo Terminal Run
SP13	Modo Test Run
SP15	Modo Test stop
SP16	Modo Terminal Program mode
SP17	Parada forzada (Forced stop)
SP20	Se ejecutó instrucción STOP
SP22	Interrupción permitida
Relevadores de supervisión del sistema	
SP36	Configuración de Override
SP37	Error en control de barrido
SP40	Error crítico
SP41	Error no crítico
SP42	Error de diagnóstico
SP44	Error de memoria de programa
SP45	Error de entradas o salidas
SP46	Error de comunicación
SP50	Se ejecutó instrucción FAULT
SP51	Time our del Watchdog

SP52	Error de syntaxis
SP53	No puede resolver la lógica
SP54	Error de comunicación
SP56	Table instruction overrun
Relevadores de estado del acumulador	
SP60	El acumulador es menor que el valor
SP61	El acumulador es igual al valor
SP62	El acumulador es mayor que el valor
SP63	El resultado en el acumuladore es cero
SP64	Ocurrió un “pide prestado mitad”
SP65	Ocurrió un “pide prestado”
SP66	Ocurrió un “carry mitad”
SP67	Ocurrió un “carry”
SP70	El resultado eis negativo (signo)
SP71	Eroor de referencia del puntero
SP73	Desbordamiento
SP75	El dato no está en BCD
SP76	Instruccion Load es cero

Códigos de error del PLC DL06

Estos errores pueden ser generados por la CPU o por el programador D2-HPP, dependiendo del error real. El apéndice B entrega una descripción más completa de los códigos de error.

Los errores se pueden detectar en varios momentos. Sin embargo, la mayoría de ellos se detectan durante la energización o cuando entra al modo RUN o cuando una secuencia de teclas del programador D2-HPP da lugar a un error o a una petición ilegal.

Código	Descripción	Código	Descripción
E003	Demora en software (Timeout)	E526	La unidad está fuera de línea (Off-line)
E004	Instrucción inválida(Error de paridad en el RAM)	E527	La unidad está en línea (On-line)
E104	Falló el escribir	E528	Modo de CPU
E151	Instrucción inválida	E540	CPU bloqueada
E311	Error de comunicación 1	E541	Contraseña errada
E312	Error de comunicación 2	E542	Reponer contraseña
E313	Error de comunicación 3	E601	Memoria llena
E316	Error de comunicación 6	E602	Instrucción faltando
E320	Time out	E604	Referencia faltando
E321	Error de comunicación	E620	Sin memoria
E360	Time-out del puerto de periférico HP	E621	memoria EEPROM no está vacía
E501	Entrada incorrecta	E622	No hay EEPROM en el programador D2-HPP
E502	Dirección incorrecta	E624	Solamente memoria V
E503	Instrucción incorrecta	E625	Solamente programa
E504	Mala referencia/valor	E627	Operación de escritura mala
E505	Instrucción inválida	E628	Error de tipo de memoria (debería ser EEPROM)
E506	Operación inválida	E640	Comparación incorrecta
E520	Operación incorrecta – CPU en Run	E650	Error de sistema del programador
E521	Operación incorrecta – CPU en Test Run	E651	Error de ROM del programador
E523	Operación incorrecta– CPU en Test Program	E652	Error de RAM del programador
E524	Operación incorrecta – CPU en Program		
E525	Conmutador de mode no en posición TERM		

Códigos de errores del programa

La tabla siguiente lista códigos de sintaxis del programa y de error durante el modo RUN. La detección de error ocurre durante la transición del modo Program a RUN o cuando se usa AUX 21 - verifique el programa. La CPU también activará SP52 y almacenará el código de error en V7755. El Apéndice B entrega una descripción más completa de los códigos de error.

Código	Descripción	Código	Descripción
E4**	No hay un programa en la CPU	E438	Dirección inválida IRT
E401	Falta una instrucción END	E440	IDirección inválida de datos
E402	Falta un LBL	E441	ACON/NCON en el cuerpo principal del programa
E403	Falta un RET	E451	Numeración incorrecta de MLS/MLR
E404	Falta un FOR	E453	Falta un temporizador o contador
E405	Falta un NEXT	E454	Uno de los contactos de TMRA está faltando
E406	Falta un IRT	E455	Uno de los contactos de CNT está faltando
E412	SBR / LBL >64	E456	Uno de los contactos de SR está faltando
E421	Referencia de etapas duplicada	E461	Mas de 9 niveles han sido almacenados en el stack
E422	Referencia de SBR/LBL duplicada	E462	No hay un almacenamiento correcto en el stack
E423	Existe un lazo NEXT/LOOP en otro	E463	No se ha usado una instrucción STR/STRN en renglón
E431	Dirección inválida ISG/SG	E464	Falta un circuito en el programa
E433	Dirección inválida ISG / SG	E471	Referencia de bobina duplicada
E434	Dirección inválida RTC	E472	Referencia de temporizador duplicada
E435	Dirección inválida RT	E473	Referencia de contador duplicada
E436	Dirección inválida INT	E499	Uso de la instrucción PRINT inválida
E437	Dirección inválida IRTC		

Indicadores LED de la CPU

Los PLCs DL06 tienen indicadores LED en el frente para ayudarle a determinar problemas potenciales con el sistema. En la operación normal, solamente los indicadores RUN y PWR estarán encendidos. La tabla abajo es una referencia rápida a problemas potenciales.

Estado del indicador	Problemas potenciales
PWR (LED verde apagado)	Voltaje del sistema no está correcto La fuente de poder del PLC ha fallado (la que genera 24 y 5 VCC)
RUN (Green LED off)	Error de programación de la CPU (CPU en modo program)
CPU (Red LED on)	Interferencia de ruido electricomagnético La CPU está defectuosa
CPU (Blinking Red LED)	La batería de respaldo tiene bajo voltaje (vea la página 4-8)

Indicador PWR

En general hay tres razones por las que el estado del LED PWR en la CPU esté apagado:

1. La alimentación a la unidad es incorrecta o no está conectada.
2. La fuente de poder interna del PLC está con falla.
3. Otros componentes hacen que la fuente de poder interna no encienda.

Si el voltaje de alimentación no está correcto, el PLC puede no funcionar correctamente o puede no encender. Use las reglas siguientes para corregir el problema.

ADVERTENCIA: Para reducir al mínimo el riesgo de choque eléctrico, desconecte siempre la energía del sistema antes de examinar el cableado físico.

1. Primero, desconecte la energía externa.
2. Verifique que todos los interruptores o fusibles externos estén intactos.
3. Verifique todo el cableado entrante por posibles conexiones sueltas. Si usted está utilizando un bloque de terminales separado, compruebe esas conexiones para saber si hay exactitud e integridad.
4. Si las conexiones están aceptables, vuelva a conectar la energía del sistema y verifique que el voltaje en la entrada del PLC DL06 está dentro de lo especificado. Si el voltaje no está correcto, apague el sistema y corrija el problema.
5. Si todo el cableado está conectado correctamente y la alimentación está dentro de lo especificado, la fuente interna del PLC pueden estar defectuosa.

La mejor manera de comprobar para saber si hay un PLC dañado es substituir el PLC por uno que se sepa que está funcionando para considerar si ésto corrige el problema. Los conectores removibles en el DL06 hacen esto relativamente fácil. Si ha habido una sobretension transitoria importante en la alimentación, es posible se haya dañado la fuente de poder interna del PLC. Si usted sospecha ésta es la causa del daño de la fuente de poder, considere instalar un filtro acondicionador de CA para atenuar picos de voltaje perjudiciales en el futuro.



El indicador LED RUN

Si la CPU no entra en el modo RUN (de funcionamiento, es decir el indicador LED RUN está apagado), el problema está generalmente en el programa, a menos que la CPU tenga un error fatal. Si ha ocurrido un error fatal, el LED CPU debe estar encendido (usted puede utilizar un dispositivo de programación para determinar la causa del error).

Ambos dispositivos de programación, el programador D2-HPP y *DirectSOFT*, mostrarán un mensaje de error que describe el problema. Dependiendo del error, puede también haber una función AUX que usted puede utilizar para ayudar para diagnosticar el problema. El error de programación más común es "Missing END Statement", esto es, al programa le falta la instrucción END. Todos los programas requieren una declaración END para terminar adecuadamente. Una lista completa de códigos de error se puede encontrar en el apéndice B.

El indicador CPU

Si el indicador CPU está encendido, ha ocurrido un error fatal en la CPU. Generalmente éste no es un problema de programación sino una falla real de hardware. Usted puede apagar y prender el PLC para tratar de limpiar el error. Si el error se limpia, usted debe supervisar el sistema y determinar qué causó el problema. Encontrará que este problema es a veces causado por ruido eléctrico de alta frecuencia introducido en la CPU de una fuente exterior. Compruebe su sistema de tierra e instale filtros anti-ruídos eléctricos si se sospecha la conexión a tierra. Si apagar y prender el PLC no limpia el error o si el problema vuelve, usted debe substituir la CPU.

Si el indicador de la CPU está centelleando, la batería de respaldo tiene el voltaje bajo (vea la página 4-8).

Problemas de comunicación

Si usted no puede establecer comunicaciones con la CPU, compruebe estos artículos:

- Está desconectado un cable.
- El cable tiene un conductor abierto o se ha hecho errada la conexión.
- El cable está terminado incorrectamente o está colocado a tierra.
- El dispositivo conectado no está funcionando a la velocidad correcta (9600 baud, por ejemplo).
- El dispositivo conectado con el puerto está enviando datos incorrectamente, o puede ser que otra operación está funcionando en el dispositivo.
- Hay una diferencia de voltaje de aterramiento entre los dos dispositivos.
- Ruido eléctrico está causando errores intermitentes.
- El PLC tiene un puerto de comunicación dañado y se debe substituir.

Para problemas de comunicación con *DirectSOFT* en una computadora personal, vea el manual *DirectSOFT*. Incluye una sección de localización de fallas que puede ayudarle a diagnosticar problemas de la PC en la configuración del puerto de comunicación, la dirección o conflictos de interrupción, etc.

Búsqueda de problemas de entradas y salidas (E/S)

Causas posibles

Si usted sospecha un error de E/S, hay varias cosas que podrían causar el problema:

- Error de la configuración de E/S de alta velocidad.
- Un fusible quemado en su máquina o panel (el DL06 no tiene fusibles internos de E/S)
- Un bloque de terminales suelto.
- La fuente auxiliar de 24 VCC ha fallado.
- El circuito de entradas o de salidas ha fallado.

Algunos pasos de verificación rápidos

Al localizar averías del PLC DL06, sepa los hechos siguientes que pueden ayudarle a corregir rápidamente un problema de entradas y salidas.

- Los errores de configuración de HSIO se confunden comúnmente por fallas de E/S durante el desarrollo del programa. Si las entradas y salidas en cuestión están en X0-X2, o Y0-Y1, compruebe todas las direcciones de parámetros enumeradas en el capítulo 3 que se apliquen al modo de HSIO que usted ha seleccionado.
- Los circuitos de salidas no pueden detectar puntos de salidas en cortocircuito o abiertos. Si usted sospecha unos o más puntos con falla, mida la caída de tensión desde el común al punto sospechoso. Recuerde que cuando usa un voltímetro digital, se debe considerar la corriente de fuga de un dispositivo de salida tal como un triac o un transistor. Un punto que está apagado puede parecer encendido si no se conecta ninguna carga al punto.
- Los indicadores de estado del punto de E/S son indicadores del lado de la lógica. Esto significa que el LED que indica estado encendido o apagado refleja el estado del punto con respecto a la CPU. En un punto de salida los indicadores de estado podrían funcionar normalmente mientras que el dispositivo de salida real (transistor, triac etc.) podría estar dañado. Con un punto de entrada, si el indicador LED está encendido, el circuito de entrada está funcionando probablemente correctamente. Verifique que se apague el LED cuando se quita la señal de entrada.
- La corriente de fuga puede ser un problema cuando se conectan dispositivos del campo a una entrada o una salida. Pueden ser generadas señales de entrada falsas cuando la corriente de fuga de un dispositivo de salida es bastante grande para activar el dispositivo de entrada conectado. Para corregir esto instale una resistencia en paralelo a la entrada o a la salida del circuito. El valor de esta resistencia dependerá de la cantidad de corriente de fuga y del voltaje aplicado pero normalmente una resistencia de 10 a 20 kohm trabajará bien. Verifique que la potencia de la resistencia esté correcta para su aplicación.
- Ya que hay bloques de terminales removibles en el DL06, el método más fácil para determinar si un circuito de E/S ha fallado es substituir la unidad si usted tiene repuestos. Sin embargo, si usted sospecha que un dispositivo de campo es defectuoso, ese dispositivo puede causar la misma falla en el PLC reemplazado también. Como precaución, usted puede comprobar los dispositivos o las fuentes de alimentación conectadas con las entradas o salidas con falla antes de substituir la unidad por un repuesto.

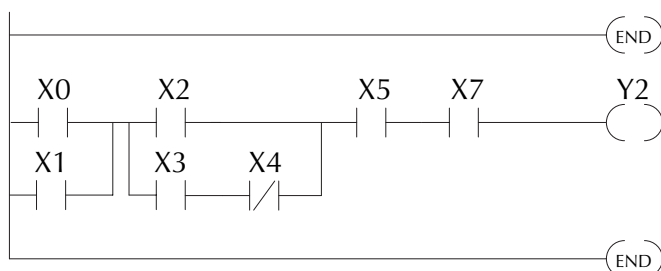
Los puntos de salidas se pueden configurar encendidos o apagados en los PLCs DL06. ¶Si usted desea hacer una verificación de entradas y salidas independientemente del programa en uso, siga el procedimiento abajo:

Paso	Acción
1	Use un programador o DirectSOFT para comunicarse en línea al PLC..
2	Cambie al modo a Program.
3	Vaya a la dirección 0 (Es decir, al primer renglón el la programma ladder).
4	Insiera una instrucción END en la dirección 0. (Esto hará que la ejecución de programa ocurra solamente en la dirección 0 y evite que el programa en uso encienda o apague algunas entradas o salidas).
5	Cambie el modo a RUN.
6	Use el aparato de programación para configurar (ON u OFF) los puntos usted desea probar.
7	Cuando usted termine de probar los puntos de E/S, elimine la declaración END en la dirección 0.



ADVERTENCIA: Dependiendo de su uso, forzar puntos de E/S puede causar una operación imprevisible en la máquina, lo que puede dar lugar a un riesgo de daños corporales o del equipo. ¶ Asegúrese de haber tomado todas las medidas de seguridad adecuadas antes de probar cualquier punto de E/S.

Secuencia de teclado en el programador usada para probar un punto de salida



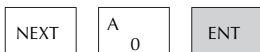
Insiera una instrucción END al comienzo del programa. Esto desactiva el resto del programa.

Use la siguiente secuencia de teclas desde un visor limpio en el programador.



16P STATUS
BIT REF X

Use las teclas PREV o NEXT para seleccionar el tipo de datos Y



Y 10 Y0
□□□□□□□□□□□□□□□□

Use las teclas de flachas para seleccionar un punto
Luego use ON y OFF para cambiar el estado



Y2 es ahora ON
Y 10 Y0
□□□□□□□□□□□□■□□

Búsqueda de fallas de ruido

Problemas eléctricos de ruido electromagnético

El ruido es uno de los problemas más difíciles de diagnosticar. El ruido eléctrico puede entrar en un sistema de muchas maneras y este cae en una de dos categorías, conducido o radiado. Puede ser difícil determinar cómo el ruido está entrando al sistema pero las acciones correctivas para cualquiera de los tipos de problemas de ruido son similares.

- El ruido conducido es cuando la interferencia eléctrica se introduce en el sistema por un conductor unido, una conexión de un panel, etc. Puede entrar a través de un circuito de E/S, de una conexión de la alimentación, de la conexión a tierra de la comunicación, o de la conexión de tierra del chasis.
- El ruido radiado es cuando la interferencia eléctrica se introduce en el sistema sin una conexión eléctrica directa, de una manera semejante como ondas de radio.

Reducción del ruido eléctrico

Aunque el ruido eléctrico no puede ser eliminado puede ser reducido a un nivel que no afecte el sistema.

- La mayoría de los problemas de ruido resultan de un sistema de conexión a tierra incorrecto. Una buena tierra puede ser la manera más eficaz de corregir problemas de ruido. Si no hay una tierra disponible, instale una barra de tierra tan cerca del sistema como sea posible. Asegúrese que todos los conductores de tierra van a un solo punto y no son encadenados a partir de un dispositivo a otro. Atierre los gabinetes metálicos del sistema. Un alambre suelto puede actuar como una antena, introduciendo ruido en el sistema. Por lo tanto, apriete todas las conexiones en su sistema. Los conductores de tierra sueltos son más susceptibles al ruido que los otros alambres en su sistema. Repase el capítulo 2 en la instalación, cableado, y especificaciones si usted tiene dudas con respecto a cómo aterrizar el sistema.
- El ruido eléctrico puede entrar al sistema a través de la alimentación para los circuitos del PLC y de entradas y salidas. La instalación de un transformador de aislación para todas las fuentes de CA puede disminuir este problema. Las fuentes de corriente continua deben ser fuentes de buena calidad, bien aterradas también.
- Separe los cables de las entradas de los cables para las salidas. Nunca instale cables de baja tensión de entradas y salidas cerca de cables de alta potencia y alto voltaje.

Partida de una máquina y búsquedas de fallas del programa

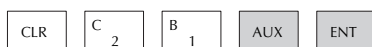
Los PLCs DL06 tienen varias funciones que pueden ayudarle a eliminar errores de su programa antes y durante el arranque de una máquina. Esta sección discute los asuntos siguientes que pueden ser muy provechosos:

- Verificación del programa
- Verificación de referencias duplicadas.
- Instrucciones especiales.
- Correcciones del programa durante el funcionamiento del PLC.
- Forzar puntos de entradas y salidas.

Verificación de sintaxis

Aunque el programador y *DirectSOFT* permiten verificación de errores durante la entrada del programa, se puede querer comprobar un programa que ha sido modificado. Ambos dispositivos de programación ofrecen una manera de comprobar la sintaxis del programa. Por ejemplo, se puede usar AUX 21- CHECK PROGRAM, para comprobar la sintaxis del programa con un programador o usted se puede usar la opción del menú Diagnósticos del PLC dentro de *DirectSOFT*. Esta verificación encontrará una variedad amplia de errores de programación. El ejemplo siguiente muestra cómo utilizar la verificación de sintaxis con un programador.

Use AUX 21 para ejecutar verificación de sintaxis



AUX 21 CHECK PRO
1:SYN 2:DUP REF

Seleccione verificación de sintaxis (Seleccionado por defecto)

ENT (Puede ser que no obtenga el mensaje BUSY si el programa no es largo)

BUSY

Uno de dos mensajes aparecerá:

Mensaje de error (ejemplo)

\$00050 E401
MISSING END
(muestra la dirección donde falta)

Mensaje de sintaxis correcta

NO SYNTAX ERROR
?

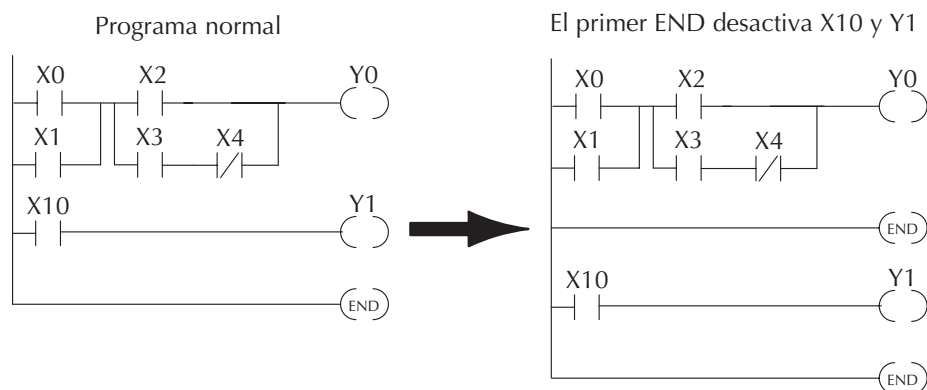
Vea la sección de códigos de error para un listado completo de los códigos de error de programación. Si tiene un error, apriete CLR y el programador exhibirá la dirección de la instrucción donde ocurrió el error. Corrija el problema y continúe operando la función de verificación de sintaxis hasta que aparezca el mensaje NO SYNTAX ERROR.

Instrucciones especiales

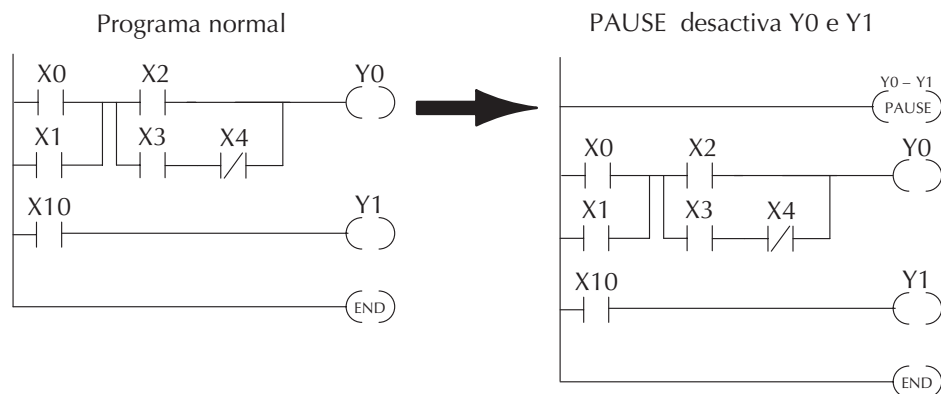
Hay varias instrucciones que se pueden utilizar para ayudarle a eliminar errores de su programa durante operaciones de arranque de una máquina o proceso.

- END
- PAUSE
- STOP

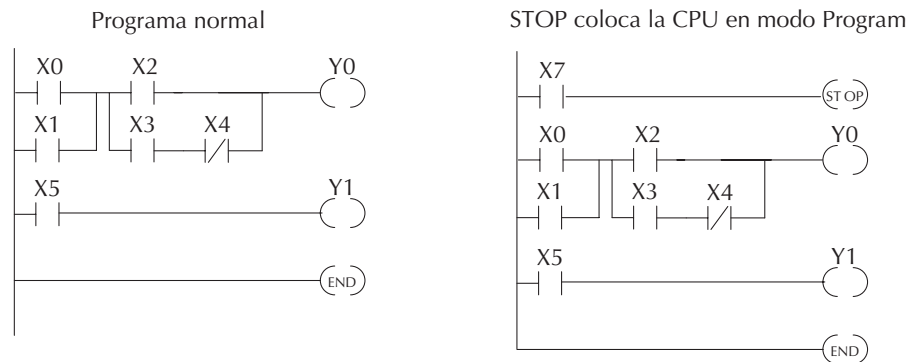
Instrucción END: Si usted necesita una manera de inhabilitar rápidamente una parte del programa, coloque una instrucción END antes de la porción que debe ser desactivada. Cuando la CPU encuentra la instrucción END, asume que es el final del programa. El diagrama siguiente muestra un ejemplo.



Instrucción PAUSE: Esta instrucción proporciona una manera rápida de permitir que las entradas (u otra lógica) funcionen mientras que inhabilita salidas seleccionadas. La memoria imagen de salida aún se actualiza, pero no los circuitos de salida. Por ejemplo, usted podría hacer esta condición agregando un contacto de entrada o un CR para controlar la instrucción con un interruptor o un dispositivo de programación. O podría agregar la instrucción sin ninguna condición de modo que las salidas seleccionadas estarían inhabilitadas siempre.



Instrucción STOP: A veces durante el arranque de una máquina usted necesita una manera de rápidamente apagar todas las salidas y volver al modo Program. Usted puede utilizar la instrucción STOP. Cuando se ejecuta esta instrucción, la CPU sale automáticamente del modo RUN y entra en modo Program. Recuerde, todas las salidas se apagan durante modo Program. El diagrama siguiente muestra un ejemplo de una condición que vuelva la CPU al modo Program.

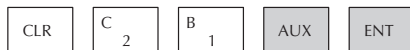


En el ejemplo mostrado arriba, usted podría accionar X7, que ejecutaría la instrucción STOP. La CPU entraría en modo Program y todas las salidas serían apagadas.

Verificación de referencias duplicadas

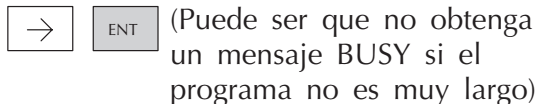
Usted puede también comprobar si hay aplicaciones múltiples de la misma bobina de salida. Ambos dispositivos de programación ofrecen una manera de comprobar si existe esta condición. Por ejemplo, usted puede hacer AUX 21-CHECK PROGRAM para comprobar si hay referencias duplicadas en un programador o usted puede utilizar la opción del menú PLC>Diagnostics dentro de *DirectSOFT*. El ejemplo siguiente muestra cómo realizar la verificación de referencias dobles con un programador.

Use AUX 21 para ejecutar verificación sintáctica



AUX 21 CHECK PRO
1:SYN 2:DUP REF

Seleccione "DUP REF"



BUSY

Aparecerá uno de estos dos mensajes:

Mensaje de error (ejemplo)
(Muestra la dirección con el error)

\$00024 E471
DUP COIL REF

Mensaje "No hay refs. duplicadas"

NO DUP REFS
?

Si tiene un error, apriete CLR y el programador mostrará la instrucción donde ocurrió el error. Corrija el problema y continúe operando la verificación de referencias dobles hasta que no se encuentre ninguna referencia duplicada.



NOTA: Usted puede usar la misma bobina en más de una localización, especialmente en los programas que contienen instrucciones de etapas e instrucciones OROUT. La verificación de referencias duplicadas encontrará ocurrencias, que es aceptable.

Modificaciones del programa durante el modo RUN

El PLC DL06 permite realizar cambios al programa en uso durante el modo RUN. Éstas correcciones no son sin algunos efectos en el funcionamiento. En ese momento, la CPU se interrumpe momentáneamente (y las salidas se mantienen en su estado actual) hasta que el cambio del programa se haya completado. Esto significa que, si la salida está apagada, permanecerá apagada hasta que el cambio del programa se complete. Si la salida está encendida, seguirá estando encendida.



ADVERTENCIA: Solamente personal autorizado y familiar con todos los aspectos del uso debe realizar cambios al programa. Los cambios durante modo de funcionamiento se hacen efectivos inmediatamente. Asegúrese completamente de considerar el impacto de cualquier cambio para reducir al mínimo el riesgo de daños corporales o al equipo. Hay algunos cambios operacionales importantes durante tiempo de corrección del programa:

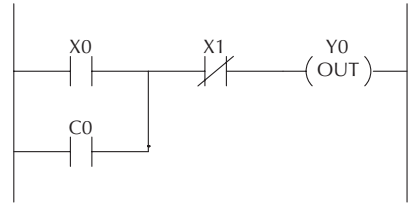
1. Si hay un error de sintaxis en la nueva instrucción, la CPU no entrará al modo RUN.
2. Si usted suprime una referencia de bobina de salida y la salida estaba encendida en ese momento, la salida permanecerá encendida hasta que se fuerce apagada con un dispositivo de programación.
3. Los cambios de estado en las entradas no se reconocen durante el tiempo de modificación del programa, de modo que, si usted está utilizando una operación de alta velocidad y cambia el estado de una entrada crítica, la CPU pueden no notar el cambio.

No todas las instrucciones se pueden corregir durante una sesión de modificación durante el modo RUN. La lista siguiente muestra las instrucciones que se pueden modificar.

Pneumónico	Descripción	Pneumónico	Descripción
TMR	Temporizador	OR, ORN	Or greater than or equal or less than (Comparative Boolean)
TMRF	Temporizador rápido	LD	Copiar datos al acumulador (constante)
TMRA	Temporizador acumulador de tiempo	LDD	Copiar palabras dobles al acumulador (constante)
TMRAF	Temporizador acumulador rápido	ADDD	Sumar palabras dobles (constante)
CNT	Contador	SUBD	Restar palabras dobles (constante)
UDC	Contador incremental\decremental	MUL	Multiplicar (constante)
SGCNT	Contador de etapas	DIV	Dividir (constante)
STR, STRN	Store, Store not (Boolean)	CMPD	Compare con acumulador (constante)
AND, ANDN	And, And not (Boolean)	ANDD	And con acumulador (constante)
OR, ORN	Or, Or not (Boolean)	ORD	Or con acumulador (constante)
STRE, STRNE	Store equal, Store not equal	XORD	Exclusive or con acumulador (constante)
ANDE, ANDNE	And equal, And not equal	LDF	Copiar puntos discretos al acumulador
ORE, ORNE	Or equal, Or not equal	OUTF	Copie el acumulador a puntos discretos
STR, STRN	Store greater than or equal Store less than (Booleano Comparativo)	SHFR	Shift accumulator right
AND, ANDN	And greater than or equal And less than (Booleano Comparativo)	SHFL	Shift accumulator left
		NCON	Constante numérica

Ejemplo de edición durante modo RUN

Utilizaremos "el programa" mostrado para describir cómo trabaja este proceso con un programador D2-HPP. En el ejemplo, cambiaremos X0 a C10. Note, el ejemplo asume que ya ha colocado la CPU en modo RUN.



Use la tecla **MODE** para seleccionar edición durante RUN (Run time edits).



MODE CHANGE
RUN TIME EDIT?

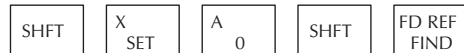
Apriete **ENT** para confirmar Run time edits

(Note, el LED RUN en el D2-HPP comienza a destellar para indicar que pueden ser hechos cambios en modo RUN).



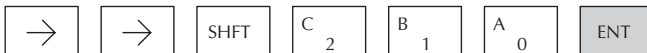
MODE CHANGE
RUNTIME EDITS

Encuentre la instrucción que usted desea cambiar (X0).



\$00000 STR X0

Apriete la tecla "flecha" para moverse a la X. Luego enter el contacto nuevo (C10).



RUNTIME EDIT?
STR C10

Apriete **ENT** para confirmar el cambio.

(Note, cuando se aprieta ENT, aparece la próxima dirección).



OR C0

Forzando puntos de entradas y salidas

Hay muchas veces que usted necesita la capacidad de forzar un punto de E/S para ser encendido o apagado, especialmente durante el arranque de una máquina y la localización de problemas. Antes de que usted utilice un dispositivo de programación para forzar cualquier tipo de datos, es importante entender cómo el PLC DL06 procesa el pedido de forzar.



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

Hay dos tipos de forzar disponibles con el PLC DL06. (El capítulo 3 proporciona una descripción detallada de cómo la CPU procesa cada tipo de petición de forzar).

- **Forzar regular:** Este tipo de forzar puede cambiar temporalmente el estado de un bit discreto. Por ejemplo, usted puede desear forzar una entrada a encendida, aunque está apagada realmente. Esto le permite cambiar el estado del punto 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 de la CPU. Esto es útil sobre todo durante situaciones de prueba cuando usted necesita forzar un bit a encendido para accionar otro evento.
- **Bit Override :** Bit override se puede activar punto por punto usando AUX 50 con el programador o por una opción de menú en DirectSOFT. Usted puede utilizar bit override con los tipos de datos X, Y, C, T, CT, y S. El bit override inhabilita básicamente cualquier cambio por la CPU al punto discreto. Por ejemplo, si usted permite bit override para X1 y X1 está apagada en ese entonces, la CPU no cambiará el estado de X1. Esto significa que incluso si X1 se enciende, la CPU no reconocerá el cambio. Por lo tanto, si usted utilizara X1 en el programa, sería evaluada siempre "OFF" en este caso. Si X1 estuviera encendida cuando el bit override fue permitido, después X1 sería evaluada siempre como "ON".

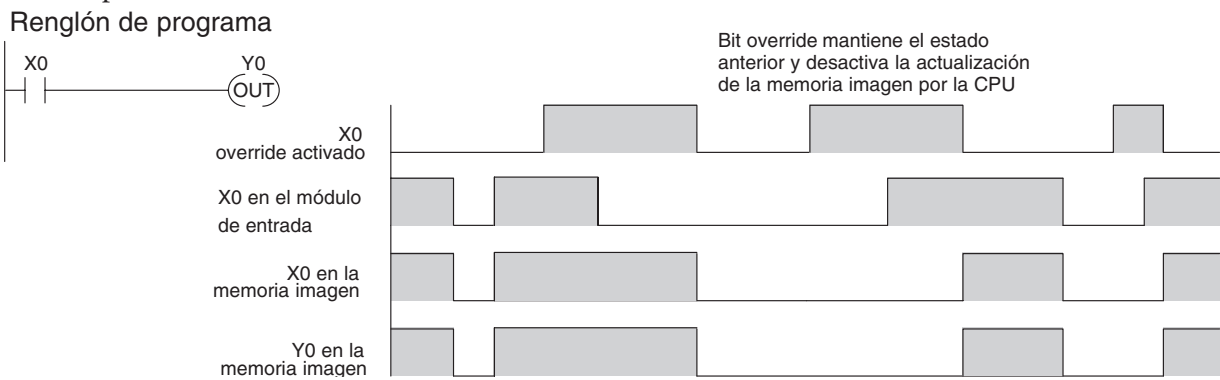
Hay una ventaja disponible cuando usted utiliza la característica de bit override. El forzar regular no es desactivado porque se permite el bit override.

Por ejemplo, si usted permitiera el bit override para Y0 y estuviera apagada en ese entonces, la CPU no cambiaría el estado de Y0.

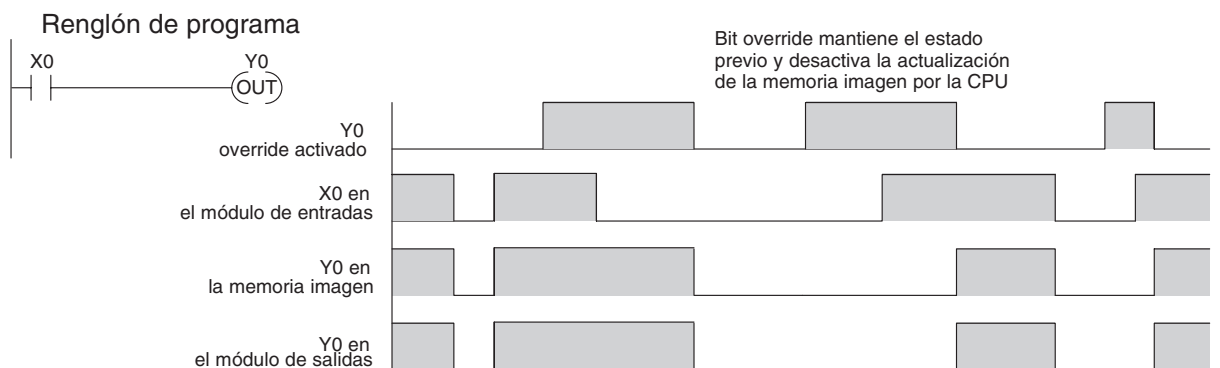
Sin embargo, usted puede aún utilizar un dispositivo de programación para cambiar el estado. Si usted utiliza el dispositivo de programación para forzar la salida Y0 a encendida, permanecerá encendida y la CPU no cambiará el estado de Y0. Si usted entonces fuerza Y0 a apagada, la CPU mantendrá Y0 como apagada.

La CPU nunca actualizará el punto con los resultados del programa en uso o de la actualización de E/S hasta que la condición de bit override sea sacada desde ese punto.

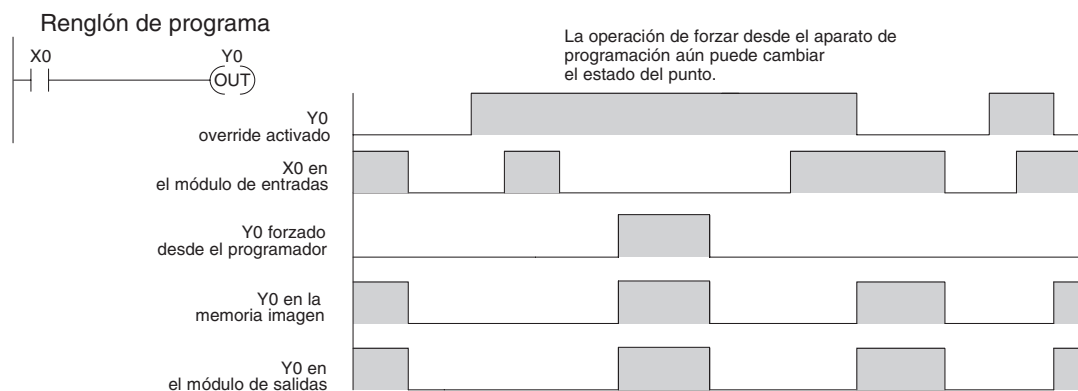
Los diagramas siguientes muestran cómo el bit override funciona para los puntos de entrada y de salida. El ejemplo utiliza un renglón simple, pero los conceptos son similares para cualquier tipo de memoria de bit.



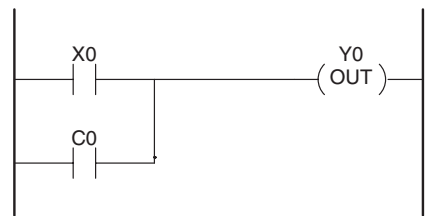
El diagrama siguiente muestra cómo el bit override funciona para un punto de salida. Note que el bit override mantiene la salida en el estado actual. Si la salida es encendida cuando se permite el bit override, entonces la salida permanece encendida. Si está apagada, entonces la salida permanece apagada.



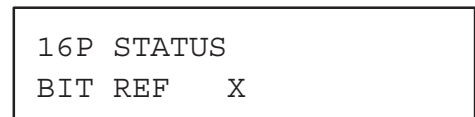
El diagrama siguiente demuestra cómo usted puede utilizar un dispositivo de programación conjuntamente con el bit override para cambiar el estado del punto. Recuerde, el bit override desactiva cambios de la CPU solamente. Usted puede todavía utilizar un dispositivo de programación para forzar el estado del punto. Además, puesto que el bit override mantiene el estado actual, esto permite forzar verdaderamente. El ejemplo mostrado es para un punto de salida, pero usted puede también utilizar los otros tipos de datos de bit.



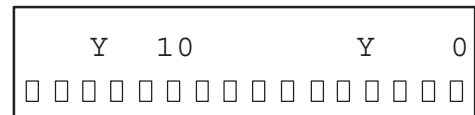
Los diagramas siguientes muestran un ejemplo simple de cómo se podría utilizar el programador del DL06 para forzar un punto de E/S. Recuerde, si usted está usando el bit override, la CPU conservará el valor forzado hasta que usted inhabilite el bit override o hasta que usted quite el forzado. La memoria imagen no será actualizada con el estado del módulo de entradas. También, la solución del programa en uso no será utilizada para actualizar la memoria imagen de la salidas. El ejemplo asume que usted ya ha colocado la CPU en modo RUN.



Con un visor limpio, utilice la siguiente secuencia de teclas:



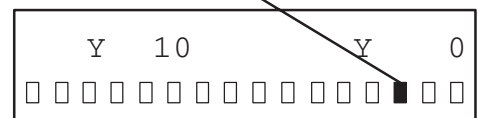
Use las teclas PREV o NEXT para seleccionar el tipo de datos Y. (Una vez que aparezca Y, presione 0 para comenzar en Y0)



Use las teclas de flecha para seleccionar el punto, después use ON y OFF para cambiar el estado.



Y2 está ON ahora

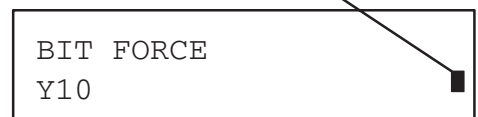


Forzar normal con acceso directo

De un visor limpio, use las teclas siguientes para forzar Y10 ON. El rectángulo sólido indica que el punto está ON o encendido .



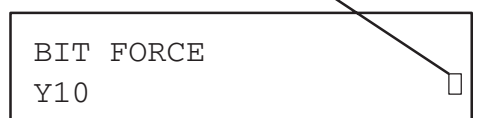
Rectángulo sólido significa que el punto está ON



De un visor limpio, use las teclas siguientes para forzar Y10. El rectángulo sin relleno indica que el punto está OFF o apagado.

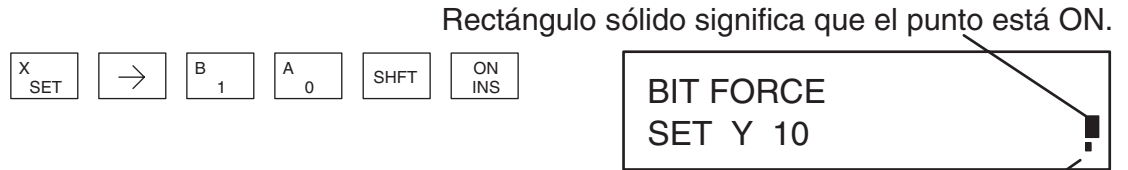


Rectángulo sin relleno significa que el punto está OFF.



Forzar con el bit override

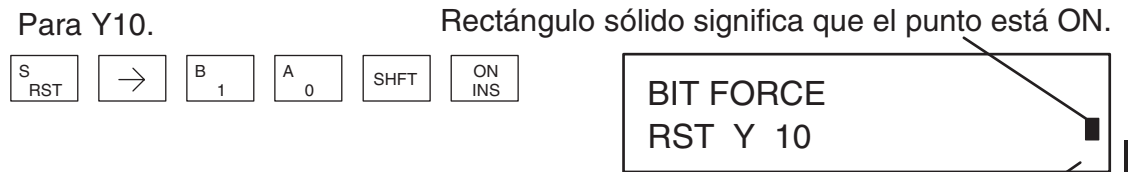
Desde un visor limpio, use las teclas siguientes para activar el bit override para Y10.



El cuadro pequeño indica que el bit override está on.

Note, en este punto usted puede utilizar las teclas PREV y NEXT para navegar a las memorias adyacentes y usar la tecla SHFT ON para activar el bit override.

Desde un visor limpio, use las teclas siguientes para desactivar el bit override para Y10. El rectángulo sólido indica que el puntop está ON.



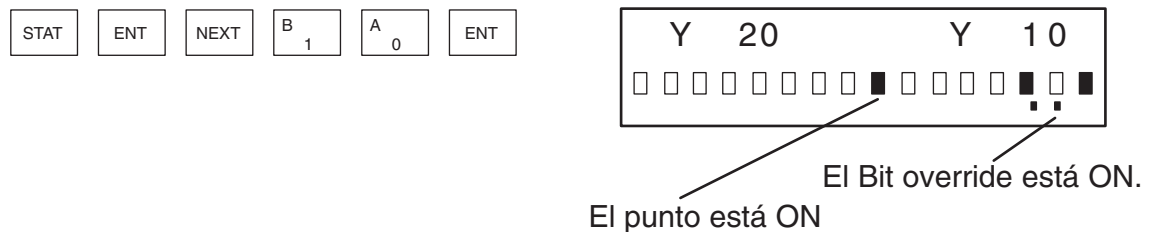
Si el cuadro pequeño no aparece, indica que el bit override está OFF.

Como el ejemplo de arriba, usted puede utilizar las teclas PREV y NEXT para navegar a las memorias adyacentes y usar la tecla SHFT OFF para desactivar el bit override

Indicadores de bit override

Los indicadores de estado del Bit Override también se muestran en el visor del programador. En la figura de abajo están las teclas para llamar el visor de estado de Y10 - Y20.

Desde un visor limpio, use las teclas siguientes para mostrar el estado de Y10 – Y20.



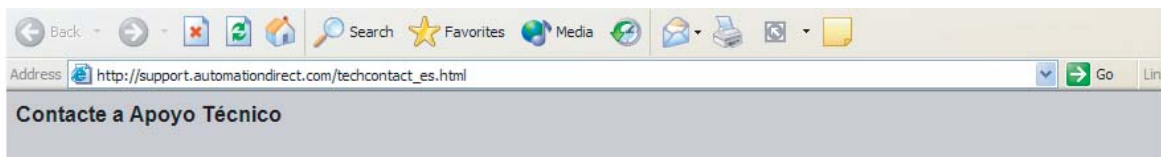
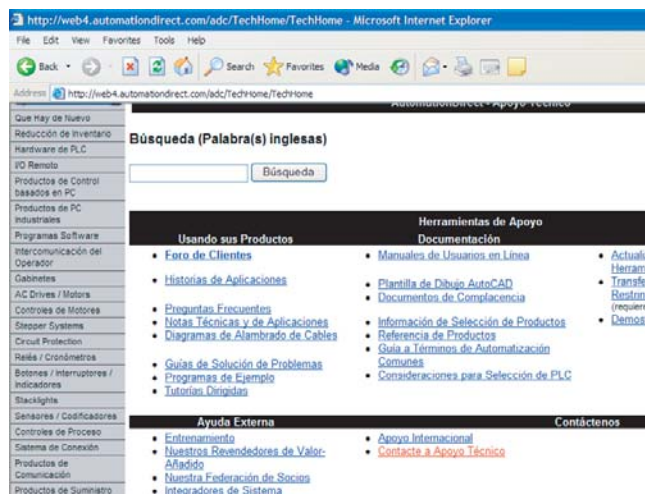
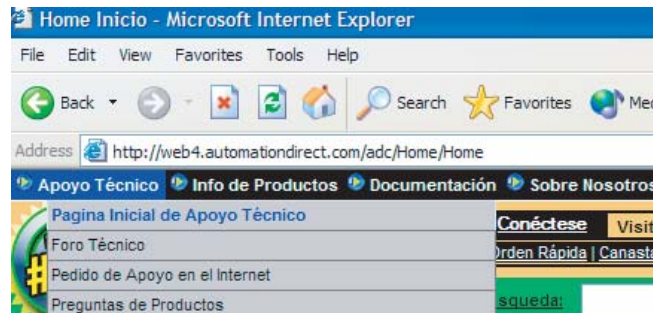
Contactando apoyo técnico en AUTOMATION DIRECT.

Si todo falla, Ud. puede buscar consejos, ejemplos, conversar con otros usuarios en nuestro foro, (en inglés) y obtener una serie de informaciones específicas o también contactar apoyo técnico en **AUTOMATIONDIRECT** a través de Internet o por teléfono. Para contactar Apoyo Técnico por Internet, siga las siguientes instrucciones:

Visite nuestro sitio de Internet en www.automationdirect.com y seleccione español. Luego coloque el cursor del mouse sobre Apoyo Técnico. Aparecerá un sub menú, y en él la primera línea será Página inicial de Apoyo Técnico. Haga clic en esta línea y aparecerá una nueva página.

La figura a la izquierda es la página de apoyo técnico, que ofrece todo lo descrito allí; con el desarrollo de nuestro sitio se puede encontrar que hay adiciones o modificaciones de asuntos. En el caso de querer contactarnos, por favor haga clic al enlace “[Contacte a Apoyo Técnico](#)”.

Esta acción lo llevará a la pagina general de información de como contactarnos, mostrada en la figura de abajo.



Teléfono **001-770-844-4200**

Nuestras horas de apoyo técnico son de 9am a 6pm Hora del Este de Lunes a Viernes.

FAX **001-770-886-3199**

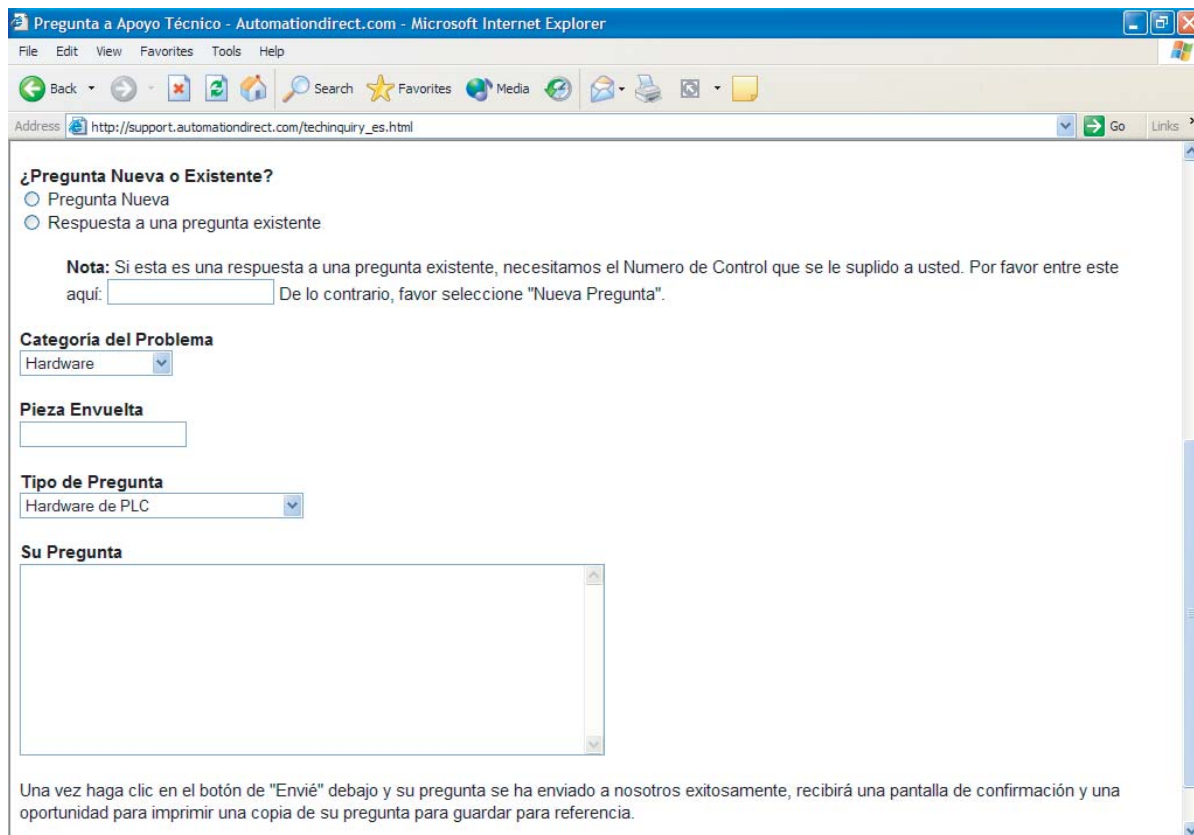
Correo Electrónico: **[Envíenos un correo electrónico](#)**

Foro de Discusión: **[Foro de Clientes de AutomationDirect](#)**

Por favor note que mientras que los miembros de nuestro equipo de Apoyo Técnico a veces participan en temas en el Foro, la intención del Foro es tener grupos de nuestros clientes discutiendo acerca de sus aplicaciones, programación, u otros problemas que puedan tener usando o seleccionando productos de AutomationDirect. Clientes que necesiten una ayuda más inmediata o ayuda más definida deben usar las opciones de teléfono, fax, o correo electrónico.

En este momento Ud. puede decidir escribir un correo electrónico con su pregunta específica. Le pedimos que coloque el máximo de datos en su pregunta, porque, aunque tratamos de entender sus problemas, no podemos adivinar lo que puede estar errado en su problema específico si no nos explica correctamente el ambiente donde está trabajando. Esto nos permitirá responder solamente una vez con una posible solución a su problema.

Si hace clic con el mouse en Envíenos un correo electrónico, aparecerá el siguiente formulario:



The screenshot shows a Microsoft Internet Explorer browser window with the title "Pregunta a Apoyo Técnico - Automationdirect.com". The address bar shows the URL "http://support.automationdirect.com/techinquiry_es.html". The form content is as follows:

¿Pregunta Nueva o Existente?
 Pregunta Nueva
 Respuesta a una pregunta existente

Nota: Si esta es una respuesta a una pregunta existente, necesitamos el Numero de Control que se le suplió a usted. Por favor entre este aquí: De lo contrario, favor seleccione "Nueva Pregunta".

Categoría del Problema
Hardware

Pieza Envuelta

Tipo de Pregunta
Hardware de PLC

Su Pregunta

Una vez haga clic en el botón de "Envíe" debajo y su pregunta se ha enviado a nosotros exitosamente, recibirá una pantalla de confirmación y una oportunidad para imprimir una copia de su pregunta para guardar para referencia.

Escriba su pregunta y envíela siguiendo las instrucciones en la pantalla. Le responderemos típicamente en hasta 12 horas en los días útiles.

PANEL CON VISOR LCD



En este capítulo...

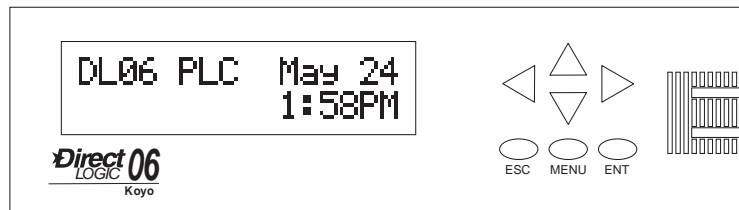
Introducción al visor LCD del DL06	10-2
Teclado del visor LCD	10-2
Instalación de encaje rápido	10-3
Prioridad de la pantalla	10-4
Navegación por el menú	10-5
Confirme PLC, nivel de versión de firmware, uso de memoria, etc.	10-6
Examinando lo que tiene en las ranuras del PLC	10-8
Supervisión y cambio de valores	10-10
Monitor de bits	10-13
Cambiando fecha y hora	10-14
Definiendo la contraseña y bloqueando cambios	10-17
Historia de errores	10-20
Teclas de Prueba, bascule la luz y el zumbador (Beeper)	10-21
Información de memoria de PLC para el visor LCD	10-22
Cambiando la pantalla original de fábrica	10-25
Instrucción del panel con visor LCD (LCD)	10-26

Introducción al panel con visor LCD del PLC DL06

El panel con visor LCD del DL06 tiene un visor de 2 filas de 16 caracteres, que se monta directamente en la frente del PLC DL06. El visor LCD es iluminado por atrás para fácil lectura en la mayoría de las situaciones de iluminación.

Hay múltiples maneras de interactuar con el panel de visor del LCD:

- Teclado numérico incorporado
- Con la instrucción ladder LCD
- Use instrucciones para cambiar el estado de bits en direcciones de memoria especificadas



Las siete teclas de función en la fase del panel con visor LCD dan acceso de usuario al ajuste de hora y fecha, valores de datos de memoria o el estado de entradas y salidas, etc. Un programador u operador con experiencia y con autorización de contraseña puede:

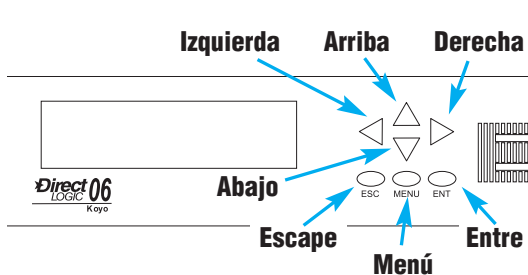
- Cambiar la fecha o la hora, minuto y segundo
- Supervisar o cambiar valores de memoria (inclusive palabras dobles)
- Forzar el estado de bits individuales (hasta 16 por pantalla)
- Ver la historia de códigos de error
- Definir o cambiar la contraseña
- Prender o apagar la iluminación trasera o el zumbador.

Hay usos potenciales para el visor LCD del DL06 que varían extensamente. Un programador puede cambiar los valores para establecer los procesos de batelada o el valor prefijado de temporizadores o contadores de máquina para fabricar productos diferentes. El personal de mantenimiento puede comunicarse con el PLC para identificar problemas en la máquina. Los mensajes de LCD se pueden preprogramar para visualizar eventos de proceso o alarmas. El LCD puede satisfacer estas y muchas otras necesidades.

10

Teclado del visor LCD

El teclado del panel con visor LCD tiene siete teclas que usted puede usar para navegar por la jerarquía del menú. Cada pantalla mostrada tiene un conjunto específico de teclas activas asociadas. Todas las otras teclas (no asociadas con la pantalla corriente) quedan inactivas.

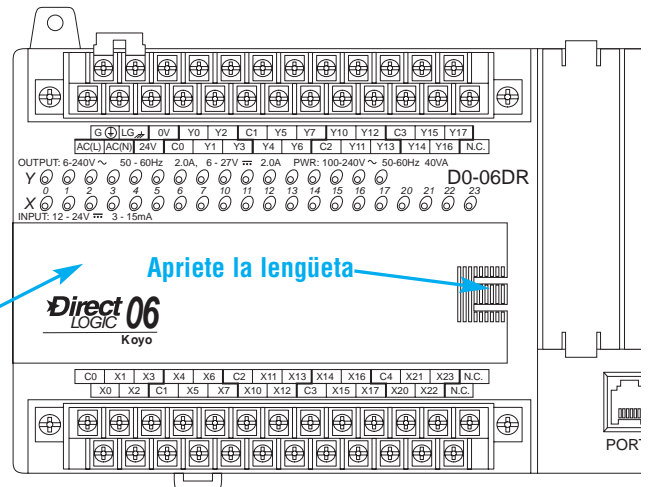


Teclas de función		
Nombre	Rótulo	Función
Flecha arriba	ninguno	Vaya a selección arriba o aumente el valor
Flecha abajo	ninguno	Vaya a selección abajo o disminuya el valor
Flecha izquierda	ninguno	Vaya al próximo dígito a la izquierda
Flecha derecha	ninguno	Vaya al próximo dígito a la derecha
Escape	ESC	Vuelva a la pantalla previa o al próximo nivel de arriba en la jerarquía del menú.
Menú	MENU	Vaya a la pantalla siguiente o al próximo nivel de abajo en la jerarquía del menú.
Entre	ENT	Entre the domain of the menu screen selected or save new value

Instalación de encaje rápido

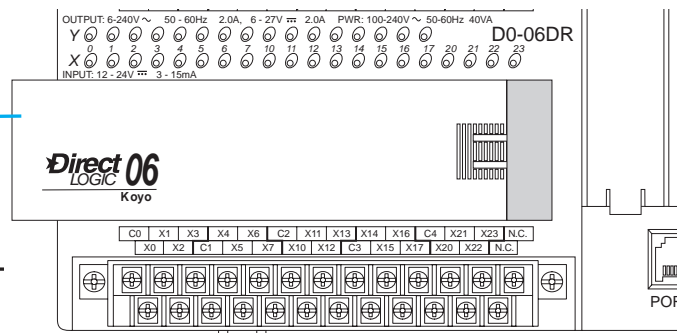
El Panel con visor LCD se instala fácilmente en cualquier PLC DL06. Quite la cubierta plástica (localizada entre los terminales de entradas y salidas). Apriete la lengüeta que cierra la cubierta para liberarla y luego deslice la cubierta a la izquierda aproximadamente 3/8 de pulgada.

Cubierta plástica



La cubierta ahora se puede remover directamente de la ranura del DL06.

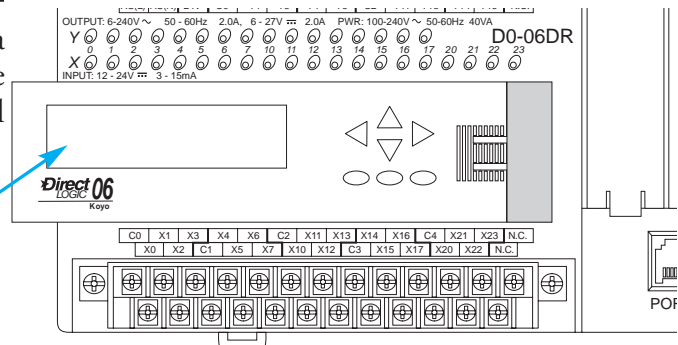
Deslice y levante la cubierta



ADVERTENCIA: Desconecte la alimentación al PLC antes de instalar o quitar el panel con visor LCD.

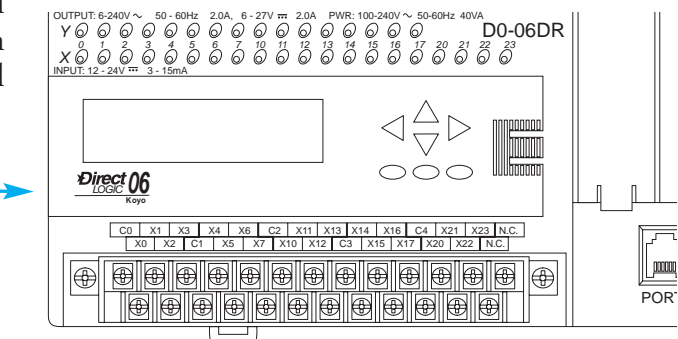
Coloque el panel con visor LCD sobre la apertura pero desviada aproximadamente 3/8 de pulgada a la izquierda. El Panel debe entrar fácilmente en la apertura.

Coloque el panel LCD sobre la apertura



Deslice el panel a la derecha hasta que el lado izquierdo del panel esté nivelado con el lado izquierdo del PLC. El conector del panel hará clic en su lugar cuando entre.

Deslice el panel LCD hasta que entre en su lugar



Prioridad de la pantalla

El panel con visor LCD mostrará uno de las siguientes posibilidades (a menos que el PLC no esté energizado):

- Pantalla definida por el usuario u original de fábrica.)
- Selección de menú
- Mensaje desde el programa ladder
- Mensaje de error

El teclado incorporado le permite navegar por estos mensajes:

Después de la energización del PLC aparece el mensaje por defecto normalmente. El mensaje por defecto es colocado en la fábrica pero puede ser definido por el usuario. La configuración de un mensaje por defecto describe mas tarde en este capítulo

Si ocurre un error de sistema, el mensaje del error se sobrepone al mensaje por defecto (u otra pantalla corrientemente en el visor),y el código del error correspondiente es mostrado para propósito de diagnóstico.

D	L	0	6		P	L	C			M	a	y	0	8	
								1	3	:	5	7	:	0	1

D	i	a	g	n	o	s	t	i	c		E	r	r	o	r
E	4	*	*		N	O		P	R	O	G	R	A	M	

Navegación por el menú

Comenzando en la pantalla por defecto, cada vez usted aprieta la tecla de MENU el visor se desplazará a la próxima opción del menú. Las teclas de flecha hacia arriba y flecha hacia abajo desplazan también la lista de menús (en la dirección indicada por la flecha), *pero usted debe apretar inicialmente la tecla de MENU (en la pantalla por defecto) activar el arriba y teclas de flecha hacia abajo.*

Hay siete selecciones incorporadas de menús. Algunos de los menús tienen sub menús. Los menús y sub menú son descritos en este capítulo. Cada selección del menú requiere que usted apriete la tecla ENTER para ver o cambiar las colocaciones o los valores dentro del dominio de esa selección principal de menú.

Siete elecciones del menú

Apretando y manteniendo apretada la tecla de MENU causará que el visor se desplazará por las opciones siguientes de menú:

- M1 : PLC information
- M2 : System configuration
- M3 : Monitor
- M4 : Calendar read/write
- M5 : Password read/write
- M6 : Error history read
- M7 : LCD test and set

M	E	N	U		S	C	R	E	E	N					
>	M	1	:	P	L	C		I	N	F	O	.			
>	M	2	:	S	Y	S	T	E	M		C	F	G		
>	M	3	:	M	O	N	I	T	O	R					
>	M	4	:	C	A	L	E	N	D	A	R		R	/	W
>	M	5	:	P	A	S	S	W	O	R	D		R	/	W
>	M	6	:	E	R	R		H	I	S	T	O	R	Y	
>	M	7	:	L	C	D		T	E	S	T	&	S	E	T

En esta sección usamos las ilustraciones del teclado de Panel de Visor de LCD y área de visor para mostrar cómo navegar por la jerarquía del menú.

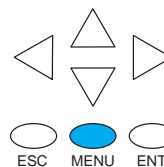
El ejemplo de abajo muestra la pantalla por defecto de la fábrica como SCREEN 1 (Pantalla 1) y la pantalla principal de la entrada de menú como SCREEN 2 (Pantalla 2).

La ilustración del teclado numérico entre ejemplos de pantallas indica que apretando la tecla MENU causa una transición de la Pantalla 1 a la Pantalla 2. Este tipo de representación se usa en esta sección. Cuando se esté dentro de la jerarquía de menú, la tecla ESC vuelve el visor a la pantalla previa.

Pantalla 1 - Configuración de fábrica

D	L	0	6		P	L	C			M	a	y		0	8
								1	4	:	1	2	:	0	1

Apriete la tecla sombreada para navegar de la pantalla 1 a la pantalla 2.



Pantalla 2

M	E	N	U		S	C	R	E	E	N				
>	M	1	:	P	L	C		I	N	F	O	.		

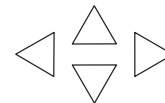
Confirme PLC, versión de firmware, uso de memoria, etc.

Menú 1, M1: INFORMACION de PLC.

Desde la pantalla por defecto, apriete la tecla de MENU una vez para llegar a la opción de menú INFORMACION de PLC.

Pantalla original de fábrica

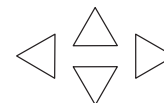
D	L	0	6	P	L	C	M	a	y	0	8
							1	4	:	1	2
									:	0	1



Paso 1.1

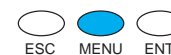
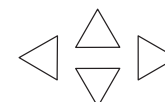
Apriete ENT para seleccionar esta selección de menú. La primera pantalla dentro de la selección de INFORMACION de PLC es M1:PLC TYPE. Esta selección muestra el modelo del PLC.

M	E	N	U	S	C	R	E	E	N				
>	M	1	:	P	L	C	I	N	F	O	.		



Paso 1.2

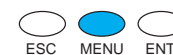
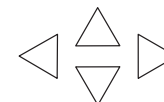
M	1	:	P	L	C	T	Y	P	E				
						D	0	-	0	6	D	D	1



Paso 1.3

Apriete MENU otra vez para ir a PLC MODE

M	1	:	P	L	C	M	O	D	E						
													R	U	N



El PLC MODE es el modo, que puede ser RUN, STOP (en modo STOP o PROGRAM), TEST-STOP (Modo de prueba parado) o TEST-RUN (modo de prueba en RUN) . Ud. puede colocar el PLC DL06 en el modo TEST-RUN desde el modo TEST STOP.

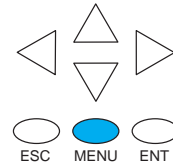
Nota: Los ejemplos de pantalla de menú mostrados en esta sección asumen que la contraseña no está activada. Si la contraseña está activada, el usuario recibirá un mensaje en el Panel del Visor para entrar la contraseña en ese momento. Los usuarios sin autorización de contraseña tendrán acceso a un número limitado de pantallas.



Apriete MENU otra vez para ir a FIRMWARE REV, que es la revisión de la versión del firmware.

Paso 1.4

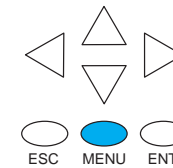
M	1	:	F	I	R	M	W	A	R	E		R	E	V	.
												V	1	.	000



Apriete MENU otra vez para ir a LADDER MEMORY USED. (MEMORIA DE PROGRAMA USADA). Se muestra el número de palabras usado y el número disponible en el PLC.

Paso 1.5

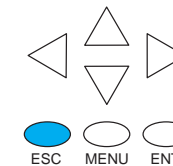
M	1	:	L	A	D	D	E	R		M	E	M	O	R	Y
U	S	E	D						2	1	/		7	6	80



Apriete MENU otra vez para ir a la próxima pantalla LADDER PASSWORD, ACTIVATED o NOT ACTIVATED (Contraseña del programa). Esta es la última pantalla del menú de INFORMACION de PLC y se explica por sí mismo

Paso 1.6

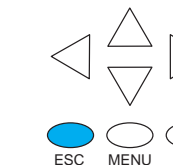
M	1	:	L	A	D	D	E	R		P	A	S	S	W	D
										N	O	T		A	C



Vuelve al paso 1.1

Apriete ESC para salir del menú M1 y regresar al menú principal.

M	E	N	U		S	C	R	E	E	N					
>	M	1	:	P	L	C		I	N	F	O	.			



Pantalla original

Apriete la tecla ESC 2 veces para volver a la pantalla por defecto

D	L	0	6		P	L	C		M	a	y		0	8	
									1	4	:		2	2	:

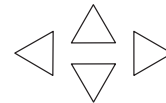
Examinando lo que tiene en las ranuras del PLC

Menu 2, M2:SYSTEM CFG.

En la pantalla por defecto, apriete MENU para llegar dos veces para llegar a la opción de menú M2:SYSTEM CFG. (Configuración de sistema).

Paso 2.1

>	M	1	:	P	L	C		I	N	F	O	.			
>	M	2	:	S	Y	S	T	E	M		C	F	G	.	



Paso 2.2

Apriete ENT para entrar a la selección de menú SYSTEM CFG.

M	2	:	O	P	T	I	O	N		S	L	O	T		1
			D	0	-	D	E	V	N	E	T	S			

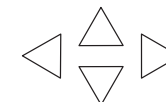


Nota: Este es solamente un ejemplo: Evidentemente presentará lo que exista en la ranura. .

Al apretar la tecla MENU 4 veces se podrá ver lo que hay en cada ranura. Aparece el modelo del módulo en la segunda línea o una indicación de que la ranura está vacía.

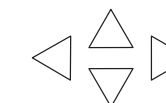
Paso 2.3

M	2	:	O	P	T	I	O	N		S	L	O	T		2	
			E	M	P	T	Y		I	/	O		S	L	O	T



Paso 2.4

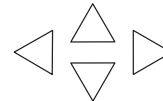
M	2	:	O	P	T	I	O	N		S	L	O	T		3
			F	0	-	0	4	A	D	-	1				



Apriete la tecla ESC 2 veces para volver a la pantalla por defecto

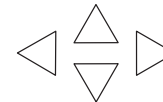
Paso 2.5

M	2	:	O	P	T	I	O	N	S	L	O	T	4	
			E	M	P	T	Y	I	/	O	S	L	O	T



Vuelve al paso 2.1

>	M	1	:	P	L	C	I	N	F	O	.		
>	M	2	:	S	Y	S	T	E	M	C	F	G	.



Vuelve a la pantalla original

D	L	0	6		P	L	C		M	a	y	0	8		
								1	4	:	5	7	:	2	1

Supervisión y cambio de valores

Menu 3, M3:MONITOR

Desde la pantalla por defecto, apriete MENU tres veces para llegar a la opción de menú M3:MONITOR

El M3: El sub menú MONITOR contiene la supervisión de datos de números y de bits. Le permite examina el contenido de memoria o punteros,

El formato por defecto es BCD/hexadecimal, pero el formato M 3: puede ser cambiado a decimal colocando el bit 8 de V7742 en ON.Por favor vea el mapa de memoria del PLC DL06 para los rangos .

Supervisión de datos

Seleccione V para memorias y P para puntero. Apriete MENU para cambiar el tipo de datos, o apriete ENT para designar la memoria cuyos datos quiere ver o cambiar.

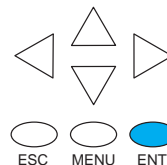
Valores de memoria V

Use la tecla de flecha a la derecha o la flecha hacia la izquierda para mover el cursor al dígito que usted quiere cambiar Use la tecla de flecha hacia arriba o tecla de flecha hacia abajo para cambiar el dígito. La dirección de memoria V se expresa como un número octal de modo que usted no verá 8's ni 9's.

Esta pantalla le permite ver dos localizaciones adyacentes de memoria V en formato BCD. La palabra más baja está a la derecha. Al apretar ENT se puede cambiar el valor del dato en esta pantalla; en la palabra más baja. En este nivel de la jerarquía de menú se puede usar también las tecla de flecha hacia arriba y hacia abajo para desplazarse a otras localizaciones de memoria.

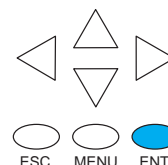
Paso 3.1

```
> M 2 : S Y S T E M   C F G .
> M 3 : M O N I T O R
```



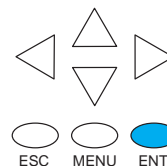
Paso 3.2

```
M 3 : > D A T A   M O N I T O R
      > B I T     M O N I T O R
```



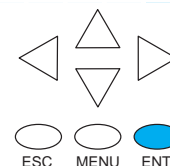
Paso 3.3

```
M 3 : D A T A   T Y P E   V
A D D R E S S   0 0 0 0 0
```



Paso 3.4

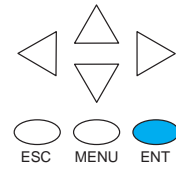
```
M 3 : D A T A   T Y P E   V
A D D R E S S   0 0 0 0 0
```



Paso 3.5

```
M 3 : V   1   V   0
V A L   0 0 0 0   0 0 0 0
```

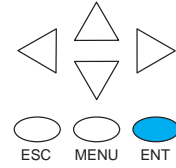
El valor del dato es de cuatro dígitos en la longitud para BCD/hexadecimal a menos que el bit 8 de V7742 esté ON. El bit 8 de V7742 cambia el formato de datos a decimal (cinco dígitos).



Paso 3.6

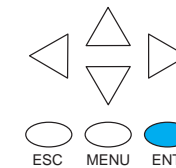
Use la tecla de flecha a la derecha o hacia la izquierda para mover el cursor al dígito que quiere cambiar. Use la tecla de flecha para arriba o hacia abajo para mover a otro dígito. El valor de la memoria V se expresa como un número BCD de modo que usted verá los valores (en el rango: 0 - F) disponibles para cada dígito. El formato de datos se puede cambiar a decimal poniendo el bit 8 de V7742 ON.

M	3	:	D	A	T	A				V					0		
			C	H	G	=		0	0	0	0			0	0	0	0



Paso 3.7

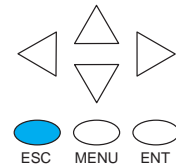
M	3	:	D	A	T	A				V					0		
			C	H	G	=		A	F	0	6			0	0	0	0



Paso 3.8

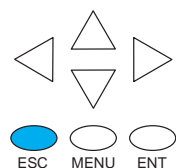
M	3	:	V					1		V					0		
			V	A	L			0	0	0	0			A	F	0	6

10



Vuelve al paso 1.1

M	3	:	D	A	T	A				T	Y	P	E			V	
			A	D	D	R	E	S	S				0	0	0	0	0



Vuelve a la pantalla por defecto

Apriete la tecla ESC cinco (5) veces para volver a la pantalla por defecto.

D	L	0	6		P	L	C			M	a	y		0	8
								1	5	:	0	2	:	1	3

Valores del puntero

Apriete ESC dos veces para volver a la pantalla del paso 3.3 con el cursor en la V, como es mostrado. Use la tecla de la flecha para arriba o para abajo para cambiar la V a P. Ahora se muestra el valor del puntero.

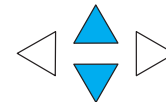
Use las teclas de la flecha para arriba o para abajo para cambiar el dígito correspondiente. Use la tecla de la flecha para la izquierda o la derecha para moverse de un dígito a otro.

En el paso 3.7a, puede usar la tecla de flecha para arriba y para abajo para cambiar palabras de datos. Cada vez que usted aprieta la tecla de flecha para arriba o hacia abajo, dirección de incrementa o decrementa en una palabra de 16 bits (las direcciones se expresan en octal).

Para cambiar del supervisión de datos a bit, apriete ESC tres veces para volver a la pantalla por defecto (Cinco veces para volver a la pantalla por defecto).

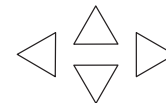
Vuelva al paso 3.3

M	3	:	D	A	T	A		T	Y	P	E					V
A	D	D	R	E	S	S					0	0	0	0	0	0



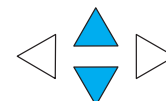
Paso 3.4a

M	3	:	D	A	T	A		T	Y	P	E					P
A	D	D	R	E	S	S					0	0	0	0	0	0



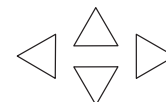
Paso 3.5a

M	3	:	D	A	T	A		T	Y	P	E					P
A	D	D	R	E	S	S					0	0	0	0	0	0



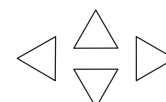
Paso 3.6a

M	3	:	D	A	T	A		T	Y	P	E					P
A	D	D	R	E	S	S					1	0	0	0	0	0



Paso 3.7a

M	3	:	D	A	T	A				P	1	0	0	0	0	0
			(V	0	0	0	0	0)		2	0	0	0	0



Vuleva al paso 3.3

M	3	:	>	D	A	T	A			M	O	N	I	T	O	R
			>	B	I	T				M	O	N	I	T	O	R

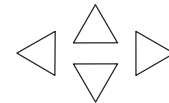
Cambiando fecha y hora

Menu 4, M4 : CALENDAR R/W

Desde la pantalla por defecto, apriete la tecla MENU cuatro veces para llegar al paso 4.1.

Paso 4.1

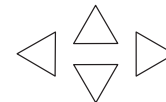
```
> M 3 : D A T A   T Y P E
> M 4 : C A L E N D A R   R / W
```



Paso 4.2



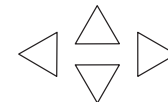
```
M 4 : D A T E   0 5 - 0 8 - 0 2
      T I M E   0 1 : 2 1 : 2 8
```



Paso 4.3



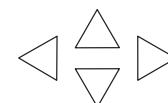
```
M 4 : > C H A N G E   D A T E
      > C H A N G E   T I M E
```



Paso 4.4



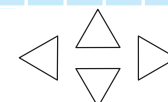
```
M 4 : D A T E   M M - D D - Y Y
      C H G =   0 5 - 0 8 - 0 2
```



Paso 4.5



```
M 4 : D A T E   M M - D D - Y Y
      S E T ?   0 5 - 0 8 - 0 2
```



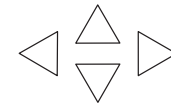
En el Paso 4.4, use las teclas flechas hacia arriba y hacia abajo para cambiar el valor por el mes, el día, o el año. Use de la tecla de flecha a la izquierda y a la derecha para moverse entre los dígitos diferentes en la fecha. Después que haga los cambios necesarios usando las teclas de flecha, apriete la tecla ENT para registrar los cambios.

Se le preguntará si usted quiere colocar la fecha al valor escogido. Apriete ENT una vez si la fecha es correcta. Usted volverá automáticamente al paso 4.2 y se muestra la fecha nueva.

Para cambiar la hora o el formato de fecha/hora, apriete ENT para regresar al paso 4.2 ,

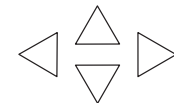
Vuelva al paso 4.2

M	4	:	D	A	T	E		0	5	-	0	8	-	0	2
			T	I	M	E		0	1	:	2	1		P	M



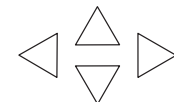
Use las teclas de flecha para arriba o para abajo o la tecla MENU para desplazarse en las diferentes opciones del menú.

M	4	:	>	C	H	A	N	G	E		D	A	T	E	
			>	C	H	A	N	G	E		T	I	M	E	



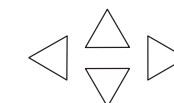
En este punto en nuestro ejemplo, nosotros cambiaremos la hora .

M	4	:	>	C	H	A	N	G	E		T	I	M	E	
			>	C	H	A	N	G	E		F	O	R	M	T



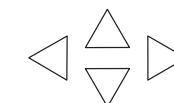
En el paso 4.4, use las teclas de flecha hacia arriba y hacia abajo para cambiar el valor de la hora, el minuto, o el segundo. Use las teclas a la izquierda y a la derecha para moverse entre los dígitos diferentes de la hora. Después de hacer los cambios necesarios usando las teclas de flecha, apriete la tecla ENT para registrar los cambios.

M	4	:	T	I	M	E		H	H	:	M	M	:	S	S
			C	H	G	=		1	3	:	5	3	:	3	2



Se le preguntará si quiere poner la fecha al valor escogido. Apriete ENT otra vez si la fecha está correcta. Usted volverá automáticamente al paso 4.2, y aparecerá la hora nueva.

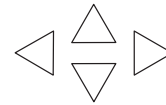
M	4	:	T	I	M	E		H	H	:	M	M	:	S	S
			S	E	T	?		1	3	:	5	3	:	3	2



Si quiere cambiar el formato de la fecha o la hora, vuelva al paso 4.2 y apriete ENT.

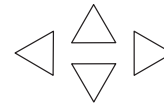
Vuelva al paso 4.2

```
M 4 : D A T E 0 5 - 0 8 - 0 2
      T I M E 0 1 : 2 1 P M
```



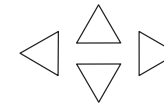
Apriete ENT, MENU, MENU para llegar a la selección del menú para cambiar el formato de fecha u hora. Apriete ENT de nuevo para llegar a la localización de cambio de formato.

```
M 4 : > C H A N G E F O R M T
      > C H A N G E D A T E
```



Apriete ENT nuevamente para entrar al formato de cambio de la fecha, o apriete MENU, ENT para cambiar el formato de tiempo.

```
M 4 : > D A T E F O R M A T
      > T I M E F O R M A T
```



En el paso 4.4, use las teclas de flecha para arriba o para abajo para desplazarse por los formatos de fecha. Las opciones son las siguientes:

MM-DD-YY (formato de EEUU)

DD-MM-YY (formato europeo)

YY-MM-DD (formato asiático)

Apriete la tecla ENT para salvar el cambio de formato.

```
M 4 : D A T E F O R M A T
      C H G = M M - D D - Y Y
```

```
M 4 : T I M E F O R M A T
      C H G = H H : M M : S S
```

Si ha escogido hacer un cambio de formato de hora, sus opciones son:

HH:MM US (12 horas 12:00 - 11:59AM/PM formato EEUU)

HH:MM AS (12 horas 00:00 - 11:59AM/PM formato asiático)

HH:MM:SS (formato de 24 horas)

Apriete la tecla ENT para salvar los cambios de formato. Apriete ESC hasta que la pantalla por defecto reaparezca. .

Formatos de variables de fecha y hora

_date:us	Formato EEUU	MM/DD/YY
_date:e	Formato europeo	DD/MM/YY
_date:a	Formato asiático	YY/MM/DD
_time:12	Formato de 12 horas	HH:MMAM/PM
_time:24	Formato de 24 horas	HH:MM:SS

Definiendo la contraseña y bloqueando cambios

Menu 5, M5 : PASSWORD R/W

El panel con visor LCD tiene su propia protección de contraseña separada de la protección de la "contraseña del PLC". Una contraseña del panel con visor LCD se puede usar para prevenir cambios no autorizados a la configuración de la hora y la fecha y también a valores de datos de memoria V. Las personas con autorización de contraseña pueden cambiar la hora, la fecha, los valores de memoria V, los bits, etc.

La contraseña LCD inhibe al personal no autorizado de modificar los datos en el DL06 con el teclado numérico de LCD. Aunque la contraseña de LCD se bloquea, el usuario puede modificar todavía los datos en el DL06 con *DirectSOFT* o el D2-HPP. El panel con visor LCD no permite usar una contraseña de varios niveles.

Sólo el menú 5 en el panel LCD puede modificar la contraseña LCD.

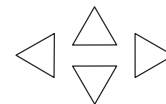


ADVERTENCIA: La protección de contraseña disponible en *DirectSOFT* o el programador portátil no previene cambios en el panel con visor LCD. Para prevenir los cambios con el panel de visor LCD, es necesario usar la contraseña de LCD.

Paso 5.1

Use la tecla MENU para navegar en la opción M5 del menú. Apriete ENT para llegar a la pantalla mostrada en el paso 5.2.

```
> M 4 : C A L E N D A R   R / W
> M 5 : P A S S W O R D   R / W
```



Paso 5.2

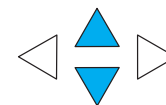
Cuando se coloca una contraseña y no se hace el bloqueo (Lock) del visor se permitirá tener acceso a todas las características del LCD.

Use las teclas de flecha hacia arriba o hacia abajo para cambiar entre PASSWD CHG? u LOCK/UNLOCK

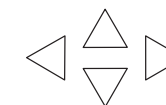
(es decir, cambio de contraseña o bloqueo o no).

8 ceros remueven la contraseña. Si la contraseña está compuesta de 8 ceros, el visor no se bloqueará

```
M 5 : > P A S S W D   C H G ?
      > L O C K / U N L O C K ?
```



```
M 5 : > P A S S W D   C H G ?
      > L O C K / U N L O C K ?
```



Capítulo 10: Panel de visor LCD

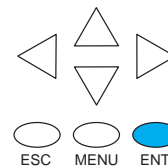
Use las teclas flecha hacia arriba o hacia abajo para desplazar por elecciones de número, y usar las teclas flecha hacia la derecha y hacia la izquierda para mover de una posición de un dígito a otro.

M	5	:	P	S	W	D	*	*	*	*	*	*	*	*
			C	H	G	=	0	0	0	0	0	0	0	0

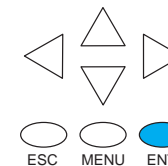


Nota: Es importante registrar la contraseña donde no se olvide y entregar la contraseña sólo a personal calificado. El acceso total al panel con visor LCD da acceso para cambiar los valores de datos dentro del PLC.

M	5	:	P	S	W	D	*	*	*	*	*	*	*	*
			C	H	G	=	2	1	7	0	8	3	0	3

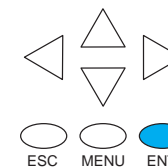


M	5	:	P	S	W	D	*	*	*	*	*	*	*	*
			S	E	T	?	2	1	7	0	8	3	0	3



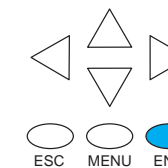
Vuelva al paso 5.2

M	5	:	>	P	A	S	S	W	D	C	H	G	?		
			>	L	O	C	K	/	U	N	L	O	C	K	?



No es posible bloquear el visor sin colocar una contraseña en el LCD. Es posible colocar una contraseña sin cerrar el visor, sin embargo, al no hacerlo no protegerá los datos sensibles. Apriete la tecla ENT en el Paso 5.2, y el visor se bloquea. Si usted no desea bloquear el visor en este punto, apriete ESC.

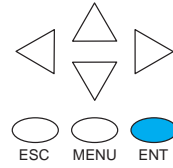
M	5	:	S	T	A	T	:	U	N	L	O	C	K	E	D
			E	N	T		T	O		L	O	C	K		



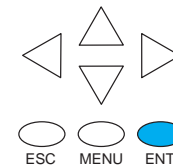
Antes asignar una contraseña, usted puede seleccionar "LOCK/UNLOCK" apretando ENT en el paso 5.2.

Vuelva al paso 5.2

M	5	:	>	P	A	S	S	W	D	C	H	G	?		
			>	L	O	C	K	/	U	N	L	O	C	K	?



M	5	:	S	T	A	T	:	U	N	L	O	C	K	E	D
			E	N	T	T	O	L	O	C	K				



Aquí, el mensaje en el visor le pide entrar una contraseña.

M	5	:	P	S	W	D	*	*	*	*	*	*	*	*
			L	O	C	K	0	0	0	0	0	0	0	0

Historia de errores

Menú M6: ERR HISTORY

De la pantalla por defecto, apriete la tecla MENU seis veces para llegar al paso 6.1.

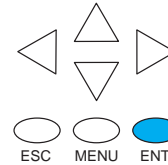
La pantalla de la Historia de Errores mostrará "NO ERROR" si no hay registro de errores. Si los errores han ocurrido, ellos pueden ser identificados por su Código de Error. La tabla del Código del Error (vea el apéndice correspondiente) explicará la fuente del mensaje de error. Se muestran los últimos 16 mensajes. Los mensajes de Error se desplazan cuando llega un nuevo mensaje de error.

Para revisar mensajes de error use la tecla flecha hacia abajo para desplazar por el registro histórico de mensajes de error.

Default screen

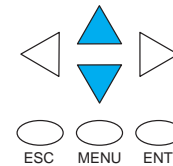
Paso 6.1

```
> M 5 : P A S S W O R D   R / W
> M 6 : E R R   H I S T O R Y
```



```
M 6 : E R R O R   H I S T O R Y
      N O   E R R O R
```

```
D i a g n o s t i c   E r r o r
E 4 * *   N O   P R O G R A M
```



```
M 6 : E r r .   0 5 - 2 2 - 0 2
      E 4 0 1   1 0 : 4 3 A M
```

Teclas de prueba, bascule la Luz y el zumbador (Beeper)

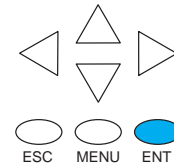
Menú 7, M7 : LCD TEST&SET

Esta selección de menú le da una oportunidad de:

- Probar cada tecla del panel LCD para asegurarse que el PLC recibe entradas apropiadamente
- Active el sonido del zumbador
- Active la iluminación trasera del LCD

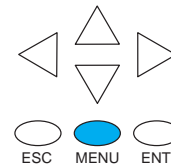
Haga una selección de menú apretando la tecla ENT.

>	M	6	:	E	R	R		H	I	S	T	O	R	Y	
>	M	7	:	L	C	D		T	E	S	T	&	S	E	T



Apriete ENT para entrar la PRUEBA de TECLA del LCD. Todas las teclas se pueden probar para la función apropiada en este menú.

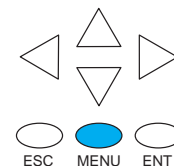
M	7	:	L	C	D		T	E	S	T	&	S	E	T	
			>	L	C	D		K	E	Y		T	E	S	T



Para volver al menú, apriete la tecla ESC dos veces o mantenga apretada la tecla ESC hasta que reaparezca el menú.

Apriete ENT para entrar el menú de prueba de iluminación trasera.

M	7	:	L	C	D		T	E	S	T	&	S	E	T	
			>	B	A	C	K		L	I	G	H	T		



El zumbador piezo-eléctrico se puede configurar para proporcionar la realimentación del pulsador.

M	7	:	L	C	D		T	E	S	T	&	S	E	T	
			>	B	E	E	P								

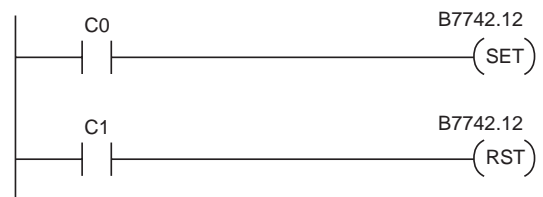
Los sufijos S, C0 y 0 alteran la presentación de ceros y de espacios principales. S remueve espacios a la izquierda y justifica a la izquierda el resultado. C0 substituye espacios a la izquierda por ceros. 0 es una modificación de C0. 0 elimina cualquier cero a la izquierda en la versión del formato C0 y los convierte a espacios.

Memoria reservada para el panel de visor de LCD

Dos registros de memoria V se reservan para hacer cambios a funciones del panel LCD vía lógica ladder. V7742 puede aceptar bits que se pueden cambiar en el diagrama ladder. El bit controla formatos de datos, la iluminación trasera y el beeper. Esto esta definido en la tabla en la próxima página. El otro registro reservado es V7743. Este registro se usa para escribir un mensaje de pantalla por defecto diseñado por el cliente para el LCD. Un programa de ejemplo para este propósito se ilustra luego en este capítulo.

Dirección de memoria V	Contenidos
V7742	Varias indicaciones LCD
	<ul style="list-style-type: none"> • Formato de fecha y hora • Menú de operación original de fábrica • Formato de datos de la supervisión de datos • Indicación de estado de contraseña LCD • Key press acknowledgement buzzer on/off setting • Configuración de la luz trasera
V7743	Localización del mensaje por defecto (escribiendo 0 a esta dirección vuelve el mensaje por defecto al valor original de fábrica)

El segmento de programa siguiente usa las instrucciones SET y RST para prender y apagar el bit 12 de V7742. Cuando C0 está ON, el bit 12 está prendido. El bit 12 prende el beeper o zumbador en el panel del visor de LCD. El contacto C1 repone el bit 12 al estado OFF.



Definiciones de los bits de V7742

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V7742	*	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit 1, 0	Formato de fecha del visor (Valor de fábrica = 00)		
	00, 11	=	Mes/Día/Año (Formato de los E.E.U.U.)
	01	=	Día/Mes/Año (Formato europeo)
	10	=	Año/Mes/Día(Formato asiático)
Bit 3, 2	Formato de hora del visor (Valor de fábrica= 00)		
	00, 11	=	HH:MM:SS (Formato 24 hour format)
	01	=	HH:MM PM/AM (Formato de 12 horas - Formato EE UU- 12:00 - 11:59)
	10	=	HH:MM PM/AM (Formato 12 horas Formato asiático - 00:00 - 11:59)
Bit 6 - 4	Configuración del menú por defecto (Valor de fábrica = 000)		
	000	=	La secuencia del menú por defecto, comienza secuencia del menú con el menú 1
	001	=	La secuencia del menú comienza con el menú 1
	010	=	La secuencia del menú comienza con el menú 2
	011	=	La secuencia del menú comienza con el menú 3
	100	=	La secuencia del menú comienza con el menú 4
	101	=	La secuencia del menú comienza con el menú 5
	110	=	La secuencia del menú comienza con el menú 6
	111	=	La secuencia del menú comienza con el menú 7
Bit 8	Formato de datos del visor (Valor de fábrica = 0)		
	0	=	Formato BCD/HEX (0000 - FFFF)
	1	=	Formato Decimal (00000 - 65535)
Bit 9	Re-escritura del nuevo mensaje (Valor de fábrica = 0)		
	0	=	El nuevo mensaje del LCD limpia ambas líneas de mensaje anterior
	1	=	El nuevo mensaje del LCD deja el mensaje anterior y sobrescribe el carácter especificado
Bit 11	LCD password status flag (Solo para leer)		
	0	=	La contraseña desbloquea
	1	=	La contraseña bloquea
Bit 12	Control del estado del indicador de señal sonora (Valor de fábrica= 0)		
	0	=	Señal sonora APAGADA
	1	=	Señal sonora ENCENDIDA (el LCD suena durante estado ON de esta indicación)
Bit 13	Control de la señal sonora del teclado (Valor de fábrica = 0)		
	0	=	Señal sonora APAGADA
	1	=	Señal sonora ENCENDIDA (señal suena cuando se presionan las teclas)
Bit 14	LCD back light setting flag (Valor de fábrica = 1)		
	0	=	Lámpara trasera del LCD apagada
	1	=	Lámpara trasera del LCD encendida
Bit 15	Indicación del estado de instalación del LCD (Solo para leer)		
	0	=	El LCD no está instalado
	1	=	El LCD está instalado

Cambiando los mensajes originales de la pantalla

Aparece la pantalla por defecto durante la energización. El mensaje por defecto es configurado en la fábrica pero puede ser cambiado por el cliente. Un método de personalizar el mensaje por defecto que aparece usa la instrucción VPRINT. La instrucción VPRINT se describe en el capítulo de instrucciones.

Mensaje de fábrica original

D	L	0	6		P	L	C			M	a	y	0	8	
								1	4	:	2	0	:	4	9

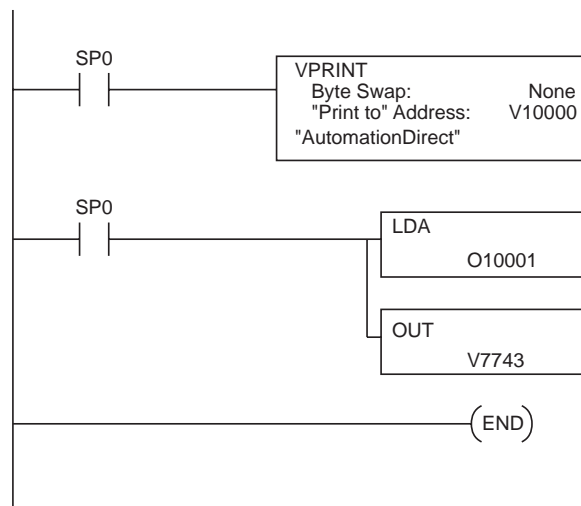
Programa ejemplo para colocar el mensaje de pantalla por defecto

El programa siguiente se puede usar para configurar el mensaje de pantalla por defecto. Este programa usa la instrucción VPRINT para cargar texto ASCII a una localización designada de memoria V y para mostrar la fecha actual.

Se usan las instrucciones LDA y OUT para apuntar a la memoria V (+1 ubicación) donde el texto se localiza. La memoria V7743 se reserva para el puntero al mensaje por defecto.



Nota: La instrucción VPRINT agrega un prefijo que no se imprime de una palabra (2 bytes). Por esta razón la instrucción LDA apunta a la memoria V10001 en vez de V10000.



V10000	00h	16h
V10001	u	A
V10002	o	t
V10003	a	m
V10004	i	t
V10005	n	o
V10006	i	D
V10007	e	r
V10010	t	c
V10011		
V10012		
V10013		
V10014		
V10015		
V10016		
V10017		
V10020		

Después de hacer funcionar este programa, apriete MENU, y luego ESC o apague y prenda el PLC. El mensaje nuevo por defecto debe verse como indicado. Vea las instrucciones del Menú 4 para cambiar la información de fecha y hora.

A	u	t	o	m	a	t	i	o	n	D	i	r	e	c	t



Nota: Es posible volver a la pantalla por defecto de la fábrica escribiendo 0 a V7743. .

Instrucción del panel con visor LCD del PLC DL06(LCD)

De la carpeta del proyecto *DirectSOFT*, use el navegador de instrucciones para localizar la instrucción LCD. Cuando seleccione la instrucción LCD y Hacer clic on OK, aparecerá el cuadro de diálogo LCD.

La instrucción LCD se inserta en el programa ladder con la caja de diálogo mostrada a la derecha. El diálogo se utiliza para especificar un mensaje que se exhibirá en la línea 1 o la línea 2 del panel LCD

S	I	U	D	G	E		P	I	T		A	L	A	R	M
E	F	F	L	U	E	N	T		O	V	E	R	F	L	O

Fuente del mensaje del visor LCD

El texto del mensaje puede generarse a partir de uno de dos lugares. Puede ser entrada directamente de la instrucción como secuencia de texto del literal (Vea la figura A), o puede generarse como texto ASCII almacenado en una localización de memoria V (Vea la figura B). En el último caso, es necesario especificar su dirección inicial y longitud de memoria V dentro de la caja de diálogo.

Las secuencias de texto de la pantalla pueden incluir datos de variables en la memoria del PLC. Cualquier valor de memoria V o de fecha y hora se puede colocar en el texto mostrado en el visor.

Figura A

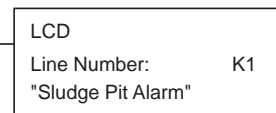
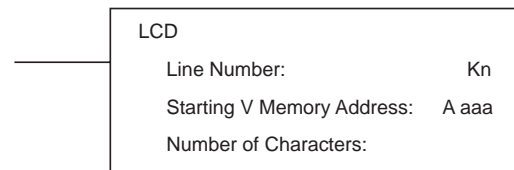


Figura B



Nota: La instrucción del panel con visor LCD es manejada por DirectSOFT, versión 4.0 o más nueva. No es posible usarla con el programador D2-HPP.

Códigos de caracteres ASCII

Los caracteres ASCII se pueden escribir directamente a las direcciones de la memoria V y después mostrar usando la instrucción LCD. La tabla a la derecha muestra el código BCD/HEX de dos dígitos para cada carácter disponible a ser mostrado.

Vea la lista completa en el apéndice G.

Ejemplo:

Para exhibir una A mayúscula, escriba el número hexadecimal 41 en la dirección de memoria identificada por la instrucción LCD.

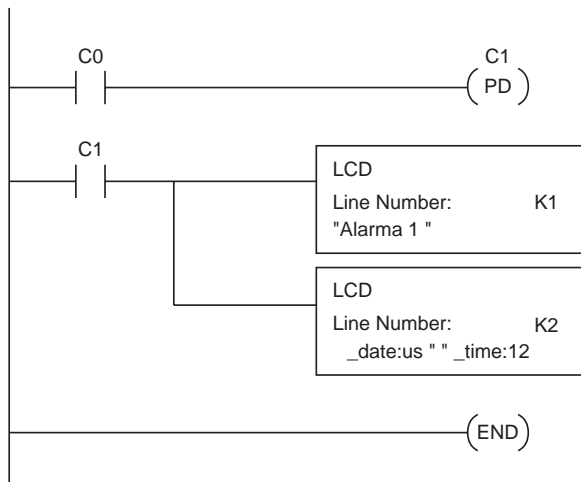
Conversión Caracteres ASCII (BCD/HEX)

		Primer dígito						
		2	3	4	5	6	7	
Segundo dígito	0	00	0A	0P	0Y	0F		
	1	!1	A0	a9				
	2	"2	B0	b9				
	3	#3	C0	c9				
	4	\$4	D0	d9				
	5	%5	E0	e9				
	6	&6	F0	f9				
	7	'7	G0	g9				
	8	(<8	H0	h9				
	9)9	I0	i9				
	A	*A	J0	j9				
	B	+B	K0	k9				
	C	,C	L0	l9				
	D	-D	M0	m9				
	E	.E	N0	n9				
	F	/F	O0	o9				

Programa ejemplo: Alarma mostrada con la la fecha y hora que ocurrió

El programa siguiente mostrará el mensaje "Alarma 1" en la línea K1 de la pantalla del visor y en la línea K2 la fecha y la hora

Se utiliza la instrucción diferencial positiva (PDd) o "one-shot", para se muestre el mensaje pero no bloquee otros mensajes u opciones del menú. Apriete la tecla MENÚ o ESC para que el texto del mensaje del alarma desaparezca



A	I	a	r	m	a		1												
0	5	/	0	8	/	0	2			5	:	2	3	P	M				

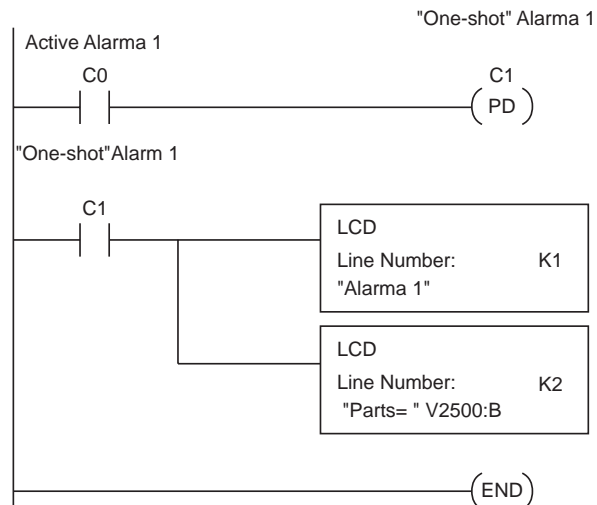
Programa ejemplo: alarma con datos embudidos de la memoria V

En este ejemplo, el texto de la notificación de alarma se muestra junto con el contenido de V2500. El sufijo "B" se agrega a la dirección de memoria (V2500:B) para hacer que los datos sean mostrados como número BCD.

En el primer ejemplo, el texto de alarma se carga directamente con la instrucción LCD. En el segundo ejemplo, el texto de alarma se carga en la memoria V y se usa la instrucción LCD para apuntar a ese texto.



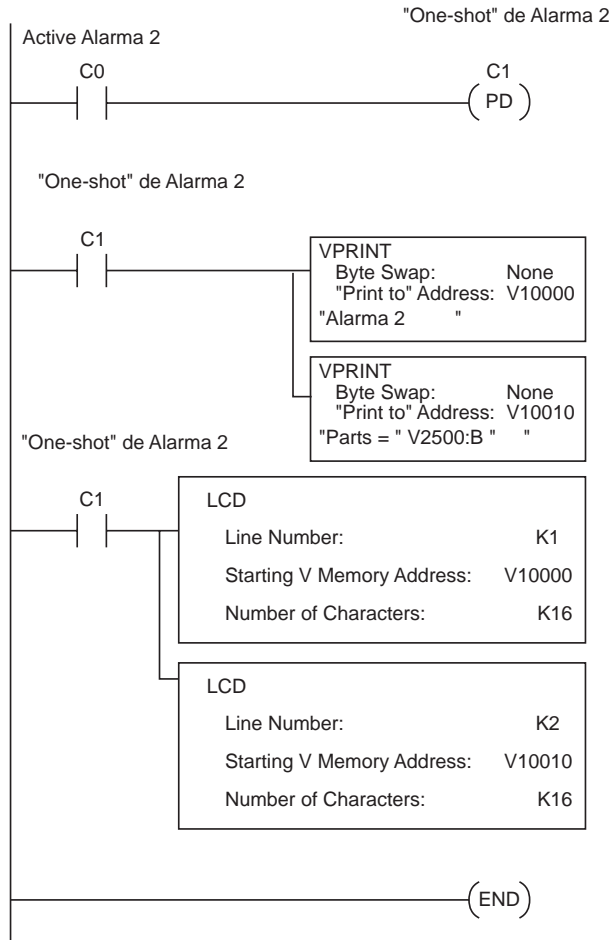
Nota: Al usar la instrucción LCD para mostrar V2000:R, hay un límite de tres caracteres de texto porque V2000:R utiliza 13 caracteres.



A	l	a	r	m	a		1										
P	a	r	t	s		=		2		4		3		7			

Programa ejemplo: Texto de alarma con datos embudidos de la memoria V

Este ejemplo de programa utiliza la instrucción VPRINT para escribir el texto ASCII (en la secuencia apropiada de caracteres) a V10000 y a V10010. La instrucción LCD se utiliza como un puntero a la dirección de memoria V en donde reside el texto para cada línea del visor.



A	l	a	r	m	a		2										
P	a	r	t	s		=		3	5	8	9						

10

RECOPIACIÓN Y EJEMPLOS



CAPÍTULO 11

En este capítulo...

Introducción	10-2
Que es memoria V	11-2
Familiarización con <i>DirectSOFT</i>	11-4
Conexión al PLC 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 E/S remotas	11-17
Concepto de una instalación controlada por PLC's	11-18
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 interface de operador	11-51
Ejemplo 5: Uso de E/S análogas	11-53
Ejemplo 6: Uso de comunicaciones seriales	11-55
Ejemplo 7: Uso de lazo de control PID	11-58

Introducción al capítulo 11

En este capítulo revisaremos todos los conceptos de los otros capítulos. *Este capítulo no existe en el manual en inglés.* Una vez que se hayan repasados conceptos generales, que también se aplican a otras familias de PLCs *DirectLOGIC*, mostraremos una selección de ejemplos explicados para ayudar a entender la técnica de programación a los nuevos programadores.

Puesto que usted está leyendo este capítulo, debe significar que ha leído el resto del libro y ahora usted tiene cierta experiencia con la programación de PLCs. Pero, puede ser que no haya sido nuestro PLC con el cual usted haya trabajado. Hay algunas cosas sobre PLCs *DirectLOGIC* que son diferentes que otros PLCs, y este capítulo puede ayudarle a entender nuestra orientación. Puede ser también que haya repeticiones de conceptos.

Que es memoria V

La memoria V en los PLCs *DirectLOGIC* son los registros, o palabras de 16 bits para todas nuestras familias de PLCs (excepto el 330/340 PLCs). Estos registros son referidos como dirección de sistema octal, es decir, no hay 8's o 9's. Por ejemplo, si usted trata de entrar la dirección de memoria V2019, usted tendrá un error.

Muchas direcciones de memoria V se pueden acceder también en los bits individuales de los cuales la palabra está constituida. El PLC DL06 puede utilizar lo que se llama **bit-of-word**, tal cual como las CPUs D0-05, D2-250-1, D2-260 o la D4-450. Esto significa que usted puede especificar el bit de una palabra particular para un dispositivo booleano, como un contacto normalmente abierto. De modo que el primer bit de V2000 sería mostrado como B2000.0.

Nota: El bit en la palabra no es octal, así que las opciones son B2000.0 hasta B2000.15. También, al entrar **bit-of-word** usted realmente entra V2000.0 y cuando usted presiona ENTER el código cambia a B2000.0, de modo que usted pueda decir que es una dirección de bit-of -word.

La configuración del PLC es hecha de preferencia con el programa *DirectSOFT*. Existe la posibilidad de bajar el programa desde nuestro sitio de Internet en forma gratuita, para efectos de demostración y éste trabaja normalmente con la única limitación de que sólo se puede programar hasta 100 palabras. El manual de *DirectSOFT* está incluido en el "folder" o carpeta HELP del programa de demostración y se encuentra cuando ya se haya instalado el programa.

¡Las instrucciones por defecto en los PLCs *DirectLOGIC* usan números BCD! El apéndice J detalla más sobre el sistema numérico BCD.

El concepto a recordar aquí es que usted debe usar instrucciones matemáticas que correspondan a sus datos. Ocurre frecuentemente que se tiene una instrucción LD, luego una operación matemática, y luego una instrucción OUT en otra dirección de memoria V. Muchos clientes llaman a apoyo técnico quejándose de que el resultado que están consiguiendo es el mismo número que fue cargado con LD originalmente. Esto es generalmente debido a que uno o más de sus números están en el formato incorrecto para la instrucción usada. En *DirectSOFT* la manera más rápida de determinar en qué formato están datos específicos de las direcciones de memoria V es abrir una ventana de **Data View** y ver el mismo dato en varios formatos hasta que



usted encuentra el correcto. Cualquier número que no sea una configuración de bits legítima en BCD será tratado como hexadecimal por el PLC para propósito de exhibición en el formato BCD/Hexadecimal.

Tenemos detallado más adelante un ejemplo para ver los datos en **Data View** en *DirectSOFT* en los varios formatos disponibles. Usted puede entrar la misma dirección de memoria V en varias líneas en **Data View** y cambiar cada uno a un formato diferente. Éste es un método excelente para entender las diferencias entre formatos.

Los temporizadores y contadores en el PLC DL06 ven el tiempo y cuentan en formato BCD.

El valor de referencia (Set point, en inglés) de PID, la variable de proceso y la salida son todos los números binarios/decimales.

Muchas direcciones de memoria V son direcciones dedicadas del sistema que no pueden ser cambiadas por código y tienen funciones especiales, como por ejemplo, hora y fecha. Un listado de estas direcciones de memoria comienza en la página 3-30 del manual.

Bits X (entradas)

Los bits X corresponden a las entradas físicas en los módulos del PLC. Las entradas análogas pueden usar bits X para obtener datos traducidos a las direcciones de memoria V, dependiendo del tipo de PLC. Los bits X pueden también ser vistos o manipulados como palabras. Vea la página 3-32 del manual y las tablas de bits de X y de Y. Note que a la derecha de X0-X17 dice V40400. V40400 se compone de los bits de entradas X0-X17. Y también observe que estos bits son OCTALES, que no hay allí ningún ocho o nueve.

Bits Y (salidas)

Los bits Y corresponden a las salidas físicas en los módulos del PLC. Éstos son iguales que los bits de X, excepto que la referencia de la dirección de memoria V comienza en V40500.

Bits C (relevadores de control)

Los relevadores de control son los que utiliza el programador para los relevadores internos en sus programas. Este PLC tiene 1024 de éstos, lo que es más que suficiente para muchas de las aplicaciones. Vea la página 4-35 del manual. Aquí vemos la dirección para los bits C, notamos que son referidos en formato octal y que pueden también ser referidos por la dirección de memoria V comenzando con V40600. Esta funcionalidad es muy útil para muchas de nuestras interfaces de operador que escriben a datos a nivel de bits a las direcciones de palabra.

Bits SP (Estado)

Todos nuestros PLCs, (excepto D3-330/340), comparten un sistema bastante estándar de bits de estado. Éstos tienen funciones que se extienden desde un bit siempre ON (SP1) a bits de estado para los puertos de comunicación de PLCs (SP112). Hay una lista de estos relevadores de propósitos especiales en el apéndice D de este manual.

Bits Tx y CTx

Estos bits son contactos de se activan cuando los temporizadores y los contadores llegan al valor de actuación o valor predefinido. Tienen la misma definición que la instrucción de que se está manejando.

Alimentación de energía

Le recomendamos que utilice una protección de sobretensión para el funcionamiento de la computadora con *DirectSOFT5*. Un protector de sobretensión protegerá su computadora contra la mayoría de las sobretensiones. Sin embargo una fuente de alimentación continua (UPS) proporcionará la mejor protección. Una UPS proporciona aislación completa entre la fuente de corriente alterna y la computadora y tiene reserva de batería para las condiciones de falta de energía y de baja tensión.

Familiarización con *DirectSOFT*

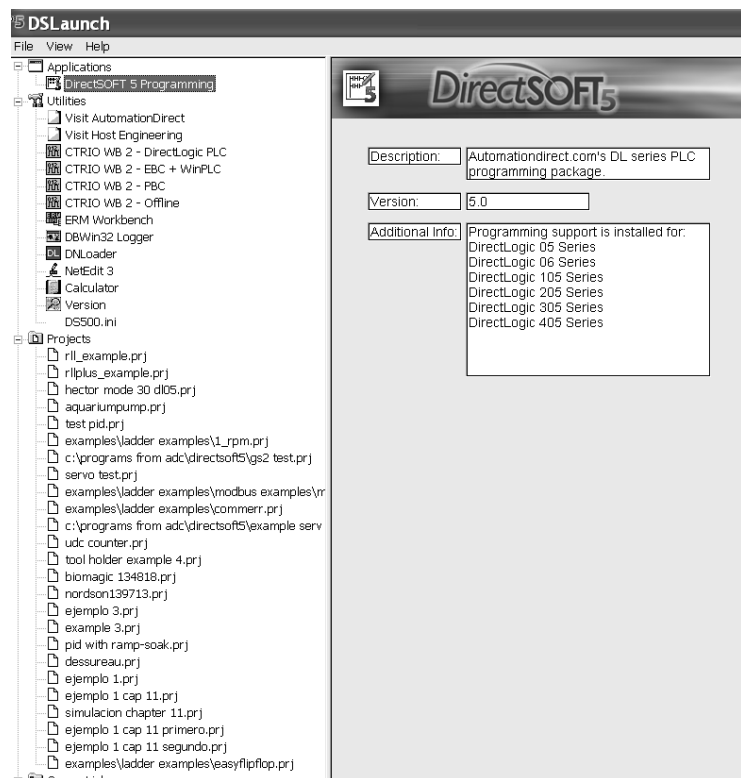
El software de programación *DirectSOFT5* funciona con el sistema de operación Windows 2000, XP y Vista y se usa para configurar el PLC DL06. Tome un momento para estudiar el manual de referencia de *DirectSOFT*. Verifique los requisitos al elegir su configuración de PC. Una vez que esté instalado el software de programación *DirectSOFT* en su computadora, usted deseará comenzar a usarlo. Le recomendamos leer el manual del programa, que está como archivo en formato pdf dentro del folder HELP en *DirectSOFT*.

Antes de comenzar a hacer o corregir un programa, usted necesita abrir *DirectSOFT*. Haga clic en Start (INICIAR) en la esquina más baja a la izquierda del monitor de la computadora. Luego vaya a **Programs**, coloque el cursor en *DirectSOFT5*, luego haga clic en **DSLlaunch** (con la figura de un cohete) en el submenú. Aparecerá la siguiente ventana de **DSLlaunch**. De esta ventana, se pueden partir utilidades adicionales, por ejemplo, CTRIO WB, ERM Workbench, etc., a partir de un lugar central. Este mismo lugar se utiliza para crear y para manejar programas del PLC y las comunicaciones entre su computadora personal y el PLC.

Note las diversas áreas en la ventana de la figura adyacente.

Applications Éstas son las aplicaciones instaladas actualmente en *DirectSOFT*. Son visibles en el menú bajo la carpeta (folder) o icono **Applications** y se conectan a las aplicaciones que se han diseñado para iniciarse desde *DirectSOFT*. Por ejemplo, para crear un nuevo programa haga clic doble en *DirectSOFT programming*.

Utilities Hay varias utilidades disponibles bajo el folder/carpeta **Utilities**. Algunas se pueden comprar en **AutomationDirect**,



por ejemplo, *KEPDirect*. Otras utilidades vendrán con el software de programación *DirectSOFT*. Estas utilidades son *ERM Workbench*, *CTRIO Workbench* y *NetEdit3*.

Projects Se crean proyectos en *DirectSOFT*. Un proyecto (también llamado un documento) es el nombre colectivo para su programa y toda su documentación. Cuando usted crea un nuevo proyecto, o trabaja en un proyecto existente, usted verá una enumeración en el menú bajo el folder/ícono **Projects** por nombre. Los documentos se enumeran en el orden "más recientemente usados".

Comm Links Los "enlaces" o links sirven para establecer parámetros para los puentes de comunicaciones entre su PC y uno o más PLCs. Los enlaces no son solamente para los programas de control. Realmente son puentes de comunicaciones (es decir, el enlace entre la computadora y la impresora). Cualquier aplicación puede usar el enlace. Cuando usted crea enlaces, aparecerán en el menú bajo el folder o ícono **Comm links** (enlaces de comunicación).

Conexión al PLC desde una computadora con *DirectSOFT*.

Normalmente Ud. se conectará a una computadora con *DirectSOFT* con un cable serial D2-DSCBL, conectado al puerto 1 del PLC. (Puede ser hecho también a través de Ethernet). El switch de selección de modo en el PLC debe estar en **TERM**.

Para poder establecer la comunicación debe usar el **Link Wizard**, que se activa desde el diálogo **DS Launch**, haciendo clic en el archivo **COMM LINKS**, como se muestra en la figura adyacente.

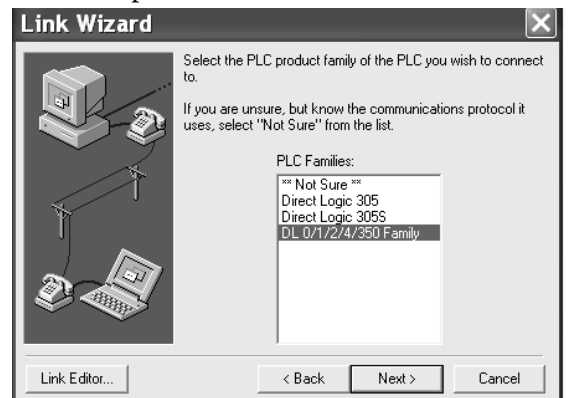
Al hacer esto, aparece un pequeño mensaje que dice **ADD LINK** (que significa Agregue una conexión). Haga clic nuevamente sobre él y aparecerá el diálogo de la figura adyacente.

Esto le permite seleccionar el puerto de comunicación de la computadora. Normalmente es **COM1**. Luego de seleccionarlo, haga clic en **NEXT>**.

Si usa un adaptador de USB, puede ser que obtenga otro puerto en la computadora. Vea que es necesario hacer, en ese caso, en el manual del programa *DirectSOFT*.

Luego viene otro cuadro de diálogo, que le permite seleccionar la familia del PLC.

Seleccione el grupo de familias **DL 0/1/2 4/350 only** y luego haga clic en **NEXT>**.



Aparecerá el próximo cuadro de diálogo que le permite seleccionar el protocolo y la dirección del nodo.

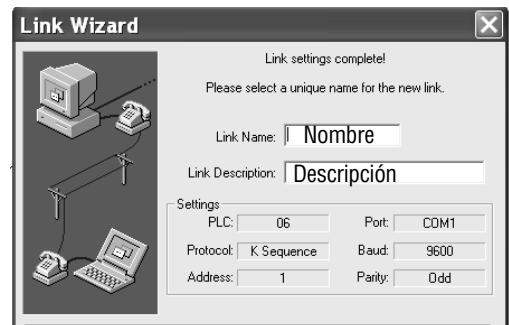
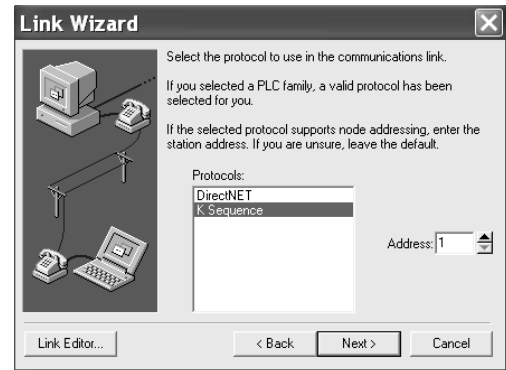
Seleccione **K-sequence** en este caso. Use el nodo 1 que es el valor por defecto y luego haga clic en **NEXT>**.

Esto hará que la computadora encuentre el PLC. Si hay errores, la computadora le informará sobre eso. Vea más detalles en el manual de *DirectSOFT*. Normalmente no hay ningún problema de conexión.

Por último, coloque un nombre al enlace como se muestra en la figura adyacente.

Cada enlace debe tener un nombre único. Puede tener hasta 16 caracteres. Haga clic en el botón **FINISH** para volver a la ventana de *DSLlaunch*. Es creado un archivo con el nombre que Ud. asignó, y queda debajo de la carpeta **Comm Link**.

Al lado izquierdo del archivo aparece una luz, que puede ser verde, amarilla o roja. Si está verde, el enlace está activo.



Comenzar a crear o modificar un programa

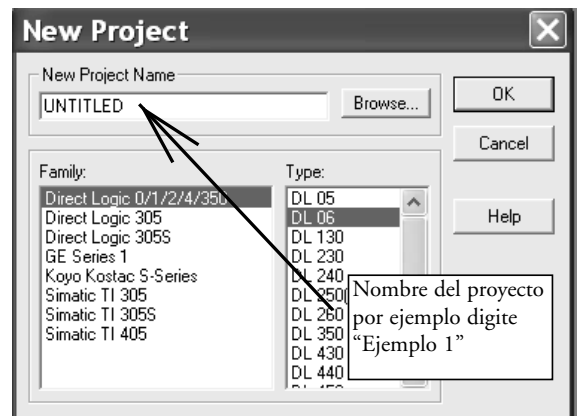
Los pasos siguientes le mostrarán los pasos básicos para modificar programas con *DirectSOFT*. Esto le dará los fundamentos para comenzar a usar de modo que usted pueda modificar un programa.

Paso 1: Entre en el modo Program

Para comenzar el programa *DirectSOFT*, haga clic en el icono *DSLlaunch5*; aparece un diálogo cuya parte izquierda muestra una lista de carpetas; Para comenzar un nuevo programa (proyecto), haga clic doble en *DirectSOFT Programming*, situado en la carpeta **Applications** del menú.

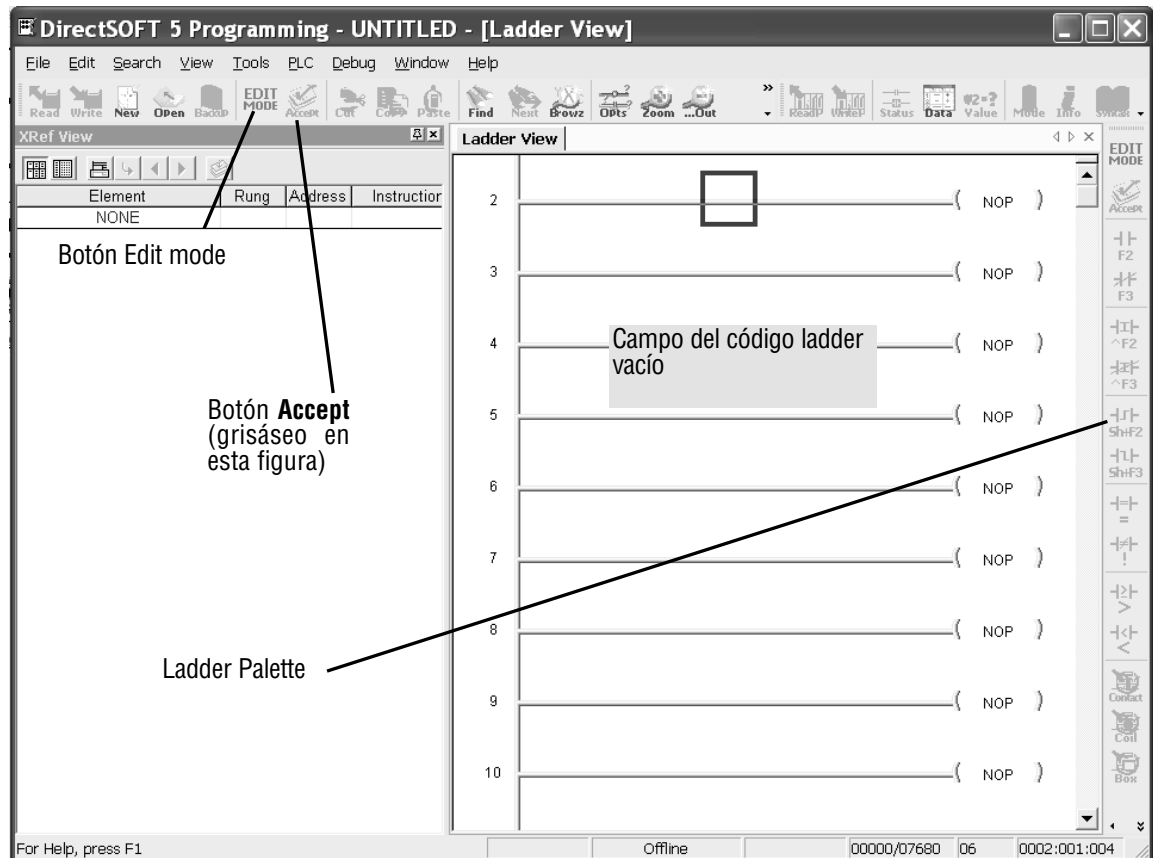
Paso 2: Comience un nuevo proyecto

Usted debe ahora ver la ventana **New Project**. Usted puede nombrar un proyecto usando cualquier combinación de 15 caracteres (incluyendo espacios). "EJEMPLO 1" es el nombre del proyecto a ser usado para este ejemplo. Mueva la barra de selección a la **PLC Family** (familia del PLC) y al **CPU Type** (tipo de CPU). Haga clic en **OK** después de usted ha hecho sus selecciones de familia y tipo. Para este ejemplo, use el grupo de PLCs que pertenece a las familias DL05/06/105/DL205/DL405. Tenga presente



que las mnemónicas disponibles, las reglas de proceso e incluso las características de la barra de la herramienta se adaptan a la selección de familia y tipo que usted hizo.

Después de hacer clic en OK para entrar el nombre del proyecto, usted verá los renglones de lógica ladder listos para ser modificados, tal como en la figura de abajo.



Éste es solamente el modo de leer en este punto (pero no modificar). En este modo, el cursor se presenta siempre vacío y no se puede hacer programación. Solamente se puede observar un proyecto. Si usted es programador "con experiencia", puede ser que no le guste la forma de presentación.

Esto sería un buen momento para seleccionar las opciones de colores. Por favor, vea al manual de usuario de programación del software *DirectSOFT*, para configurar el aspecto de la ventana de programación.

Paso 3: Vaya al modo EDIT

El modo de modificación o **Edit Mode** se utiliza para escribir el programa de control. Usted tiene la opción de entrar el modo de modificar el programa de tres maneras, siendo lo más común hacer clic en el botón **Edit Mode** en la barra superior de herramientas. Será un botón con letras rojas y fondo transparente. Otra manera de cambiar el modo de modificación es hacer clic en la barra de menú superior en **EDIT** y luego seleccione **Edit Mode**. La última manera de entrar en el modo de corrección es apretar las teclas **Ctrl + E** (presione las teclas control y E simultáneamente).

DirectSOFT indicará el modo de modificación activo (Edit mode) cuando la caja del cursor se convierte en sólida y el botón **Edit Mode** se hace blanco. La herramienta **LADDER PALETTE** también aparecerá en la parte lateral de la ventana de programación, la cual muestra símbolos de programación, tal como un contacto normalmente abierto.

Paso 4: Usando la Ladder Palette para entrar el primer elemento en el renglón

La **Ladder Palette** puede ser muy útil, especialmente al comienzo mientras aprende a usar el programa ladder con *DirectSOFT*. Más adelante, usted puede preferir utilizar las teclas especiales (**Hot Keys**). Estas se muestran en cada botón de símbolos y aparecen siempre cuando el cursor está en el botón de símbolos (por ejemplo, F3 coloca un contacto normalmente cerrado).

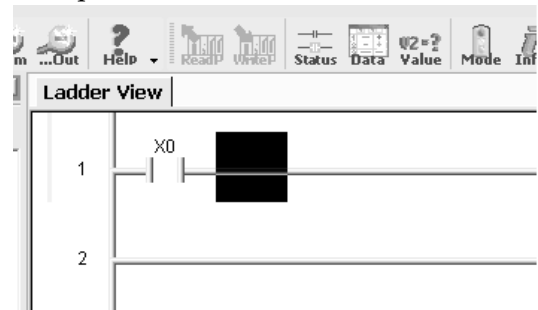
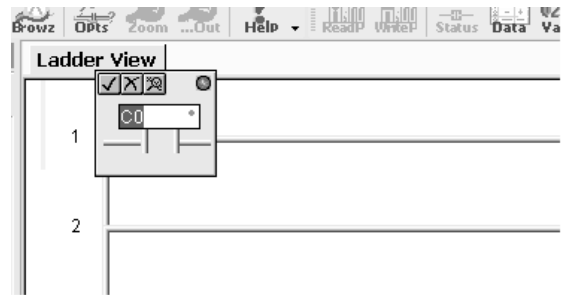
Vea el manual de usuario de programación del software *DirectSOFT* para más detalles. Use la **Ladder Palette** para incorporar la primera instrucción del programa. Primero, mueva el cursor a la localización deseada para el primer elemento. Esto se hace con el mouse o con las flechas en el teclado hacia arriba y hacia abajo. Al usar el mouse, coloque simplemente la flecha del mouse en el punto donde usted quiera que el elemento sea puesto y haga clic el botón de mouse izquierdo.

En este ejemplo, será colocado un contacto normalmente abierto en la primera posición respecto al renglón 1. Coloque el cursor al principio del renglón y haga clic en el símbolo “contacto normalmente abierto” en la **Ladder Palette**.

Paso 5: Entrando entradas del tipo X

Usted verá que el cursor cambia a una caja de diálogo con un contacto abierto, una ventana con el cursor de texto que centellea en la dirección **C0** (destacada) y un indicador verde. Si el punto verde del elemento cambia al rojo, significa que la dirección es incorrecta, inválida o un carácter incorrecto. Por ejemplo, si usted teclea la letra **O** en vez del dígito **0**, el indicador se torna rojo y permanecería rojo hasta que usted corrija su error. Entre **X0** mientras **CO** está destacado. Después de que haya entrado la dirección y el indicador de error es verde, haga clic en la marca de verificación (✓) o presione la tecla **ENTER**.

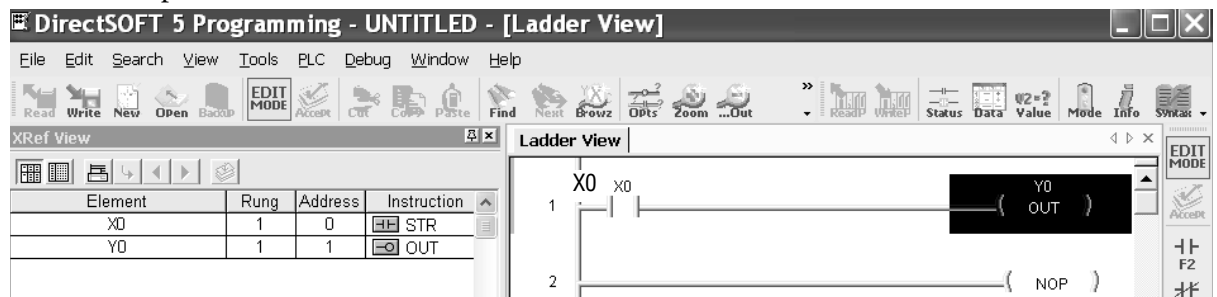
Se ha entrado la instrucción y el cursor se ha movido a la posición siguiente de entrada. Note la barra vertical amarilla que aparece al lado del renglón. Puede ser que éste no sea un manual en colores, pero se ve una barra vertical coloreada en el ejemplo de la pantalla. La barra amarilla indica que se ha entrado una instrucción o instrucciones, pero que el programa no se ha aceptado (compilado) de modo que usted pueda salvar el programa revisado al disco duro. Los renglones que se han aceptado ya en memoria compilada tendrán una barra verde en lugar de amarilla. Sin la compilación, usted no verá los iconos para **Save to Disk** (Salvar al disco duro) o **Save to PLC** (Salve al PLC).



Esto significa que para salvar su programa en el disco o en el PLC, usted tendrá que compilar el programa apretando **ACCEPT** primero. Por ejemplo, si usted desea parar de trabajar *DirectSOFT* ahora, usted primero debería compilar todos los renglones

Paso 6: Entrando salidas del tipo Y

Después, mueva el cursor al extremo del renglón, usando las teclas **CTRL + la flecha ->** del teclado, sobre la instrucción **NOP**. Haga clic en el botón **BROWSE Coils** (Navegar las bobinas) en **Tool Palette**. El **instruction browser** aparecerá con la bobina estándar seleccionada por defecto. Haga clic en **OK** para entrar una bobina estándar.

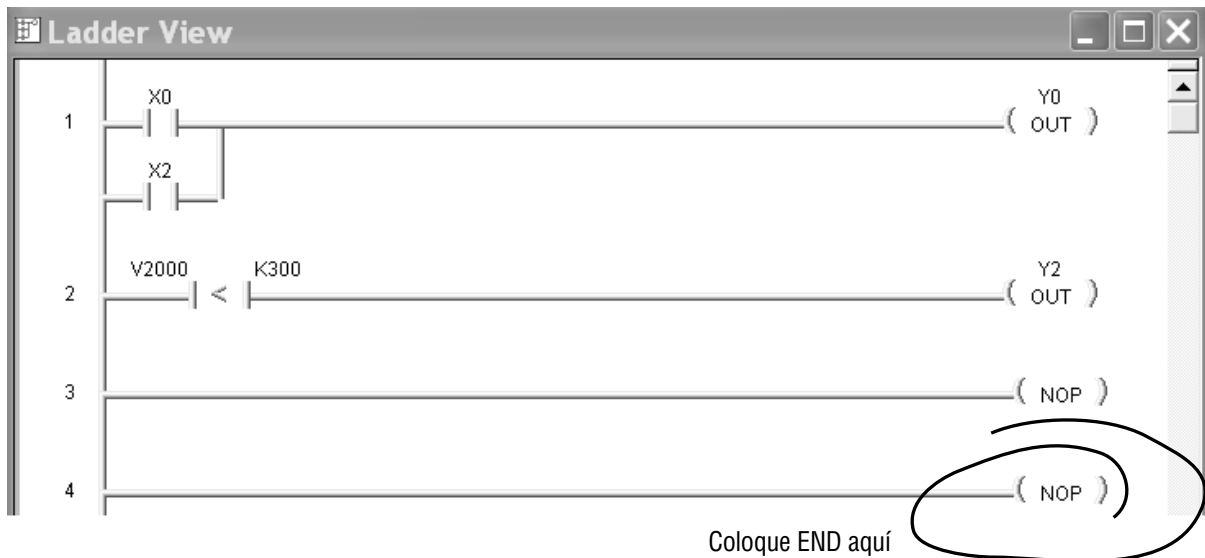


Paso 7: Cuadro de diálogo de entrada del elemento

El navegador de instrucciones será substituido por la caja de la entrada del elemento. Note que la instrucción se llama **OUT**. Esta es la bobina **OUT** (Hay otra instrucción **OUT**, que es de caja). La dirección por defecto, **C0**, estará destacada. Teclee **Y0 > Enter**. Cuando la dirección es entrada correctamente, el indicador de error estará verde. Si quiere agregar otro contacto en paralelo con el primero, apriete la tecla **Enter** y se abrirá un espacio debajo de este renglón. Allí Ud. puede colocar otro contacto, como **X2**, de la misma forma. Luego cierre el circuito usando las teclas **CTRL + flecha para arriba** simultáneamente. Acaba de programarse el renglón 1. Ud. puede programar cuantos reglones sean necesarios para su programa. Apriete el botón **Accept**. De la misma forma se programan instrucciones caja, tales como **LD** o **MUL** u otras.

Digamos que quiere ahora entrar una comparación de un valor en **V2000** con una constante en el renglón 2. Ud. va colocar el cursor en el segundo renglón y hacer un clic en la **Ladder Palette**, en el símbolo de **menor que** (**-|<|-**) o apretar la tecla **<**. La caja de diálogo le presenta dos campos; en el primero Ud. coloca **V2000** (donde dice **TA0**) y en el segundo campo coloque **K300**, siendo **300** la constante. Mueva el cursor a la derecha y allí puede colocar una bobina **Y1**, por ejemplo, haciendo clic en donde dice **Coil**. Nuevamente aparecerá el diálogo **Instruction Browser**. Esta instrucción compara el contenido de **V2000** con la constante **3000**. Si la comparación es verdadera, la bobina **Y1** será activada.

Estos renglones se podrían descargar o bajar al PLC, excepto que falta un elemento. El programa **siempre** se debe terminar con un renglón con la bobina **END**. de otra forma, tendrá un error al compilar. Vea como queda la programación en la figura de la próxima página, antes de colocar la instrucción **END**.



Paso 8: Entre la bobina END

Para programar este renglón, mueva el cursor de modo que quede sobre la instrucción **NOP** en el renglón 4, y haga clic en el botón **COIL**. Aparecerá la ventana **Instruction Browser**. Mueva las flechas hacia arriba o hacia abajo o use el mouse para seleccionar **Program Control** en la sección de **COIL CLASS** (clase de bobina) de la ventana. **END** estará al comienzo de la lista de las bobinas y estará destacado. Haga clic en **OK** para entrar el elemento. Acepte el programa. Hay ahora varios renglones programados en este ejemplo. Este programa se puede descargar a un PLC de la forma que está o, si desea, se pueden agregar renglones adicionales al programa. La bobina **END** debe estar al final del programa. Continúe practicando lo que se ha discutido antes de continuar. Esta es una de las formas de escribir el programa en *DirectSOFT*.

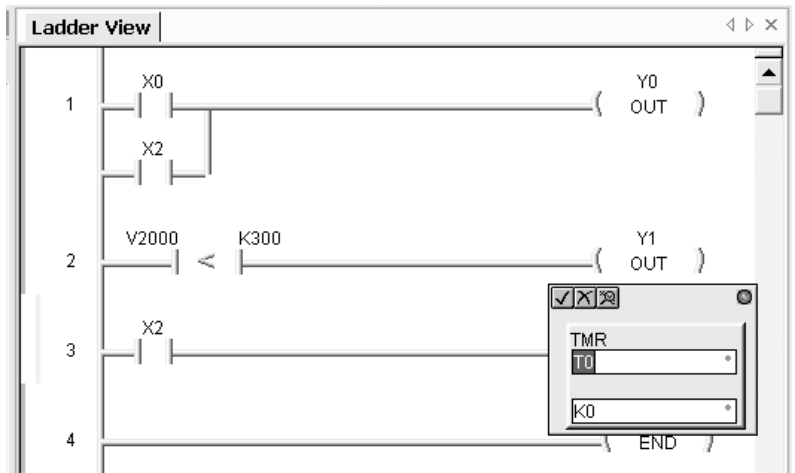
Paso 9: Entre un temporizador

En este ejemplo, será colocado un contacto normalmente abierto en la primera posición del renglón 3, como condición para activar el temporizador. Coloque el cursor al principio del renglón y haga clic en el símbolo “contacto normalmente abierto” en la **Ladder Palette** (o apriete F2).

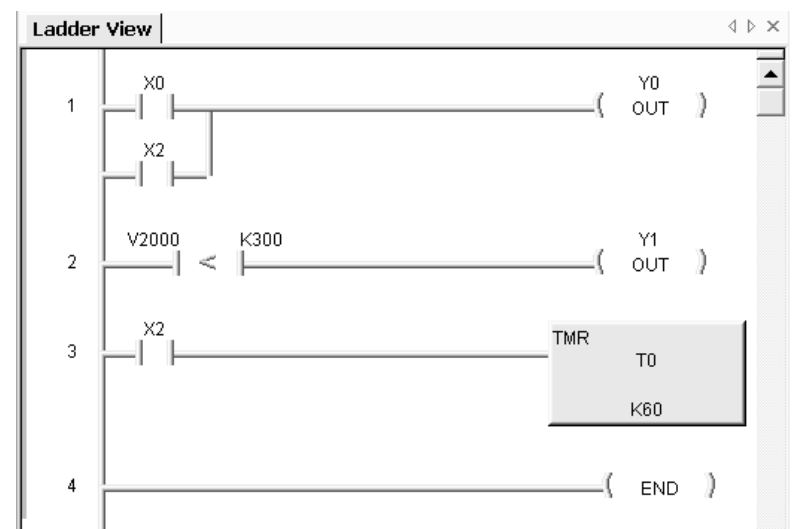
Usted verá nuevamente que el cursor cambia a una caja de diálogo con un contacto abierto, una ventana con el cursor de texto que centellea en la dirección **C0** (destacada) y un indicador verde. Entre **X2** donde **C0** está destacado. Después de que haya entrado la dirección y el indicador de error es verde, haga clic en la marca de verificación (✓) o presione la tecla **ENTER**.

Para programar un temporizador de 10 segundos, mueva el cursor de modo que quede sobre la instrucción **NOP** en el renglón, y haga clic en el botón **Box** en la **Ladder Palette**. Aparecerá la ventana **Instruction Browser**. Mueva las flechas hacia arriba o hacia abajo o use el mouse para seleccionar **Timer/Counter/SR** en la sección de **BOX CLASS** (clase de box) de la ventana. **TMR** estará en la lista de boxes; haga clic para destacarlo. Haga clic en **OK** para entrar el elemento. Aparecerá el diálogo mostrado en la figura de la próxima página.

Note que hay dos campos a ser llenados. El superior determina la identificación del temporizador (T0, en este caso). El campo inferior define el tiempo que toma en activarse el contacto T0. Cada unidad es de 0,1 segundos, de modo que para hacer 6 segundos debe escribirse allí el valor 60, o K60, siendo K el método normal de indicar una constante. Este campo también puede tener una variable, un registro tal como V4000. Después de escribir estos valores, acepte esta instrucción haciendo **Enter**. El programa quedará como en la figura adyacente. Haga clic en el botón **ACCEPT** en el menú barra de herramientas para compilar el programa. Este programa se puede descargar a un PLC de la forma que está o, si desea, se pueden agregar renglones adicionales al programa. Note que la instrucción **END** está colocada al final del programa.



Continúe practicando lo que se ha discutido antes de continuar. Esta es una de las formas de escribir el programa en *DirectSOFT*.



Paso 10: Acepte para compilar y luego salve el programa

Continuaremos con este ejemplo para hacer las cosas simples. El programa necesita ser aceptado para ser descargado al PLC. Note que los dos botones de diskette a la izquierda del menú barra de herramientas que permiten leer desde el disco (**READ FROM DISK**) o escribir al disco (**WRITE TO DISK**), no están "grises".



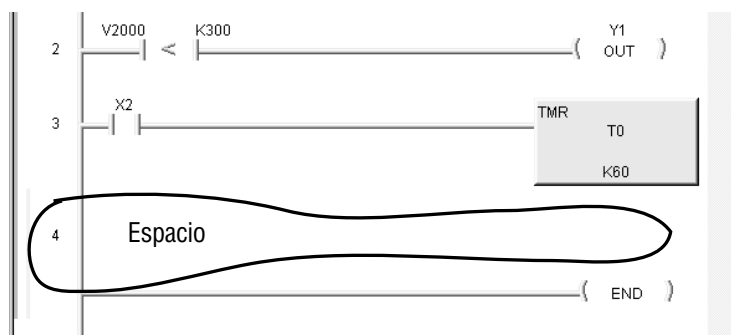
En este caso, usted deseará hacer clic el botón de escribir para salvar el programa al disco duro (no es necesario salvar el programa al disco para descargar el programa a un PLC). Es una buena práctica salvar su trabajo mientras modifica un programa. Puede haber una equivocación ocasionalmente y puede desear restaurar el programa al estado que estaba antes de que sucediera la equivocación. Para hacer esto, se puede hacer clic en el botón READ (leer) y el programa previamente salvado aparecerá en la pantalla y la programación puede continuar.

Paso 11: Programe un contador

En este ejemplo, vamos a abrir un espacio entre el renglón 3 y el 4. Para eso, coloque el cursor al comienzo del renglón 4 y apriete la tecla **Enter**. Esta acción abre un espacio y muestra el número 4 arriba del renglón que tiene la instrucción END. Luego será colocado un contacto normalmente abierto en la primera posición del renglón 4, como el pulso para activar el contador. Coloque el cursor al principio del renglón y apriete **F2**.

Usted verá nuevamente que el cursor cambia a una caja de diálogo con un contacto abierto, una ventana con el cursor de texto que centellea en la dirección **C0** (destacada) y un

indicador verde. Entre **X3** donde está **CO**. Después de que haya entrado la dirección y el indicador de error es verde, presione la tecla **ENTER**.



Para programar un contador de hasta 9999 pulsos, haga una línea apretando las teclas **CTRL+flecha ->** simultáneamente y haga clic en el botón **Box** en la **Ladder Palette**. Aparecerá la ventana **Instruction Browser**. Mueva las flechas hacia arriba o hacia abajo o use el mouse para seleccionar **Timer/Counter/SR** en la sección de **BOX CLASS** (clase de box) de la ventana. **CNT** estará en el primer lugar de la lista de boxes; haga clic para destacarlo. Haga clic en **OK** para entrar el elemento. El contador tiene un contacto de reset. Use **X4** para volver el valor de conteo a cero. El valor corriente del contador se ve en **V1000**. para el contador **CT1** estará en **V1001**, y así sucesivamente. Esta acción programa el contador. Si Ud necesita contar más de 9999, considere usar el condador **UDC**, que puede contar hasta 99999999.

Paso 12: Acepte para compilar y luego salve el programa

Continuaremos con este ejemplo para hacer las cosas simples. El programa necesita ser aceptado para ser descargado al PLC. Note que los dos botones de diskette a la izquierda del menú barra de herramientas que permiten leer desde el disco (**READ FROM DISK**) o escribir al disco (**WRITE TO DISK**), no están "grises". En este caso, usted deseará hacer clic el botón de escribir para salvar el programa al disco duro. Es una buena práctica salvar su trabajo mientras modifica un programa.

Ejercicios para el nuevo programador:

¿Ud. entiende el direccionamiento de PLCs, no es cierto? Que bien, porque ahora usted va a poder escribir un programa. Como primera acción, puede ser que sea provechoso leer algunas notas que tenemos para usar *DirectSOFT* y que se encuentran en nuestro sitio de Internet. Hay documentos en inglés que le permiten conocer más de *DirectSOFT*, entre ellos:

AN-DS-002 es un documento que se ocupa de varios problemas de localización de averías que hemos podido documentar para *DirectSOFT*.

AN-DS-003 es un documento que trata de como comenzar y como usar *DirectSOFT*. Revise este documento en su totalidad. Muchos de los trucos de allí le ahorrarán tiempo y le ayudarán a solucionar problemas más eficientemente.

Es necesario también que lea el manual de los módulos de señales análogas D0-OPTIONS-M-SP.

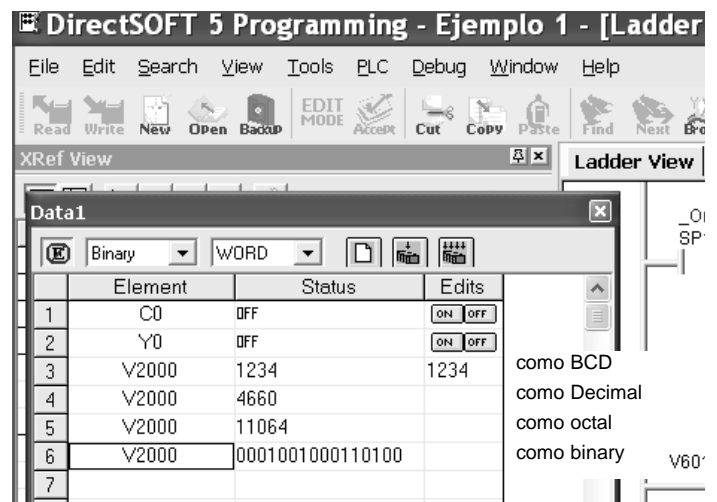
Ahora que usted está poco más familiarizado con el software, escribamos (o hagamos lógica) de lo siguiente (Damos una solución de algunas de estas tareas):

- Conéctese con el PLC y luego abra Data View, haciendo clic en el icono “Data”, y entre V2000 en varias líneas.

Cambie el formato en cada línea a un formato diferente y luego entre forzando datos a V2000 en varios formatos y observe las diferencias. Observe la figura adyacente.

Data View tiene normalmente 2 columnas, pero puede hacer que aparezca la columna 3 haciendo clic en el icono amarillo con una letra E.

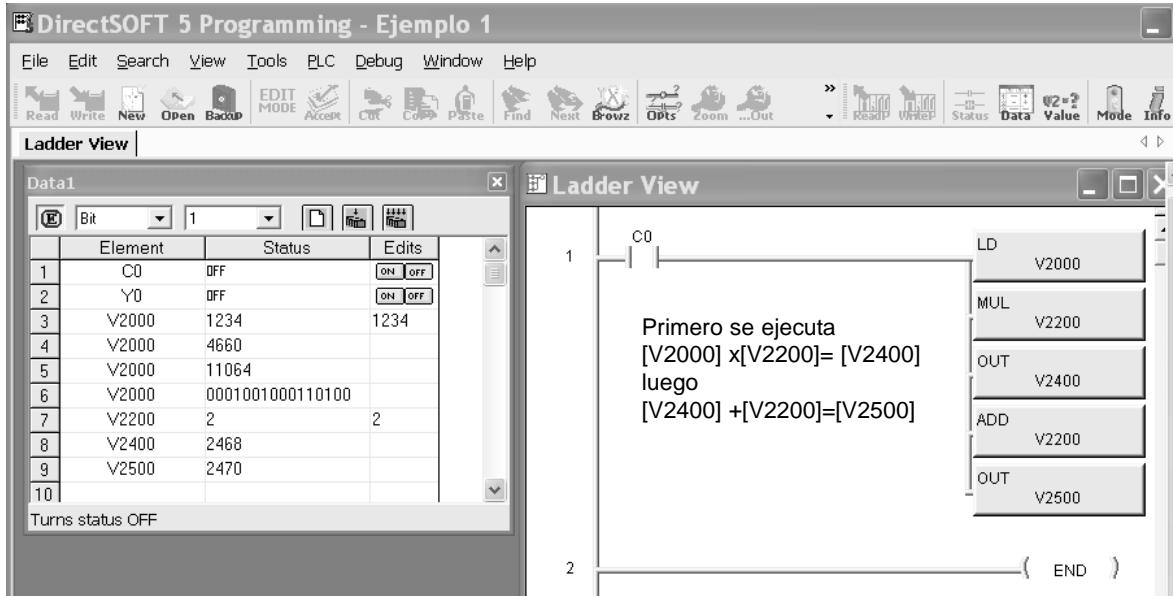
Como puede observar, el valor 1234 en BCD puede ser visto también como 4660 en formato decimal, o como 11064 en formato octal o aún en binario, como secuencia de ceros y unos.



- Haga la misma cosa para las instrucciones binarias/decimales y nuevamente para instrucciones reales (punto flotante).
- Escriba el código para cambiar el formato de datos de varios formatos uno a otro. Use las instrucciones BIN, BCD, BTOR, y RTOB para cambiar datos de formato a formato y viceversa.
- Escriba lógica usando instrucciones BCD tal como MUL y ADD y fuerce los valores V2000 y V2200 correctamente ajustados a formato BCD así como también datos incorrectamente ajustados a formato y vea los resultados. Accione C0 desde la columna Edit. Vea la figura de la página siguiente. Cada vez que acciona C0 para ON, el PLC ejecuta la operación.

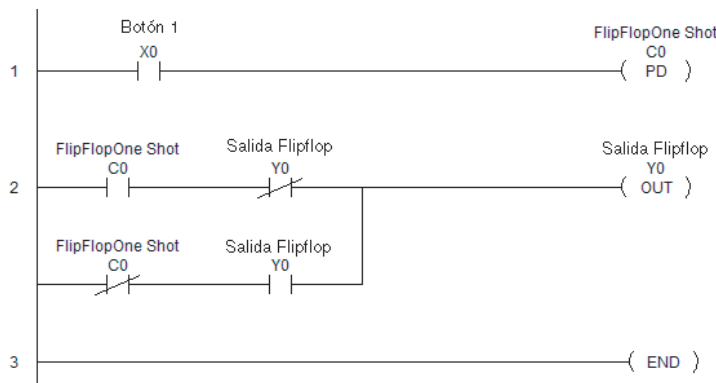
La operación funciona de la siguiente manera: cada vez que el renglón se hace verdadero, la instrucción LD (load) coloca el valor de V2000 en el acumulador del PLC. Luego la

instrucción MUL (multiplicación en BCD) toma el valor en el acumulador y lo multiplica por el argumento de la instrucción MUL (V2200 en este caso). Luego la instrucción OUT lee lo que está en el acumulador y lo copia el argumento de la instrucción (V2400 en este caso). Luego la instrucción ADD suma en BCD el valor del argumento (V2200 en este caso) con el contenido del acumulador y lo coloca nuevamente en el acumulador. Por último, la



instrucción OUT copia a la memoria del argumento de OUT lo que está en el acumulador

- Escriba el código para ejecutar un temporizador T6 en que la salida sea activada cuando la entrada C10 se activa. El temporizador debe comenzar a contar el tiempo cuando la entrada se apaga. La salida debe apagarse cuando el temporizador llega al tiempo prefijado. Este temporizador se conoce en inglés como OFF-DELAY TIMER. Hay dos ejemplos en el sitio de Internet de **AUTOMATIONDIRECT**>Tech support>Example programs> documento EP-MISC-012. Esta área es una buena fuente para obtener ejemplos.
- Escriba la lógica para hacer que un botón de entrada encienda una salida cuando se aprieta una vez y luego apague la salida cuando se apriete nuevamente. Esto es el circuito **flip flop**. Asegúrese de que sean eliminados los rebotes del contacto de modo que al apretar el botón no haga que el circuito complete un ciclo. Usted tal vez quiera usar la bobina PD (One shot) y contactos diferenciales (o un pulso en la transición de OFF para ON o “one shot”).



- Escriba el código para hacer que la instrucción Shift register (SR) cambie la posición de un bit repetidamente desde C0 hasta C16. Utilice el relevador especial SP4 (con ciclos de 1 segundo) para producir el cambio. Cada vez que un bit se cambia en C16, C0 debe activarse otra vez. Éste es un método de generar un circuito que completa un ciclo, útil para renglones de lógica de comunicaciones y otras características.
- Escriba la lógica usando un contador que vuelva a cero cada 16 conteos. Use el relevador especial SP4. Esta lógica se puede también usar como un circuito que completa un ciclo. Algunos clientes prefieren el SR y algunos prefieren contadores.

Resumen de las particularidades de las familias de PLCs *Direct*LOGIC

- Las instrucciones aritméticas por defecto son BCD, y hay instrucciones que pueden modificar el número para otros formatos, tales como binario o real.
- Las funciones aritméticas trabajan con el acumulador de modo que usted tiene que cargar un valor en el acumulador antes de que usted pueda realizar una operación en ella en un bloque de aritmética. El resultado residirá en el acumulador, de modo que debe mover el resultado a una dirección de memoria con una instrucción OUT.
- Los valores prefijados de temporizadores y contadores deben tener el formato BCD. Además, los valores actuales de temporizadores y contadores están en formato BCD (incluso las constantes).
- Los temporizadores TMRA y contadores UDC toman (2) palabras consecutivas cada uno, de modo que si usted tiene un TMRA T0 usted no puede tener un T1.
- Las instrucciones DRUM (TAMBOR) toman cuatro direcciones consecutivas de contadores.
- Las instrucciones de temporizadores, contadores y tambores no se deberían usarse en lógica de subrutinas o de etapas. Su comportamiento puede ser comprometido por estar localizado en etapas o subrutinas.
- Las asignaciones de entradas y salidas en todos los PLCs (excepto la familia 305) serán asignadas de izquierda a derecha por el tipo de módulo. De modo no hay diferencia en la asignación para 2 módulos de entradas y luego 2 módulos de salidas versus entradas-salidas-entradas-salidas. *Por esta razón se recomienda NO dejar ranuras vacías a la izquierda de cualquier módulo de entradas y salidas*, ya que al agregar un módulo del mismo tipo a la izquierda de un módulo cambiará las asignaciones de entradas y salidas de todos los módulos que vienen a continuación.
- Los módulos análogos tendrán asignadas X e Y incluso si se programan usando el método del puntero.

Programación de entradas y salidas análogas con PLCs *Direct*LOGIC

Ahora usaremos entradas y salidas análogas. Obtenga un módulo de entradas y de salidas de voltaje tal como el F0-2AD2DA-2. Éste es más fácil de usar para conectar uno al otro puesto que los módulos de voltaje no tienen tipo de salidas surtidoras/drenadoras. Usted no podría conectar una señal no aislada de 4-20 mA con una entrada de 4-20 mA en nuestros PLCs a menos que fuera un módulo aislado como algunos de nuestros módulos 4-20 mA. Esto puede también ser un problema al conectar con otros dispositivos, y nuestros clientes tienen con

frecuencia este problema.

La mayoría de nuestros módulos análogos requieren que se alimente 24VCC a dos terminales en el módulo.

También, cada módulo de señales análogas debe ser configurado por ladder.

Hay dos tipos de configuración para los módulos análogos en nuestros PLC's DL06; con IBOX y con el método del puntero.

La instrucción IBOX es la más fácil. También, el PLC DL06 usa el método del puntero. **¡El método del puntero también es fácil!**

Los módulos análogos reciben una asignación de bits X o Y de la misma forma que son hechos en los módulos discretos. Vea el manual de módulos análogos D0-OPTIONS-M-SP para más información.

En el método del puntero, la programación es simple. Usted programa un solo renglón para cada módulo. Este renglón contiene la información que le dice al PLC cuántos canales usted desea, donde usted desea los datos localizados y en qué formato usted desea los datos.

Una advertencia es que los números de la ranura comienzan en 1 y 1 es la ranura más a la izquierda en el PLC, mirando el PLC por la frente (en el PLC DL06; esto no es válido para PLCs de otras familias). Obviamente, si se selecciona la dirección dedicada incorrecta porque se tiene el módulo en una ranura diferente, las entradas no trabajarán.

Una palabra sobre números negativos

¡No hay números negativos en BCD! En BCD los números negativos son representados generalmente activando el bit más significativo de los 16 bits para ese canal. Esto tiene el efecto de sumar una constante 8000 al número. De modo que 8035 es el negativo 35.

En números decimales (o binarios, como se define en el manual), los números negativos se indican con el complemento de 2, que es el número está invertido (todo los 1s cambian a ceros, a todos los ceros cambian a 1s) y luego se suma un valor binario de 1. Esta conversión es también necesaria si usted desea exhibir un número negativo en un panel táctil (o touch screen). Sepa que la función PID considera números negativos como magnitud más signo. En magnitud más signo, el bit más significativo de la palabra decimal que está encendido indica un número negativo. El rango es entonces -32767 a 32767.

Entonces, aquí están sus tareas sobre módulos análogos:

- Conecte un canal de salida de su módulo de salida de voltaje con un canal de entrada de su módulo de entradas de voltaje. Escriba un programa que permite que todos los canales en ambos módulos tengan los datos del BCD.
- Vaya a Data View en *DirectSOFT* y envíe los datos al canal de la salida que usted ha conectado al canal de entrada y observe los datos recibidos para ese módulo en **Data View**.
- Abra el circuito del canal o lleve la señal fuera del rango y observe el resultado.

Una palabra sobre programación de PID

¿Qué hace el control PID? Aprenda sobre control PID en el capítulo 8 de este manual. No hay programación de PID en nuestros PLCs, ya que realmente no hay una instrucción lógica para el PID. En nuestros PLCs el PID es una función que está activada o desactivada. Usted puede enviarle datos de lógica y puede también activar el o los lazos con la programación.

Aquí están los principales asuntos para el PID:

- No se olvide de entrar un valor máximo de salida diferente de cero, que es el valor por defecto; si usted no lo hace, el PID parecerá que está trabajando, pero la salida nunca cambiará.
- No cambie los bits de cambio de modo manual/auto/cascade permanentemente. Estos bits necesitan ser encendidos con un pulso one shot. El código activa básicamente el bit SET y después el PLC cambia modo y resetea el bit.
- El lazo PID necesita que la variable de referencia (Set point) y de proceso estén en formato binario o decimal (magnitud más signo si es negativa). La salida PID está también en decimal. Cerciórese de que su código análogo esté instalado como decimal o agregue el código convertido desde BCD a decimal antes de enviar al PID.
- El modo **bumpless 1** cambiará el SP para hacerlo igual al PV al cambiar al modo de funcionamiento. Esto incomoda a clientes que no están escribiendo permanentemente a su registro de SP. Usted puede desactivar esto seleccionando el modo **bumpless 2**.
- **Bias Freeze** le dice al PID que pare de calcular un **bias** adicional si la salida está en el máximo o el mínimo (Esto debería estar activado por defecto). Y los clientes con lazos de control PID con una rampa de larga duración necesitan activar esto para estar en el lado seguro.
- Cada lazo **PID** necesita tener un valor específico de ganancia (GAIN) diferente de cero o de otra forma nunca funcionará. La ganancia debe ser ajustada durante la sintonización (tuning) junto con el valor de Reset y de Rate.
- Vea un ejemplo de PID en el ejemplo 7 de este capítulo.

El PLC DL06 puede tener entradas y salidas remotas

El PLC DL06 tiene dos puertos seriales, que permiten hacer comunicación serial con el protocolo MODBUS RTU como maestro o esclavo. Esta función le permite añadir más de 2000 entradas y salidas remotas hasta una distancia de hasta 1000 metros con el sistema de cableado RS-422 or S-485, disponible en el puerto 2 del PLC. Evidentemente esto también se puede instalar en el mismo gabinete del PLC DL06. Un esclavo es, por ejemplo, otro PLC DL06 o un PLC DL05 o un variador de frecuencia GS2-25P0, que son declarados esclavos.

También **Automation Direct** tiene un producto modular llamado Terminator I/O, que es un sistema de entradas y salidas no inteligentes, pero que acepta el protocolo MODBUS como esclavo hasta una velocidad de transmisión de 38.4 KBaud y puede aceptar fácilmente hasta 90 nodos. Vea más informaciones de Terminator I/O en nuestro sitio de Internet.

Concepto de una instalación controlada por PLC's.

El proyecto de control de una máquina o un sistema de equipos a ser controlado por uno o más PLCs es tarea de uno o más técnicos, quienes deben entender los detalles de operación de la máquina o el sistema. La programación del PLC es una tarea, entre muchas otras tareas, en el diseño de un sistema, que debe ser ejecutada para la implementación de un proyecto. Así como se requiere estudios para definir el proceso, para escoger la mejor solución, para seleccionar cada una de las piezas que componen un sistema, también es necesario planear la selección del sistema de control con la parte de operación (HMI), uno o más PLCs, sensores, contactores, transductores, válvulas solenoides, motores y actuadores entre otro equipo. y después como será alambrado el sistema, cuántas y qué tipo de entradas y salidas se requieren.

No es el propósito de este capítulo describir cada una de las etapas anteriores a la programación de un PLC; sin embargo, describiremos aquí en general acciones que es necesario implementar para tener un sistema de control con PLCs hecho en forma exitosa.

Cualquier sistema de control tendrá como entradas: comandos hechos por un operador, entre otras, que son hechos a través de sistemas llamados paneles de operador o, más en general, interfaces hombre máquina (H. M. I.), definición de los estados de la maquinaria o de un sistema, tales como el estado de motores (funcionando o parados), conmutadores para determinar si una determinada máquina puede funcionar antes que otra, sensores de proximidad que determinan la posición de una determinada parte de la maquinaria, sensores fotoeléctricos que detectan luz, para contar u otras funciones, dependiendo de lo que se cruce frente a estos sensores, conmutadores que indican si el nivel de agua está más bajo que un cierto nivel prefijado, confirmación de válvulas solenoides activadas, velocidad de la maquinaria, posición de ejes o de discos, temperaturas, presiones, niveles, corrientes, etc. y salidas discretas como señales para hacer funcionar un motor girando en un sentido o el otro, activar válvulas, lámparas de indicación de estado, sirenas para llamar la atención de un determinado paso o partida de motores o salidas análogas para actuación de variadores de frecuencia, válvulas para actuadores de control de flujo, aparatos de potencia de control de potencia para controlar calentamiento o refrigeración, e incluso transferencia en tiempo real de informaciones para sistemas que adquisición de datos.

La programación de un PLC es una de las últimas etapas de un proyecto de control, ya que primeramente se tiene que definir cada uno de los componentes del PLC y cada uno de los sensores, contactores, el alambrado a cada uno de los sensores, a los paneles de operador, a los centros de controles de motores, a los variadores de frecuencia, a los encoders para medir velocidad o posición etc.

Tomemos por ejemplo el caso de un sistema simple de control de una planta de cemento; una planta de cemento tiene típicamente unos 300 a 500 motores, unas 150 mediciones de temperatura, presión, flujo, unos 1000 switches de diferentes tipos, unas 200 válvulas solenoides y otros componentes los cuales deben ser comandados, debe verse el estado de la mayoría de ellos, y debe tenerse un sistema para eso. Aquí los PLCs juegan un papel importante en el control de este sistema. Cualquier defecto en una o más de las máquinas significa que hay que tomar acción inmediata mente para evitar parar la planta y perder producción o incluso dañar otras máquinas.

Considerando toda la información descrita anteriormente, estos son los pasos básicos para poder hacer un desarrollo del programa.



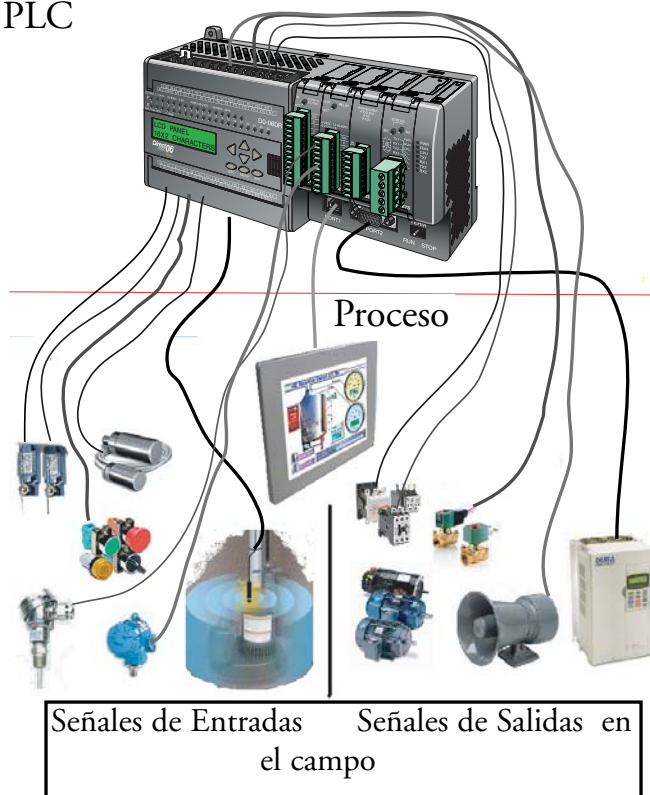
ADVERTENCIA: Es su responsabilidad diseñar un ambiente seguro de operación para el personal y el equipo y debe ser su meta primaria durante la planificación de sistema e instalación. Los sistemas de automatización pueden fallar y poder tener como resultado situaciones que pueden causar heridas graves al personal o daño al equipo. No confíe solamente en el sistema de automatización para hacer seguro el ambiente de operación. Usted debe usar artefactos electro-mecánicos externos, tales como conmutadores de límite, independientes de la aplicación del PLC para suministrar protección para cualquier parte del sistema que pueda causar heridas o daños personales. Cada aplicación de automatización es diferente, así que puede haber requerimientos especiales para su aplicación particular.

Etapa 1 Descripción sumaria de alto nivel de la aplicación

Tenga un documento con una primera descripción de su aplicación; debe hacerse en los términos más sencillos posibles (descripción en texto, una lista de instrucciones, un diagrama de flujo o cualquiera que le sea cómodo). Evite la implementación de detalles específicos en este punto. Si usted no puede describir su aplicación, usted no puede PLC programarla. Si su aplicación es grande y compleja, desglóselas en partes más pequeñas.

Casi todos programas de aplicación tienen tres componentes: Entradas, Proceso, Salidas.

- ¿Cuales son sus entradas en la aplicación? En el puerto serial, entradas físicas, memoria del PLC.
- ¿Qué proceso se debe hacer? Analice la serie de datos, cálculo de valores, construya una cadena de enclavamiento, etc.
- ¿Dónde van las salidas de la aplicación? activan salidas como motores, válvulas solenoides, envían datos procesados por vías de comunicación, datos a ser enviados por el puerto serial, leen y escriben valores a memorias del PLC, imprimen datos a resultados de cálculos, etc.



Etapa 2 Revise su descripción

Lea su primera descripción.

- ¿Le hace sentido a usted?
- ¿Tiene definidos los componentes correctos del PLC?
- ¿Se han tomado en cuenta consideraciones para hacer el sistema seguro para el personal y el equipo?
- ¿Es la velocidad de procesamiento un factor a considerar?

- ¿Dónde están los cuellos de botella en la aplicación?
- ¿Hay una manera más sencilla de alcanzar los mismos resultados?
- ¿Hay una manera más rápida de alcanzar los mismos resultados?
- ¿Cuales son las consideraciones o requisitos para esta aplicación?
- ¿Han sido alcanzadas las consideraciones o requisitos especiales para esta aplicación?

Si las preguntas han causado cambios en partes de la aplicación, repita los pasos 1 y 2 hasta que usted esté satisfecho y luego proceda a la etapa siguiente.

Etapas 3 Descripción de nivel eléctrico de la aplicación

Refine su primera descripción de la aplicación con detalles específicos usando los términos más sencillos (descripción en texto, una lista de declaraciones, un diagrama de flujo o cualquier otro que haga que Ud. se sienta cómodo). Debe tener claro toda la parte de hardware que será usada.

Etapas 4 Codifique y arme una aplicación de prueba.

En esta etapa Ud. comenzará a trabajar con el o los PLCs. Asegúrese que conoce bien el PLC y todas sus características y configuraciones. Escoja y verifique los módulos de entradas y salidas adecuados para cada función. Entienda todas las opciones de alambrado. Entienda las opciones para programación. Entienda la necesidad de hacer mantención y búsqueda de problemas en el sistema cuando éste ya esté trabajando.

Cada programa tiene varias instrucciones que se deben incluir. El programa puede trabajar sin estas instrucciones pero es buena práctica siempre que programe incluir estas instrucciones, porque le pueden salvar tiempo en la depuración de un programa durante la implantación. Documente la codificación, es decir, coloque apodos en cada entrada y salida, y haga comentarios en cada rung según la lógica que ud está usando.

Etapas 5 Escriba código para el próximo componente del programa de la aplicación.

Mientras más sencillo es el componente, mejor. Documente el programa mientras lo ejecuta.

Etapas 6 Pruebe todas las combinaciones posibles de entradas y la salidas de los componentes de la aplicación.

Etapas 7 Repita las etapas 5 y 6 para cada componente de la aplicación

Etapas 8 Pruebe la aplicación completamente con cada combinación posible antes de colocar en funcionamiento el programa.

Verifique que la documentación esté correcta, para que todas las personas que vayan a usarlo entiendan bien cada uno de los procedimientos.

Etapas 9 Colocación en funcionamiento:

Esta es la etapa donde se descubrirá si se ha hecho un buen programa o todavía es necesario cambiar algunas partes del programa. Típicamente esta parte es muy esforzada porque el dueño, (llamemos dueño al que tiene más interés que el proceso esté funcionando), quiere tener la máquina o el conjunto de máquinas trabajando en el más corto tiempo posible. En este período es donde el ingeniero electricista sufre todas las culpas del atraso del proyecto, aunque no sea verdad. Es muy conveniente evitar esos problemas al máximo.

Ejemplos de programas ladder con PLC DL06

Ejemplo 1.

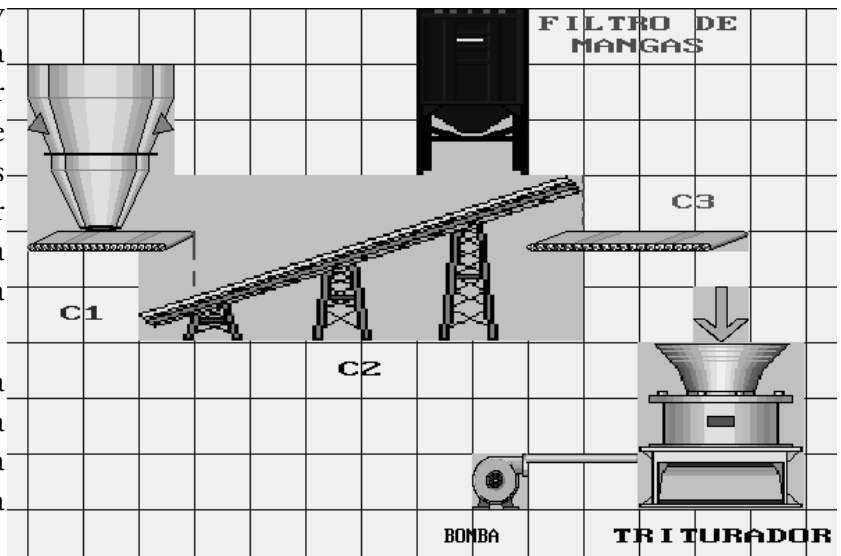
A continuación presentamos como hacer un proyecto que debe ser definido por un técnico que entienda el proceso a ser controlado. Hay muchas etapas iniciales antes de hacer el sistema de control y de ejecutar la programación del PLC:

Como ejemplo de la metodología para poder hacer un proyecto industrial o comercial con un PLC mostraremos un ejemplo simple de un sistema de molienda de mineral con 3 correas transportadoras. En los próximos ejemplos trabajaremos sobre el mismo concepto.

Etapas 1. Este control debe tener un enclavamiento de partida de tres correas transportadoras de mineral que deben partir una luego de otra con un retraso entre cada una para alimentar un triturador de mineral y

además, al ser activada una parada de emergencia por cuerda, todas las correas que están en el lado “aguas arriba” van a parar inmediatamente. Vea la figura adyacente con la idea descrita.

Cada correa transportadora C1, C2 o C3 tiene asociada una parada de emergencia por cuerda a lo largo de la correa.



En la partida, el operador apretará un botón que hará sonar una sirena por 45 segundos, para alertar a las personas que el sistema va a partir. Al final de los 45 segundos parte la bomba de lubricación del triturador y al mismo tiempo el ventilador de un filtro de mangas para sacar el polvo que será creado en el transporte y en la molienda, si el aire para limpiar las mangas tiene la presión suficiente.

Si la presión de lubricación llega al valor mínimo aceptable y el ventilador está corriendo, parte el triturador de mineral y después de un cierto tiempo partirán las otras correas para eliminar cualquier material que pueda haber quedado en la parada anterior en la secuencia C3, C2 y C1 con un intervalo de tiempo de cada una.

La parada normal hará que la correa C1, pare primeramente y permitirá que las otras queden vacías antes de parar. Cada correa tiene un motor de inducción de una velocidad solamente, en este ejemplo. Por definiciones del proceso, el tiempo de parada está definido con un cierto tiempo entre correas (puede ser otro valor cualquiera diferente que el que se ha programado). Luego se para el triturador y después de un cierto tiempo, la bomba de lubricación del triturador.

Etapa 2 Revisemos la descripción:

- Lea la descripción. *Falta definir el lugar del pupitre de control; Eso debe ser definido por la conveniencia de la operación y las distancias envueltas.*
- ¿Le hace sentido a usted? *Parece ser correcto pero falta saber qué viene después del triturador. Tendría sentido colocar un enclavamiento para el sistema que va después del triturador. Sería interesante mostrar al operador qué equipo está funcionando.*
- ¿Tiene definidos los componentes correctos del PLC? *NO, los definiremos mas adelante:*
- ¿Se han tomando en cuenta consideraciones para hacer el sistema seguro para el personal y el equipo? *Se han considerado paradas de emergencia en la correas; no se han considerado protecciones en el triturador ni en el filtro de mangas. Posiblemente existan protecciones mecánicas para impedir el acceso. Debe haber alarmas cuando falla la presión de aire o de lubricación. Si para el triturador, deben parar inmediatamente las correas, Si se para el ventilador, debe pararse la correa C1 y luego las otras hasta parar el triturador vacío.*
- ¿Es la velocidad de procesamiento un factor a considerar? *No en este momento.*
- ¿Dónde están los cuellos de botella en la aplicación? *Posiblemente en la posibilidad de atorar el triturador con material. El triturador tiene una capacidad de 300 Toneladas por hora y si se sobrepasa este límite el motor puede parar y lo peor es que el triturador quede lleno y al quedar lleno no puede partir tan fácilmente.*
- ¿Hay una manera más sencilla de alcanzar los mismos resultados? *Bueno, el sistema está ya definido; no hay por ahora opciones en este caso. Sin embargo, la solución de control puede tener más de una alternativa. De hecho existen otras alternativas, pero no las discutiremos aquí.*
- ¿Hay una manera más rápida de alcanzar los mismos resultados? *Este no es un sistema en que la velocidad de procesamiento del control juegue un papel importante, por lo menos en esta etapa.*

Etapa 3 Descripción de nivel eléctrico de la aplicación

Este es una definición de las necesidades para la parte eléctrica de potencia:

Alimentación trifásica 380 Volt, 50 Hz, 20 MVA de potencia de cortocircuito

Motor de la correa alimentadora C1	20 HP, 1500 rpm
Motor de la correa transportadora C2	50 HP, 1500 rpm
Motor de la correa transportadora C3	10 HP, 1500 rpm
Motor de la bomba	3 HP, 1000 rpm
Motor del triturador	500 HP, 1500 rpm con 6,6 kV; control separado
Motor del ventilador del filtro de mangas	30 HP, 1500 rpm

Necesitamos de un centro de control de motores con 5 partidores en 380 Volt; el control del triturador en media tensión, los fusibles o interruptores necesarios, todo instalado en uno o más gabinetes metálicos, de acuerdo a normas preestablecidas de seguridad, una sirena de alarma, un pupitre de control con botones y lámparas, el PLC para hacer el control, incluyendo el programa de control, un diagrama de alambrado de cada componente, instrucciones de instalación y de

operación, después de tener claro el concepto de control. El control del triturador solamente necesita un contacto que se cierra para partir, y se para al abrir este contacto.



Nota: Este es solamente un ejemplo. AUTOMATIONDIRECT no toma ninguna responsabilidad en la implementación de un sistema similar. Este ejemplo solamente sirve para mostrar la metodología.

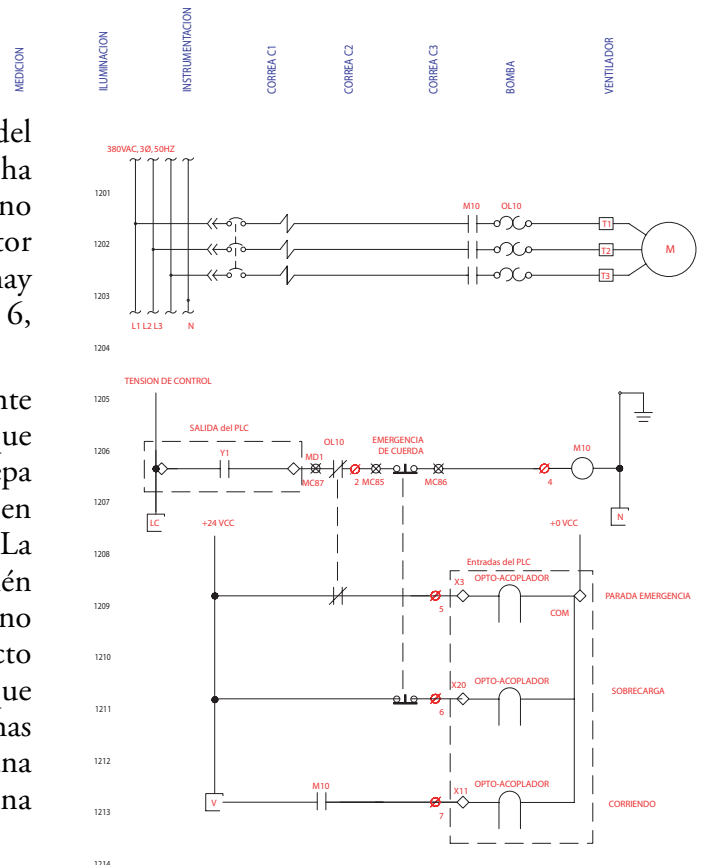
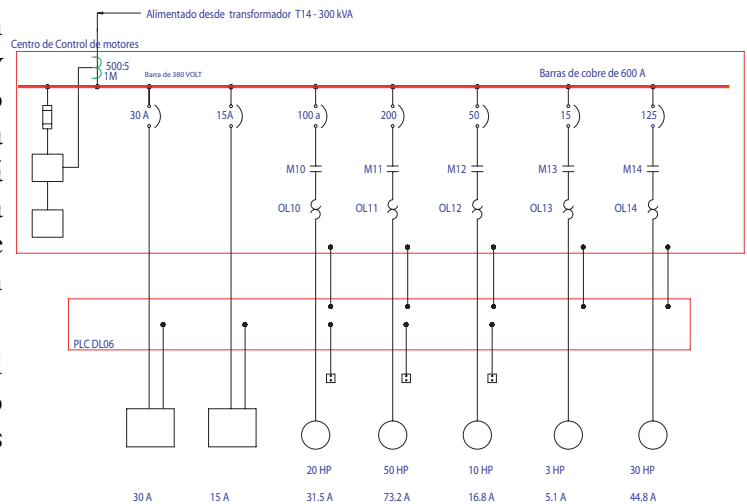
En la figura adyacente se muestra un diagrama unifilar de potencia y como se ha desarrollado el cableado de control de cada motor; esto da una idea de como será interconectado el PLC a cada entrada y cada salida. Naturalmente es necesario más documentación que ésta.

El botón de emergencia principal debe entrar en el PLC y al mismo tiempo debe desenergizar todas las salidas del PLC.

En la segunda figura de esta página se muestra una conexión típica de cada partidor de motor de 380 Volt.

Note que el relevador de sobrecarga OL10 abre el circuito de la bobina del partidor físicamente, es decir, se ha optado en este caso que el PLC no intervenga en la apertura del contactor del partidor del motor; sin embargo, hay aún una señal de sobrecarga en el borne 6, que va al PLC.

Esta señal se usa en este caso solamente como indicación para el operador, lo que puede ser conveniente para que éste sepa que el motor ha sido sobrecargado y en este caso no es un problema eléctrico. La parada de emergencia por cuerda también se ha implementado en el cableado y no en el PLC. Por último, el contacto auxiliar del contactor M10 confirma que el contactor se ha cerrado. Hay muchas otras soluciones de control. Ésta es una muy simple y no constituye una recomendación.



Capítulo 11: Recopilación y ejemplos

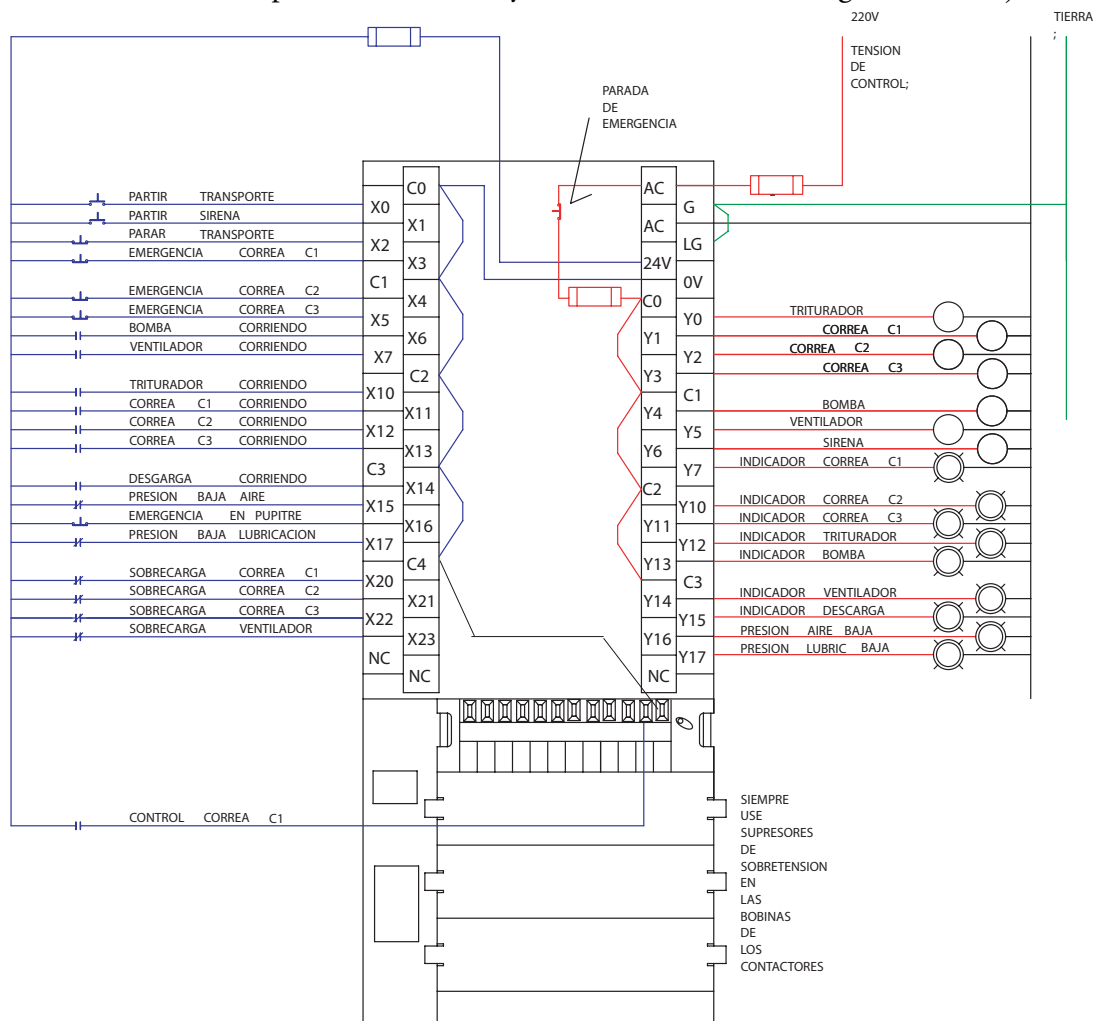
Los diagramas anteriores se deben ejecutar realmente lo más detallados posibles, para evitar que en el futuro los electricistas que deban mantener el sistema se confundan con el concepto. En los treinta años de experiencia del autor se ha visto mucho estas situaciones y vale la pena recalcar mucho este asunto. Sin embargo, no es el propósito de este manual entrar a discutir estos detalles.

En primera instancia escogeremos el PLC D0-06AR para este ejemplo. Este PLC tiene una alimentación de 100 a 240 VAC, 43 a 63 Hz y entradas de 90 a 120 Volt así como salidas entre 17 a 240 VCA.

La primera consideración a ser hecha es escoger 110 o 220 Volt para el voltaje de control. Ya que un voltaje de 380 Volt nos entrega 220 Volt entre una fase y tierra, parece ser más lógico escoger 220 Volt, pero tenemos la limitación que las entradas sólo pueden llegar hasta 120 Volt.

En ese caso, preferiremos hacer un sistema con entradas de 24 Volt CC y salida de 220 Volt. El PLC tiene una fuente de poder de 24 Volt, 300 mA. Cada entrada llega a 8 mA cuando está activada y ya que tenemos 20 entradas usaremos solamente 160 mA en el peor de los casos. De modo que usaremos el PLC D0-06DR. Puede considerarse un módulo de entrada.

Haremos el cableado de potencia, entradas y salidas de acuerdo al diagrama de abajo.



Asignación de las estradas discretas

Las entradas serán asignadas por una convención arbitraria, tal como la tabla de abajo: En general se recomienda dejar algunas entradas disponibles para el futuro, en caso de modificación del proyecto, adición de una máquina en el futuro, etc. En este caso, si fueran necesarias otras entradas o salidas, se puede adicionar otro módulo de acuerdo a la necesidad, ya que el PLC DL-06 tiene 4 ranuras para módulos opcionales.

Se ve la necesidad de instalar los botones en un lugar conveniente. Decidimos en este ejemplo que un operador tendrá un pupitre en una localización central, con los botones X0, X1, X2 y X100, además de indicaciones de funcionamiento a ser definidas de acuerdo entre el técnico de proceso y el diseñador del sistema de control. En este caso, agregaremos el módulo D0-08TR.

Note que el proceso no siempre es bien definido por el ingeniero de proceso y por lo tanto es necesario tener siempre una realimentación. Hay varias preguntas que el programador debe hacer al “dueño”, para evitar responsabilidades después que el proceso está funcionando y no está adecuadamente diseñado. No se puede culpar al ejecutor del sistema de control de las fallas de diseño para el operador, que no permitan que el operador esté en control del sistema.

Terminal	Apodo	Descripción
X0	Partir	Botón de partida del operador en el pupitre de control N.A
X1	Sirena	Botón para activar la sirena para comunicación N.A.
X2	Parar normal	Botón de parada en el pupitre de control N.C.
X3	Emergencia de C1	Conmutador de emergencia en la correa C1, N.C
X4	Emergencia de C2	Conmutador de emergencia en la correa C2 N.C
X5	Emergencia de C3	Conmutador de emergencia en la correa C3 N.C
X6	Bomba corriendo	Contacto auxiliar del contacto del motor de la bomba N.A
X7	Ventilador corriendo	Contacto auxiliar del contacto del motor de la bomba N.A.
X10	Triturador corriendo	Contacto auxiliar del contacto del motor de la bomba N.A
X11	C1corriendo	Contacto auxiliar del contactot del motor de la correa C1 N.A.
X12	C2 corriendo	Contacto auxiliar del contacto del motor de la correa C2 N.A
X13	C3 corriendo	Contacto auxiliar del contacto del motor de la correa C3 N.A
X14	Enclavamiento después del triturador	Contacto auxiliar N.C.de un contacto del sistema de descarga
X15	Falta de presión de aire del filtro	Presostato de baja presión de aire.(N.C está normal)
X16	Parada de emergencia principal	Botón de emergencia en el pupitre
X17	Presión de lubricación normal	Presión del aceite de lubricación del triturador (N.C. está normal)
X20	Sobrecarga de la correa C1	Contacto abierto del disparador térmico de sobrecorriente
X21	Sobrecarga de la correa C2	Contacto abierto del disparador térmico de sobrecorriente
X22	Sobrecarga de la correa C3	Contacto abierto del disparador térmico de sobrecorriente
X23	Sobrecarga del ventilador	Contacto abierto del disparador térmico de sobrecorriente
X100	Control de la correa C1	Botón en el pupitre para parar temporalmente las correas
X101	Sobrecarga de la bomba	Contacto abierto del disparador térmico de sobrecorriente
X102	Sobrecarga del triturador	Contacto abierto del disparador térmico de sobrecorriente

Asignación de salidas discretas

Las salidas serán asignadas por convención como la tabla de abajo:

Terminal	Apodo	Descripción
Y0	Triturador parte	Esta salida parte y hace funcionar el triturador
Y1	C1 activada	Esta salida parte y hace funcionar la correa C1
Y2	C2 activada	Esta salida parte y hace funcionar la correa C2
Y3	C3 activada	Esta salida parte y hace funcionar la correa C3
Y4	Bomba activada	Esta salida parte y hace funcionar la bomba
Y5	Ventilador parte	Esta salida parte y hace funcionar el ventilador
Y6	Sirena activada	Esta salida parte y hace funcionar la sirena
Y7	Indicador de C1	Prende la lámpara C1
Y10	Indicador de C2	Prende la lámpara C2
Y11	Indicador de C3	Prende la lámpara C3
Y12	Indicador de triturador	Prende la lámpara del triturador
Y13	Indicador de bomba	Prende la lámpara de la bomba
Y14	Indicador de ventilador	Prende la lámpara del ventilador
Y15	Indicador descarga OK	Lámpara indicando que la descarga del triturador está corriendo
Y16	Presión baja de aire	Prende la lámpara para indicar presión baja
Y17	Lubricación sin presión	Prende la lámpara para indicar falla en la lubricación
Y100	Indicador sobrecarga C1	Lámpara indicando que la correa C1 paró por sobrecarga
Y101	Indicador sobrecarga C2	Lámpara indicando que la correa C2 paró por sobrecarga
Y102	Indicador sobrecarga C3	Lámpara indicando que la correa C3 paró por sobrecarga
Y103	Indicador sobrecarga bomba	Lámpara indicando que la bomba paró por sobrecarga
Y104	Indicador sobrecarga ventilador	Lámpara indicando que el ventilador paró por sobrecarga
Y105	Indicador sobrecarga Triturador	Lámpara indicando que el triturador paró por sobrecarga

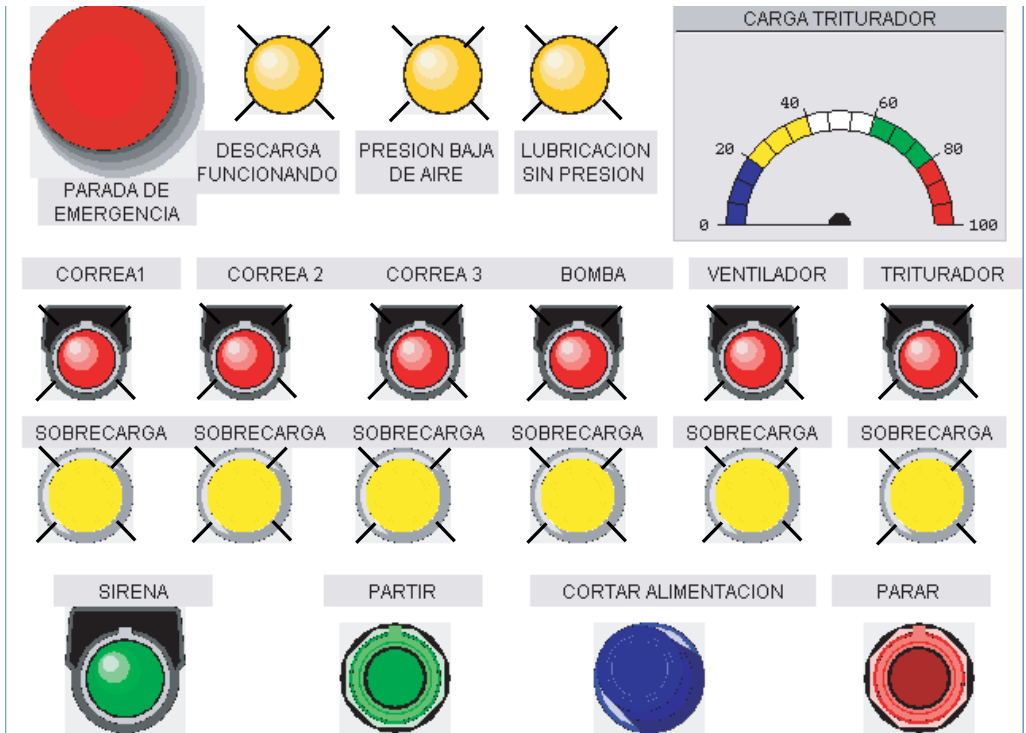
En general se recomienda dejar algunas salidas disponibles para el futuro, en caso de modificación del proyecto, adición de una máquina en el futuro, etc. En este caso, si fueran necesarias otras entradas o salidas, se puede adicionar uno o más módulos de acuerdo a la necesidad, ya que el PLC DL-06 tiene 4 ranuras para módulos opcionales. De hecho, en los siguientes ejemplos, agregaremos más módulos.

Como definimos en la página anterior, un operador tendrá en el pupitre las indicaciones de funcionamiento (definidas de acuerdo entre el técnico de proceso y el diseñador del sistema de control). Digamos que las asignadas arriba fueron ya definidas.

Es muy importante definir estas indicaciones basadas en experiencias anteriores. No hay nada mejor que en un proyecto se use la experiencia de personal que haya trabajado en sistemas similares. Evidentemente es necesario ser inventivo en el caso de un sistema completamente nuevo.

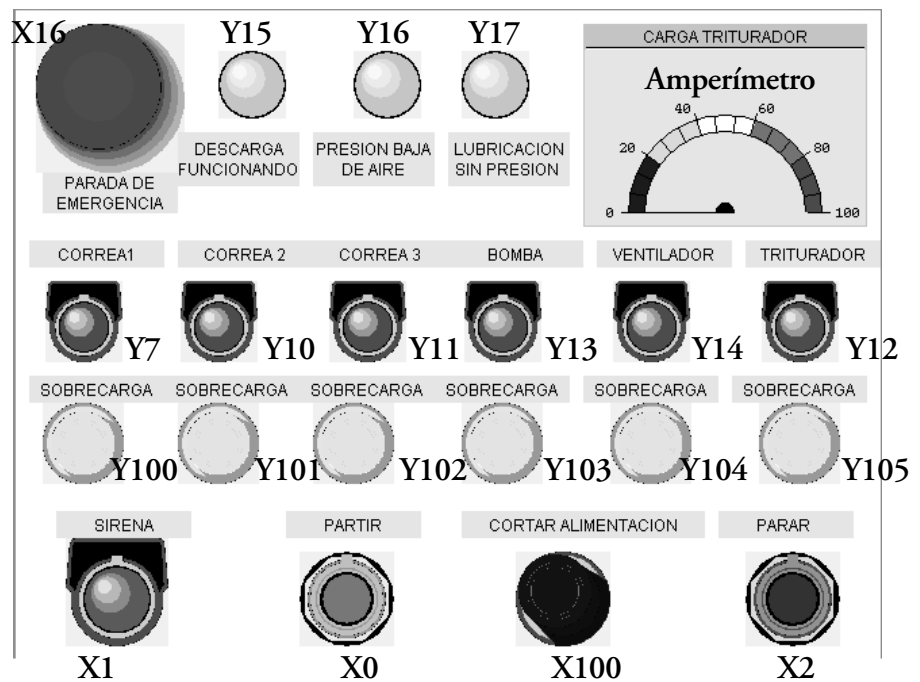
El pupitre de control estará en una parte central de la planta, donde el operador pueda tener acceso al personal de apoyo, si fuera necesario. Vea en la próxima página más informaciones sobre el pupitre.

Todos los indicadores iluminados tienen el símbolo mostrado arriba. Lo que no está indicado significa un botón o selector.



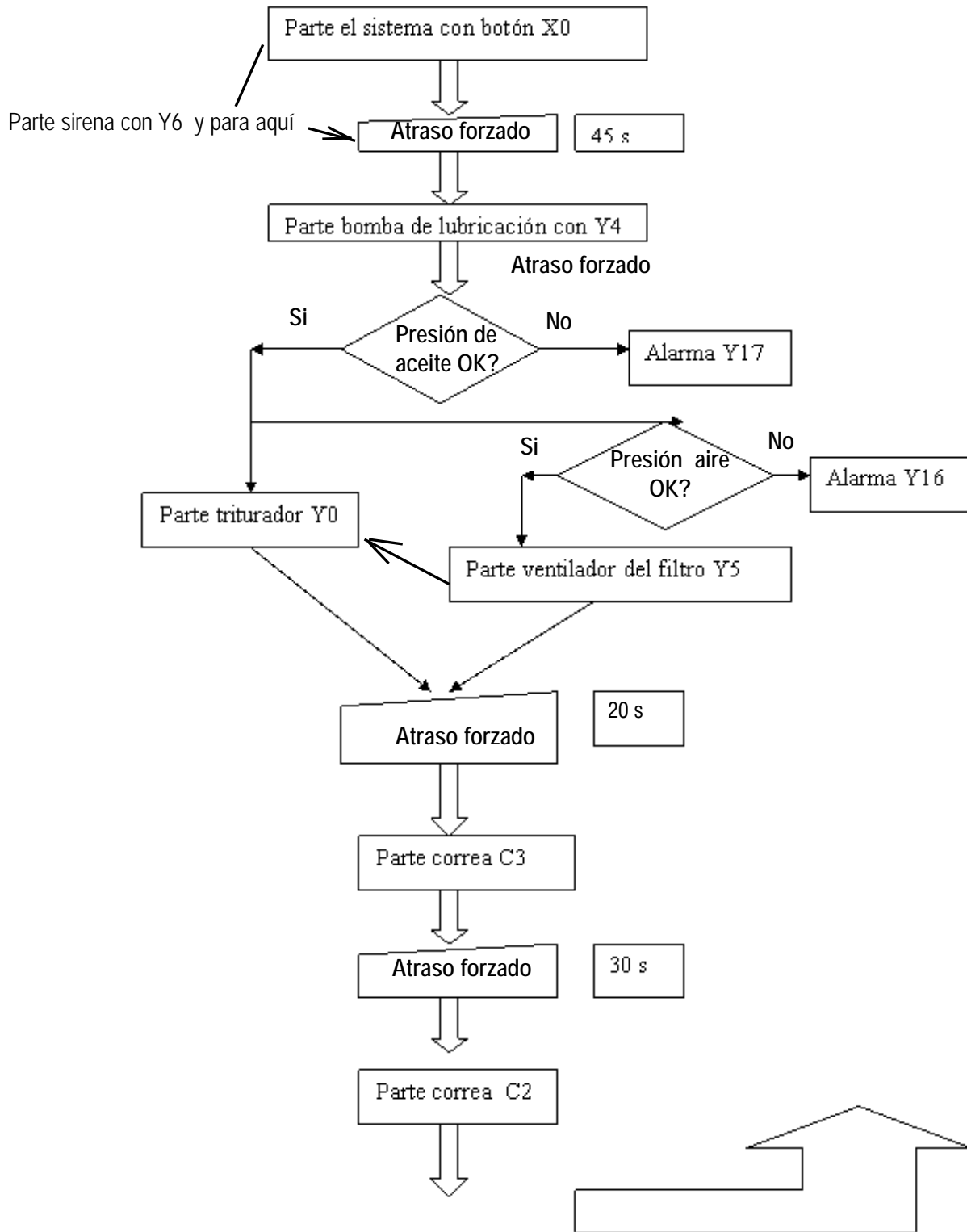
Vista frontal del pupitre

El pupitre tiene asociadas las entradas y salidas mostrada en la figura de abajo. Note que necesitaremos más salidas de relevadores que las previstas, para activar la indicaciones Y100 hasta Y105. Agregaremos el módulo D0-08TR en la ranura 2.

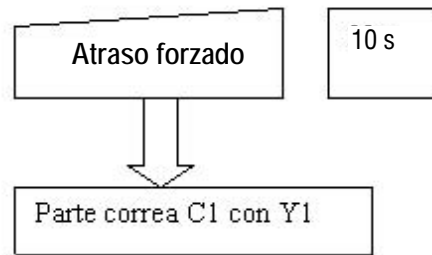


Definición de entradas y salidas del PLC

Diagrama de bloque de la operación



11



El diagrama de bloques muestra que el sistema está funcionando en este momento.

Observaciones para el desarrollo del programa

Note los siguientes puntos cuando se observa el diagrama:

- La bomba necesita de unos segundos para crear presión de aceite de lubricación.
- La presión de aire es generada por un compresor fuera del sistema, pero lo interesante es que haya presión de aire necesario para la limpieza de las mangas del filtro. No es necesario saber si el compresor está funcionando. Esto enclava la partida del ventilador del filtro de mangas.
- Aquí no aparece el diagrama de operación de parada normal, de paradas por cuerda de las correas transportadoras en emergencia o en caso de falla de cualquier motor por sobrecarga y será por lo tanto necesario hacer otro diagrama; en ese momento, haremos lo definido en el diagrama, que es típico en el desarrollo del programa ladder. Nada errado con tratar varias veces. La experiencia naturalmente ayuda a hacer el desarrollo lo más simple posible, pero en esta descripción de la ejecución del programa mostraremos cada etapa y como se desarrolla el proyecto.

Observe en las próximas páginas la primera tentativa de ejecutar el programa.

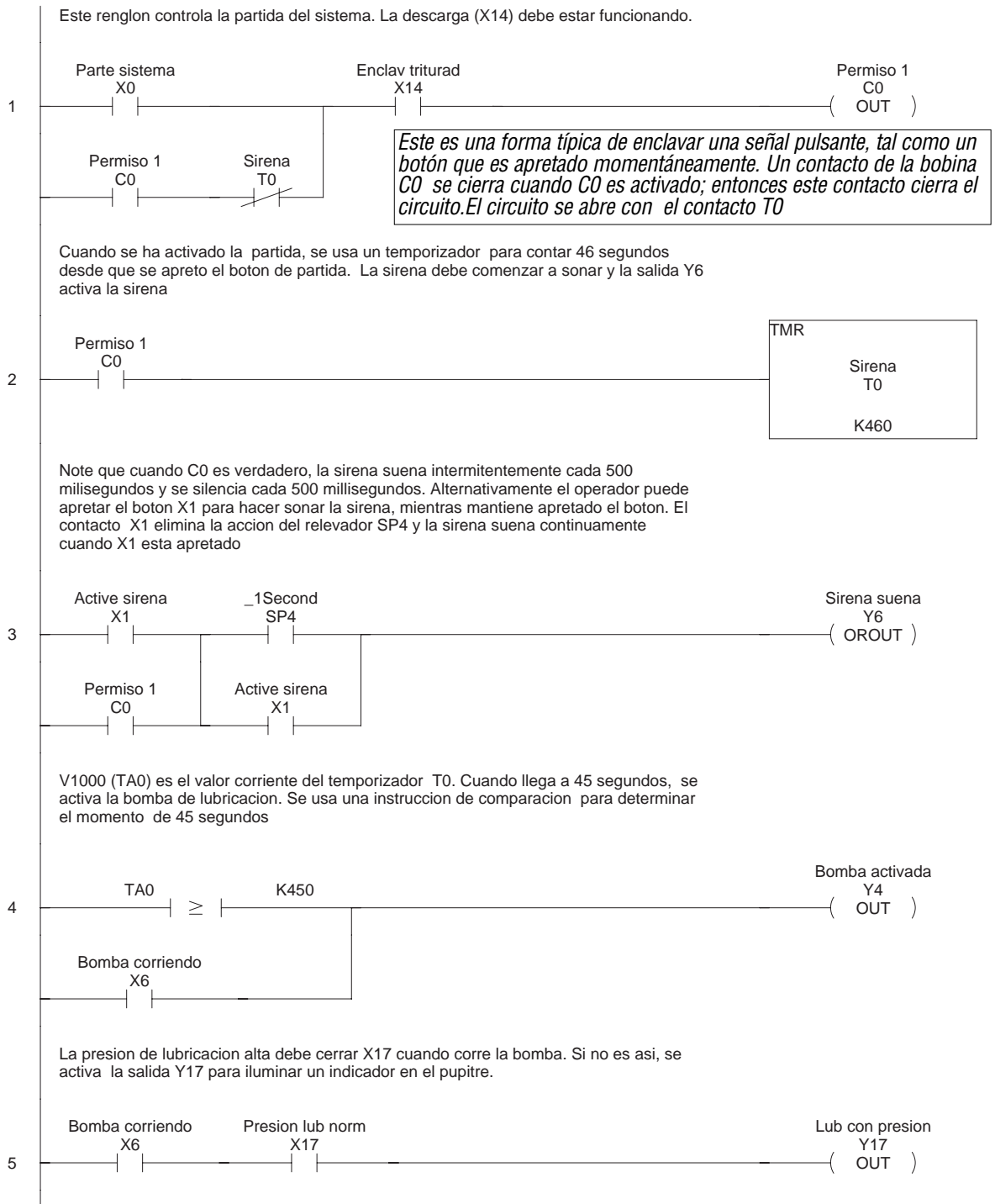
Note que cada partidador de motor tiene un contacto auxiliar que informa al PLC que realmente el contactor se ha cerrado. Esto es necesario para estar seguro que el motor está corriendo. Hay también otras formas de tener la confirmación de que el motor corre normalmente. Esta operación es remota y por lo tanto el operador no vé si una de las correas transportadoras o la bomba realmente están funcionando.

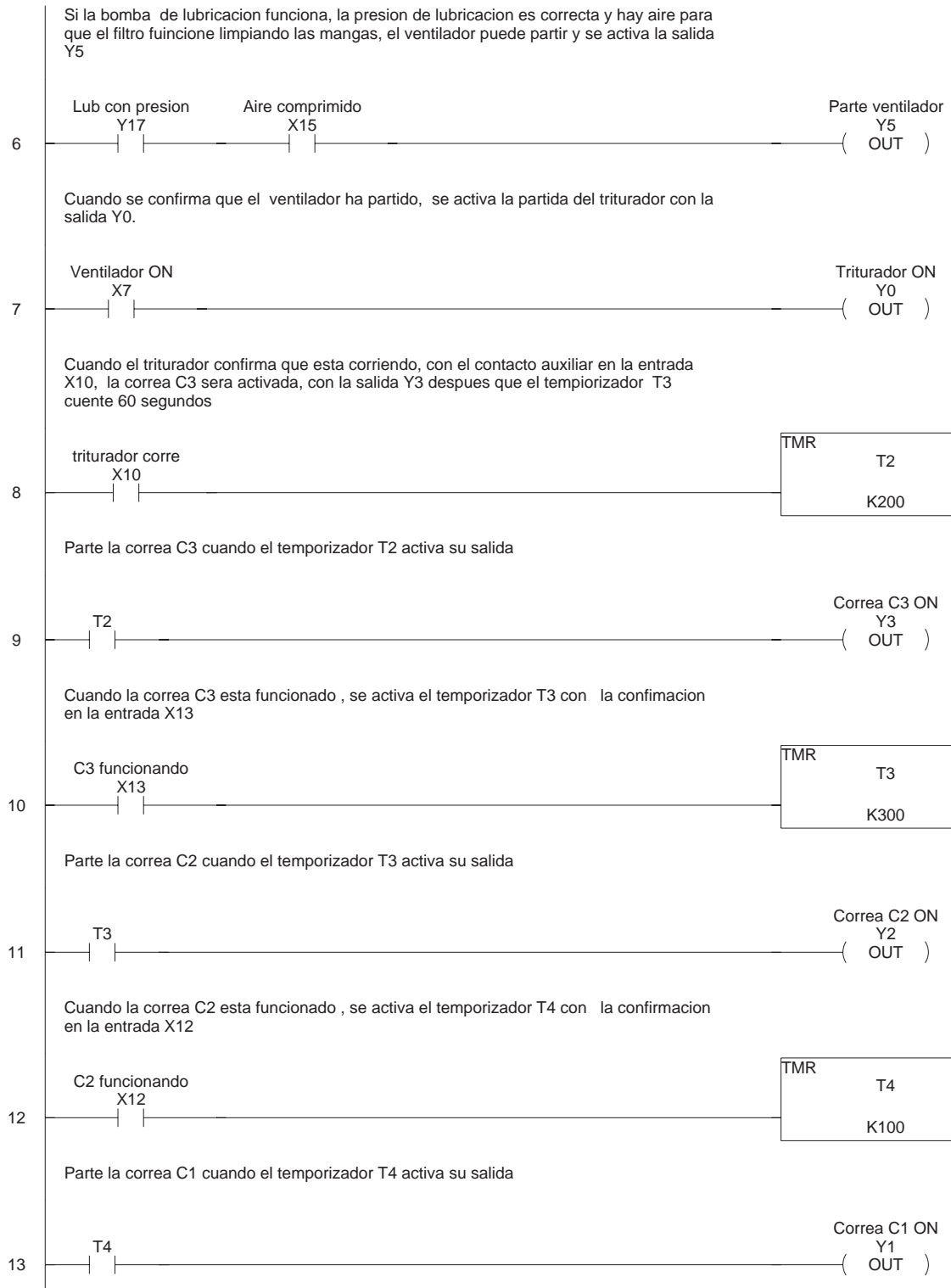
Note también que el pupitre tiene indicaciones sobre el funcionamiento de cada elemento de la operación y hay que asociarles una entrada o una salida.

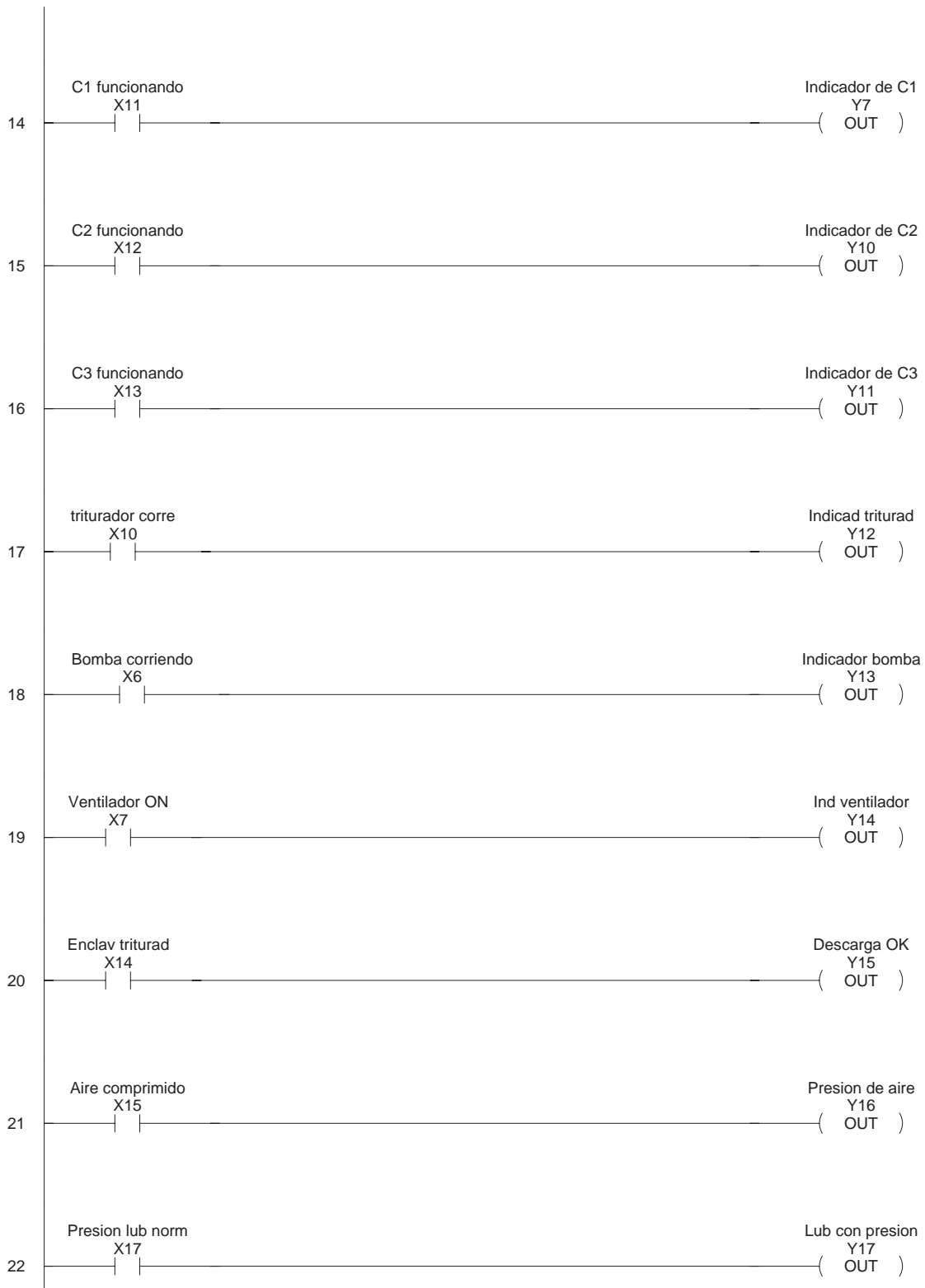
En general, no se tiene disponible durante la programación el centro de control de motores, el pupitre, los presostatos y ninguno de los elementos que completan el circuito de control.

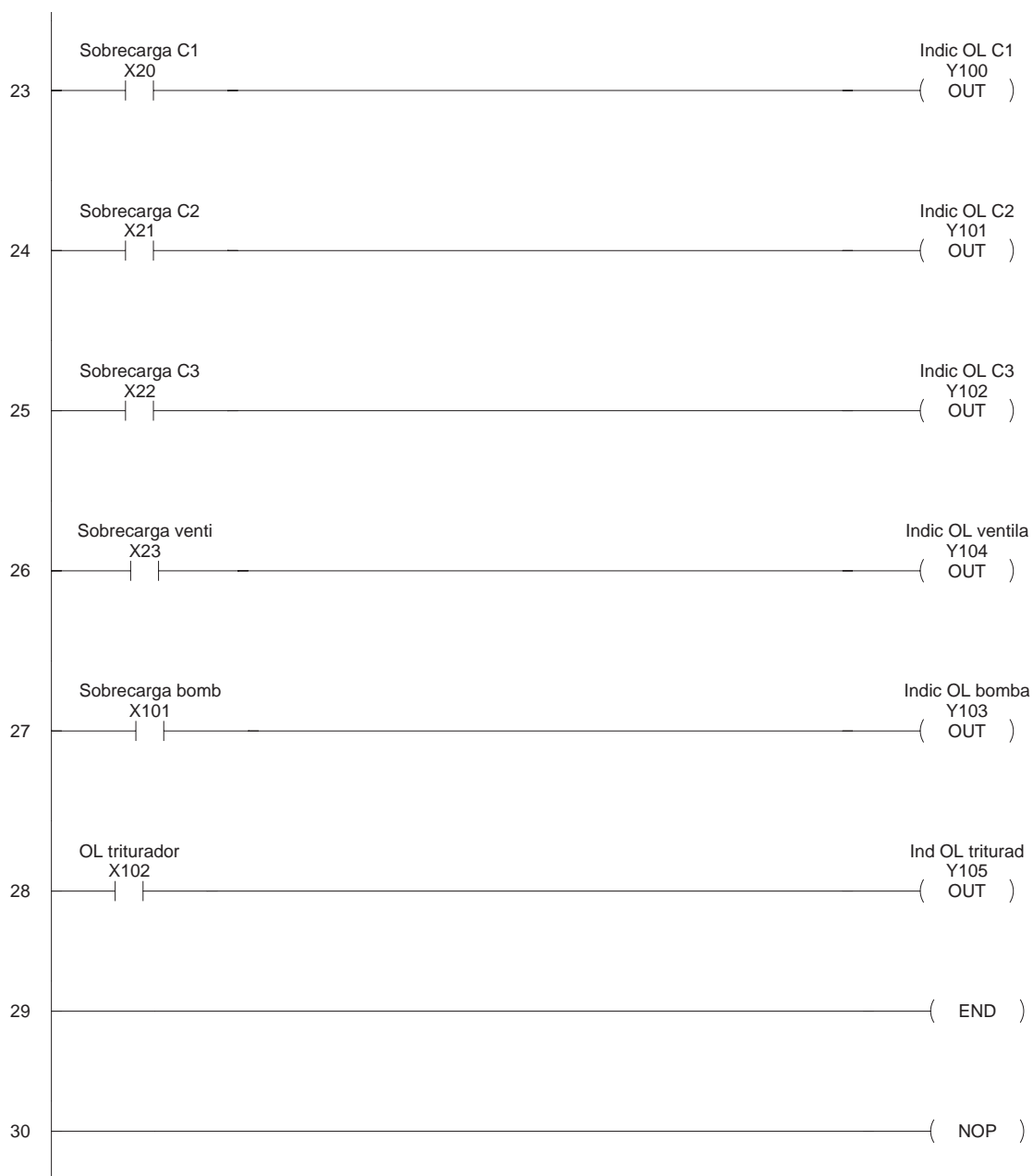
Por eso es muy deseable tener herramientas que puedan hacer la simulación del sistema. En el caso de este ejemplo, se conectó un segundo PLC DL06 con un módulo simulador (D0-08SIM) que permitió simular el proceso. Si Ud tiene solamente un PLC, puede hacer la simulación directamente en el mismo PLC, o parcialmente con **Data View**, y luego, cuando todo funciona ya en la planta, se borra el programa de simulación.

A continuación presentamos el diagrama ladder para esta parte. Hay explicaciones en cada rung. Otras explicaciones pueden ser encontradas en cuadros específicos.









El concepto de control del sistema y el correspondiente programa ladder en el PLC debe entenderse bien, entenderse como se ejecuta éste en el PLC, debe simularse y posiblemente corregir errores que puedan existir y luego obtener que éste funcione, antes de ir a una segunda revisión.

Como habíamos dicho, no están programadas las diversas paradas ni las protecciones, etc. Eso es lo que veremos en las próximas páginas. Note que estamos haciendo el programa basado en el diagrama de bloques creado inicialmente.

El programa puede ser modificado para aceptar las nuevas condiciones de parada.

El sistema debe causar paradas de los equipos por parada normal, por sobrecarga del triturador, por falla del sistema de descarga, por parada de emergencia desde el pupitre o por las cuerdas de emergencia al lado de las correas transportadoras o por defecto de uno de los motores de los equipos involucrados.

Analicemos cada caso:

1. Parada normal (cuando se aprieta el botón asociado a la entrada X2 en el pupitre)

Al apretar el botón X2 debe suceder que:

- a) Se para la correa C1 inmediatamente, al momento de apretar el botón
- b) Se espera un tiempo para que se vacíe de material la correa C1 (digamos 12 segundos) y se para la correa C2.
- c) Se espera un tiempo para que se vacíe de material la correa C2 (digamos 25 segundos) y se para la correa C3.
- d) Se espera un tiempo para que se vacíe de material el triturador (digamos 15 segundos) y se para el triturador y luego de 30 segundos la bomba de lubricación.
- e) Se espera que la lubricación pierda la presión para parar el ventilador del filtro de mangas.

2. Parada por triturador cargado sobre el límite (cuando la carga pasa de 95%)

Se espera que el triturador trabaje entorno del 90% de la carga máxima. El operador de la planta debe observar esta condición con el amperímetro del pupitre para determinar como está la carga del triturador. El PLC no puede tomar ninguna acción aquí porque no se ha definido ningún criterio para hacer ésto. En un ejemplo posterior veremos como automatizar este evento, lo que libera tareas del operador. Cuando la carga llegue a 90%, el operador debe parar la alimentación de material y para eso debe apretar un botón de parada de alimentación que debe parar las correas C1, C2 y C3 por un tiempo adecuado, que estimamos ahora en 60 segundos. Después de ese período, la correa C3 parte, luego la C2 y por ultimo la C1.

Este botón no había sido previsto al comienzo del desarrollo del proyecto y como sabemos que podemos aumentar la cantidad de entradas y salidas, lo haremos agregando un módulo D0-10ND3 y la entrada a ser usada será X100.

3. Parada de emergencia desde el pupitre.

En ese caso, el operador puede hacer que todas las máquinas paren inmediatamente, por alguna causa de peligro a una persona o por posibilidad de destrucción de l equipo.

En ese caso, todos los motores deben parar. Recuerde que ésta es una operación simple para mostrar como se hace la programación de un PLC en un proyecto de control. Talvez, dependiendo de otras condiciones definidas por el personal de proceso, el ventilador y la bomba del triturador no sean necesarias que paren. Para este caso simple. también se pararán estos motores.

4. **Parada por falla de la descarga.** Esto es lo mismo que una parada de emergencia. En este caso se cubre también una falla de energía eléctrica.

4. **Parada por accionamiento de la emergencia de cuerda de las correas.**

En ese caso, el PLC recibe la información de que se ha parado una de las correas y las correas “aguas arriba” deben parar inmediatamente. Será necesario que se inspeccione la causa de la parada y se tomen las medidas necesarias para reiniciar la operación nuevamente.

5. **Parada por defecto de uno o mas equipos** (típicamente una desconexión por sobrecarga)

En este ejemplo, tenemos las siguientes condiciones:

- Cada uno de los relevadores de sobrecarga de los partidores de motores.
- Si falla el aire comprimido para funcionamiento del filtro de mangas.
- Si falla la presión de lubricación del triturador.

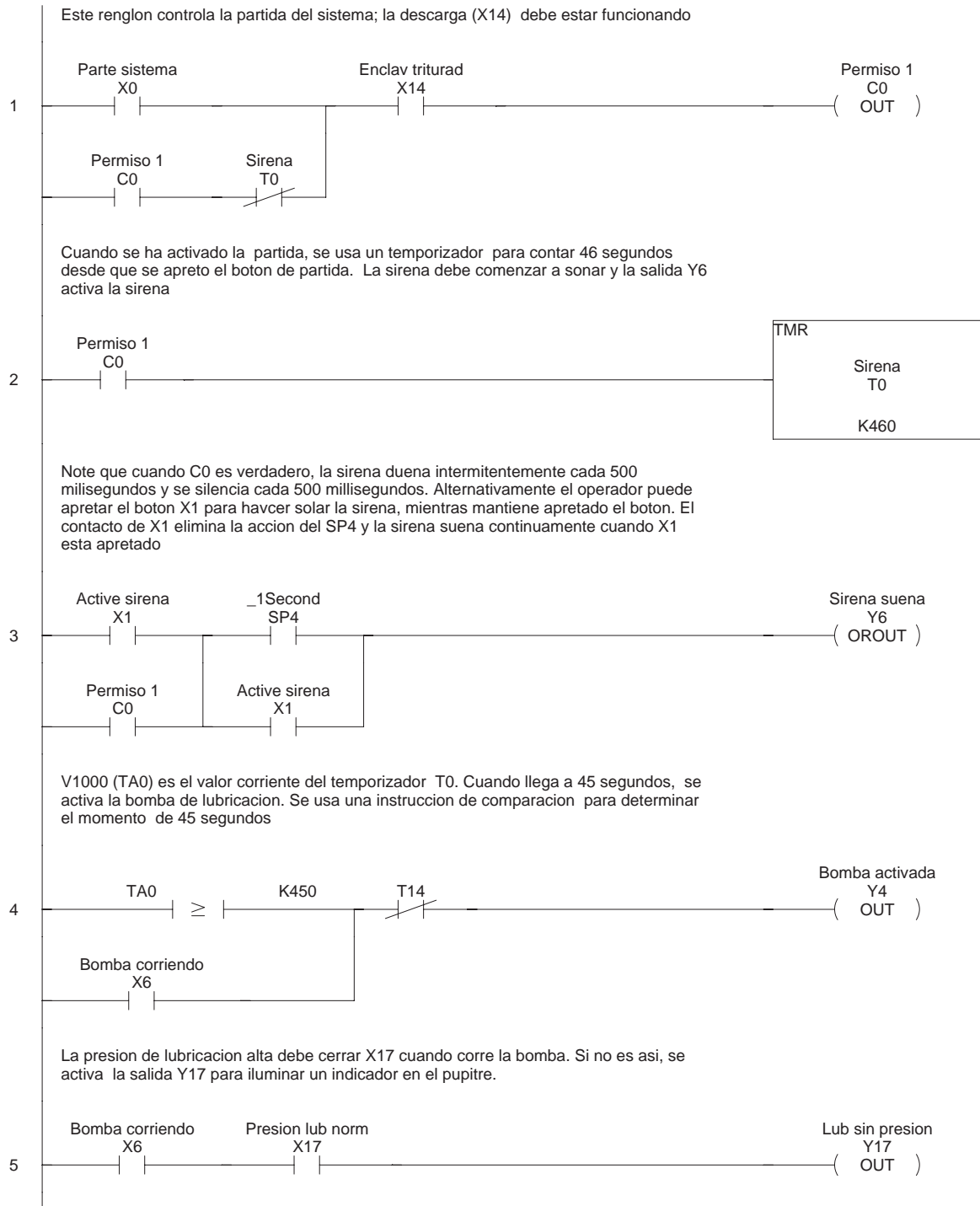
Veamos la condición en cada una de las posibilidades:

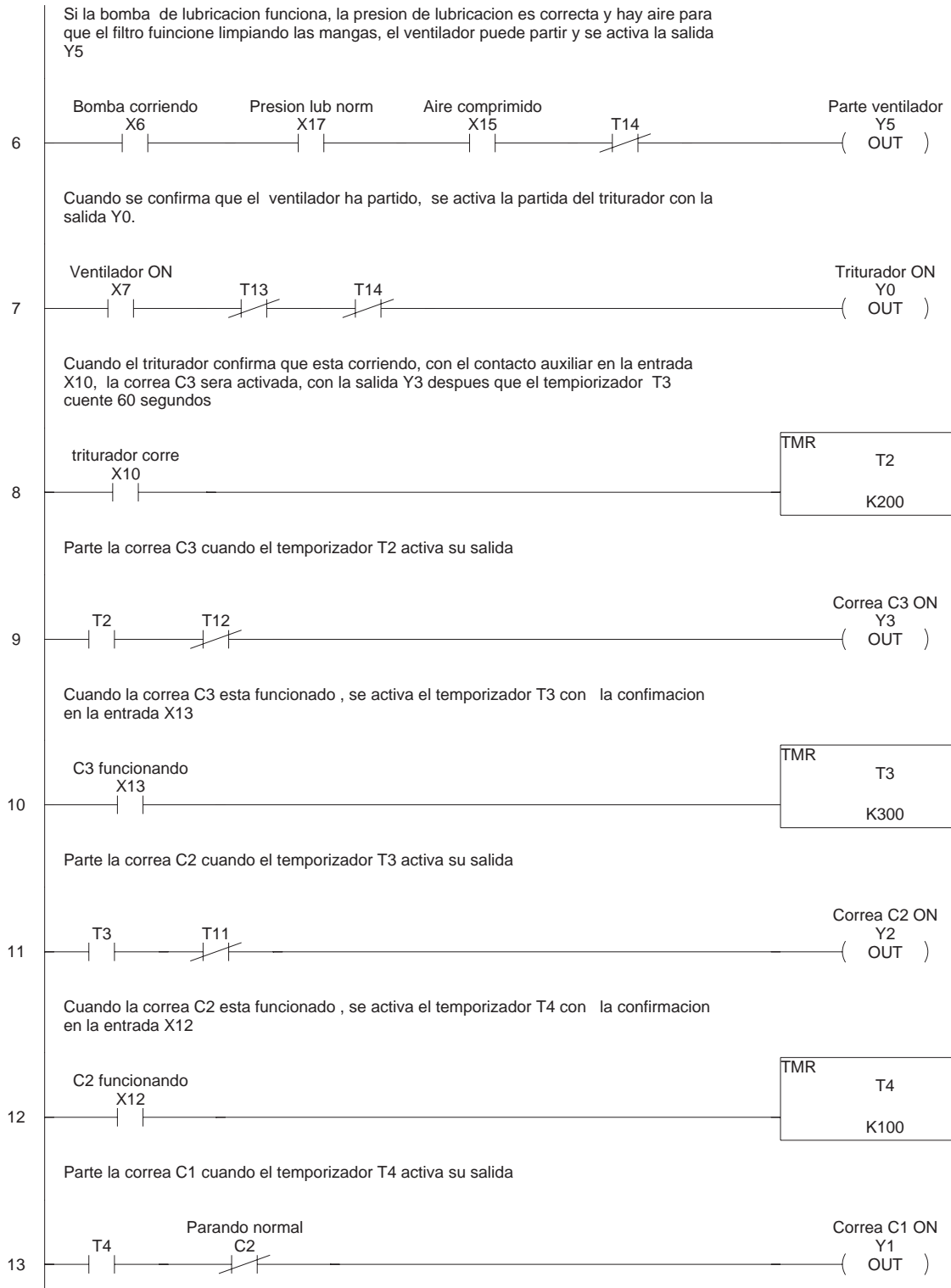
- **Desconexión por sobrecarga del triturador:** Al parar el triturador, deben parar todos los equipos “aguas arriba” inmediatamente, excepto el ventilador del filtro y la bomba de lubricación.
- **Desconexión por sobrecarga del ventilador:** Al parar el ventilador, se pierde la acción de limpieza del filtro y por lo tanto deben parar todos los equipos inmediatamente.
- **Desconexión por sobrecarga de la bomba de lubricación:** Al parar el triturador, deben parar todos los equipos “aguas arriba” inmediatamente.
- **Desconexión por sobrecarga una o más correas:** Al parar una de las correas, deben parar las correas “aguas arriba”.
- **Desconexión por falla de presión de aire:** Al no tener aire, se para la acción del filtro de mangas. Deben parar todos los equipos “aguas arriba”.

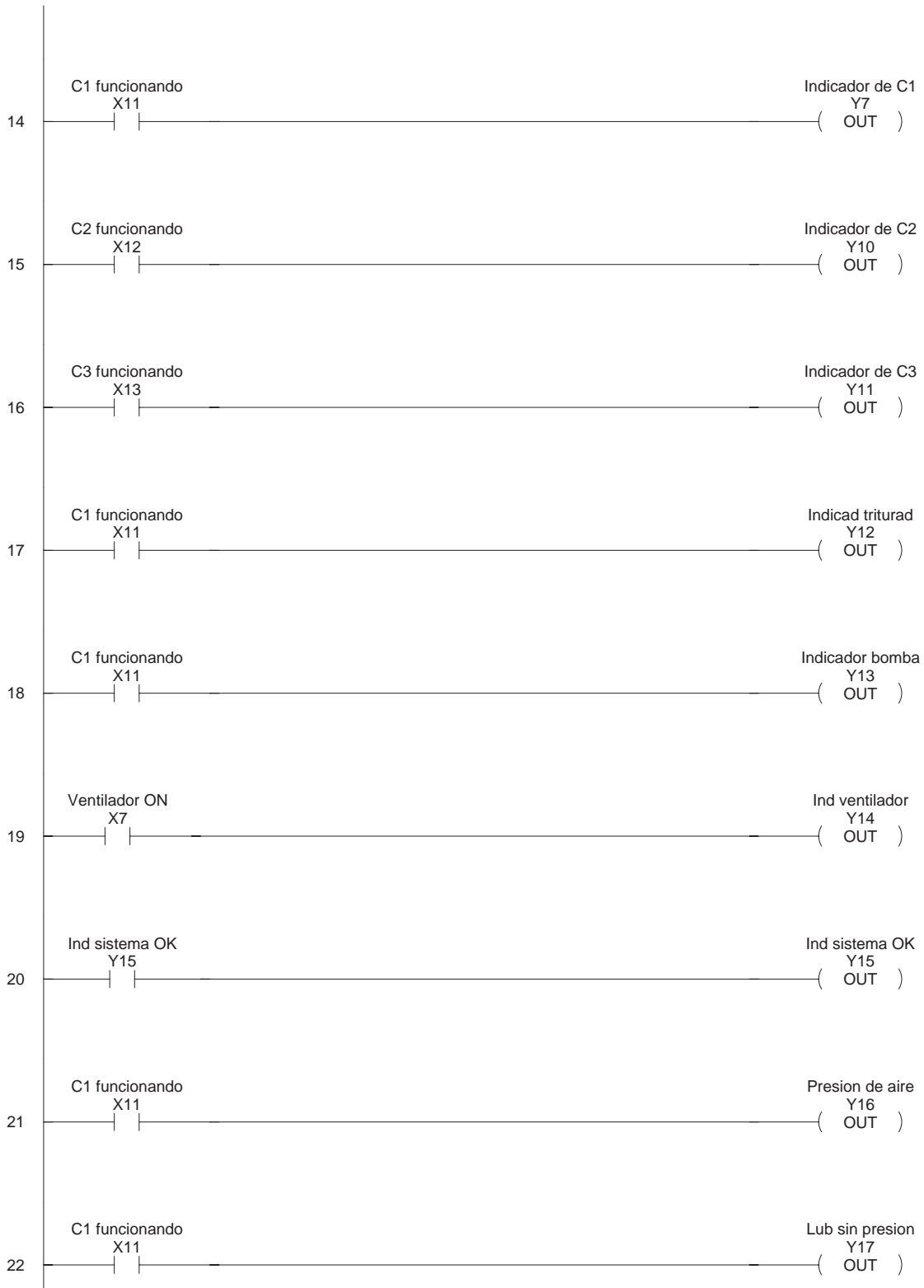
La primera corrección al programa hecho inicialmente tiene que ver con la parada normal. Crearemos un bit C16, que se hace ON al apretar el botón de parada del pupitre X2 y cuando éste esté activado, podrán ejecutarse las acciones de parada de los motores.

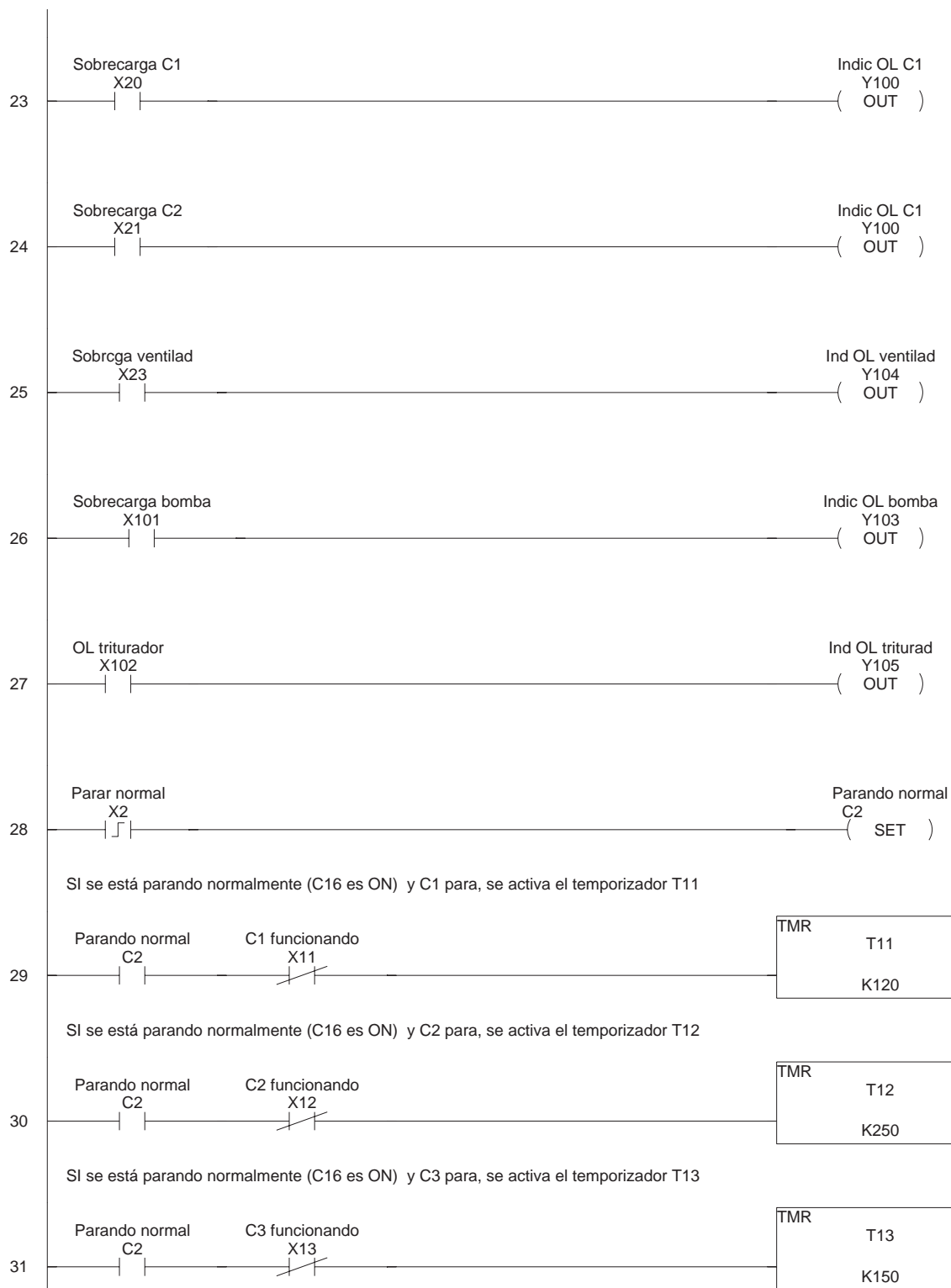
En las próximas páginas mostramos como ha sido modificado el programa para cubrir esta parte.

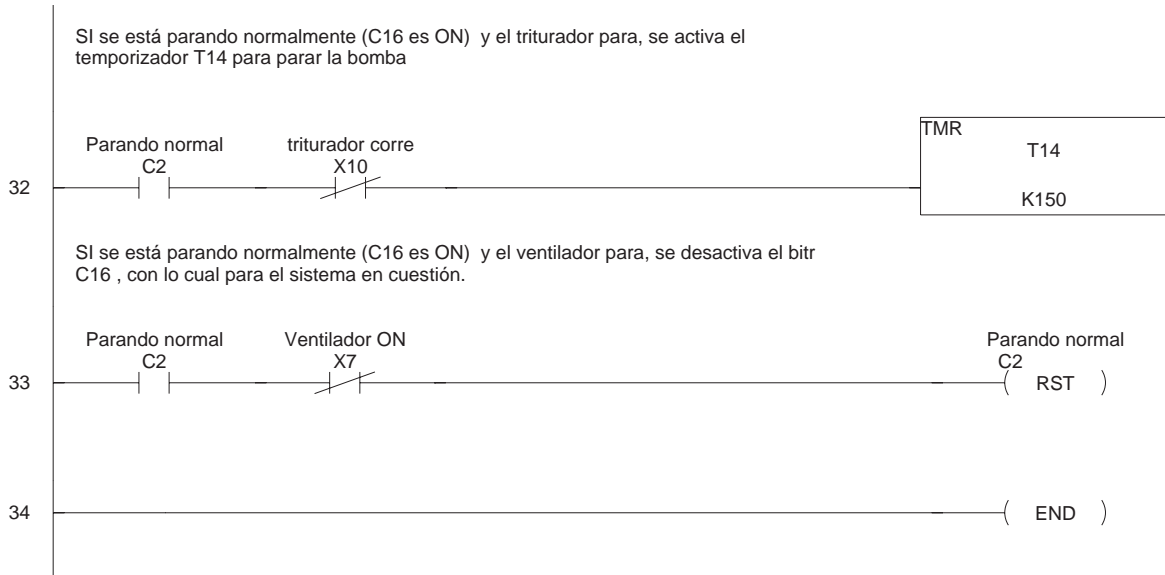
Esta parte define como fue modificado el programa ladder para incluir la parada normal.





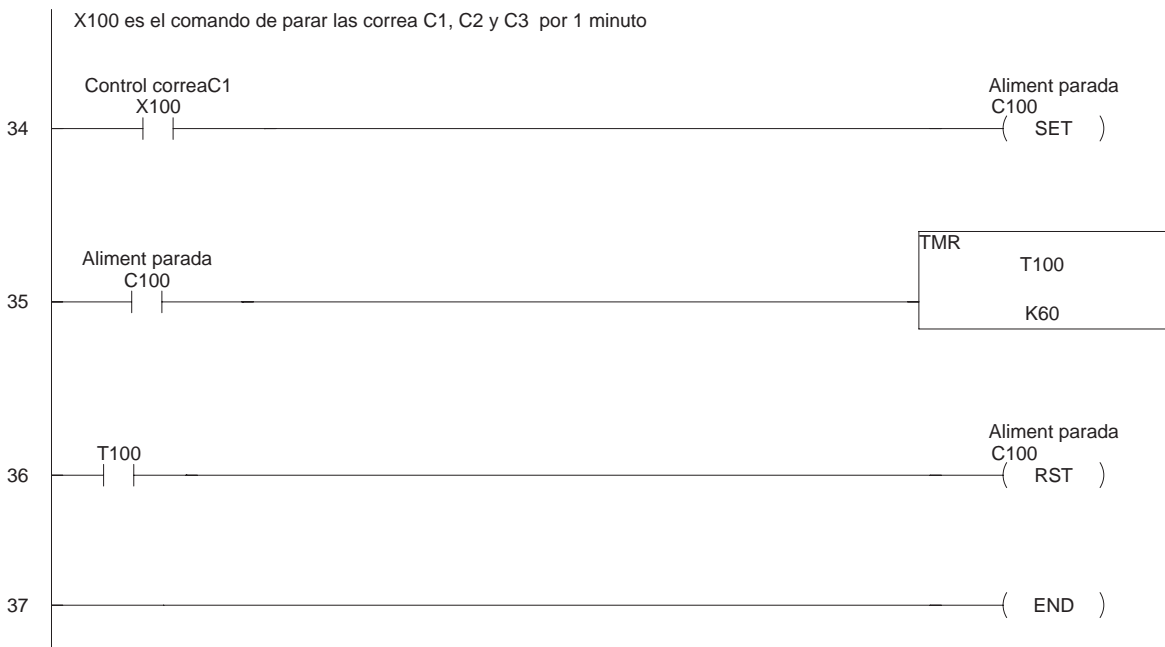






2. Parada por triturador sobrecargado.

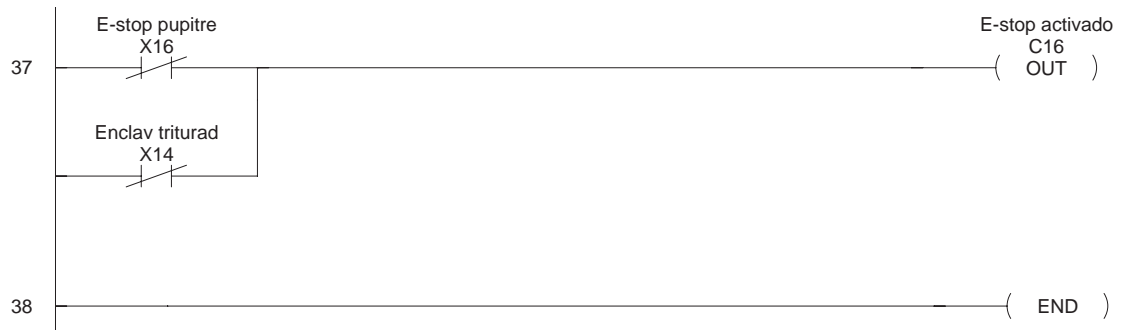
Esta condición debe bajar temporalmente el flujo de alimentación de material al triturador. Las correas C1, C2 y C3 deben parar por un cierto tiempo para que el triturador consuma el material que está procesando. Posiblemente 1 minuto puede obtener el resultado esperado, pero naturalmente este valor se podrá ajustar durante la operación, cambiando el valor en el programa. Vea qué renglones son necesarios en el próximo segmento de programa.



C100 enclava las salidas de las correas en los renglones 9, 11 y 13. No es necesario mostrar todo el programa nuevamente.

3. Parada de emergencia desde el pupitre o falla de la descarga

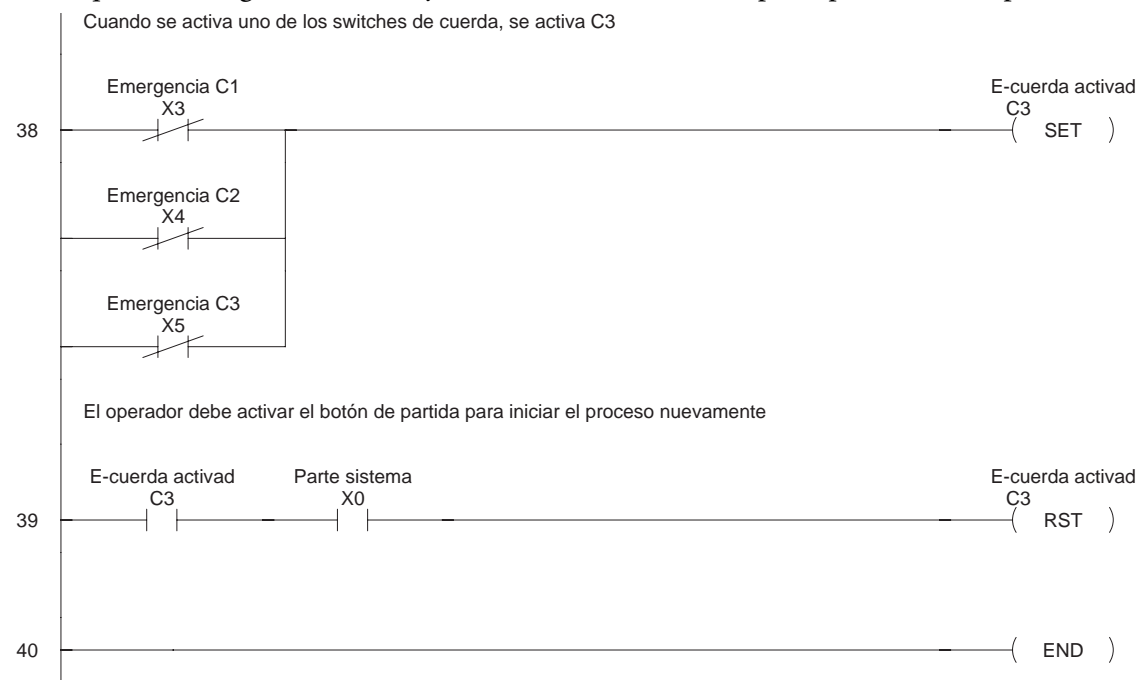
Vea que solamente es necesario colocar una condición de parada causada por el botón X16 o el botón X14. Uno de los botones activa la bobina de relevador interno C16, que abre el circuito en el renglón 4. No es necesario mostrar todo el programa nuevamente.



4. Parada de emergencia de cuerda de las correas transportadoras

En este caso, las entradas X3, X4 y X5 generan la condición de parada de cada una de las correas. La señal X3 para la correa C1. La señal X4 para la correa C2. La señal X5 para la correa C3.

Note que los renglones 8, 10 y 12 fueron cambiados para poder hacer parar la correa



correspondiente. No es necesario mostrar todo el programa nuevamente.

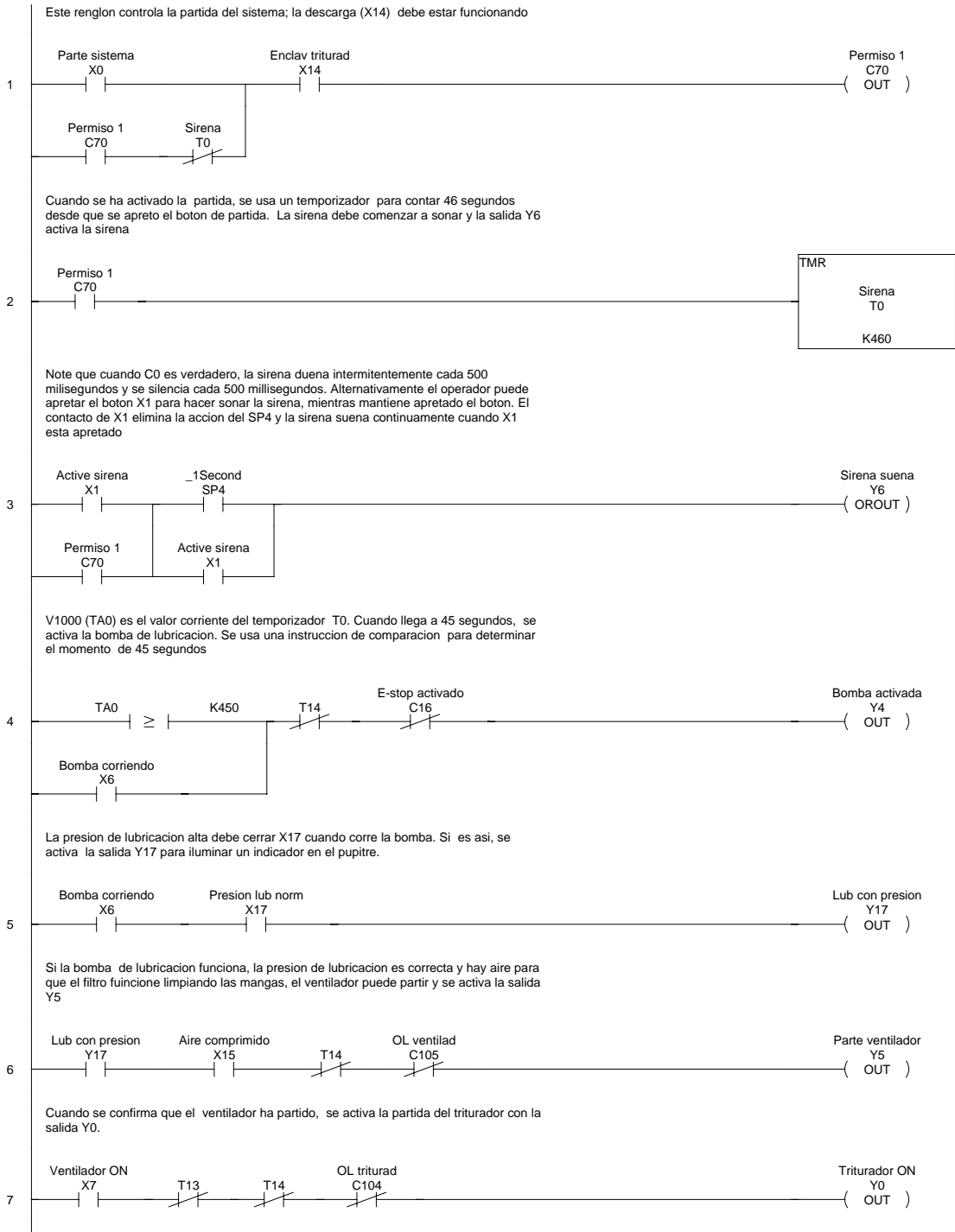
5. Parada por defecto de uno o más equipos

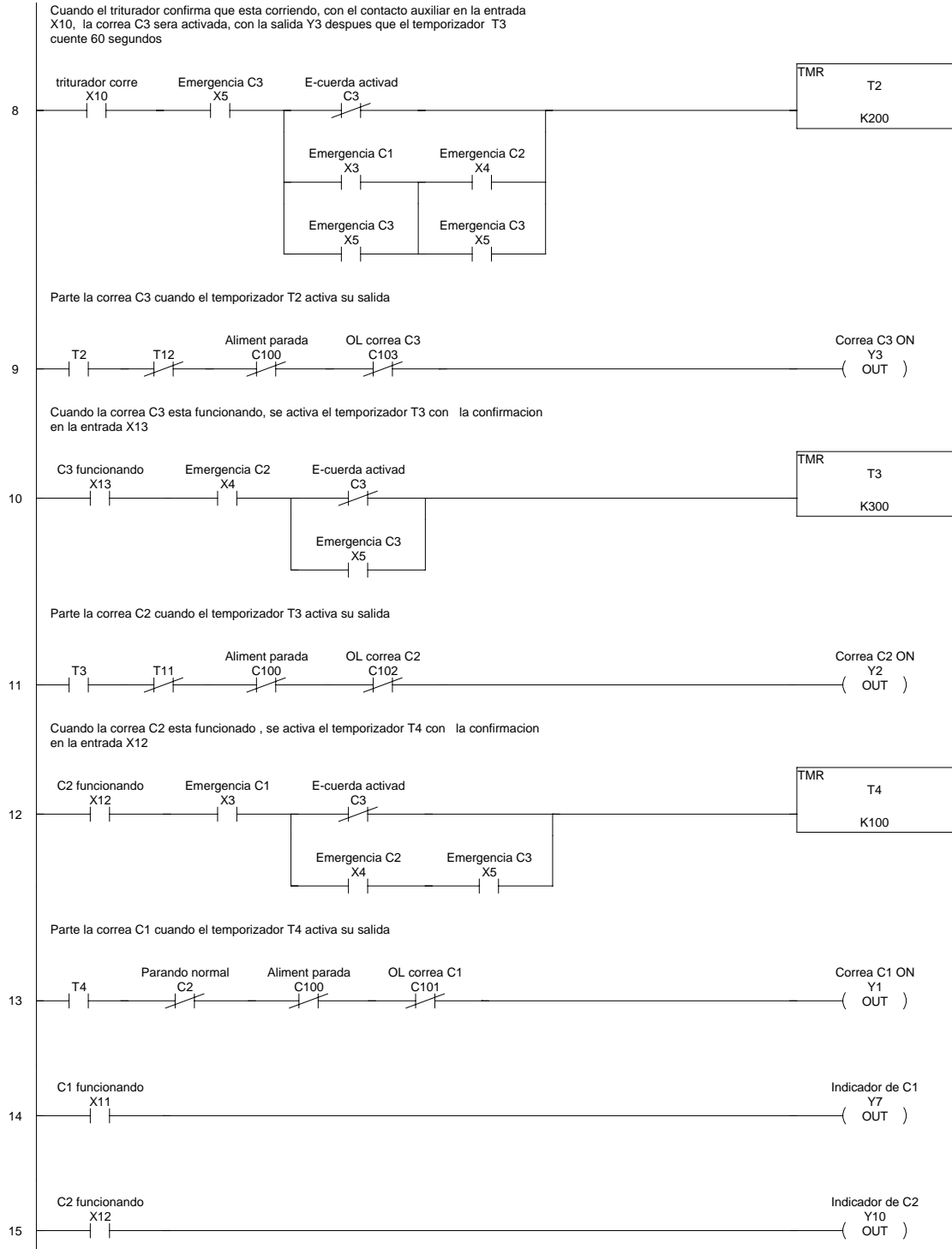
Vea en el diagrama de las próximas páginas como fueron implementados estos eventos.

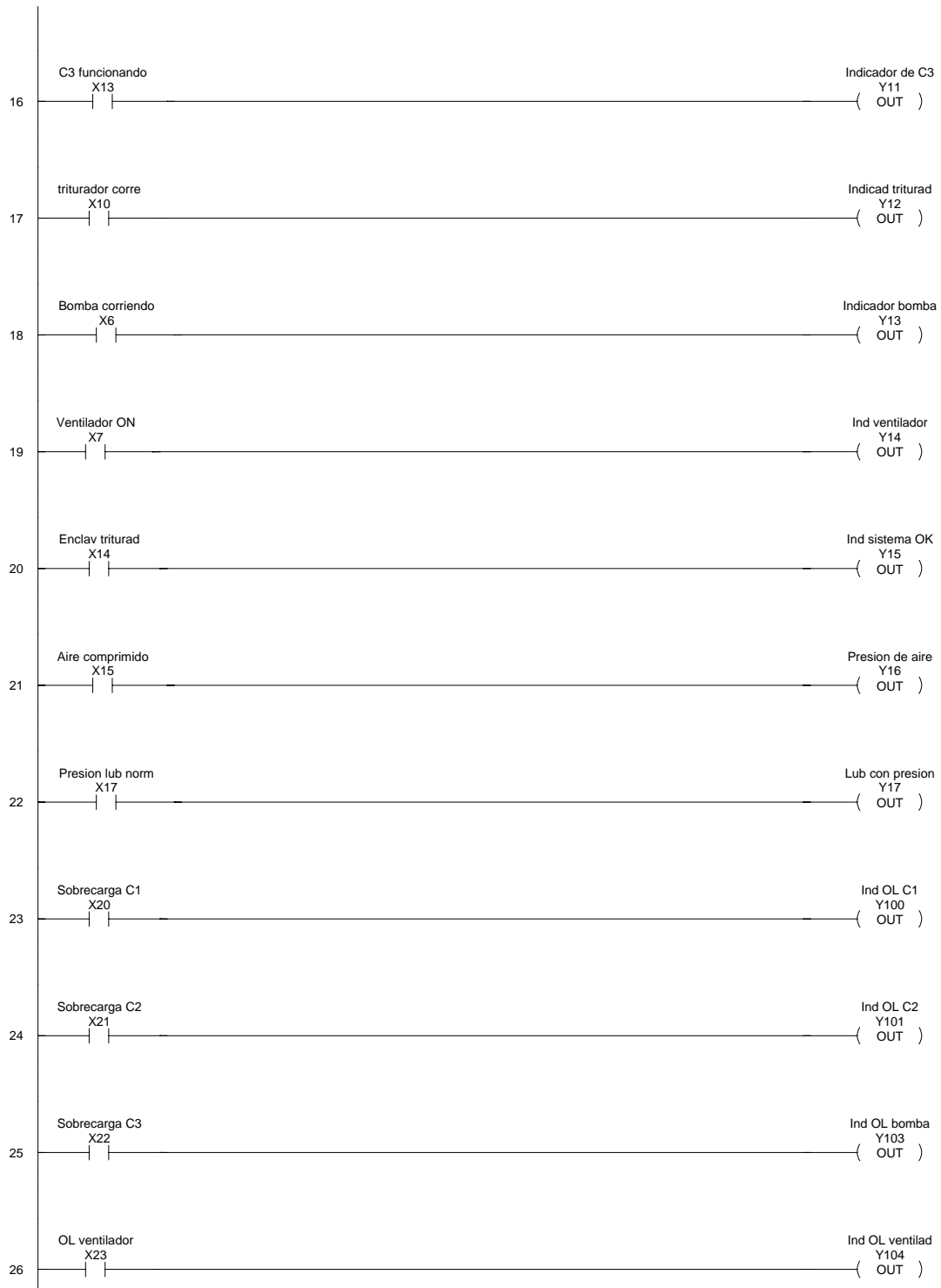
Éstos son:

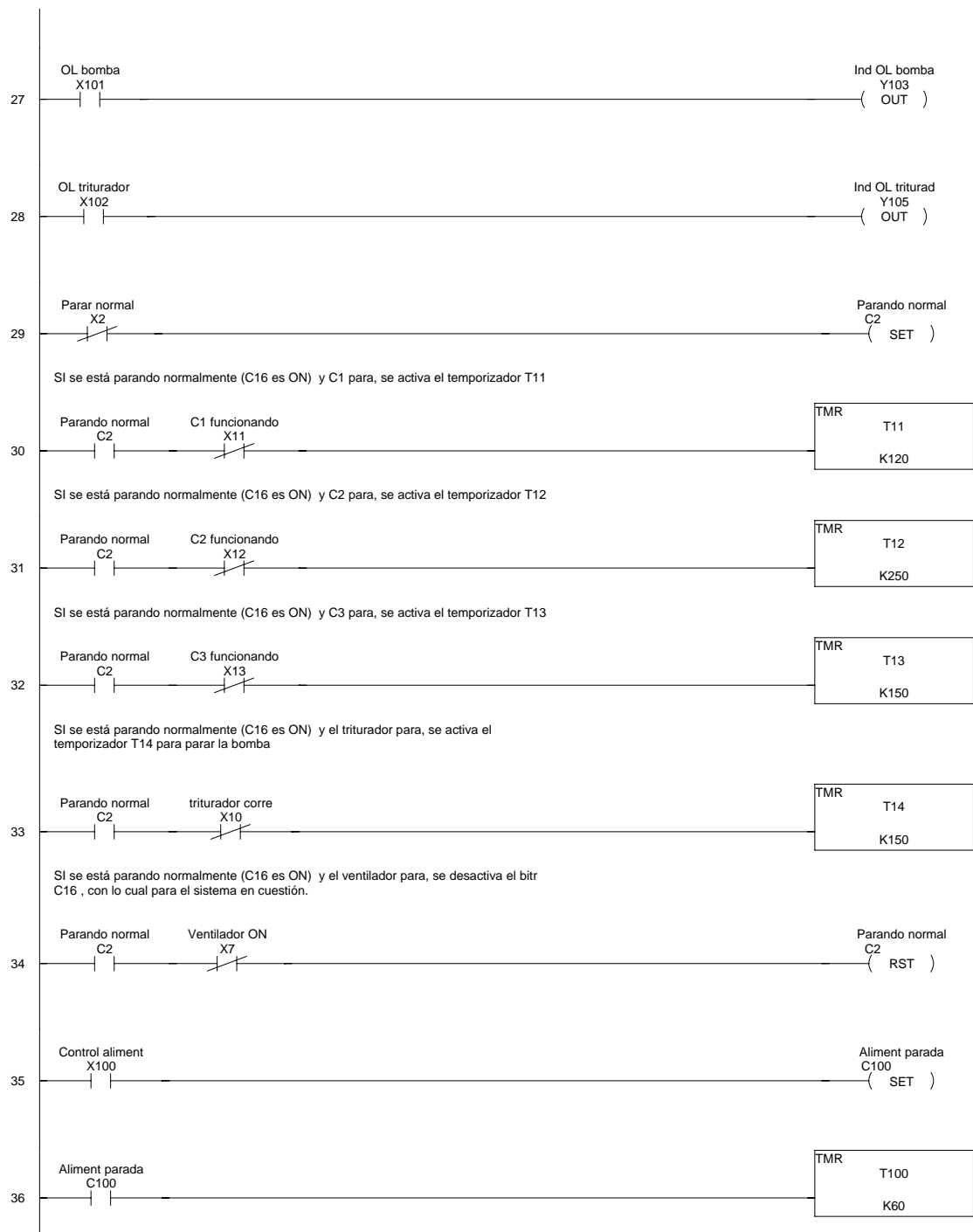
Capítulo 11: Recopilación y ejemplos

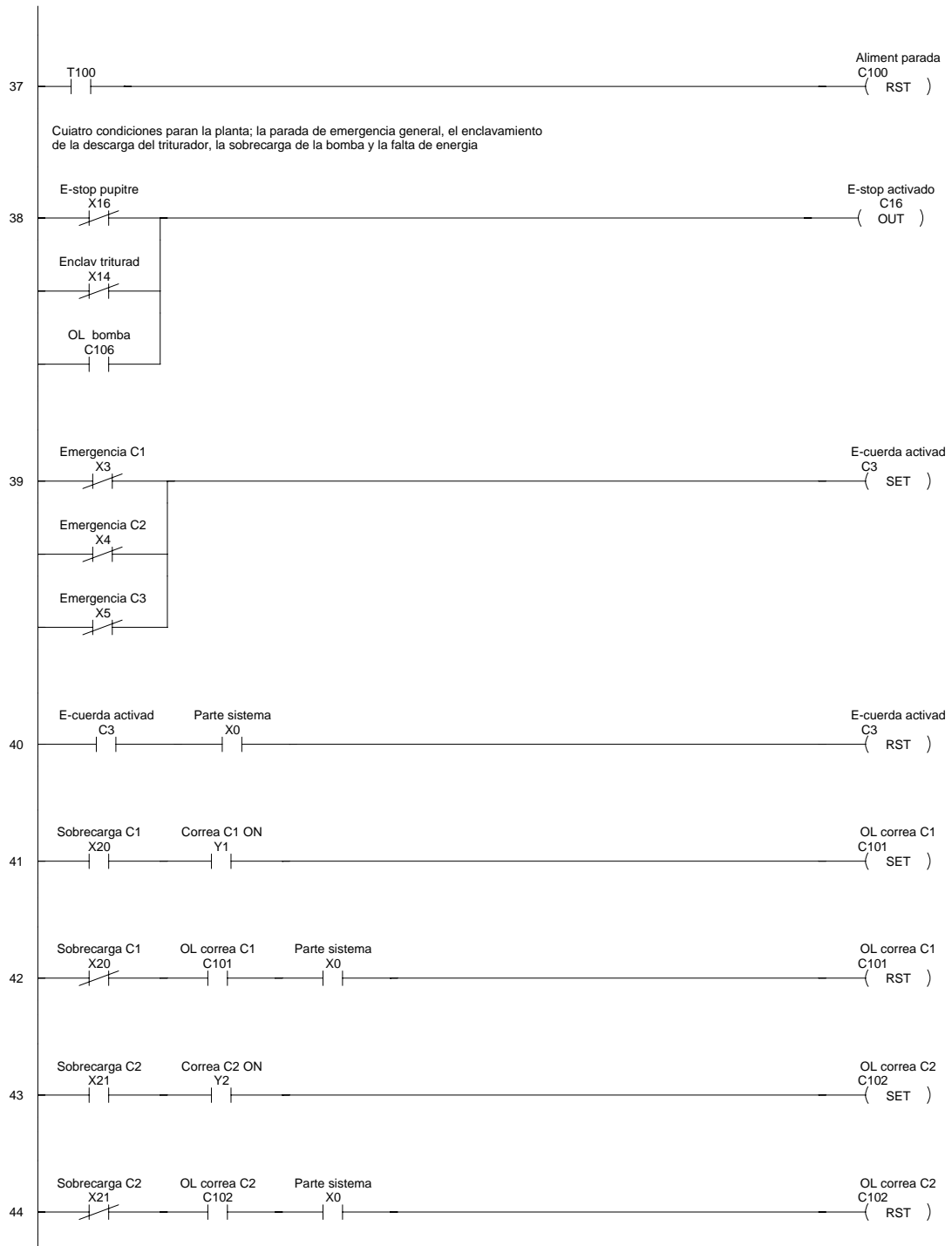
- sobrecarga de cada uno de los motores; en los motores cada confirmación debe causar un enclavamiento con el sistema y parar los demas equipos.
- falla del aire comprimido para la limpieza de las mangas del filtro. Debe causar la parada general de la planta.

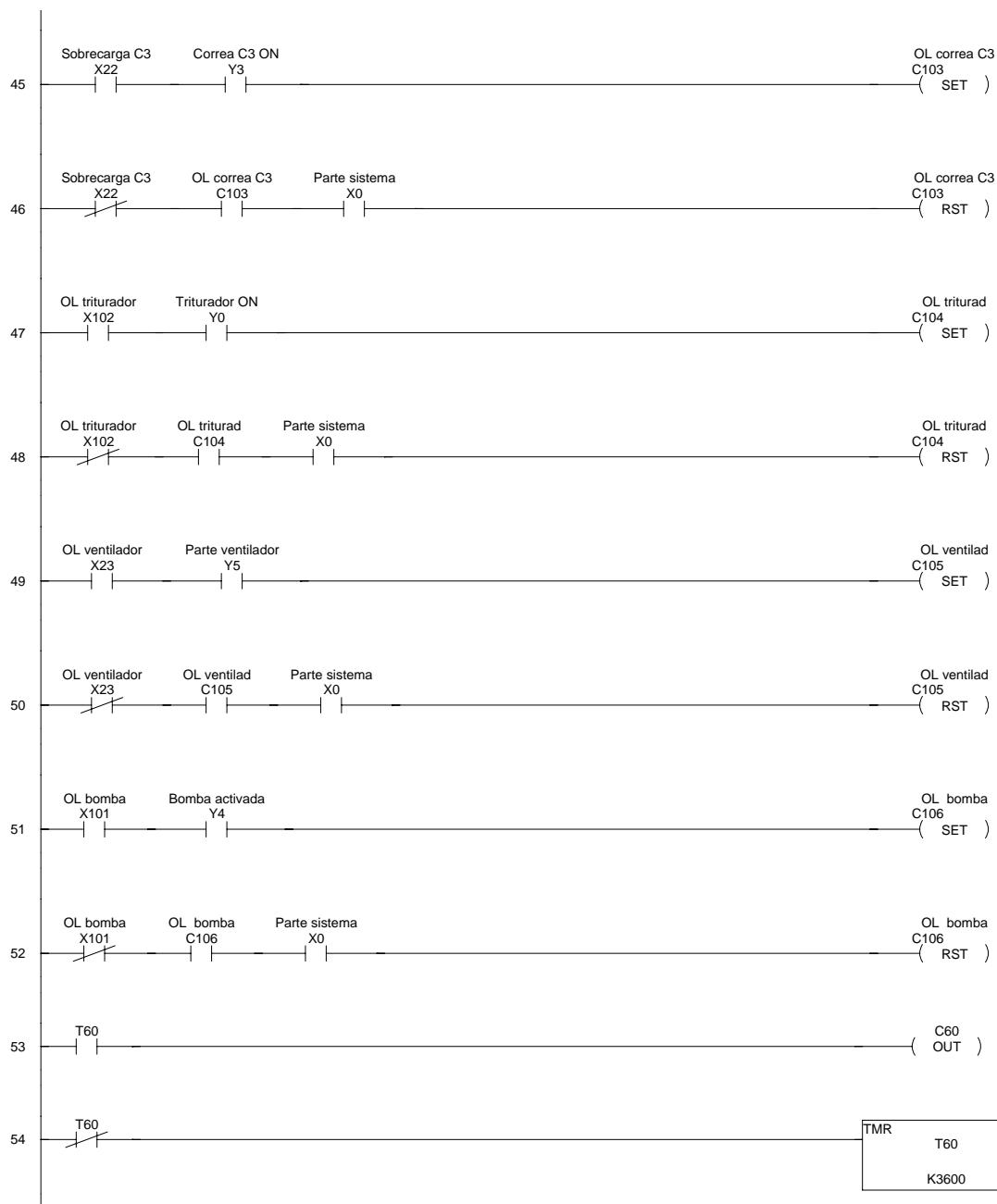


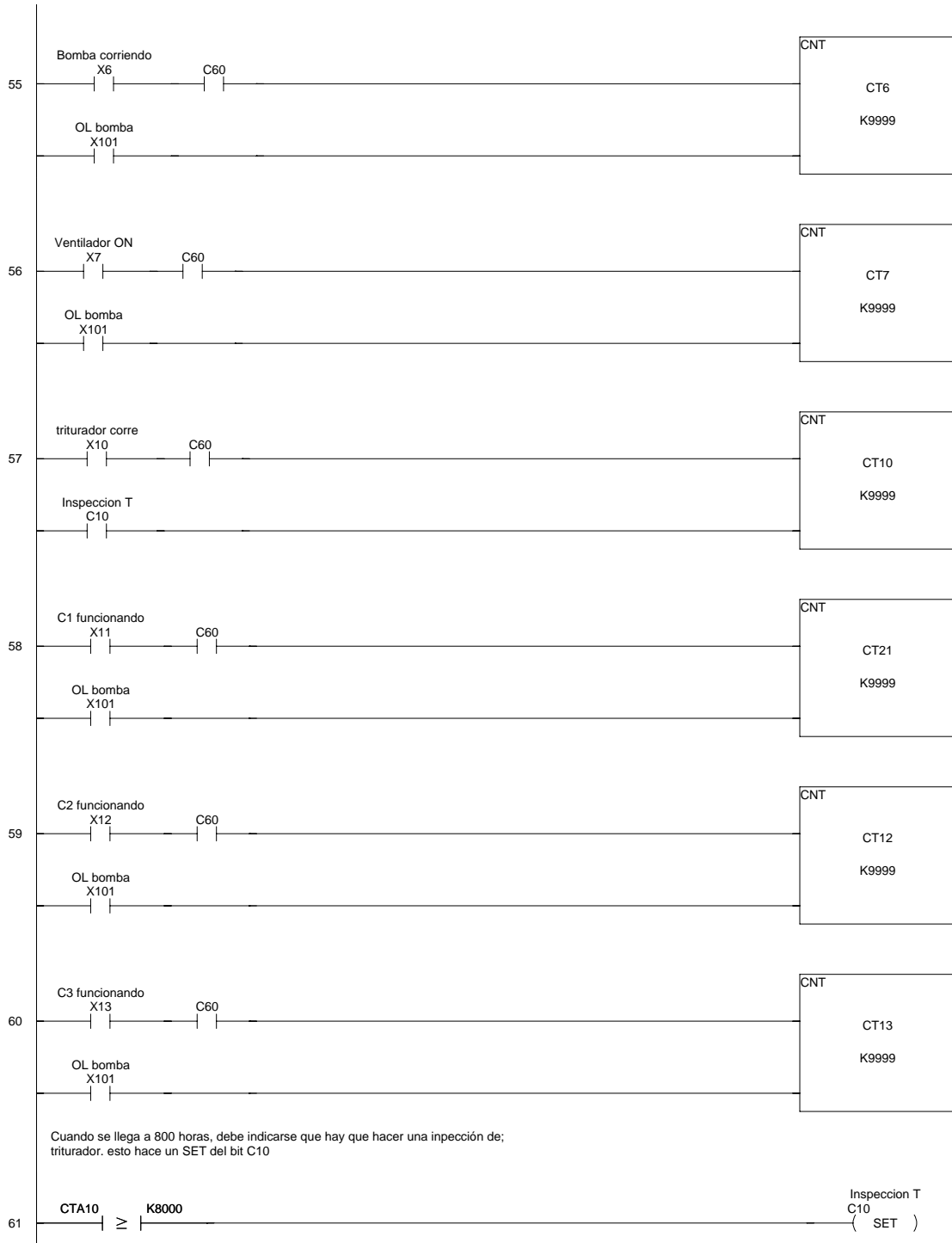


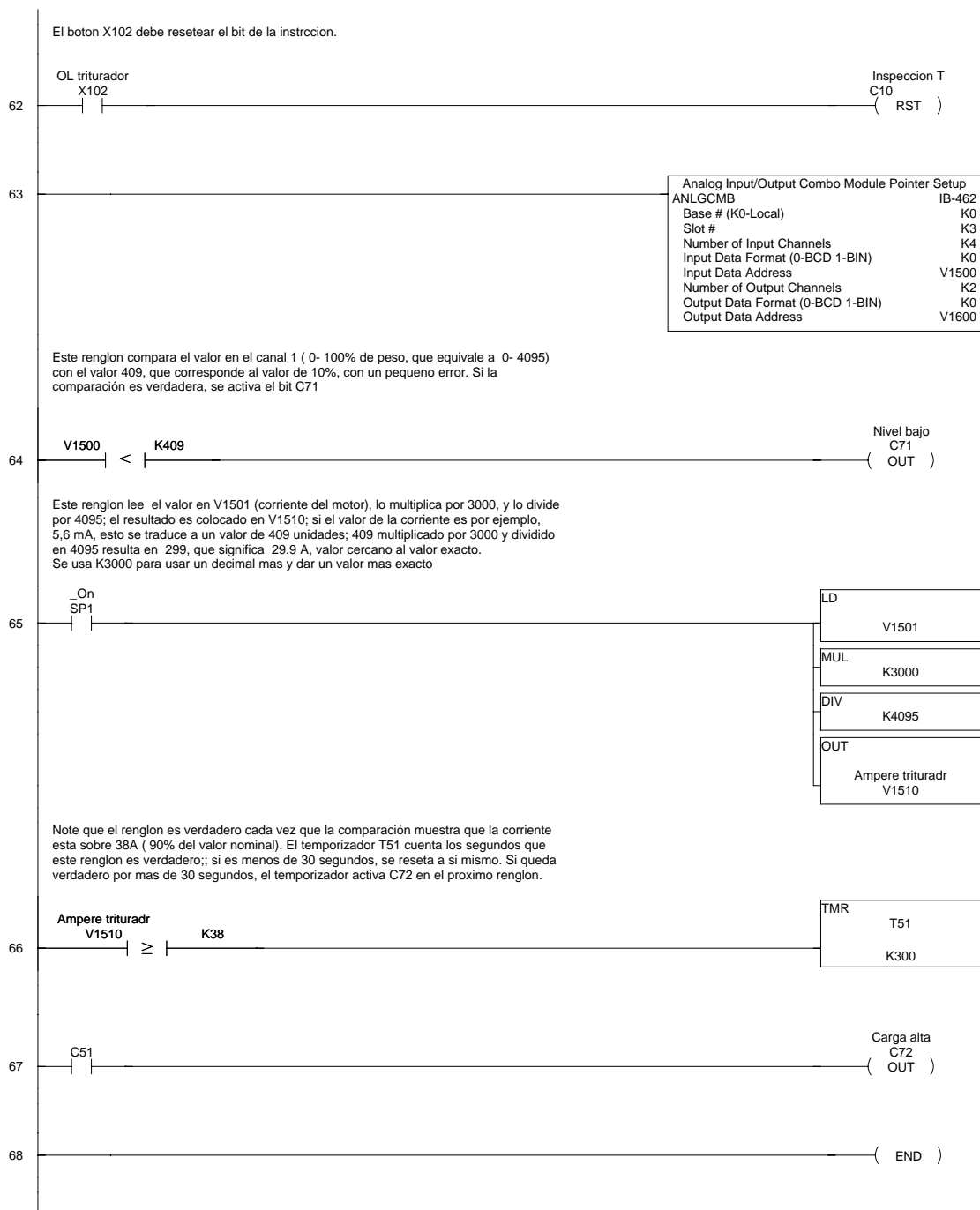












Pasamos ahora a desarrollar otros conceptos requeridos en la implementación de programas con el PLC DL06.

Ejemplo 2: Consideremos usar contadores en el mismo ejemplo. Una función bastante normal es medir las horas que un motor ha funcionado, por ejemplo, para ayudar al departamento de mantenimiento a saber cuando es necesario revisar la grasa en los rodamientos de los motores, o para determinar cuando es el día que corresponde a una inspección después de 1000 horas, por ejemplo.

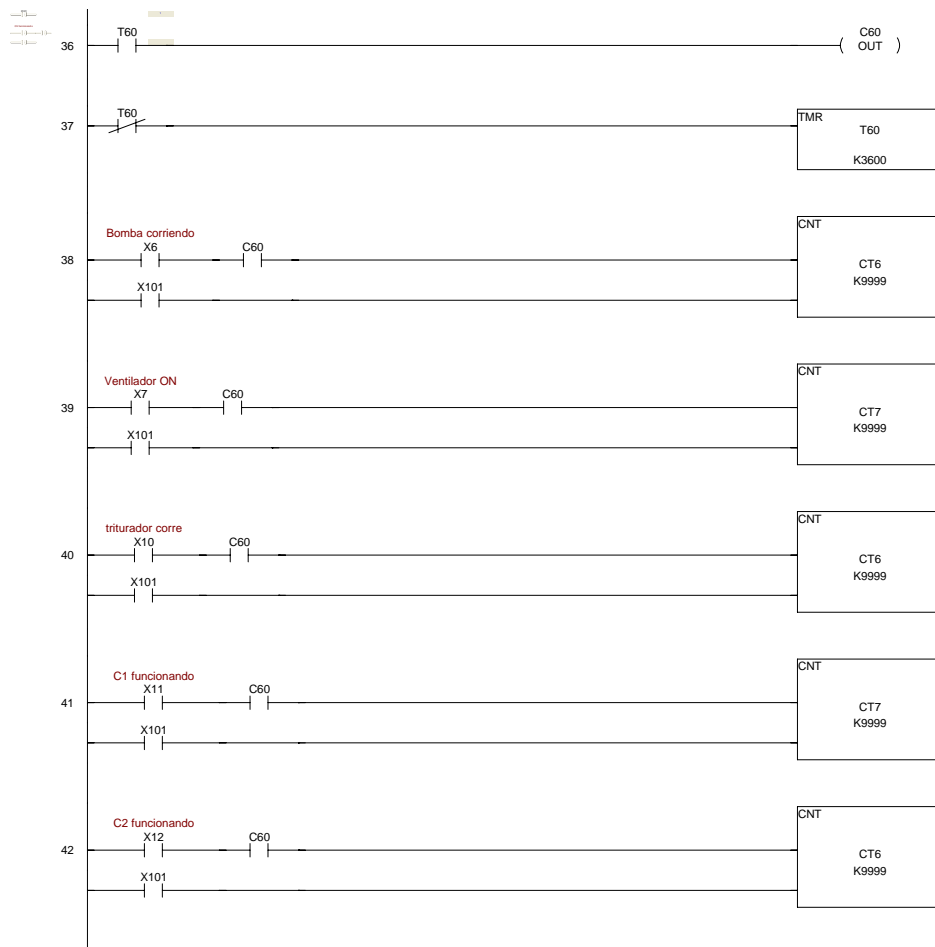
Esto no es un valor exacto hasta el segundo de modo que no sería necesario contarlos, y en ese caso usaremos cada décimo de hora como unidad.

Cada confirmación de funcionamiento de un motor tendrá asociado un temporizador. Recuerde que hay 256 temporizadores y 128 contadores en este PLC. Hasta ahora se han usado solamente unos 10 temporizadores y ningún contador.

Crearemos primero un generador de un pulso con C60 a cada 6 minutos (1 décimo de hora) con el temporizador T60. Luego cada entrada de confirmacion de motor corriendo recibe un contador que cuenta en décimos de hora;

Si el contador CT6 cuenta 134 pulsos, esto significa que el motor de la bomba ha corrido 13.4 horas. y así sucesivamente.

Note que podemos contar solamente hasta 999 horas y 9 décimos; esto equivale a unos 41 días;

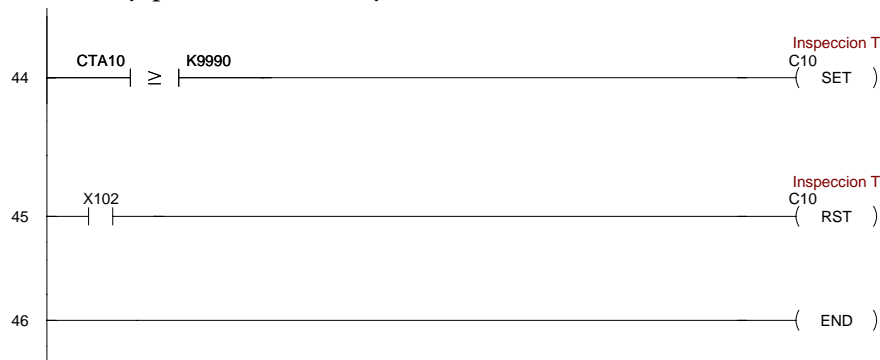


posiblemente sea interesante tener más horas. En ese caso se puede usar el contador UDC, que permite contar hasta 99999999 pulsos. Esto se ha implementado en el caso del triturador.

Note que deberíamos crear un nuevo botón, el botón X101, no previsto en la especificación inicial, para llevar a cero los contadores. También se podría hacer un “reset” del valor cada Lunes a las 6 :00 de la mañana, con el reloj que tiene el PLC. Dejaremos esto para el lector.

Ejemplo 3: Usando la cantidad de horas acumuladas, podemos hacer comparaciones, para mostrar como se operan con esas instrucciones.

El bit C10 puede ser conectado a una salida real (que no ha sido considerada) o a la sirena, con un pulso de 100 ms, cada minuto, para avisar que es tiempo de inspeccionar el triturador. Cuando eso haya sido hecho, el electricista (no el operador) debe apretar el botón X102 para resetear el bit y parar el aviso. Hay varias otras formas. Solamente se muestra una posibilidad.



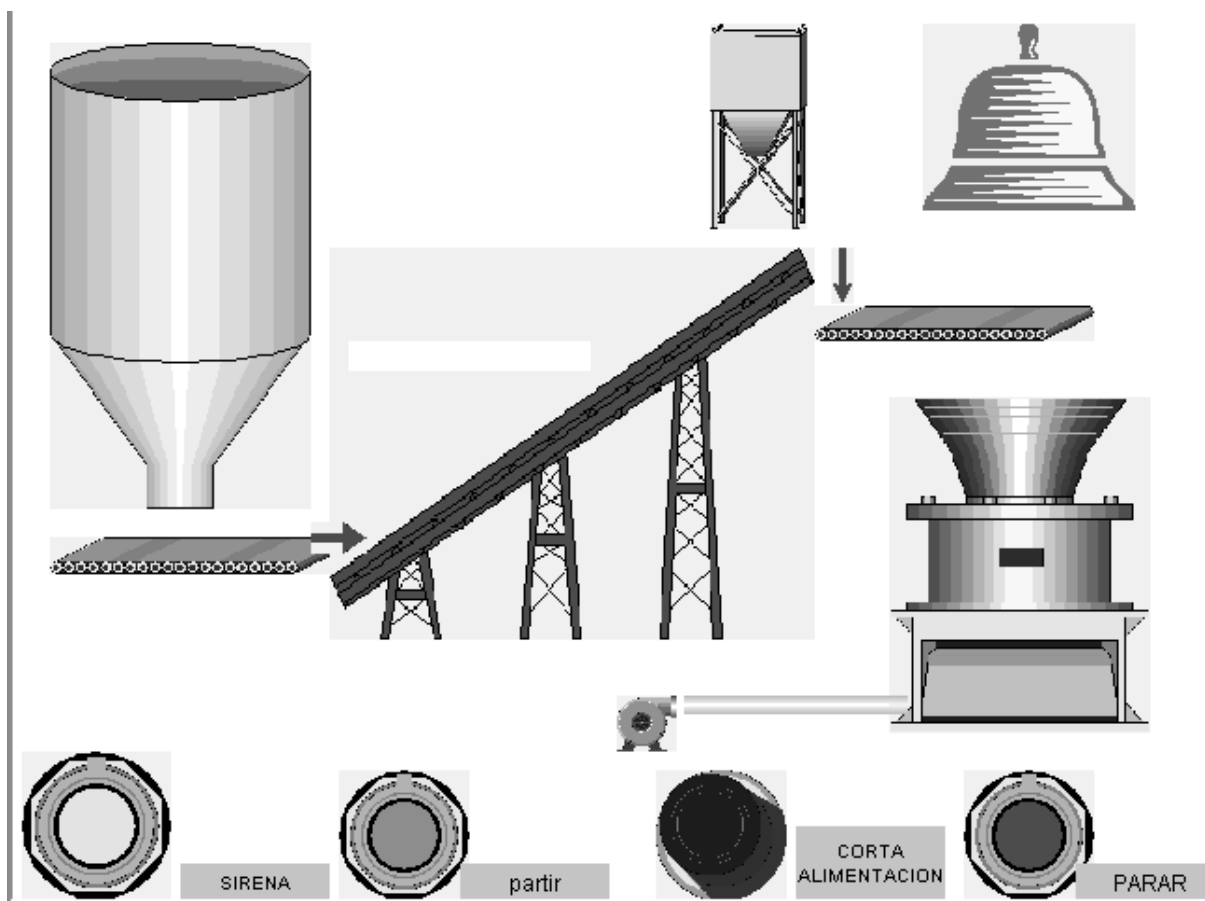
Ejemplo 4; Digamos que se requiere que el operador tenga más informaciones del comportamiento del sistema y para eso instalaremos una interface de operador C-more de 10 pulgadas EA7-T10C, lo que permite tener mas adquisición de datos; los botones e indicadores con lámparas serán reemplazados en ese panel. Observaremos que esta acción libera ahora entradas y salidas físicas del PLC.

En este caso, crearemos una pantalla en el panel con las siguientes asociaciones:

- X0-cambiando a---> C0=====>Objeto bitmap botón “Partir”.
- X1-cambiando a---> C1 =====>Objeto bitmap botón “Sirena”.
- X2-cambiando a---> C2 =====>Objeto bitmap botón “Parar”.
- X6----->Objeto bitmap =====>Bomba corriendo.
- X7----->Objeto bitmap =====>Ventilador corriendo
- X10-----> Objeto bitmap =====>”Triturador corriendo”.
- X11-----> Objeto bitmap =====>” Correa C1 funcionando”.
- X12- ----->Objeto bitmap =====>.”Correa C2 funcionando”.
- X13----->Objeto bitmap =====>”Correa C3 funcionando”.
- X14----->Objeto bitmap =====>”Descarga funcionando”.

- X15----->Objeto bitmap =====>"Presión de aire correcta".
- X17----->Objeto bitmap =====>"Presión de lubricación correcta".
- X20----->Objeto bitmap =====>"Sobrecarga Correa C1".
- X21----->Objeto bitmap =====>"Sobrecarga Correa C2".
- X22----->Objeto bitmap =====>"Sobrecarga Correa C3".
- X23----->Objeto bitmap =====>"Sobrecarga ventilador".
- X100----->Objeto bitmap =====>"Corta alimentación". (un botón)
- X101----->Objeto bitmap =====>"Sobrecarga de bomba".
- X102----->Objeto bitmap =====>"Sobrecarga del triturador".

Ésto se muestra en la figura a continuación:



Como puede notar, el uso de un panel de interface de operador puede mostrar el proceso de una forma mucho más clara que usando solamente indicadores, no se necesitan usar tantas salidas físicas para encender las lámparas de los indicadores, se puede conectar solamente un cable a PLC en vez de hacer un alambrado punto a punto. la pantalla ocupa en espacio menor. Se deja todavía el botón de parada de emergencia principal por razones de seguridad.

La conexión al PLC es hecha a través del puerto 2, como conexión serial RS-232. Se pueden crear más pantallas con gráficos de tendencia, alarmas, etc.

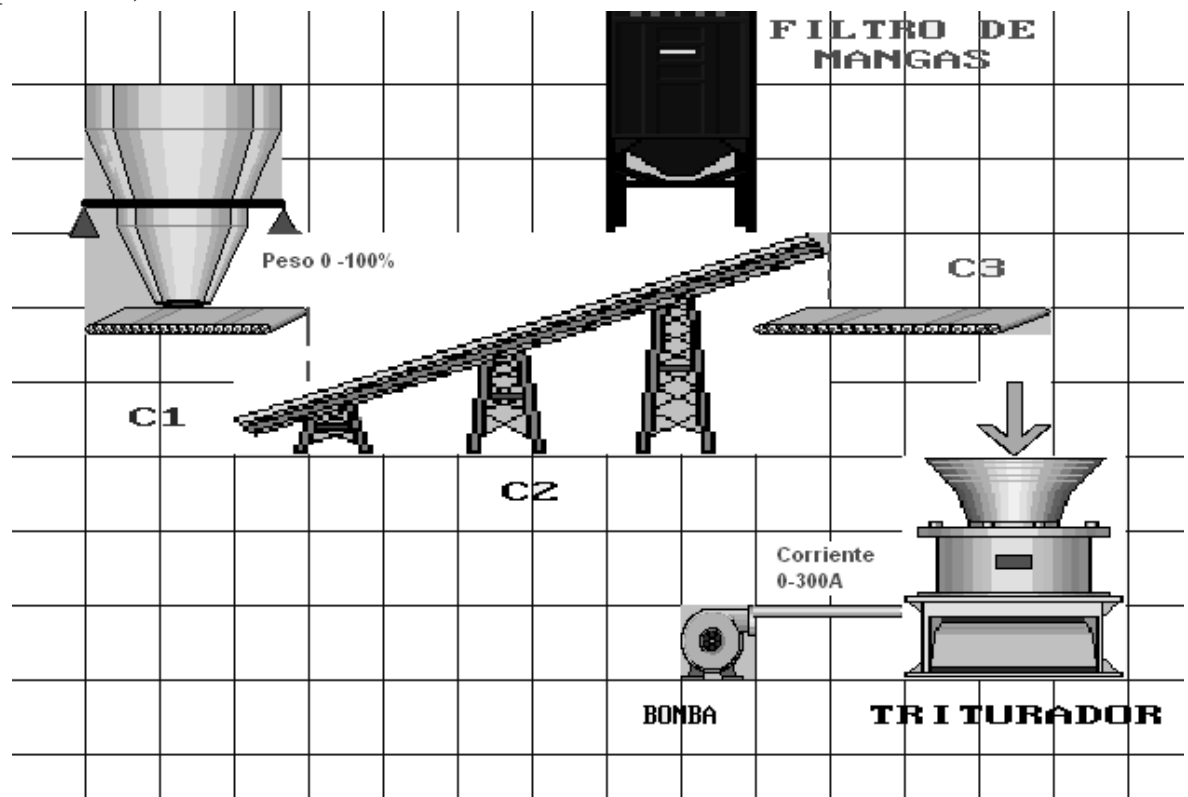
Vea más detalles en el manual del panel C-More.

Ejemplo 5: Hagamos ahora un ejemplo en que queremos analizar el comportamiento de señales análogas. Utilizando un transductor de corriente podemos ver el valor de corriente en el motor del triturador y determinar el nivel de mineral en el “Depósito de mineral” en la alimentación para evitar que las correas y el triturador corran vacíos. En ese caso, aceptaremos una señal de menos de 10% del nivel como depósito de mineral vacío y la corriente del motor por 30 segundos menor que 50 % de la corriente nominal indicará que el sistema está corriendo sin mineral. Debe ser generada una alarma. Veamos primero que necesitamos en el sistema.

- Un transductor con señal de 4-20 mA que indique el peso del material en el depósito de mineral (u otro similar tal como nivel). 20 mA indica 100% del peso de mineral.
- Un transductor de corriente 4-20 mA que indique la corriente del motor del triturador, como medio de indicar el consumo. El motor tiene una corriente nominal de 42 A y el partidor usa un transductor de corriente con relación 0-300A/4-20 mA.

A continuación tenemos un diagrama, que muestra el arreglo de este ejemplo.

Los transductores serán conectados a un módulo de entradas análogas instalado en una de las ranuras del PLC DL06, tal como el F0-04AD2DA-1 (Salidas a ser usadas por otras aplicaciones).



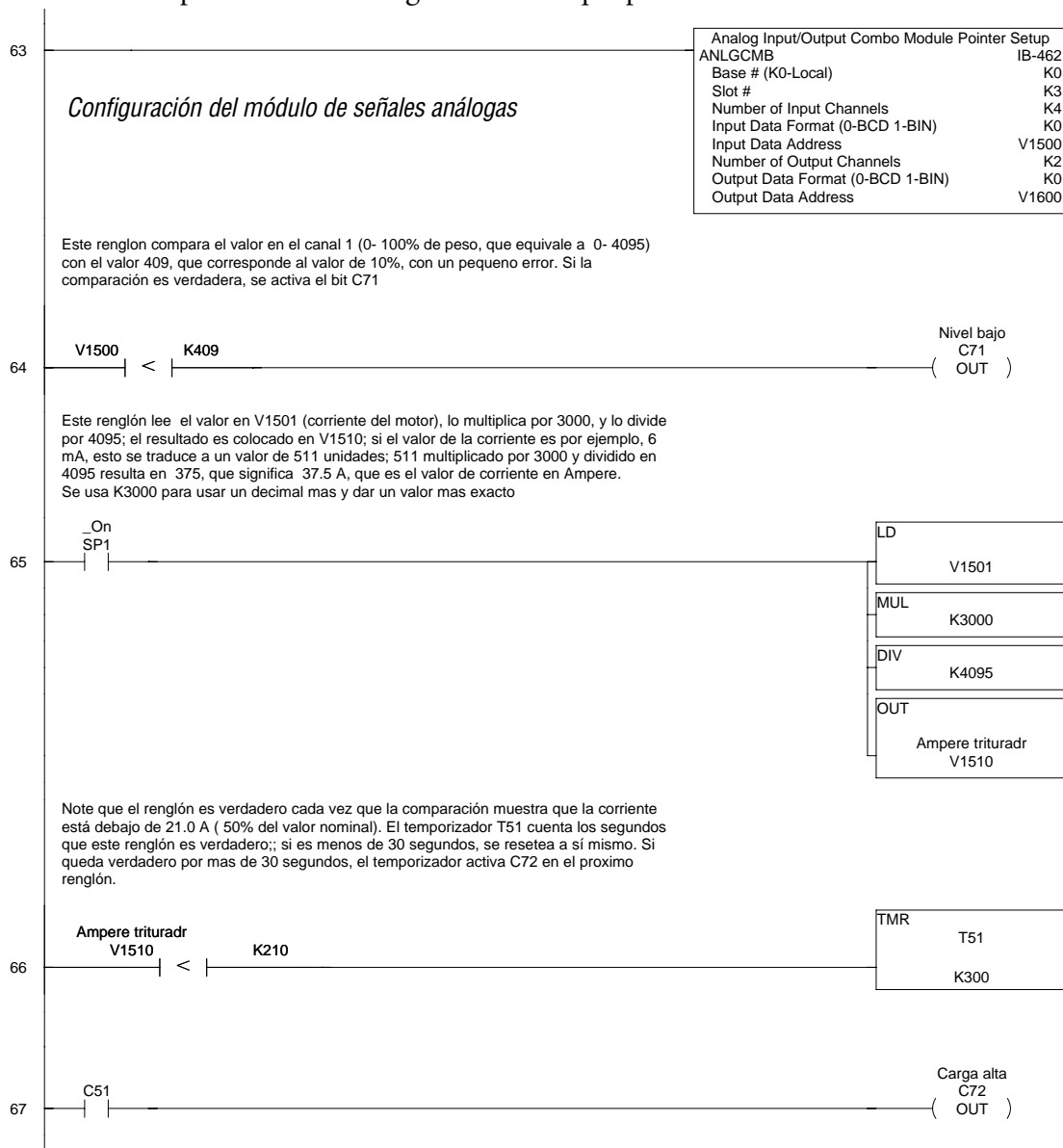
Capítulo 11: Recopilación y ejemplos

Este módulo debe ser configurado en el PLC para que el PLC sepa en qué ranura está colocado, en que registros se van a leer los datos y en qué formato van a ser leídos los niveles de señal.

Seleccionaremos el rango V1500 hasta V1503, correspondiente a los canales 1, 2, 3 y 4 para leer los datos en BCD. Vea más explicaciones en el programa. V1500 es el nivel de material y V1501 es la corriente del motor

Usamos una comparación para determinar que el nivel está a 10% o menor para activar una alarma. Esta salida es directa; en el caso de la corriente del motor del triturador, es necesario colocar un temporizador que cuente al menos 30 segundos para generar esa alarma. Usamos también una codificación para hacer escala para presentar el valor de corriente directamente en décimos de Ampere.

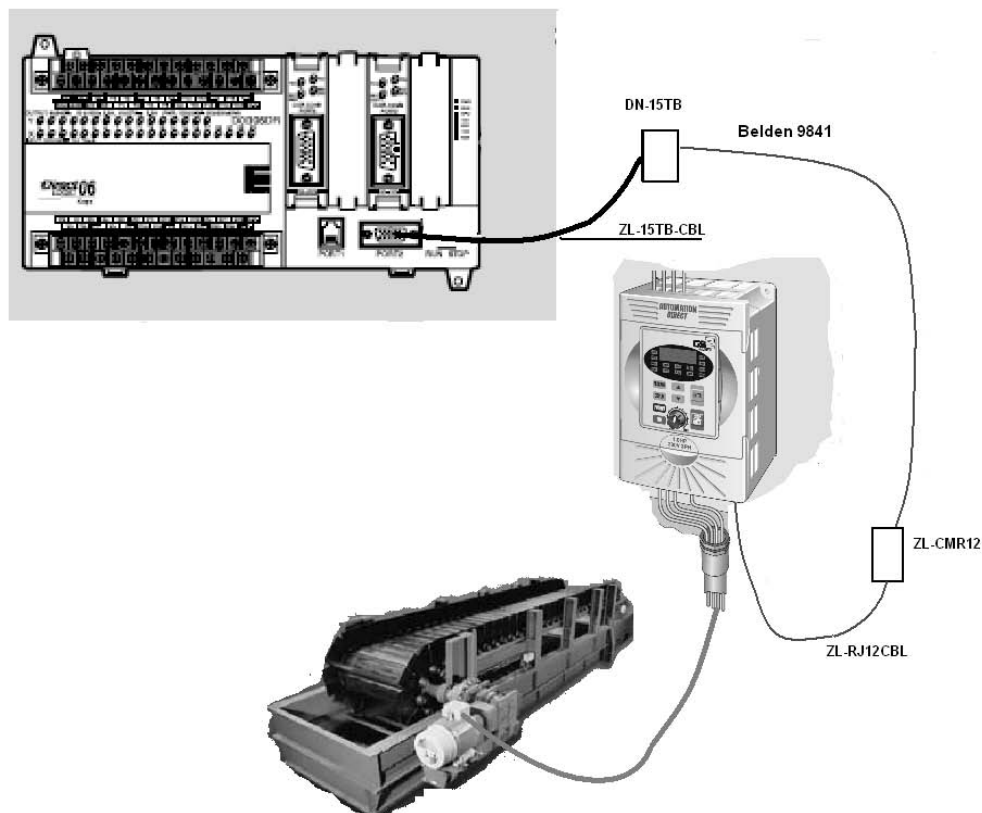
A continuación presentamos el diagrama ladder que permite hacer este control.



Ejemplo 6: Digamos que queremos controlar el flujo de material de la correa alimentadora C1 ajustando la velocidad en un rango de velocidad de 40 a 100% y para eso colocaremos un controlador en el motor de la correa. Éste tendrá un variador de frecuencia GS2-23P0 en el motor de la correa.

Este variador de frecuencia será conectado como esclavo en una red MODBUS RS-485. El PLC usará el puerto 2 para transmitir la información. Instalaremos el variador de frecuencia cerca del motor para evitar problemas en la transmisión de datos. Si se ha ocupado el puerto 2 del PLC DL06 para otro uso, se podría usar un módulo D0-DCM para usarlo como maestro en la red al variador. Vea mas datos del variador de frecuencia en el manual que existe en español en nuestro sitio de Internet.

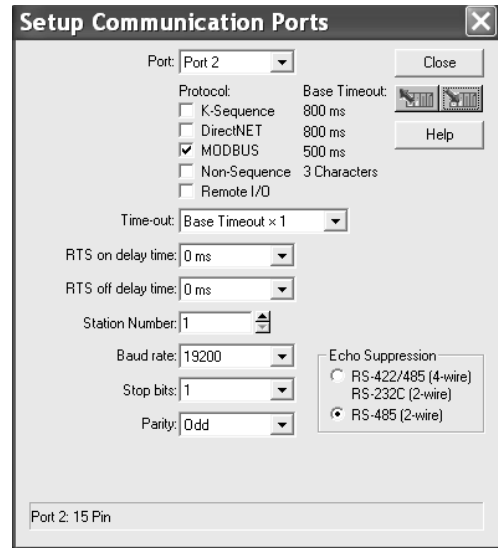
Necesitamos escribir a los registros de la referencia de velocidad, dirección 42331, y al registro de comando de partir y parar, en la dirección 42332. Los otros parametros tales como aceleracion, datos del motor, etc, se colocan directamente en el variador. Solamente le informaremos al variador cuando partir y parar y a qué velocidad. El maestro es el PLC DL06



y el puerto 2 debe ser configurado para trabajar con el protocolo MODBUS RTU y debe definirse la velocidad de transmisión, el tratamiento de la falla de comunicación y en general los demás datos de comunicación. Esto se hace con el programa *DirectSOFT*, como se muestra a continuación.

La configuración del puerto 2 se podría hacer por código ladder, para evitar que se pueda perder esta configuración si faltara la energía eléctrica por más de 4 días y no se ha instalado una batería en el PLC.

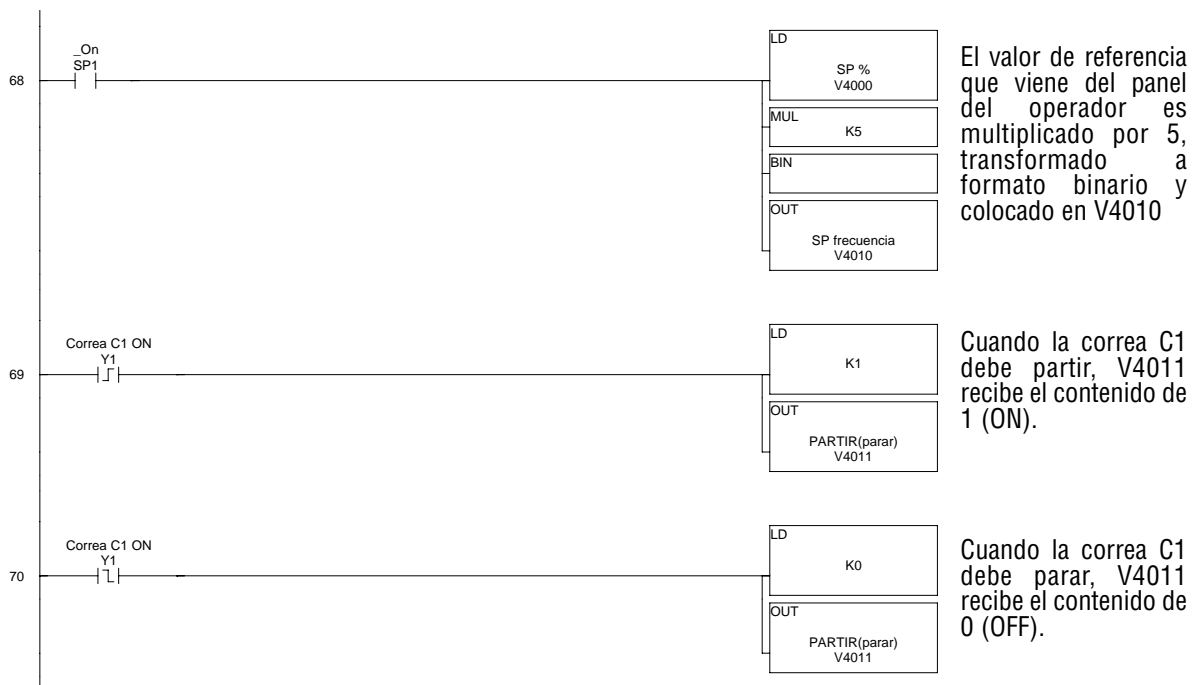
Estando conectado el PLC a la PC con *DirectSOFT*, vaya al menú **PLC**, luego **SETUP** y por último “**Setup sec. comm port...**”. Aparecerá el diálogo mostrado en la figura adyacente.

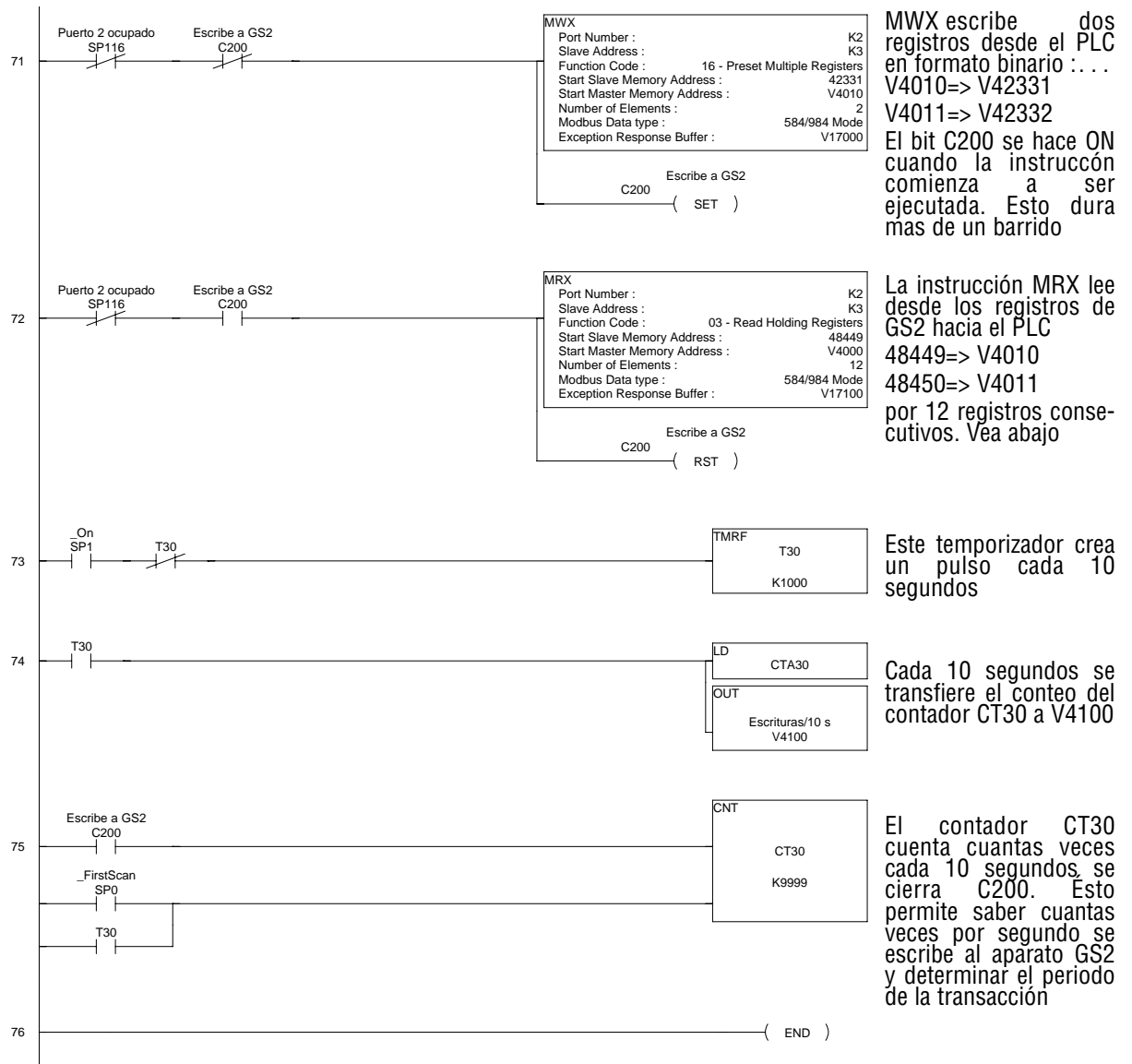


Coloque los datos como mostrado en la figura y asegúrese que los mismos datos está colocados en el variador de frecuencia. Luego es necesario crear el código ladder para permitir que la información sea transmitida. El valor de referencia para velocidad es dado por el panel de pantalla táctil C-more, en % en el registro V4000, como valor BCD de 16 bits. Debemos transformar ésto a frecuencia. También, el mínimo valor es 40%. Eso será ejecutado en el Panel C-more y solamente esperamos un número entre 40 a 100%. Esto corresponde a frecuencias entre 20 a 50Hz. El variador de frecuencia acepta valores decimales (no BCD) y en décimos de Hz. Por lo tanto debemos hacer una operación en el PLC para transformar el valor en % en frecuencia. Eso es fácil multiplicando el valor por 5 ($100 \times 5 = 500$, que es la frecuencia máxima). Colocaremos ese valor en V4010.

Recuerde que habíamos controlado el motor con la salida Y1. Esta salida aún puede ser usada para colocar los datos en el puerto 2 para la transmisión. Cuando se activa, el variador de frecuencia debe partir y cuando se desactiva debe parar. Este bit debe colocar un 1 en la memoria designada para tener este dato, cuando se parte, que por conveniencia lo haremos en V4011. Cuando debe parar, el valor en V4011 debe ser 0.

Vea en el código a continuación como ésto fue implementado en *DirectSOFT*.





Este simple programa permite escribir datos al esclavo 142 veces cada 10 segundos, o 14,2 veces por segundo, cada 70 milisegundos. Esta es la tasa de transferencia con una velocidad de transmisión de 19,2 kbps, para este ejemplo específico.

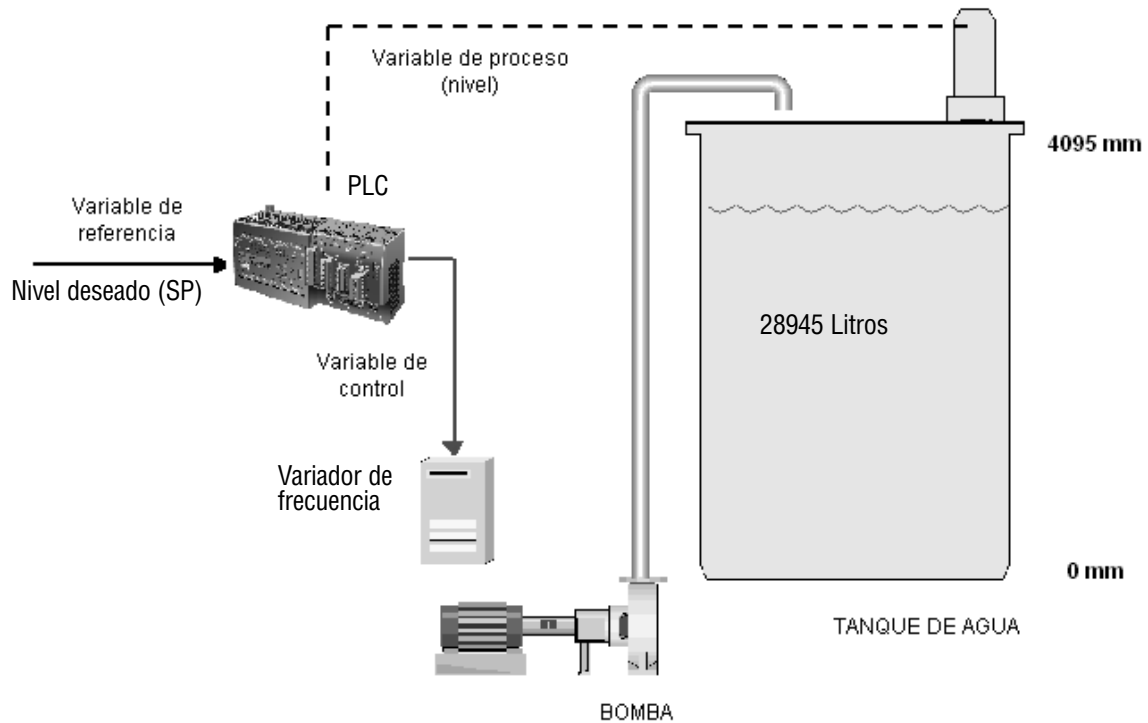
Por otra parte, el PLC tiene un barrido promedio de 4 ms con 76 renglones, como hecho arriba. Aquí se puede ver que las transacciones duran más de un barrido del PLC.

Las lecturas colocan los datos del estado del variador de frecuencia GS2, así como también la frecuencia de referencia y de salida, la corriente de salida, el voltaje de la barra de corriente continua, el voltaje de salida, el valor estimado de rpm del motor, un valor a escala de la frecuencia, el ángulo del factor de potencia y el porcentaje de carga estimado.

Estos valores se pueden mostrar en el panel del operador o hacer uso de los valores para otras funciones.

Ejemplo 7: Control de un nivel de un tanque de agua usando simulación con números reales.

Este ejemplo considera un lazo de control PID para controlar el nivel de un tanque de agua cilíndrico, que tiene un consumo variable, y una alimentación con una bomba centrífuga activada por un variador de frecuencia. El nivel se mide con un transductor de salida 0-10 Volt.



En este ejemplo se ha simulado el nivel del lazo de control usando código ladder.

La variable de proceso es el nivel del tanque, definido como 0-4095 unidades en V7003 (mm). La referencia de nivel es dado en V7002, en mm, las mismas unidades que la variable de proceso. La salida del PID es expresada en 0-4095 unidades (pero 1750 es 100% de velocidad).

El consumo o perturbación (cantidad de agua que sale del tanque) está definida en V6010, en el rango 120-620 litros por minuto; es decir, siempre hay un consumo variable de por lo menos 120 L/m. El tanque tiene un diámetro de 3 metros y contiene hasta 28945 litros. Por lo tanto, la altura máxima es de aproximadamente 4.09 metros (o 4095 milímetros).

La bomba puede entregar hasta 1200 litros por minuto, de modo que se llena en 26.8 minutos si se comienza desde que el tanque esté vacío y el consumo sea 120 litros por minuto. La bomba gira a un máximo de 1750 rpm.

La relación adaptada entre el caudal de la bomba y la velocidad del motor es:

$$Q \text{ [litros/minuto]} = 1200 * n^2 / 1750^2 \text{ siendo } n \text{ la velocidad de la bomba en rpm.}$$

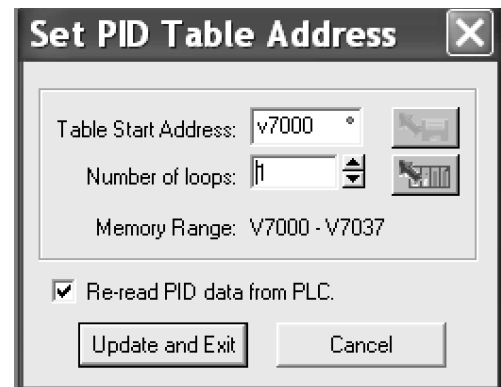
La tabla de PID está en V7000. La tabla de ramp/soak está en el rango V5000 hasta V5037

The PLC tiene un módulo de entradas y salidas como definido anteriormente, con señales de 0- 10 Volt. El transductor de presión entrega una señal de 0 -10 Volt, que corresponde a una señal interna en V2000, de 0- 4095. La salida también será de 0-10 Volt, y para eso es necesario

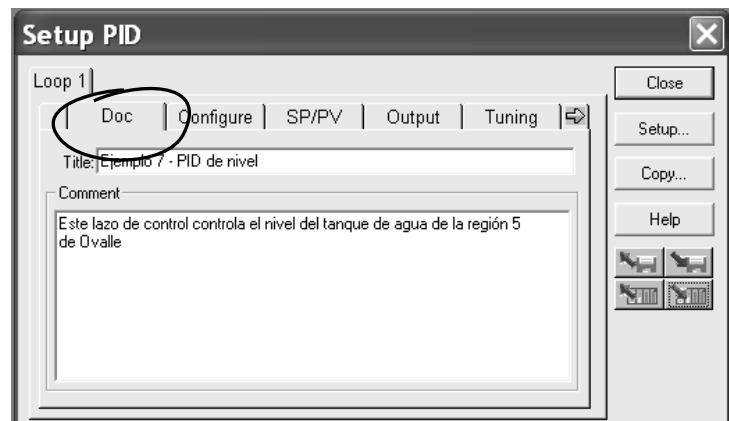
colocar una señal del control de la velocidad del motor escribiendo en la memoria V2010. Antes de ejecutar el programa, es necesario configurar el lazo PID, de la siguiente forma:

Abra el diálogo de setup PID

Conéctese al PLC. Haga clic en el menú PLC, luego en **Setup** y por último en **PID**. Aparece el cuadro de diálogo **PID Table address**, como se muestra en la figura adyacente. Este diálogo le permite definir en que registro comenzará el grupo de parametros de los lazos PID. Cuando haya definido los dos valores que necesita, haga clic en el botón **Update and Exit**, lo que salva la localización de los parámetros y la cantidad de lazos. En este caso elegiremos V7000 y 1 lazo de control.



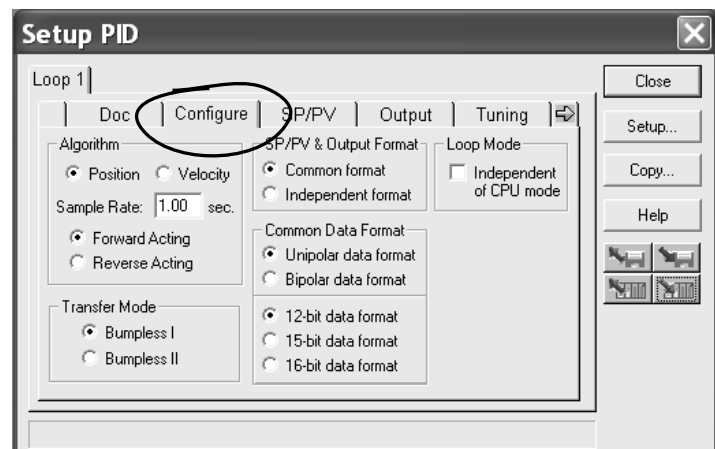
Cuando hace clic en el botón indicado, aparecerá el cuadro de diálogo como se muestra en la figura adyacente. Observe que hay “lengüetas” que seleccionan diversos campos. Por defecto viene en **Doc**. Coloque el título del lazo en el campo **Title**, y comentarios en el campo **Comments**. Esto es opcional.



Luego hay otra lengüeta llamada **Configure**. Este diálogo le muestra como seleccionar las características del algoritmo PID a ser usado.

En este caso, sabemos que debe ser colocado como se muestra en la figura adyacente. Vea más explicaciones en el capítulo 8.

En este caso, la acción directa (**Forward acting**) es necesaria. Si el error (SP-PV) aumenta, la variable de proceso PV aumentará. Usaremos un tiempo de muestreo (**sampling time**) de 1 segundo, ya que no es necesario hacer el lazo más rápido. Usamos también **common format**, ya que la entrada y salida tienen el mismo formato (12 bits, BCD).



Luego debe seleccionar la lengüeta **SP/PV**. Note que las direcciones de estos valores han sido ya seleccionadas.

Aquí, en ese caso, no es necesario cambiar ningún valor. Note que es posible seleccionar la raíz cuadrada (SQUARE ROOT) de la variable de proceso. Esto se hace en el caso de, por ejemplo, placas orificio, que miden presión para determinar flujo.

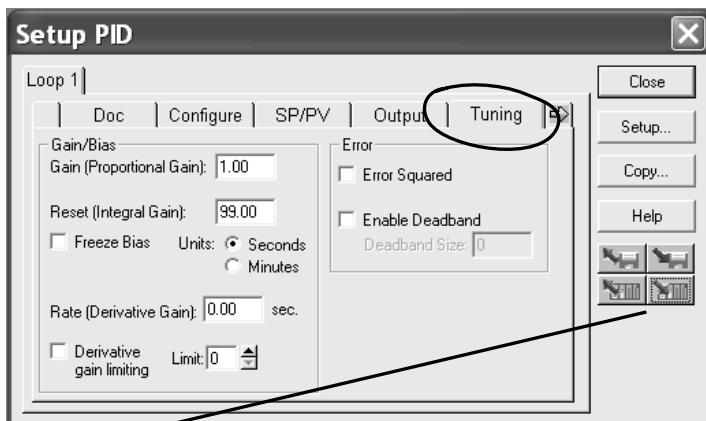
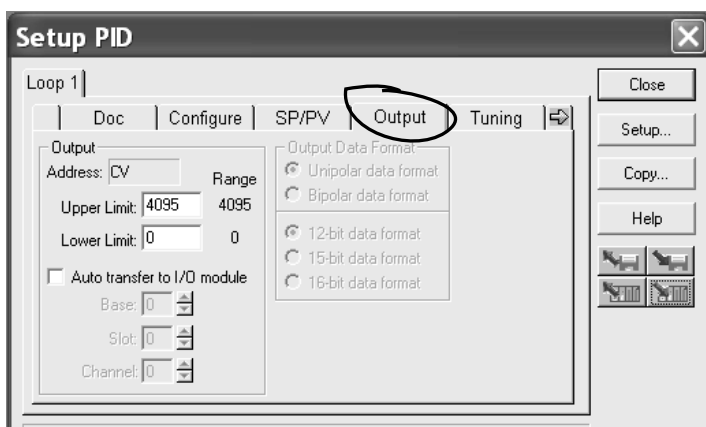
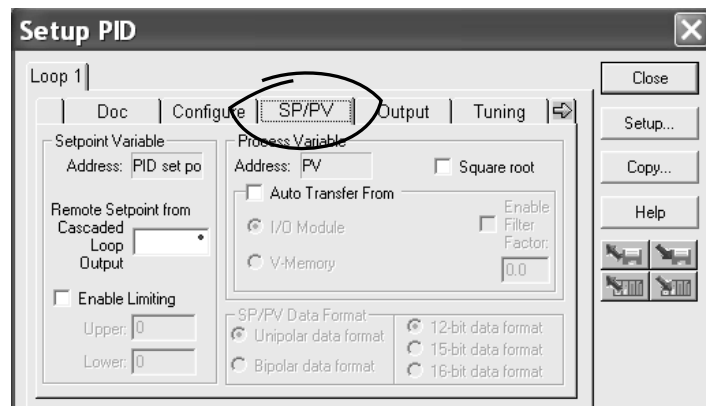
La próxima lengüeta es la lengüeta **Output** (Salida de control) que configura como será el valor numérico de la salida del controlador PID. En el campo **Upper limit** seleccionaremos el valor 4095, en este caso, para limitar la salida al valor máximo que acepta el módulo de salida.

Puede ser que se quiera que la velocidad mínima de la bomba sea de 20%; en ese caso seleccionaríamos un límite inferior. Pero no es necesario en este caso.

No hay otro ajuste que hacer en este caso. Note que siempre el valor límite máximo es 0, por defecto, y si no se coloca un valor diferente, la salida del lazo PID no trabajará.

La próxima lengüeta es la lengüeta **Tuning** (sintonización del lazo) que configura los valores de ganancia proporcional (Gain) y el valor integral (Reset), que son determinados posteriormente, cuando ya esté conectado el sistema de bombeo al PLC. Por ahora, colocaremos el valor 1 en la ganancia y 99 segundos en el reset. No usaremos el factor derivativo. Se puede considerar el uso de freeze bias, pero no lo usaremos en este caso. El lector puede probar el efecto de este acción. Como se hará una simulación, mostraremos esto como si fuera una salida física. Salve los resultados haciendo clic en el botón indicado aquí.

La próxima lengüeta es la lengüeta **Alarms** (alarmas del lazo) que configura los valores límites alarma para este lazo específico, Cada alarma genera un bit activado en uno de los registros de



parámetros del lazo PID. Vea mas detalles en el capítulo 8. No cubriremos esto en este ejemplo. Por último viene la lengüeta **Ramp/Soak**, que es la programación de hasta 8 segmentos de valores de referencia. Analizaremos este asunto más adelante. Usaremos esto solamente para mostrar como funciona el ramp/soak.

A continuación mostraremos como se hace el programa ladder para interactuar con el lazo PID y como se hace la simulación; V2000 lee los rpm del variador de frecuencia. V2010 es el valor que genera la referencia de frecuencia en el variador de frecuencia. Usaremos un panel de interface de operador EA7-T10C, para entregar el nivel deseado en V7002 (SP), que es la variable en el lazo PID que acepta el valor de referencia, así como también los comandos de partir y parar la bomba y de como cambiar del modo Manual a Automático o viceversa.

En este ejemplo, usamos números de punto flotante, conocidos como números reales. Recordemos que, en realidad, es una representación de un número real, pero éste tiene algún error de truncado, que es pequeño, pero hace que el cálculo no sea exacto. En este caso, el error no es importante.

Lo que se hace primeramente en el programa es configurar el módulo de señales análogas con la instrucción ANLGCMB en el renglón 1.

El renglón 2 define constantes en el primer barrido del PLC, usando SP0.

El renglón 3 genera un pulso cada segundo, para minimizar el tiempo de barrido promedio. Note que se hace un enclavamiento con el momento en que el algoritmo del lazo PID se ejecuta. Luego se programa en el renglón 4 el consumo de agua en litros por minuto, ya que la entrada está en el rango 0-4095, se multiplica por el factor $500/4095 = 0.1221$ para obtener litros por minuto, y luego se le agrega 60, para poder considerar que el consumo será al menos 60 litros por minuto.

El renglón 5 lee los rpm desde el variador de frecuencia, como señal 0-10 Volt, que es leído en el rango 0-4095, lo transforma en rpm con el factor $1750/4095 = 0.42735$. Luego es elevado al cuadrado para considerar la fórmula de flujo, y se obtiene el flujo en litros por minuto en la memoria V6000, como número de punto flotante.

El renglón 6 calcula el aporte de agua neto, ya que se alimenta el tanque con la bomba, pero al mismo tiempo hay agua saliendo del tanque. El aporte se coloca en V6016, en litros por minuto, pero se transforma en litros por segundo, para permitir el cálculo del volumen de agua en el tanque, a ser hecho posteriormente.

El renglón 7 calcula el volumen del tanque en litros. El aporte se coloca en V6016, en litros y por facilidad de supervisión con Data View, se coloca también en V6032.

El renglón 8 compara el volumen de agua con la capacidad del tanque.

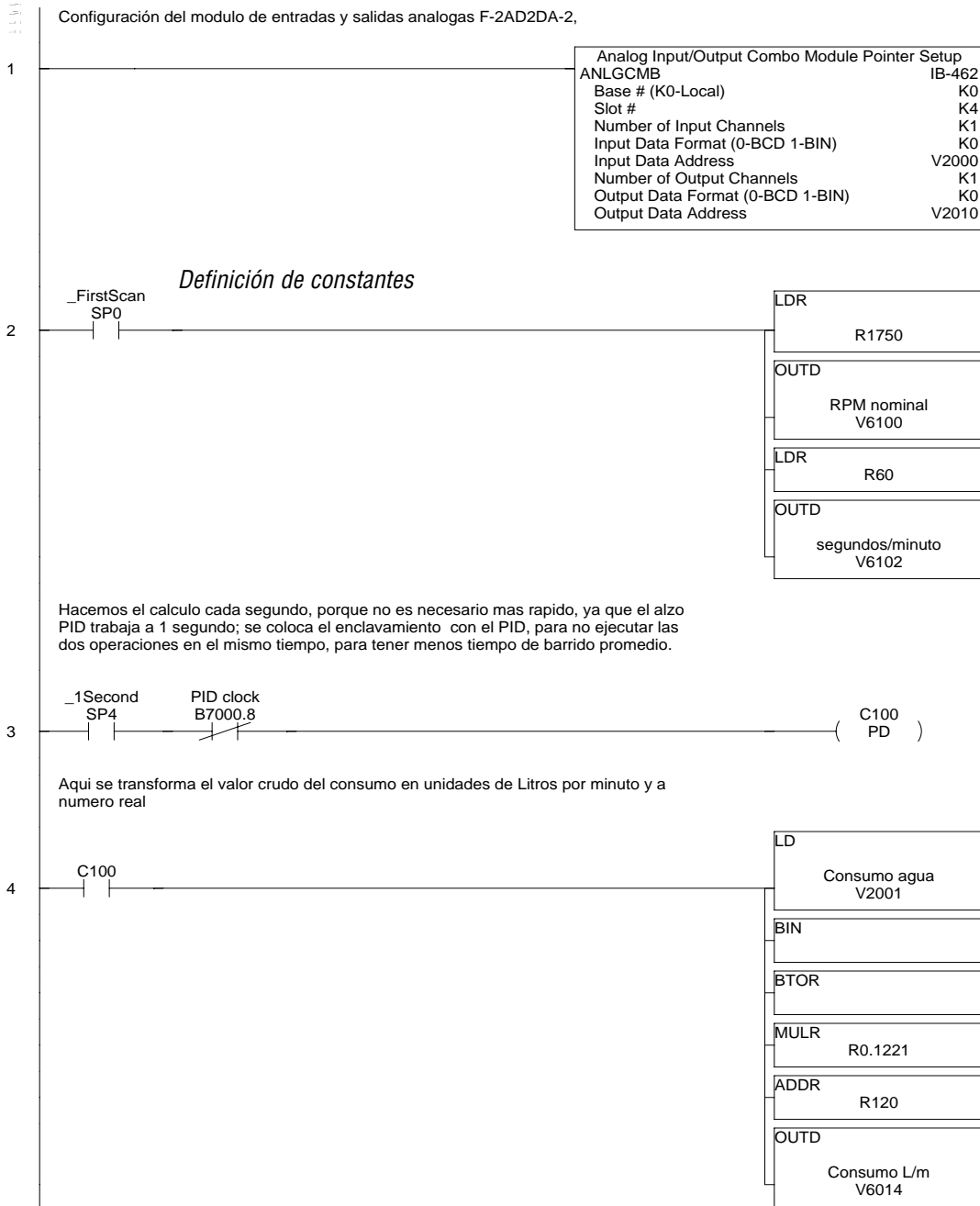
El renglón 9 recibe el resultado de la comparación y se coloca en la memoria el máximo valor que el tanque puede contener, para simular rebalse del tanque.

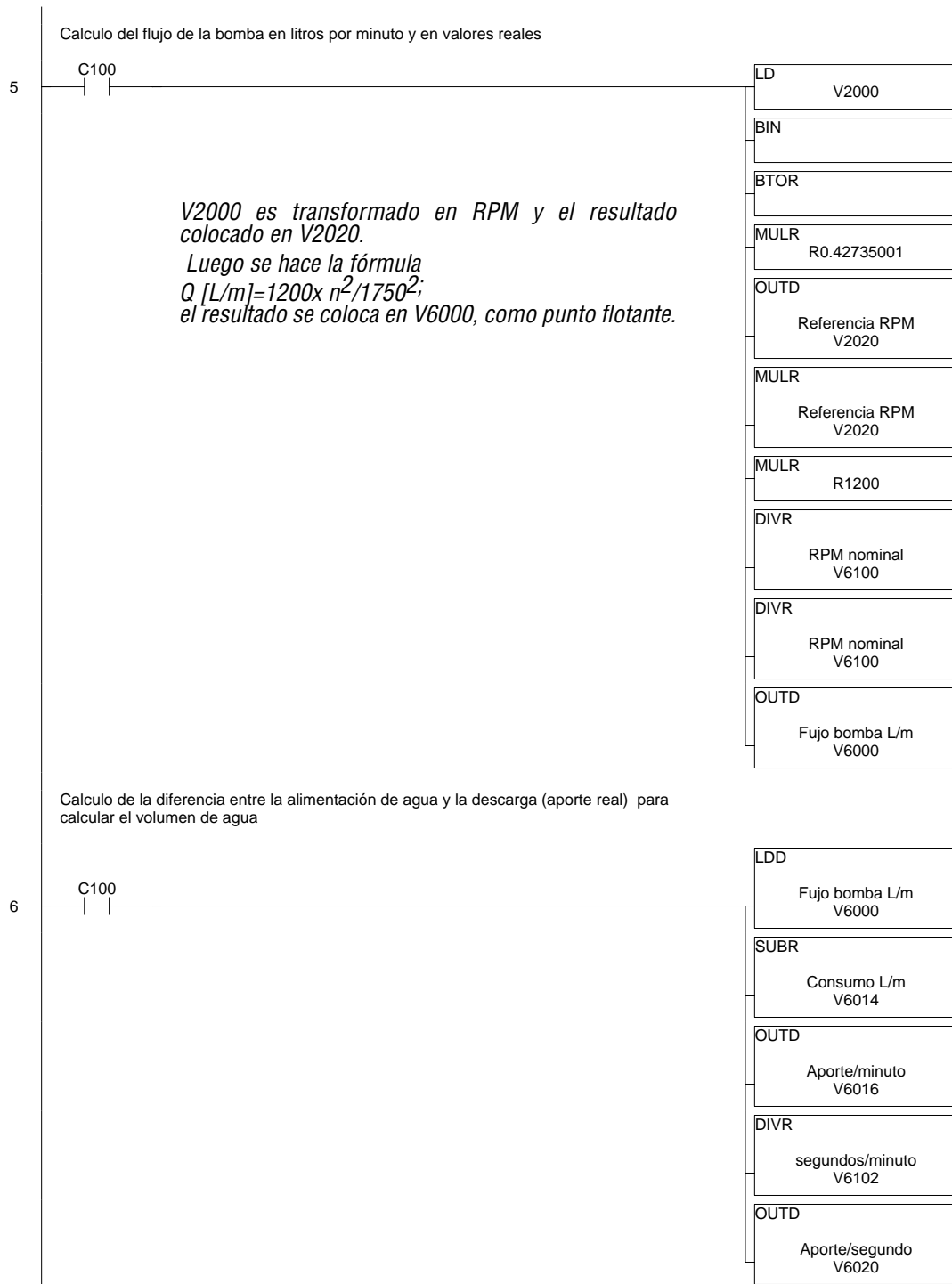
El renglón 10 hace el cálculo del nivel directamente en milímetros y el resultado es colocado en V6040; este valor tiene el formato de número de punto flotante. Este valor se transforma en valor decimal con la instrucción RTOB y se coloca directamente en V7003, la dirección del lazo PID que acepta la variable de proceso.

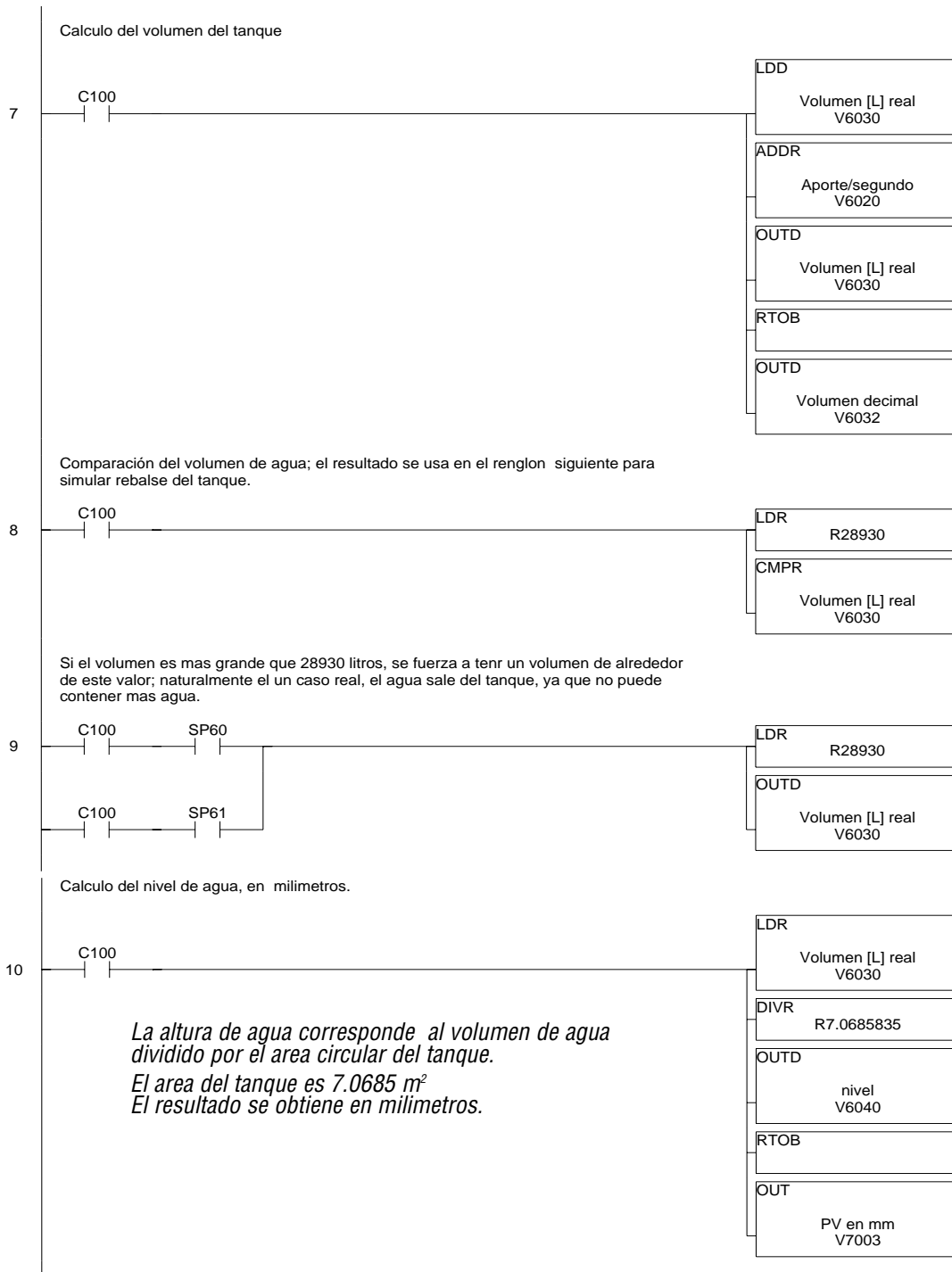
Capítulo 11: Recopilación y ejemplos

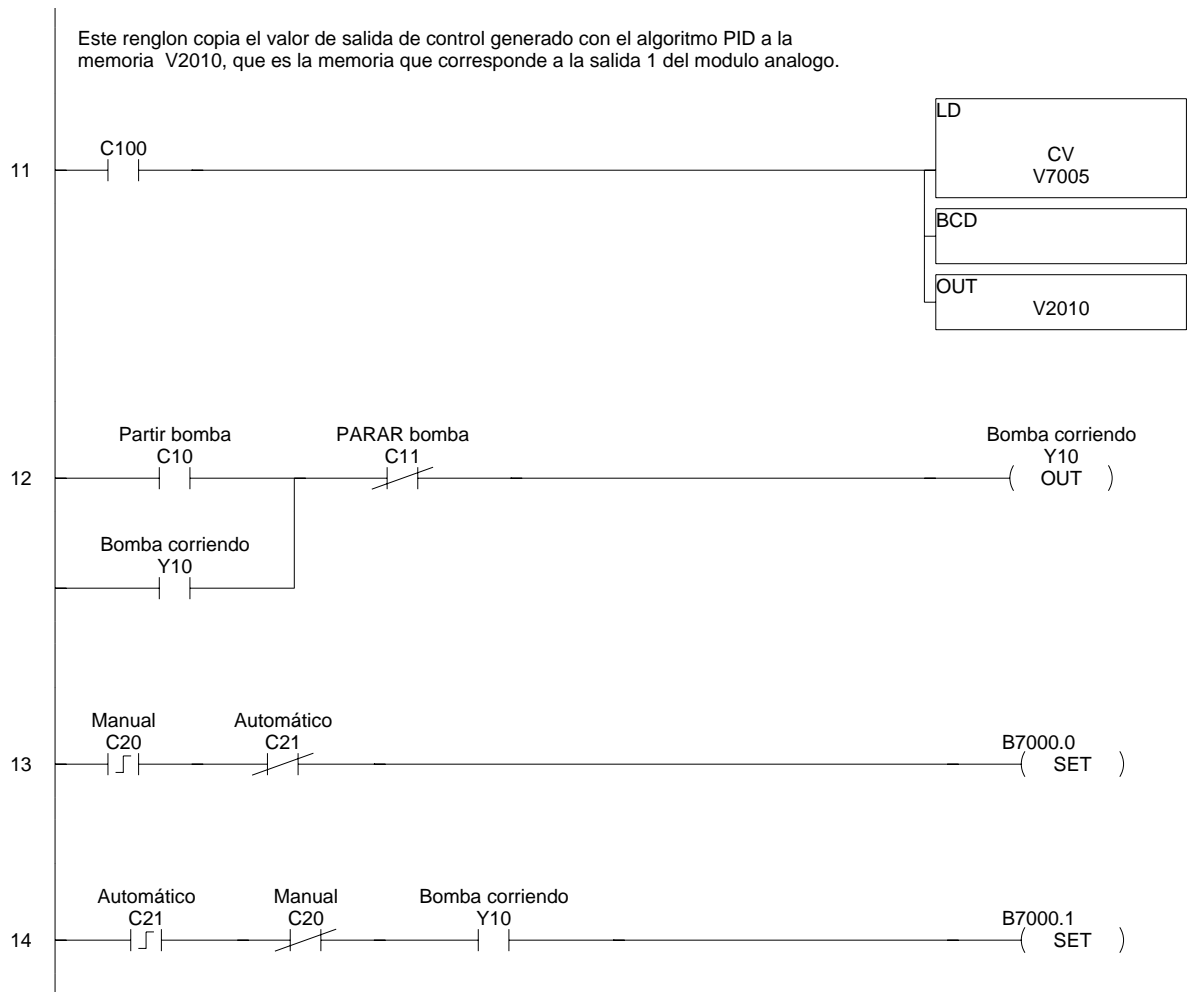
Otro asunto a considerar es el comando de partida y parada de la bomba, y también como comandar cuando el lazo de control va al modo Manual o al Automático. En Manual, el algoritmo que calcula la salida del lazo de control no es ejecutado. En Automático, la salida debe afectar el lazo de acuerdo a lo programado y hay una salida de control que depende del error (SP-PV) y de los valores de ganancia y de reset.

Usaremos botones en el panel de interface del operador para partir y parar la bomba, así como también para pasar de Manual a Automático. Se muestra en el diagrama ladder como se programa esta función. Vea a continuación el diagrama ladder correspondiente:









No se ha colocado la instrucción END porque el programa sigue posteriormente para mostrar el uso de la función ramp/soak.

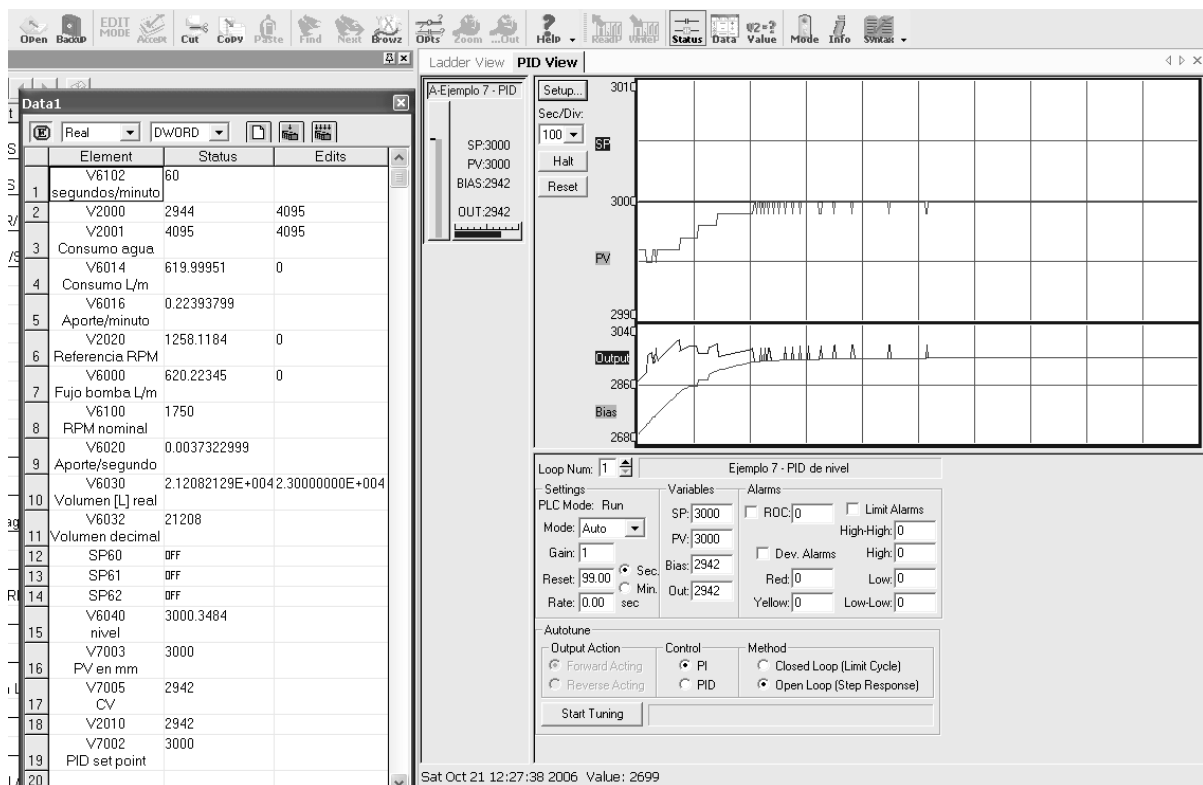
Note que hicimos, de propósito, cálculos sin usar Iboxes, para demostrar como se hace el cálculo con versiones de *DirectSOFT* anteriores a version 5. Naturalmente los cálculos pueden ser hechos con Iboxes. Se puede ver que el tiempo de barrido promedio es de 3 ms, con lo cual conseguimos el objetivo. Ésto no es importante en este caso, pero sirve para mostrar trucos para acortar el tiempo de barrido.

Teniendo el programa hecho, asumiremos que la salida del módulo de señales análogas y la entrada son equivalentes en RPM y por lo tanto conectaremos la salida 0-10 Volt a la entrada 0 -10 Volt. En la práctica, esto se conectará directamente al variador de frecuencia.

Sintonización (Tuning)

Corresponde hacer la sintonización (tuning); haremos el método de sintonización automática. Para eso colocaremos el valor de la variable de proceso (PV) lo mismo que el valor de referencia (SP), por ejemplo, en 3000 mm, usando los valores de ganancia y de reset que habíamos definido anteriormente. Usamos PID View y DataView para ayudar al proceso.

Colocamos en funcionamiento la bomba y cambiamos el modo desde **Manual** (que es el valor por defecto) a **Automático (Auto)**, con ayuda del panel de interface de operador, si es que está conectado, o con Data View, o aún con la ventana de PID View. Ud puede, en este caso, cambiar el modo directamente en el campo **Mode** haciendo clic en la flecha de menú al lado derecho del campo **Mode**.

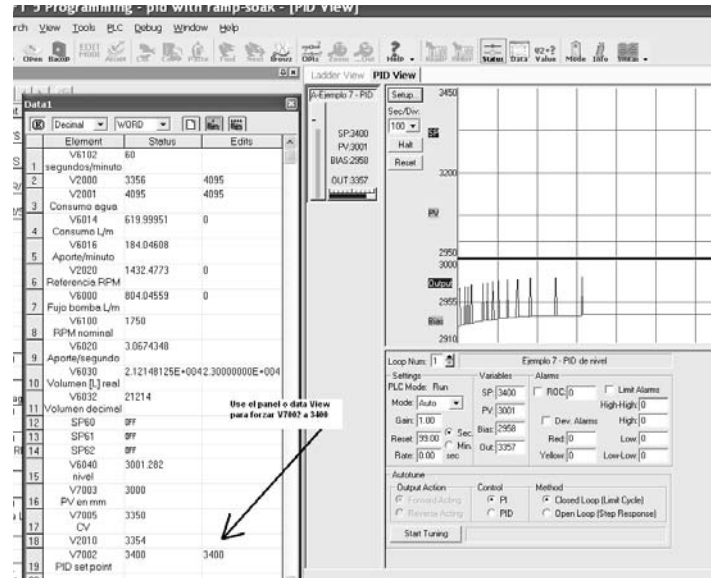


Como se ve en la figura de arriba, Data View puede colocar los valores que interesan para observar el proceso.

Con Data View se pueden forzar otros valores, se puede cambiar el estado de los bits C, etc.

El diálogo PID View permite visualizar la variable de proceso PV, el valor de referencia (SP), la salida de control CV y el bias, convenientemente en varios rangos de tiempo, hasta 200 segundos por división. También es posible cambiar valores y modos y es una herramienta indispensable para sintonización.

La próxima acción es activar el “autotuning” o sintonización automática; para eso usamos el campo **Autotune**; seleccionamos el método closed loop, el modo PI (no es necesario el efecto derivativo en este caso), colocamos una referencia más alta que la variable de proceso, por ejemplo, 3400 y se hace clic en el botón **Start Tuning** para que el PLC comience a determinar los valores de ganancia y de reset aplicando escalones de referencia. Esto toma cierto tiempo. En este caso tomó unos 5 a 8 minutos debido a que el actuador (la bomba) es relativamente pequeña. El diálogo de PID View avisa cuando al proceso acaba diciendo “Autotune complete, check PID parameters”.

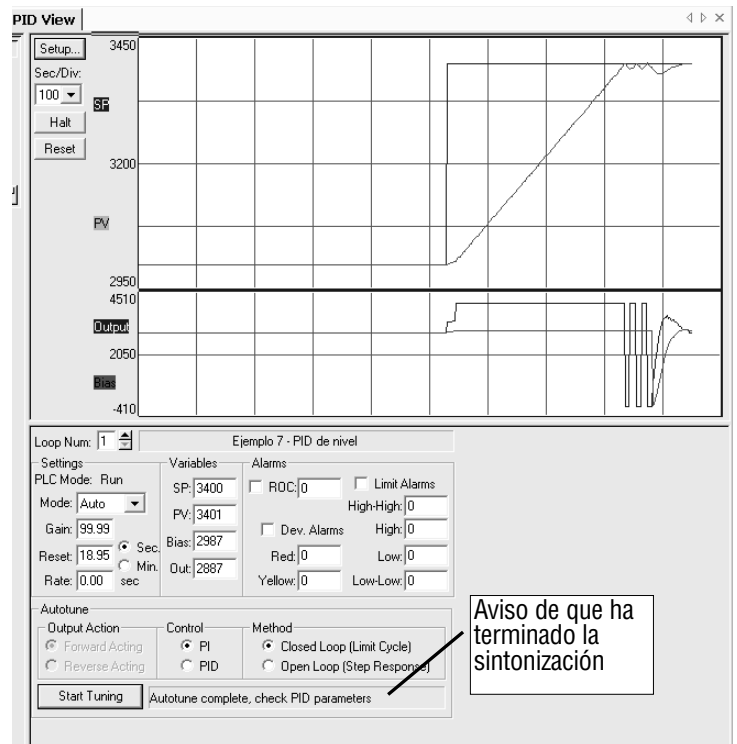


En general, la determinación de los valores de ganancia y reset son adecuados para la operación estable del control. Esta determinación es un valor calculado por el PLC considerando el atraso de la señal y la inclinación de la rampa producida por la variable de proceso al imponer referencias escalón por el PLC, durante este procedimiento.. Este valor no es el óptimo, sino es el que el PLC cree que es el mejor valor. Otras veces es necesario hacer una sintonización “manual”, para mejorar esos valores. El procedimiento para hacer esta operación está explicado en el capítulo 8 de este manual.

Observe las otras figuras que se muestran a continuación.

En esta figura se vé cuando el proceso de determinación de los valores ha terminado y aparece al lado del botón **Start Tuning** el mensaje que ha terminado la sintonización.

La próxima acción es verificar que el lazo de control trabaja en todo el rango. Por eso se han probado otros

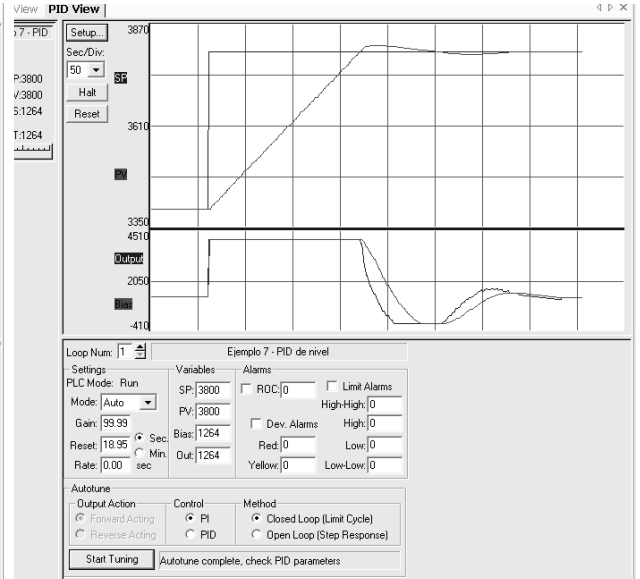
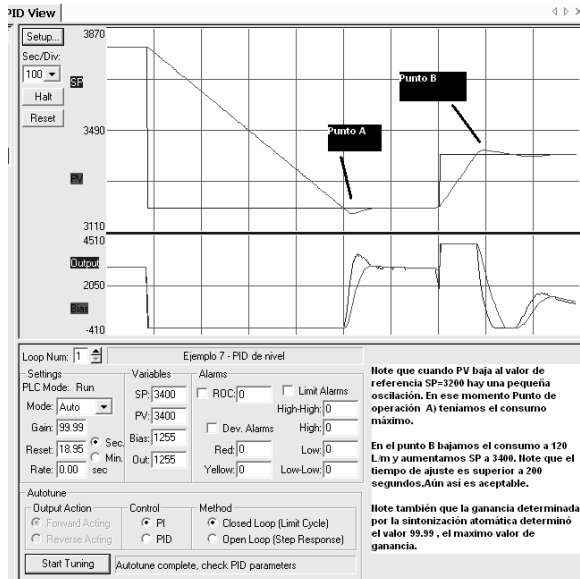
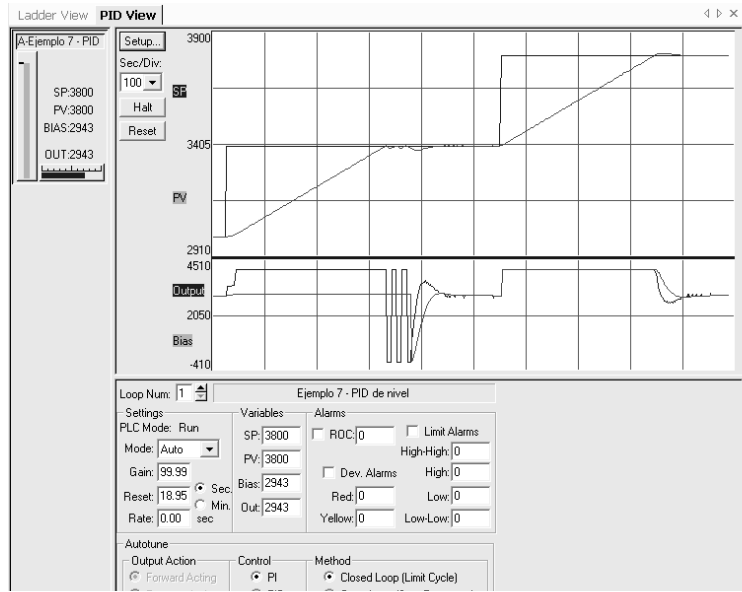


puntos de referencia y se ha cambiado el consumo de agua, del máximo al mínimo.

Se espera que el nivel del tanque se mantenga en aproximadamente 3800 mm.

Por eso se ha probado en el entorno de este valor con valores de 3200, 3400 y 3800 mm.

Las figuras muestran que el comportamiento es razonable. Aún existe “overshoot” y la variable de proceso se estabiliza en unos 200 segundos. En este caso no tiene

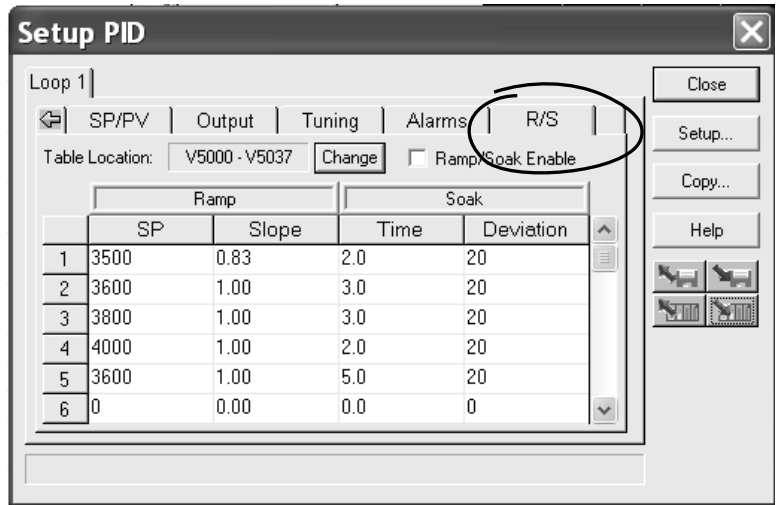


importancia, pero puede haber casos en que sea necesario no tener ningún “overshoot”. También, en otros casos, la respuesta debe ser más rápida y para eso, naturalmente, el actuador debe proporcionar mas corrección (la bomba debe tener mas capacidad de bombeo, es este caso).

El muestreo del control PID puede ser ajustado hasta 50 milisegundos, (lo que permite hacer que el PLC DL06 calcule la salida hasta 20 veces por segundo).

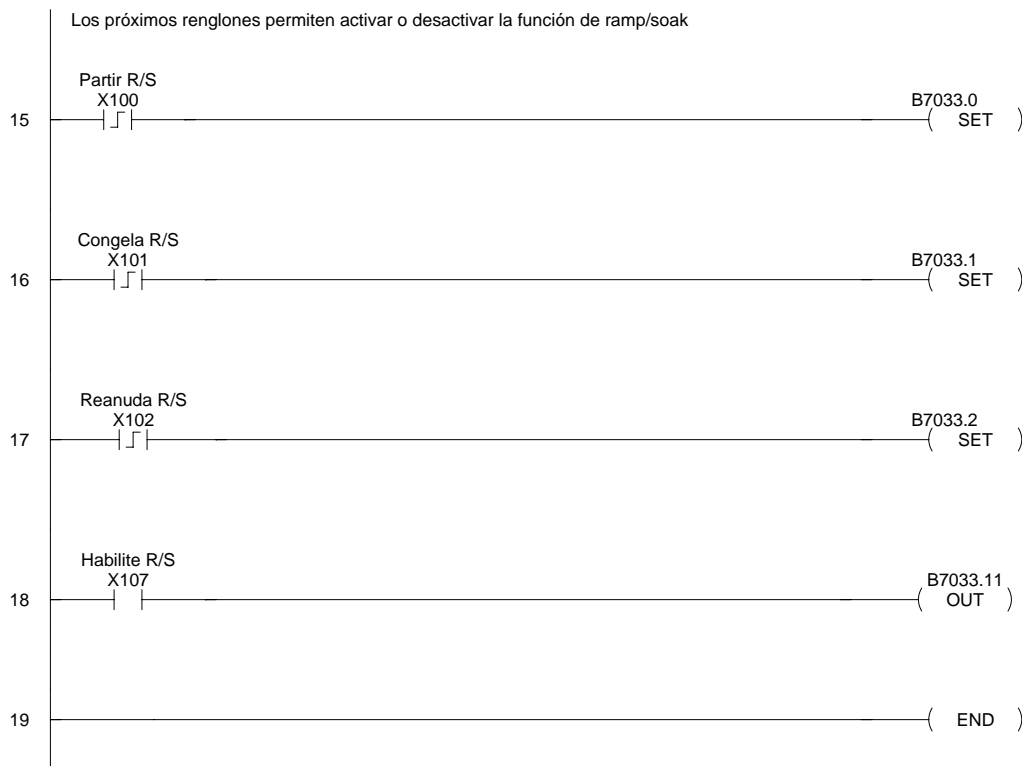
Uso de ramp/soak

Veamos ahora un ejemplo de uso de ramp/soak. recordemos que esta función es la programación de la referencia del lazo PID variable durante el tiempo. Para programar estos tiempos, se usa el diálogo de **Setup PID**, en la lengüeta de R/S. Vea la figura adyacente para las próximas explicaciones:



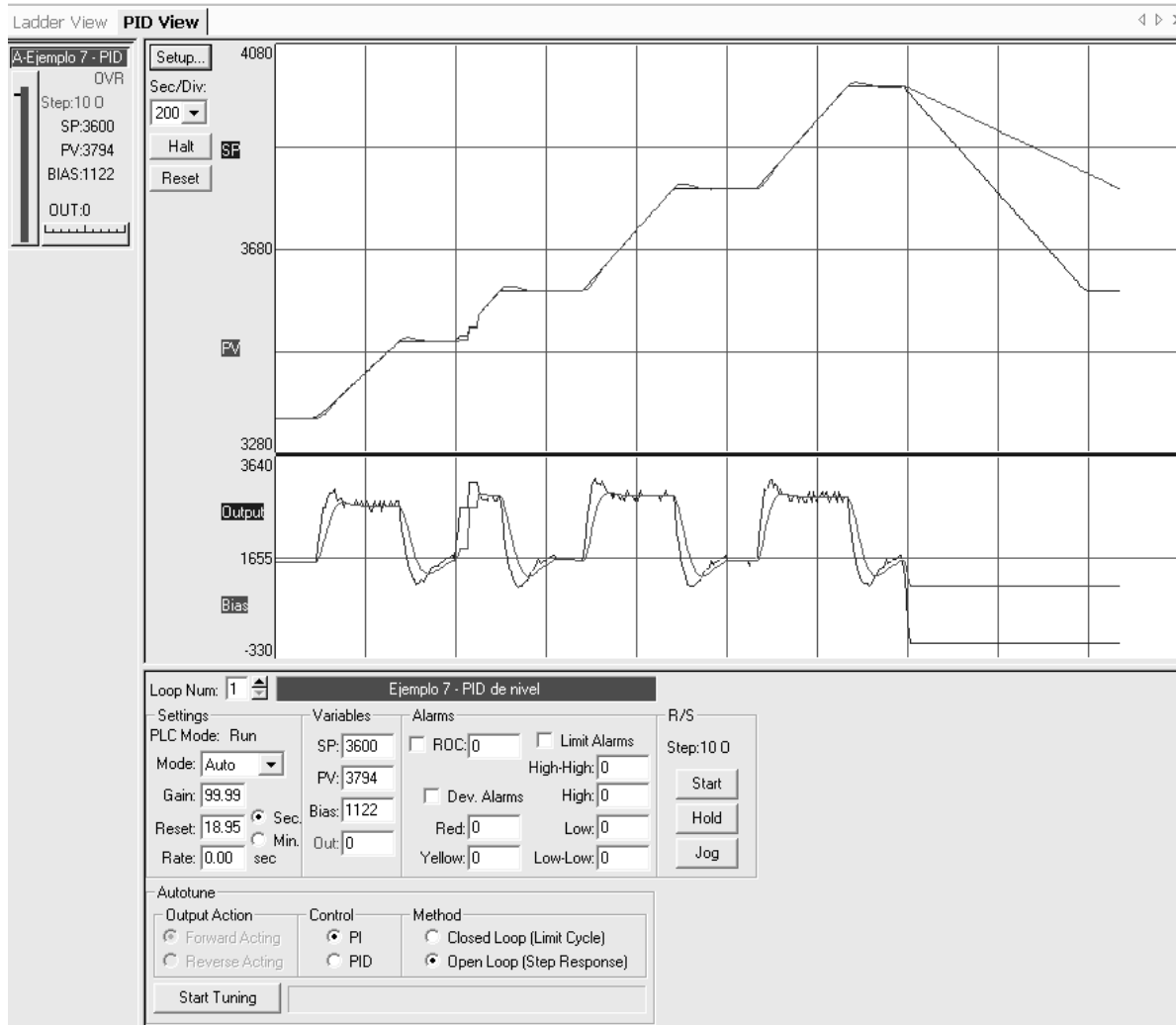
Consideraremos en la primera etapa que la primera rampa es de 50 segundos (0.83 minutos) yendo de donde se encuentre la variable de proceso hasta 3600 mm. Luego tendremos un valor fijo de 2 minutos (soak), Luego viene la segunda etapa, con 1 minuto de rampa y 3 minutos de soak en el valor 3800 . Y así sucesivamente.

Es necesario hacer código ladder para poder partir y parar esta función. Eso se hace de la forma mostrada en el diagrama siguiente:



Note que hemos usado las entradas X100 hasta X107. La razón de eso es que hemos usado el módulo D0-08SIM, que permite simular entradas. Naturalmente pueden ser usadas las entradas X0 hasta X23, u otras del tipo C0, que hayan sido consideradas con el panel de interface de operador.

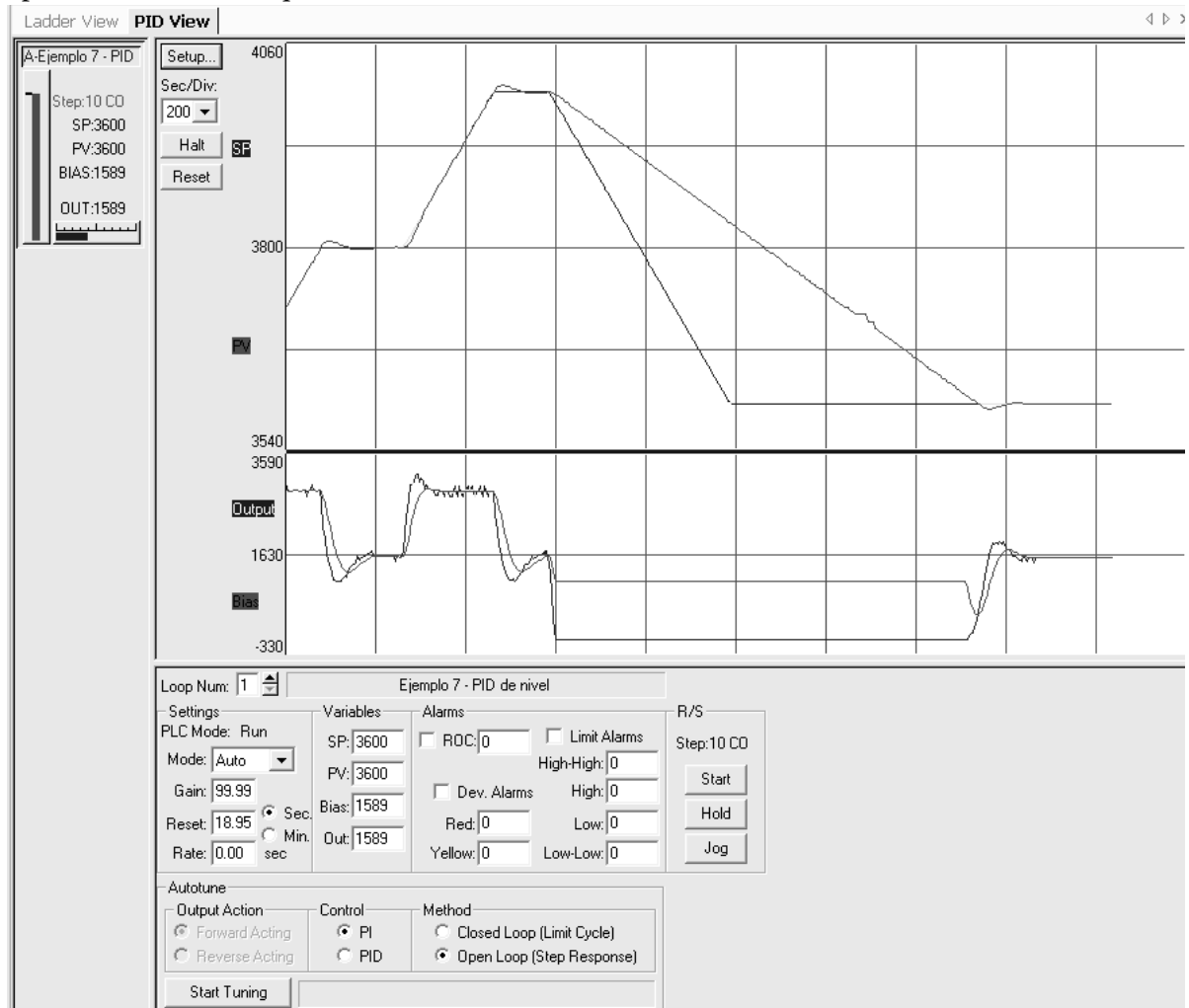
Vea en la próxima figura el resultado de esta función. Más explicaciones pueden ser encontradas en el capítulo 8.



Observaciones de la figura de arriba, con ramp/soak:

Note que la referencia sigue la programación de la variable de proceso. Inicialmente el valor de referencia estaba en 3350 mm. Cuando se activó la función ramp/soak la salida cambió para activar la bomba para dar mas flujo para que la variable de proceso siga la referencia. Eso es posible sin problemas en la subida. En la disminución del valor de referencia, el consumo, aunque sea el máximo, no permite que la variable de proceso siga la referencia, porque, aunque la salida se vaya a cero (bomba parada) el volumen del tanque no disminuye tan rápido como es deseado. Esta es una consideración importante en los lazos de control PID.

Note también, en la próxima figura, que cuando la variable de proceso llega al valor de referencia, la bomba comienza a girar a la velocidad tal que compensa el consumo, es decir, 1589 rpm equivalentes ($1859/4095 \times 1750$). Esta es una excelente solución, económica, para mantener el nivel del agua en el tanque a la altura deseada en función del tiempo. Este proceso de control PID también es una mucha mejor solución que hacer que la bomba parta y pare en función de 2 presostatos en el tanque.



Esto termina los ejemplos que han sido desarrollados en este capítulo. Esperamos que sean de utilidad para aprender a programar el PLC DL06. Depende de la inteligencia del programador hacer el programa de modo que haga las funciones deseadas, que se pueda entender, dependiendo de cuán bien sea documentado el diagrama. Existen muchas posibilidades de programación y un programa es válido cuando hace las funciones que se necesitan, con el mínimo de codificación, con un tiempo de barrido aceptable. Es muy conveniente que se pueda entender por otras personas. Por eso debe colocarse información en el diagrama. Note, por otra parte, que la documentación no se almacena en el PLC; solamente es almacenado el diagrama ladder en el PLC. Toda la documentación se almacena en la memoria del disco de la computadora.

Página dejada en blanco intencionalmente

FUNCIONES AUXILIARES



En este apéndice...

Introducción	A-2
AUX 2* — Operaciones de programación	A-4
AUX 3* — Operaciones de memoria V	A-4
AUX 4* — Configuración de E/S	A-4
AUX 5* — Configuración de la CPU	A-5
AUX 6* — Configuración del programador portátil	A-8
AUX 7* — Operaciones de EEPROM	A-8
AUX 8* — Operaciones de contraseña	A-9

Introducción

A

Propósito de las funciones auxiliares

Muchas tareas de configuración de la CPU implican el uso de funciones auxiliares(AUX.). Las funciones AUX realizan diversas operaciones, incluyendo limpieza de la memoria del programa, exhibición del tiempo de barrido y copia de programas a EEPROM en el programador portátil. Se dividen en categorías que afectan diversos recursos del sistema. Usted puede tener acceso a las funciones AUX de *DirectSOFT* o del programador D2-HPP. Los manuales de esos productos le proporcionan procedimientos paso a paso para tener acceso a las funciones AUX.. Algunos de estas funciones AUX. se han diseñado específicamente para la configuración del programador, así que no serán necesarias (o disponibles) en el paquete *DirectSOFT*. Aunque este apéndice proporciona muchos ejemplos de cómo funcionan, usted debe complementar esta información con la documentación del dispositivo de programación escogido.

NOTA: El programador portátil puede tener funciones AUX.adicionales que no existen con los PLCs DL06.

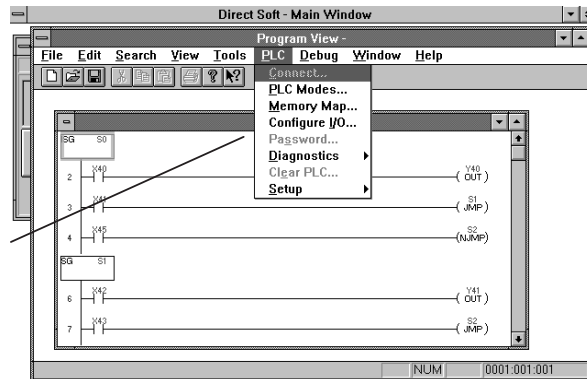
Funciones AUX y descripción		DL06
AUX 2* — Operaciones RLL		
21	Verifique el programa	*
22	Cambie referencia	*
23	Borre un rango de programa ladder	*
24	Borre todo el programa ladder	*
AUX 3* — Operaciones de memoria V		
31	Borre la memoria V	*
AUX 4* — Configuración de E/S		
41	Mestre la configuración de E/S	*
AUX 5* — Configuración de la CPU		
51	Modifique el nombre del programa	*
53	Muestre el tiempo de barrido	*
54	Initialize Scratchpad	*
55	Ajuste el tiempo de Watchdog	*
56	Configure el puerto 2	*
57	Configure rangos retentivos	*
58	Pruebe operaciones	*
59	Configuración de Override	*
5B	Configuración del a interfase HSIO	*
5D	Configuración de control de barrido	*

Funciones AUX y descripción		DL06
AUX 6* — Configuración de programador HPP		
61	Muestre los nos. de revisión	*
62	Active o desactive Beeper	HP
65	Haga funcionar diagnóstico	HP
AUX 7* — Operaciones EEPROM		
71	Copie memoria de la CPU a EEPROM de programador	HP
72	Escriba EEPROM a la CPU	HP
73	Compare la CPU al EEPROM del programador	HP
74	Verifique que el EEPROM está vacío.	HP
75	Apague el EEPROM del programador	HP
76	Muestre el tipo de EEPROM (CPU y HPP)	HP
AUX 8* — Operaciones de contraseña		
81	Modifique contraseña	*
82	Desbloquee la CPU	*
83	Bloquee la CPU	*

* - Soportada por el aparato
 HP - Función del programador portátil

Acceso a funciones AUX con *DirectSOFT*

DirectSOFT permite usar varias opciones de menú durante la programación en línea y fuera de línea. Algunas de las funciones AUX son solamente disponibles durante la programación en línea y otras solamente durante fuera de línea y algunas durante ambos casos. La figura siguiente muestra un ejemplo del menú de las operaciones del PLC disponible dentro de *DirectSOFT*.



Acceso a funciones AUX con el programador portátil

Se puede también tener acceso a las funciones AUX usando un programador portátil. Además, recuerde que algunas de las funciones AUX están solamente disponibles en el programador portátil. A veces el nombre o la descripción de la función AUX no puede caber en el visor. Si desea ver la descripción completa, apriete las teclas de flecha para moverse a la izquierda o a la derecha. También, dependiendo de la visión actual, puede tener que presionar CLR más de una vez.



AUX FUNCTION SELECTION
AUX 2* RLL OPERATIONS

Utilice NXT o PREV para moverse en secuencia a través de los menús



AUX FUNCTION SELECTION
AUX 3* V OPERATIONS

Presione ENT para seleccionar sub-menús



AUX 3* V OPERATIONS
AUX 31 CLR V MEMORY

Usted puede también entrar el número AUX. exacto para ir directo al sub-menu.

Entre el número AUX. directamente



AUX 3* V OPERATIONS
AUX 31 CLR V MEMORY

AUX 2* — Operaciones de programa ladder RLL

Las funciones auxiliares de operación RLL le permiten realizar varias operaciones en el programa ladder.

AUX 21 Check Program

El programador y *DirectSOFT* comprueban automáticamente si hay errores durante la entrada del programa. Sin embargo, puede haber ocasiones cuando usted desea comprobar un programa que ha estado ya en la CPU. Hay dos tipos de verificaciones disponible:

- Sintaxis
- Referencias duplicadas

La verificación sintáctica encontrará una variedad amplia de errores de programación, tales como falta de la instrucción END. Si usted realiza esta verificación y obtiene un error, vea el apéndice B para un listado completo de los códigos de error de programación. Corrija el problema y después continúe ejecutando la verificación sintáctica hasta que aparezca el mensaje “NO SYNTAX ERROR”.

Use la verificación de referencias duplicadas para verificar que usted no ha utilizado la misma referencia de bobina de salida más de una vez. Note, testa función AUX también encontrará las mismas salidas incluso si se han utilizado instrucciones OROUT, lo que es perfectamente aceptable.

Esta función AUX está disponible en el sub-menu PLC>Diagnostics dentro de *DirectSOFT*.

AUX 22 Change Reference

Habrà probablemente un momento en que usted necesite cambiar una referencia de dirección de E/S o de relevadores de control. AUX 22 permite cambiar rápidamente y fácilmente todas las ocurrencias, (dentro de un rango de direcciones) de una instrucción específica. Por ejemplo, usted puede substituir cada instancia de X5 por X10.

AUX 23 Clear Ladder Range

Sucede mucho para solucionar nuevos problemas de aplicación que se toman programas existentes y se agregan o se quitan ciertas partes. Usando AUX 23 usted puede seleccionar y suprimir una parte del programa. *DirectSOFT* no tiene un menú para esta función AUX, pero usted puede apenas seleccionar la parte adecuada del programa y modificarlo con las herramientas de modificación.

AUX 24 Clear Ladders

AUX 24 borra el programa entero de la memoria de la CPU. Antes de que usted entre un nuevo programa, usted debe tener siempre la memoria ladder limpia. Esta función AUX está disponible en el sub-menu del PLC> Clear dentro de *DirectSOFT*.

AUX 3* — Operaciones de memoria V

AUX 31 Clear V Memory

AUX 31 borra toda la información de las direcciones de memoria V disponibles para uso general. Esta función AUX está disponible en el sub-menu del PLC PLC>Clear PLC dentro de *DirectSOFT*.

AUX 4* — Configuración de entradas y salidas

AUX 41 Show I/O Configuration

Esta función AUX le permite mostrar la configuración corriente en el DL06. Tanto el programador como *DirectSOFT*, mostrarán la configuración de entradas y salidas.

AUX 5* — Configuración de la CPU

Las funciones AUX auxiliares siguientes permiten que usted configure, que vea o que cambie la configuración de la CPU.

AUX 51 Modifique el nombre del programa

Los PLCs DL06 pueden usar un nombre de programa para el programa en la CPU o un programa almacenado en EEPROM en el programador. (note, no puede tener programas múltiples almacenados en EEPROM). El nombre del programa puede tener hasta 8 caracteres en longitud y puede utilizar cualquiera de los caracteres disponibles (A-z, 0-9). AUX 51 le permite entrar un nombre de programa. Puede también realizar esta operación dentro de *DirectSOFT* usando el sub-menú **PLC>Setup**. Cuando haya entrado un nombre de programa, puede solamente limpiar el nombre usando AUX 54 para renombrar la memoria del sistema. Asegúrese que entiende los efectos posibles de AUX 54 antes de que usted la utilice!

AUX 53 Tiempo de barrido del PLC

AUX 53 muestra el tiempo actual, mínimo y máximo del barrido. Los tiempos mínimo y máximo son los que han ocurrido desde la última transición de modo Program al modo RUN. Usted puede también realizar esta operación dentro de *DirectSOFT* usando el sub-menú **PLC>Diagnostics**.

AUX 54 Initialize Scratchpad

La CPU mantiene parámetros del sistema en un área de memoria designada a menudo el "scratchpad". En algunos casos, usted puede realizar cambios a la configuración del sistema que será almacenado en la memoria. Por ejemplo, si usted especifica un rango de relevadores de control (CRs) como retentivos, se almacenan estos cambios en esta memoria.



NOTE: Puede ser que nunca tenga que usar esta característica a menos que haya realizado cambios que afectan la memoria del sistema.

AUX 54 coloca la memoria de sistema a los valores originales de fábrica. Usted puede también realizar este operación dentro de *DirectSOFT* usando el sub-menú **PLC>Setup**.

AUX 55 Configura el Watchdog Timer

Los PLCs DL06 tienen un contador de tiempo de "perro guardián" que se utiliza para supervisar el tiempo del barrido. El valor original de fábrica es 200 ms. Si el tiempo de barrido excede el límite de tiempo del perro guardián, la CPU deja automáticamente el modo RUN y entra en modo Program. El programador muestra el siguiente mensaje E003 S/W TIMEOUT cuando ocurre un barrido de un tiempo mayor que el watchdog.

Use AUX 55 para aumentar o para disminuir el valor del contador de tiempo del perro guardián. Usted puede también realizar esta operación dentro de *DirectSOFT* usando el sub-menú **PLC>Setup**.

AUX 56 CPU Network Address

Puesto que la CPU DL06 tiene 2 puertos, usted puede usar el programador para configurar la dirección de red para el puerto 2 y los parámetros de comunicación. Los ajustes por defecto son:

- Dirección de la estación: 1
- Modo Hexadecimal
- Paridad Odd.

Se puede utilizar este puerto con o el programador, *DirectSOFT*, o como puerto para *DirectNET* y MODBUS.



NOTA: Usted necesitará solamente usar este procedimiento si usted tiene el puerto 2 conectado con una red. Si no, los valores por defecto trabajarán muy bien.

Use AUX 56 para configurar los parámetros de dirección y de comunicación de la red. Se puede también realizar esta operación dentro de *DirectSOFT* usando **PLC>Setup**.

AUX 57 Set Retentive Ranges

Los PLCs DL06 proporcionan ciertos rangos de memoria retentiva por defecto. Algunos de las direcciones retentivas de memoria son respaldadas por un condensador y otros están en memoria FLASH permanente. Las direcciones de la memoria FLASH son V7400 a V7577. Los rangos por defecto son convenientes para muchas aplicaciones, pero usted puede cambiarlos si su aplicación requiere rangos retentivos adicionales o no necesita retentividad. Los ajustes por defecto son:

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

Use AUX 57 para cambiar los rangos retentivos. Usted puede también realizar esta operación dentro de *DirectSOFT* usando el sub-menú **PLC>Setup**.



ADVERTENCIA: Los PLCs DL06 no tienen RAM con batería. Un condensador de gran capacidad mantendrá los valores en el evento de falta de energía, pero solamente hasta 3 semanas (El tiempo de retención de la memoria puede ser tan corto como 4 días en una temperatura de funcionamiento de 60 grados C).

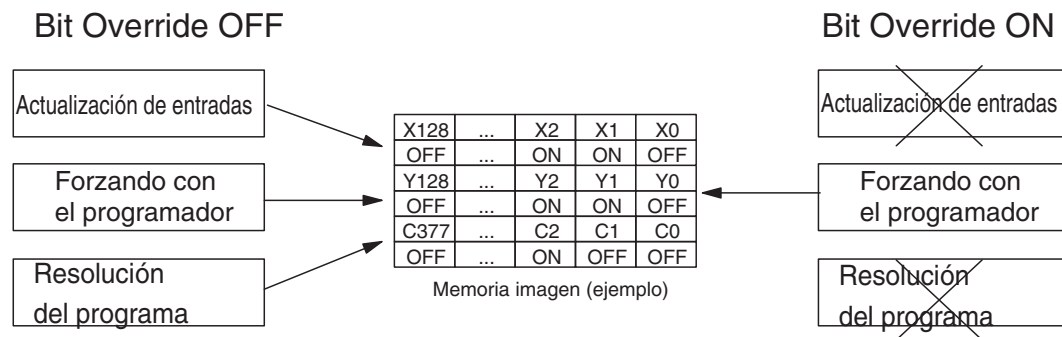
AUX 58 Test Operations

AUX 58 se usa para eliminar la inhabilitación de la función de la instrucción PAUSE. Use AUX 58 para programar una sola salida o un rango de salidas que funcionarán normalmente cuando esos puntos están dentro del alcance de la instrucción PAUSE.

AUX 59 Bit Override

El bit override puede ser activado en punto a punto usando AUX 59 del programador o por una opción del menú dentro de *DirectSOFT*. El bit override inhabilita básicamente cualquier cambio de estado a un punto discreto por la CPU. Por ejemplo, si usted activa el bit override para X1 y X1 está apagada en ese entonces, entonces la CPU no cambiará el estado de X1. Esto significa que incluso si X1 se hace ON, la CPU no reconocerá el cambio. De modo que, si usted utilizara X1 en el programa, sería evaluada siempre como "OFF" en este caso. Por supuesto, si X1 estuviera encendida cuando el bit override fue activada, entonces X1 sería evaluado siempre como "ON".

Hay una ventaja cuando usted utiliza la función de bit override. El forzar regular no es desactivado porque se activa el bit override. Por ejemplo, si usted permitiera el bit override para Y0 y estuviera apagado 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 usa el dispositivo de programación para forzar la salida Y0 encendida, permanecerá encendida y la CPU no cambiará el estado de Y0. Si usted luego fuerza Y0 a apagada, la CPU mantendrá Y0 como apagada. La CPU nunca actualizará el punto con los resultados del programa de aplicación o de la actualización de E/S hasta que el bit override se sesactive en el punto. El diagrama siguiente muestra una breve descripción de la función de bit override. Note que la CPU no actualiza la memoria imagen cuando se activa el bit override.



AUX 5B Configuración HSIO

Se usa AUX 5B con la función de E/S de alta velocidad (HSIO) para seleccionar la configuración. Usted puede elegir el tipo de contador, configurar los parámetros, etc. Vea el capítulo 3 para una descripción completa de cómo seleccionar las varias características del contador.

AUX 5D Seleccione el modo de barrido del PLC

El PLC DL06 tiene dos modos de barrido del programa: fijo y variable.

En modo fijo, el tiempo de barrido se agranda al tiempo que usted especifica (en milisegundos). Si el tiempo real de barrido es más grande que el tiempo fijo de barrido, después se muestra el código de error 'E504 BAD REF/VAL'. En modo de barrido variable, la CPU comienza cada barrido cuando las actividades del barrido anterior terminan.

AUX 6* — Configuración del programador D2-HPP

Las funciones auxiliares siguientes permiten que usted configure, que vea o que cambie la configuración del programador D2-HPP.

AUX 61 Muestre el no. de versión

Como con la mayoría de los productos industriales de control, hay casos cuando se agregan características y mejoras. Estas nuevas características trabajan a veces solamente con ciertos lanzamientos de firmware. Usando AUX 61 usted puede ver rápidamente los números de revisión de los firmwares de la CPU y del programador. Esta información (para la CPU) está también disponible dentro de *DirectSOFT* del sub-menu de PLC>Diagnostics.

AUX 62 Zumbador activado o desactivado

El programador tiene un zumbador (beeper) que da confirmación de que la tecla fue apretada. Usted puede utilizar la función auxiliar AUX 62 para prender o apagar al beeper.

AUX 65 Diagnósticos

Si se piensa que el programador no está funcionando correctamente, puede usar AUX 65 para probar un programa de diagnóstico. Puede verificar los siguiente:

- Teclado.
- Visor.
- Indicadores LED y contraluz.
- Verificación del EEPROM del programador.

AUX 7* — Operaciones con EEPROM

Las funciones auxiliares siguientes permiten que usted mueva el programa ladder de una área a otra y que realice otras tareas de mantención del programa.

Áreas de memoria transferibles

Muchas de estas funciones AUX permiten que usted copie diversas áreas de la memoria a y desde la CPU y el programador. La tabla siguiente muestra las áreas que pueden ser copiadas.

Opción y tipo de memoria	Rango por defecto del DL06
1:PGM — Programa	\$00000 – \$02047
2:V — Memoria V	\$00000 – \$07777
3:SYS — Sistema	No es seleccionable. Copia parámetros del sistema
4:etc (All)— Programa, sistema y memoria V no volátil solamente	No es seleccionable

AUX 71 CPU al EEPROM del programador

AUX71 copia información de la memoria de la CPU a un EEPROM instalado en el programador. El programador puede copiar diversas porciones de memoria EEPROM (HP) a la memoria de la CPU según lo mostrado en la tabla anterior.

AUX 72 EEPROM del programador a la CPU

AUX 72 copia información del EEPROM instalado en el programador a la memoria de la CPU en el DL06. Se pueden copiar diversas porciones de la memoria EEPROM (HP) a la memoria de CPU según lo mostrado en la tabla anterior.

AUX 73 Compare EEPROM a la CPU

AUX 73 compara el programa en el programador (EEPROM) con el programa de la CPU.¶ Usted puede comparar diversos tipos de información según lo mostrado previamente.

AUX 74 Verifique que el EEPROM está vacío

AUX 74 permite que usted compruebe el EEPROM en el programador para cerciorarse de esté vacío. Es una buena idea usar esta función siempre que usted comience a copiar un programa entero a un EEPROM en el programador.

AUX 75 Borre el EEPROM

AUX 75 permite borrar todos los datos en el EEPROM localizado en el programador. Se debe utilizar esta función AUX antes de copiar un programa desde la CPU.

AUX 76 Muestre el tipo de EEPROM

Usted puede utilizar AUX 76 para determinar rápidamente qué tamaño de EEPROM está instalado en el programador.

AUX 8* — Operaciones de contraseña

Hay varias funciones AUX disponibles que se pueden usar para modificar o para activar la contraseña de la CPU. Usted puede usar estas funciones durante comunicaciones en línea con la CPU o puede también utilizarlos con un EEPROM instalado en el programador durante operación off-line. Esto le permitirá desarrollar un programa en el programador y que incluya la protección de contraseña.

- AUX 81 — Modificar la contraseña.
- AUX 82 — Desbloquee la CPU
- AUX 83 — Bloquee la CPU

AUX 81 Modique la contraseña

Usted puede utilizar AUX 81 para permitir una medida adicional de protección entrando una contraseña que prevenga operaciones desautorizadas de la máquina. La contraseña debe ser un código numérico de ocho dígitos (0-9). Una vez que usted haya entrado una contraseña, usted puede sacarla entrando ceros (00000000). (éste es el valor original de la fábrica).

Una vez que usted haya entrado una contraseña, se puede trabar la CPU contra el acceso. Hay dos maneras de trabar o bloquear la CPU con el programador.

- La CPU estará siempre bloqueada después de apagar el PLC y encenderlo nuevamente (si una contraseña está presente).
- Usted puede utilizar AUX 82 y AUX 83 para trabar y para desbloquear la CPU.

Usted puede también entrar o modificar una contraseña dentro de *DirectSOFT* usando el sub-menu de **PLC>Password**. Esta característica trabaja levemente diferentemente en *DirectSOFT*. Una vez que usted haya entrado una contraseña, la CPU se traba automáticamente cuando usted sale del software. También será bloqueada si el PLC es apagado.



ADVERTENCIA: Asegúrese de tomar nota de la contraseña antes de que trabe la CPU. Una vez que la CPU esté bloqueada no puede ver, cambiar o borrar la contraseña. Si usted no recuerda la contraseña, usted tiene que enviar el CPU a **AUTOMATIONDIRECT** para retirar la contraseña. En ese caso, el programa también se borraré.



NOTA: Hay una protección del programa ladder en dos niveles con la contraseña del PLC DL06. Esto permite que haya protección de contraseña pero que no trabe el puerto de comunicación a una interfase de operador. La contraseña para este caso puede ser creada usando una letra mayúscula "A" seguido por siete caracteres numéricos (e.g. A1234567).

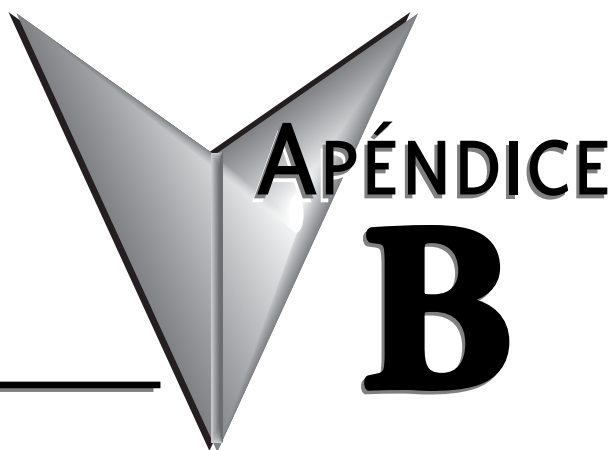
AUX 82 Destrahe la CPU

AUX 82 se puede utilizar para destrabar una CPU que ha sido protegida con una contraseña. *DirectSOFT* pedirá automáticamente que usted incorpore la contraseña si usted trata de comunicarse con una CPU que contenga una contraseña.

AUX 83 Trabe la CPU

AUX 83 se puede utilizar para trabar una CPU que ha sido protegida con una contraseña. Una vez que la CPU sea bloqueada, usted tendrá que entrar una contraseña para acceder. Recuerde, esto no es necesario con *DirectSOFT*, puesto que la CPU se traba automáticamente siempre que usted salga del paquete de software.

CÓDIGOS DE ERROR DEL PLC DL06



En este apéndice...

Códigos de error del PLC DL06B-2

Códigos de error del PLC DL06

B

Código de error del DL06	Descripción
E001 ERROR FATAL DE LA CPU	Se puede posiblemente borrar el error apagando y preñdiendo el PLC. Si el error vuelve, substituya el PLC DL06.
E003 TIME-OUT TIME-OUT	Este error ocurrirá si el tiempo de barrido del programa excede el tiempo asignado al temporizador del watchdog. SP51 estará encendido y el código de error estará almacenado en V7755. Para corregir este problema use AUX 55 para hacer mas grande el tiempo asignado al temporizador de watchdog.
E041 VOLTAJE DE BATERÍA DEL PLC BAJO	El voltaje de la batería del PLC DL06 está bajo (menos de 2.5 Volt) y se debe substituir. SP43 estará encendido y el código de error estará almacenado en V7757.
E104 FALLO UNA ESCRITURA	No fue completada una escritura al PLC DL06 .Apague y encienda el. Si el error vuelve, substituya el PLC DL06.
E151 COMMANDO INCORRECTO	Un error de paridad ha ocurrido en el programa en uso. SP44 estará encendido y el código de error será almacenado en V7755. Este problema puede posiblemente ser debido al ruido eléctrico. Borre la memoria y baje el programa otra vez. Corrija cualquier problema de aterramiento. Si el error vuelve, substituya el PLC DL06.
E155 FALLA DE RAM	Ha ocurrido un error en la comprobación(CHECKSUM) en el sistema RAM. SP44 estará encendido y el código de error será almacenado en V7755. Este problema puede ser debido a una batería con voltaje bajo, a ruido eléctrico o a una falla en la RAM de la CPU. Borre la memoria y baje el programa otra vez. Corrija cualquier problema de aterramiento. Si el error vuelve, substituya el PLC DL06.
E2** FALLA DE UN MODULO DE E/S	Un módulo de E/S ha fallado. Ejecute AUX42 para determinar el error real.
E202 FALTA UN MODULO DE E/S	Un módulo de E/S no ha podido comunicarse con el DL06 o falta en la ranura. SP45 estará encendido y el código de error será almacenado en V7756. Ejecute AUX42 para determinar la localización de la ranura y de la base del módulo que genera el error.
E210 FALLA EN LA ALIMENTACION	Una falta de alimentación de corta duración ocurrió en la línea que proveía potencia al DL06
E252 CONFIGURACIÓN DE E/S NUEVA	Este error ocurre cuando la verificación automática de la configuración está activada el DL06 y la configuración real de E/S ha cambiado moviendo los módulos en una base o cambiando tipos de módulos en una base. Se pueden volver los módulos a la posición original o ejecutar AUX45 para aceptar la nueva configuración. SP47 estará encendido y el código de error estará almacenado en V7755
E262 DIRECCION DE E/S FUERA DEL RANGO	Se ha encontrado una dirección de E/S fuera del rango en el programa en uso. Corrija la dirección inválida en el programa. SP45 estará encendido y el código de error estará almacenado en V7755
E263 DIRECCION DE E/S CONFIGURADA ESTA FUERA DEL RANGO	Se han asignado direcciones fuera de rango mientras se hace configuración de E/S manual..Corrija las asignaciones de dirección usando AUX46.
E311 ERROR 1 DEL PROGRAMADOR	Una petición del programador no pudo ser procesada por el DL06. Limpie el error y revise la petición. Si el error continúa substituya el DL06. SP46 está encendido y el código de error estará almacenado en V7756.
E312 ERROR 2 DE COMUNICACIÓN CON EL PROGRAMADOR	Un error de datos fue encontrado durante comunicaciones con el DL06. Limpie entre los dos dispositivos, substituya el programador i si es necesario substituya el DL06. El código de error estará almacenado en V7756.
E313 ERROR 3 DE COMUNICACION CON EL PROGRAMADOR	Un error de dirección fue encontrado durante comunicaciones con el DL06. Limpie el error y revise la petición. Si el error continúa verifique el cable entre los dos dispositivos, substituya el programador, después si hay necesidad substituya el DL06. El código de error será almacenado en V7756.
E316 HP COMM ERROR 6	Un error de modo fue encontrado durante comunicaciones con el DL06. Limpie el error y revise la petición. Si el error continúa substituya el programador, después en caso de necesidad substituya el DL06. El código de error estará almacenado en V7756.

Códigos de error del PLC DL06, continuado

Código de error del DL06	Descripción
E320 HP COMM TIME-OUT	El DL06 no respondió a la petición de comunicación del programador. Compruebe que el cable está correcto y no defectuoso. Apague y prenda el sistema. Si el error continúa, primero substituya el DL06 y después el programador en caso de necesidad.
E321 ERROR DE COMUNICACION	Un error de datos fue encontrado durante la comunicación con el DL06. Asegúrese que el cable es correcto y no defectuoso. Apague y prenda el sistema y si el error continúa primero substituya el DL06 y después el programador si fuera necesario.
E4** NO HAY UN PROGRAMA	Existe un error de sintaxis en el programa en uso. El más común es que falta la instrucción END. Ejecute AUX21 para determinar cual serie de errores se está señalando por medio de un relevador especial. SP52 estará encendido y el código de error estará almacenado en V7755.
E401 FALTA LA INSTRUCCION END	Todos los programas de uso deben terminar con una declaración END. Entre la instrucción END en la dirección apropiada en su programa. SP52 estará encendido y el código de error estará almacenado en V7755.
E402 FALTA LBL	Una instrucción MOVMC o LDLBL fue utilizada sin la etiqueta apropiada. Vea el capítulo 5 para los detalles en estas instrucciones. SP52 estará encendido y el código de error estará almacenado en V7755..
E403 FALTA RT	Una subrutina en el programa no termina con la instrucción RT. SP52 estará encendido y el código de error estará almacenado en V7755..
E404 FALTA FOR	Una instrucción NEXT no tiene la instrucción correspondiente FOR. SP52 estará encendido y el código de error estará almacenado en V7755.
E405 FALTA NEXT	La instrucción FOR no tiene la instrucción correspondiente NEXT. SP52 estará encendido y el código de error estará almacenado en V7755.
E406 FALTA IRT	Una rutina de interrupción en el programa no termina con la instrucción IRT. SP52 estará encendido y el código de error será almacenado en V7755.
E412 SBR/LBL>256	Hay más de 256 instrucciones SBR o DLBL en el programa. Este error también aparece si hay más de 4 instrucciones INT usadas en el programa. SP52 estará encendido y el código de error estará almacenado en V7755.
E421 REFERENCIA DUPLICADA DE ETAPAS	Dos o más etiquetas SG o ISG existen en el programa en uso con el mismo número. Un número único se debe permitir para cada etapa y la etapa inicial. SP52 estará encendido y el código de error estará almacenado en V7755.
E422 DUPLICATE LBL REFERENCE	Existen dos o más instrucciones LBL en el programa en uso con el mismo número. Un número único se debe permitir para cada uno y la etiqueta. SP52 estará encendido y el código de error será almacenado en V7755.
E423 LAZO NEXT/FOR EN OTRO LAZO	No se permiten lazos que programan un lazo de FOR/NEXT dentro de otro .SP52 estará encendido y el código de error estará almacenado en V7755.
E431 DIRECCION INVALIDA ISG/SG	No se debe poner una instrucción de ISG o SG después de la declaración END (por ejemplo dentro de una subrutina). SP52 estará encendido y el código de error estará almacenado en V7755.
E432 DIRECCION INVALIDA DE SALTO (GOTO)	Una instrucción LBL que corresponde a una instrucción GOTO no se debe programar después de la instrucción END tal como adentro una subrutina. SP52 estará encendido y el código de error estará almacenado en V7755.
E433 DIRECCIÓN INVÁLIDA DE SUBROUTINA	Una instrucción SBR se debe programar después de la declaración END, no en el cuerpo principal del programa o en una rutina de interrupción. SP52 estará encendido y el código de error estará almacenado en V7755.
E434 DIRECCION INVÁLIDA RTC	Una instrucción RTC se debe programar después de la declaración END, no en el cuerpo principal del programa o en una rutina de interrupción. SP52 estará encendido y el código de error estará almacenado en V7755.

Apéndice B: Códigos de error

B

Código de error del DL06	Descripción
E435 DIRECCION INVÁLIDA RT	Una instrucción RT se debe programar después de la declaración END, no en el cuerpo principal del programa o en una rutina de interrupción. SP52 estará encendido y el código de error estará almacenado en V7755
E436 DIRECCION INVÁLIDA INT	Una instrucción INT se debe programar después de la declaración END, no en el cuerpo principal del programa. SP52 estará encendido y el código de error estará almacenado en V7755.
E437 DIRECCION INVÁLIDA IRTC	Una instrucción IRTC se debe programar después de la declaración END, no en el cuerpo principal del programa o en una rutina de interrupción. SP52 estará encendido y el código de error estará almacenado en V7755.
E438 DIRECCION INVÁLIDA IRT	Una instrucción IRT se debe programar después de la declaración END, no en el cuerpo principal del programa o en una rutina de interrupción. SP52 estará encendido y el código de error estará almacenado en V7755.
E440 DIRECCION INVÁLIDA DE DATOS	O la instrucción DLBL se ha programado en el área principal del programa (no después de la declaración END) o la instrucción DLBL está en un renglón que contiene el contactos de entradas.
E441 ACON/NCON	Una instrucción ACON o un NCON se debe programar después de la declaración END, no en el cuerpo principal del programa. SP52 estará encendido y el código de error estará almacenado en V7755.
E451 MLS/MLR CON ERROR	Las instrucciones MLS se deben numerar en orden ascendente de arriba para abajo.
E452 X USADA COMO SALIDA	Un tipo de datos X se está utilizando como salida de bobina.
E453 FALTA UN TEMPORIZADOR O CONTADOR	Se está utilizando un contacto de un temporizador o de un contador donde no existe el elemento asociado.
E454 TMRA CON ERROR	Falta uno de los contactos de una instrucción MRA.
E455 CONTACTO DE CNT/UDC FALTANDO	Falta uno de los contactos de una instrucción CNT o UDC..
E456 FALTANDO CONTACTO DE SR	Falta uno de los contactos de una instrucción SR..
E461 STACK OVERFLOW	Se han almacenado en el stack más de nueve niveles de lógica . Compruebe el uso de las instrucciones OR STR y AND STR .
E462 STACK UNDERFLOW	Se ha almacenado en el stack un número no aceptable de los niveles de lógica .Asegúrese que el número de instrucciones AND STR y OR STR corresponde al número de instrucciones STR.
E463 ERROR LOGICO	No fue utilizada una instrucción STR/STRN para comenzar un renglón la lógica ladder.
E464 CIRCUITO FALTANDO EN RENGLÓN	No se ha terminado correctamente un renglón de lógica ladder.
E471 REFERENCIA DUPLICADA DE BOBINAS DE SALIDA	Dos o más instrucciones OUT se refieren al mismo punto de E/S.
E472 REFERENCIA DUPLICADA DE UN TEMPORIZADOR	Dos o más instrucciones TMR se refieren al mismo número.
E473 REFERENCIA DUPLICADA DE UN CONTADOR	Dos o más instrucciones CNT se refieren al mismo número.
E480 DIRECCION INVALIDA DE CV	La instrucción CV se está usando en una rutina de interrupción de subrutina o de programa. La instrucción CV se puede utilizar solamente en el área principal del programa (antes de la declaración END).
E481 INSTRUCCION CON CONFLICTO	Existe una instrucción entre etapas de convergencia.
E482 INSTRUCCIONES CV EXCEDIDAS	La cantidad de instrucciones CV es más de 17.

Código de error del DL06	Description
E483 DIRECCION INVALIDA DE CV JUMP	CVJMP se ha utilizado en una subrutina o una rutina de interrupción del programa.
E484 L:A INSTRUCCION CV ESTÁ FALTANDO	CVJMP no es precedido por la instrucción CV. Una instrucción CVJMP debe seguir inmediatamente la instrucción CV.
E485 ESTÁ FALTANDO UNA INSTRUCCIÓN REQUERIDA.	No se ha colocado una instrucción CV JMP entre CV y la instrucción [SG, ISG, ST BLK, END BLK, END].
E486 DIRECCION INVALIDA CALL BLK	CALL BLK se está usando se utiliza en una subrutina o una rutina de interrupción del programa. La instrucción CALL BLK se puede utilizar solamente en el área principal del programa (antes de la declaración END).
E487 ESTÁ FALTANDO UNA INSTRUCCIÓN ST BLK	La instrucción CALL BLK no es seguida por una instrucción ST BLK.
E488 DIRECCION INVALIDA ST BLK	La instrucción ST BLK está siendo usada en una subrutina o una interrupción de programa. Otra instrucción ST BLK se está usando entre las instrucciones CALL BLK y END BLK.
E489 REFERENCIA DUPLICADA CR	El relevador de control usado para la instrucción BLK se está utilizando como salida en otra parte.
E490 ESTÁ FALTANDO INSTRUCCION SG	La instrucción BLK no es seguida inmediatamente por la instrucción SG.
E491 DIRECCION INVALIDA DE LA INSTRUCCION ISG	Hay una instrucción ISG entre las instrucciones ST BLK y END BLK.
E492 DIRECCION INVALIDA DE END BLK	La instrucción END BLK se está usando en un subrutina o una rutina de interrupción del programa. La instrucción END BLK no es seguida por una instrucción ST BLK.
E493 ESTÁ FALTANDO UNA INSTRUCCIÓN END	La instrucción [CV, SG, ISG, ST BLK, END] debe seguir inmediatamente la instrucción END BLK.
E494 ESTÁ FALTANDO LA INSTRUCCION END BLK	La instrucción ST BLK no es seguida por una instrucción END BLK..
E499 INSTRUCCION PRINT	Uso inválido de la instrucción PRINT. Las comillas (") y/o los espacios no fueron entrados o entraron incorrectamente.
E501 ENTRADA INCORRECTA	Un teclado o una serie de teclados inválidos fue entrado en el programador.
E502 DIRECCION INCORRECTA	Una dirección inválida fue entrada en el programador.
E503 COMANDO INCORRECTO	Un comando inválido fue entrado en el programador.
E504 REF/VALOR INCORRECTO	Un número inválido de un valor o de una referencia fue entrado con una instrucción.
E505 INSTRUCCION INVALIDA	Una instrucción inválida fue entrada en el programador.
E506 OPERACION INVALIDA	Se trató de hacer una operación inválida por el programador.
E520 OP-RUN INCORRECTA	Se trató de hacer una operación que es inválida en el modo RUN por el programador.
E521 OP-TRUN INCORRECTA	Se trató de hacer una operación que es inválida en el modo TEST RUN por el programador.
E523 OP-TPGM INCORRECTA	Se trató de hacer una operación que es inválida en el modo TEST PROGRAM por el programador.
E524 OP-PGM INCORRECTA	Se trató de hacer una operación que es inválida en el modo PROGRAM por el programador.

Apéndice B: Códigos de error

B

Código de error del DL06	Descripción
E525 CONMUTADOR DE MODO	Se trató de hacer una operación mientras el conmutador de modo DL06 estaba en una posición diferente de la posición TERM.
E526 OFF LINE	El programador está en el modo OFF LINE. Para cambiar al modo ON LINE utilice la tecla MODE.
E527 ON LINE	El programador está en el modo ON LINE. Para cambiar al modo OFF LINE utilice la tecla MODE.
E528 MODO DE LA CPU	La operación que se trata de hacer no se permite durante una modificación en modo RUN.
E540 CPU BLOQUEADA	El PLC DL06 ha sido bloqueado con contraseña .Para desbloquear el PLC DL06 use AUX 82 con la contraseña.
E541 CONTRASEÑA INCORRECTA	La contraseña usada para desbloquear el PLC DL06 con AUX 82 es incorrecta.
E542 CONTRASEÑA INCORRECTA	El DL06 fue energizado con una contraseña inválida y cambió la contraseña a 00000000. Se puede volver a entrar una contraseña usando AUX81.
E601 MEMORIA LLENA	Se trató de entrar una instrucción que requería más memoria que la disponible en el DL06.
E602 INSTRUCCION NO ENCONTRADA	Una función de búsqueda fue realizada y la instrucción no fue encontrada.
E603 ESTAN FALTANDO DATOS	Una función de búsqueda fue realizada y los datos no fueron encontrados.
E604 ESTA FALTANDO REFERENCIA	Una función de búsqueda fue realizada y la referencia no fue encontrada.
E610 TIPO DE E/S INCORRECTO	El programa en uso se ha referido a un módulo de E/S con un tipo incorrecto de módulo.
E620 OUT OF MEMORY	Se ha tratado de transferir más datos entre el DL06 y el programador de lo que el dispositivo de recepción puede aceptar.
E621 EEPROM NO VACIA	Se ha tratado de escribir a un EEPROM no vacío en el programador. Borre la memoria del EEPROM y después escriba nuevamente.
E622 NO HAY EEPROM EN D2-HPP	Se ha tratado de transferir datos sin EEPROM instalada en el programador (o posiblemente un EEPROM defectuoso).
E623 EEPROM SOLO CON SISTEMA	Fue solicitada una función al programador con una EEPROM que contiene solamente información del sistema ..
E624 EEPROM SOLO CON MEMORIA V	Fue solicitada una función al programador con una EEPROM que contiene solamente datos de memoria V.
E625 EEPROM SOLO CON PROGRAMA	Fue solicitada una función al programador con una EEPROM que contiene solamente datos de programa..
E627 BAD WRITE	Se trató de escribir a una EEPROM defectuosa en el programador. Sustituya la EEPROM si fuera necesario.
E628 ERROR DE TIPO DE EEPROM	Se está utilizando el tipo incorrecto de EEPROM.
E640 ERROR DE COMPARACION	Al comparar entre la EEPROM del programador y el DL06 se encontró un error.
E642 ERROR DE CHECKSUM	Fue detectado un error cuando los datos eran transferidos a la EEPROM del programador. Verifique cables y revise la operación.
E650 ERROR DE SISTEMA DEL D2-HPP	Ha ocurrido un error de sistema en el programador. Apague y encienda el programador. Si el error vuelve sustituya el programador.
E651 ERROR DE ROM DEL D2-HPP	Ha ocurrido un error de la ROM en el programador. Apague y encienda el programador. Si el error vuelve sustituya el programador.
E652 ERROR DE RAM DEL D2-HPP	Ha ocurrido un error de la RAM en el programador. .Apague y encienda el programador. Si el error vuelve, sustituya el programador.

TIEMPOS DE EJECUCIÓN DE CADA INSTRUCCIÓN



En este apéndice...

Introducción	C-2
Tiempos de ejecución de cada instrucción	C-3

Introducción

Este apéndice contiene varias tablas que entregan los tiempos de ejecución de las instrucciones en los PLCs DL06. Muchos de los tiempos de ejecución dependen del tipo de datos usados con la instrucción. Las memorias se pueden clasificar en los tipos siguientes:

- Memorias de datos (palabra).
- Memorias de bits.

Memorias de datos o Memoria V

Algunas direcciones de memoria se consideran registros de datos, tales como valores corrientes de temporizadores o contadores. La memoria normal de usuario se clasifica como registro de datos. Note que se puede cargar una configuración de bits en estos tipos de memoria, aunque su uso primario es para registros de datos. Las direcciones siguientes son registros de datos:

Registros de datos	DL06
Valores corrientes de temporizadores	V0 - V377
Valores corrientes de contadores	V1000 - V1177
Palabras de datos del usuario	V400 - V677 V1200 - V7377 V10000 - V17777

Memorias de bits

Recuerde que algunos de los puntos discretos tales como X, Y, C, etc. están relacionados automáticamente en la memoria V. Los bits siguientes contienen estos datos:

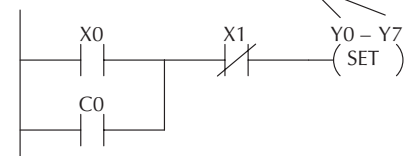
Registros de bits	DL06
Puntos de entradas (X)	V40400 - V40437
Puntos de salidas (Y)	V40500 - V40537
Relevadores de control (C)	V40600 - V40677
Etapas (S)	V41000 - V41077
Bits de estado de temporizadores	V41100 - V41177
Bits de estado de contadores	V41140 - V41147
Relevadores especiales (SP)	V41200 - V41237

Cómo leer las tablas

Algunas instrucciones pueden tener más de un parámetro. Por ejemplo, la instrucción SET mostrada en el programa ladder a la derecha puede configurar un solo punto o un rango de puntos.

En estos casos, los tiempos de ejecución dependen de la cantidad y del tipo de parámetros. Las tablas de tiempo de ejecución enumeran los tiempos de ejecución para ambas situaciones, según lo mostrado abajo:

Dos localizaciones de datos disponibles



SET	1er #: X, Y, C, S	9.2 μ s
	2ndo #: X, Y, C, S (N puntos)	9.6 μ s + 0.9 μ s x N
RST	1er #: X, Y, C, S	9.2 μ s
	2ndo #: X, Y, C, S (N puntos)	9.6 μ s + 0.9 μ s x N

La ejecución depende de la cantidad de localizaciones y tipos de datos usados

Tiempos de ejecución de instrucciones

Instrucciones booleanas

Instrucciones booleanas		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
STR	X, Y, C, T, CT, S, SP, GX, GY	0,67 μ s	0,00 μ s
STRN	X, Y, C, T, CT, S, SP, GX, GY	0,67 μ s	0,0 μ s
OR	X, Y, C, T, CT, S, SP, GX, GY	0,51 μ s	0,51 μ s
ORN	X, Y, C, T, CT, S, SP, GX, GY	0,55 μ s	0,55 μ s
AND	X, Y, C, T, CT, S, SP, GX, GY	0,42 μ s	0,42 μ s
ANDN	X, Y, C, T, CT, S, SP, GX, GY	0,51 μ s	0,51 μ s
ANDSTR	Ninguno	0,37 μ s	0,37 μ s
ORSTR	Ninguno	0,37 μ s	0,37 μ s
OUT	X, Y, C, GX, GY	1,82 μ s	1,82 μ s
OROUT	X, Y, C, GX, GY	2,09 μ s	2,09 μ s
NOT	Ninguno	1,04 μ s	1,04 μ s
SET	1o. #: X, Y, C, S, 2ndo. #: X, Y, C, S (N pt)	9,2 μ s 9,6 μ s + 0,9 μ s x N	1,0 μ s 1,1 μ s
RST	1o. #: X, Y, C, S, GX, GY 2ndo. #: X, Y, C, S (N pt), GX, GY	9,2 μ s 9,6 μ s + 0,9 μ s x N	1,0 μ s 1,1 μ s
	1o. #: T, CT, GX, GY 2ndo. #: T, CT (N pt), GX, GY	25,7 μ s 16,8 μ s + 2,7 μ s x N	1,1 μ s 1,4 μ s
PAUSE	1o: Y 2ndo: Y (N puntos)	5,6 μ s 9,2 μ s + 0,3 μ s x N	5,4 μ s 4,8 μ s

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
STRE	1o. V Reg Datos.	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,7 µs	27,7 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
	P:Indir. (Bit)	V:Reg Datos	29,9 µs	29,9 µs
V:Reg bits		29,9 µs	29,9 µs	
K:Constante		27,7 µs	27,7 µs	
P:Indir. (Datos)		51,0 µs	51,0 µs	
P:Indir. (Bit)		51,0 µs	51,0 µs	
STRNE	1o. V: Reg Datos.	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos.	30,3 µs	30,3 µs
		V:Reg bits	30,3 µs	30,3 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
	P:Indir. (Bit)	V:Reg Datos.	30,3 µs	30,3 µs
V:Reg bits		30,3 µs	30,3 µs	
K:Constante		27,4 µs	27,4 µs	
P:Indir. (Datos)		51,0 µs	51,0 µs	
P:Indir. (Bit)		51,0 µs	51,0 µs	

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación (cont.)			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
ORE	1o. V Reg Datos	2ndo. V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos	30,3 µs	30,3 µs
		V:Reg bits	30,3 µs	30,3 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	50,4 µs	50,4 µs
		P:Indir. (Bit)	50,4 µs	50,4 µs
	P:Indir. (Bit)	V:Reg Datos	30,3 µs	30,3 µs
V:Reg bits		30,3 µs	30,3 µs	
K:Constante		27,4 µs	27,4 µs	
P:Indir. (Datos)		50,4 µs	50,4 µs	
P:Indir. (Bit)		50,4 µs	50,4 µs	
ORNE	1o. Reg Datos.	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos.	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
	P:Indir. (Bit)	V:Reg Datos.	29,9 µs	29,9 µs
V:Reg bits		29,9 µs	29,9 µs	
K:Constante		27,4 µs	27,4 µs	
P:Indir. (Datos)		51,0 µs	51,0 µs	
P:Indir. (Bit)		51,0 µs	51,0 µs	

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación (cont.)			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
ANDE	1o. V Reg Datos.	2ndo. V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
	P:Indir. (Bit)	V:Reg Datos	29,9 µs	29,9 µs
V:Reg bits		29,9 µs	29,9 µs	
K:Constante		27,4 µs	27,4 µs	
P:Indir. (Datos)		51,0 µs	51,0 µs	
P:Indir. (Bit)		51,0 µs	51,0 µs	
ANDNE	1o. V: Reg Datos.	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos.	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
	P:Indir. (Bit)	V:Reg Datos.	29,9 µs	29,9 µs
V:Reg bits		29,9 µs	29,9 µs	
K:Constante		27,4 µs	27,4 µs	
P:Indir. (Datos)		51,0 µs	51,0 µs	
P:Indir. (Bit)		51,0 µs	51,0 µs	

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
STR	1o. T, CT	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V Reg Datos	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs	
	V:Reg bits	7,6 µs	7,6 µs	
	K:Constante	4,8 µs	4,8 µs	
	P:Indir. (Datos)	30,2 µs	30,2 µs	
	P:Indir. (Bit)	30,2 µs	30,2 µs	
P:Indir. (Datos)	V:Reg Datos.	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	
P:Indir. (Bit)	V:Reg Datos	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	
STRN	1o. T, CT	2ndo.o.. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg Datos.	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
STRN (cont.)	1o. V: Reg bits	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos.	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
P:Indir. (Bit)		51,0 µs	51,0 µs	
P:Indir. (Bit)	V:Reg Datos.	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	

Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
OR	1o. T, CT	2ndo. V Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V Reg Datos.	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
P:Indir. (Bit)	V:Reg Datos	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación (cont.)			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
ORN	1o. T, CT	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg Datos	V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs	
	V:Reg bits	7,6 µs	7,6 µs	
	K:Constante	4,8 µs	4,8 µs	
	P:Indir. (Datos)	30,2 µs	30,2 µs	
	P:Indir. (Bit)	30,2 µs	30,2 µs	
P:Indir. (Datos)	V:Reg Datos.	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	
P:Indir. (Bit)	V:Reg Datos.	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	

C

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación (cont.)		DL06		
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
AND	1o. T, CT	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V:Reg Datos.	V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
	P:Indir. (Datos)	V:Reg Datos	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
P:Indir. (Bit)	V:Reg Datos	29,9 µs	29,9 µs	
	V:Reg bits	29,9 µs	29,9 µs	
	K:Constante	27,4 µs	27,4 µs	
	P:Indir. (Datos)	51,0 µs	51,0 µs	
	P:Indir. (Bit)	51,0 µs	51,0 µs	

Instrucciones booleanas de comparación

Instrucciones booleanas de comparación (cont.)			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
ANDN	1o. T, CT	2ndo. V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
V: Reg Datos.	V: Reg Datos.	V:Reg Datos	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
V: Reg bits	V: Reg bits	V:Reg Datos.	7,6 µs	7,6 µs
		V:Reg bits	7,6 µs	7,6 µs
		K:Constante	4,8 µs	4,8 µs
		P:Indir. (Datos)	30,2 µs	30,2 µs
		P:Indir. (Bit)	30,2 µs	30,2 µs
P:Indir. (Datos)	P:Indir. (Datos)	V:Reg Datos.	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs
P:Indir. (Bit)	P:Indir. (Bit)	V:Reg Datos.	29,9 µs	29,9 µs
		V:Reg bits	29,9 µs	29,9 µs
		K:Constante	27,4 µs	27,4 µs
		P:Indir. (Datos)	51,0 µs	51,0 µs
		P:Indir. (Bit)	51,0 µs	51,0 µs

Instrucciones Inmediatas

Instrucciones inmediatas		DL06	
Instrucción	Tipos de de datos	Ejecución	Sin ejecución
LDI	V	20.6 µs	1.1 µs
LDIF	1r. #: Y 2o. #: K Constante	26.6 µs+0.9µs x N	1.4 µs
STRI	X	19.3 µs	19.3 µs
STRNI	X	19.4 µs	19.4 µs
ORI	X	19.1 µs	18.7 µs
ORNI	X	19.2 µs	18.9 µs
ANDI	X	18.7 µs	18.7 µs
ANDNI	X	18.8 µs	18.8 µs
OUTI	Y	25.5 µs	25.5 µs
OROUTI	Y	25.7 µs	25.7 µs
OUTIF	1r. #: Y 2o. #: Y (N pt)	66.1 µs+0.9µs x N	1.4 µs
SETI	1r. #: Y 2o. #: K Constant	23.1 µs, 22.8 µs+1.4µsXN	0.9 µs, 0.9 µs
RSTI	1r. #: Y 2o. #: Y (N pt)	23.2 µs, 22.8 µs+1.4µsXN	0.9 µs, 0.9 µs

Instrucciones booleanas Bit of Word

Instrucciones booleanas Bit of Word		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
STRB	V:Reg Datos.	3,1 µs	3,1 µs
	V:Reg bits	3,1 µs	3,1 µs
	P:Indir. (Datos)	30,0 µs	30,0 µs
	P:Indir. (Bit)	30,0 µs	30,0 µs
STRNB	V:Reg Datos.	3,0 µs	3,0 µs
	V:Reg bits	3,0 µs	3,0 µs
	P:Indir. (Datos)	29,8 µs	29,8 µs
	P:Indir. (Bit)	29,8 µs	29,8 µs
ORB	V:Reg Datos.	2,9 µs	2,9 µs
	V:Reg bits	2,9 µs	2,9 µs
	P:Indir. (Datos)	29,9 µs	29,9 µs
	P:Indir. (Bit)	29,9 µs	29,9 µs
ORNB	V:Reg Datos.	2,8 µs	2,8 µs
	V:Reg bits	2,8 µs	2,8 µs
	P:Indir. (Datos)	29,6 µs	29,6 µs
	P:Indir. (Bit)	29,6 µs	29,6 µs
ANDB	V:Reg Datos	2,8 µs	2,8 µs
	V:Reg bits	2,8 µs	2,8 µs
	P:Indir. (Datos)	29,6 µs	29,6 µs
	P:Indir. (Bit)	29,6 µs	29,6 µs
ANDNB	V:Reg Datos.	2,7 µs	2,7 µs
	V:Reg bits	2,7 µs	2,7 µs
	P:Indir. (Datos)	29,6 µs	29,6 µs
	P:Indir. (Bit)	29,6 µs	29,6 µs
OUTB	V:Reg Datos.	3,1 µs	3,4 µs
	V:Reg bits	3,1 µs	3,4 µs
	P:Indir. (Datos)	30,3 µs	30,7 µs
	P:Indir. (Bit)	30,3 µs	30,7 µs
SETB	V:Reg Datos.	13,4 µs	3,4 µs
	V:Reg bits	13,4 µs	3,4 µs
	P:Indir. (Datos)	41,1 µs	29,1 µs
	P:Indir. (Bit)	41,1 µs	29,1 µs
RSTB	V:Reg Datos.	13,5 µs	1,4 µs
	V:Reg bits	13,5 µs	1,4 µs
	P:Indir. (Datos)	41,3 µs	29,1 µs
	P:Indir. (Bit)	41,3 µs	29,1 µs

Temporizador, Contador y Shift Register

Temporizador, Contador y Shift Register			DL06	
Instrucción	Tipos legales de datos		Ejecución	Sin ejecución
TMR	1o. T	2ndo. V: Reg Datos.	26,8 µs	7,3 µs
		V: Reg bits	26,8 µs	7,3 µs
TMRF	T	K: Constante	20,0 µs	4,8 µs
		P: Indir. (Datos)	45,6 µs	30,2 µs
		P: Indir. (Bit)	45,6 µs	30,2 µs
		V: Reg Datos.	51,4 µs	7,3 µs
		V: Reg bits	51,4 µs	7,3 µs
TMRA	T	K: Constante	48,4 µs	4,6 µs
		P: Indir. (Datos)	75,9 µs	30,2 µs
		P: Indir. (Bit)	75,9 µs	30,2 µs
		V: Reg Datos.	48,9 µs	7,3 µs
		V: Reg bits	48,9 µs	7,3 µs
TMRAF	T	K: Constante	45,0 µs	4,6 µs
		P: Indir. (Datos)	75,9 µs	30,2 µs
		P: Indir. (Bit)	75,9 µs	30,2 µs
		V: Reg Datos.	54,2 µs	7,3 µs
		V: Reg bits	54,2 µs	7,3 µs
CNT	CT	K: Constante	50,3 µs	4,6 µs
		P: Indir. (Datos)	81,2 µs	30,2 µs
		P: Indir. (Bit)	81,2 µs	30,2 µs
		V: Reg Datos.	25,8 µs	7,3 µs
		V: Reg bits	25,8 µs	7,3 µs
SGCNT	CT	K: Constante	22,2 µs	4,6 µs
		P: Indir. (Datos)	53,5 µs	30,2 µs
		P: Indir. (Bit)	53,5 µs	30,2 µs
		V: Reg Datos.	27,3 µs	7,3 µs
		V: Reg bits	27,3 µs	7,3 µs
UDC	CT	K: Constante	23,5 µs	4,6 µs
		P: Indir. (Datos)	54,9 µs	30,2 µs
		P: Indir. (Bit)	54,9 µs	30,2 µs
		V: Reg Datos	39,8 µs	7,3 µs
		V: Reg bits	39,8 µs	7,3 µs
SR	C (N puntos a deslocar)	K: Constante	35,4 µs	4,6 µs
		P: Indir. (Datos)	67,8 µs	30,2 µs
		P: Indir. (Bit)	67,8 µs	30,2 µs
		V: Reg Datos	17,8 µs +	9,8 µs
		V: Reg bits	0,9 µs x N	

Instrucciones de datos de acumulador

Instrucciones datos de acumulador/stack		DL06		
Instrucción	Tipos de datos legales		Ejecución	Sin ejecución
LD	V:Reg Datos.		11,8 µs	1,0 µs
	V:Reg bits		11,8µs	1,0 µs
	K:Constante		9,0 µs	1,0 µs
	P:Indir. (Datos)		33,9 µs	0,9 µs
	P:Indir. (Bit)		33,9 µs	0,9 µs
LDD	V:Reg Datos.		12,2 µs	1,0 µs
	V:Reg bits		12,2 µs	1,0 µs
	K:Constante		9,0 µs	1,0 µs
	P:Indir. (Datos)		37,8 µs	0,9 µs
	P:Indir. (Bit)		37,8 µs	0,9 µs
LDF	1o. X, Y, C, S T, CT, SP	2ndo. K:Constante	20,5 µs+0,9 µs×N	0,9 µs
LDA	O: (Constante octal para dirección)		10,4 µs	1,0 µs
LDR	V:Reg Datos.		29,5 µs	1,0 µs
	V:Reg bits		29,5 µs	1,0 µs
	K:Constante		25,5 µs	1,0 µs
	P:Indir. (Datos)		54,9 µs	1,0 µs
	P:Indir. (Bit)		54,9 µs	1,0 µs
LDSX	K: Constante		14,6 µs	1,0 µs
LDX	V:Reg Datos.		10,8 µs	1,0 µs
	V:Reg bits		10,8 µs	1,0 µs
	P:Indir. (Datos)		45,2 µs	1,0 µs
	P:Indir. (Bit)		45,2 µs	1,0 µs
OUT	V:Reg Datos.		9,3 µs	1,0 µs
	V:Reg bits		9,3 µs	1,0 µs
	P:Indir. (Datos)		35,2 µs	0,9 µs
	P:Indir. (Bit)		35,2 µs	0,9 µs
OUTD	V:Reg Datos.		10,2 µs	1,0 µs
	V:Reg bits		10,2 µs	1,0 µs
	P:Indir. (Datos)		35,8 µs	0,9 µs
	P:Indir. (Bit)		35,8 µs	0,9 µs
OUTF	1o. X, Y, C	2ndo. K:Constante	54 µs+1,0 µs×N	0,9 µs
OUTL	V:Reg Datos.		13,5 µs	1,0 µs
	V:Reg bits		13,5 µs	1,0 µs
OUTM	V:Reg Datos.		13,7 µs	1,0 µs
	V:Reg bits		13,7 µs	1,0 µs
OUTX	V:Reg Datos.		17,2 µs	1,0 µs
	V:Reg bits		17,2 µs	1,0 µs
	P:Indir. (Datos)		43,4 µs	1,0 µs
	P:Indir. (Bit)		43,4 µs	1,0 µs
POP	Ninguno		8,4 µs	1,0 µs

Instrucciones lógicas

Instrucciones lógicas (acumulador)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
AND	V:Reg Datos.	7,9 µs	1,0 µs
	V:Reg bits	7,9 µs	1,0 µs
	P:Indir. (Datos)	33,4 µs	0,9 µs
	P:Indir. (Bit)	33,4 µs	0,9 µs
ANDD	V:Reg Datos.	8,9 µs	1,0 µs
	V:Reg bits	8,9 µs	1,0 µs
	K:Constante	5,7 µs	1,0 µs
	P:Indir. (Datos)	34,4 µs	0,9 µs
	P:Indir. (Bit)	34,4 µs	0,9 µs
ANDF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	21,6 µs + 0,9 µs x N	1,0 µs
ANDS	Ninguno	10,0 µs	1,0 µs
OR	V:Reg Datos	8,1 µs	1,0 µs
	V:Reg bits	8,1 µs	1,0 µs
	P:Indir. (Datos)	33,8 µs	0,9 µs
	P:Indir. (Bit)	33,8 µs	0,9 µs
ORD	V:Reg Datos.	9,0 µs	1,0 µs
	V:Reg bits	9,0 µs	1,0 µs
	K:Constante	5,8 µs	1,0 µs
	P:Indir. (Datos)	34,5 µs	0,9 µs
	P:Indir. (Bit)	34,5 µs	0,9 µs
ORF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	20,9 µs + 0,9 µs x N	1,0 µs
ORS	Ninguno	10,2 µs	1,0 µs
XOR	V:Reg Datos.	8,0 µs	1,0 µs
	V:Reg bits	8,0 µs	1,0 µs
	P:Indir. (Datos)	33,6 µs	0,9 µs
	P:Indir. (Bit)	33,6 µs	0,9 µs
XORD	V:Reg Datos.	9,0 µs	1,0 µs
	V:Reg bits	9,0 µs	1,0 µs
	K:Constante	5,4 µs	1,0 µs
	P:Indir. (Datos)	34,4 µs	0,9 µs
	P:Indir. (Bit)	34,4 µs	0,9 µs
XORF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	20,9 µs + 0,9 µs x N	1,0 µs
XORS	Ninguno	10,1 µs	1,0 µs
CMP	V:Reg Datos.	9,4 µs	1,0 µs
	V:Reg bits	9,4 µs	1,0 µs
	P:Indir. (Datos)	34,9 µs	0,9 µs
	P:Indir. (Bit)	34,9 µs	0,9 µs
CMPD	V:Reg Datos.	9,9 µs	1,0 µs
	V:Reg bits	9,9 µs	1,0 µs
	K:Constante	6,7 µs	1,0 µs
	P:Indir. (Datos)	35,4 µs	1,0 µs
	P:Indir. (Bit)	35,4 µs	1,0 µs

Instrucciones lógicas (Acumulador) (cont.)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
CMPF	1o.: X, Y, C, S	20,9 μ s + 1,0 μ s x N	1,0 μ s
	T, CT, SP, GX, GY		
CMPR	2ndo.: K:Constante		
	V:Reg Datos.	42,8 μ s	1,0 μ s
	V:Reg bits	42,8 μ s	1,0 μ s
	K:Constante	38,4 μ s	1,0 μ s
	P:Indir. (Datos)	69,0 μ s	1,0 μ s
P:Indir. (Bit)	69,0 μ s	1,0 μ s	
CMPS	Ninguno	11,2 μ s	1,0 μ s

Instrucciones aritméticas

Instrucciones aritméticas (Acumulador)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
ADD	V:Reg Datos.	78,4 μ s	0,9 μ s
	V:Reg bits	78,4 μ s	0,9 μ s
	P:Indir. (Datos)	101,2 μ s	0,9 μ s
	P:Indir. (Bit)	101,2 μ s	0,9 μ s
ADDD	V:Reg Datos.	83,3 μ s	0,9 μ s
	V:Reg bits	83,3 μ s	0,9 μ s
	K:Constante	67,7 μ s	0,9 μ s
	P:Indir. (Daa)	101,2 μ s	0,9 μ s
	P:Indir. (Bit)	101,2 μ s	0,9 μ s
SUB	V:Reg Datos.	77,4 μ s	0,9 μ s
	V:Reg bits	77,4 μ s	0,9 μ s
	P:Indir. (Datos)	95,1 μ s	0,9 μ s
	P:Indir. (Bit)	95,1 μ s	0,9 μ s
SUBD	V:Reg Datos.	82,5 μ s	0,9 μ s
	V:Reg bits	82,5 μ s	0,9 μ s
	K:Constante	66,0 μ s	0,9 μ s
	P:Indir. (Datos)	99,7 μ s	0,9 μ s
	P:Indir. (Bit)	99,7 μ s	0,9 μ s
MUL	V:Reg Datos.	266,1 μ s	0,9 μ s
	V:Reg bits	266,1 μ s	0,9 μ s
	K:Constante	286,9 μ s	0,9 μ s
	P:Indir. (Datos)	290,0 μ s	0,9 μ s
	P:Indir. (Bit)	290,0 μ s	0,9 μ s
MULD	V:Reg Datos.	839,1 μ s	0,9 μ s
	V:Reg bits	839,1 μ s	0,9 μ s
	P:Indir. (Datos)	863,1 μ s	0,9 μ s
	P:Indir. (Bit)	863,1 μ s	0,9 μ s
DIV	V:Reg Datos.	363,9 μ s	0,9 μ s
	V:Reg bits	363,9 μ s	0,9 μ s
	K:Constante	384,4 μ s	0,9 μ s
	P:Indir. (Datos)	419,8 μ s	0,9 μ s
	P:Indir. (Bit)	419,8 μ s	0,9 μ s
DIVD	V:Reg Datos.	398,3 μ s	0,9 μ s
	V:Reg bits	398,3 μ s	0,9 μ s
	P:Indir. (Datos)	390,9 μ s	0,9 μ s
	P:Indir. (Bit)	390,9 μ s	0,9 μ s

Instrucciones aritméticas

Instrucciones aritméticas (Acumulador)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
INC	V:Reg Datos	48,5 µs	1,0 µs
	V:Reg bits	48,5 µs	1,0 µs
	P:Indir. (Datos)	74,7 µs	1,0 µs
	P:Indir. (Bit)	74,7 µs	1,0 µs
DEC	V:Reg Datos.	47,5 µs	1,0 µs
	V:Reg bits	47,5 µs	1,0 µs
	P:Indir. (Datos)	71,5 µs	1,0 µs
	P:Indir. (Bit)	71,5 µs	1,0 µs
INCB	V:Reg Datos.	13,2 µs	1,0 µs
	V:Reg bits	13,2 µs	1,0 µs
	P:Indir. (Datos)	38,6 µs	0,9 µs
	P:Indir. (Bit)	38,6 µs	0,9 µs
DECB	V:Reg Datos.	13,2 µs	1,0 µs
	V:Reg bits	13,2 µs	1,0 µs
	P:Indir. (Datos)	38,0 µs	0,9 µs
	P:Indir. (Bit)	38,0 µs	0,9 µs
ADDB	V:Reg Datos.	24,9 µs	1,0 µs
	V:Reg bits	24,9 µs	1,0 µs
	K:Constante	23,5 µs	1,0 µs
	P:Indir. (Datos)	51,1 µs	1,0 µs
	P:Indir. (Bit)	51,1 µs	1,0 µs
ADDBD	V:Reg Datos.	24,4 µs	1,0 µs
	V:Reg bits	24,4 µs	1,0 µs
	K:Constante	20,7 µs	1,0 µs
	P:Indir. (Datos)	50,7 µs	1,0 µs
	P:Indir. (Bit)	50,7 µs	1,0 µs
SUBB	V:Reg Datos.	24,7 µs	1,0 µs
	V:Reg bits	24,7 µs	1,0 µs
	K:Constante	23,3 µs	1,0 µs
	P:Indir. (Datos)	50,6 µs	1,0 µs
	P:Indir. (Bit)	50,6 µs	1,0 µs
SUBBD	V:Reg Datos.	24,2 µs	1,0 µs
	V:Reg bits	24,2 µs	1,0 µs
	K:Constante	20,2 µs	1,0 µs
	P:Indir. (Datos)	50,2 µs	1,0 µs
	P:Indir. (Bit)	50,2 µs	1,0 µs
MULB	V:Reg Datos.	10,8 µs	1,0 µs
	V:Reg bits	10,8 µs	1,0 µs
	K:Constante	8,2 µs	1,0 µs
	P:Indir. (Datos)	37,1 µs	1,0 µs
	P:Indir. (Bit)	37,1 µs	1,0 µs
DIVB	V:Reg Datos.	28,7 µs	1,0 µs
	V:Reg bits	28,7 µs	1,0 µs
	K:Constante	26,1 µs	1,0 µs
	P:Indir. (Datos)	54,9 µs	1,0 µs
	P:Indir. (Bit)	54,9 µs	1,0 µs
ADDR	V:Reg Datos.	48,1 µs	1,0 µs
	V:Reg bits	48,1 µs	1,0 µs
	K:Constante	41,7 µs	1,0 µs
	P:Indir. (Datos)	74,3 µs	1,0 µs
	P:Indir. (Bit)	74,3 µs	1,0 µs

Apéndice C: Tiempos de ejecución de instrucciones

Instrucciones aritméticas (Acumulador)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
SUBR	V:Reg Datos.	50,1 μ s	1,0 μ s
	V:Reg bits	50,1 μ s	1,0 μ s
	K:Constante	58,7 μ s	1,0 μ s
	P:Indir. (Datos)	76,3 μ s	1,0 μ s
	P:Indir. (Bit)	76,3 μ s	1,0 μ s
MULR	V:Reg Datos.	54,2 μ s	1,0 μ s
	V:Reg bits	54,2 μ s	1,0 μ s
	K:Constante	42,7 μ s	1,0 μ s
	P:Indir. (Datos)	80,4 μ s	1,0 μ s
	P:Indir. (Bit)	80,4 μ s	1,0 μ s
DIVR	V:Reg Datos.	50,1 μ s	1,0 μ s
	V:Reg bits	50,1 μ s	1,0 μ s
	K:Constante	58,7 μ s	1,0 μ s
	P:Indir. (Datos)	76,3 μ s	1,0 μ s
	P:Indir. (Bit)	76,3 μ s	1,0 μ s
ADDF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	109,3 μ s + 0,9 μ s x N	1,0 μ s
SUBF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	107,3 μ s + 0,9 μ s x N	1,0 μ s
MULF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	352,5 μ s + 0,9 μ s x N	1,0 μ s
DIVF	1o.: X, Y, C, S T, CT, SP, GX, GY 2ndo.: K:Constante	477,3 μ s + 0,8 μ s x N	1,0 μ s
ADDS	None	99,5 μ s	1,0 μ s
SUBS	Ninguno	97,5 μ s	1,0 μ s
MULS	Ninguno	342,5 μ s	1,0 μ s
DIVS	Ninguno	467,3 μ s	1,0 μ s
ADDBS	Ninguno	24,3 μ s	1,0 μ s
SUBBS	Ninguno	23,7 μ s	1,0 μ s
MULBS	Ninguno	11,7 μ s	1,0 μ s
DIVBS	Ninguno	29,7 μ s	1,0 μ s
SQRTR	Ninguno	87,9 μ s	1,0 μ s
SINR	Ninguno	226,8 μ s	1,0 μ s
COSR	Ninguno	213,1 μ s	1,0 μ s
TANR	Ninguno	285,5 μ s	1,0 μ s
ASINR	Ninguno	489,8 μ s	1,0 μ s
ACOSR	Ninguno	508,3 μ s	1,0 μ s
ATANR	Ninguno	317,1 μ s	1,0 μ s

Instrucciones diferenciales

Instrucciones diferenciales		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
PD	X, Y, C	14,4 µs	14,4 µs
STRPD	X, Y, C, S, T, CT	5,4 µs	5,4 µs
STRND	X, Y, C, S, T, CT	7,3 µs	7,3 µs
ORPD	X, Y, C, S, T, CT	6,8 µs	5,2 µs
ORND	X, Y, C, S, T, CT	7,1 µs	4,9 µs
ANDPD	X, Y, C, S, T, CT	6,8 µs	5,2 µs
ANDND	X, Y, C, S, T, CT	7,1 µs	4,9 µs

Instrucciones de bits

Instrucciones de bits (Acumulador)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
SUM	Ninguno	6,7 µs	1,0 µs
SHFR	V:Reg Datos. (N bits)	12,1 µs + 0,1 x N	0,9 µs
	V:Reg bits (N bits) K:Constante (N bits)	8,4 µs + 0,1 x N	
SHFL	V:Reg Datos. (N bits)	12,1 µs + 0,1 x N	0,9 µs
	V:Reg bits (N bits) K:Constante (N bits)	8,4 µs + 0,1 x N	
ROTR	V:Reg Datos. (N bits)	16,4 µs	1,0 µs
	V:Reg bits (N bits)	16,4 µs	1,0 µs
	K:Constante (N bits)	12,9 µs	1,0 µs
ROTL	V:Reg Datos. (N bits)	16,4 µs	1,0 µs
	V:Reg bits (N bits)	16,4 µs	1,0 µs
	K:Constante (N bits)	12,7 µs	1,0 µs
ENCO	Ninguno	33,9 µs	0,9 µs
DECO	Ninguno	5,7 µs	1,0 µs

Instrucciones de conversión de números

Instrucciones de conversión de números		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
BIN	Ninguno	100,2 µs	0,9 µs
BCD	Ninguno	95,2 µs	0,9 µs
INV	Ninguno	2,5 µs	1,0 µs
BCDPL	Ninguno	75,6 µs	1,0 µs
ATH	V	25,4 µs	1,0 µs
HTA	V	25,4 µs	1,0 µs
GRAY	Ninguno	110,8 µs	1,0 µs
SFLDGT	Ninguno	23,1 µs	1,0 µs
BTOR	Ninguno	18,6 µs	1,0 µs
RTOB	Ninguno	8,6 µs	1,0 µs
RADR	Ninguno	51,4 µs	1,0 µs
DEGR	Ninguno	81,5 µs	1,0 µs

Instrucciones de tablas de memorias

Instrucciones de tablas		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
MOV	Move V:Reg Datos. a V:Reg Datos Move V:Reg bits a V:Reg Datos Move V:Reg Datos. a V:Reg bits Move V:Reg bits a V:Reg bits N=# de palabras	60,2 µs+9,5 x N	0,9 µs
MOVMC	Move V:Reg Datos a E ² Move V:Reg bits a E ² Move desde E ² a V:Reg Datos Move desde E ² s a V:Reg bits N= #de palabras	35 µs + 10,4 µs x N	0,9 µs
LDLBL	K	6,4 µs	1,3 µs
FILL	V: Reg Datos V:Reg bits	29,4 µs + 8,0 µs x N	1,0 µs
	K:Constante	26,2 µs + 8,0 µs x N	1,0 µs
	P:Indir. (Datos) P:Indir. (bit)	55,1 µs + 8,0 µs x N	1,0 µs
FIND	V: Reg Datos (N bits)	66,8 µs	1,0 µs
	V:Reg bits (N bits)	66,8 µs	1,0 µs
	K:Constante(N bits)	64,0 µs	1,0 µs

Instrucciones de tablas (cont.)		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
FDGT	V: Reg Datos (N bits)	66,1 µs	1,0 µs
	V:Reg bits (N bits)	66,1 µs	1,0 µs
	K:Constante(N bits)	55,2 µs	1,0 µs
FINDB	V: Reg Datos (N bits)	210,8 µs	1,0 µs
	V:Reg bits (N bits)	210,8 µs	1,0 µs
	P:Indir. (Datos)	237,0 µs	1,0 µs
	P:Indir. (Bit)	237,0 µs	1,0 µs
TTD	V: Reg Datos	66,9 µs	1,0 µs
	V:Reg bits	66,9 µs	1,0 µs
RFB	V: Reg Datos	66,8 µs	1,0 µs
	V:Reg bits	66,8 µs	1,0 µs
STT	V: Reg Datos	67,8 µs	1,0 µs
	V:Reg bits	67,8 µs	1,0 µs
	K:Constante	65,0 µs	1,0 µs
RFT	V: Reg Datos	51,1 µs	1,0 µs
	V:Reg bits	51,1 µs	1,0 µs
ATT	V: Reg Datos	53,5 µs	1,0 µs
	V:Reg bits	53,5 µs	1,0 µs
	K:Constante	50,8 µs	1,0 µs
TSHFL	V: Reg Datos	134,0 µs	1,0 µs
	V:Reg bits	134,0 µs	1,0 µs
TSHFR	V: Reg Datos	133,9 µs	1,0 µs
	V:Reg bits	133,9 µs	1,0 µs
ANDMOV	V: Reg Datos	80,2 µs	1,0 µs
	V:Reg bits	80,2 µs	1,0 µs
ORMOV	V: Reg Datos	80,4 µs	1,0 µs
	V:Reg bits	80,4 µs	1,0 µs
XORMOV	V: Reg Datos	80,4 µs	1,0 µs
	V:Reg bits	80,4 µs	1,0 µs
SWAP	V: Reg Datos	84,1 µs	1,0 µs
	V:Reg bits	84,1 µs	1,0 µs
SETBIT	V: Reg Datos (N bits)	59,5 µs	1,0 µs
	V:Reg bits (N bits)	59,5 µs	1,0 µs
RSTBIT	V: Reg Datos (N bits)	59,5 µs	1,0 µs
	V:Reg bits (N bits)	59,5 µs	1,0 µs

Instrucciones de control de la CPU

Instrucciones de control de la CPU		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
NOP	Ninguno	1,1 µs	1,1 µs
END	Ninguno	24,0 µs	24,0 µs
STOP	Ninguno	10,0 µs	1,1 µs
RSTWT	Ninguno	5,9 µs	2,2 µs

Instrucciones de control del programa

Instrucciones de control del programa		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
GOTO	K	5,1 µs	4,8 µs
LBL	K	5,7 µs	0,0 µs
FOR	V, K	125,9 µs	14,5 µs
NEXT	Ninguno	64,4 µs	64,4 µs
GTS	K	27,5 µs	14,8 µs
SBR	K	1,5 µs	1,5 µs
RTC	Ninguno	25,7 µs	12,1 µs
RT	Ninguno	21,2 µs	21,2 µs
MLS	K	(1-7) 35,2 µs	35,2 µs
MLR	K	(0-7) 30,9 µs	30,9 µs

Instrucciones de interrupción

Instrucciones de interrupción		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
ENI	Ninguno	24,2 µs	2,7 µs
DISI	Ninguno	9,4 µs	2,3 µs
INT	O(0,1)	7,5 µs	-
IRTC	Ninguno	0,9 µs	1,3 µs
IRT	Ninguno	6,6 µs	-

Instrucciones de red de comunicación

Instrucciones de red		DL06	
Instrucción	Tipos de datos legales	Ejecución	Sin ejecución
RX	X, Y, C, T, CT, SP, S, \$	852,0 µs	4,4 µs
	V:Reg Datos.	852,0 µs	4,4 µs
	V:Reg bits	852,0 µs	4,4 µs
	P:Indir. (Datos)	868,2 µs	4,2 µs
	P:Indir. (Bit)	868,2 µs	4,2 µs
WX	X, Y, C, T, CT, SP, S, \$	1614,0 µs	4,4 µs
	V:Reg Datos.	1614,0 µs	4,4 µs
	V:Reg bits	1614,0 µs	4,4 µs
	P:Indir. (Datos)	1630,0 µs	4,4 µs
	P:Indir. (Bit)	1630,0 µs	4,4 µs

Instrucciones de entradas y salidas inteligentes

Instrucciones de red		DL06	
Instrucción	Tipos de datos legales	Ejecución	Sin ejecución
RD	V:Reg Datos.	385,7 μ s	1,2 μ s
	V:Reg bits	385,7 μ s	1,2 μ s
WT	V:Reg Datos.	385,6 μ s	1,2 μ s
	V:Reg bits	385,6 μ s	1,2 μ s

Instrucciones de mensajes

Instrucciones de mensaje		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
FAULT	V:Reg Datos.	65,0 μ s	4,4 μ s
	V:Reg bits	65,0 μ s	4,4 μ s
	K:Constante	204,7 μ s	4,4 μ s
DLBL	K	–	–
NCON	K	–	–
ACON	A	–	–
PRINT	ASCII	631,0 μ s	3,6 μ s

Instrucciones RLL *plus*

Instrucciones RLL ^{plus}		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
ISG	S	44,0 μ s	41,1 μ s
SG	S	44,0 μ s	41,1 μ s
JMP	S	76,0 μ s	9,3 μ s
NJMP	S	77,4 μ s	9,3 μ s
CV	S	42,1 μ s	27,5 μ s
CVJMP	S	89,5 μ s	17,6 μ s
BCALL	C	22,1 μ s	22,6 μ s
BLK	C	17,1 μ s	14,6 μ s
BEND	Ninguno	8,7 μ s	0,0 μ s

Instrucciones Drum

Instrucciones Drum		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
DRUM	CT	840,0 μ s	339,6 μ s
EDRUM	CT	753,2 μ s	357,0 μ s
MDRMD	CT	411,3 μ s	216,4 μ s
MDRMW	CT	378,6 μ s	147,0 μ s

Instrucciones de fecha y hora

Instrucciones de fecha y hora		DL06	
Instrucción		Ejecución	Sin ejecución
DATE	V:Reg Datos. V:Reg bits	24,0 µs	1,2 µs
TIME	V:Reg Datos. V:Reg bits	50,8 µs	1,2 µs

Instrucciones MODBUS

Instrucciones MODBUS		DL06	
Instrucción		Ejecución	Sin ejecución
MRX	Input, Input Register Coil, Holding Register	120,2 µs	1,3 µs
MWX	Input, Input Register Coil, Holding Register	21,3 µs	1,3 µs

Instrucciones ASCII

Instrucciones ASCII		DL06	
Instrucción	Tipos legales de datos	Ejecución	Sin ejecución
AIN	V	13,9 µs	12,0 µs
AFIND	V	111,5 µs	1,3 µs
AEX	V	111,7 µs	1,3 µs
CMPV	V	12,2 µs	1,3 µs
SWAPB	V	109,8 µs	1,3 µs
VPRINT	Datos de texto	161,6 µs	1,3 µs
PRINTV	V	163,3 µs	1,3 µs
ACRB	V	3,9 µs	1,1 µs

RELEVADORES ESPECIALES



En este apéndice...

Relevadores especiales del PLC DL06D-2

Relevadores especiales del PLC DL06

Los "Relevadores especiales " son contactos que son configurados por el sistema operativo de la CPU para indicar que ha ocurrido un evento particular del sistema. Estos contactos están disponibles para uso en su programa ladder. Se puede ahorrar mucho tiempo de programación sabiendo el contacto especial correcto del relevador a ser usado en una situación particular . Ya que la CPU activa y desactiva contactos de relevadores especiales, el programa ladder tiene que usarlos solamente como entradas lógicas.

D

Relevadores de partida y de pulsos de tiempo definido

SP0	Primer barrido	Se activa solamente en el primer barrido después de energizar el PLC o una transición de modo Program para modo RUN. El relevador se desactiva en el segundo ciclo de barrido. Es útil donde una función necesita ser realizada solamente al partir el programa.
SP1	Siempre ON	Proporciona un contacto siempre activado para hacer que instrucción se ejecute en cada barrido.
SP2	Siempre OFF	Proporciona un contacto siempre desactivado.
SP3	Pulso en 1 minuto	Activado por 30 segundos y apagado por 30 segundos.
SP4	Pulso en 1 segundo	Activado por 0,5 segundos y apagado por 0,5 segundos.
SP5	Pulso en 100 ms	Activado por 50 ms y apagado por 50 ms.
SP6	Pulso en 50 ms	Activado por 25 ms y apagado por 25 ms
SP7	Barrido alternado	Se activa en un barrido y no en el próximo.

Relevadores de estado de la CPU

SP11	Modo RUN forzado	Activado cuando el conmutador de modo está en la posición RUN y la CPU está funcionando.
SP12	Modo Terminal run	Activado cuando el conmutador de modo está en la posición TERM y la CPU está funcionando.
SP13	Modo Test run	Activado cuando la CPU está funcionando en modo TEST RUN.
SP15	Modo Test stop	Activado cuando la CPU está funcionando en modo TEST STOP.
SP16	Modo Terminal PGM	Activado cuando el conmutador de modo está en la posición TERM y la CPU está en modo Program.
SP17	Parada forzada	Activado cuando el conmutador de modo está en la posición STOP
SP20	Modo de parada forzada	Activado cuando la instrucción STOP fue ejecutada
SP22	Interrupción habilitada	Activado cuando se ha hecho posible las interrupciones usando la instrucción ENI.

Supervisión del sistema

SP36	Override setup	Activado cuando se utiliza la función Override.
SP37	Controlador de barrido	Activado cuando el tiempo de barrido es > que el tiempo prescrito de barrido.
SP40	Error crítico	Activado cuando ocurrió un error crítico tal como pérdida de comunicación de E/S.
SP41	Advertencia (Warning)	Activado cuando ha ocurrido un error no crítico.
SP42	Error de diagnóstico	Activado cuando ocurre un error del diagnóstico o un error de sistema.
SP43	Error de batería	Activado cuando el voltaje de la batería de la CPU está bajo
SP44	Error de memoria de programa	Activado cuando ha ocurrido un error de memoria tal como un error de paridad.
SP45	Error de entradas y salidas	Activado cuando ocurre un error de E/S tal como un fusible quemado.
SP46	Error de comunicaciones	Activado cuando ocurre un error en cualquiera de los puertos de la CPU.
SP50	Instrucción FAULT activada	Activado cuando se ejecuta una instrucción FAULT.
SP51	Watch Dog timeout	Activado si el temporizador del watchdog de la CPU es sobrepasado.
SP52	Error gramatical	Activado si ha ocurrido un error gramatical mientras la CPU está funcionando o si se hace funcionar la verificación sintáctica.V7755 contendrá el código de error exacto.
SP53	Error de resolución de lógica	Activado si la CPU no puede resolver la lógica.
SP54	Error de comunicación	Activado cuando se ejecutan las instrucciones RX, WX con parámetros errados.
SP56	Table instruction overrun	Activado cuando las instrucciones RX o WX, se ejecutan con parámetros incorrectos..

D

Estado del acumulador

SP60	Valor menor que	Activado cuando el valor del acumulador es menos que el valor de la instrucción.
SP61	Valor igual a	Activado cuando el valor del acumulador es igual que el valor de la instrucción.
SP62	Valor mas grande que	Activado cuando el valor del acumulador es mayor que el valor de la instrucción.
SP63	Valor cero	Activado cuando el resultado de la instrucción es cero (en el acumulador).
SP64	Half borrow	Activado cuando la instrucción de resta de 16 bits da lugar a un "pedir prestado".
SP65	Borrow	Activado cuando la instrucción de resta de 32 bits da lugar a un "pedir prestado".
SP66	Half carry	Activado cuando la instrucción de suma de 16 bits da lugar a un "lleva a".
SP67	Carry	Activado cuando la instrucción de suma de 32 bits da lugar a un "lleva a".
SP70	Signo negativo	Activado cuando el valor en el acumulador es negativo.
SP71	Error de Puntero	Activado cuando la memoria V especificada por un puntero (p) es inválida.
SP72	No. de punto flotante	Activado cuando el valor en el acumulador es un número válido de punto flotante.
SP73	Overflow	Activado si ocurre desbordamiento (overflow) en el acumulador cuando una suma o resta con signo da lugar a un bit de signo incorrecto.
SP74	Underflow	Activado en cualquier momento que una operación de punto flotante da lugar a un error de desbordamiento inferior (underflow).
SP75	Error de datos	Activado si se espera un número BCD y se encuentra un número que no es BCD.
SP76	Carga un cero	Activado cuando cualquier instrucción carga un valor cero en el acumulador.

Estado de entradas HSIO

SP100	Activado cuando X0 está encendida	Activado cuando X0 está encendida
SP101	Activado cuando X0 está encendida	Activado cuando X1 está encendida

Relevador especial de salida de pulsos HSIO

SP104	Perfil completado	Activado cuando se termina el perfil de salidas del pulso.(modo 30)
--------------	-------------------	---

Relevador de supervisión de comunicación

SP116	Puerto de la CPU ocupado	Activado cuando el puerto 2 es el maestro y los datos se están enviando.
SP117	Error de comunicaciones del Puerto 2	Activado cuando el puerto 2 es el maestro y hay un error de comunicación.

Relevador de supervisión de comunicaciones de las ranuras del PLC

SP120	Ranura 1 ocupada	Puerto 2 con H0-ECOM y D0-DCM
SP121	Error en la ranura 1	Puerto 2 con H0-ECOM y D0-DCM
SP122	Ranura 2 ocupada	Puerto 2 con H0-ECOM y D0-DCM
SP123	Error en la ranura 2	Puerto 2 con H0-ECOM y D0-DCM
SP124	Ranura 3 ocupada	Puerto 2 con H0-ECOM y D0-DCM
SP125	Error en la ranura 3	Puerto 2 con H0-ECOM y D0-DCM
SP126	Ranura 4 ocupada	Puerto 2 con H0-ECOM y D0-DCM
SP127	Error en la ranura 4	Puerto 2 con H0-ECOM y D0-DCM

Relevadores especiales para módulos opcionales

SP140-237	Ranura 1	Relevador especial para módulo opcional
SP240-337	Ranura 2	Relevador especial para módulo opcional
SP340-437	Ranura 3	Relevador especial para módulo opcional
SP430-537	Ranura 4	Relevador especial para módulo opcional

Relevadores especiales de detección del Contador 1 en el Modo 10

SP540	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3631/3630
SP541	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3633/3632
SP542	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3635/3634
SP543	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3637/3636
SP544	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3641/3640
SP545	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3643/3642
SP546	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3645/3644
SP547	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3647/3646
SP550	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3651/3650
SP551	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3653/3652
SP552	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3655/3654
SP553	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3657/3656
SP554	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3661/3660
SP555	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3663/3662
SP556	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3665/3664
SP557	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3667/3666
SP560	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3671/3670
SP561	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3673/3672
SP562	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3675/3674
SP563	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3677/3676
SP564	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3771/3770
SP565	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3703/3702
SP566	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3705/3704
SP567	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3707/3706

D

Relevadores especiales de detección del Contador 2 en el Modo 10

SP570	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3711/3710
SP571	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3713/3712
SP572	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3715/3714
SP573	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3717/3716
SP574	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3721/3720
SP575	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3723/3722
SP576	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3725/3724
SP577	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3727/3726
SP600	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3731/3730
SP601	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3733/3732
SP602	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3735/3734
SP603	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3737/3736
SP604	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3741/3740
SP605	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3743/3742
SP606	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3745/3744
SP607	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3747/3746
SP610	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3751/3750
SP611	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3753/3752
SP612	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3755/3754
SP613	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3757/3756
SP614	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3761/3760
SP615	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3763/3762
SP616	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3765/3764
SP617	Valor corriente= Valor predefinido	Activado cuando el valor corriente es igual al valor en V3767/3766

Vea también la página 3-14 del manual D0-OPTIONS-M-SP, para más relevadores especiales; éstos están relacionados al módulo F0-04AD-1.

ENTRADAS DE ALTA VELOCIDAD Y SALIDAS DE TREN DE PULSOS (HSIO)



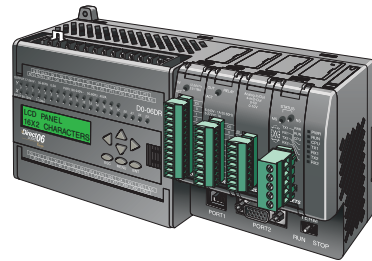
En este capítulo...

Introducción	E-2
Escogiendo el modo de operación HSIO	E-4
Modo 10: Contador de alta velocidad	E-7
Modo 20: Contador incremental y decremental	E-24
Valores prefijados y relevadores especiales	E-27
Modo 30: Salidas de tren de pulsos	E-38
Modo 40: Interrupciones de alta velocidad	E-64
Modo 50: Captura de pulsos de entrada	E-69
Modo 60: Entradas discretas filtradas	E-73

Introducción

Soluciones de control de movimiento embutidas

Muchas aplicaciones de control de máquinas requieren varios tipos de supervisión de alta velocidad y control. Estas aplicaciones usualmente tienen que ver con un tipo de control del movimiento o interrupciones de alta velocidad. El PLC DL06 resuelve este problema, normalmente de costo alto, con funciones ya incluidas en la CPU. Existe también un módulo opcional que permite hacer estas funciones, es el módulo H0-CTRIO, que no es descrito en este libro.



Las funciones disponibles del PLC para alta velocidad son:

- Un contador de alta velocidad, hasta 7 kHz máximo, con hasta 24 valores prefijados de contador y con una subrutina de interrupción con un contador incremental solamente y con vuelta a 0 del valor corriente del contador.
- Entrada de encoder en cuadratura para medir conteo y dirección a favor y en contra de los punteros del reloj (7 kHz máximo), contador incremental o decremental, con vuelta a 0.
- Entradas de interrupción de alta velocidad para respuesta inmediata a tareas críticas o sensibles al tiempo.
- Función de captura de un pulso de corta duración para supervisar un punto de entrada teniendo un ancho del pulso de como mínimo 100 microsegundos.
- Filtrado discreto programable (con hasta 99 ms) para asegurar una señal de entrada integral (este es el modo normal para las entradas X0-X3)

Las funciones de salidas de tren de pulsos son:

- Pulsos de salida programables (10 kHz máximo) con tres tipos de perfil, incluyendo movimiento trapezoidal y control de velocidad

Disponibilidad de las funciones de E/S de alta velocidad (HSIO).

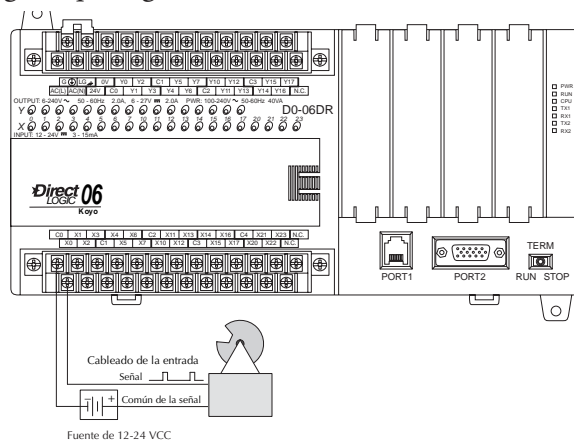
IMPORTANTE: Note las siguientes restricciones en la disponibilidad de las funciones:

- Las entradas de alta velocidad están disponibles solamente en PLCs DL06 con entradas C.C.
- Las salidas de tren de pulsos están disponibles solamente en los PLCs DL06 con salidas en C.C.
- Solamente puede ser usada una función de alta velocidad en un tiempo dado. No se pueden usar entradas y salidas de alta velocidad simultáneamente.

Especificaciones				
Modelo del DL06	Tipo de entrada	Tipo de salida	Salida alta velocidad	Salida de pulsos
DO-06AA	CA	CA	No	No
DO-06AR	CA	Relevador	No	No
DO-06DA	CC	CA	Si	No
DO-06DD1	CC	CC	Si	Si
DO-06DD2	CC	CC	Si	Si
DO-06DR	CC	Relevador	Si	No
DO-06DD1-D	CC	CC	Si	Si
DO-06DD2-D	CC	CC	Si	Si
DO-06DR-D	CC	Relevador	Si	No

Circuitos dedicados de entradas y salidas de alta velocidad

La tarea principal de la CPU es ejecutar el programa y leer y escribir todos los puntos de entradas y salidas durante cada barrido. Para poder ejecutar los eventos de alta velocidad, el DL06 incluye un circuito especial que está dedicado a una porción de las entradas y salidas. Vea al diagrama de la figura que sigue:



El circuito de alta velocidad (HSIO) trabaja con las cuatro primeras entradas (X0-X3) y las primeras dos salidas (Y0-Y1); podemos pensar este circuito como un ayudante de la CPU. En la operación normal (llamada modo 60) el circuito HSIO trabaja normalmente de modo que todas 20 entradas se comportan igualmente y todas las 16 salidas hacen lo mismo.

Cuando la CPU es configurada en cualquier otro modo HSIO, el circuito HSIO impone una función especializada en las entradas y salidas dedicadas para este fin. El circuito HSIO *opera independientemente del barrido de programa* de la CPU.

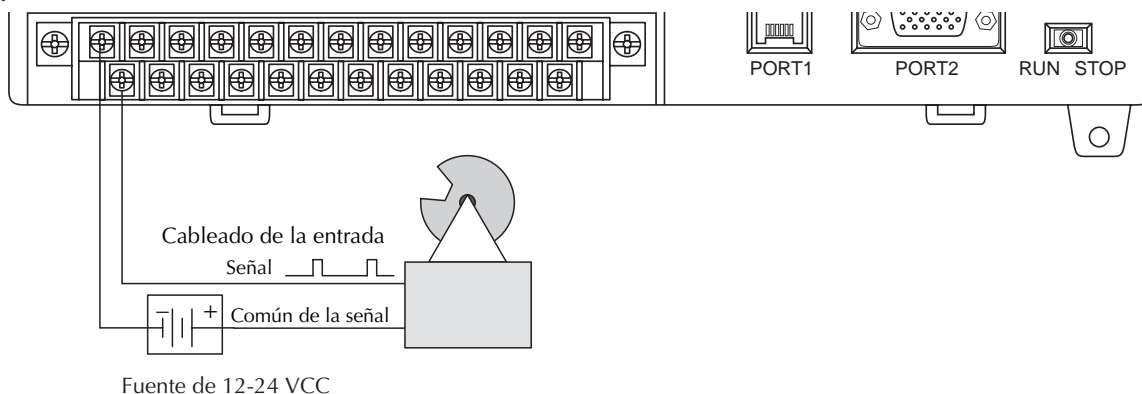
Esto permite una medida precisa y la captura de las entradas de alta velocidad cuando la CPU está ocupada con la ejecución del programa.

Diagramas de cableado para cada modo HSIO

Después de escoger el modo HSIO apropiado para su aplicación, usted debe referirse a la sección en este capítulo para el modo específico.

Cada sección incluye diagramas de cableado para ayudar a conectar los puntos de entrada y salida de alta velocidad correctamente a los aparatos de campo.

Un ejemplo de un contador de alta velocidad de una entrada está mostrado en la figura adjacente.



Selección del modo de operación HSIO

Entendiendo los seis modos

Los circuitos de alta velocidad de entrada y salida operan en uno de los seis modos básicos mostrados en la lista de más abajo. El número en la columna izquierda es el número del modo (más tarde usaremos estos números para configurar el PLC).

Escoja uno de los modos siguientes de acuerdo a la función primaria que usted quiere ejecutar. Usted simplemente puede usar todas las 20 entradas y las 16 salidas como entradas y salidas normales con el modo 60.

Modos básicos de E/S de alta velocidad		
Modo		Características de cada modo
10	Contador de alta velocidad	2 contadores de 7 kHz con 24 valores prefijados y entrada de vuelta a 0, con interrupción con valores prefijados.
20	Contador incremental/decremental	2 contadores de 7 kHz con 24 valores prefijados y entrada de vuelta a 0, con interrupción con valores prefijados.
		Entrada en cuadratura, canal A y B, ambos sentidos
30	Salida de tren de pulsos	Control de motores stepper- señales de pulsos y dirección, perfil programable, pulsos hasta 10 kHz
40	Interrupción de alta velocidad	Genera una interrupción por tiempo o entrada
50	Captura de pulso corto	Captura pulsos de corta duración en una entrada definida
60	Entrada filtrada	Rechaza pulsos de corta duración en entradas definidas

Al escoger uno de los modos de entrada y salidas de alta velocidad, las entradas y salidas listadas en la tabla operan sólo como está definido por la función.

Si una entrada no está definida para trabajar en un modo particular, normalmente va a operar como una entrada filtrada, en forma normal. Similarmente las salidas operarán normalmente a menos que el modo de salidas de tren de pulsos sea seleccionado.

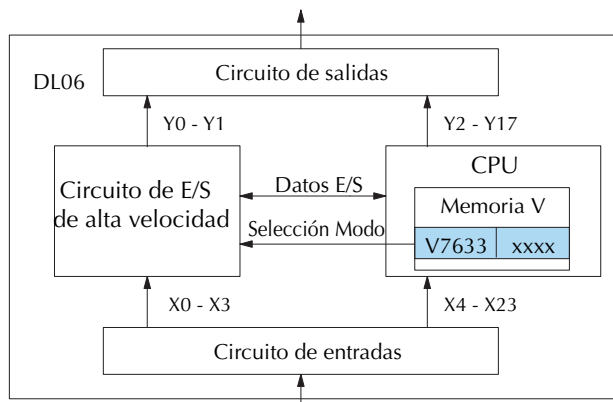
Uso de los puntos de entradas y salidas físicas							
Modo		Puntos de entradas CC				Puntos de salidas CC	
		X0	X1	X2	X3	Y0	Y1
10	Contador de alta velocidad	Contador 1	Contador 2, Interrupción, Entrada de pulsos o entrada filtrada	Reset 1, Interrupción, Entrada de pulsos, Entrada filtrada	Reset 2, Interrupción, Entrada de pulsos, Entrada filtrada	Salida Normal	Salida Normal
20	Contador incremental o decremental	Incrementa	Decrementa	Reset, Entrada de pulsos, entrada filtrada	Entrada de pulsos, entrada filtrada	Salida Normal	Salida Normal
	Contador incremental o decremental (Con encoder en cuadratura)	Entrada de la fase A	Entrada de la fase B				
30	Salida de tren de pulsos	Entrada de pulsos o entrada filtrada	Entrada de pulsos o entrada filtrada	Entrada de pulsos o entrada filtrada	Entrada de pulsos o entrada filtrada	Salida de tren de pulsos o pulsos CW	Salida de dirección o pulsos CCW
40	Interrupción de alta velocidad	Interrupción	Interrupción, Entrada de pulsos o entrada filtrada	Interrupción, Entrada de pulsos o entrada filtrada	Interrupción, Entrada de pulsos o entrada filtrada	Salida Normal	Salida Normal
50	Captura de pulso corto	Entrada de pulsos	Interrupción, Entrada de pulsos o entrada filtrada	Interrupción, Entrada de pulsos o entrada filtrada	Interrupción, Entrada de pulsos o entrada filtrada	Salida Normal	Salida Normal
60	Entrada filtrada	Entrada filtrada	Entrada filtrada	Entrada filtrada	Entrada filtrada	Salida Normal	Salida Normal

Modo normal de fábrica

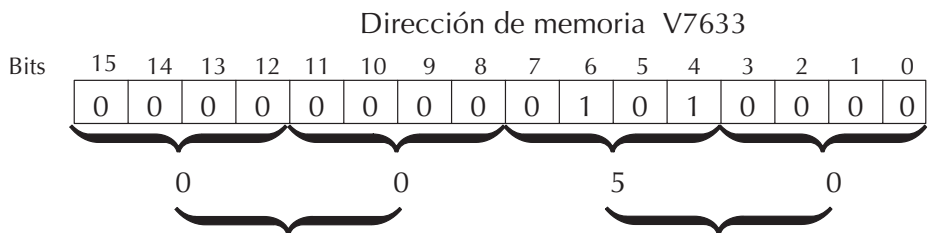
El modo 60 (entradas filtradas) es el modo normal. El DL06 es inicializado en este modo en la fábrica y en cualquier momento en que hace una acción de "Initialize scratchpad". En la condición por defecto, X0 hasta X3 son entradas filtradas (demora de 10 milisegundos) y Y0 hasta Y1 son salidas normales.

Configuración del modo HSIO

Si usted ha escogido un modo de alta velocidad para su aplicación, entonces estará listo para proceder a configurar el PLC para que opere adecuadamente. En el diagrama de bloques mostrado abajo, note el detalle de la memoria en el bloque de la CPU. La dirección de memoria V7633 determina el modo funcional del circuito de alta velocidad. Este es el valor de memoria más importante para las funciones HSIO



El contenido de V7633 es una palabra de 16 bits, y acepta valores como un número BCD. La figura a continuación define cada nivel para entender como es formada la palabra.



Misceláneos (BCD)

- 00 = No es usado (valor original)
- 10 = Activar aviso de batería
- 20 = Energización en modo RUN
- 30 = Aviso de batería y energización en modo RUN activados

Configuración HSIO (BCD)

- 00 = No es usado
- 10 = Contador de alta velocidad
- 20 = Contador incremental decremental
- 30 = Salidas de tren de pulsos
- 40 = Interrupciones de alta velocidad
- 50 = Captura de pulsos
- 60 = Entradas filtradas (valor original)

Los bits 0-7 definen el número del modo (00, 10, 20 hasta 60) a ser colocados en BCD. Por ejemplo el dato "0050" muestra el modo 50 con captura de pulsos habilitado.

Configuración de las entradas X0-X3

Usted también necesita programar las próximas cuatro direcciones además de V7633 en ciertos modos de acuerdo a la función deseada para los puntos X0-X3. Puede ser que otras direcciones pudieran necesitar configuración adicional dependiendo del modo HSIO (Vea la sección correspondiente para el modo particular HSIO).

	Memoria V	
Modo	V7633	xxxx
X0	V7634	xxxx
X1	V7635	xxxx
X2	V7636	xxxx
X3	V7637	xxxx

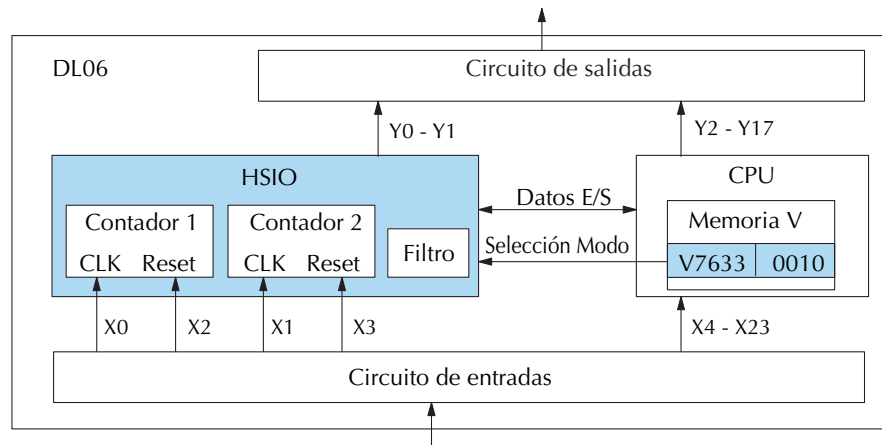
Modo 10: Contador de alta velocidad

Propósito

El circuito HSIO contiene dos contadores de alta velocidad. Un tren de pulsos desde una fuente externa (X0) hace contar el contador en cada transición de 0 a 1. El contador sólo cuenta para arriba, de 0 hasta 9999 9999. El contador compara el valor corriente con hasta 24 valores prefijados que usted defina. El propósito de los valores prefijados es causar una acción e cuando se llegue a puntos específicos, haciéndolo ideal para aplicaciones de, por ejemplo, cortar a la medida. Usa los contadores CT174 y CT176 en la CPU.

Diagrama de bloques funcional:

Vea el diagrama de abajo. Cuando el byte inferior del modo HSIO en la palabra V7633 contiene un número 10 en BCD, el contador de alta velocidad en el circuito HSIO es habilitado. X0 y X1 automáticamente se tornan las entradas de pulso para el contador de alta velocidad, incrementando el contador en cada transición de 0 a 1. La entrada externa del comando de vuelta a 0 del contador en X2 y X3 es el modo normal de configuración para el modo 10.



Usted puede usar X2 y X3 como entradas normales filtradas, en vez de usarlas como entradas de reset dedicadas (que tornan 0) de cada contador. De esta manera el comando de reset del contador debe ser generado en la lógica ladder.

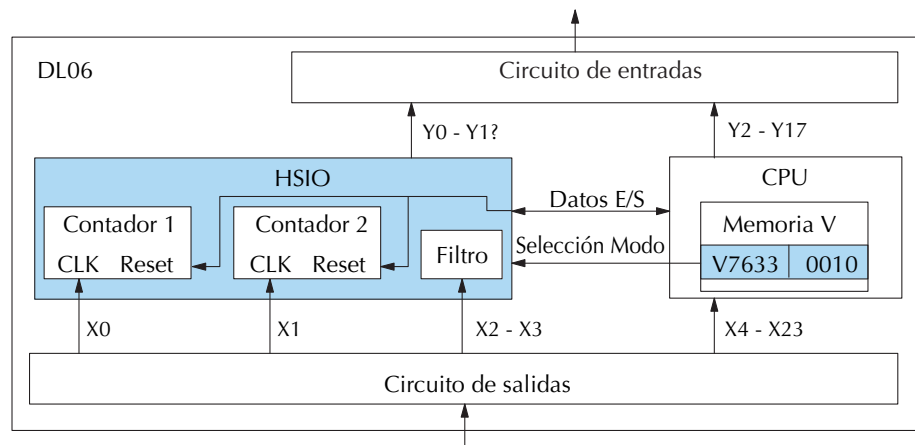
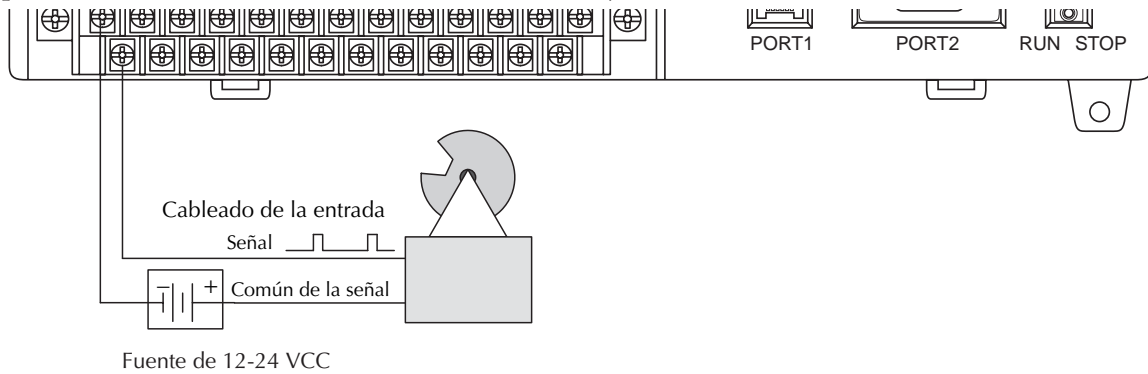


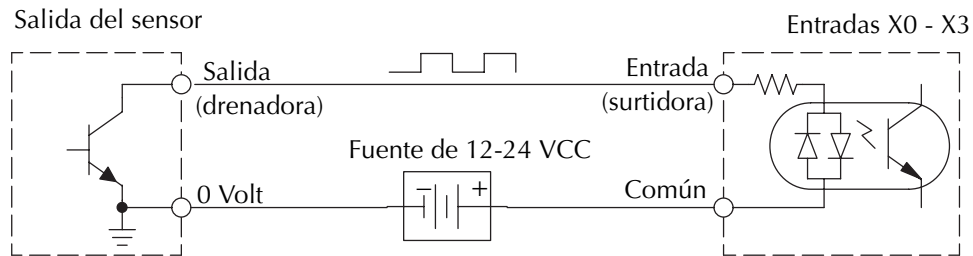
Diagrama de cableado

Se muestra abajo un diagrama general de cableado para contadores o encoders al DL06 en el modo 10 HSIO. Se pueden usar muchos aparatos que generan pulsos como sensor de proximidad, sensores magnéticos u ópticos, etc. Los aparatos con salidas drenadoras (de colector abierto tipo NPN) son probablemente la mejor selección para hacer la interfase. Si el contador es un surtidor a las entradas, debe entregar 12 a 24 volt de corriente continua. Note que los aparatos con salidas de 5 Volt surtidoras no trabajarán con las entradas del PLC DL06.

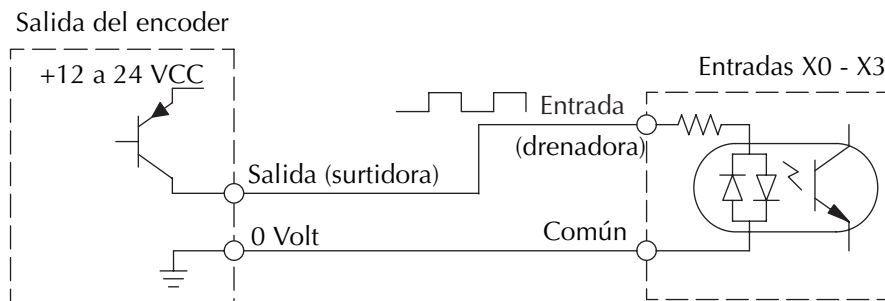


Interconexión a las entradas de conteo

Las entradas de C.C. del DL06 son flexibles en detectar flujo de corriente en cualquier dirección, así que pueden ser conectadas a un sensor con salidas drenadoras o surtidoras. En el circuito siguiente, un contador tiene salidas de transistor de colector abierto, tipo NPN. Drena la corriente del punto de entrada del PLC que surte corriente. La fuente de poder puede ser el FA-24PS u otra fuente (+12VDC o +24VDC), si se cumplen las especificaciones de entrada.

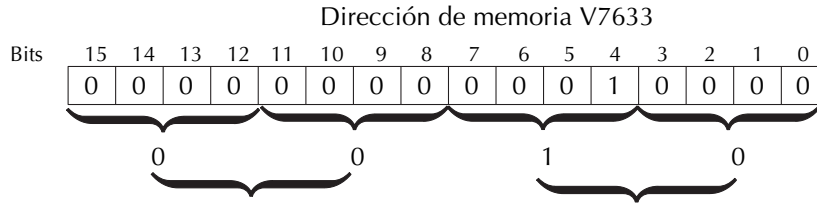


En el diagrama de abajo se tiene un encoder con salidas de transistor de emisor abierto PNP. Surte corriente a la entrada del PLC que drena la corriente a 0 Volt. Ya que el encoder surte corriente, no se requiere ninguna fuente de poder adicional. Sin embargo, observe que la salida del encoder debe ser 12 a 24 Volt (Los encoders de 5V no funcionan).



Configuración del modo 10

La memoria V7633 permite hacer la selección del modo HSIO. Use el BCD 10 en el byte inferior de V7633 para seleccionar este modo.



Vea la página 3-6 para las funciones de los bits 8 a 15

Configuración del modo 10 (BCD)
10 = Contador de alta velocidad

Escoja el método más conveniente para programar V7633 de lo siguiente:

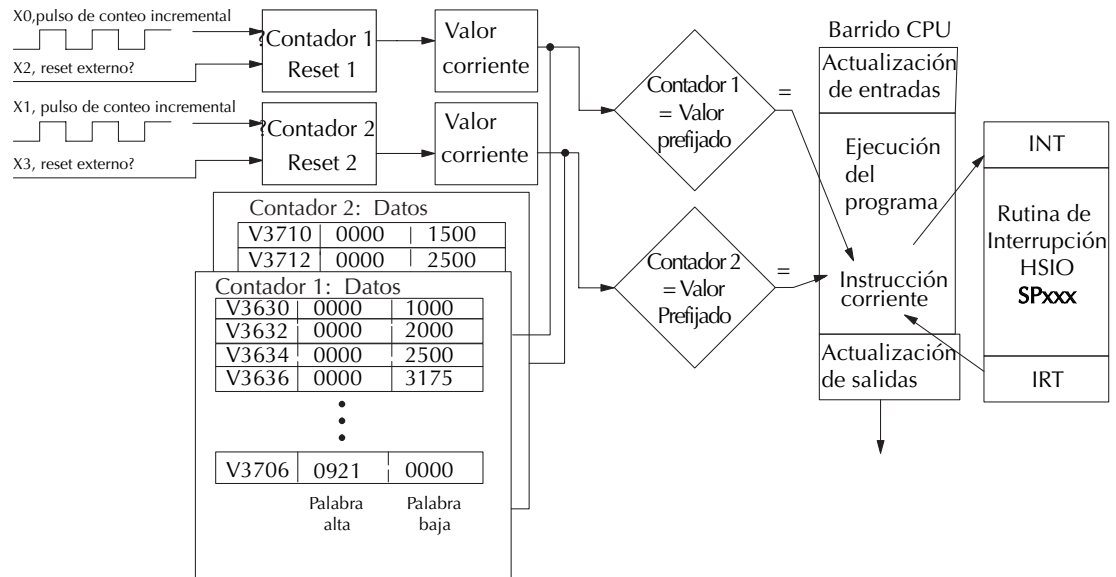
- Incluya instrucciones LD y OUT en su programa ladder.
- Use el programa *DirectSOFT* (Menús Memory Editor o Data View)
- Use el programador D2-HPP

Recomendamos usar el primer método de modo que la configuración HSIO sea parte integral de su programa. Se muestra un ejemplo más adelante en esta sección de como hacer ésto.

Relevadores especiales y valores prefijados.

Los valores prefijados son usados para causar una acción especial cuando el contador llegue al valor prefijado. Vea la figura más abajo. Cada contador tiene 24 valores prefijados que usted puede programar. Los valores prefijados son números de palabra doble (32 bits) de modo que ocupan dos palabras. El usuario selecciona los valores prefijados y el contador continuamente compara el valor corriente del contador con el valor prefijado; cuando los dos son iguales un contacto de relevador especial es activado y la ejecución del programa salta a una rutina de interrupción.

Recomendamos usar relevadores especiales en la rutina de servicio de interrupción para causar la acción inmediata que usted desee. Después que el servicio de interrupción ha sido completado, la CPU retorna al programa prosiguiendo la ejecución del programa desde el punto de interrupción. La función de comparación queda lista para el próximo evento de valor prefijado.



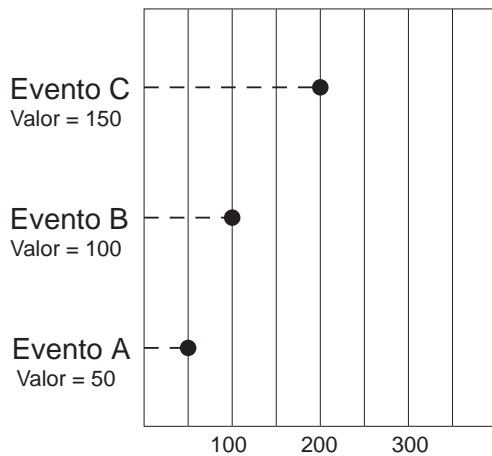
Valores prefijados incrementales y absolutos.

Hay 2 modos de valor prefijado disponibles; uno es absoluto y el otro es incremental. Los valores prefijados se colocan en una tabla de memorias contiguas.

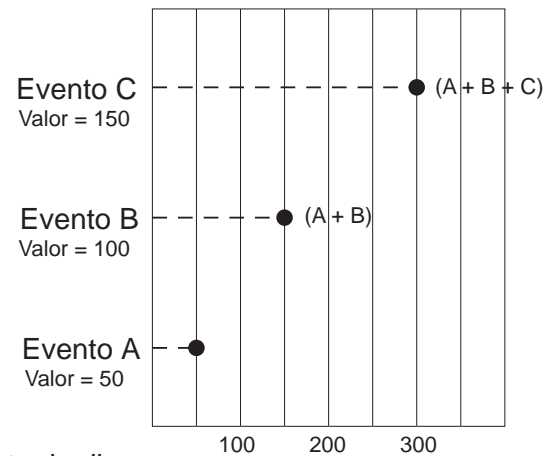
En el modo absoluto cada valor prefijado es tratado como valor total. En el modo incremental los valores prefijados son acumulados.

Los valores prefijados incrementales representan el número de conteo de pulsos entre eventos.

Valores prefijados absolutos



Valores prefijados incrementales



● = Punto de disparo

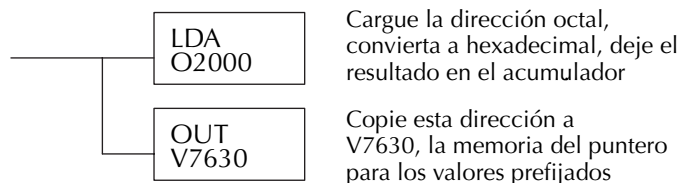
En el ejemplo de la figura, los valores prefijados, se definen en 50, en 100 y en 150 pulsos.

La diferencia entre los modos incremental y absoluto es que los valores prefijados absolutos disparan eventos cuando llegan al valor prefijado 50, 100 y 150 pulsos.

Los valores prefijados incrementales generan eventos cuando los valores llegan al valor prefijado 50 para el primer evento, a 150, que es el resultado de la suma del primer valor prefijado y el segundo, y otro evento cuando llega a 300, que es el resultado de la suma del primer, segundo y tercer valor prefijado..

Dirección inicial de los datos de valores prefijados

La memoria V7630 es el puntero de la dirección de memoria que contiene el inicio de la tabla de datos de valores prefijados. La dirección de partida normal para tabla de valor prefijados es de V3630 (normal después de hacer la initialize Scratchpad). Sin embargo usted puede cambiar esto programando un valor diferente en el valor V7630. Use las instrucciones que LDA y OUT como mostrado abajo.



Puntero de tabla de valores prefijados

V7630	2000
-------	------

Tabla de valores

V2001	V2000	0000	1000
V2003	V2002	0000	2000
V2005	V2004	0000	2500
V2007	V2006	0000	3175

⋮

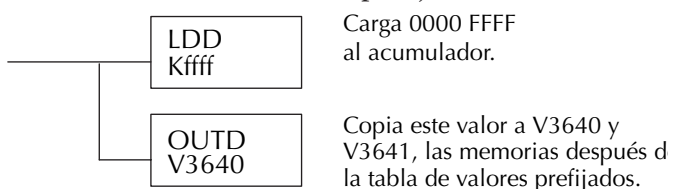
V2077	V2076	0000	0000
-------	-------	------	------

Usando menos de 24 valores prefijados

Cuando se usan todos los puntos de valor prefijados, la CPU sabe automáticamente cuando se llega al final de la tabla. Sin embargo, cuando se usan menos de 24 valores prefijados, es necesario indicar a la CPU que ha llegado al último valor prefijado. La forma de indicar el fin de los valores prefijados es colocar uno de los siguientes códigos de fin de tabla en la próxima dirección de memoria:

Código de fin de tabla	Modo aplicable	Significado
0000 FFFF	Absoluto e Incremental	Indica el fin de la tabla de valores prefijados
0000 00FF	Incremental	Señala el fin de valores prefijados y los reinicia. No vuelve a 0 el conteo acumulado de pulsos de CT174 o CT176.
0000 FF00	Incremental	Señala el fin de valores prefijados, los parte nuevamente y vuelve a 0 los pulsos contados acumulados de CT174 o CT176.

Como es mostrado en la tabla de arriba, cada una de las señales de fin de tabla tiene un significado diferente. Use la instrucción LDD Kffff para indicar el código de fin de tabla en el próximo par de memorias en la tabla de valores prefijados. En el ejemplo, se usan cuatro valores prefijados. El valor 0000 FFFF en la memoria V3641 y V3640 indican que el valor prefijado anterior fue el último valor prefijado.



Ejemplo de tabla original de valores prefijados

V3631	V3630	0000	1000
V3633	V3632	0000	2000
V3635	V3634	0000	2500
V3637	V3636	0000	3175
V3641	V3640	0000	FFFF

En el modo absoluto, el contador y el total acumulado son vueltos a 0 cada vez que es alcanzado un valor prefijado. En modo incremental, se puede escoger no volver a 0 el contador o el total acumulado o se puede escoger volver a 0 solamente el contador o se puede volver el contador y el total acumulado a 0 cuando es leído el código de fin de tabla. En el ejemplo FFFF ha sido colocado en V3640 ya que el último valor prefijado estaba en V3636 y estábamos usando menos de 24 valores prefijados.



NOTA: En modo absoluto cada valor prefijado debe ser mayor que el valor previo. Si un valor prefijado es menor que un anterior, la CPU no puede comparar aquel valor ya que el contador sólo cuenta hacia arriba.

Relevadores especiales de comparación

La tabla siguiente muestra una lista de todas las direcciones de memoria de valor prefijado para cada contador de alta velocidad. Cada uno ocupa dos memorias de 16 bits. El numero del relevador especial se muestra en la columna adyacente.

Podemos llamar a esos relevadores contactos de "igualdad" porque ellos sólo son verdaderos (se cierran) cuando el valor corriente del contador de alta velocidad es igual al valor prefijado y permanece cerrado. Cada contacto permanece cerrado hasta que el contador llega ser igual al próximo valor prefijado.

Tabla de memoria de valores prefijados

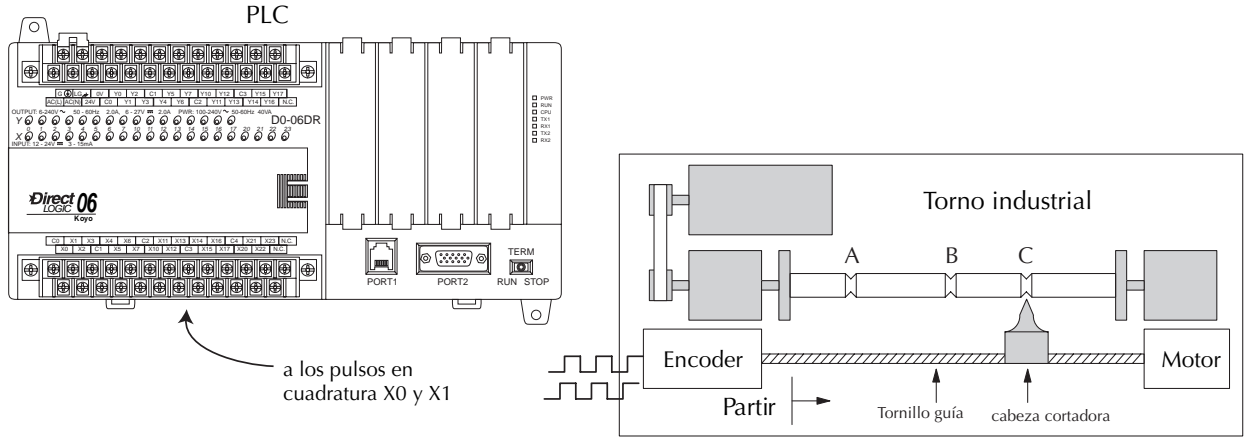
Valor del contador 1	Memoria del valor prefijado	Memoria del valor prefijado	Valor del contador 2	Memoria del valor prefijado	Memoria del valor prefijado
1	V3631 / V3630	SP540	1	V3711/V3710	SP570
2	V3633 / V3632	SP541	2	V3713/V3712	SP571
3	V3635 / V3634	SP542	3	V3715/V3714	SP572
4	V3637 / V3636	SP543	4	V3717/V3716	SP573
5	V3641 / V3640	SP544	5	V3721/V3720	SP574
6	V3643 / V3642	SP545	6	V3723/V3722	SP575
7	V3645 / V3644	SP546	7	V3725/V3724	SP576
8	V3647 / V3646	SP547	8	V3727/V3726	SP577
9	V3651 / V3650	SP550	9	V3731/V3730	SP600
10	V3653 / V3652	SP551	10	V3733/V3732	SP601
11	V3655 / V3654	SP552	11	V3735/V3734	SP602
12	V3657 / V3656	SP553	12	V3737/V3736	SP603
13	V3661 / V3660	SP554	13	V3741/V3740	SP604
14	V3663 / V3662	SP555	14	V3743/V3742	SP605
15	V3665 / V3664	SP556	15	V3745/V3744	SP606
16	V3667 / V3666	SP557	16	V3747/V3746	SP607
17	V3671 / V3670	SP560	17	V3751/V3750	SP610
18	V3673 / V3672	SP561	18	V3753/V3752	SP611
19	V3675 / V3674	SP562	19	V3755/V3754	SP612
20	V3677 / V3676	SP563	20	V3757/V3756	SP613
21	V3701 / V3700	SP564	21	V3761/V3760	SP614
22	V3703 / V3702	SP565	22	V3763/V3762	SP615
23	V3705 / V3704	SP566	23	V3765/V3764	SP616
24	V3707 / V3706	SP567	24	V3767/V3766	SP617

Las direcciones consecutivas mostradas arriba para cada relevador es asignada por la CPU como direcciones por defecto. El puntero para el inicio de esas direcciones es almacenado por CPU en la memoria V7630. Para el segundo contador, use V7631 como el puntero para la tabla de valores prefijados. Aún existen solamente 24 valores prefijados. Si usted tiene un conflicto de direcciones porque ya fue hecho un programa que ocupa estas direcciones, usted puede cambiar la tabla de direcciones normal solamente haciendo que su programa tome un valor inicial diferente en V7630 para el contador 1 o en V7631 para el contador 2. Para cambiar la localización de la tabla use las instrucciones LDA y OUT como mostrado anteriormente.

Cálculo de valores prefijados

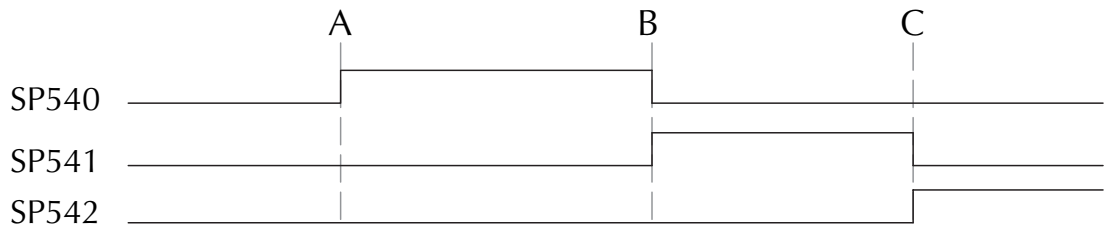
Los valores prefijados ocupan dos palabras cada uno. Ellos pueden estar en el rango de -8388608 hasta +8388607, de la misma forma como es el rango del valor del contador de alta velocidad. Todos los 24 valores son valores absolutos, esto es, cada uno es un desvío del valor 0 del contador.

En el torno industrial del diagrama, el PLC supervisa la posición del tornillo de guía contando los pulsos A, B y C a lo largo del viaje lineal y la cabeza de corte empuja el material de trabajo y hace una ranura.



El diagrama a continuación muestra la duración de cada cierre del contacto de "igualdad", que es el relevador especial SP40, SP41 o SP42. Cada contacto permanece cerrado hasta que el próximo cierre. Todos se abren cuando el valor corriente del contador vuelve a cero.

Relevadores especiales de igualdad



NOTA: Cada valor prefijado sucesivo debe ser dos números mayor que el valor prefijado anterior. En el ejemplo del torno industrial, B debe ser mayor que $A + 2$ y C mayor que $B + 2$.

Configuración de las entradas X.

Las opciones de entradas discretas para el modo contador de alta velocidad están mostradas en la tabla inferior. La entrada X0 es dedicada a la primera entrada del contador. La entrada X1 puede ser el pulso para el segundo contador o para una entrada filtrada. La sección en el modo 60 al final de este capítulo describe la programación de las constantes de tiempo del filtro. Las entradas X2 y X3 puede ser configuradas como una entrada que vuelve a cero el contador, con o sin la opción de interrupción. La opción de interrupción permite que las entradas X2 y X3 causen una interrupción de la misma forma que los valores prefijados hacen pero no hay un contacto de relevador especial sino que las entradas X2 y X3 serán verdaderas durante la rutina de interrupción, por un barrido o finalmente, X2 y X3 puede ser dejados simplemente como entradas filtradas.

Opciones de entradas			
Entrada	Memoria configuración	Función	Código hexadecimal
X0	V7634	Pulso del contador #1	0001 (absoluto) (original)
			0101 (incremental)
X1	V7635	Pulso del contador #2	0001 (absoluto) (original)
			0101 (incremental)
		Interrupción	0004
		Entrada de pulsos	0005
		Entrada filtrada	xx06, xx = tiempo de filtrado 0 - 99 ms (BCD)
X2	V7636	Reset del contador #1 (sin interrupción)	0007* (original) 0207*
		Reset del contador #1 (con interrupción)	0107* 0307*
		Interrupción	0004
		Entrada de pulsos	0005
		Entrada filtrada	xx06, xx= tiempo de filtrado 0 - 99 ms (BCD)
X3	V7637	Reset del contador #2 (sin interrupción)	0007* (original) 0207*
		Reset del contador #2 (con interrupción)	0107* 0307*
		Interrupción	0004
		Entrada de pulsos	0005
		Entrada filtrada	xx06, xx= tiempo de filtrado 0 - 99 ms (BCD)

* Con el reset (valor de vuelta a 0) del contador usted tiene la opción de una vuelta a 0 normal o una vuelta a 0 rápida. Sin embargo la vuelta a 0 rápida no reconoce valores prefijados cambiados durante la ejecución de programa.

Cuando se configura el número 0007 o 0107 en V7636 o en V7637 y se cambian los valores prefijados durante la ejecución del programa, el PLC DL06 reconoce los valores prefijados cambiados durante el tiempo de reset.

Cuando el número se configura 0207 o 0307 en V7636 o V7637 la CPU no verifica por un cambio en el valor prefijado de modo que el PLC DL06 tiene un tiempo de vuelta a cero más rápido.

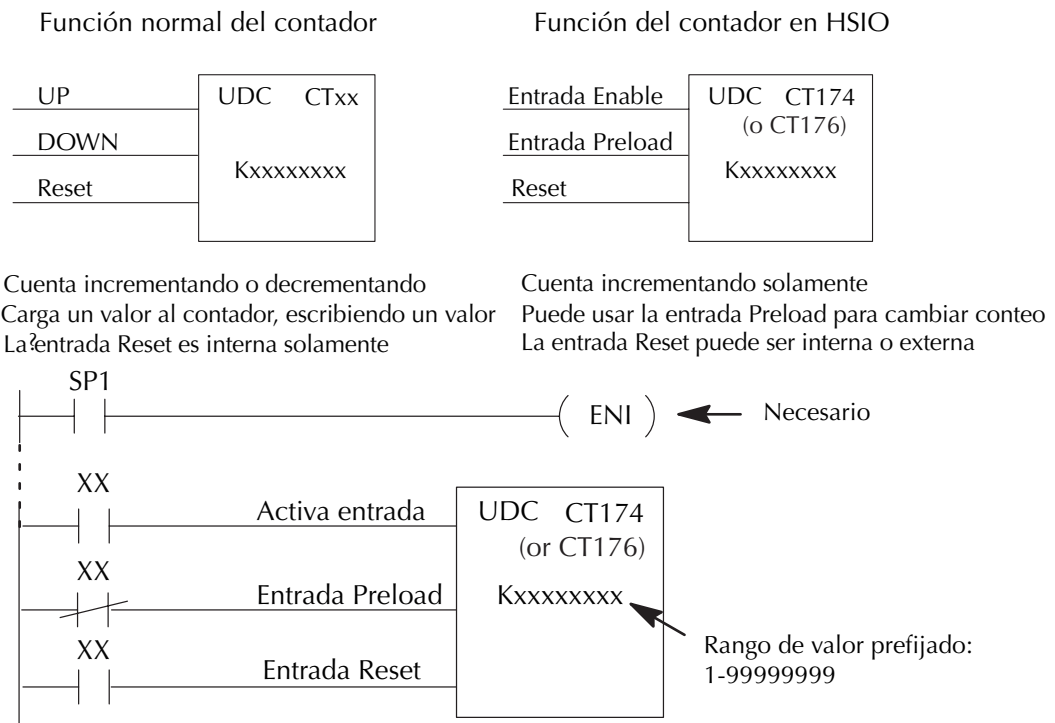
Escribiendo el programa de control.

El símbolo para la instrucción de contador es UDC (contador incremental-decremental).

El DL06 puede tener hasta 128 contadores, apodados CT0 hasta CT177. El contador de alta velocidad en el circuito HSIO es accesado en logica ladder usando CT174 y CT176. Usa sólo las memorias del contador CT174 hasta CT177 exclusivamente cuando el modo HSIO es activo (de otra forma, CT174 hasta CT177 están disponibles para uso como contador normal).

El contador HSIO necesita dos memorias porque es un contador de doble palabra. Tiene tres entradas como mostrado. La primera entrada ENABLE (habilitar) permite contar cuando está activa. El punto medio es usado para transferir un nuevo valor corriente en el contador (preload input). La última entrada es el comando de vuelta 0. La entrada del medio debe ser falsa cuando el contador está contando.

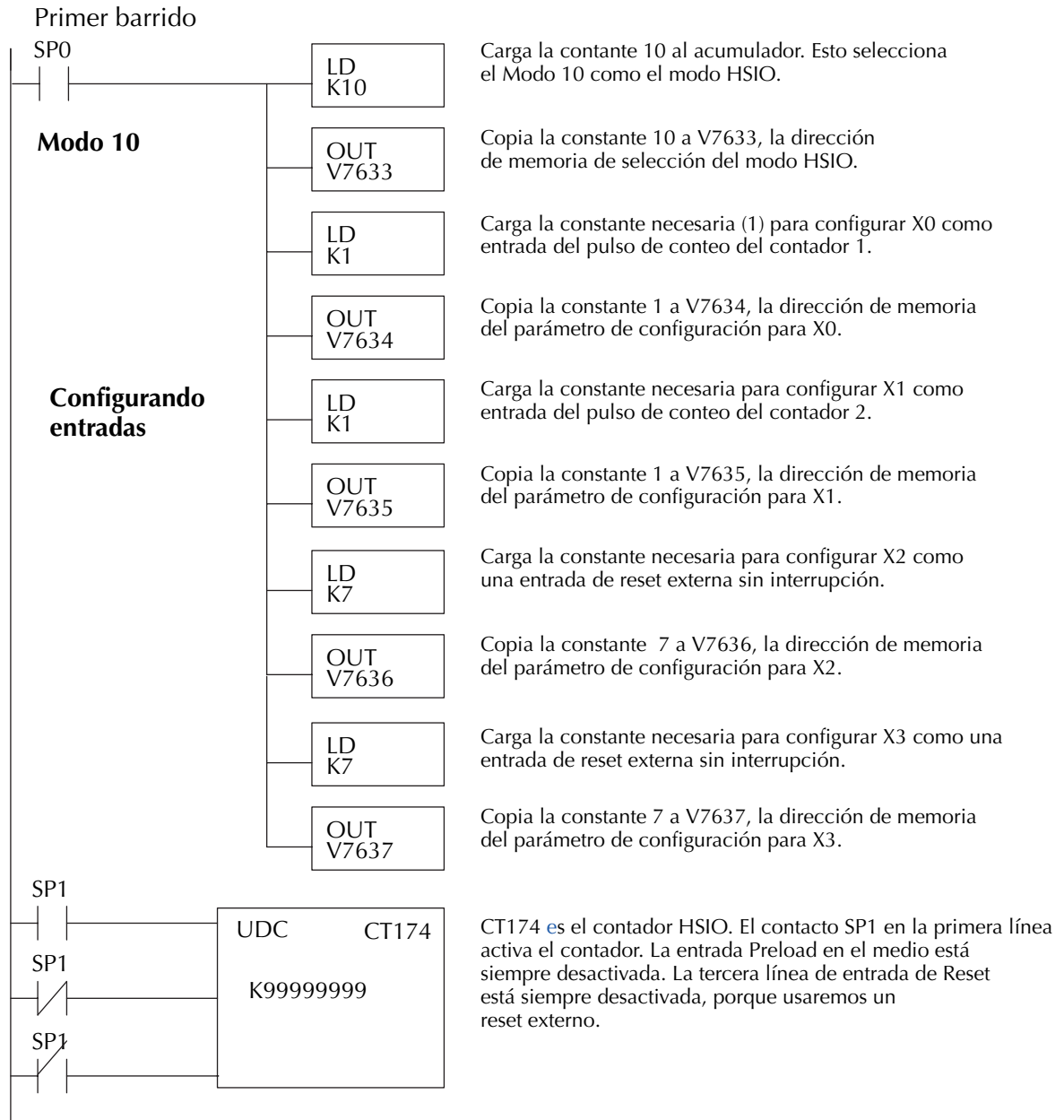
La próxima figura muestra como el contador HSIO aparecerá en un programa con diagrama ladder. Note que la instrucción ENI debe ser ejecutada antes que el contador llegue al primer valor prefijado. Hacemos esto normalmente durante la energización usando el relevador SP0. Cuando se usa el contador pero no los valores prefijados y la interrupción podemos omitir la instrucción ENI.



Quando la entrada Enable (de habilitación) es ON, el contador incremental-decremental CT174 responderá a pulsos en X0 y incrementará ese valor. El contador CT176 responderá los pulsos en X1 e incrementará ese valor. El contacto de vuelta a cero se comporta de una forma lógica OR con la entrada física. X2 (cuando seleccionado) vuelve a cero el contador 1. La entrada X3 (cuando seleccionada) vuelve a cero el contador 2, de modo que el contador de alta velocidad puede recibir un comando de vuelta a 0 de cualquiera de los contactos del renglón en el diagrama ladder o puede recibir un comando externo de vuelta a 0 de X2 o X3, si ha configurado X2 o X3 como un comando externo.

Ejemplo 1 modo 10. Contador sin valor prefijado

El siguiente ejemplo es la forma mas simple de usar los contadores de alta velocidad que no usan valores prefijados y relevadores especiales en la rutina de interrupción. El programa configura el circuito HSIO para operación en modo 10 de modo que X0 es automáticamente la entrada del contador para el primer contador y X1 es la entrada del contador para el segundo contador. Use la instrucción compare-double (CMPD) para causar una acción a un cierto valor de conteo. Note que esto permite tener más de 24 valores prefijados. También se configura X2 y X3 para ser un comando externo de vuelta a cero del contador.

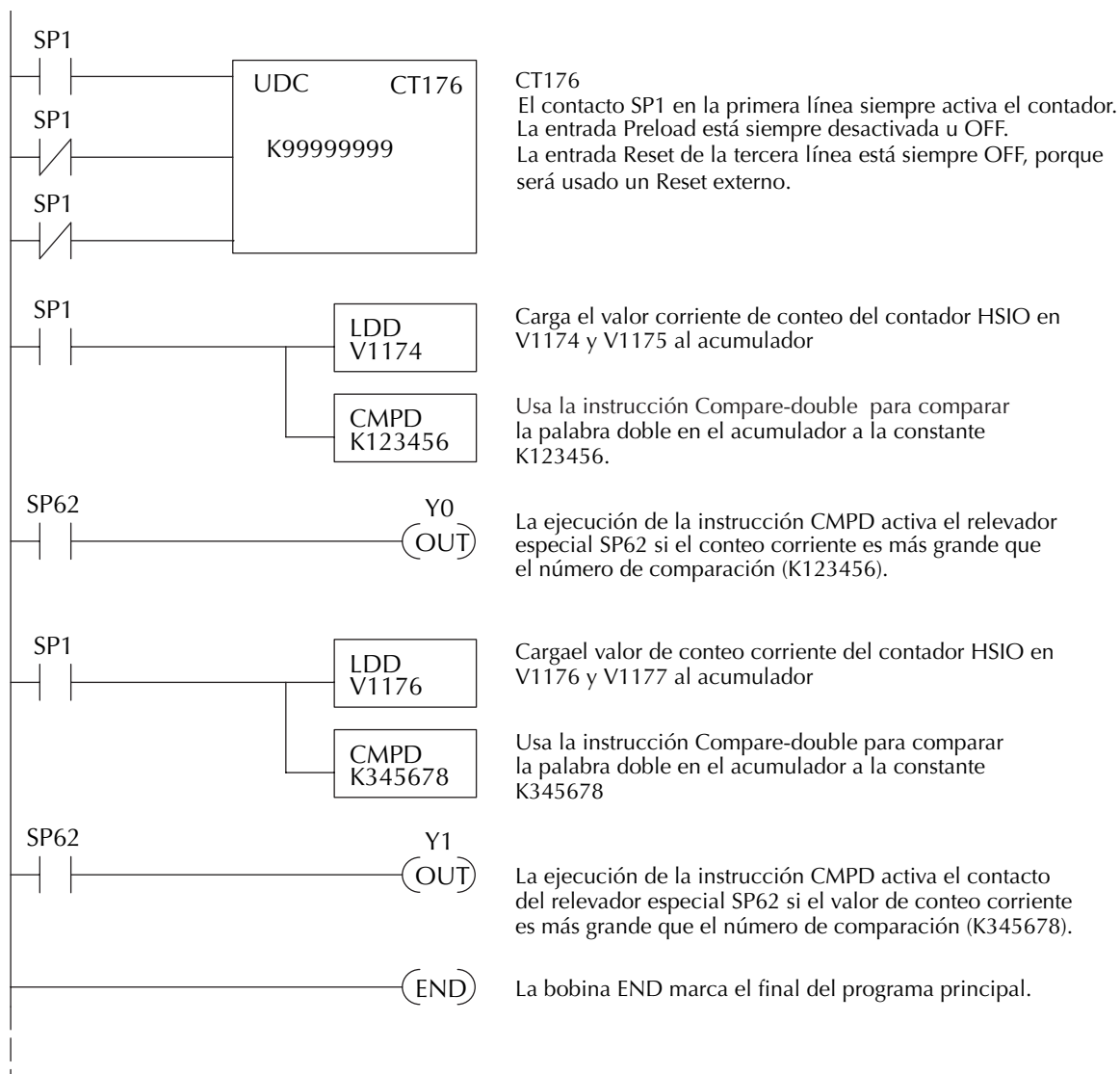


continúa en la próxima página

Programa ejemplo continuado

La instrucción de comparación doble en el código de abajo usa el valor corriente del contador HSIO para hacer verdadero Y1. Esta técnica se puede usar para hacer más de 24 comparaciones pero es dependiente del tiempo de barrido. Sin embargo use los 24 valores prefijados con la rutina de interrupción si su aplicación necesita un tiempo de respuesta rápido como mostrado en el próximo ejemplo.

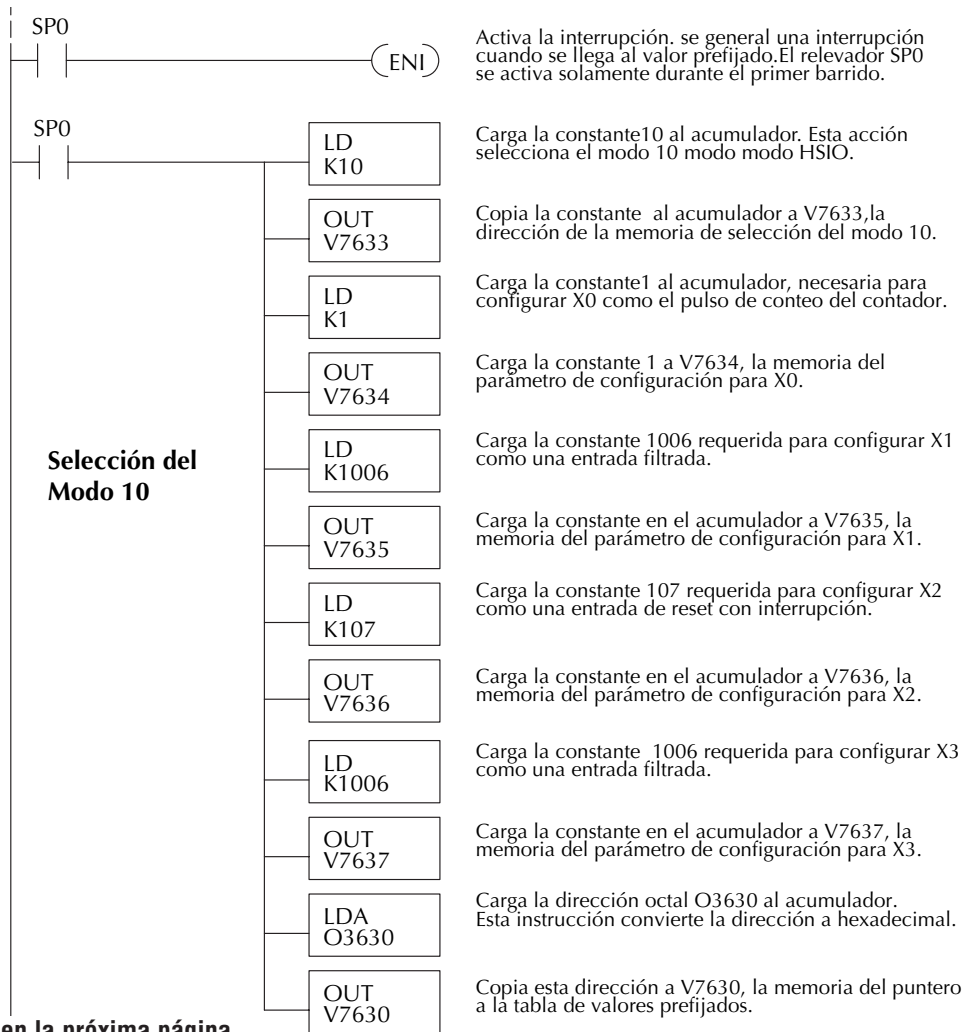
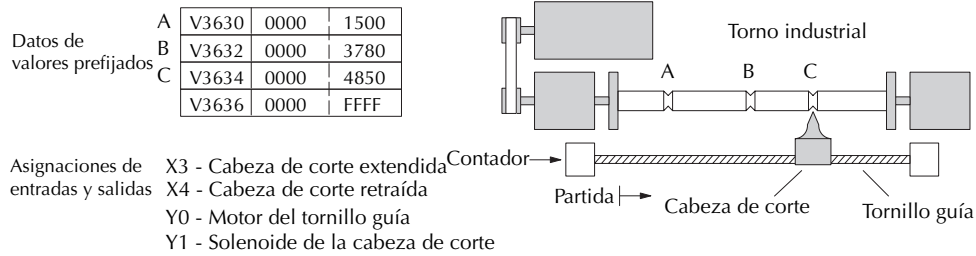
continúa desde la página anterior



Ejemplo 2 del modo 10: Contador con valores prefijados.

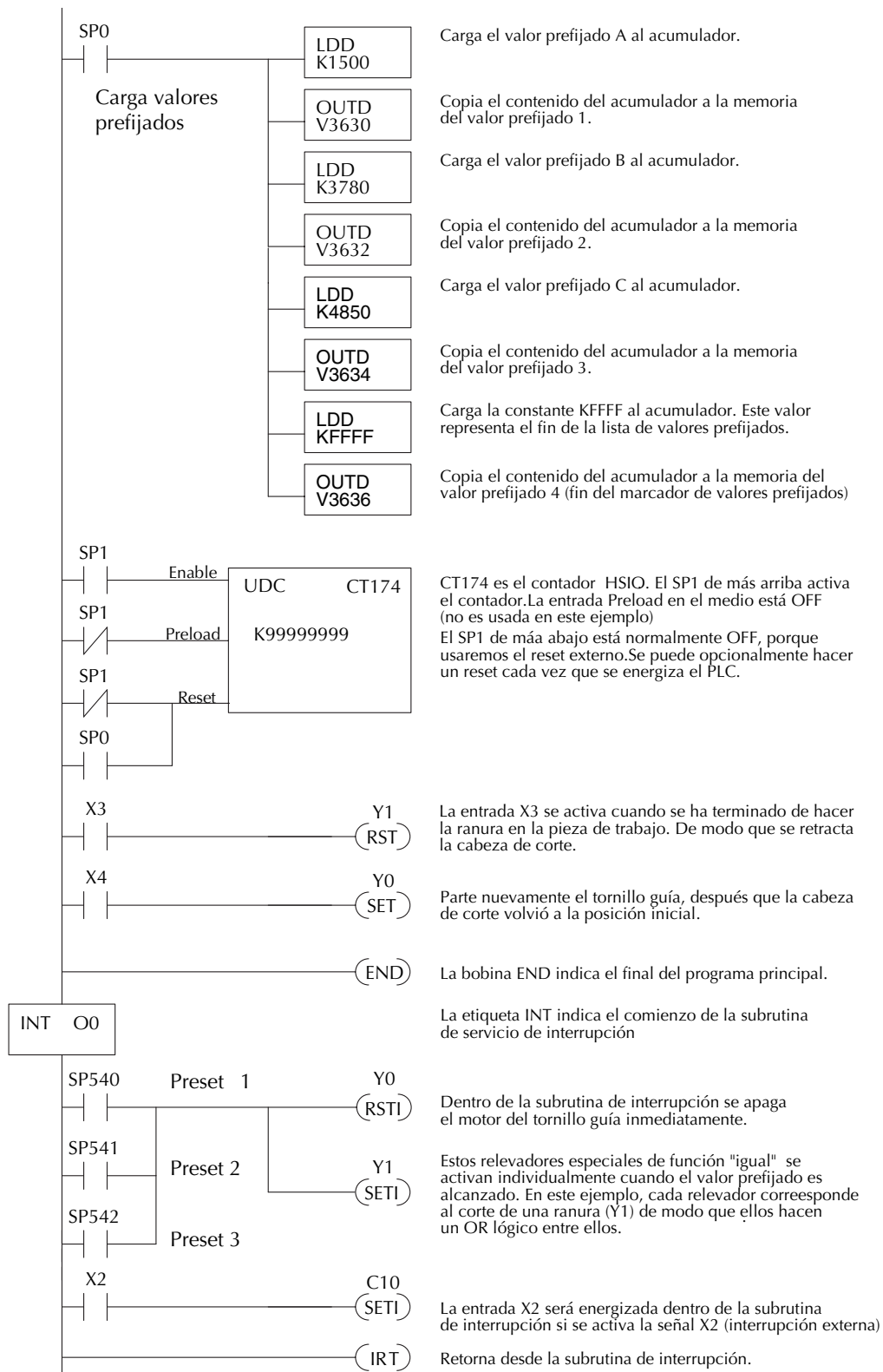
El siguiente ejemplo muestra como programar un circuito HSIO para disparar tres valores prefijados. Usted tal vez se recuerde del ejemplo del torno industrial al comienzo de este capítulo. Este ejemplo muestra como controlar la cabeza cortadora del torno para hacer tres ranuras en la pieza en que se trabaja en posiciones bien precisas. Cuando el tornillo guía gira, el contador genera pulsos que el DL06 puede contar.

Los tres valores prefijados A, B y C representan las posiciones (el número de pulsos) que corresponden a cada una de las tres ranuras. En este ejemplo es usado sólo un contador . El segundo contador puede ser usado de la misma manera.



continúa en la próxima página

continuado desde la página anterior



E

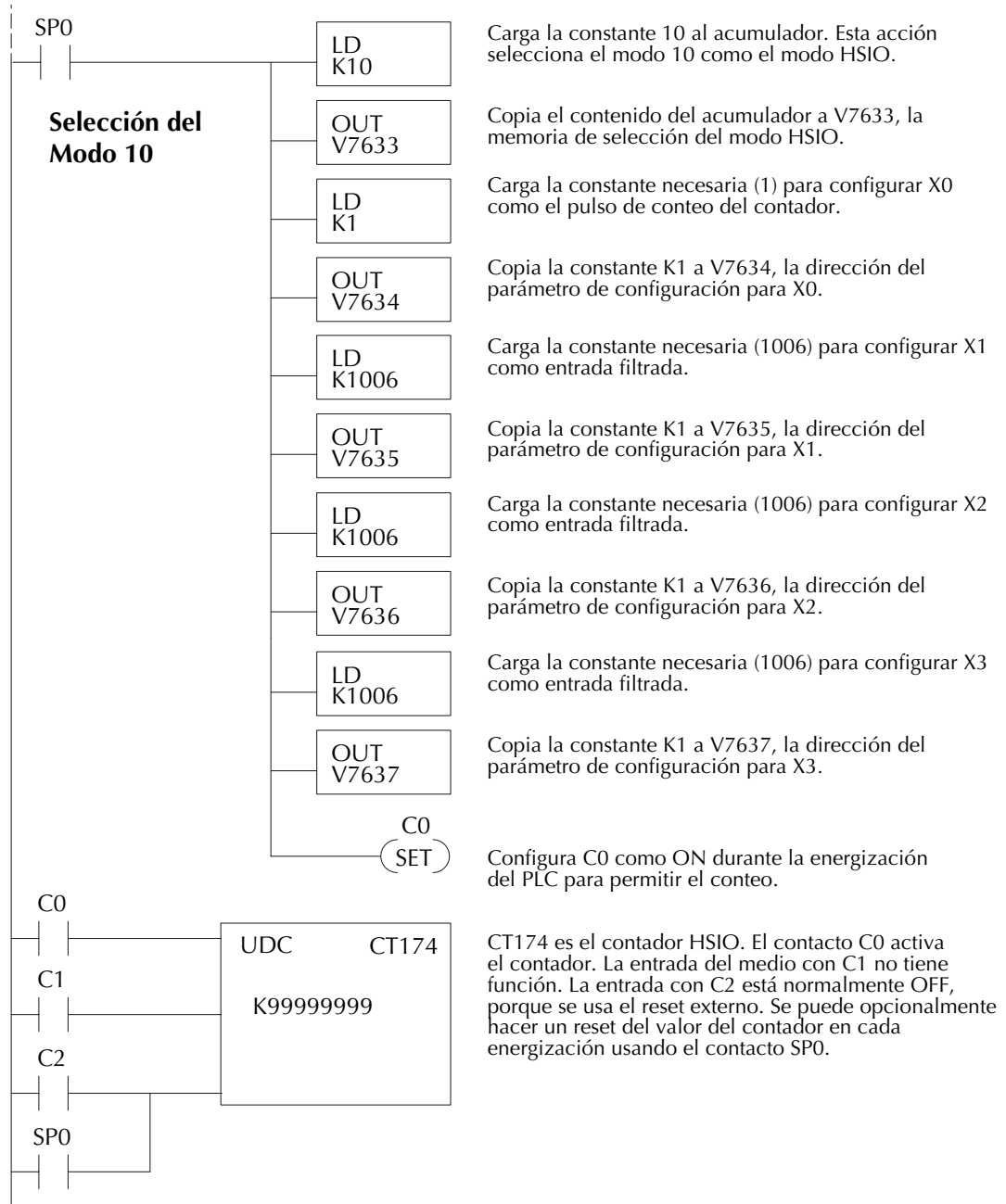
Algunas aplicaciones requieren un tipo diferente de acción indicada para cada valor prefijado. Es posible distinguir en la rutina de interrupción un valor prefijado de otro, al hacer verdadero una salida única para cada contacto de comparación con el relevador especial SPxx. Podemos determinar la fuente de la interrupción examinando los contactos individualmente así como también en X2. El contacto X2 será verdadero (dentro de la rutina de interrupción solamente) si la interrupción fue causada por el comando externo de vuelta a 0 con la entrada X2.

E

El resto de la página fue dejado en blanco intencionalmente

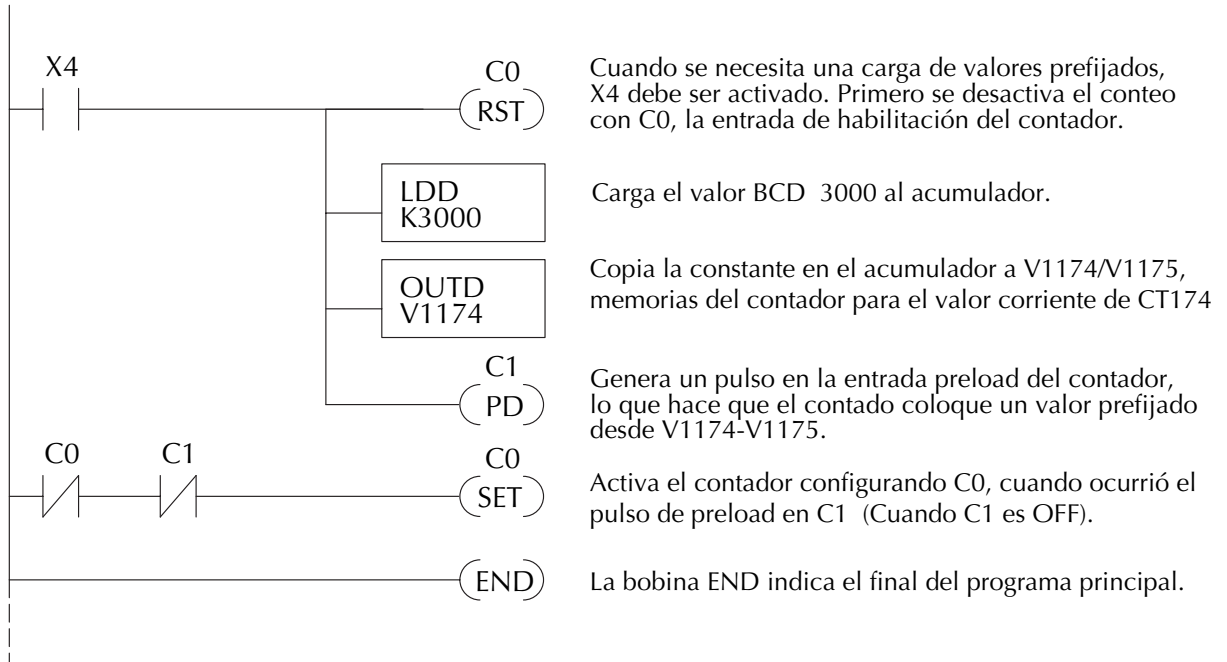
Ejemplo 3 modo 10: contador con valor inicial.

El siguiente ejemplo muestra como se puede cargar el valor de conteo con otro valor. Cuando la entrada de comando preload (X4 en este ejemplo) es energizado, deshabilitamos el contador de contar con C0. Cuando escribimos el valor K3000 a la memoria del contador (V1076-V1077) cargamos el valor corriente del contador con K3000. Cuando el comando Preload X4 es falso, el contador continúa contando cualquier pulso pero ahora partiendo desde el valor 3000. En este ejemplo sólo un contador de alta velocidad es usado. El segundo contador puede ser usado de la misma forma.



continúa en la próxima página

continuado desde la página anterior



Cuando se necesita una carga de valores prefijados, X4 debe ser activado. Primero se desactiva el conteo con C0, la entrada de habilitación del contador.

Carga el valor BCD 3000 al acumulador.

Copia la constante en el acumulador a V1174/V1175, memorias del contador para el valor corriente de CT174

Genera un pulso en la entrada preload del contador, lo que hace que el contado coloque un valor prefijado desde V1174-V1175.

Activa el contador configurando C0, cuando ocurrió el pulso de preload en C1 (Cuando C1 es OFF).

La bobina END indica el final del programa principal.

Búsqueda de problemas en el modo 10.

Si está teniendo problemas con la operación del modo 10 por favor estudie los siguientes síntomas y las causas posibles. Los problemas comunes están listados a continuación:

Síntomas: el contador no cuenta.

Causas posibles:

1. **Sensor y cableado** – Verifique que el encoder, sensor de proximidad o generador de pulsos realmente enciende el LED de estado en X0 (contador 1) y X1 (contador 2). El problema puede ser debido a un problema de cableado por ser circuito drenador o surtidor. Verifique la conexión de señal a tierra. También verifique que la duración del pulso es suficientemente larga para que el PLC lo pueda reconocer.
2. **Configuración** – Use la ventana **Data View** para verificar los parámetros de configuración. V7633 debe ser configurado con el valor 10 y V7634 debe ser colocado 1 o 101 para activar el primer contador de alta velocidad. V7635 debe ser configurado como 1 o 101 para habilitar el segundo contador de alta velocidad.
3. **Parado en cero sin contar**– Verifique el estado de la entrada del comando de vuelta a 0, X2 y X3. Si X2 es verdadero, el contador no cuenta por qué está siendo forzado a estar en cero.
4. **Programa ladder** – Asegúrese de que esté usando el contador CT174 y CT176 en su programa. La entrada superior es la señal de habilitación para el contador. Debe ser verdadera antes que el contador cuente. La entrada del medio es una entrada sin uso. La entrada inferior es el comando de vuelta a 0 y debe estar falso durante el conteo.

Síntoma: el contador cuenta pero los valores prefijados no funcionan

Causas posibles :

1. **Configuración** – Asegúrese que el valor prefijado es correcto. Los valores prefijados son valores de 32 bits, que tienen un rango de 0 hasta 9999 9999. Asegúrese que usted escribe todos los 32 bits a la localización reservada usando las instrucciones LDD y OUTD. Use solamente direcciones pares, V3630 hasta V3767. Si usa menos de cuatro valores prefijados, asegúrese de colocar 0000FFFF, 0000FF00, o 000000F en la localización después del último valor prefijado usado.
2. **Rutina de interrupción** – Sólo use la interrupción No. 0. Asegúrese que es la interrupción ha sido habilitada ejecutando la instrucción ENI antes de hacer la interrupción. La rutina de interrupción debe ser colocada después del programa principal, usando la etiqueta INT y terminando con un retorno de interrupción IRT.
3. **Relevadores especiales**– Verifique los números de relevadores especiales en su programa. Use SP540 para el valor prefijado 1, SP541 para el valor prefijado 2 y así sucesivamente. Recuerde que se usa sólo un contacto cerrado del relevador especial cada vez. Cuando el valor del contador llega al próximo valor prefijado, el contacto del relevador especial que estaba cerrado ahora se abre y el próximo se cierra.

Síntomas: el contador cuenta pero no vuelve a 0.

Causas posibles :

1. Verifique el estado del indicador LED de X2 (contador 1) y X3 (contador 2) para asegurarse que está activo cuando usted necesite un comando de vuelta a 0. O si usted está usando un comando de vuelta a 0 interno, use *Directsoft* para supervisar la entrada de vuelta a 0 al contador.

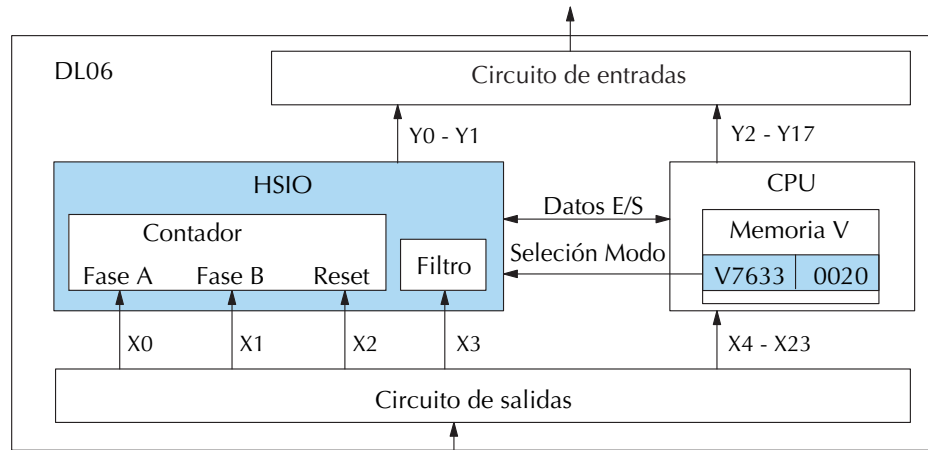
Modo 20: Contador incremental y decremental.

Propósito del modo 20

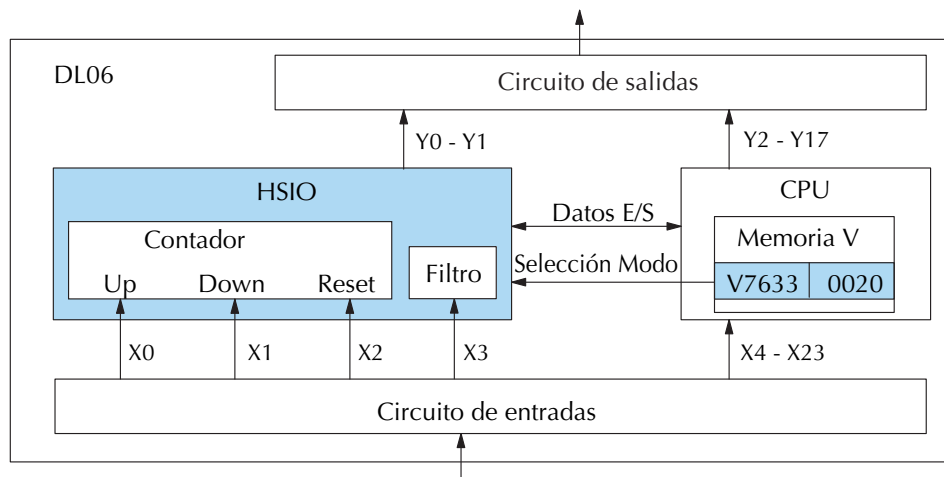
El contador en el circuito HSIO puede contar una señal para arriba y para abajo desde dos fuentes separadas (esto es, 2 encoders de un canal) o dos pulsos de señal en cuadratura. Las señales en cuadratura normalmente son generadas por encoders incrementales, que puede ser rotatorios o lineales. Este contador tiene un rango de -8388608 hasta 8388607. Usando CT174 y CT175, el contador en cuadratura puede contar a una frecuencia de hasta 7 kHz.

Diagrama de bloques funcional

El diagrama de abajo muestra la función de alta velocidad en el modo 20. Cuando el byte menos significativo de la memoria V7633 contiene un número BCD "20", se activa el contador incremental-decremental en el circuito HSIO. Para contar en cuadratura, la entrada X0 es dedicada a la señal en cuadratura fase A, y la entrada X1 recibe la señal de la fase B. X2 queda dedicado al comando de vuelta a 0 y es activo cuando X2 se cierra.



Para conteo normal incremental, la entrada X0 es dedicada a la señal de conteo incremental y la entrada X1 es dedicada a la señal de conteo para abajo o decremental. La entrada X2 es el comando de vuelta a 0 y es activo cuando X2 se cierra.



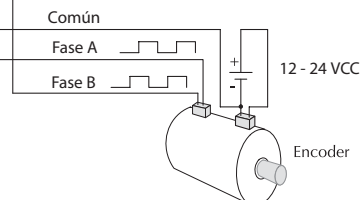
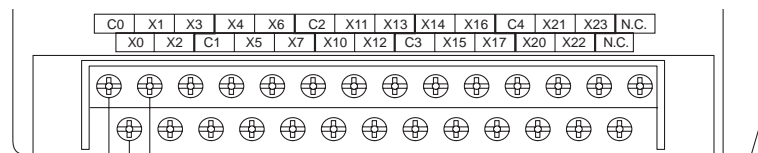
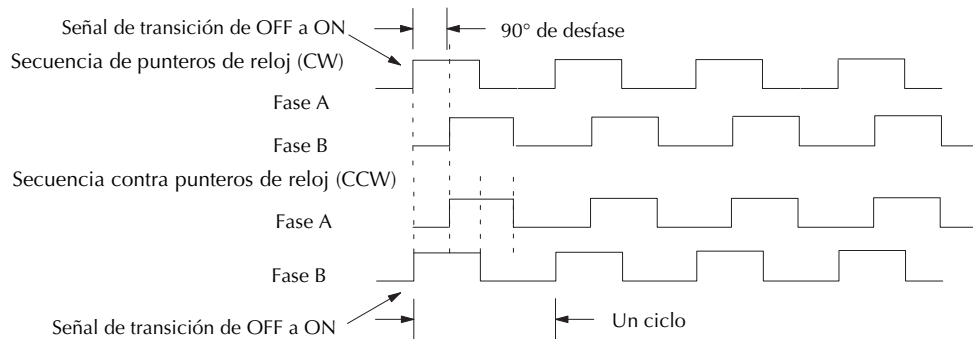
Señal en cuadratura del encoder.

Las señales en cuadratura contiene la información de posición y dirección y la frecuencia representa la velocidad del movimiento. La fase A y B mostradas abajo están desfasadas en 90° y es por eso que existe el nombre cuadratura.

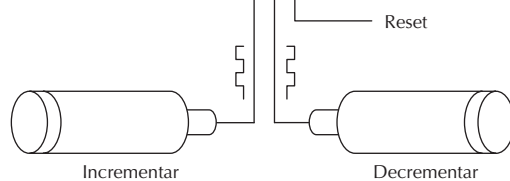
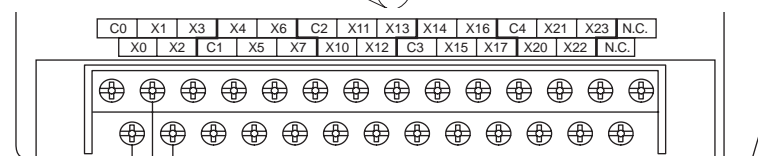
Cuando sucede una transición de OFF para ON de la fase A antes que la transición de OFF para ON de la fase B, el contador cuenta para arriba (indica movimiento a favor de los punteros del reloj por convención). Si la transición de OFF para ON de la fase B sucede antes que la transición de la fase A, el contador cuenta hacia abajo (indica movimiento en contra de los punteros el reloj)

Diagrama de cableado

Se muestra abajo un diagrama de cableado para encoders al DL06 en el modo 20. La mejor opción para usar encoders es que tengan salidas drenadoras (colector abierto NPN); si el encoder es surtidor debe entregar de 12 a 24 Volt de corriente continua. Por favor note que encoders con salidas surtidoras de 5 Volt no trabajarán con las entradas del DL06.



Entrada de encoder en cuadratura



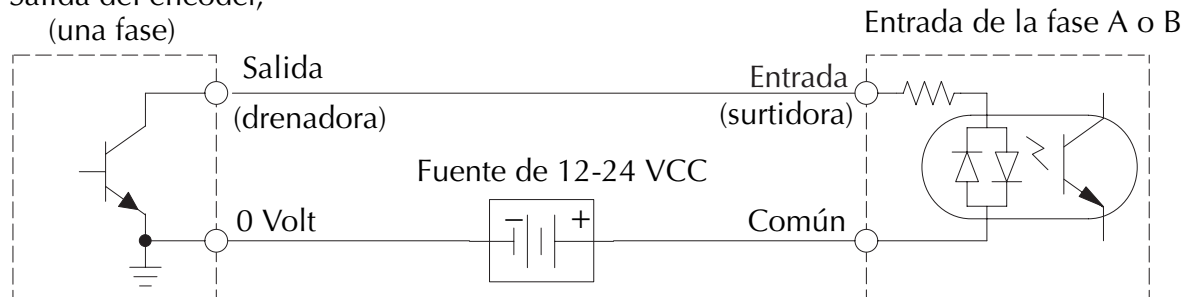
Entrada de sensores de proximidad

Conexión de salidas de un encoder

Las entradas de C.C. de los PLCs DL06 son flexibles ya que pueden detectar flujo de corriente en cualquier dirección, de modo que pueden ser conectados a circuitos drenadores o surtidores. En el circuito siguiente, un encoder tiene salidas de transistor de colector abierto NPN. Drena la corriente del punto de entrada del PLC, que surte la corriente.

La fuente de alimentación puede ser la fuente auxiliar de +24VCC u otra fuente (+12VCC o +24VCC), si se cumplen las especificaciones de entrada.

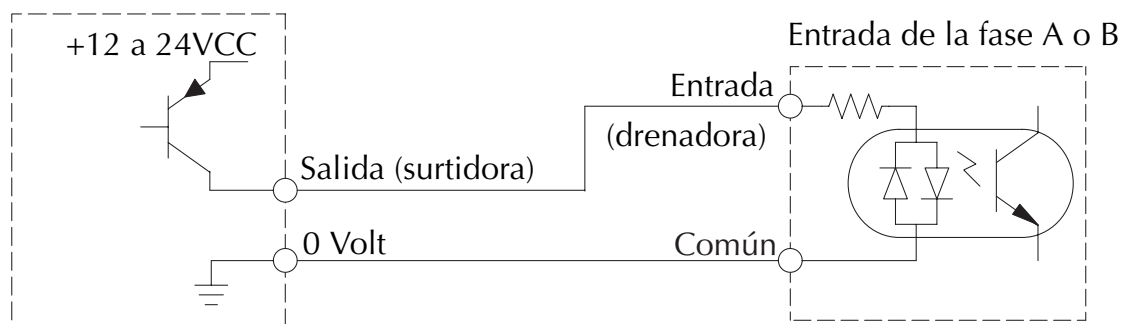
Salida del encoder,
(una fase)



En el circuito siguiente, un encoder tiene salidas de transistor de emisor abierto PNP. Surte corriente a la entrada del PLC, que drena la corriente a 0 Volt.

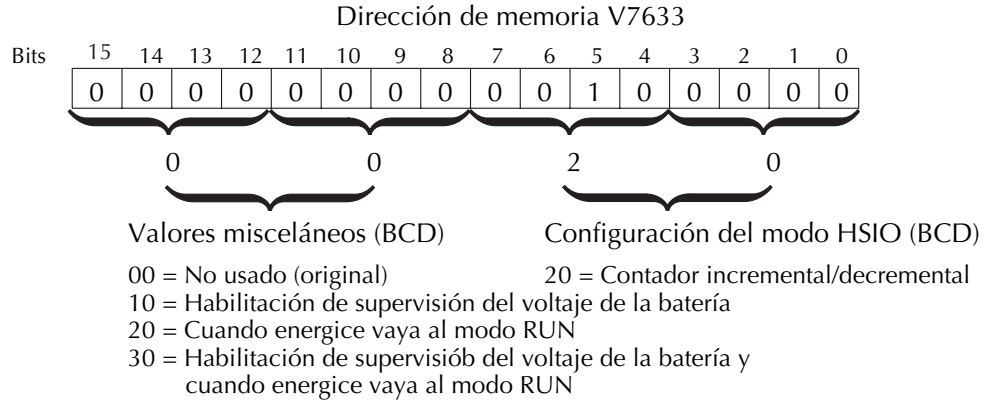
Ya que el encoder surte corriente, no se requiere ninguna fuente de alimentación adicional. Sin embargo, observe que la salida del encoder debe ser 12 a 24 volt (Pulsos de salidas de encoders de 5 VCC no operarán correctamente).

Salida del encoder,
(una fase)



Configuración del modo 20

Recuerde que la memoria V7633 es la de selección del modo remoto HSIO. Use el número 20 BCD en el byte menos significativo en V7633 para seleccionar este modo.



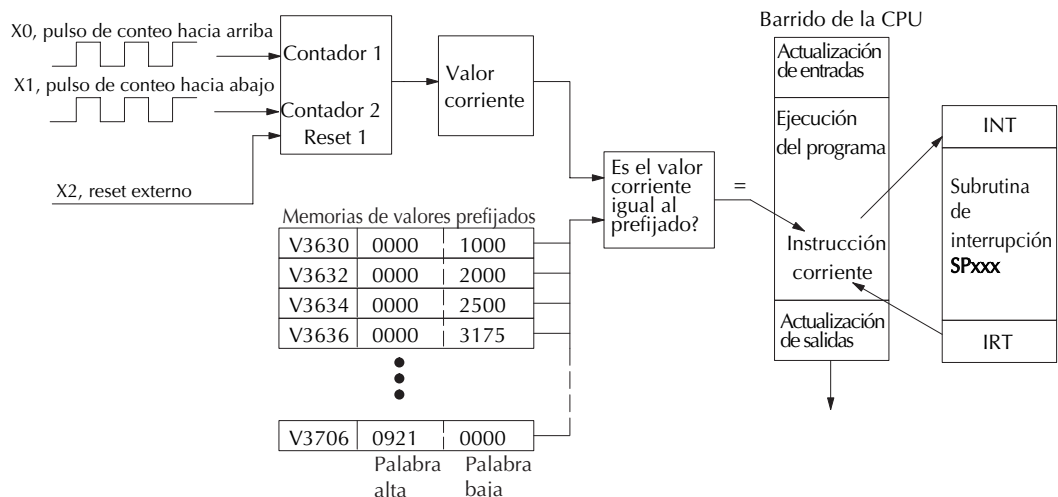
Escoja el método más conveniente de programar el valor V7633 entre:

- Incluya las instrucciones LD y OUT en su programa ladder
- Use el programa *DirectSOFT* (Menús Memory Editor o Data View)
- Use el programador portátil D2-HPP

Recomendamos usar el método de instrucciones en su programa para que siempre esté incluido en el programa. Un programa de ejemplo se muestra más tarde en esta sección.

Relevadores especiales y valores prefijados del modo 20

El objetivo de contar es causar una acción particular al alcanzar el conteo el valor prefijado. Refiérase a la figura de más abajo. Cada contador tiene 24 valores prefijados que usted puede programar. Un valor prefijado es un número que usted selecciona y almacena de modo que el contador continuamente compare el valor corriente con el valor prefijado. Cuando los dos son iguales se energiza un relevador especial y la ejecución del programa salta a una rutina de interrupción. Recomendamos usar los relevadores especiales en la rutina de servicio de interrupción para causar cualquier acción inmediata que se desee. Después que la rutina de interrupción ha sido completada, la CPU vuelve al programa principal, retomando la ejecución del programa desde el punto de interrupción. La función de comparación está lista para el próximo evento el valor prefijado.



Configuración de entradas X.

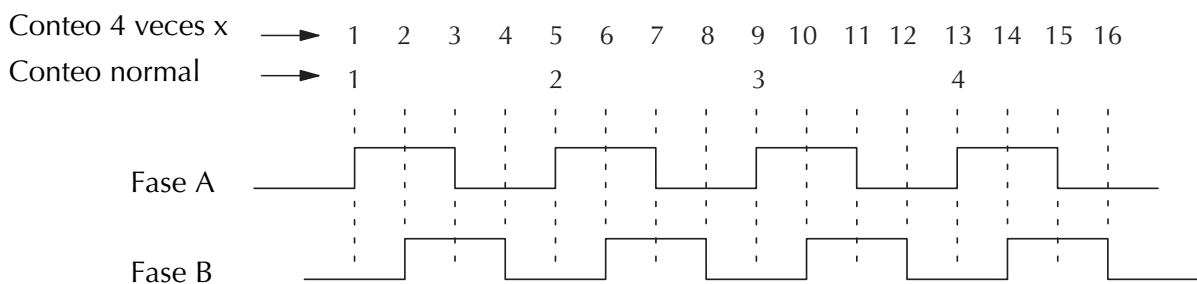
Las opciones de entradas discretas configurables están listadas en la tabla de más abajo. La sección de la operación en el modo 60 al final de este capítulo describe la programación de las constantes de tiempo del filtro.

Contador incremental y decremental del modo 20.

E

Entrada	Memoria de configuración	Función	Código hexadecimal
X0	V7634	Conteo incremental	0202 (estandar, absoluto)
			0302 (estandar, incremental)
		Fase A	0002 (cuadratura, absoluto) (valor original)
			0102 (cuadratura, incremental)
			1002 Conteo 4x (cuadratura, absoluto) *
1102 Conteo 4x(cuadratura, incremental) *			
X1	V7635	Conteo decremental o Fase B	0000
X2	V7636	Reset del contador (sin interrupción)	0007** (valor original) 0207**
		Reset del contador (con interrupción)	0107** 0307**
		Entradas de pulsos	0005
		Entrada filtrada	xx06 (xx = tiempo de filtro, 0 - 99ms (BCD))
X3	V7637	Entradas de pulsos	0005
		Entrada filtrada	xx06 (xx=tiempo de filtro, 0-99ms (BCD) (original)

* Con esta característica, se puede contar 4 veces mas con el mismo encoder. Vea la forma en que es hecho en el diagrama inferior.



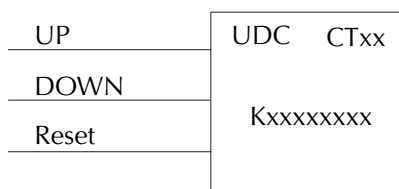
** Con el comando de reset Ud. puede escoger las opciones normal o mas rápida. Sin embargo, el comando de vuelta a 0 mas rápido no reconoce cambios de los valores prefijados durante la ejecución del programa. Cuando se coloca '0007' o '0107' en V7636 y los valores prefijados se cambian durante la ejecución del programa, el DL06 reconoce el valor cambiado durante el comando de vuelta a 0. Cuando se coloca '0207' o '0307' en V7636 y los valores prefijados se cambian durante la ejecución del programa, el DL06 no verifica si ha cambiado el valor cambiado durante el comando de vuelta a 0, pero tiene un tiempo más corto de vuelta a 0.

Escribiendo el programa de control del modo 20.

El símbolo para la instrucción de contador es UDC (contador incremental-incremental). El DL06 puede tener hasta 128 contadores, apodados CT0 hasta CT177. El contador de alta velocidad en el circuito HSIO es accesado en el diagrama usando UDC CT174.

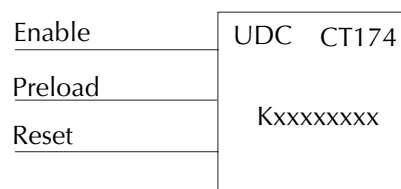
Usa las memorias del contador CT174 y CT175 exclusivamente cuando el modo HSIO es activo (de otra forma, CT174 y CT175 están disponibles para uso como contador normal). El contador HSIO necesita dos memorias porque es un contador de doble palabra. Tiene tres entradas como mostrado en el diagrama de abajo. La primera entrada es ENABLE (habilitar) permite contar cuando está activa. El punto medio es usado para transferir el valor al contador. La última entrada es el comando de vuelta a 0. La entrada ENABLE debe ser verdadera para que el contador cuente.

Función de contador normal



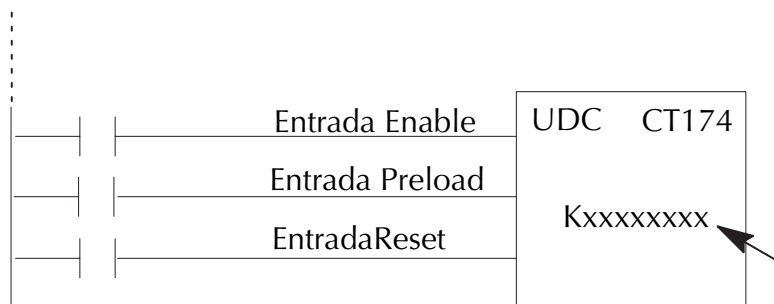
Cuenta incrementando y decrementando
Use Preload escribiendo al valor
Entrada Reset es sólo interna

Función de contador HSIO



Cuenta incrementando y decrementando (desde X0, X1)
Puede usar Preload para cambiar valor corriente
Entrada Reset puede ser interna o externa

La próxima figura muestra como el contador HSIO aparecerá en un programa con diagrama ladder.

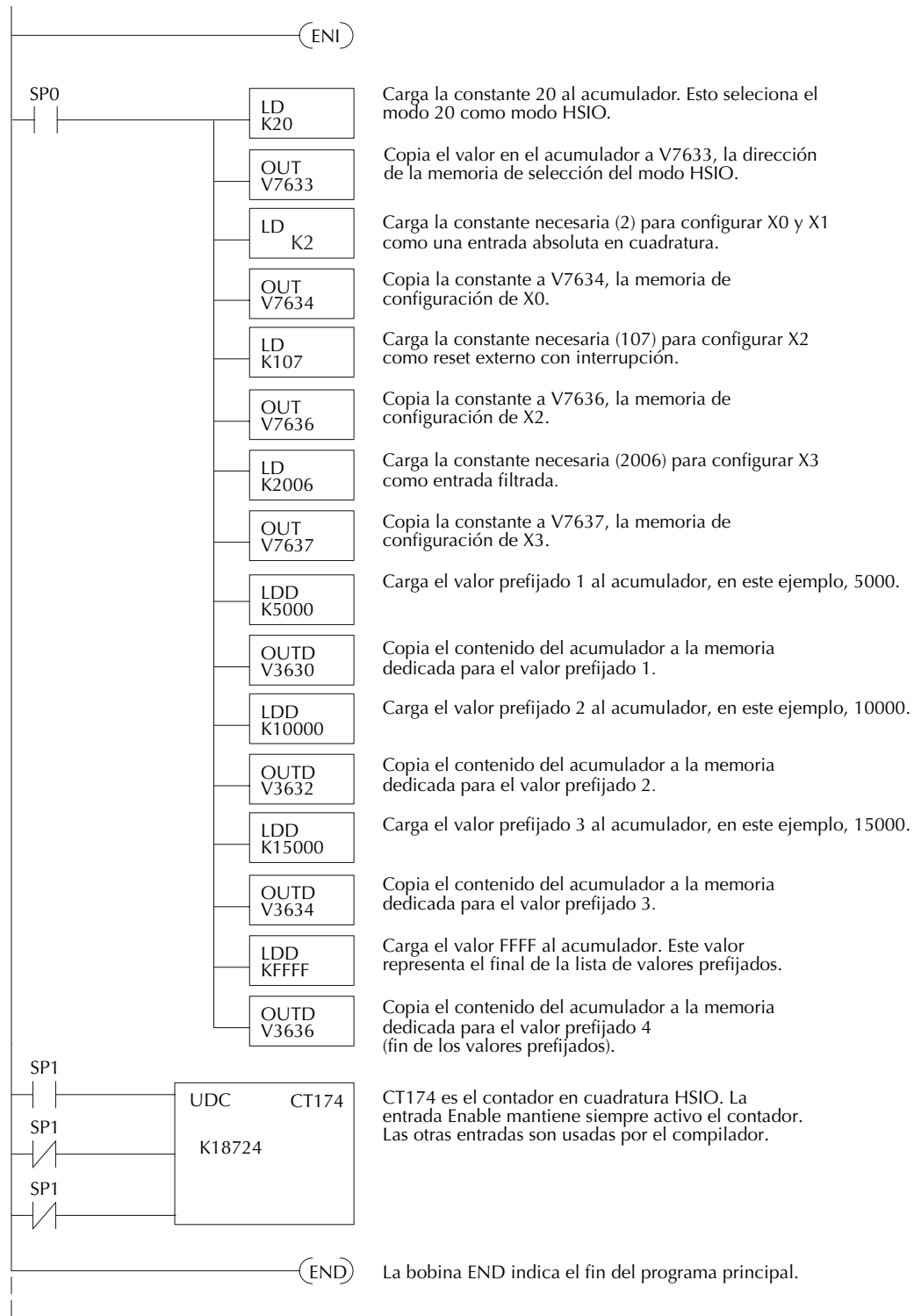


Rango de valor prefijado:
-8388608 a 8388607

Cuando la entrada Enable está verdadera, el contador responderá a los pulsos en cuadratura en X0 y X1, incrementando o decrementando el valor corriente del contador en CT174 y CT175. El contacto de vuelta a 0 puede recibir un comando desde la lógica o de una entrada externa X2.

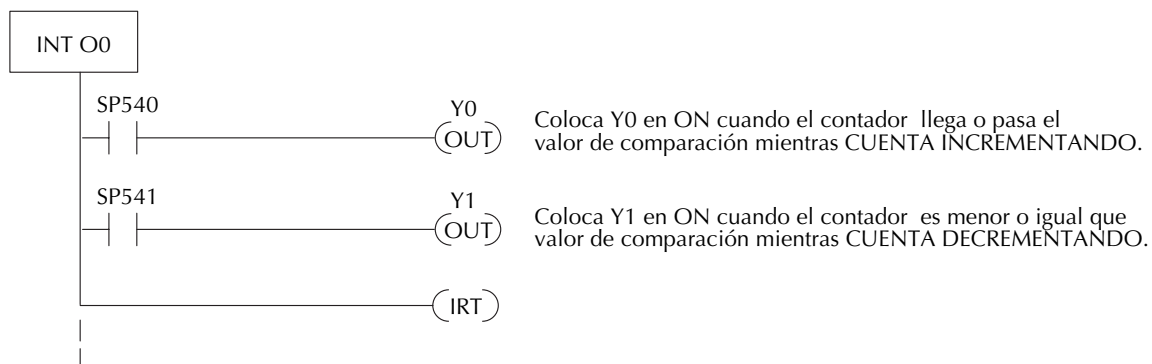
Ejemplo 1 modo 20. Contador en cuadratura con interrupción

El siguiente ejemplo es la forma simple de usar un contador de alta velocidad con una interrupción.



continúa en la próxima página

continúa desde la página anterior



Las instrucciones LDA han configurado la memoria V según lo requerido, es decir, 20 en V7633 para el modo y 0202 en V7634 para indicar el modo absoluto de contador incremental y decremental con valor prefijado.

Al colocar 0107 en V7636 se selecciona un comando de vuelta a 0 externo para el contador CT174 y ejecutará la interrupción en la transición de falso para verdadero.

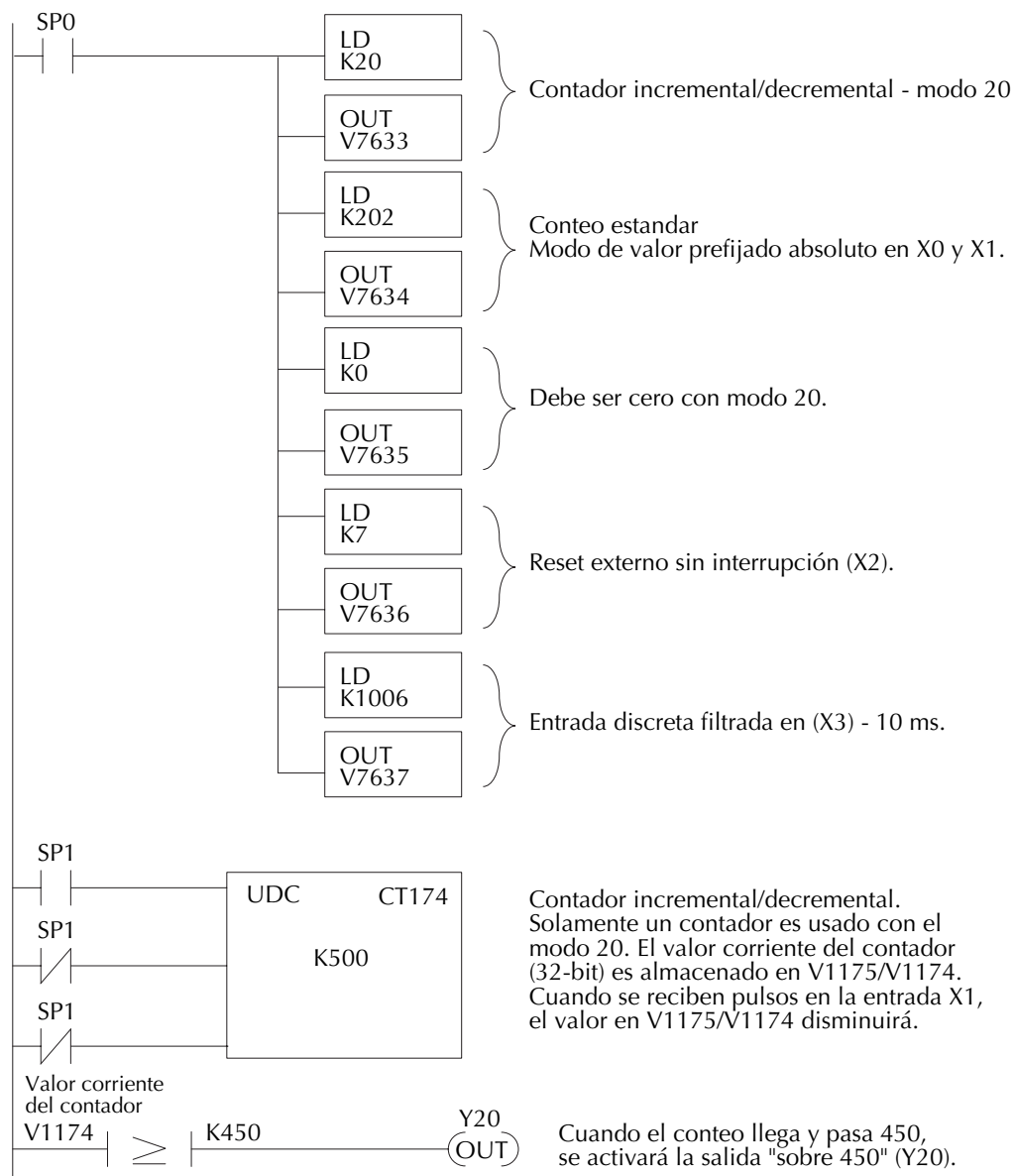
Los valores prefijados para el conteo han sido almacenados en las direcciones V3630 hasta V3635. La dirección par que sigue contiene FFFF para indicar que no hay más valores prefijados.

E

Ejemplo 2 modo 20. Contador incremental y decremental con entradas normales

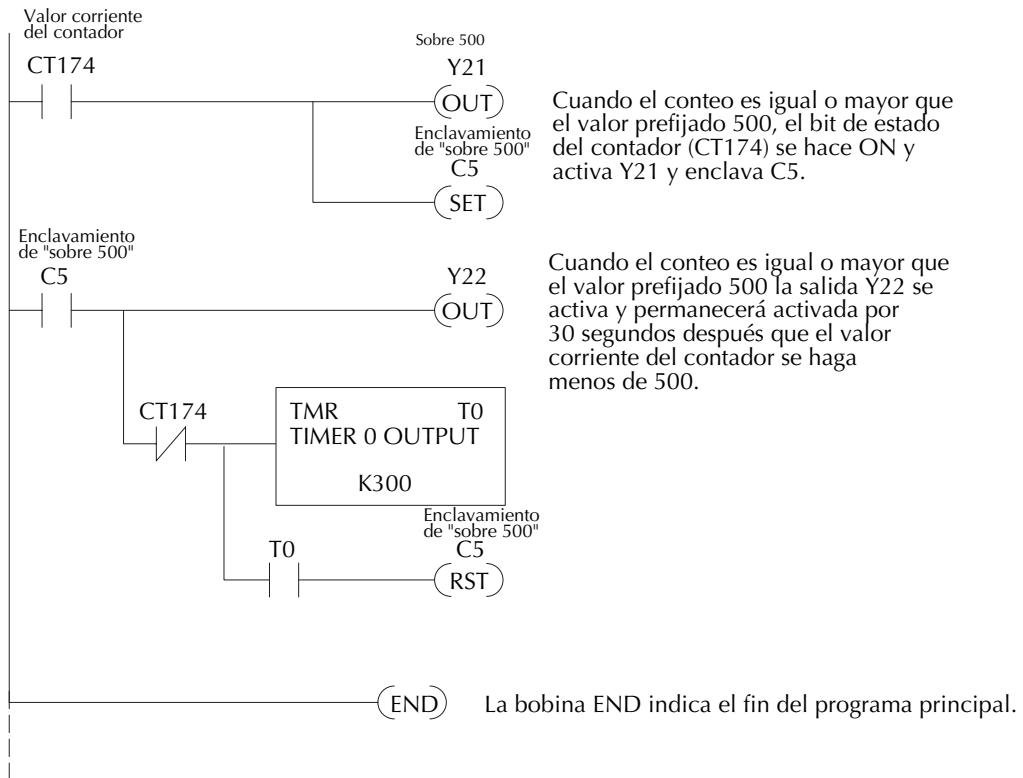
En este ejemplo, hay una correa transportadora "A" que transporta botellas para ser inspeccionadas. Durante el curso del proceso, un sensor cuenta a las botellas que van en la correa "A" para la inspección y otro sensor cuenta cuántas botellas son retiradas de la línea de productos terminados. Cuando se ha alcanzado 500 botellas en el proceso, un indicador luminoso "sobre 500" prende y se activa un portón que cambia la ruta para enviar las botellas entrantes a la correa transportadora "B". El portón que cambia la ruta permanecerá activado por 30 segundos después que la correa transportadora "A" contenga menos de 500 botellas.

El programa de abajo muestra cómo se puede hacer una lógica ladder para hacer esta función. Note el uso de V1174. Esta dirección de memoria almacena el conteo corriente de CT174 que se usa con el DL06.



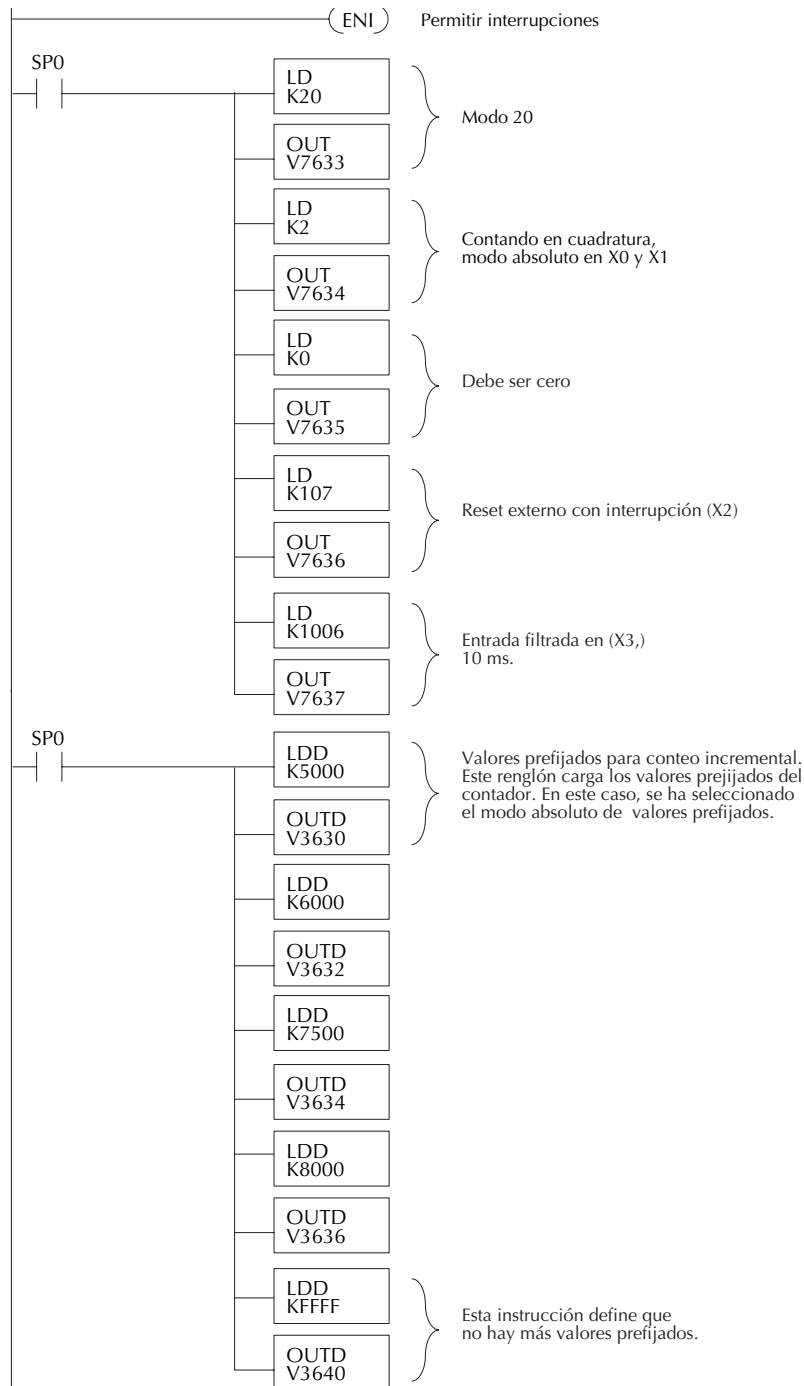
Continúa en la próxima página.

continúa desde la página anterior



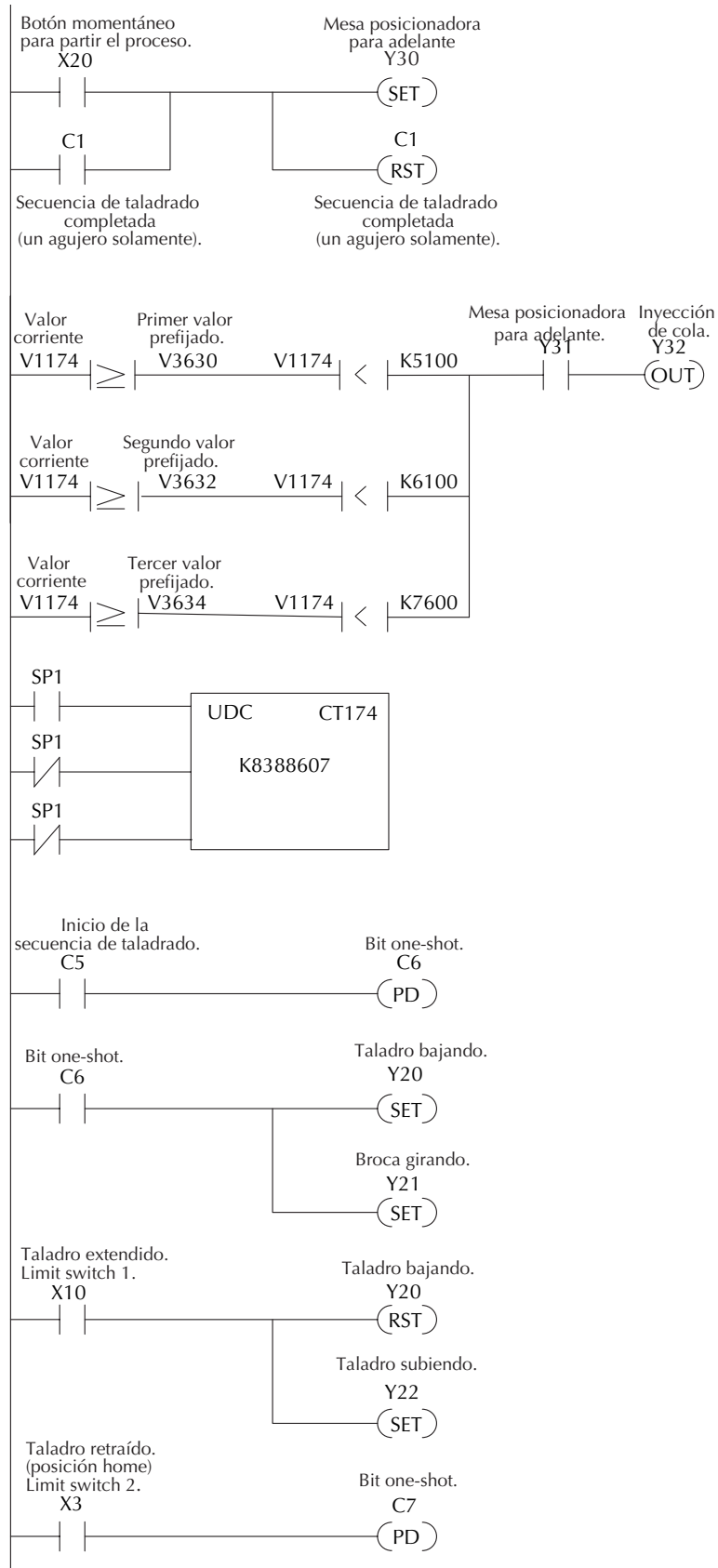
Ejemplo 3: modo 20 - Contador en cuadratura

En este ejemplo, una pieza de madera es taladrada con 3 agujeros y luego los agujeros son inyectados con cola, para usar tarugos a ser colocados en otra etapa. En este caso, un encoder en cuadratura es conectado a una mesa posicionadora que mueve horizontalmente la pieza y un taladro sube y baja adecuadamente. La mesa posicionadora se detendrá y el taladro bajará para perforar un agujero en la posición exacta. Después que los 3 agujeros son taladrados, la mesa posicionadora retrocede e inyecta cola en los mismo agujeros.



Continuado en la próxima página.

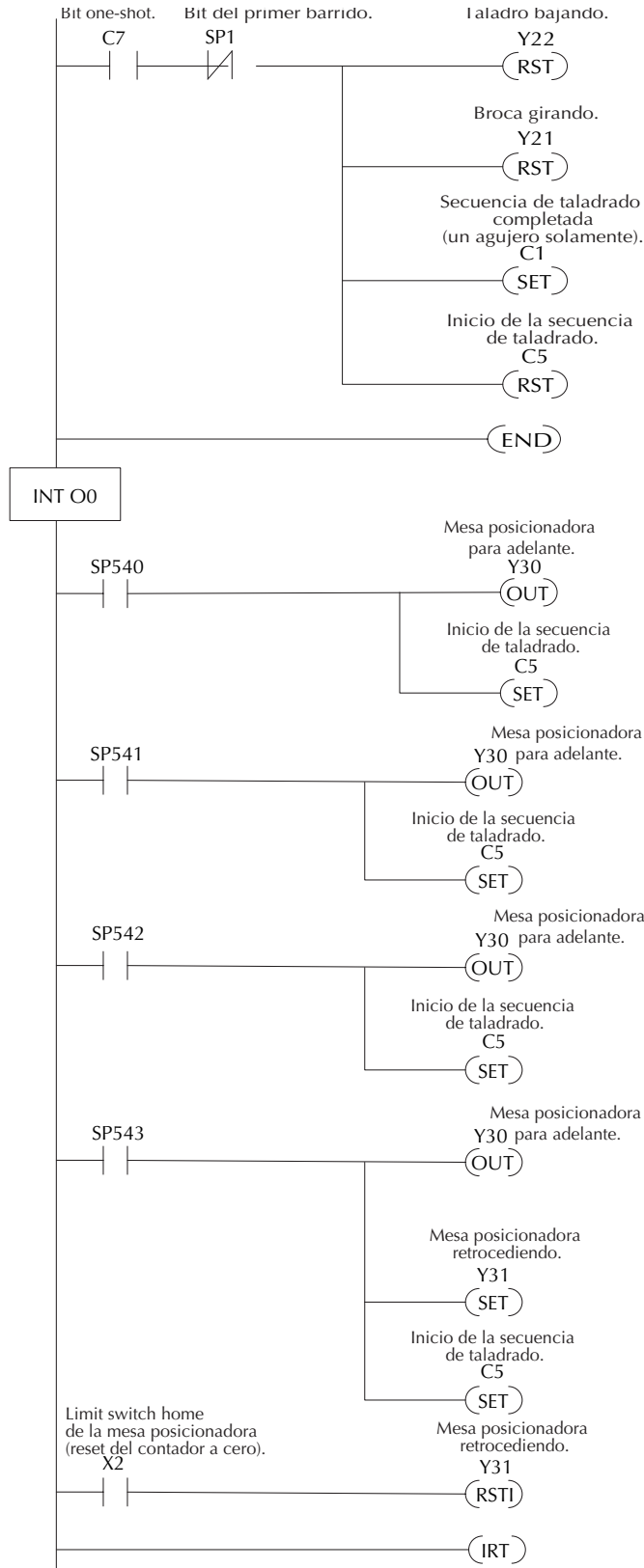
continúa desde la página anterior



Continúa en la próxima página.

E

Continuado desde la página anterior.



Búsqueda de problemas en el modo 20

Si usted tiene problemas con la operación del Modo 20, estudie los siguientes síntomas y las causas posibles. Los problemas más comunes se listan abajo.

Síntoma: El contador no cuenta.

Posibles causas:

1. **El sensor de campo y cableado** – Verifique que el encoder u otra entrada de un aparato en el campo prende realmente y el LED de estado de X0 y X1 se ilumina en el PLC. Un encoder incremental normal prende alternamente los LEDs en X0 y X1 cuando se gira lentamente (1 RPM). O, el problema podría ser debido a un problema de cableado con fuentes drenadoras o surtidoras. Verifique la conexión a tierra de la señal. Verifique también que el pulso en el tiempo, el ciclo, el nivel de voltaje y la frecuencia están dentro de las especificaciones de entrada.
2. **Configuración** – Asegúrese que todos los parámetros de configuración están correctos. V7633 debe ser colocado en 20, y V7634 debe ser colocado en "0002" para habilitar la entrada de la fase A y V7635 debe ser colocado en "0000" para habilitar la entrada de la Fase B.
3. **Se ha quedado en Reset** – Verifique el estado de la entrada Reset, X2. Si X2 está ON, el contador no contará porque está en reset.
4. **El programa ladder** – Asegúrese que usa el contador CT174 en su programa. La entrada principal es la señal de habilitación para el contador. Debe estar ON para que el contador cuente. La entrada del medio es una entrada falsa y debe estar apagada para que el contador cuente. La entrada del abajo es el reset del contador y debe estar apagada durante el proceso de contar.

Síntoma: El contador cuenta en la dirección incorrecta (para arriba en vez de hacia abajo, y viceversa).

Posibles causas:

1. **Definición del canal A y B** – Es posible que los canales A y B del encoder estén cableados al revés de la rotación deseada y la orientación que desea. Solamente cambie las entradas X0 y X1 y se invertirá la dirección en que cuenta.

Síntoma: El contador cuenta hacia arriba y abajo pero no hace reset.

Posibles causas:

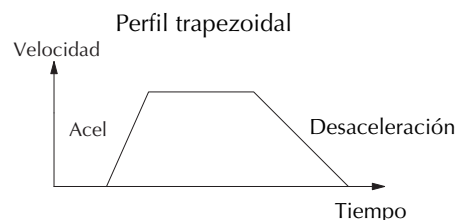
1. Verifique el LED indicador de estado de X2 para asegurarse que está activo cuando quiere que haya un reset. Verifique también que la memoria V7636 de configuración para X2 es configurada como 7. O, si usted usa un reset interno, use el menú DataView de *DirectSOFT* para controlar la entrada reset al contador.

Modo 30: Salida de tren de pulsos

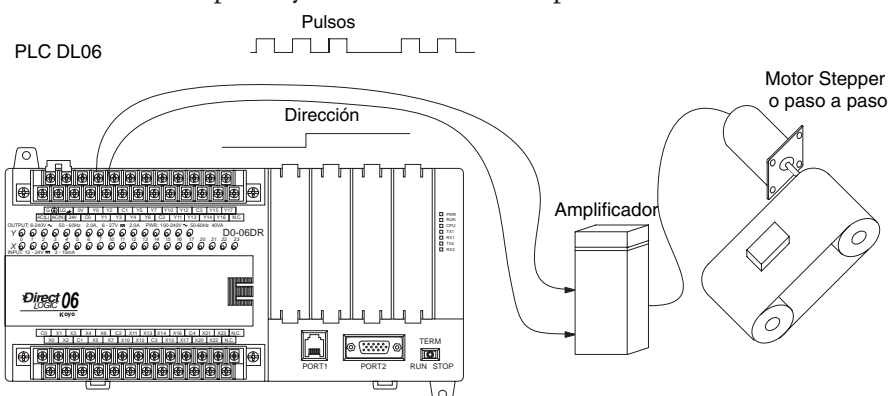
Propósito del modo 30.

El circuito HSIO en el modo 30 genera pulsos de salidas adecuados para control de un circuito abierto de un sistema de movimiento y posición de eje simple. Genera señales de pulso y dirección que pueden conectar a sistemas de accionamiento de motores y ejecutar varios tipos de control de movimiento. Usando la salida en el modo 30 usted puede seleccionar tres perfiles detallados más abajo en este capítulo:

- **Perfil Trapezoidal automático** – Con rampa de aceleración hasta una velocidad definida y luego una rampa de desaceleración.
- **Perfil trapezoidal paso a paso** – Aceleración y desaceleración por pasos de acuerdo a una definición del usuario incluyendo una velocidad definida.
- **Control de velocidad** – solamente usa velocidad y dirección.

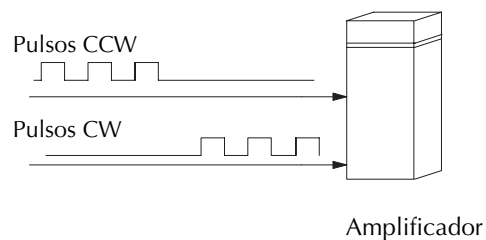


El circuito HSIO llega a ser un generador de pulsos de alta velocidad (hasta 10 KHz.). Al programar aceleración y la desaceleración, posición y velocidad, la función HSIO funciona automáticamente para definir todo el perfil de movimiento. La figura de abajo muestra el PLC DL06 generando señales de pulso y dirección a un amplificador de un accionamiento de un



sistema con un motor paso a paso (stepper motor). Los pulsos producen el perfil independientemente y sin interrupción de la ejecución del programa en la CPU.

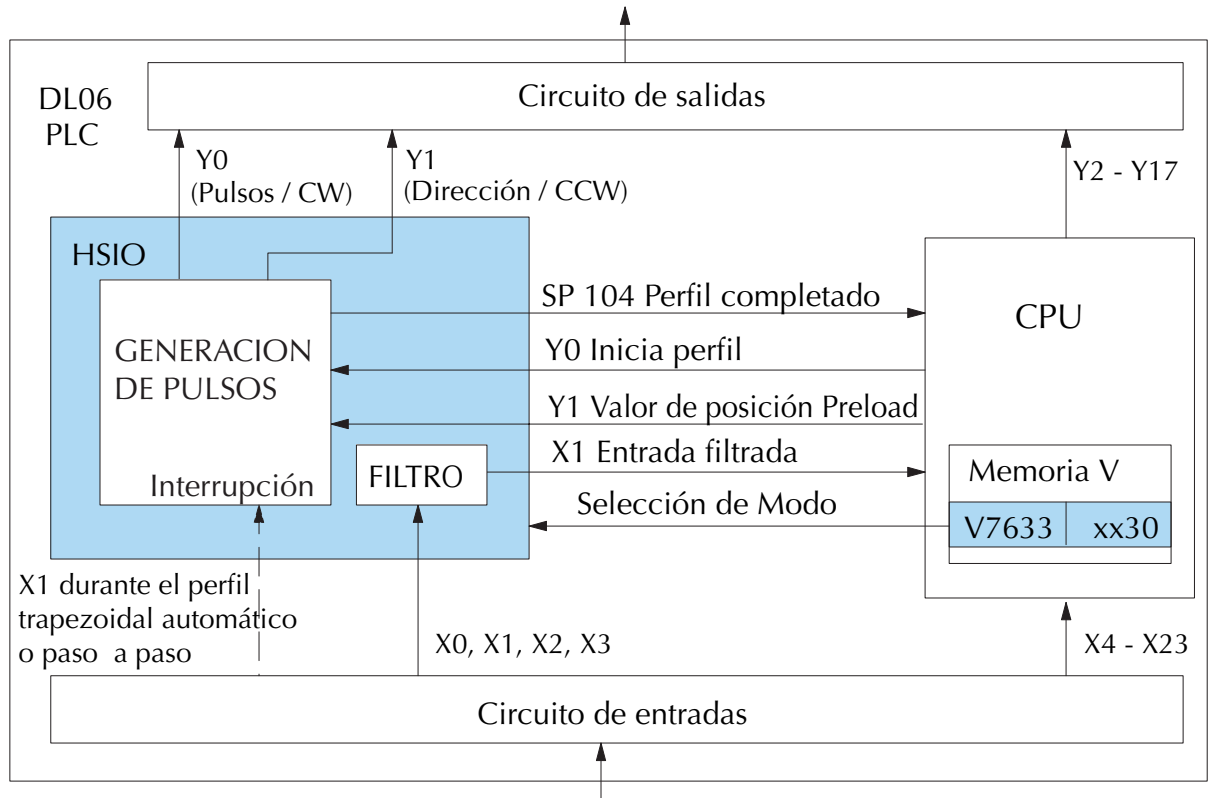
En la figura de arriba, el PLC DL06 genera señales de pulso y dirección. Cada pulso representa el menor incremento de movimiento para el sistema de posicionamiento (tal como un sistema paso a paso). Alternativamente, el modo de salidas de tren de pulsos puede ser configurado para entregar señales de pulsos a favor de los punteros del reloj (CW) y en contra de los punteros del reloj (CCW) como se muestra en la figura HSIO adyacente.



NOTA: La salida de pulsos está diseñada para sistemas de circuito abierto. Esto, más la velocidad mínima de 40 pulsos por segundos lo hace incapaz de hacer control de servomotores.

Diagrama de bloques funcional del modo 30.

El diagrama abajo muestra las funciones del modo 30. Cuando el byte inferior de la memoria V7633 contiene un "30" BCD, se activa la función de generación de un tren de pulsos. Las salidas de pulsos usan los terminales Y0 e Y1 en el conector de salida. Recuerde que sólo se puede tener esta función si las salidas son del tipo de corriente continua.



NOTA IMPORTANTE: : En el modo de salida de pulsos, Y0 e Y1 son redefinidas o son usadas en dos maneras diferentes. Las referencias físicas se refieren a los terminales, mientras que las referencias lógicas se refieren a las entradas y salidas en el programa. Por favor lea los párrafos de abajo para entender este punto muy crucial.

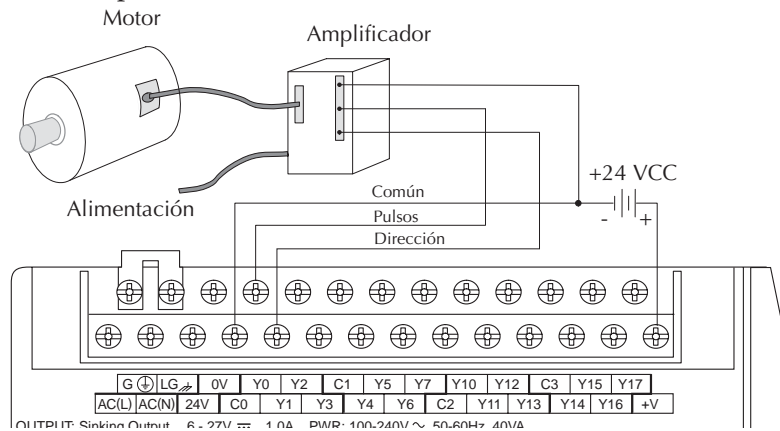
Note las asignaciones de puntos de entradas y salidas y uso en el diagrama de arriba :

- X0, X1, X2 y X3 pueden ser entradas filtradas o entradas de pulso en el modo de salida de pulsos y están disponibles como contactos de entradas al programa ladder.
- X1 se comporta como una interrupción externa para el generador de pulsos para perfiles automáticos trapezoidal o paso a paso. En otros modos, puede ser usado como entrada filtrada o entrada de pulso igual que X0 (modo de registro mostrado arriba).
- Las referencias Y0 e Y1 son usadas en dos formas diferentes: En el conector de salidas discretas, Y0 e Y1 entrega los pulsos al sistema de movimiento. El programa ladder usa las referencias lógicas Y0 e Y1 para las funciones de "iniciar perfil" y "escribir el valor de posición" en el modo 30.

Esperamos que la explicación de arriba explique porqué algunos nombres de referencia de entradas y salidas tienen dos significados en el modo de pulso de salida. Por favor lea el resto de esta sección con cuidado para evitar confusión en cual función de entradas o de salidas están siendo discutidas.

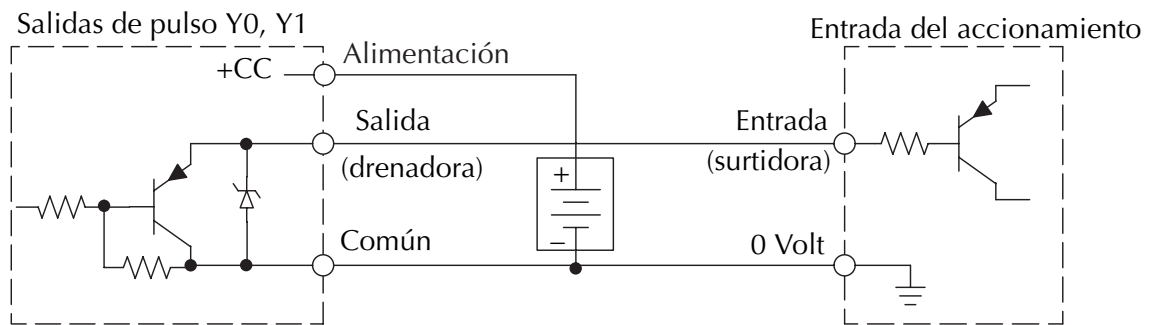
Diagrama de cableado del modo 30

El diagrama de cableado generalizado mostrado abajo muestra las salidas Y0 e Y1 conectadas a las entradas del amplificador de un sistema de control de movimiento

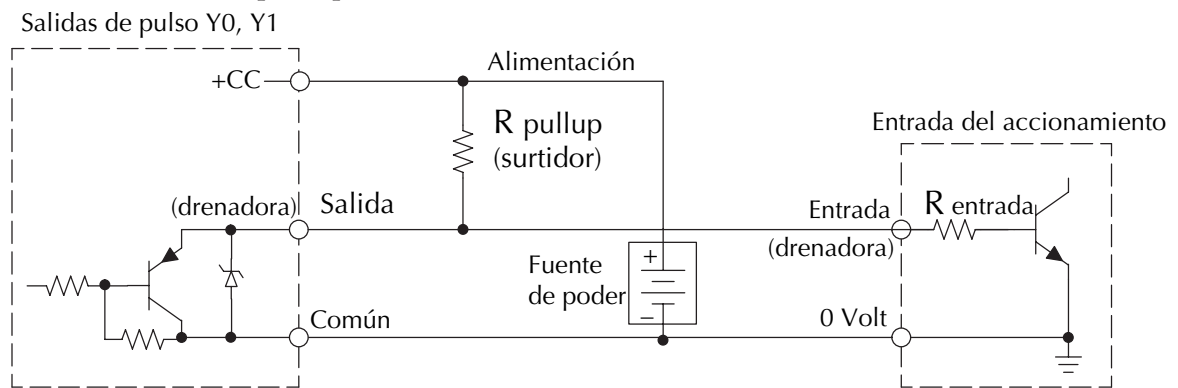


Conexión de entradas de un accionamiento

Las señales de pulso de las salidas Y0 y Y1 irán típicamente a los circuitos de entrada de un accionamiento amplificador según lo mostrado arriba. Será útil tener los diagramas esquemáticos del circuito equivalente del amplificador del accionamiento. El diagrama siguiente muestra cómo conectar un circuito de entrada del accionamiento del tipo surtidor.



El circuito siguiente muestra cómo conectar un circuito de entrada del accionamiento del tipo drenador usando una resistencia pullup. Vea por favor el capítulo 2 para aprender cómo calcular e instalar la resistencia pull up.



Especificaciones del perfil de movimiento.

Lea la tabla siguiente para este objetivo:

Especificaciones del perfil de movimiento	
Parámetro	Especificación
Perfiles	Automático trapezoidal-Rampa de aceleración/Velocidad/ Rampa de desaceleración
	Trapezoidal paso a paso-Aceleración de paso/desaceleración
	Control de velocidad- Solamente velocidad y dirección
Rango de posición	-8388608 hasta 8388607
Posicionamiento	Comando absoluto o relativo
Rango de velocidad	40 Hz hasta 10 kHz
Memorias	V3630 hasta V3652 (Tabla de parámetros del perfil)
Posición corriente	CT174 y CT175 (V1174 y V1175)

Configuración de las entradas y salidas físicas.

Las opciones de configuración de entradas y salidas discretas para el modo 30 están listadas en la tabla de más abajo. La CPU usa el contacto SP104 para informar que el perfil se ha completado. La memoria V7632 es usada para seleccionar pulso y dirección o los dos sentidos de rotación externa CW o CCW. La entrada X1 está dedicada como interrupción para uso en el modo de registración.

Funciones lógicas de entradas y salidas

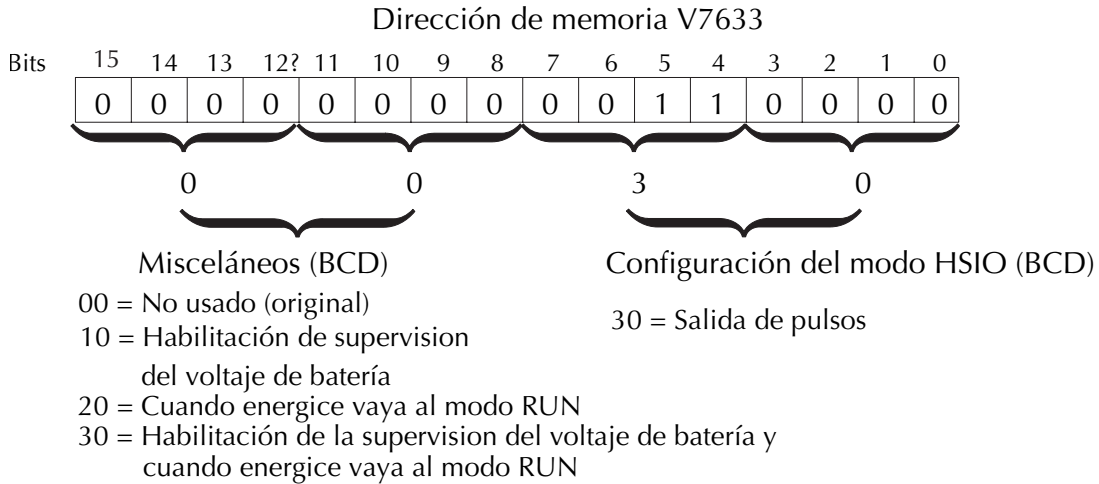
Las referencias lógicas siguientes de entradas y salidas definen las funciones que permiten una comunicación del HSIO con el programa ladder.

Configuración de entradas y salidas físicas			
Entrada	Memoria de configuración	Función	Código hexadecimal
-	V7632	Y0 = Pulso;Y1 = Dirección	0103
		Y0 = Pulsos CW;Y1 = CCW	0003 (original)
X0	V7634	Entrada de pulso	0005
		Entrada filtrada	xxDL06 (xx= tiempo de filtrado, 0-99ms BCD)
X1	V7635	Entrada de pulso	0005
		Entrada filtrada	xxDL06 (xx= tiempo de filtrado, 0-99ms BCD)
X2	V7636	Entrada de pulso	0005
		Entrada filtrada	xxDL06 (xx= tiempo de filtrado, 0-99ms BCD)
X3	V7637	Entrada de pulso	0005
		Entrada filtrada	xxDL06 (xx= tiempo de filtrado, 0-99ms BCD)

Funciones lógicas de E/S	
E/S lógicas	Función
SP104	Perfil completo - el HSIO hace ON el relevador especial SP104 en la CPU cuando el perfil termina. Se va a OFF cuando el perfil parte (Y0).
X1	Interrupción externa - si la característica de interrupción se selecciona para el perfil trapezoidal automático o el perfil trapezoidal paso a paso, el DL06 genera pulsos hasta X1 se haga ON. Después de que esté ON, la unidad genera pulsos que se definen como posición deseada.
Y0	Parte el perfil - el programa ladder hace ON Y0 para iniciar el movimiento. Si se hace OFF antes de que el movimiento termine, el movimiento para. Al hacerlo ON nuevamente comienza otro perfil, a menos que la posición corriente iguale la posición deseada.
Y1	Cargar el valor de posición - si se para el movimiento y el bit Y0, Parte el perfil, está apagado (OFF), se puede cargar un nuevo valor en CT174/CT175, y hacer ON Y1. En esa transición, el valor en CT174/CT175 se convierte en la posición corriente.

Configuración del modo 30

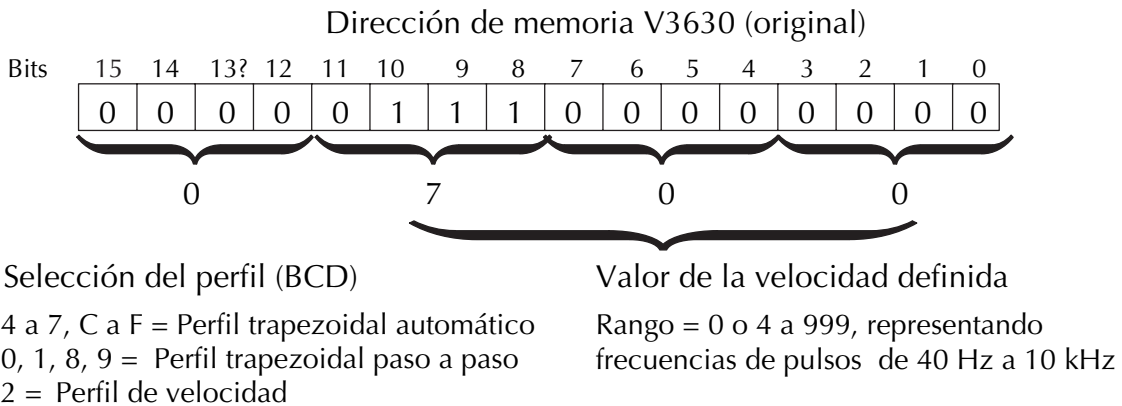
Recuerde que la memoria V7633 es la de selección del modo remoto HSIO. Vea el diagrama de abajo. Use "30" BCD en el byte inferior para seleccionar este modo.



Escoja el método más conveniente de programar el valor V7633 entre:

- Incluya las instrucciones LD y OUT en su programa ladder
- Use el programa *DirectSOFT* (Menús Memory editor o Data View)
- Use el programador portátil D2-HPP

Recomendamos usar el método de instrucciones en su programa para que siempre esté incluido en el programa. Un programa ejemplo se muestra más tarde en esta sección.



Memoria de selección de velocidad o de perfil

La primera dirección en la tabla de parámetros del perfil almacena dos importantes piezas de información. Los cuatro bits más importantes (12-15) seleccionan el tipo de perfil requerido. Los 12 bits inferiores (0-11) seleccionan la velocidad definida.

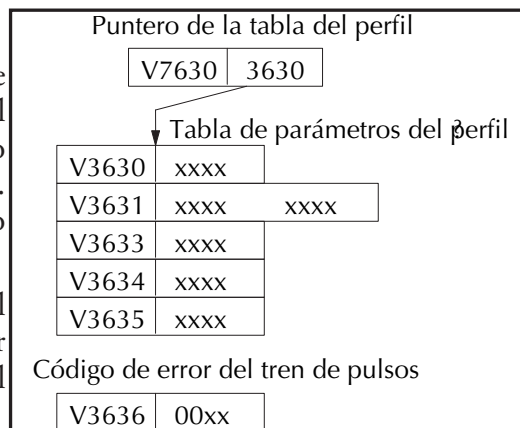
El programa ladder debe programar esta dirección antes de iniciar cualquiera de los 3 perfiles. Las instrucciones LD y OUT definen 16 bits. Por lo tanto asegúrese de que especifica completamente los valores de cuatro dígitos para el perfil cada vez.

La selección absoluta y relativa determina como el circuito HSIO interpretará su posición especificada del “blanco” (Valor de conteo a ser alcanzado). Los “blancos” de posición absolutos son referenciados a 0. Los “blancos” de posición relativos se refieren a la posición corriente (posición del blanco previa). Usted puede escoger que método de referencia es más conveniente para su aplicación.

Tabla de parámetros del perfil

La memoria 7630 es una dirección de puntero que apunta al comienzo de la tabla de parámetros del perfil. Sin embargo, usted puede cambiar esto programando un valor diferente en la memoria 7630. Recuerdese de usar la instrucción LDA, convirtiendo un número octal en hexadecimal.

Este modo usa la memoria que sigue después del final de la tabla de parámetros del perfil para indicar errores en el perfil. Vea la tabla de errores en el final de esta sección para definición de códigos de error.



Configuración del perfil automático trapezoidal

Memoria V	Función	Rango	Unidades
V3630, bits 12–15	Perfil trapezoidal automático sin velocidad final (La velocidad final es fija en 0)	4=absoluto sin interrupción 5=absoluto con interrupción* C=relativo sin interrupción D=relativo con interrupción*	–
	Perfil trapezoidal automático con velocidad final (Use V3637 para definir la velocidad final)	6=absoluto sin interrupción 7=absoluto con interrupción* E=relativo sin interrupción F=relativo con interrupción*	–
V3630, bits 0–11	Velocidad blanco	4 a 999 o 0 hasta 1000	x 10 pps
V3631 / V3632	Posición blanco**	–8388608 a 8388607	Pulsos
V3633	Velocidad inicial	4 a 100	x 10 pps
V3634	Tiempo de aceleración	1 a 100	x 100 ms
V3635	Tiempo de desaceleración	1 a 100	x 100 ms
V3636	Código de error	(Vea el final de esta sección)	–
V3637	Velocidad final	4 a 100	x 10 pps

* Si selecciona el uso de interrupción, el PLC DL06 no busca por el valor de posición blanco hasta que la señal de interrupción X1 se haga verdadera.

**Para configurar un número negativo, coloque 8 en el dígito más importante. Por ejemplo, -8388608 es 88388608 en V3631 y V3632.

Tabla de perfil trapezoidal paso a paso.

Memoria	Función	Rango	Unidades
V3630, bits 12–15	Perfil trapezoidal paso a paso	0 = Absoluto sin interrupcion 7 = Absoluto con interrupcion 8 = Relativo sin interrupcion 9 = Relativo con interrupcion *	–
V3630, bits 0–11	Velocidad blanco	4 a 999 o 0 hasta 1000	x 10 pps
V3631 / V3632	Posición blanco**	–8388608 a 8388607	Pulsos
V3633	Aceleración 1	4 a 1000	x 10 pps
V3634	Distancia 1	1 a 9999	Pulsos
V3635	Aceleración 2	4 a 1000	x 10 pps
V3636	Distancia 2	1 a 9999	Pulsos
V3637	Aceleración 3	4 a 1000	x 10 pps
V3640	Distancia 3	1 a 9999	Pulsos
V3641	Aceleración 4	4 a 1000	x 10 pps
V3642	Distancia 4	1 a 9999	Pulsos
V3643	Desaceleración 5	4 a 1000	x 10 pps
V3644	Distancia 5	1 a 9999	Pulsos
V3645	Desaceleración 6	4 a 1000	x 10 pps
V3646	Distancia 6	1 a 9999	Pulsos
V3647	Desaceleración 7	4 a 1000	x 10 pps
V3650	Distancia 7	1 a 9999	Pulsos
V3651	Desaceleración 8	4 a 1000	x 10 pps
V3652	Distancia 8	1 a 9999	Pulsos

* Si selecciona el uso de interrupción, el PLC DL06 no busca por el valor de posición blanco hasta que la señal de interrupción X1 se haga verdadera.

**Para configurar un número negativo, coloque 8 en el dígito más importante.
Por ejemplo, -8388608 es 88388608 en V3631 y V3632.

Control de velocidad

Memoria	Función	Rango	Unidades
V3630	Perfil de velocidad	Solamente 2000	–
V3631 / 3632	Selección de dirección	0=CW, 80000000=CCW,	Pulsos
V3633	Selección de dirección	4 a 1000	x 10 pps
V3636	Código de error	(Vea el fin de la sección)	–

Selección del tipo de perfil.

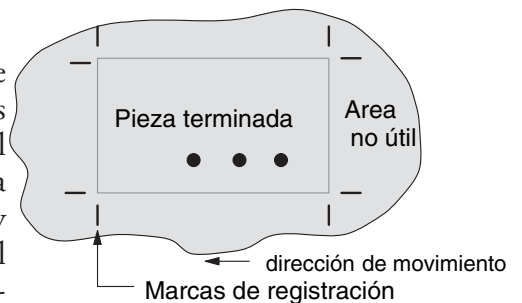
El modo de salida de pulsos genera tres tipos de perfil de movimiento. La mayoría de las aplicaciones usa un tipo solamente para todos los movimientos. Sin embargo cada movimiento puede ser diferente si fuera necesario.

- Trapezoidal automático – rampa de aceleración hasta la velocidad definida y rampa de desaceleración.
- Trapezoidal paso a paso - velocidad hasta control de posición cuando hay una interrupción.
- Control de velocidad - solamente velocidad y dirección.

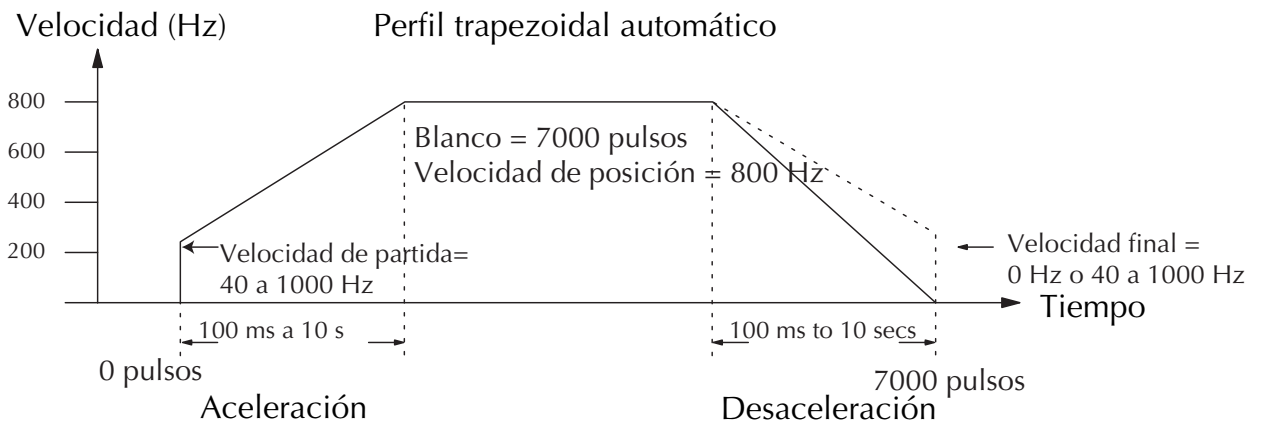
Definición del perfil trapezoidal automático.

Este perfil es el más común de los perfiles de posicionamiento. Mueve la carga (Una herramienta o el material) hasta una posición blanco definida, creando un perfil de movimiento. La rampa de aceleración es aplicada en el punto de partida. La rampa de desaceleración es aplicada retrocediendo desde la posición blanco. El resto del movimiento es usado viajando a la velocidad definida.

Los perfiles de registración resuelven una clase de problemas de control de movimiento. En algunas aplicaciones, el material de trabajo se mueve pasando al frente de una herramienta, como por ejemplo, una estación de taladrado. En la figura adyacente hay marcas de registración en el área de removido del material de trabajo, para permitir a la máquina-herramienta registrar su posición relativa al rectángulo, para perforar adecuadamente.



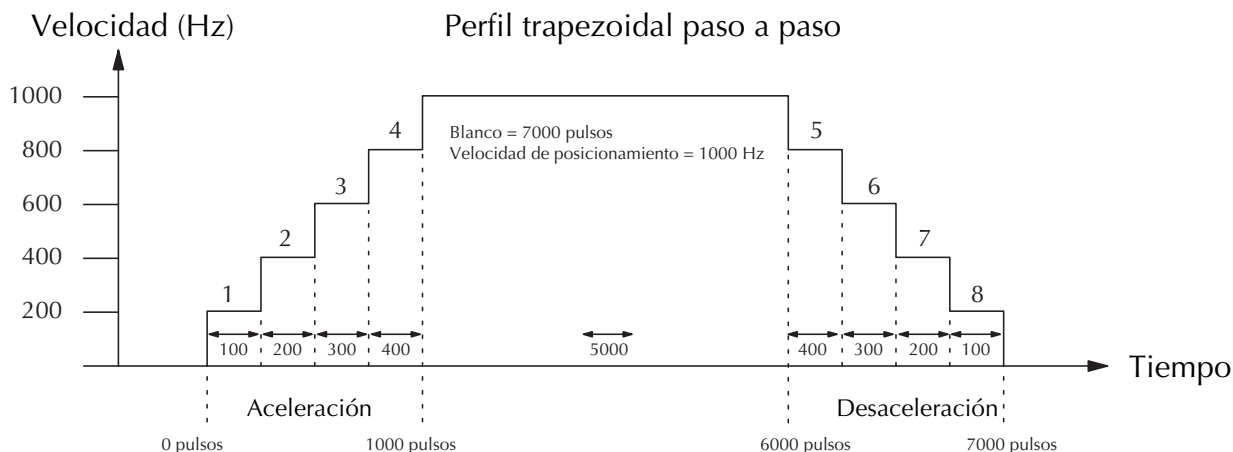
Los movimientos de búsqueda de la posición inicial (Home en inglés) permite que los sistemas de lazo abierto puedan recalibrar el valor corriente durante una energización.



El usuario determina la velocidad de partida, los tiempos de aceleración y desaceleración y el número total de pulsos. La CPU computa el perfil con esas entradas.

Definición del perfil trapezoidal paso a paso

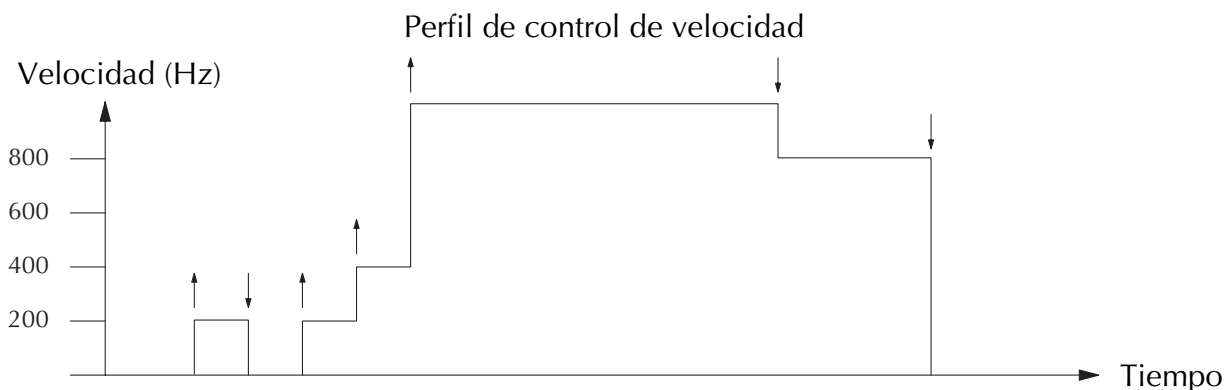
Los perfiles de registración son una combinación de modos de control de velocidad y de posición. El movimiento se inicia acelerando a una velocidad programada. La velocidad se mantiene y el movimiento es de duración indefinida. Cuando una señal de interrupción se torna verdadera (debido a detección de la registración) el perfil pasa de control de velocidad para posición. El movimiento termina a una distancia predeterminada pasado del punto de interrupción (Por ejemplo, una localización de un agujero de taladrado). La rampa de desaceleración es aplicada antes de la posición "blanco".



Defina los pasos 1 hasta 4 para aceleración gradual hasta la velocidad definida y defina los pasos 5 hasta 8 para desaceleración desde la velocidad definida. Este tipo de perfil es apropiado para aplicaciones que usan motores paso a paso de gran tamaño o cargas de mucha inercia. Sin embargo, puede ser usado para suministrar rampas en usos de motores y cargas pequeñas.

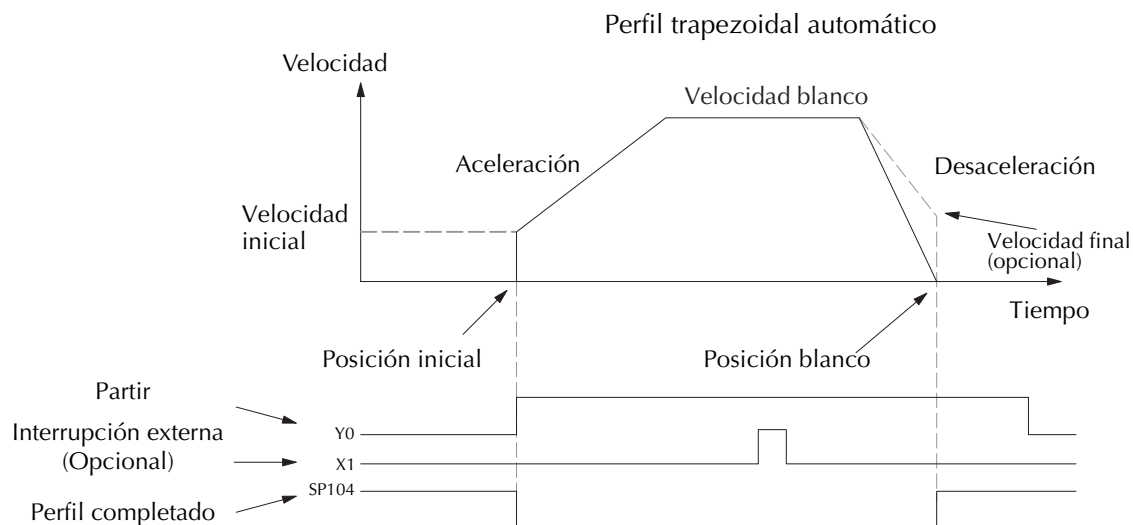
Definición de control de velocidad

En el control de velocidad se define solamente la dirección y velocidad del movimiento. No hay una posición blanco especificada de modo que el movimiento puede ser tan largo como se desee. Se define solamente el primer valor de velocidad. Los valores restantes de velocidad pueden ser cambiados mientras el movimiento está progresando. En el diagrama del perfil de abajo, las flechas indican los cambios de velocidad.



Operación del perfil trapezoidal automático

Las velocidades de inicio deben estar en el rango de 40 hasta 1000 pulsos por segundo. El resto de los parámetros del perfil están en la tabla de parámetros del perfil.



Los gráficos de señal debajo de la línea del tiempo del perfil indica el orden de eventos. El sistema usa la salida Y0 como entrada de "partir" el tren de pulsos, que inicia el movimiento siguiendo el perfil. Inmediatamente el sistema coloca en 0 la señal de perfil completado (SP104), de modo que programa ladder pueda supervisar el progreso del movimiento. Típicamente un programa ladder supervisará este contacto de modo que sepa cuando se va a iniciar el próximo movimiento del perfil.

Usted también puede usar la interrupción externa (X1). Cuando se selecciona la interrupción interna para el perfil, el PLC DL06 sigue generando pulsos hasta que X1 se haga verdadera. Luego el PLC DL06 genera pulsos definidos en la posición blanco.

Si usted tiene familiaridad con control de movimiento usted nota de que no tenemos que especificar la dirección del movimiento. La función examina la posición blanco relativa a la posición corriente y automáticamente genera las salidas en la dirección correcta al accionamiento del motor.

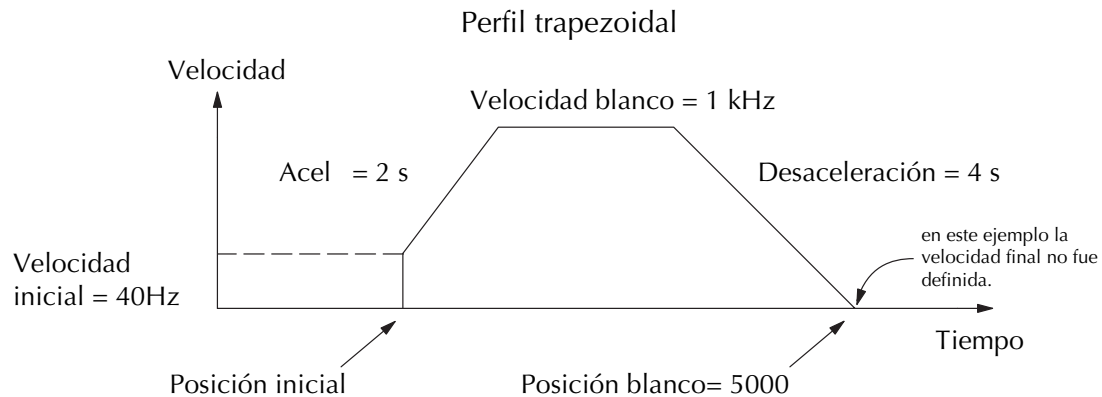
Note que el movimiento se acelera inmediatamente a la velocidad de partida. Este segmento es útil en sistemas paso a paso de modo que podamos saltar áreas de baja velocidad cuando haya problemas de torque bajo o de resonancia en el motor, lo que podría causar un bloqueo al motor. (Cuando un motor paso a paso se para, perdemos la posición de la carga en sistemas de posicionamiento de lazo abierto).

Sin embargo, es preferible no hacer la velocidad de partida muy grande porque el motor paso a paso también puede saltar algunos pulsos debido a la inercia del sistema. Usted puede también configurar la velocidad final por la misma razón.

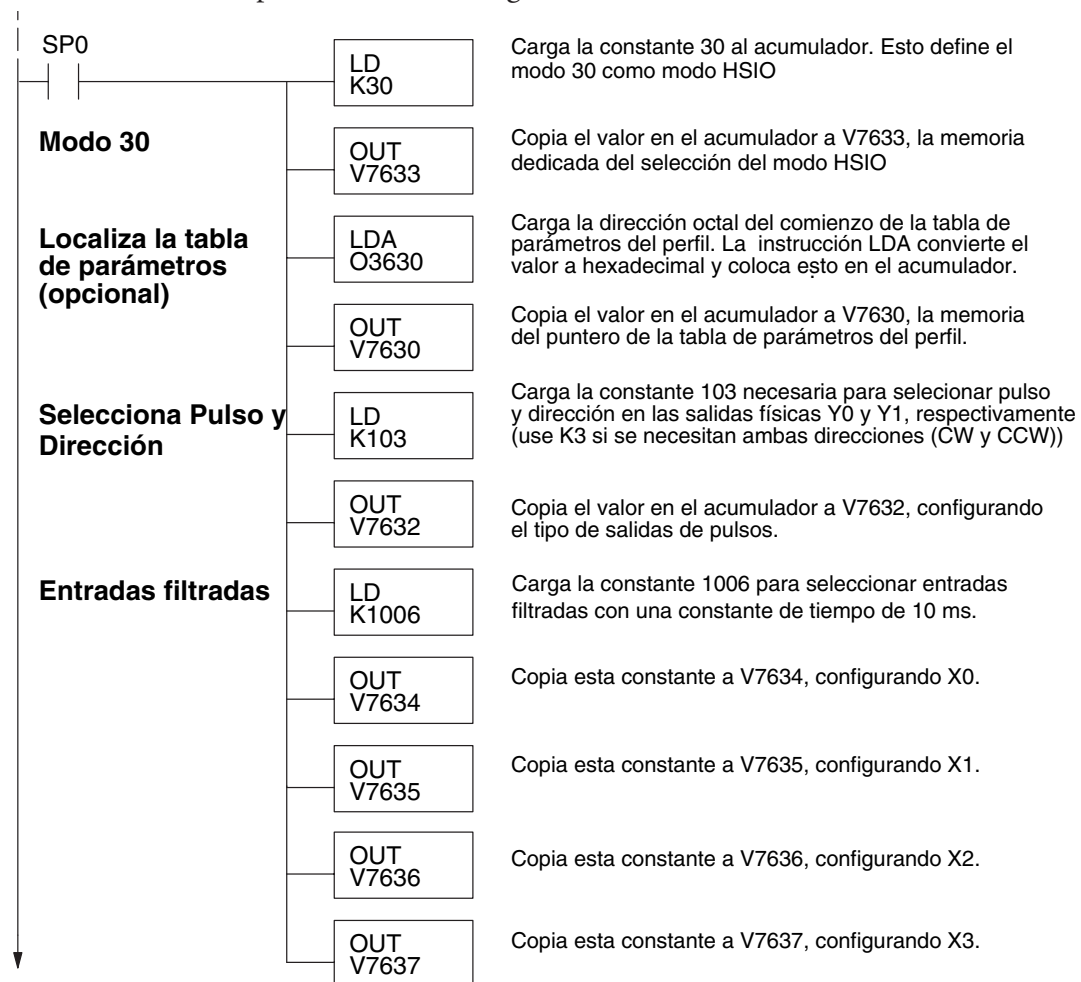
Cuando se necesite cambiar el valor de posición corriente use la salida lógica Y1 para cargar un nuevo valor en el contador HSIO. Si el programa carga un nuevo valor en CT174/CT175 (V1174/V1175), entonces activando Y1 copiará aquel valor en el contador HSIO. Esto debe suceder antes que el perfil de movimiento comience, porque el sistema ignora Y1 durante el movimiento.

Ejemplo 1 modo 30: Perfil trapezoidal automático sin interrupción externa

El perfil que queremos mostrar es dibujado en la siguiente figura. Consiste de una velocidad de inicio diferente de 0 y una velocidad definida moderada.

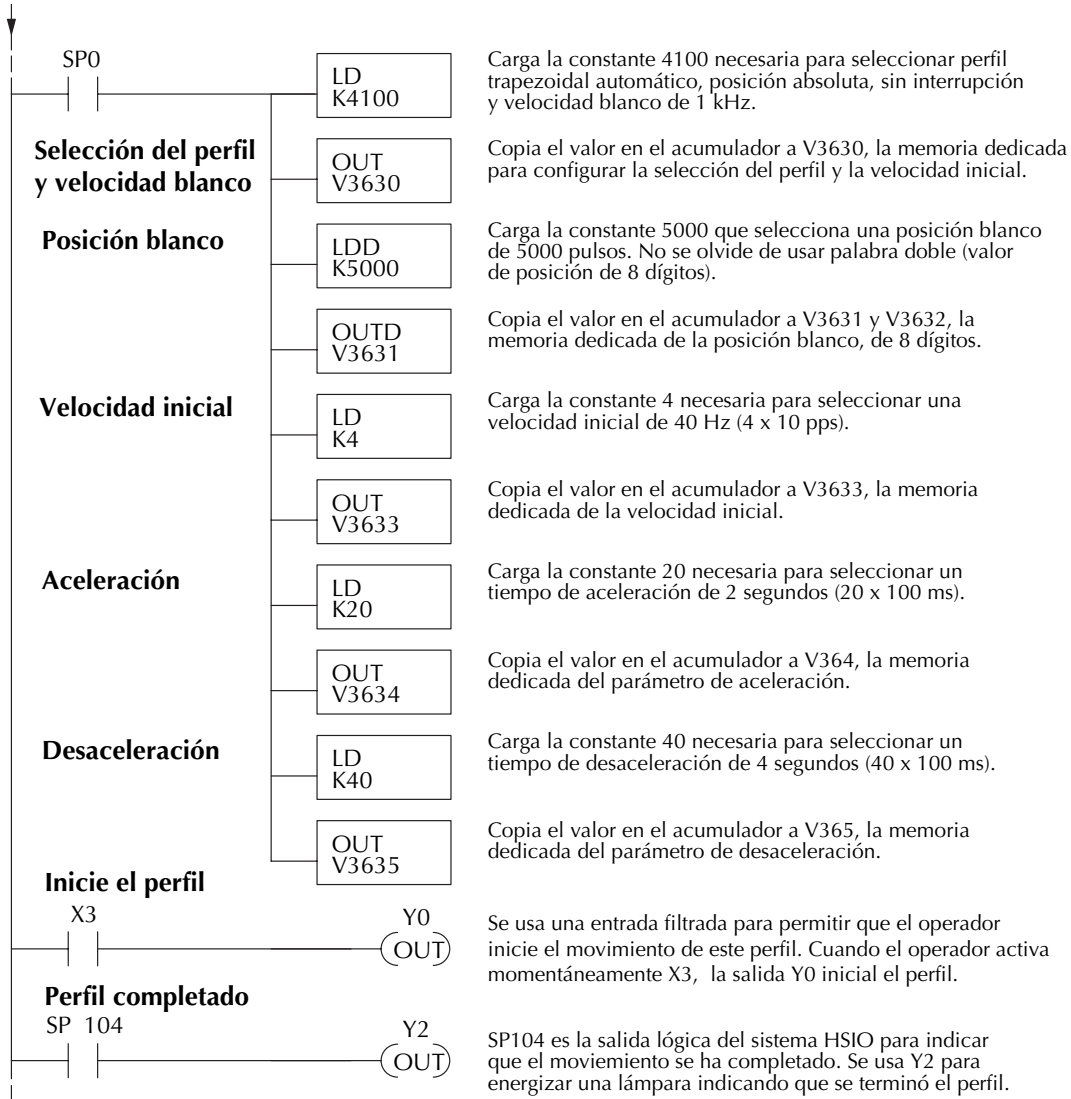


El siguiente programa ejecutará el movimiento del perfil en la figura anterior, cuando sea ejecutado. El comienzo del programa contiene todos los parámetros de configuración necesarios para el modo 30. Solamente tenemos que hacer esto una vez en el programa de modo que usamos el contacto SP0 para definir la configuración.



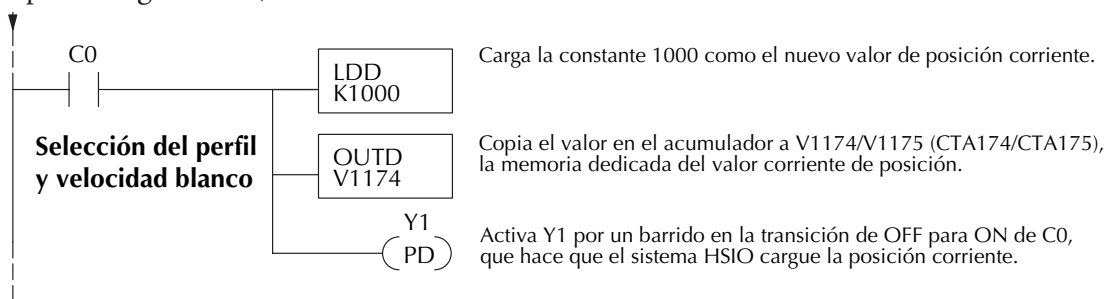
Continúa en la próxima página.

continuado desde la página anterior.



Cambie el valor de posición en cualquier momento

En cualquier momento se puede escribir una posición nueva en el valor corriente de posición. Esto a menudo se hace después que una búsqueda del punto inicial (HOME) (vea los programas de ejemplo de registración).

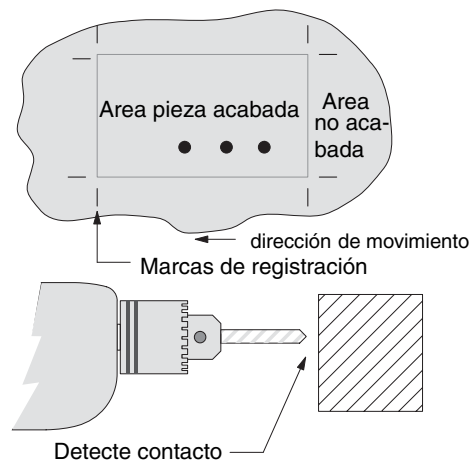


Ejemplo 2 modo 30 : Perfil trapezoidal automático con interrupción externa

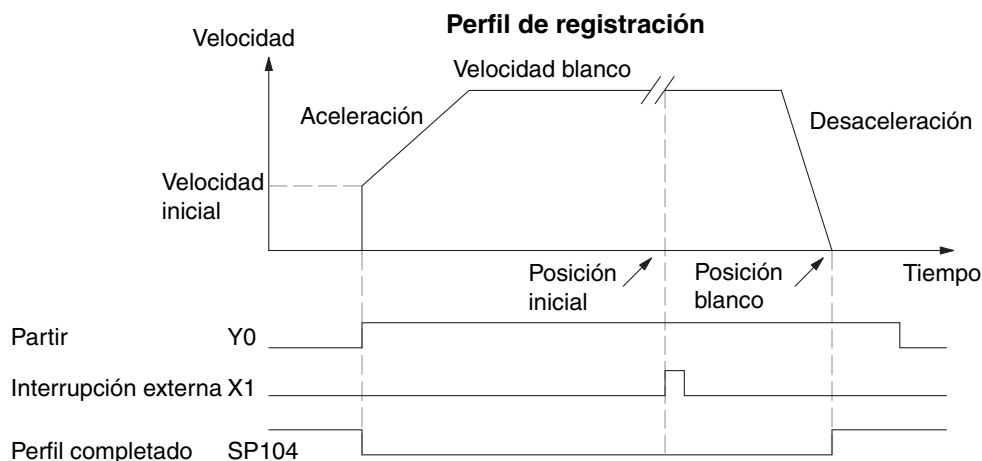
Aplicaciones de registración:

1. En una aplicación típica mostrada a la derecha, el producto bruto siendo trabajado se mueve pasando al frente de una herramienta de trabajo, tal como un taladro. Las marcas de registración en el área no útil del material de trabajo le permite a una máquina-herramienta que determine su posición relativa al rectángulo, para taladrar apropiadamente.

2. En otros ejemplos de registración, el material de trabajo es estacionario y la herramienta se mueve. Un taladro puede acercarse a la superficie de una parte en el trabajo, preparandose para taladrar un hoyo de profundidad definida. Sin embargo, la longitud de la broca del taladro disminuye gradualmente debido al uso de la herramienta. Un método para eliminar esto es detectar el momento del contacto con la superficie de la parte en cada taladrado, moviendo la broca en la pieza una distancia constante después del contacto. El contacto de detección del área de la pieza acabada marca la dirección del movimiento.



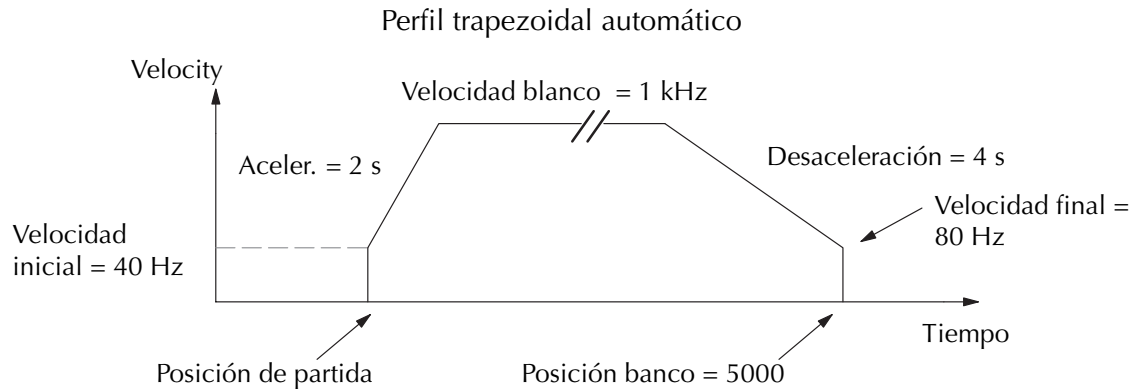
3. El movimiento de búsqueda de Home permite que un sistema de movimiento calibre su posición al inicio. En este caso, el sistema de posicionamiento hace un movimiento indefinido y espera que la pieza pase un interruptor límite. Esto crea una interrupción cuando la pieza esté en una posición conocida. Luego se para el movimiento y se cambia el valor de la posición con un número que es igual a la posición física de "Home". Cuando ocurre un pulso de interrupción en la entrada X1 física, la posición inicial es definida como el valor corriente de conteo (la posición corriente). El modo de control de velocidad cambia para control de posición, moviendo la pieza a la posición de "blanco". Note que la velocidad inicial mínima es 40 pulsos por segundo. Esta velocidad instantánea permite el uso de motores de paso a paso que pueden atascarse en velocidades bajas.



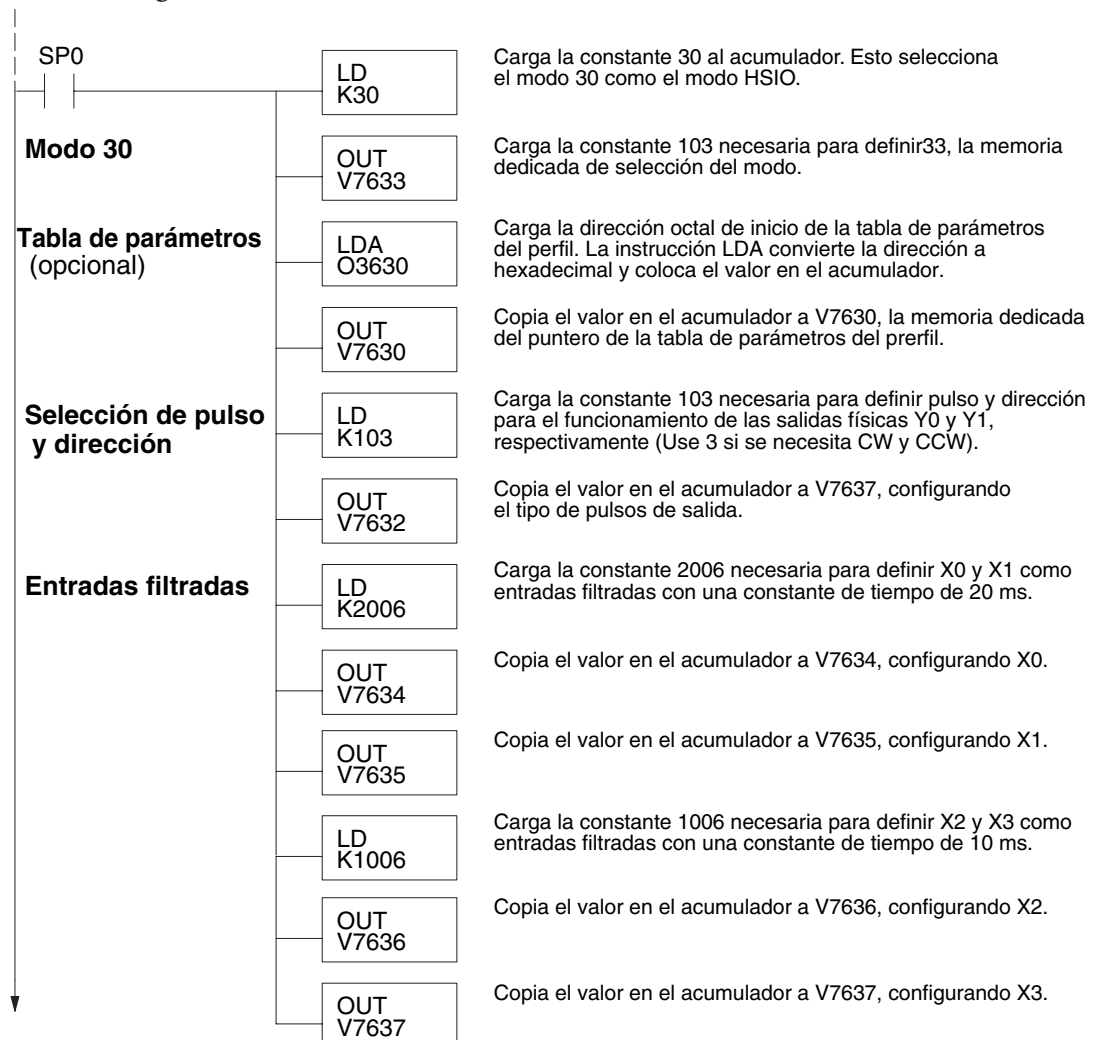
Los gráficos de comportamiento de las señales en relación al tiempo debajo del perfil indican el orden de eventos. La CPU usa la salida Y0 lógica para comenzar el perfil. Inmediatamente el sistema de entradas y salidas de alta velocidad apaga la señal de Perfil Completado (SP104), de modo que el programa ladder pueda supervisar el fin del movimiento sintiendo la señal en estado verdadero u ON.

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

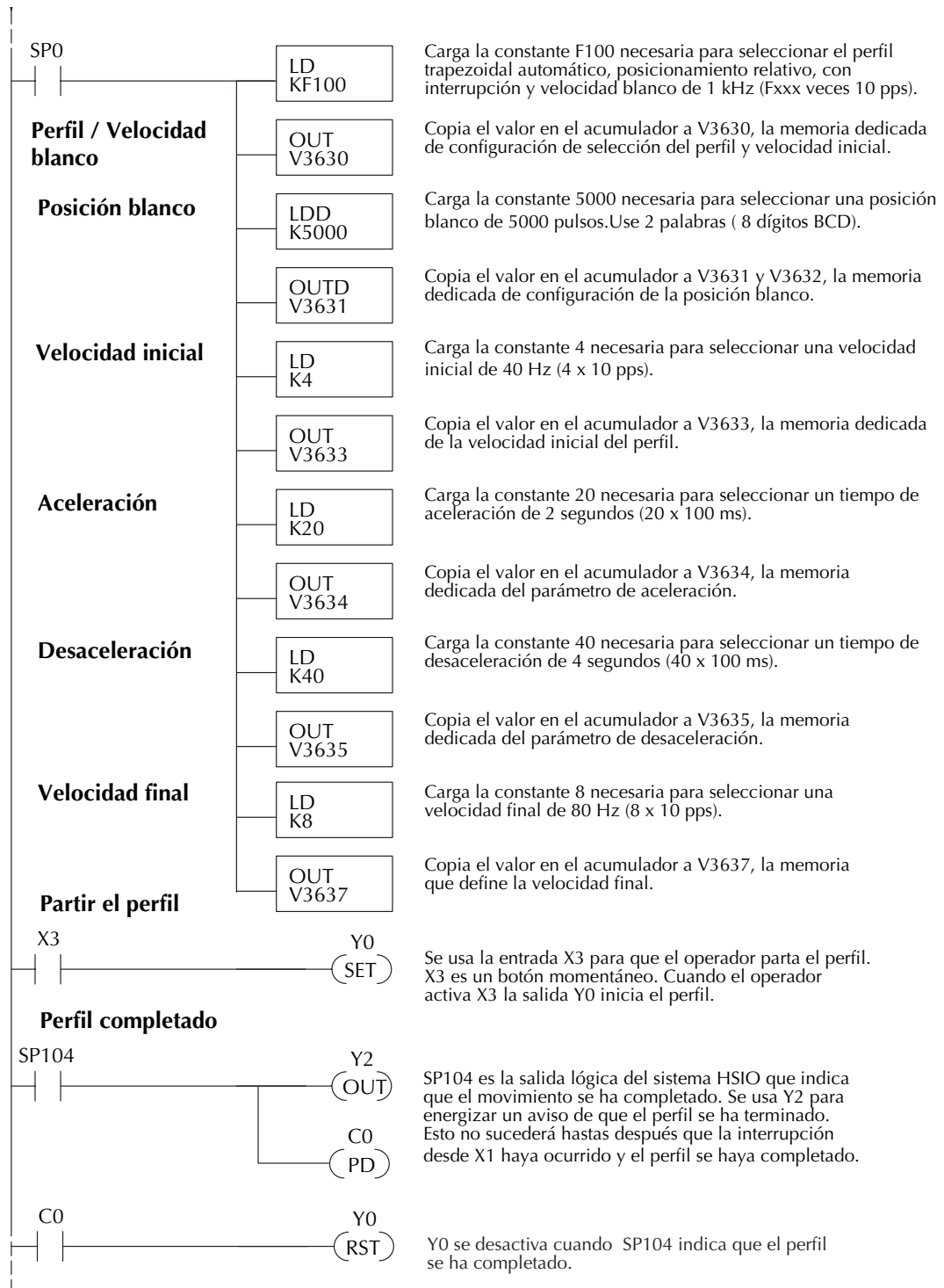
El perfil trapezoidal automático que queremos realizar está mostrado en la figura siguiente. Se compone de una velocidad inicial diferente de cero y una velocidad definida moderada.



El programa siguiente implementará el perfil mostrado, cuando sea ejecutado. El primer renglón del programa contiene todos los parámetros necesarios de configuración. Se hace una vez solamente en el programa, de modo que usamos el contacto SP0 del primer barrido para ejecutar la configuración.



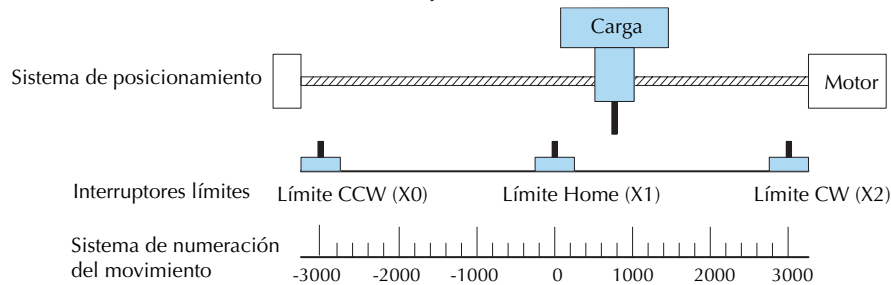
Continúa desde la página anterior



El perfil comenzará cuando la entrada de inicio (X3) se hace verdadera. Luego comienza un movimiento indefinido que dura hasta que una interrupción externa X1 ocurra. Entonces el movimiento continúa por 5000 pulsos más antes de parar.

Ejemplo 3 modo 30 : Perfil trapezoidal automático con búsqueda de "Home"

Uno de los aspectos más difíciles del control del movimiento es el establecimiento de la posición verdadera durante la energización. Esto es especialmente cierto para sistemas de lazo abierto que no tienen un sensor de realimentación de posición. Sin embargo, un interruptor de límite localizado en una ubicación exacta en el mecanismo posicionador puede informar la realimentación de la posición en un punto. Para la mayoría de los sistemas de controles paso a paso, este método es una solución buena y económica.

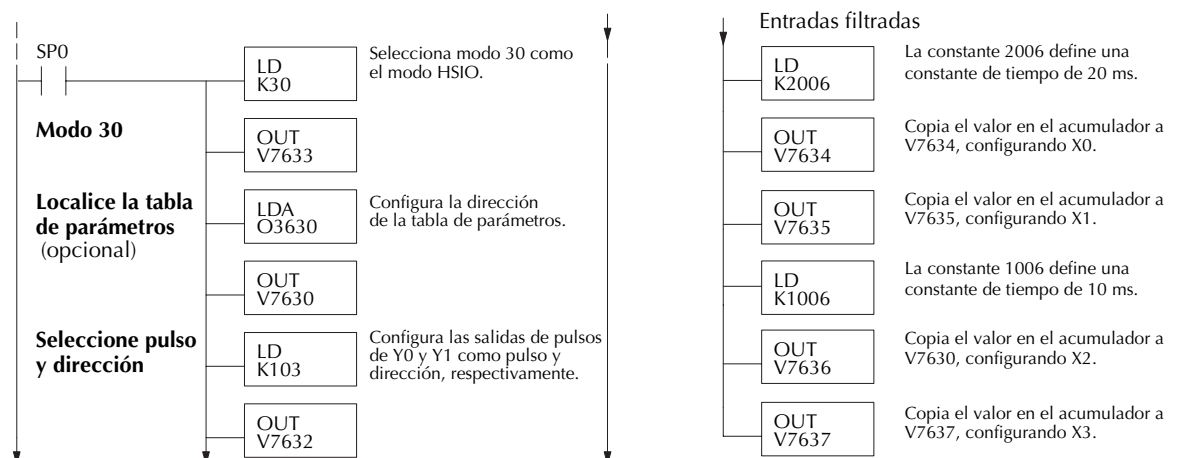


En el dibujo de arriba, la carga se mueve a la izquierda o a la derecha dependiendo de la dirección CW/CCW (a favor o en contra de los punteros del reloj) de la rotación del motor. El programa ladder del PLC recibe las señales de interruptores de límite de CW y CCW para parar el motor, antes que la pieza se salte fuera de posición y dañe la máquina. El interruptor límite "Home" se usa en energización para establecer la posición verdadera. El sistema numérico es arbitrario, dependiendo de las unidades de ingeniería de una máquina.

En la energización, no se sabe si la pieza está localizada a la izquierda o a la derecha del interruptor de límite HOME. Por lo tanto, se iniciará un perfil buscador del HOME, usando el modo de registración. El interruptor de límite home es cableado a X1, causando una interrupción. Escogemos una dirección inicial arbitraria de la búsqueda, instalándose en la dirección CW (izquierda a derecha).

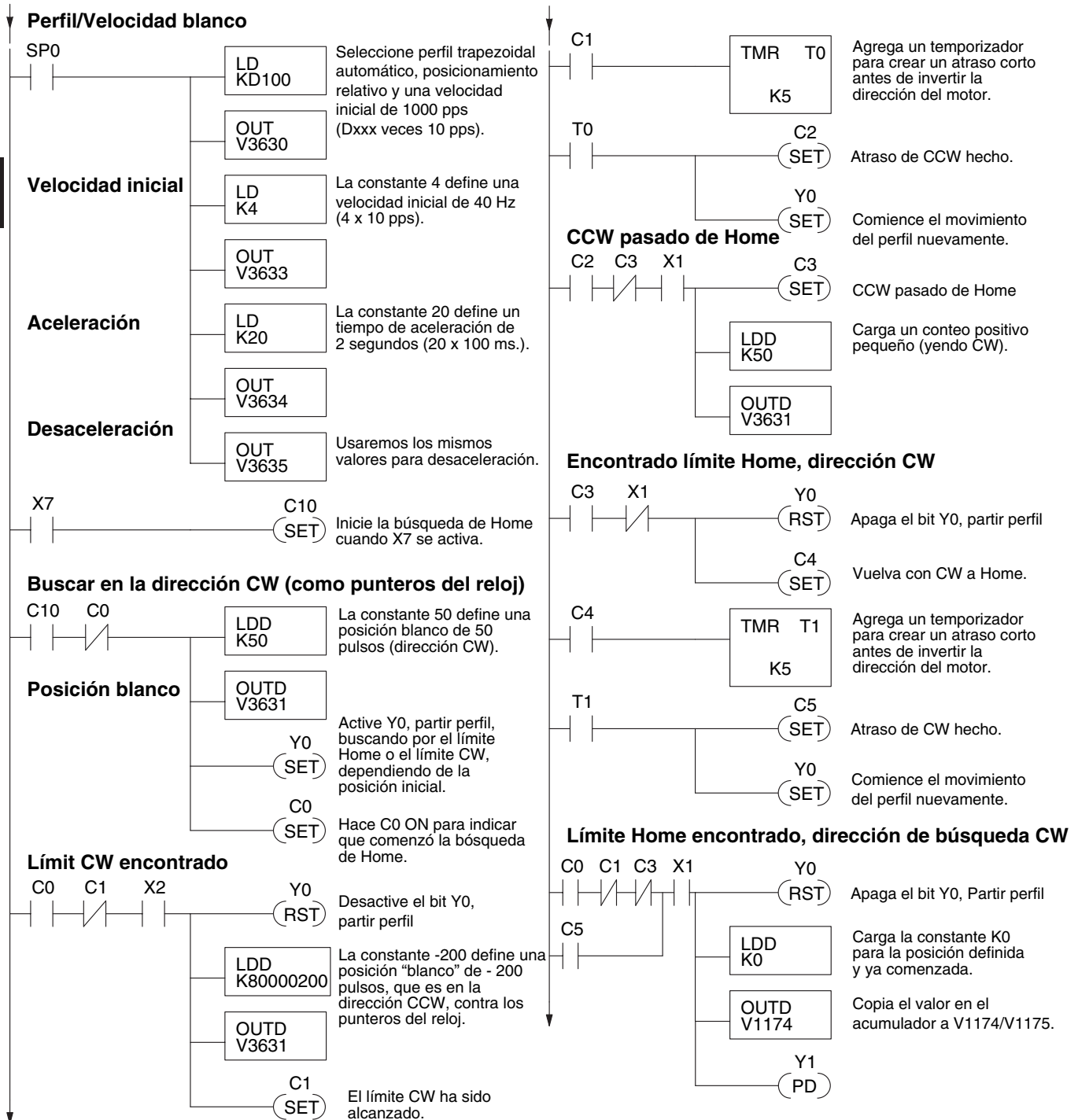
- Si el interruptor límite Home se cierra primero, entonces se para y define la posición (este valor es típicamente "0", pero puede ser diferente si es preferible).
- Sin embargo, si el interruptor de límite de CW se cierra primero, se debe invertir el motor y mover hasta que el interruptor de límite HOME cierre, parando exactamente al pasar.

¡En el caso último, se repite el primer movimiento, porque siempre se necesita hacer la aproximación final al interruptor de límite HOME *desde la misma dirección*, para que la posición física final sea la misma en cualquier caso!



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

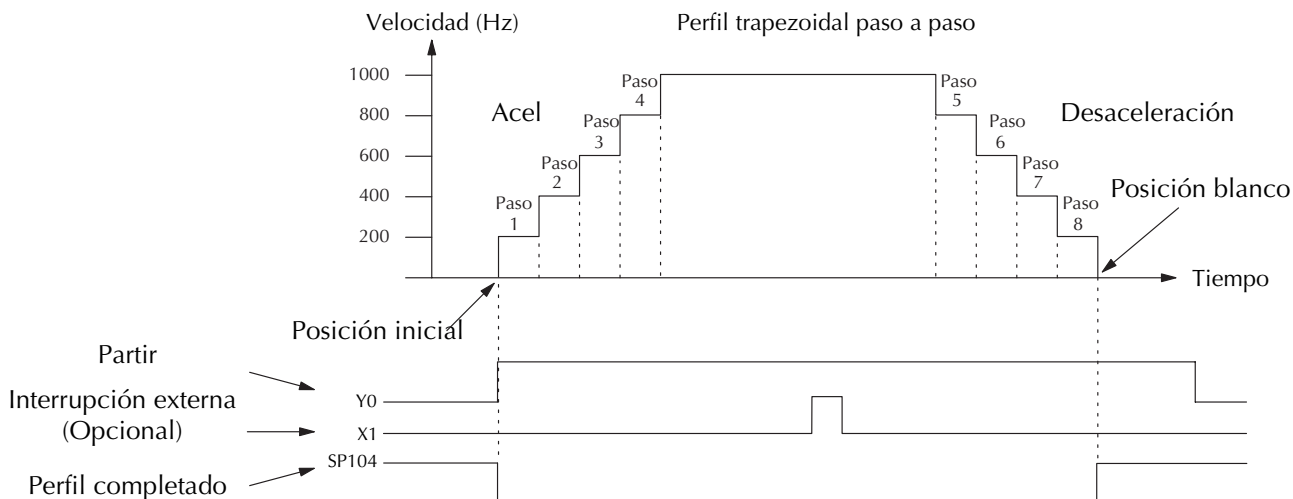
E



El perfil buscador de HOME ejecutará partes específicas del programa, basado en la orden de detección de los interruptores límite. La lógica ladder configura C0 verdadera para iniciar una búsqueda de HOME en la dirección CW. Si se encuentra el límite CW, el programa busca el HOME en la dirección CCW, pasa levemente el interruptor límite Home y hace la búsqueda final CW del HOME. Después de alcanzar la posición HOME, el último renglón cambia la posición corriente a "0".

Operación del perfil trapezoidal paso a paso.

Con el perfil trapezoidal paso a paso, usted puede controlar las rampas de aceleración y desaceleración como se quiera.



Los gráficos de comportamiento de las señales en relación al tiempo debajo del perfil indica el orden de eventos. La CPU usa la salida Y0 lógica para comenzar el perfil. Inmediatamente el sistema de entradas y salidas de alta velocidad apaga la señal de Perfil Completado (SP104), de modo que el programa ladder pueda supervisar el fin del movimiento sintiendo la señal en estado verdadero u ON.

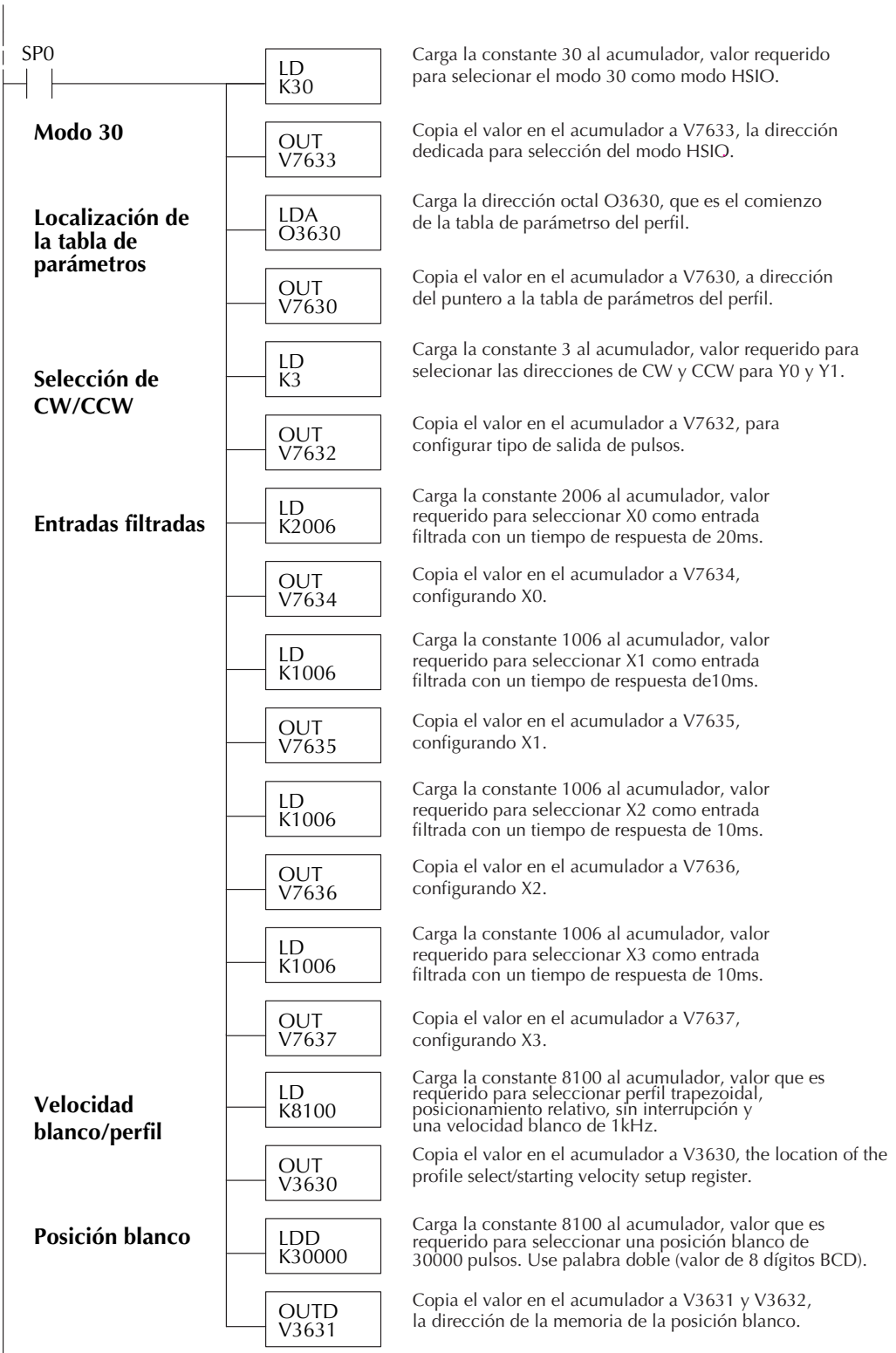
Típicamente, un programa ladder supervisará el estado de este relevador para que sepa cuando iniciar el próximo movimiento del perfil. Puede usar también la interrupción externa (X1). Una vez escogida la característica externa de interrupción para el perfil, el DL06 se mantiene generando los pulsos hasta que X1 se haga verdadero u ON. Luego el DL06 sigue generando pulsos hasta el valor definido como posición “blanco”.

Cada rampa de aceleración y desaceleración se compone de 4 pasos. Se puede establecer la velocidad y la distancia (el número de pulsos) de cada paso. No se necesita usar los 4 pasos de cada rampa.

Por ejemplo, si quiere usar sólo 2 pasos, haga cero a la velocidad y la distancia del paso de 3 y 4. Si la rampa de aceleración y de desaceleración son idénticas, puede hacer cero en todas las velocidades y los parámetros de distancia para la rampa de desaceleración.

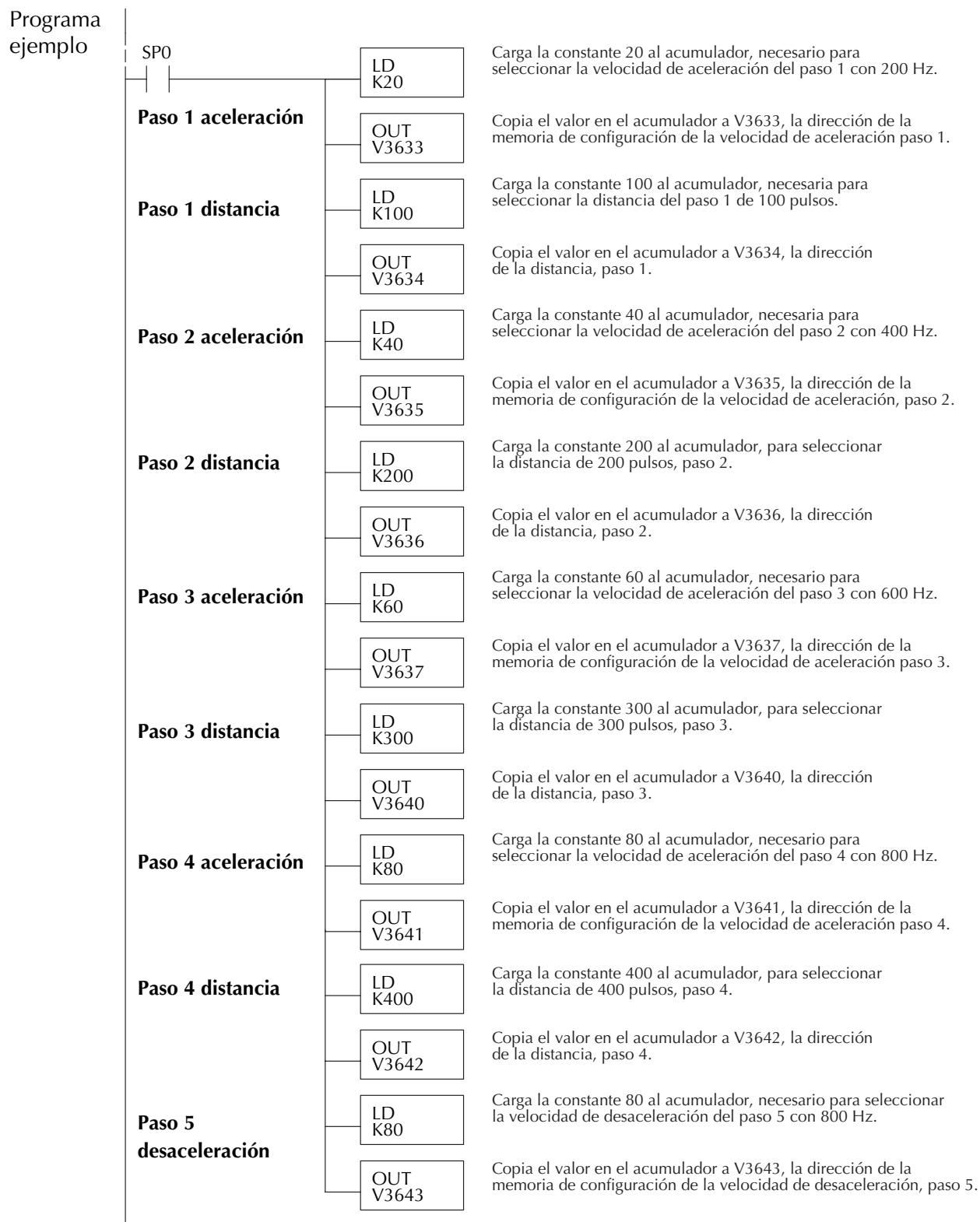
Ejemplo Programa 4 modo 30: Perfil trapezoidal paso a paso

Programa
Ejemplo



Continúa en la próxima página

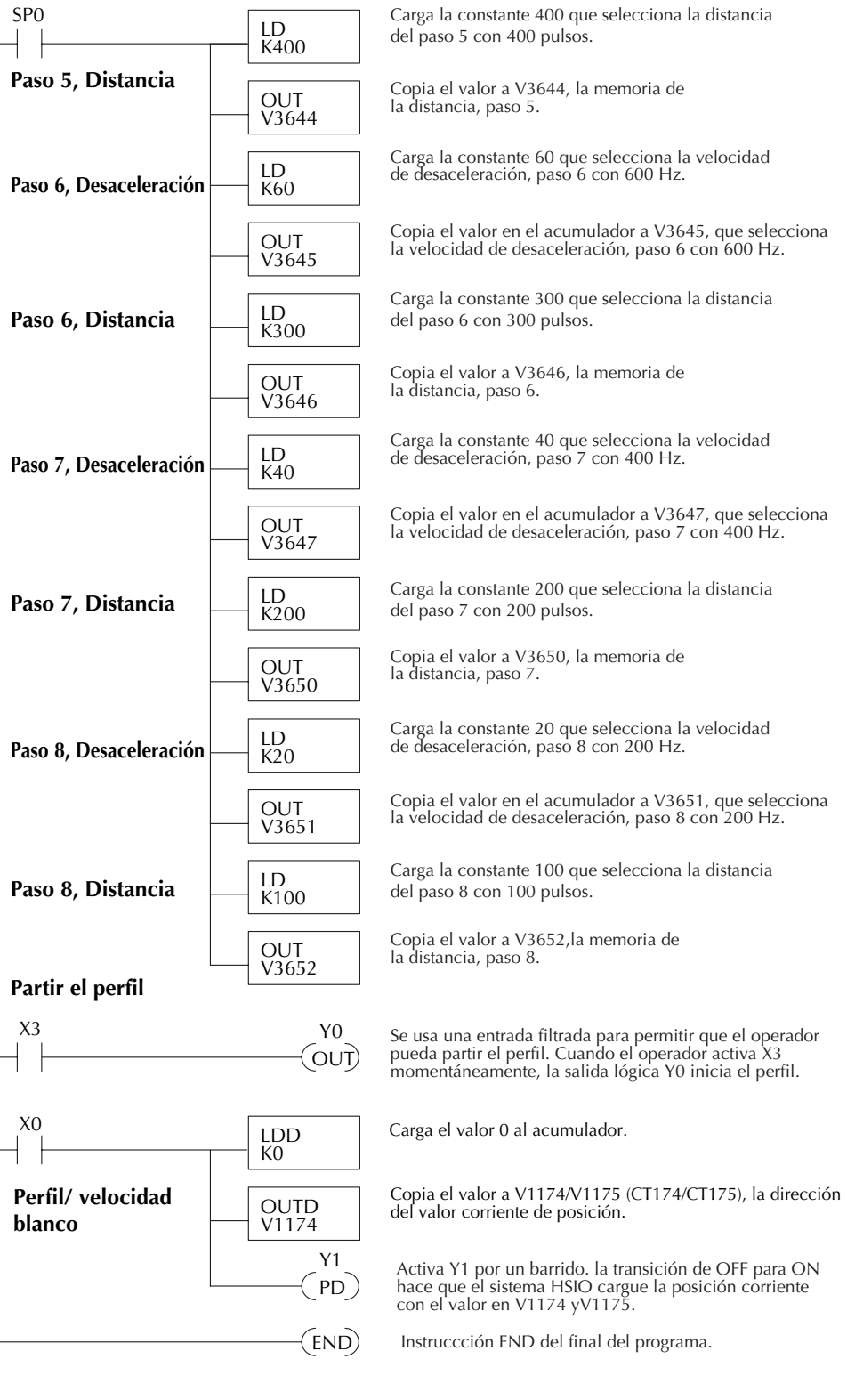
Continúa desde la página anterior



Continúa en la próxima página

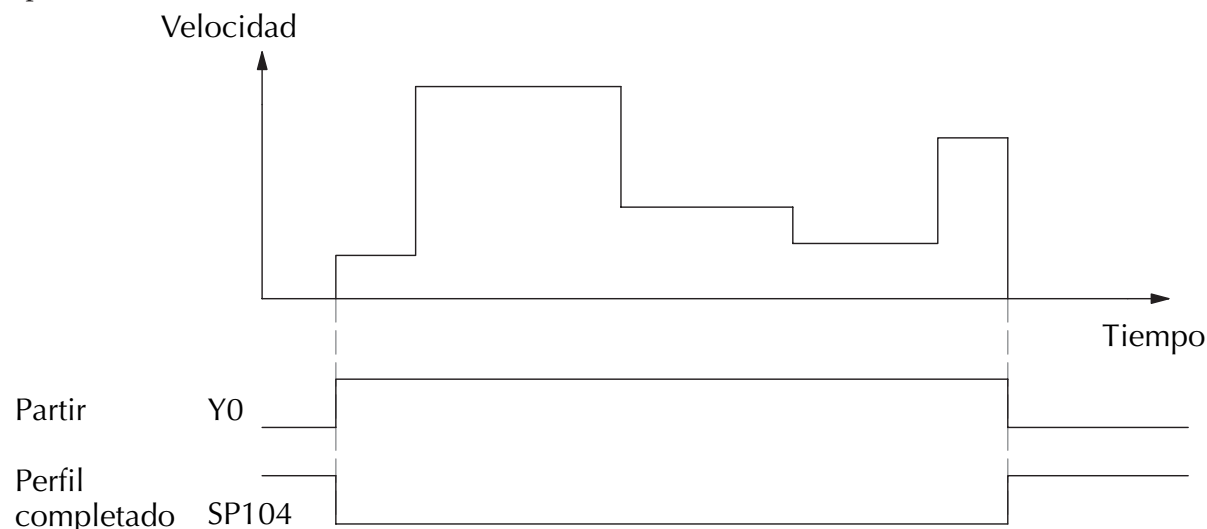
Continúa desde la página anterior

**Programa
Ejemplo**



Operación del perfil de velocidad

El perfil de velocidad se acomoda mejor a aplicaciones que envuelven movimientos pero no requieren moverse a puntos específicos. El control de velocidad de un transportador es un ejemplo típico.



El gráfico de señales debajo del perfil indica el orden de eventos. Asumiendo que la velocidad es mayor que cero, el movimiento comienza cuando la entrada de inicio (Y0) se activa. Ya que no hay una posición “blanco” final, el perfil se considera en progreso cuando la entrada Y0 (Partir) permanece activa. La entrada lógica SP104 (perfil completado) se relaciona directamente a la lógica (X0 ladder) al estado de la entrada Partir (Y0) cuando se usan perfiles de velocidad.

Mientras la entrada Partir (Y0) esté activa, el programa ladder puede comandar un cambio de velocidad escribiendo un valor nuevo al registro de velocidad (V3633 por defecto).

Está disponible el rango completo de velocidad de 40 Hz hasta 10 kHz. Note del diagrama que no hay rampas de aceleración ni desaceleración entre actualizaciones de velocidad. Así es cómo el perfil de velocidad trabaja con el sistema HSIO. Sin embargo, el programa ladder puede comandar un cambio en velocidad más gradual incrementando o decrementando el valor de velocidad más lentamente. Un contador o un temporizador pueden ser útiles para crear rampas de aceleración y desaceleración.

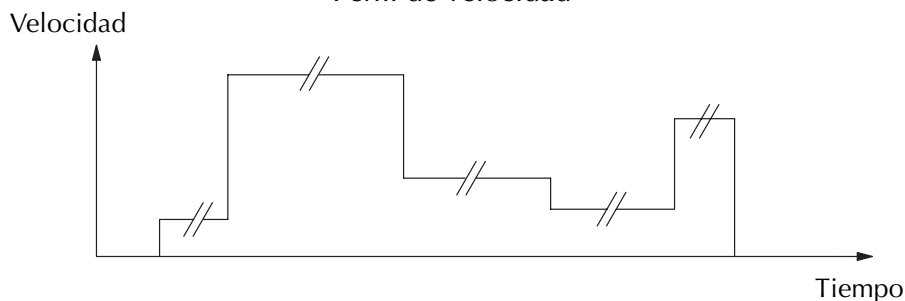
A menos que la carga deba hacer un movimiento muy complejo, es más fácil hacer que la función HSIO engendre rampas de aceleración y desaceleración escogiendo los perfiles trapecoidal o de registración en lugar de usar este perfil.

En forma diferente que en los perfiles trapecoidal y de registración, con perfiles de velocidad se debe especificar la dirección deseada del movimiento. Cargue la memoria de dirección (V3631/V3632 por defecto) con 8000 0000 hexadecimal para la dirección CCW, o 0 para la dirección de CW.

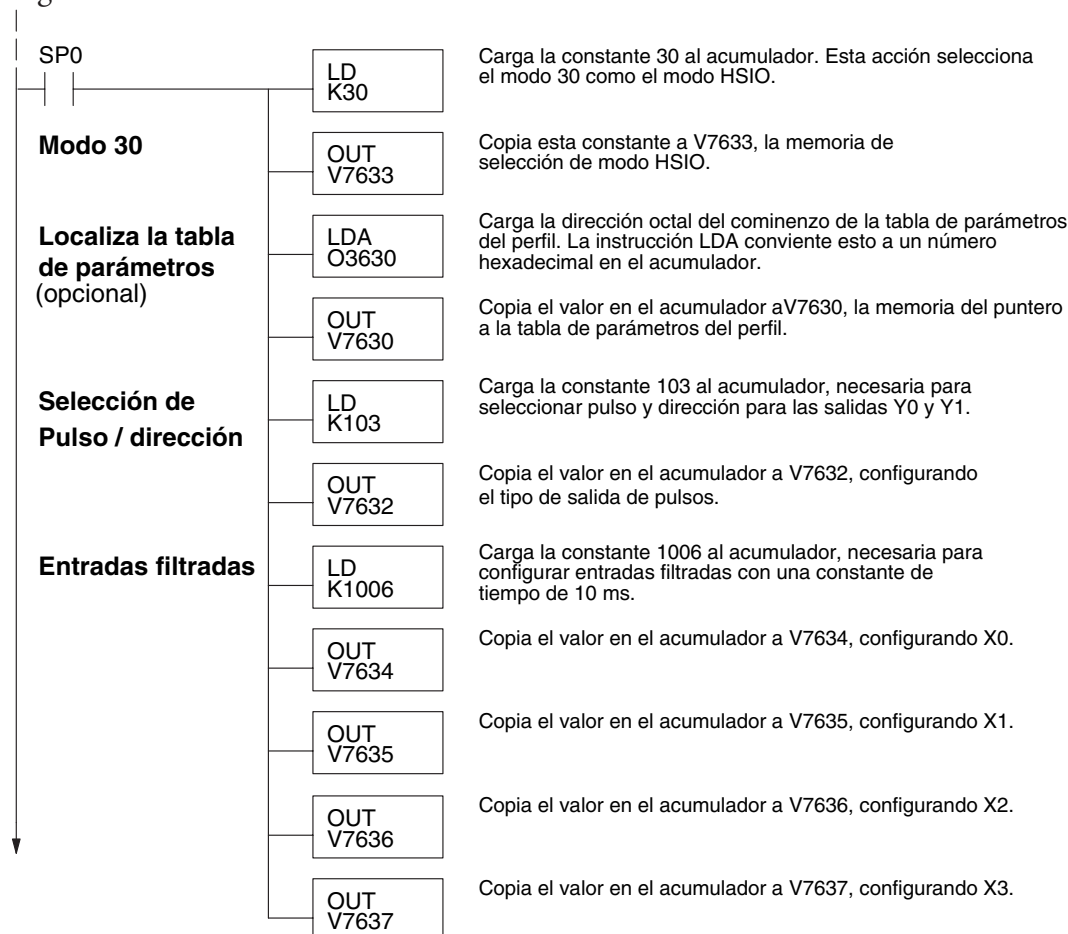
Programa Ejemplo 5 modo 30: Perfil de velocidad

El perfil de velocidad que queremos realizar es mostrado y marcado en la figura siguiente. Cada segmento de velocidad es de longitud indefinida. La velocidad sólo cambia cuándo la lógica ladder (u otro aparato que escriba en la memoria V) actualiza el parámetro de velocidad.

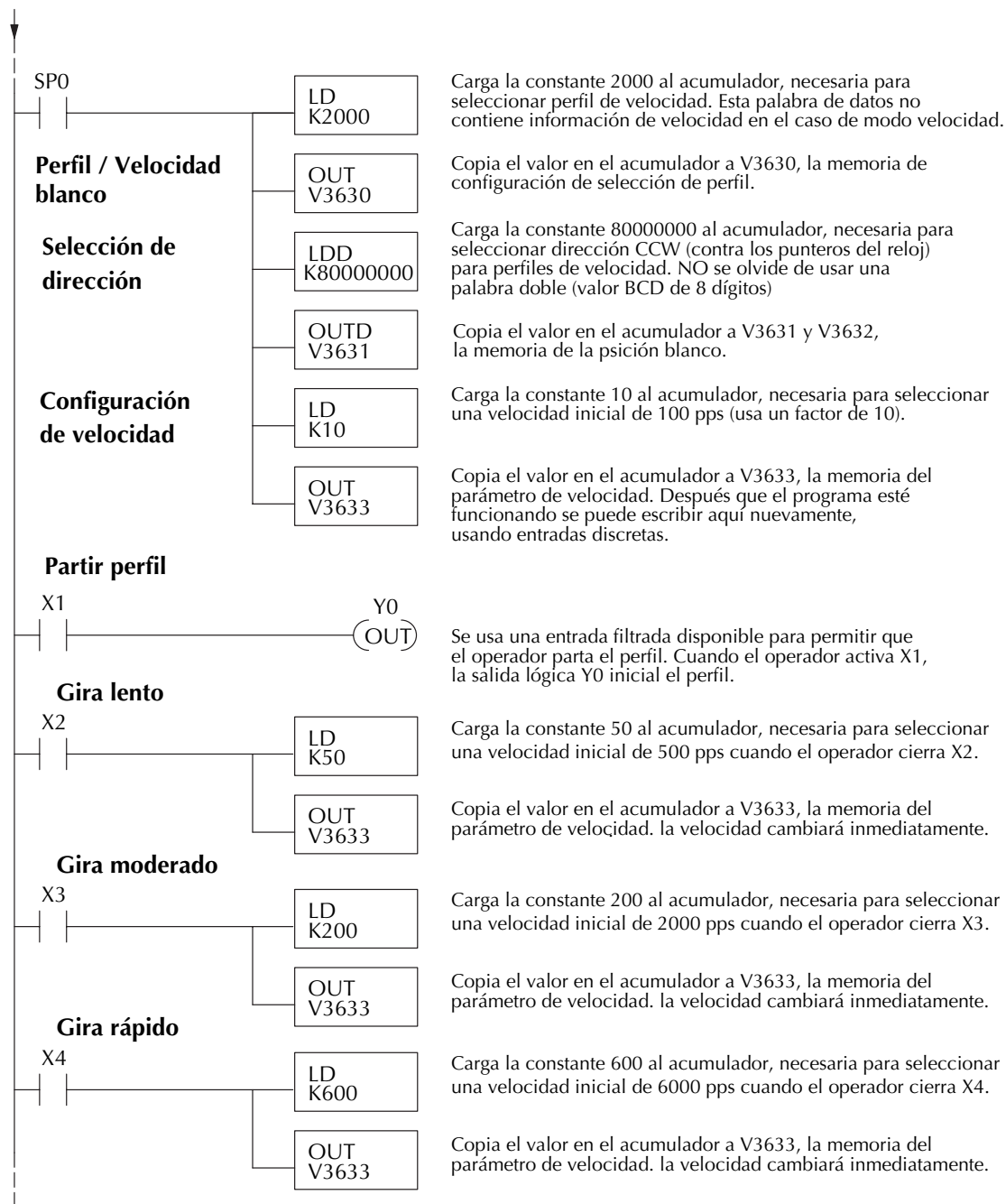
Perfil de velocidad



El siguiente programa usa entradas discretas dedicadas para definir o cargar nuevos valores de velocidad. Es instructivo tratar de hacer este programa porque se pueden crear una gran variedad de perfiles con solamente dos o tres entradas discretas. Se trata de activar solamente una de las entradas X2, X3 o X4 por vez. El comienzo del programa contiene todos los parámetros de configuración para el modo 30 de salida de pulsos. Se hace esto solamente una vez en el programa, de modo que usamos el contacto SP0 que se activa en el primer barrido para definir la configuración.



Programa ejemplo continuado



E

Códigos de error del perfil trapezoidal automático

La tabla de parámetros del perfil comenzando en V3630 (la dirección por defecto) define el perfil. Ciertos números tendrán como resultado un error cuando el sistema de entradas y salidas de alta velocidad trate de usar los parámetros para ejecutar un perfil de movimiento. Cuando un error ocurre, el sistema de entradas y salidas de alta velocidad escribe un código de error en V3636.

La mayoría de los errores pueden ser corregidos examinando los valores de la tabla de parámetros del perfil. El código de error se limpia automáticamente en la energización o en la transición del modo de Program a RUN.

Código de error	Descripción del error
0000	No hay error
0010	El código de tipo de perfil es inválido (debe ser 4,5,6,C, D, E,F)
0020	La velocidad blanco no es declarada en BCD
0021	La velocidad blanco es definida menor que 40 Hz
0022	La velocidad blanco es definida mayor que 10000 Hz
0030	El valor de posición blanco no es declarado en BCD
0032	La selección de dirección no es 0 o 80000000
0040	La velocidad inicial no está declarada en BCD
0041	La velocidad inicial es definida menor que 40 pps
0042	La velocidad inicial es definida mayor que 1000 pps
0050	El tiempo de aceleración no está en BCD
0051	El tiempo de aceleración es 0
0052	El tiempo de aceleración es mayor que 10 segundos
0060	El tiempo de desaceleración no está en BCD
0061	El tiempo de desaceleración es 0
0062	El tiempo de desaceleración es mayor que 10 segundos

Búsqueda de problemas para el modo 30

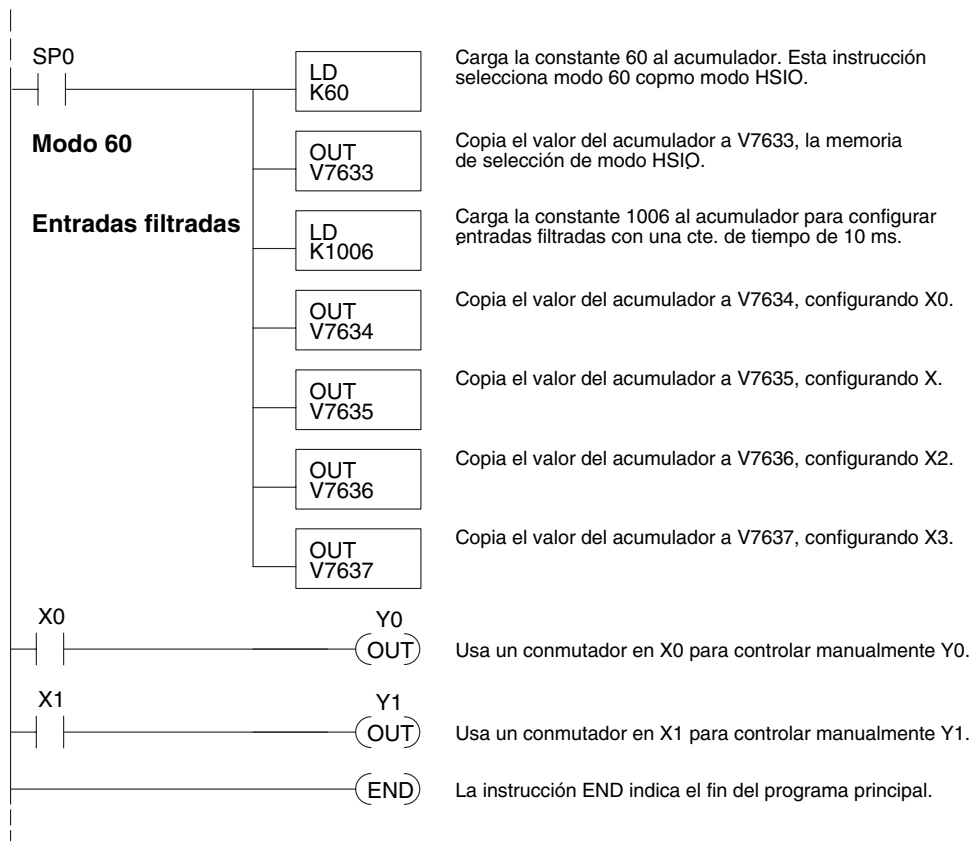
Si usted tiene problemas con la operación del modo 30, estudie por favor los síntomas siguientes y las causas posibles. Los problemas más comunes se listan abajo:

Síntoma: El motor de paso a paso no gira

Causas posibles: :

1. **Configuración** – Verifique que el sistema de entradas y salidas de alta velocidad realmente genera pulsos en las salidas Y0 y Y1. Mire el estado de los LEDs para Y0 y Y1 cuando usted comienza un perfil de movimiento. Si el LED parpadea o si está constantemente encendido, la configuración es probablemente correcta.
2. **Error de programa** – Si no hay pulsos en Y0 ni Y1 usted puede tener un error en el programa. Verifique el contenido de V3636 y vea si hay un código del error que se puede engendrar cuando el PLC trata de ejecutar el perfil. Las descripciones del código del error se dan arriba.
3. **Verifique el valor del “blanco”** – El perfil no tendrá generación de pulsos si el valor de conteo es igual al valor del blanco (Ejemplo: conteo = 0, blanco = 0)

4. **Cableado**– Verifique que el cableado al motor del paso a paso está correcto. Recuerde que se requiere la conexión de la señal de tierra del PLC al sistema del movimiento.
5. **El sistema de movimiento** – Verifique que el amplificador del accionamiento está energizado y está habilitado. Para verificar el sistema del movimiento está funcionando, puede usarse la operación del modo 60 (las entradas / salidas normales del PLC) como mostrado en el programa de prueba abajo. Con él puede controlar manualmente Y0 y Y1 con X0 y X1, respectivamente. Usar un simulador de entrada es ideal para este tipo de depuración de un programa. Con los interruptores usted puede avanzar o retroceder uno paso cada vez el motor en cualquier dirección. Si el motor no se mueve con este control simple, la operación del Modo 30 no será posible hasta que sea corregido el problema con el sistema motor del accionamiento o el cableado.



6. **Error de memoria** – Los parámetros de configuración del sistema de entradas y salidas de alta velocidad se almacenan en la memoria del sistema de la CPU. Los datos corruptos en esta área de memoria pueden intervenir a veces con la operación apropiada del sistema de entradas y salidas de alta velocidad. Si todas las otras acciones correctivas fallan, inicializando la memoria de scratchpad puede resolver el problema. Con *DirectSOFT*, escoja el menú **PLC**, luego **SETUP**, luego **"Initialize Scratchpad"**.

Síntoma: El motor gira en la dirección equivocada

Causas posibles:

1. **Cableado**– Si usted ha escogido la operación del tipo de CW/CCW solamente intercambie los alambres en las salidas Y0 y Y1.
2. **Control de dirección** – Si usted ha escogido operación de tipo de Pulso y Dirección, cambie solo el bit de dirección al estado contrario.

Modo 40: Interrupciones de alta velocidad

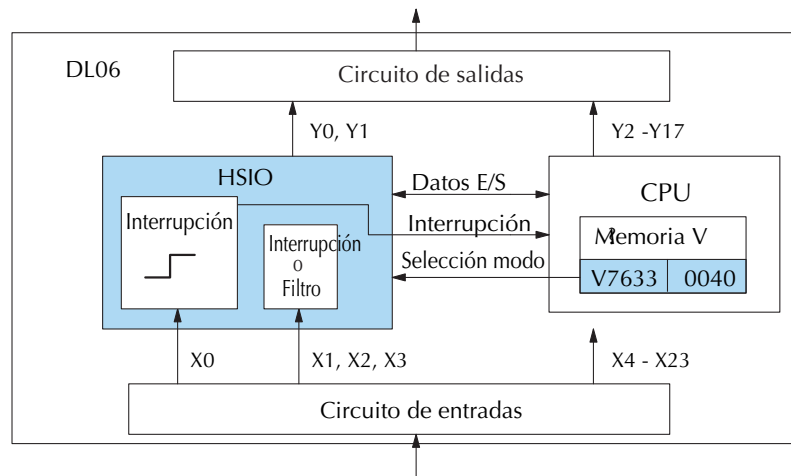
Propósito del modo 40

El Modo 40 del sistema HSIO permite una interrupción de alta velocidad al programa ladder. Esta capacidad es adecuada para elección de los siguientes escenarios:

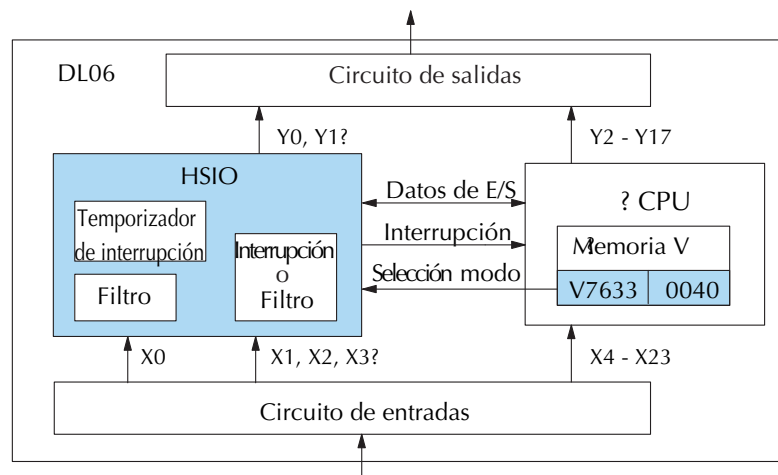
- Un evento externo debe causar una subrutina de interrupción en la CPU. Es típico usar instrucciones inmediatas de entradas y salidas en la subrutina.
- Debe ocurrir una rutina de interrupción en un tiempo definido, diferente del tiempo de barrido de la CPU (o más rápido o más lento). La interrupción por tiempo es programable de 5 a 999 ms.

Esquema de bloques funcional

El circuito de entradas y salidas de alta velocidad crea la interrupción de alta velocidad a la CPU. El esquema siguiente muestra la opción externa de interrupción, que usa X0. En esta configuración X1, X2 y X3 son interrupciones externas o entradas filtradas normales.

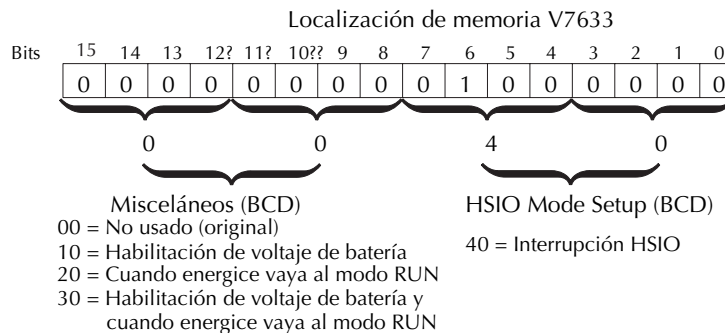


Alternativamente, usted puede configurar el circuito HSIO para generar interrupciones basadas en un temporizador, como mostrado abajo. En esta configuración, la entrada X0 es una entrada filtrada.



Configuración del Modo 40

Recuerde que V7633 es la memoria de selección del modo HSIO. Vea el esquema de abajo. Use 40 en BCD en el byte más bajo de V7633 para escoger el Modo de interrupción de alta velocidad.



Escoja el método más conveniente de programar V7633 de las siguientes opciones:

- Incluya las instrucciones LD y OUT en su programa ladder
- Use el redactor de memoria de *DirectSOFT*
- Use el programador portátil D2-HPP

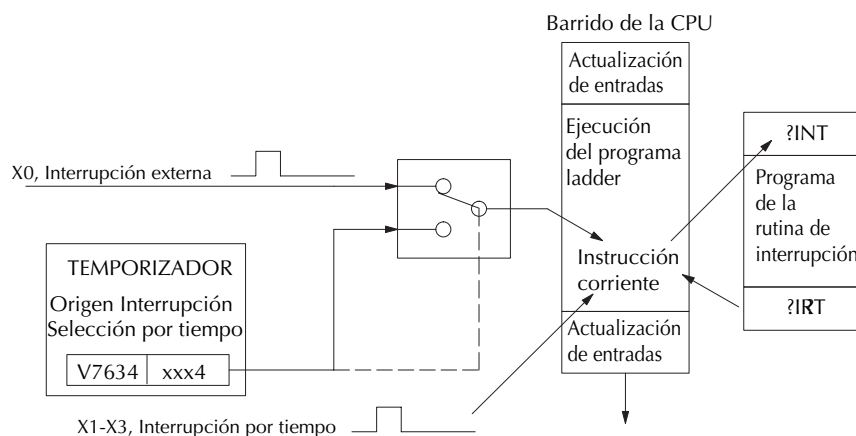
Le recomendamos usar el primer método para que la configuración de entradas y salidas de alta velocidad sea una parte integrante de su programa. Se muestra un programa de ejemplo en esta sección de cómo hacer ésto.

Las interrupciones y el programa ladder

Vea el diagrama abajo. La fuente de la interrupción puede ser externa (X0 - X3). Se puede usar un temporizador en vez de X0 como la fuente de interrupción. El parámetro de configuración en V7634 sirve para un doble propósito:

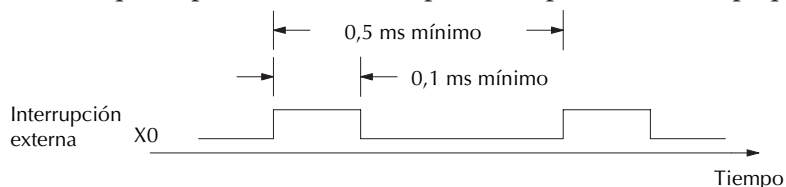
- Selecciona entre las dos fuentes de interrupción (el pulso externo o el interno). La interrupción del temporizador sólo puede ser usada con X0.
- En el caso de interrupción por tiempo, programe la base de tiempo de interrupción entre 5 y 999 ms.

La interrupción resultante usa la etiqueta (label) INT 0, 1, 2 o 3 en el programa ladder. Asegúrese de incluir la instrucción de Habilitar Interrupción (ENI) al comienzo de su programa. De otro modo, la rutina de interrupción no se ejecutará.



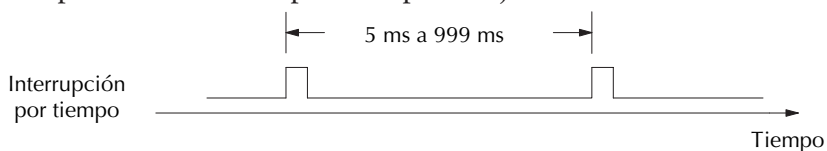
Parámetros de sincronización de interrupción externa

Las señales externas de interrupción deben reunir ciertos criterios de sincronización para garantizar que resulte una interrupción. Vea el esquema de sincronización abajo. El ancho mínimo de pulso es 0,1 milisegundo. Debe haber alguna demora antes del próximo pulso de interrupción, de modo que el período de interrupción no pueda ser más pequeño que 0,5 ms.



Parámetros de interrupción por tiempo

Cuando se selecciona la interrupción por tiempo, el sistema de entradas y salidas de alta velocidad genera la interrupción a la lógica ladder. No hay "ancho de pulso" de interrupción en este caso, pero el período de interrupción se puede ajustar de 5 a 999 ms.



Configuración INT temporizada / entrada X

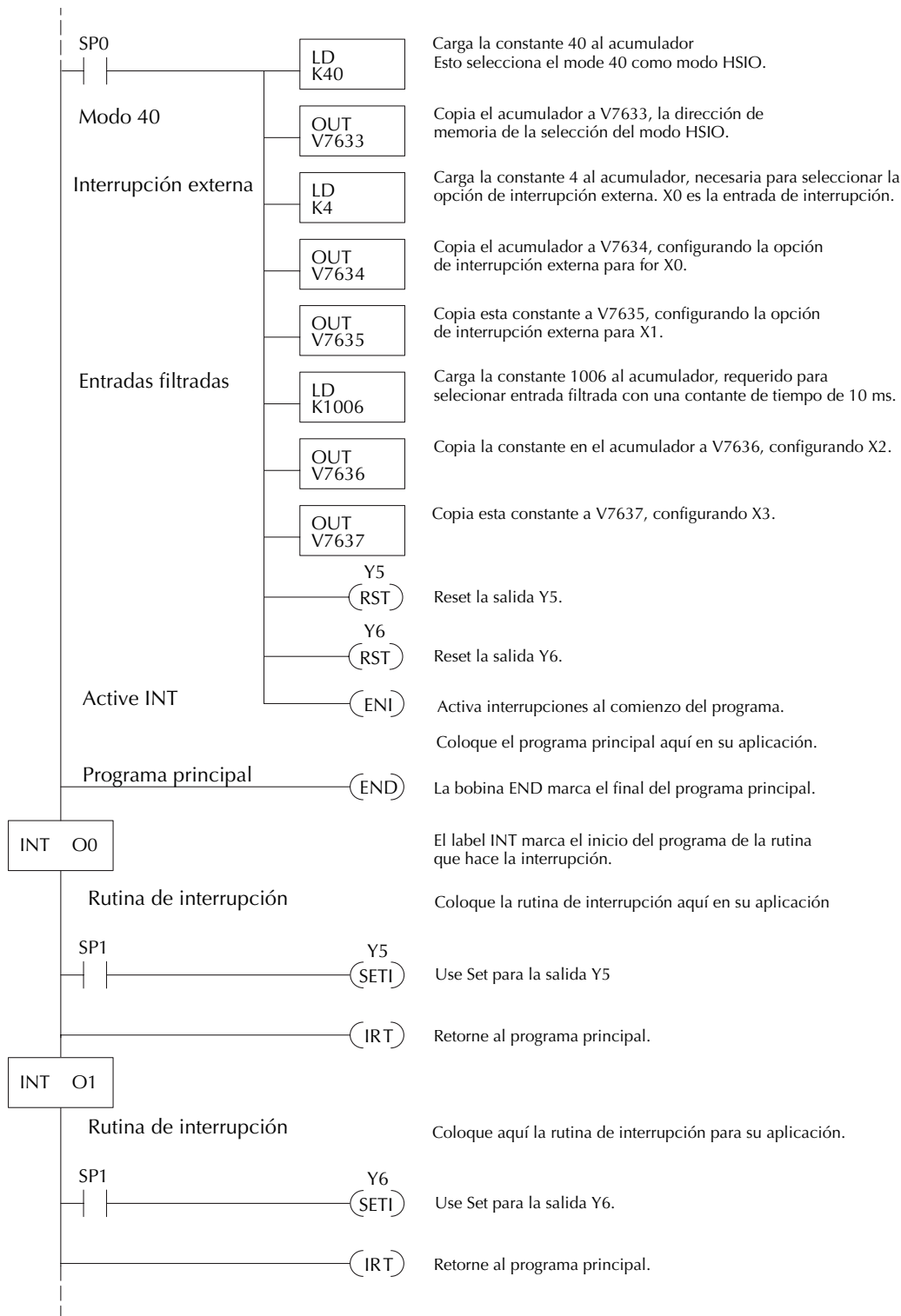
Las opciones configurables de entradas discretas para el modo de alta velocidad de interrupción se listan en la tabla de abajo. La entrada X0 es la interrupción externa cuando "0004" está en V7634. Si usted en cambio necesita una interrupción por tiempo, entonces V7634 contiene el período de tiempo de interrupción, y la entrada X0 llega a ser una entrada filtrada (usa la constante de tiempo de filtro X1 originalmente). Las entradas X0, X1, X2, y X3, pueden filtrar las entradas, teniendo registros individuales de configuración y constantes de tiempo de filtro, entradas de interrupción o entradas de contador.

Entrada	Memoria configuración	Función	Código hexadecimal
X0	V7634	Interrupción externa	0004 (por defecto)
		Interrupción x tiempo	xxx4, xxx=INT base 5 a 999 ms(BCD)
X1	V7635	Interrupción	0004 (por defecto)
		Entrada de pulsos	0005
		Entrada filtrada	xx06(xx=cte. de tiempo) 0-99 ms(BCD)
X2	V7636	Interrupción	0004 (por defecto)
		Entrada de pulsos	0005
		Entrada filtrada	xx06(xx=cte. de tiempo) 0-99 ms(BCD)
X3	V7637	Interrupción	0004 (por defecto)
		Entrada de pulsos	0005
		Entrada filtrada	xx06(xx=cte. de tiempo) 0-99 ms(BCD)

Si usted está usando solamente *uno* de los puntos para una interrupción, puede escoger un modo principal diferente (por ejemplo 10, 20, 30, 50, o 60); y entonces, configura solo uno de las terminales no tomado como interrupción. Por ejemplo, quizás quiera configurar su CPU como el modo contador (el Modo 10) y usar la señal X3 para una interrupción de alta velocidad. Usted debe leer las secciones individuales para otro modo alternativo que usted quiera escoger. Allí usted encontrará las instrucciones de cómo escoger una interrupción de alta velocidad como una función secundaria.

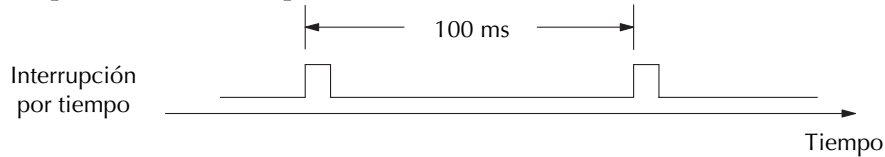
Ejemplo 1 modo 40: Interrupción externa

El programa siguiente selecciona el Modo 40 y luego escoge la opción externa de interrupción para las entradas X0 y X1. Las entradas X2 y X3 se configuran como filtradas con una constante de tiempo de 10 ms. El programa es genérico, y puede ser adaptado a su aplicación.

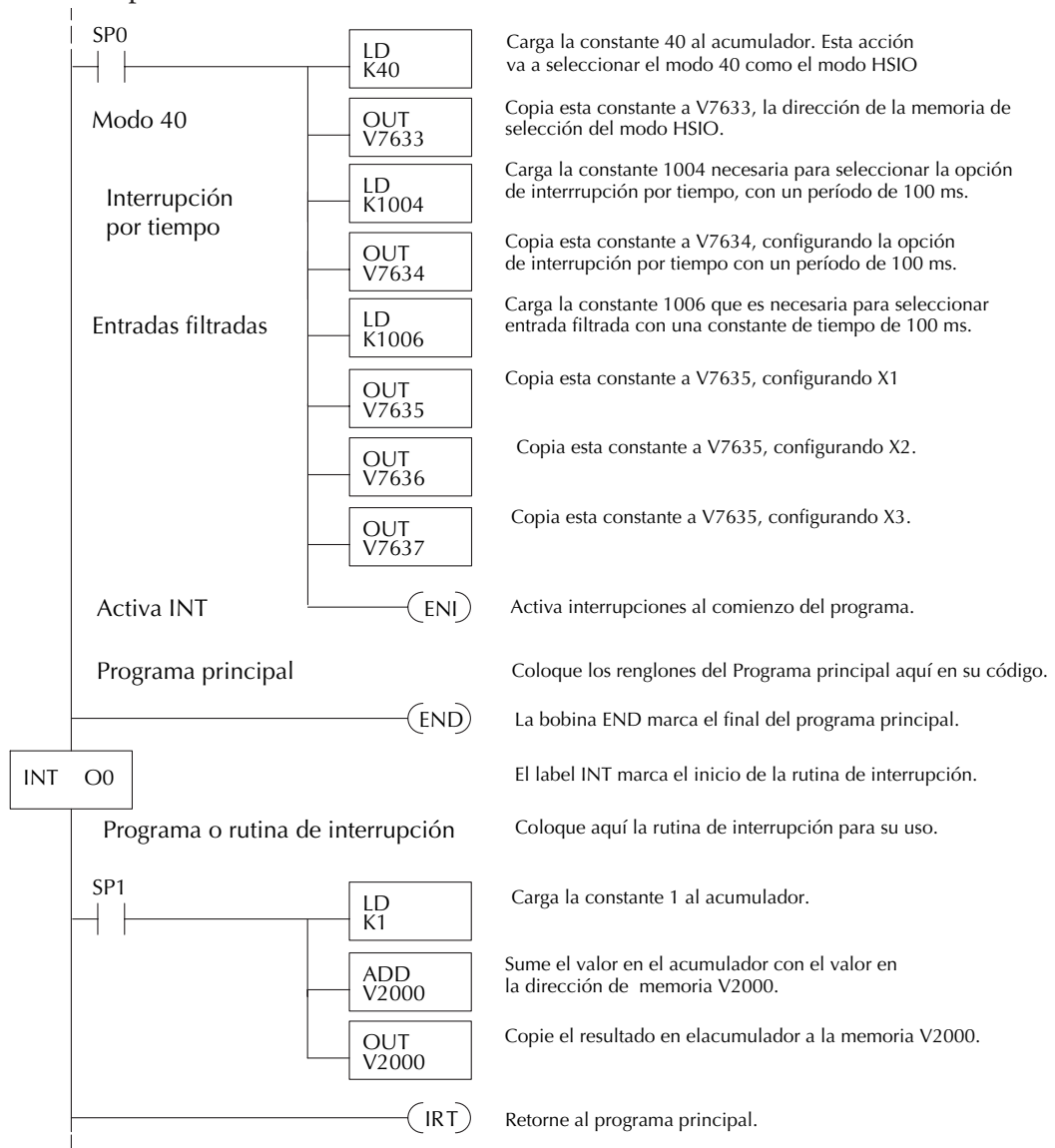


Ejemplo 2 modo 40: Interrupción por tiempo

El programa siguiente selecciona el modo 40, luego escoge la opción de interrupción por tiempo, con un período de interrupción de 100 ms.



Las entradas X1, X2, y X3, se configuran como entradas filtradas con una constante de tiempo de 10 ms. Note que X0 usa la constante de tiempo de X1. El programa es genérico y puede ser adaptado a su aplicación.



NOTA: X0 no debe ser usado en la lógica del programa principal. Sin embargo, usando X0 para hacer ON el Bit C10, por ejemplo, permitirá el uso de C10 en la lógica principal. No se olvide de apagar C10 cuando sea conveniente.

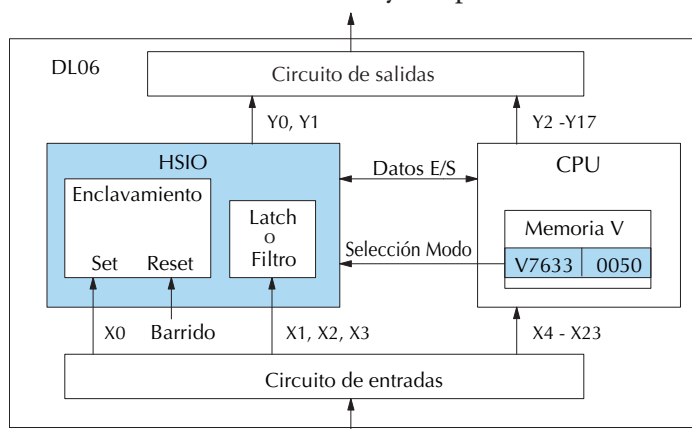
Modo 50: Entrada de captura de pulso

Propósito del modo 50

El circuito HSIO tiene un modo de operación de captura de pulso. Supervisa la señal en las entradas X0 - X3, detectando la ocurrencia de un pulso estrecho. El propósito del modo de captura de pulsos es habilitar el programa ladder a "ver" un pulso de entrada que es más corto que la duración del tiempo de barrido corriente. El circuito HSIO mantiene el pulso en las entradas X0 - X3 por un barrido. Este contacto se abre automáticamente después de un barrido.

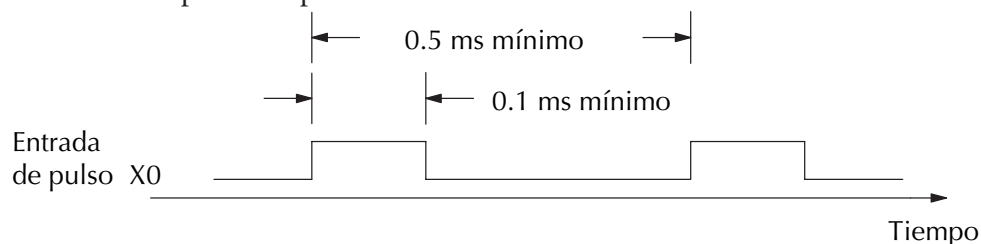
Esquema funcional de bloque

Vea el diagrama de bloque abajo. Cuando el byte más bajo de la memoria "Modo HSIO" V7633 contiene un número "50" BCD, se habilita el modo de captura de pulso en el circuito HSIO. X0 - X3 se tornan las entradas de captura de pulsos, que mantiene verdadero el pulso cada vez que se detecta una transición de falso para verdadero. El sistema HSIO mantiene el pulso solo por un barrido. Las entradas X1 X2, y X3 pueden ser filtradas también.



Parámetros de sincronización de captura de pulso

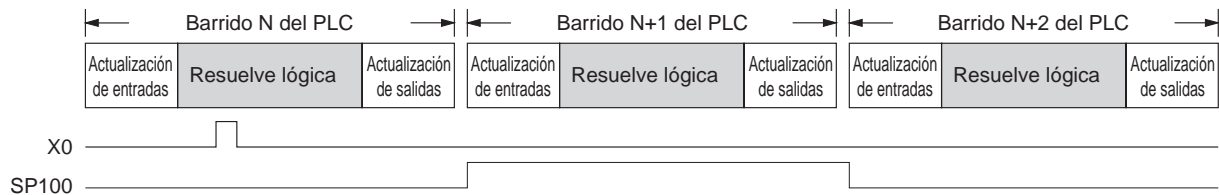
Las señales de pulsos en X0 - X3 deben reunir ciertos criterios de sincronización para garantizar que funcione una captura de pulso. Vea el diagrama de tiempo abajo. La característica de entrada en X0 es fija (no es una entrada filtrada programable). La anchura mínima de pulso es 0,1 ms. Debe haber alguna demora antes que llegue el próximo pulso de modo que el período de pulso no pueda ser más pequeño que 0,5 ms. Si el período de pulso es menor que 0,5 ms., el próximo pulso se considerará parte del pulso actual.



Nota: Las funciones de captura de pulso y de pulso filtrado son opuestas por naturaleza. La captura de pulso trata de capturar pulsos cortos mientras que la entrada filtrada trata de rechazar los pulsos cortos.

Cuando usar modo de captura de pulso

Use el modo de captura de pulso para aplicaciones donde la entrada (por ejemplo X0) no puede ser usado en el programa de usuario porque el ancho del pulso es muy corto. Use SP100 en vez de X0. El contacto SP100 permanece activado durante todo el tiempo de barrido justo después que el pulso es capturado, como mostrado en la figura adyacente.

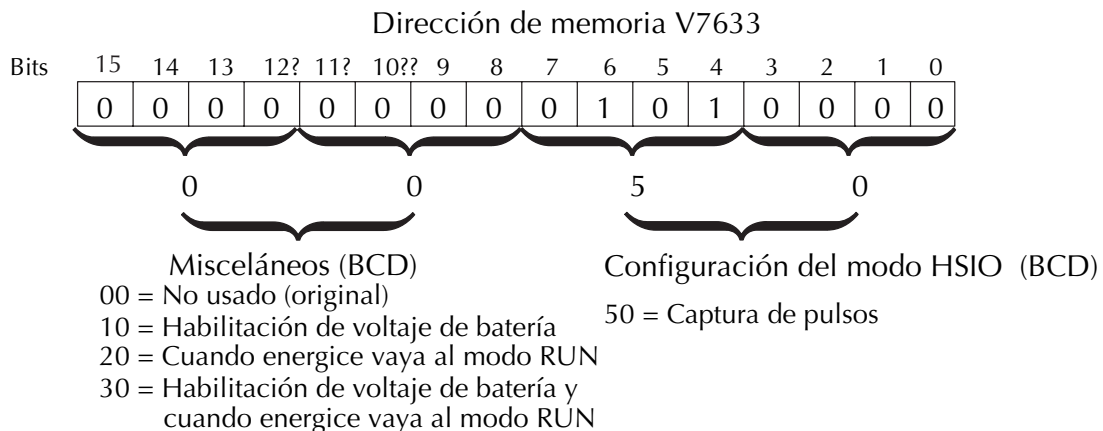


El relevador de estado para X0 es SP100. Los otros relevadores de estado se muestran en la tabla abajo.

Entrada	Relevador de estado
X0	SP100
X1	SP101
X2	SP102
X3	SP103

Configuración del modo 50

Recuérdese que la memoria V7633 es la de selección del modo remoto HSIO. Vea el diagrama de abajo. Use el número "50" BCD en el byte inferior para seleccionar este modo.



Escoja el método más conveniente de programar el valor V7633 entre:

- Incluya las instrucciones LD y OUT en su programa ladder
- Use el programa *DirectSOFT* (Menús Memory Editor o Data View)
- Use el programador portátil D2-HPP

Recomendamos usar el método de instrucciones en su programa para que siempre esté incluido en el programa. Se muestra más tarde en esta sección un programa de ejemplo.

Configuración de las entradas X

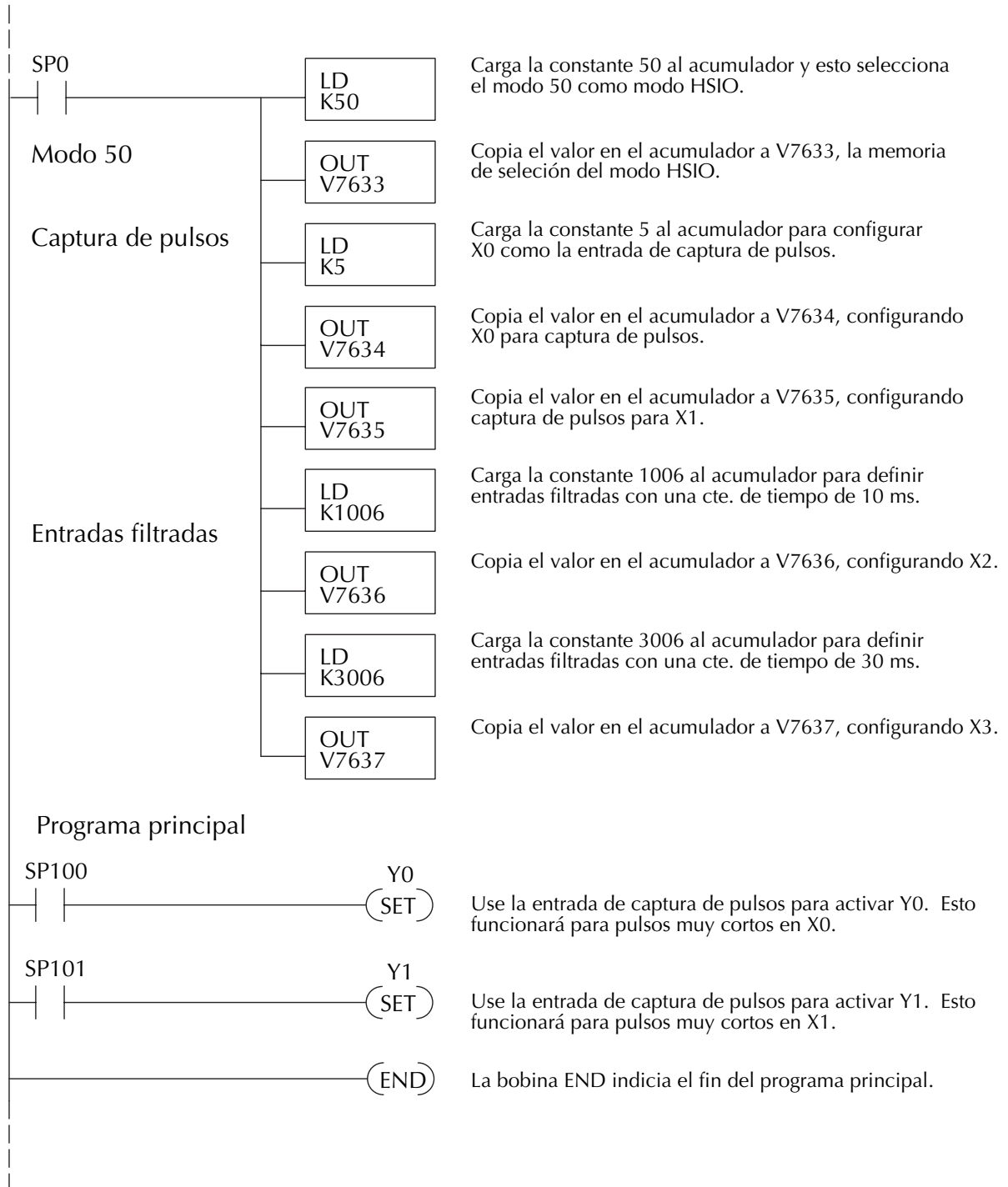
Las diversas opciones de configuración de la entrada para el modo de Captura de Pulso se listan en la tabla abajo. Cada entrada tiene su propio registro de configuración y constante de tiempo de filtro.

Entrada	Configuración	Función	Código hexadecimal
X0	V7634	Captura de pulso	0005 (Por defecto)
		Interrupción	0004
X1	V7635	Captura de pulso	0005 (Por defecto)
		Entrada filtrada	xx06 (xx = cte de tiempo) 0-99 ms(BCD)
X2	V7636	Interrupción	0004
		Captura de pulso	0005 (Por defecto)
		Entrada filtrada	xx06 (xx = cte de tiempo) 0-99 ms(BCD)
X3	V7637	Interrupción	0004
		Captura de pulso	0005 (Por defecto)
		Entrada filtrada	xx06 (xx = cte de tiempo) 0-99 ms(BCD)

Ejemplo 1 modo 50 : Captura de pulso

El programa siguiente selecciona el Modo 50, luego programa el código de captura de pulso para X0 y X1. Las entradas X2, y X3 se configuran como entradas filtradas con constantes de tiempo de 10 y 30 ms. respectivamente. El programa es genérico, y puede ser adaptado a su aplicación.

E



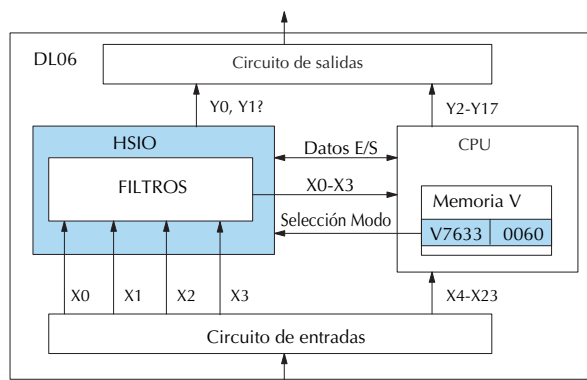
Modo 60: Entradas discretas con filtro

Propósito del modo 60

El último modo que discutiremos para el circuito de entradas y salidas de alta velocidad es el Modo 60, entradas con filtro. El propósito de este modo es permitir que el circuito de entrada rechace pulsos estrechos y acepte anchos, cuando visto por el programa ladder. Esto es útil en los ambientes especialmente ruidosos u otras aplicaciones donde el ancho de pulso es importante. En todos los otros modos en este capítulo, las funciones de las entradas X0 a X3 generalmente son especiales. Sólo las entradas que no tienen funciones especiales de alta velocidad operan como filtradas por defecto. Ahora en modo 60, las cuatro entradas X0 a X3 funcionan como entradas discretas filtradas.

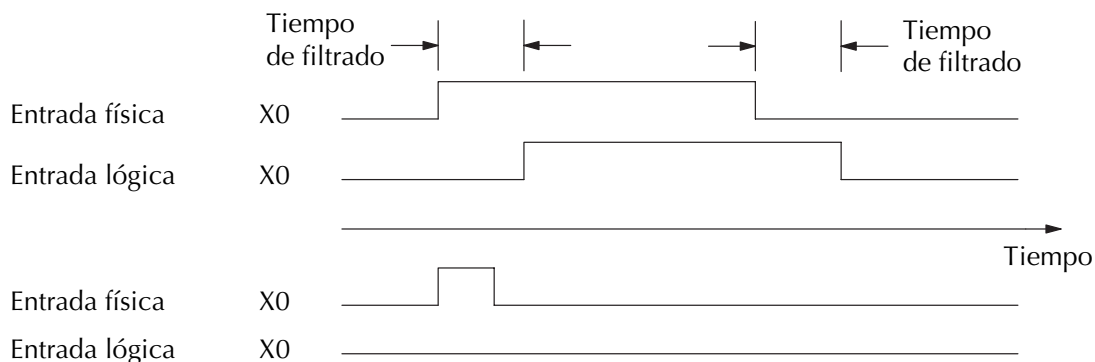
Diagrama de bloques funcional

Vea el diagrama de bloque abajo. Cuando el byte más bajo de la memoria V7633 que define el modo del sistema HSIO contiene un número "60" BCD, se habilita el filtro de la entrada en el circuito HSIO. Cada entrada X0 hasta X3 tiene su propia constante de tiempo de filtro. El circuito de filtro asigna las salidas de los filtros como las referencias lógicas X0 a X3.



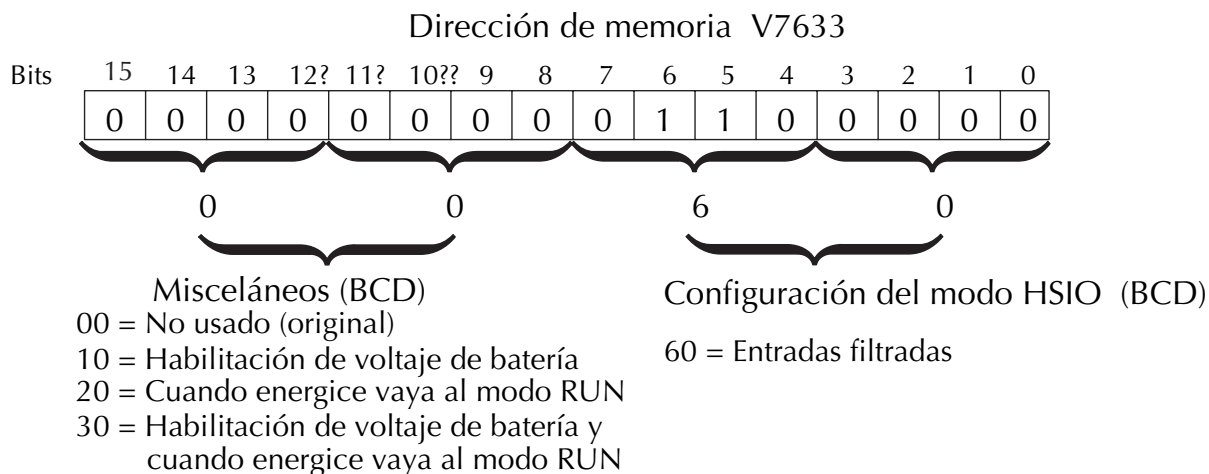
Parámetros de constante de tiempo del filtro de entrada

Los pulsos de señal en las entradas X0 - X3 son filtrados usando un tiempo de demora. En la figura abajo, el pulso de entrada en la primera línea es más largo que el tiempo del filtro. La entrada lógica resultante al programa ladder es atrasada (demorada) por el filtro de tiempo en la transición de falso para verdadero y viceversa. En las formas de ondas de mas abajo, el ancho físico del pulso de entrada es más pequeño que el tiempo del filtro. En este caso, la entrada lógica permanece en estado falso para el programa ladder (el pulso de entrada se filtró y no entró).



Configuración del Modo 60

Recuérdese que la memoria V7633 es la de selección del modo remoto HSIO. Vea el diagrama de abajo. Use el número "60" BCD en el byte inferior para seleccionar este modo.



Escoja el método más conveniente de programar el valor V7633 entre:

- Incluya las instrucciones LD y OUT en su programa ladder
- Use el programa *DirectSOFT* (Menús Memory Editor o Data View)
- Use el programador portátil D2-HPP

Recomendamos usar el método de instrucciones en su programa para que siempre esté incluido en el programa. Un programa de ejemplo se muestra más tarde en esta sección.

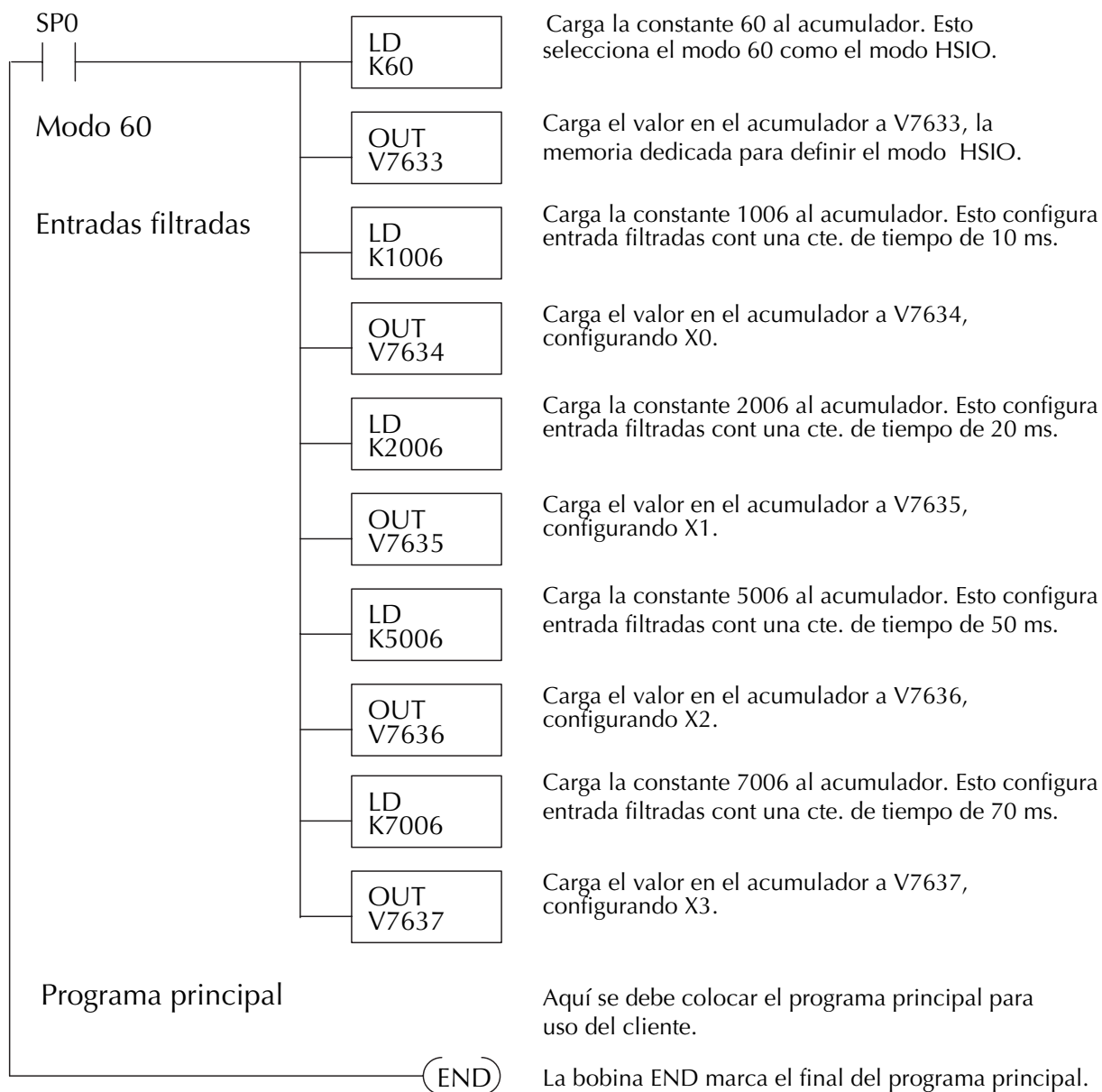
Configuración de las entradas X

Las diversas opciones de configuración de las entradas para el modo de entradas filtradas se listan en la tabla abajo. La constante de tiempo del filtro es programable de 0 a 99 ms. (la entrada actúa como una entrada discreta normal cuando la constante de tiempo es puesta a 0). El código para esta selección ocupa el byte superior de la memoria de configuración en BCD. Combinamos este número con el requerido "06" en el byte más bajo para obtener "xx06", donde xx = 0 a 99. Las entradas X0, X1, X2, y X3 pueden ser entradas filtradas. Cada entrada tiene su propio registro de configuración y constante de tiempo de filtro.

Entrada	Memoria de configuración	Función	Código hexadecimal requerido
X0	V7634	Entrada filtrada	xx06 (xx=cte. de tiempo) 0-99 ms (BCD)(por defecto)
X1	V7635	Entrada filtrada	xx06 (xx=cte. de tiempo) 0-99 ms (BCD)(por defecto)
X2	V7636	Entrada filtrada	xx06 (xx=cte. de tiempo) 0-99 ms (BCD)(por defecto)
X3	V7637	Entrada filtrada	xx06 (xx=cte. de tiempo) 0-99 ms (BCD)(por defecto)

Ejemplo modo 60 : Entradas filtradas

El programa siguiente escoge el Modo 60, luego programa las constantes de tiempo de demora de filtro para las entradas X0, para X1, para X2, y para X3. Cada constante de tiempo del filtro es diferente, para propósitos de ilustración. El programa es de otro modo genérico, y puede ser adaptado a su aplicación.



MEMORIAS DEL PLC



En este apéndice

Memorias del PLC DL06	F-2
-----------------------------	-----

Memorias del PLC DL06

Al diseñar un uso del PLC, es importante que el usuario del PLC entienda los diversos tipos de memoria en los tipos del PLC. Se utilizan dos tipos de memoria en el PLC DL06: CPU, RAM y EEPROM. Esta memoria se puede configurar por el usuario del PLC como memoria retentiva o no-retentiva.

La memoria retentiva es la memoria que es configurada por el usuario para mantener valores. La memoria no retentiva es la memoria que es configurada por el usuario del PLC para limpiar los datos cuando se apaga y se enciende el PLC o en una transición del modo PROGRAM para RUN. Los rangos retentivos se pueden configurar con el programador D2-HPP usando AUX57 o *DirectSOFT (PLC> Setup)*.

El contenido de la memoria RAM puede ser escrito a y leído un número infinito de veces, pero la memoria RAM requiere una fuente de energía para mantener el contenido de la memoria. El contenido de la memoria RAM es mantenido por la fuente de alimentación interna (5VCC) cuando el PLC es alimentado por una fuente externa, normalmente 120VCA. Cuando la energía al PLC se apaga, el contenido de la memoria RAM es mantenido por un "Super-Condensador "de respaldo. Si el condensador se descarga, el contenido de la memoria RAM se pierde. El tiempo de retención de datos de la RAM por el condensador de respaldo es de máximo 3 semanas, y mínimo de 4 días y medio (a 60°C).

El contenido de la memoria EEPROM se puede leer un número infinito de veces, pero hay un límite al número de veces que puede ser escrito (la especificación típica es 100.000 escrituras). La EEPROM no requiere una fuente de energía para mantener el contenido de la memoria y se conservará el contenido de la memoria indefinidamente.

La memoria V de usuario del PLC se almacena en la memoria volátil RAM y la memoria permanente EEPROM. Los datos que son almacenados en la memoria RAM utilizan V400-V677, V1200-V7377 y V10000-V17777. Los datos almacenados en EEPROM utilizan V7400-V7577 y V700-V777, V7600-V7777 y V36000-V37777.

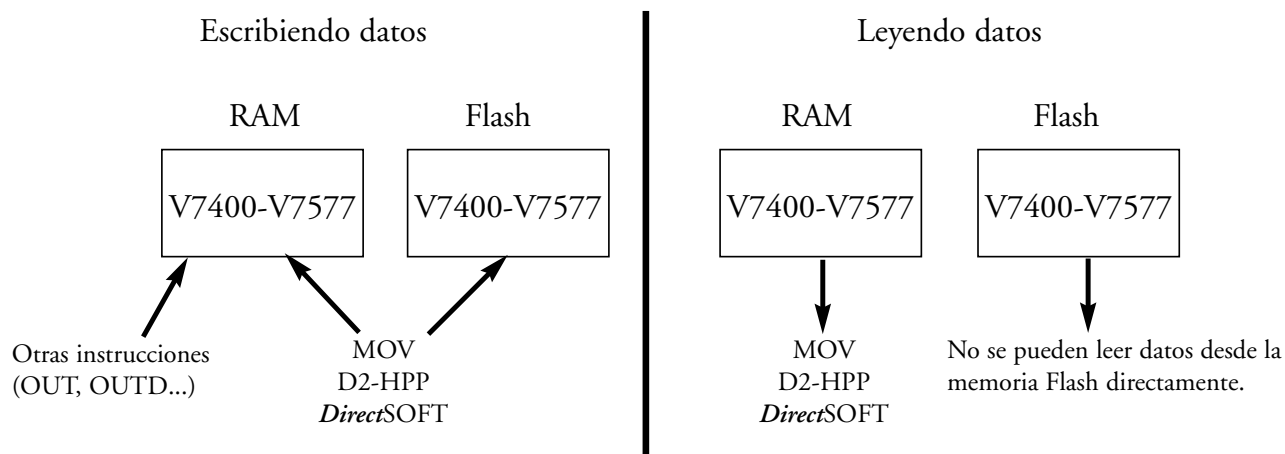
Los valores de datos que se deben conservar por largos períodos de tiempo, cuando el PLC se desconecta de la alimentación de energía, se deben almacenar en la memoria V del tipo EEPROM.

Los valores de datos que están cambiando continuamente o que se pueden inicializar con "lógica de programa " se deben almacenar en la memoria V del tipo RAM.

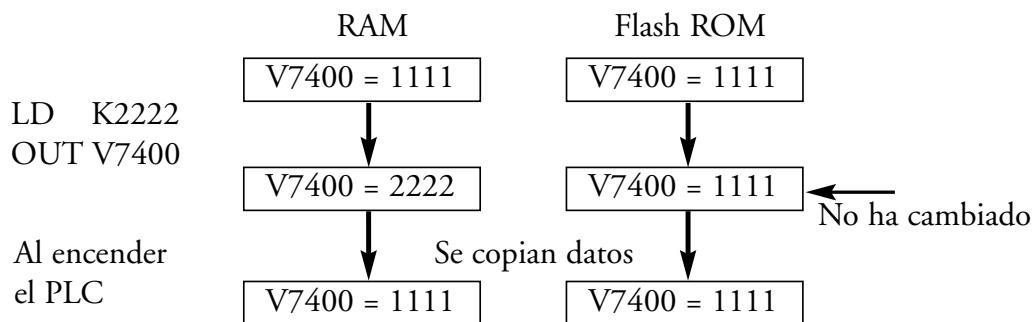
Memoria V permanente en el DL06

Hay 2 tipos de memoria asignados para el área permanente de memoria V. Estos tipos son on RAM y FLASH ROM(EEPROM). Ellos comparten las mismas direcciones de memoriaV ; sin embargo, solamente se puede usar la instrucción de MOV, el D2-hpp y *DirectSOFT* para escribir datos a la memoria Flash. Cuando usted escribe datos a la memoria Flash, los mismos datos también se escriben a la memorias RAM. Si usted usa otras instrucciones, se puede escribir solamente datos a la memoria RAM. Cuando le los datos leídos del área permanente de memoria V, los datos seleen siempre desde la memoria RAM.

Después de apagar y encender el PLC, el PLC copia siempre los datos contenidos en la memoria Flash a la memoria RAM.



Si usted utiliza instrucciones (excepto la instrucción MOV) para escribir datos en el área permanente de memoria V, usted realmente solamente actualiza los datos en RAM. Después de apagar y encender el PLC, el PLC copia los datos de la memoriaV en Flash a la memoria RAM, de modo que usted puede pensar que los datos que usted cambió han desaparecido. Para evitar este problema, recomendamos que usted use la instrucción MOV.



Parecería ser que los datos que estaban en la memoria están retornando.

TABLA ASCII



En este apéndice

Tabla de conversión ASCIIG-2

Apéndice G: Tabla de caracteres ASCII

Tabla de conversión de DECIMAL a HEXADECIMAL y a ASCII

DEC	HEX	ASCII	DEC	HEX	ASCII	DEC	HEX	ASCII	DEC	HEX	ASCII
0	00	NUL	32	20	espacio	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	“	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL	39	27	'	71	47	G	103	67	g
8	08	BS	40	28	(72	48	H	104	68	h
9	09	TAB	41	29)	73	49	I	105	69	i
10	0A	LF	42	2A	*	74	4A	J	106	6A	j
11	0B	VT	43	2B	+	75	4B	K	107	6B	k
12	0C	FF	44	2C	,	76	4C	L	108	6C	l
13	0D	CR	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

PESOS DE LOS COMPONENTES



En este apéndice

Tablas de pesos de los productosH-2

Tabla de pesos de los productos

PLC	Peso
D0-06AR	1,78 lb./807g.
D0-06DR	1,76 lb./798 g.
D0-06DR-D	1,72 lb./780 g.
D0-06AA	1,78 lb./807 g.
D0-06DA	1,76 lb./798 g.
D0-06DD1	1,68 lb./762 g.
D0-06DD1-D	1,64 lb./743 g.
D0-06DD2-D	1,64 lb./743 g.
D0-06DD2	1,68 lb./798 g.
D0-06LCD	0,12 lb./54.4g.

H

SISTEMAS NUMÉRICOS



En este apéndice...

Introducción a sistemas numéricosI-2
Sistema numérico decimalI-2
Sistema numérico octalI-2
Memorias para datos y para configuraciones del PLCI-3
Sistema numérico binarioI-3
Sistema numérico BCDI-4
Sistema numérico hexadecimalI-5
Sistema numérico real de punto flotanteI-6
Sistema numérico en representación GrayI-7
Representación del complemento de 2I-9
El cálculo del complemento de 2I-10

SISTEMAS NUMÉRICOS

Introducción a sistemas numéricos

En este apéndice describiremos algunos de los sistemas numéricos que son usados en los PLCs *DirectLogic*. Como cualquier PLC, éstos almacenan y manipulan números en forma binaria, esto es, sólo usan 1s y 0s, y entonces los unos y ceros tienen que tener una cierta convención para representar un número. Es por eso que debemos describir qué sistemas numéricos usan estos PLCs.

Describiremos el sistema decimal, el binario, el sistema octal, el sistema BCD, el sistema hexadecimal, los números reales con formato de punto flotante y el complemento de 2 y sus propiedades y como dato curioso el código Gray, que es usado con ciertos encoders.

Los PLCs ofrecen una cantidad fija de recursos dependiendo del modelo y de la configuración. Usamos la palabra "recursos" para incluir memoria variable, puntos de entradas y salidas, temporizadores, contadores, etc.

Los PLCs *DirectLogic* usan grupos de entradas y salidas en grupos de 8, es decir, todos los recursos de los PLCs son contados en el sistema octal. Es fácil para computar que los computadores cuenten en grupos de 8 porque 8 es una potencia de 2.

Sistema decimal

El sistema numérico decimal es el que se usa corrientemente y significa simplemente que se cuenta en grupos de 10 a un tiempo esto es 0 a 9, 10 a 19, 20 a 29, etc.

Este sistema es generado como una serie de números cuya base es el número 10. Esto viene de la numeración arábiga que usaba el cero, como mejoría de la numeración romana.

Para formar el número 12345, lo que realmente se hace es:

$$1 \times 10^4 + 2 \times 10^3 + 3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0 = 12345$$

Llamamos **peso** al número que multiplica cada dígito por la potencia de 10. Esto es, el peso del dígito más a la derecha (el menos significativo) es 1, el peso del segundo dígito hacia la izquierda es 10; el tercer dígito es 100, y así sucesivamente. Este concepto se usará posteriormente.

Esta explicación es necesaria para entender como funcionan los otros sistemas numéricos.

Sistema numérico octal

El sistema numérico OCTAL significa simplemente que se cuenta en grupos de 8 a un tiempo esto es 0 a 7, 10 a 17, 20 a 27, etc. usando el 8 como base.

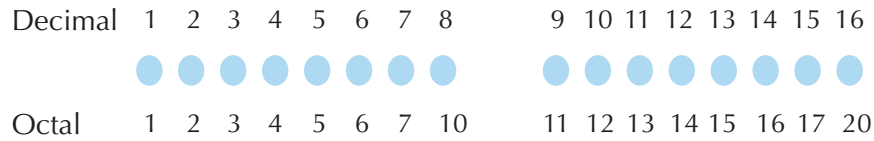
Esto es, un número octal estará formado como sigue:

$1 \times 8^4 + 2 \times 8^3 + 3 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$ que en la numeración decimal es

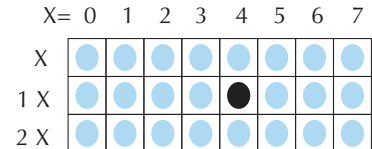
$4096 + 1024 + 192 + 32 + 5$ que equivale a 5357

Note que el peso es basado en una potencia de 8.

Decimal	1	2	3	4	5	6	7	8
	●	●	●	●	●	●	●	●
Octal	1	2	3	4	5	6	7	10



Vea la figura adyacente, donde tenemos dos grupos de 8 círculos. Contando en octal los círculos tenemos "2 0" círculos significando dos grupos de 8 y 0 grupos individuales.



Ahora los recursos del PLC se cuentan de 0 hasta siete que resulta también en un grupo de 8, el número 0 significa algo a un computador de modo que no lo saltamos; contamos de 0 a 7 y ahí entenderá que si éstos fueran contadores CT14 correspondería a lo que es la localización del círculo negro.

Memorias de datos y memorias para configuración del PLC

Las direcciones de memoria usan el mismo sistema octal, por ejemplo V2073 es una localización o dirección válida mientras que V1983 no es válida (9 y 8 no son dígitos válidos octales)



Cada localización de memoria tiene una palabra de datos y cada palabra contiene 16 bits.

Cada palabra tiene 2 bytes, es decir 2 grupos de 8 bits cada uno.

Cada byte tiene 2 nibbles, es decir, 2 grupos de 4 bits cada uno.

Los bits se muestran diagramáticamente en el diagrama abajo y el bit menos significativo (LSB) estará a la derecha, y el bit más significativo (MSB) a la izquierda. Usamos la palabra más significativa refiriéndose al peso de cada bit.



El peso de cada tipo es una característica que permite determinar el valor equivalente decimal, siendo el sistema decimal el sistema numérico que usamos los humanos.

Los datos de memorias de 16 bits son almacenados en forma binaria, pero nosotros raramente programamos las memorias de datos colocando cada bit. En vez de eso, usamos instrucciones o usamos herramientas de visualización que nos permiten trabajar con números binarios, números decimales, números BCD, números hexadecimales o incluso números reales de punto flotante. Todos estos números son convertidos y almacenados como binarios, es decir un conjunto de ceros y unos.

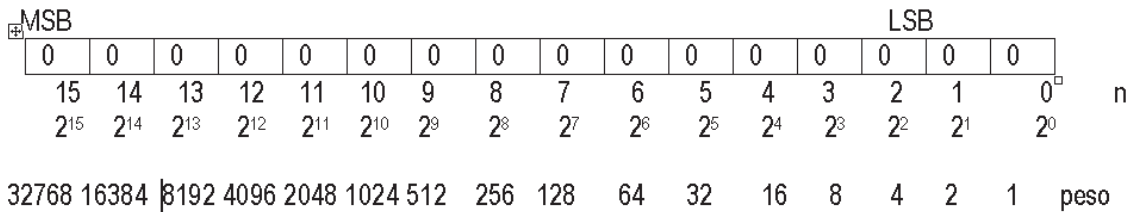
Una pregunta normalmente es, como se sabe si son números binarios, octales, decimales o hexadecimales? La respuesta es que no podemos decir cual numero es sólo mirando al conjunto de ceros y unos, pero realmente no es importante. Lo importante es que la fuente o el

mecanismo que escribe los datos en una localización o una dirección de memoria y el objeto que luego lee ellos deben tener el mismo tipo de datos o usar la misma convención. La localización de memoria es solamente una caja de almacenamiento, no convierte o mueve datos por sí mismo.

Sistema numérico binario

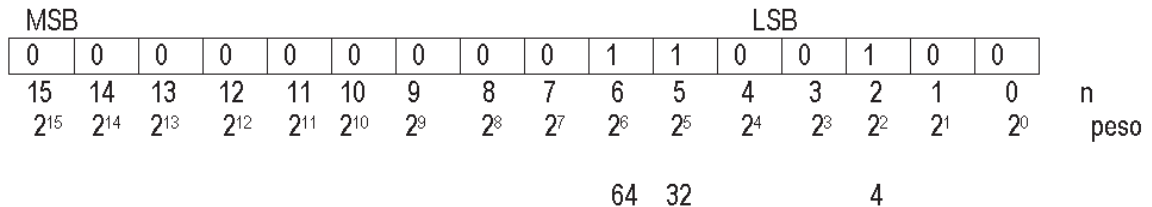
Un número en base 2 puede expresar cualquier número equivalente en decimal. Describimos como número binario a un número que es compuesto de 16 bits, cada uno teniendo un peso de 2 elevado a n siendo n la posición relativa de los bits; vea en la figura de abajo más explicaciones.

Un número binario es formado por la suma de los valores que contienen un 1 en el bit correspondiente.



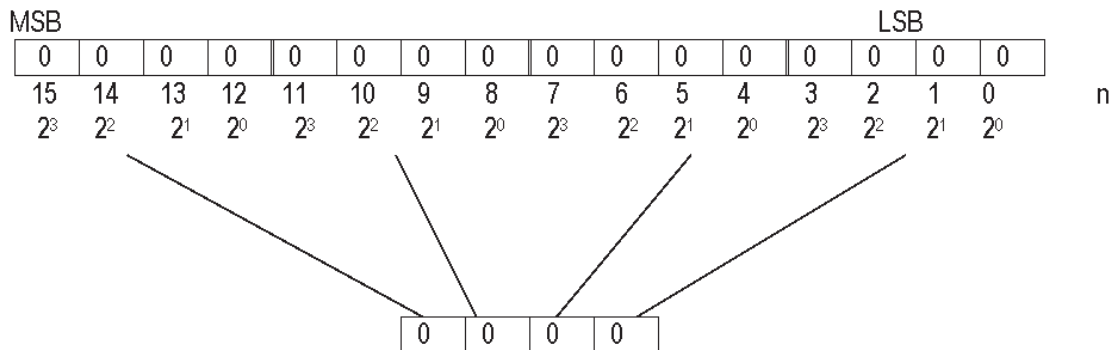
Por ejemplo, el número 100 decimal corresponde a un número en bits con un 1 en los bits 2, 5 y 6, como muestra el diagrama abajo. Aquí se suma $64+32+4=100$. Los bits que son 0 no se suman.

Un número binario también puede contener 15 bits, 12 bits, o un valor dado por convención.

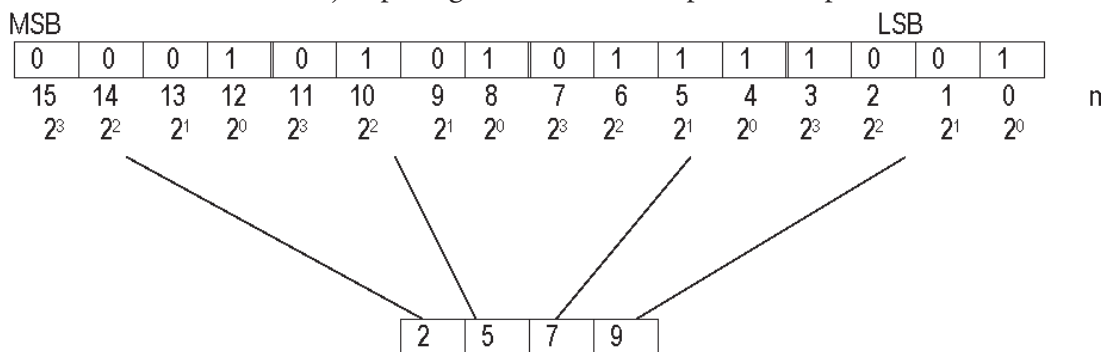


Sistema numérico BCD (Binary coded decimal)

Ya que los humanos naturalmente cuentan en el sistema numérico decimal, preferimos entrar y ver datos en el PLC en valores decimales. Sin embargo los computadores son más eficientes al usar números binarios puros. Una solución de compromiso entre los dos es la representación BCD. Un dígito BCD tiene el rango de 0 a 9 y es almacenado como cuatro bits (llamado un nibble). Esto permite que cada localización de memoria almacenen cuatro dígitos BCD, con un rango de números decimales de 0000 hasta 9999.



En un sentido binario puro, una palabra 16 bits representa un número de 0 a 65,536. Al almacenar números BCD, el rango es reducido a 0 hasta 9999. Muchas instrucciones aritméticas en el PLC usan datos BCD en que el rango es reducido a 0 a 9999. Muchas instrucciones aritméticas usan datos BCD e incluso *DirectSOFT* nos permite entrar y ver datos en el sistema BCD. *DirectSOFT* tiene instrucciones que nos permiten convertir de BCD a binario o viceversa. Vea en el ejemplo siguiente como es representado el número 2579/



Sistema numérico hexadecimal

Los números hexadecimales son similares a los números BCD, excepto que ellos utilizan todos los números binarios en cada nibble. Estos son números en base 16 de modo que se necesitan 16 dígitos diferentes. Para extender los números decimales de 0 a 9 se usan las letras A hasta F, como se muestra abajo

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

Un número hexadecimal de 4 nibbles puede representar hasta 65536 valores en una palabra de 16 bits. El rango va desde 0000_h hasta FFFF_h, siendo el sufijo _h la indicación de que el número está representado como hexadecimal. A veces los PLCs necesitan todo este rango, por ejemplo, para datos de sensores con señales análogas. El sistema hexadecimal es sólo una forma conveniente de ver datos almacenados en forma binaria.

Sistema numérico real de punto flotante

El estándar de IEEE 754 de números de punto flotante es la representación más común hoy para números reales en computadoras, inclusive PCs basados en Intel, Macintoshs, y la mayoría de las plataformas Unix.

Hay varias maneras de representar los números reales en computadoras. El sistema de coma fija (o punto como es usado en USA) coloca una coma en algún lugar entre los dígitos, por convención y es equivalente a usar enteros que representan las porciones de alguna unidad. Por ejemplo, uno quizás represente un centésimo de una unidad; si usted tiene cuatro dígitos de decimal, usted podría, por ejemplo, representar 10,82, o 00,01.

Otro enfoque sería usar números racionales, y representar cada número como la razón de dos enteros.

La representación del punto flotante representa básicamente un número real en notación científica. La notación científica representa los números como un número base y un exponente.

Por ejemplo, 123,456 podría ser representado como $1,23456 \times 10^2$. En hexadecimal, el número 123, ABC podría ser representado como $1,23ABC \times 16^2$.

¿Cual es la convención con IEEE para punto flotante de 32 bits?

Los números de punto flotante tienen tres componentes básicos: el signo, el exponente, y la mantisa. La mantisa se compone de una fracción y un dígito delantero implícito (explicado abajo). La base (2) del exponente es implícita y no hay necesidad de almacenarla. La tabla siguiente muestra la disposición los valores de punto flotantes de precisión de 32 bits. Se muestra el número de bits para cada campo (los rangos de bits están en paréntesis cuadrados):

	Signo	Exponente	Fracción	Bias
Precisión de 32 bits	1[31]	8[30-23]	23 [22-0]	127

•El bit de signo

El bit del signo es muy sencillo. 0 denota un número positivo; 1 denota un número negativo.

•El exponente

El campo del exponente necesita representar tanto exponentes positivos y negativos. Para hacer esto, se añade un número o bias al exponente verdadero para obtener el exponente almacenado. Para precisión de 32 bits, este valor es 127. Así, un exponente de cero significa que se almacena 127 en el campo del exponente. Un valor almacenado de 200 indica un exponente de $(200-127)$, o 73. Para razones no discutidas aquí, los exponentes de -127 (todos 0s) y +128 (todos 1s) son reservados para números especiales.

•La mantisa

La mantisa representa los bits de precisión del número. Se compone de un bit delantero implícito y los bits de la fracción. Para averiguar el valor del bit delantero implícito, considera que cualquier número se puede expresar en notación científica de muchas maneras diferentes. Por ejemplo, el número cinco pueden ser representado como cualquiera de éstos:

$$\begin{aligned} &5,00 \times 10^0 \\ &0,05 \times 10^2 \\ &5000 \times 10^{-3} \end{aligned}$$

Para llevar al máximo la cantidad de números representables, los números de punto flotante se almacenan típicamente en forma normalizada. Esto pone básicamente la coma después del primer dígito que no sea cero. En forma normalizada, cinco es representado como $5,0 \times 10^0$.

Una optimización gradable está disponible en base dos, ya que el único dígito que no es cero posible es 1. Así, acabamos de asumir un dígito delantero de 1, y no necesitamos representarlo explícitamente. Como resultado, la mantisa tiene efectivamente 24 bits de resolución, y 23 bits de fracción.

Para hacer un resumen:

- 1.El bit del signo es 0 para positivo, 1 para negativo.
- 2.La base del exponente es dos.
- 3.El campo del exponente contiene 127 más el exponente verdadero con precisión de 32 bits

4.El primer bit de la mantisa se asume típicamente ser 1, f, donde f es el campo de bits de fracción.

Rangos de números de punto flotante

Consideremos por un momento números de punto flotante de precisión de 32 bits. Note que tomamos esencialmente un número de 32 bits y re-distribuimos los campos para cubrir un rango más amplio. Algo tiene que ceder, y es la precisión. Por ejemplo, enteros de 32 bits regulares, con toda precisión centrada en cero, puede almacenar precisamente enteros con 32 bits de resolución. Un número de punto flotante de precisión de 32 bits, por otro lado, es incapaz de lograr esta resolución con 24 bits. Sin embargo, se aproxima este valor al truncar los valores más bajos. Por ejemplo:

```
11110000 11001100 10101010 00001111 //entero de 32 bits =
+1,1110000 11001100 10101010 x 231de número de punto flotante de precisión de 32 bits
= 11110000 11001100 10101010 00000000 // valor correspondiente
```

Esto se aproxima al valor de 32 bits, pero no nos da una representación exacta. Por otro lado, además de la habilidad de representar los componentes fraccionarios (que los enteros no pueden hacer), el valor de punto flotante puede representar números alrededor de 2127, comparado con el valor de 32 bits del máximo de enteros alrededor de 232.

Mucho más se podría hablar sobre este sistema, pero en este libro eso es lo suficiente para los propósitos a ser alcanzados. Note que en realidad no es necesario conocer la convención ya que las operaciones toman cuenta de los cálculos en forma transparente.

Número en representación Gray

El código Gray es una sucesión binaria con la propiedad que sucede sólo un cambio de bit entre cualquiera de dos elementos consecutivos.

El código Gray se puede usar para convertir la posición angular de un disco a la forma digital (con un encoder, por ejemplo). Una línea radial de sensores lee el código desde la superficie del disco y si el disco está en el medio entre dos posiciones, cada sensor quizás lea su bit de ambas posiciones al mismo tiempo pero ya que sólo hay un bit de diferencia entre entre las dos posiciones, el valor leído es garantizado ser uno de los dos valores válidos antes que alguna combinación de un tercero (inválido).

Un algoritmo posible para engendrar una sucesión de código Gray deberá cambiar un bit que tiene como resultado un código nuevo cada vez. Aquí está una sucesión Gray de cuatro bits de código engendrada de esta manera:

Decimal	No. Gray
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111

6	0101
7	0100
8	1100
9	1101
10	1111
11	1110
12	1010
13	1011
14	1001
15	1000

Algunos encoders usan la convención Gray para entregar la información y entonces es necesario entender que es este sistema numérico para poder operarlo adecuadamente. Varios de los PLCs DirectLogic tienen una instrucción que maneja este código.

Valores numéricos en módulos análogos

Volviendo al sistema BCD o binario, Ud. puede haber entrado la configuración de un módulo análogo como número BCD, o Ud. puede configurar los módulos para traer los valores en un sistema numérico binario. Todo lo que Ud. debe saber es que, si Ud. usará este valor de este módulo análogo como un valor prefijado de temporizador o de un contador, los temporizadores y los contadores funcionan con BCD.

De modo que si va a usar el valor como un valor prefijado de un temporizador, los va a tener que entrar como BCD. Pero si ellos entran como binario, la CPU tiene una conversión dentro para convertir un valor binario en BCD. Esto se usa para convertir el modelo binario para hacerlo el modelo correcto. Es su preferencia de cómo Ud. quiere que ellos entren.

Ahora hay alguna confusión entre clientes. Cuando usa software de *DirectSOFT* y Ud. va a Data View, si Ud. baja el menú y Ud. lo pide en binario, *DirectSOFT* le dará ceros y unos. Le dará una serie de ceros y unos, le mostrará exactamente lo que se está en esa palabra. Si Ud. tecldea la flecha hacia abajo y Ud. escoge tipo BCD/Hex sumará toda la serie de ceros y unos en la palabra que Ud. seleccionó y le dirá el valor en un número BCD. Le dará de vuelta el número a que equivalen estos bits. El PLC cuida de la operación para Ud.

Ud. no tiene que recordar estos sistemas numéricos semejantes a éstos. La computadora lo hace por Ud. Los suma. Si trata de evaluar una palabra en un sistema numérico binario que usa 0s y 1s, (que *DirectSOFT* se refiere a como sistema numérico decimal).

Si escoge binario en Data View de DirectSOFT, Ud. verá 0s y 1s. Si escoge decimal va a ver el equivalente de decimal, equivalente a ceros y unos en formato binario.

Representación del complemento de 2

Veamos a continuación la definición de lo que es el complemento de 2, cómo calcular este, como sumarlos, como restarlos, como multiplicarlos y como dividirlos.

La representación del complemento de 2 es usada por números con signo en la mayoría de los computadores. Esta notación le permite a un computador sumar y restar números usando la misma operación (de modo que no es necesario implementar sumadores y restadores). En este caso se necesita tener un numero fijo de bits y el bit más significativo es el bit de signo. Esta misma representación es usada para representar números positivos y negativos.

Propiedades

La representación del complemento de 2 permite el uso de operaciones aritméticas binarias en enteros con signo, permitiendo resultados correctos del complemento de 2's.

Los números positivos del complemento de 2's se representan como números binarios simples.

Los números negativos del complemento de 2's se representan como números binarios de tal modo que cuando sumados a un número positivo de la misma magnitud resulta en un cero.

Vea la tabla a continuación para entender el concepto: Es este caso se hace con 8 bits, pero puede hacerse con la cantidad de bits que sea conveniente o que se desee.

Número Entero Complemento de 2

	Con signo	Sin Signo
5	5	0000 0101
4	4	0000 0100
3	3	0000 0101
2	2	0000 0010
1	1	0000 0001
0	0	0000 0000
-1	255	1111 1111
-2	254	1111 1110
-3	253	1111 1101
-4	252	1111 1100
-5	251	1111 1011

Nota: el bit más significativo (al extremo izquierdo) indica el signo del entero; por lo tanto se llama a veces el bit del signo

Si el bit del signo es cero => entonces el número es mayor que o es igual a cero, o positivo.

Si el bit del signo es uno => entonces el número es menor que cero, o negativo.

El cálculo del complemento de 2

Para calcular el complemento de 2 de un entero, invierta el equivalente binario del número cambiando todos los unos a ceros y todos los ceros a unos (llamado también el complemento de 1's), y luego sume uno.

Por ejemplo,

$$0001\ 0001 \text{ (binario 17)} \Rightarrow 1110\ 1111 \text{ (complemento de 2 -17)}$$

$$\text{invertir (0001 0001)} \Rightarrow 1110\ 1110 \text{ (Invierta los bits)}$$

$$1110\ 1110 + 0000\ 0001 = >1110\ 1111 \text{ (Sume 1)}$$

Suma del Complemento de 2

Sigue las mismas reglas de la adición binaria.

Por ejemplo,

$$5 + (-3) = 2 \quad 0000\ 0101 = +5$$

$$+ 1111\ 1101 = -3$$

$$0000\ 0010 = +2$$

Substracción del Complemento de 2

La resta del complemento de 2 es la adición binaria del minuendo al complemento del 2's del sustraendo (agregar un número negativo es lo mismo que restar un 1 positivo).

Por ejemplo,

$$7 - 12 = (-5) \quad 0000\ 0111 = +7$$

$$+ 1111\ 0100 = -12$$

$$1111\ 1011 = -5$$

Multiplicación del Complemento de 2

Sigue las mismas reglas de la multiplicación binaria.

Por ejemplo,

$$(-4) \times 4 = (-16) \quad 1111\ 1100 = -4$$

$$\times 0000\ 0100 = +4$$

$$1111\ 0000 = -16$$

División del Complemento de 2

La división del complemento de 2 es la sustracción repetida del complemento de 2. Se calcula el complemento de 2 del divisor y luego es añadido al dividendo. Para el próximo ciclo de sustracción, el cociente reemplaza el dividendo. Esto se repite hasta que el cociente sea demasiado pequeño para la sustracción o sea cero, entonces llega a ser el resto.

La respuesta final es el suma de ciclos de sustracción más el resto.

Por ejemplo,

$$\begin{array}{r}
 7 \div 3 = 2 \text{ resto } 1 \quad 0000 \ 0111 = +7 \qquad 0000 \ 0100 = +4 \\
 \qquad \qquad \qquad + 1111 \ 1101 = -3 \qquad \qquad + 1111 \ 1101 = -3 \\
 \hline
 \qquad \qquad \qquad 0000 \ 0100 = +4 \qquad \qquad 0000 \ 0001 = +1 \text{ (resto)}
 \end{array}$$

Para hacer la extensión del signo para extender un entero con signo de 8 bits a uno de 16 bits o de 16 bits a uno de 32 bits, añada los bits adicionales en el lado izquierdo del número. Llene cada bit extra con el valor del número más pequeño los bits más significativos (el bit del signo).

Por ejemplo,

Entero con signo	Representación de 8 bits	Representación de 16 bits
- 1	1111 1111	1111 1111 1111 1111
+1	0000 0001	0000 0000 0000 0001

Esperamos que este apéndice le deje claro los sistemas numéricos que son usados en los PLCs. Se pueden consultar muchas más informaciones en libros especializados de matemáticas y debemos decir que mayores explicaciones están fuera del alcance de este manual.



Productos de AUTOMATIONDIRECT.COM y tipos de datos

PLCs *Direct*LOGIC

La familia de PLCs *Direct*LOGIC usa el sistema de numeración octal para toda las entradas y salidas. Todas las memorias V se almacenan en formato BCD a menos que sea cambiada específicamente por el programador. Esto significa que todas las operaciones aritméticas usadas en programas de lógica ladder se deben hacer con operadores BCD. Se tiene también la opción de usar números binarios o reales así como valores de doble palabra BCD en algunos de los modelos del PLC.

Para cambiar el tipo de formato los datos, usted debe usar una instrucción de programa. Por ejemplo, para cambiar un valor de formato BCD a formato binario, use una instrucción BIN. También, al cambiar el formato desde BCD a un número real o de punto flotante, use una instrucción BTOR.

Un número en formato BCD o un número en formato binario no se puede sumar a un número verdadero, o un número en formato BCD a un número de formato binario, y luego va a obtener un resultado correcto. Los formatos de datos deben ser iguales .

Éstos son algunos asuntos que debe saber en relación con que todos los datos de memoria V que están en formato BCD por defecto.

- Uno es que los módulos análogos se pueden configurar para dar resultados em formato binarios o en BCD, de modo que es necesario saber como se está usando el módulo.
- PID es otra área donde no todos los valores están en formato BCD. De hecho, casi todos los parámetros de la tabla de PID se almacenan en la memoria del PLC como números de formato binario.



NOTA: El algoritmo de PID usa magnitud más signo para los números binarios negativos, mientras que las funciones estándares usan el complemento de dos. Esto puede causar confusión mientras se buscan errores de un lazo de PID.

- Por último, cuando se usa está utilizando Data View en *Direct*SOFT. Valores en formato binario, hexadecimal y decimales se almacenan de la misma manera en el PLC y son todos son llamados formato binario. La única diferencia entre todos es siempre que se vean usando Data View. Asegúrese que ha seleccionado el formato apropiado en la ventana de Data View. También note que el valor BCD está llamado BCD/Hex.

Recuerde de la tabla 8 (página I-6), BCD y hexadecimal son realmente iguales aunque las letras A a F no están implicadas; comparten un formato de representación aunque los valores son diferentes. Aquí es donde es crucial la buena documentación del formato de datos almacenado en memoria.

DIRECTIVAS DE LA UNION EUROPEA (CE)



En este apéndice...

Directivas de la Unión Europea	J-2
Guías básicas de instalación EMC	J-4

Directivas de la Unión Europea (EU)



NOTA: La información contenida en esta sección se piensa como una regla y se basa en nuestra interpretación de varios estándares y requisitos. Puesto que los estándares reales son publicados por otros organismos, y en algunos casos agencias gubernamentales, los requisitos pueden cambiar en un cierto plazo sin previo aviso o advertencia. Cambios a los estándares pueden posiblemente invalidar la información proporcionada en esta sección.

Esta área de certificación y de aprobación es absolutamente vital a cualquier persona que desee hacer negocio en Europa. Una de las tareas claves que tuvieron los países miembros de la EU y el área económica europea (EEA) fue el requisito de traer varios estándares similares distintos a un estándar común para todos los miembros. El propósito primario de un estándar único era hacer más fácil vender y transportar mercancías entre los varios países y mantener un ambiente de trabajo y de vida seguro. Las directivas que resultaron de esta combinación de estándares ahora son requisitos legales para hacer negocio en Europa. Los productos que cumplen estas directivas deben tener una marca CE para mostrar que cumplen las reglas.

Países miembros

Con fecha 23 de Julio de 2002, los miembros del EU son Austria, Bélgica, Dinamarca, Finlandia, Francia, Alemania, Grecia, Irlanda, Italia, Luxemburgo, los Países Bajos, Portugal, España, Suecia, y el Reino Unido. Islandia, Liechtenstein y Noruega junto con los miembros de la EU constituyen el área económica europea (EEA) y todos son cubiertos por las directivas.

Directivas aplicables

Hay varias directivas que se aplican a nuestros productos. Las directivas se pueden enmendar, o agregar, como sea requerido.

- **La directiva de la compatibilidad electromagnética (EMC)** — esta directiva procura asegurarse de que los dispositivos, el equipo, y los sistemas tienen la capacidad de funcionar satisfactoriamente en un ambiente electromagnético sin introducir disturbio electromagnético intolerable a cualquier cosa en ese ambiente.
- **Directiva de la seguridad de la maquinaria** — esta directiva cubre los aspectos de seguridad del equipo, de la instalación, del etc. Hay varias áreas implicadas, incluyendo los estándares de prueba que cubren inmunidad de ruido eléctrica y la generación de ruido.
- **Directiva de baja tensión** — esta directiva es también relacionada a seguridad y cubre el equipo eléctrico que tiene gamas del voltaje de 50-1000vac y/o de 75-1500vdc.
- **Directiva de la batería** — esta directiva cubre la producción, reciclado y disposición de baterías.

Conformidad a las directivas

Ciertos estándares dentro de cada directiva requieren ya conformidad obligatoria. La directiva de EMC, que ha ganado gran atención, llegó a ser obligatoria en fecha de 1. de Enero de 1996. La directiva de baja tensión llegó a ser obligatoria con fecha de 1o. de Enero de 1997.

En última instancia, somos todos responsables de nuestros varios pedazos del rompecabezas. Como fabricantes, debemos probar nuestros productos y documentar cualesquier resultado de pruebas y/o procedimientos de instalación que sean necesarios para cumplir con las directivas. Como constructor de la máquina, usted es responsable de instalar los productos de una manera que mantenga y asegure conformidad. Usted es también responsable de probar cualquier combinación de los productos que pueden (o no pueden) cumplir con las directivas cuando se usan juntos.

El usuario final de los productos debe cumplir con cualquiera directiva que pueda cubrir mantenimiento, disposición, etc. del equipo o componentes varios. Aunque nos esforzamos en proporcionar la mejor ayuda disponible, es imposible que probemos todas las configuraciones posibles de nuestros productos con respecto a cualquier directiva específica. *Debido a esto, es en última instancia responsabilidad del cliente asegurarse de que su maquinaria (en su totalidad) cumple con estas directivas y continuará a cumplirlas con las prácticas aplicables que se requieren para la cumplimiento de éstas.*

Con fecha 10. de Enero de 1999, el DL05, DL06, DL205, DL305, y los sistemas del PLC DL405 fabricados por las industrias de Koyo Electronics o FACTS Engineering, cuando están instalados correctamente y utilizados, cumplen con los requisitos directivos de compatibilidad electromagnética (EMC), de la directiva de la baja tensión y directivas de la maquinaria en los estándares siguientes.

- **Estándares directivos de EMC relevantes a PLCs**
 - EN50081–1 Generic emission standard for residential, commercial, and light industry
 - EN50081–2 Generic emission standard for industrial environment.
 - EN50082–1 Generic immunity standard for residential, commercial, and light industry
 - EN50082–2 Generic immunity standard for industrial environment.
- **Estándares directivos de baja tensión aplicables a PLCs**
 - EN61010–1 Safety requirements for electrical equipment for measurement, control, and laboratory use.
- **Estándar específico del producto para PLCs**
 - EN61131–2 Programmable controllers, equipment requirements and tests. Este estandar reemplaza los estandares genéricos de más arriba para inmunidad y seguridad. Sin embargo, los estandares genéricos de emisiones deben ser usados junto con los siguientes estandares :
 - EN61000-3-2 Harmónicas
 - EN 61000-3-2 Fluctuaciones
- **Cuidado en descargas electrostáticas (ESD) (ESD)**

Recomendamos que todo el personal tome las precauciones necesarias para evitar el riesgo de transferencia de cargas atmosféricas dentro del gabinete de control, y deben ser proporcionadas advertencias e instrucciones claras en el exterior del gabinete. Tales precauciones pueden incluir, el uso de las correas de descarga a tierra, dispositivos similares o el accionar apagado del equipo dentro del recinto antes que la puerta se abra.
- **Cuidado en interferencia de radio (RFI)**

Esto es un producto de la clase A. En un ambiente doméstico este producto puede causar radio interferencia y en tal caso puede ser requerido que el usuario tome medidas adecuadas.

Seguridad general

- Se requieren desconectores externos, interruptores o fusibles externos para estos dispositivos.
- El desconector o el interruptor se debe montar cerca del equipo del PLC.

AUTOMATIONDIRECT está actualmente en el proceso de cambiar sus métodos de prueba de los estándares genéricos a los estándares específicos del producto.

Manual de instalación especial

Los requisitos de la instalación para cumplir con los requisitos de las directivas de la maquinaria, de la baja tensión y de EMC son levemente más complejos que los requisitos normales de la instalación encontrados en los Estados Unidos. Para ayudar con esto, hemos publicado un manual especial que usted puede pedir o bajar desde nuestro sitio de Internet:

- DA-EU-M – Manual de la instalación de CE que cubre requisitos de la instalación especial para resolver los requisitos de directivas del CE. Pida este manual para obtener la información más actualizada.

Otras fuentes de información

Aunque las directivas EMC consiguen la mayoría de la atención, otras directivas básicas, tales como la directiva de maquinaria y de baja tensión, también colocan restricciones en el constructor del panel de control. Debido a estos requisitos adicionales se recomienda que las publicaciones siguientes sean compradas y utilizadas como pautas:

- BSI publication TH 42073: Febrero 1996 – cubre la seguridad y los aspectos eléctricos de las directivas de la maquinaria.
- EN 60204-1:1992 – requisitos eléctricos generales para la maquinaria, incluyendo baja tensión y consideraciones de EMC
- IEC 1000-5-2: EMC Requisitos para aterramiento y cableado
- IEC 1000-5-1: EMC consideraciones generales

Puede ser posible que usted obtenga esta información localmente; sin embargo, la fuente oficial de directivas aplicables y de estándares relacionados es:

La oficina para las publicaciones oficiales de las Comunidades Europeas L-2985 Luxembourg; el contacto más rápido es a través de Internet : <http://euro-op.eu.int/indexn.htm>

Otra fuente es:

British Standards Institution – Sales Department

Linford Wood

Milton Keynes

MK14 6LE

United Kingdom; el contacto más rápido es a través de Internet : <http://www.bsi.org.uk>.

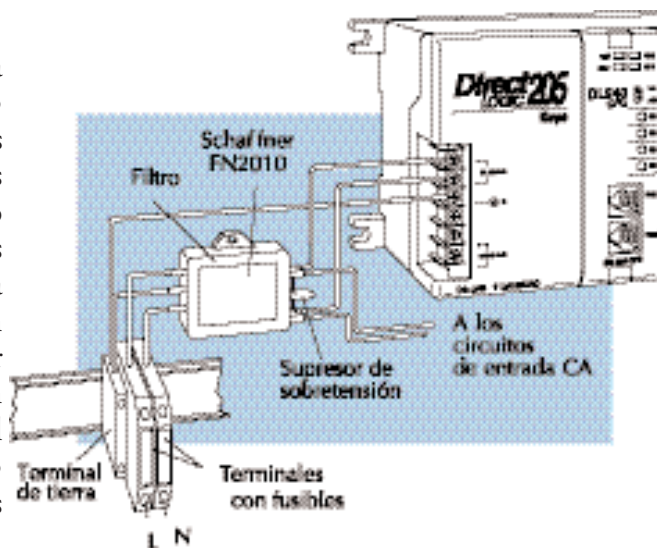
Pautas básicas de instalación de EMC

Gabinetes

La manera más simple de resolver los requisitos de seguridad de las directivas de maquinaria y de baja tensión es contener todo el equipo del control en un gabinete metálico positivo de trabar estándar industrial. Esto tiene normalmente una ventaja agregada porque también ayudará a asegurarse de que las características de EMC están en conformidad con los requisitos de las directivas EMC. Aunque las emisiones de RF del equipo del PLC, cuando sean medidas en un ambiente abierto, estuviesen bien debajo de los límites de EMC, ciertas configuraciones pueden aumentar los niveles de emisión. Los agujeros en el recinto para el paso de cables o para montar interfaces de operador, aumentarán a menudo las emisiones.

Filtros de alimentación en CA

Las fuentes de alimentación en corriente alterna de los PLCs DL05, DL06, DL205, DL305 requieren filtros adicionales para cumplir con las directivas de EMC sobre emisiones conducidas de RF. Todo el equipo del PLC se ha probado con los filtros Schaffner, que reducen los niveles de las emisiones si los filtros se ponen a tierra correctamente. Se debe seleccionar un filtro con una corriente nominal conveniente para proveer todos los módulos de las fuentes de alimentación del PLC y de entrada de CA. Sugerimos el FN2010 para los sistemas DL05/DL06/DL205 y el FN2080 para los sistemas DL305. Los sistemas DL405 no requieren filtrado adicional.



NOTA: *Muy pocos filtros de red pueden reducir el problema de emisiones a niveles insignificantes. En algunos casos, los filtros pueden aumentar las emisiones conducidas si no corresponden a las emisiones del problema.*

Supresores de sobretensión y fusibles

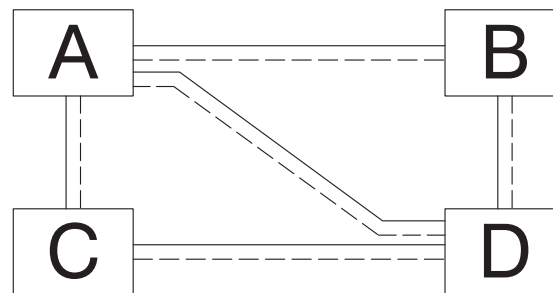
Para cumplir con los requisitos del riesgo de fuego eléctrico de los estándares de baja tensión y de la maquinaria en la directiva EN 61010-1 y EN 60204-1, limitando la energía en los circuitos "ilimitados" de las redes con los conductores de energía invertidos, es necesario colocar fusibles a las entradas de la fuente de CA y de C.C. Se debe también instalar un supresor transitorio de tensión a través de las conexiones de entrada de energía del PLC. Escoja un supresor tal como un varistor, con un grado de voltaje de trabajo 275VAC para las fuentes nominales 230V (voltaje de trabajo 150VAC para las fuentes 115V) y capacidad de alta energía (eg. 140 julios).

Los supresores transitorios se deben proteger por fusibles y la capacidad del supresor transitorio debe ser mayor que las características de fusión de los fusibles o de los interruptores para evitar un riesgo de fuego. Un arreglo recomendado de la entrada de la fuente de CA para PLCs Koyo es utilizar los terminales con fusibles TT gemelos de 3 A con indicación de fusible quemado, tal como terminales de DINectores DN-F10L, o interruptores duplos conectados a un filtro Schaffner FN2010 o equivalente, con el supresor transitorio de alta energía soldado directamente a través de los terminales de salida del filtro. Las entradas del sistema del PLC se deben también proteger contra impulsos de tensión derivando su energía de la fuente con fusibles, filtrada y con supresores de sobretensión.

Aterramiento interno del gabinete

Se debe proporcionar un bloque de terminales de tierra en cada cubículo para la conexión de todas las barras de tierra, conexiones de la tierra protectora, conductores de tierra del filtro de la red y conexiones de tierra del conjunto mecánico. Esto se debe instalar para cumplir con la seguridad y los requisitos de EMC, los estándares locales y los requisitos encontrados en las directivas de la maquinaria del IEC 1000-5-2. También se requiere que los terminales comunes de los módulos de entrada del PLC y el lado común de la fuente de las cargas controladas por módulos de salidas del PLC sean conectados con el terminal de tierra protectora.

Aterramiento equipotencial



Key ——— Cable de comunicación serial
----- Unión equipotencial

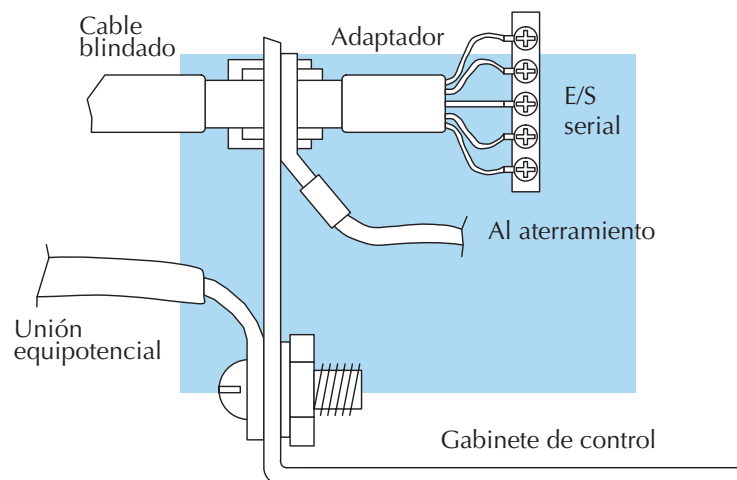
Debe hacerse la tierra del sitio adecuadamente para equipos que contienen circuitos electrónicos modernos. El uso de electrodos aislados de tierra para sistemas electrónicos es prohibido en algunos países. Asegúrese de usted ha verificado cualquier requisito para su destino particular. IEC 1000-5-2 cubre el tema de las redes de tierra equipotencial adecuadamente, pero se debe dar atención especial a los aparatos y a los cubículos de control que contienen los dispositivos de E/S, estantes remotos de E/S o tienen comunicaciones entre sistemas con el gabinete primario del sistema del PLC. Se debe instalar un conductor para conexión equipotencial junto a todos los cables seriales de comunicaciones y a cualquier artículo separado de la planta que contiene los dispositivos de E/S conectados con el PLC. El diagrama muestra un ejemplo de cuatro localizaciones físicas conectadas por un cable de comunicaciones.

Comunicaciones y cables blindados

Se recomiendan cables blindados de pares torcidos de sección mínima 24 AWG de buena calidad, con blindaje total de hoja y trenzados para las señales análogas y para comunicaciones que van fuera del recinto del PLC.

Hasta la fecha ha sido una práctica común conectar solamente una tierra a un extremo del blindaje del cable para reducir al mínimo el riesgo del ruido causado por las corrientes de tierra del lazo de la tierra entre los aparatos. El procedimiento solamente de poner a tierra un extremo, que se originó sobre todo como resultado de intentar reducir ruido en sistemas de audio, no es más aplicable en un ambiente industrial complejo.

Los cables blindados son también emisores eficientes del ruido del RF del sistema del PLC y pueden interactuar recíprocamente de una manera parásita en redes y entre las fuentes múltiples de interferencia.



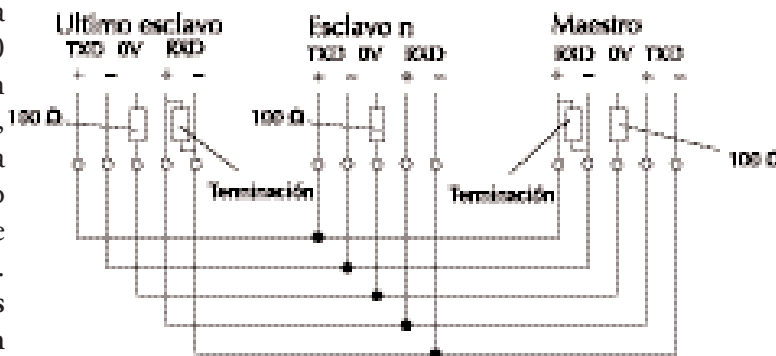
La recomendación es utilizar los cables blindados como "ductos electrostáticos" entre el aparato y los sistemas e instalar conductores de cobre en enlace equipotencial de gran sección junto a todos los cables blindados. Cuando los cables blindados pasan a través de la pared metálica de un recinto o de una máquina, es recomendado en IEC 1000-5-2 que el blindaje se conecte en el perímetro completo a la pared, preferiblemente usando un adaptador conductor y no a través de una conexión de un alambre a un perno de tierra. Los blindajes de cables se deben conectar con cada pared del gabinete o de la cubierta de la máquina que pase a través.

Cables de señales análogos y RS232

El abastecimiento de una tierra para ambos extremos del blindaje para los circuitos análogos proporciona el ambiente eléctrico perfecto para el cable del pares trenzados mientras que el lazo consiste en la señal y el retorno, en un arreglo perfectamente equilibrado del circuito, con la conexión al común del circuito de entrada hecho en los terminales del módulo. Los cables RS232 se manejan de la misma manera.

Cables multinodo

Los cables de doble par trenzados RS422 y par simple RS485 también requieren una conexión a 0V, que ha sido proporcionada a menudo en el pasado por el blindaje del cable. Ahora se recomienda que use un cable de par trenzado triple para los enlaces RS422, y el cable gemelo de pares trenzados para enlaces RS485. Esto es porque el par adicional se puede utilizar como el conductor de 0 Volt entre sistemas. Con la tierra de las fuentes de potencia CC se crean lazos de aterramiento en ambos sistemas, se crean de este modo a través de los conductores de 0 Volt. Las guías de instalación favorecen a crear lazos de tierra, que son mantenidos en una impedancia baja usando conductores de enlace equipotencial de gran sección. Para tomar en cuenta instalaciones no europeas usando tierra en solo un extremo y sitios con características de tierra deficientes, recomendamos agregar resistencias de 100 ohm en cada conexión de 0V en cables de red y de comunicaciones.



Cables blindados dentro de gabinetes

Cuando usted instala cables entre dispositivos del PLC dentro de un gabinete que también contenga equipo electrónico susceptible de otros fabricantes, recuerde que estos cables pueden ser una fuente de emisiones de radio frecuencia (RF).

Hay formas de reducir al mínimo este riesgo. Los cables estándares de datos que conectan PLCs y/o las interfases del operador se deben instalar bien lejos del otro equipo y de sus cables asociados.

Usted puede hacer cables seriales especiales donde el blindaje del cable está conectado con la tierra del gabinete en ambos extremos, de la misma manera que los cables externos están conectados.

Aislamiento de una red de comunicación

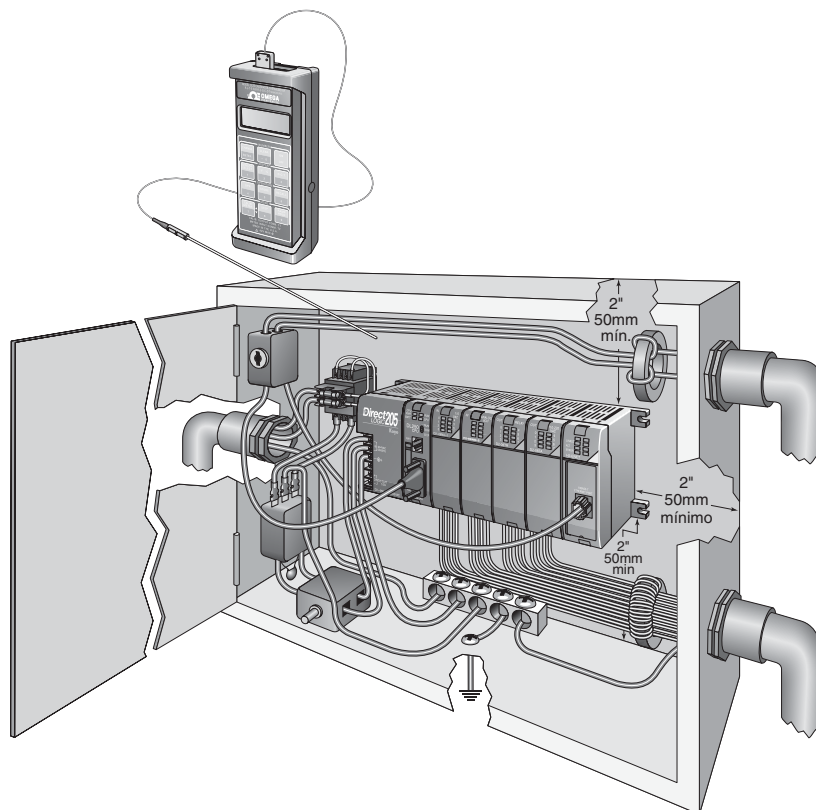
Por razones de seguridad, es un requisito específico de la directiva de maquinarias que debe ser instalado un desconectador con llave para la condición de que aisle cualquier señal de entrada de la red durante mantenimiento, para no poder recibir comandos remotos que podrían dar lugar a una operación de la maquinaria. El FA-ISOCÓN no tiene un desconectador con llave! Use un pestillo de seguridad y energice su gabinete de modo que cuando esté abierto quite energía del FA-ISOCÓN. Para evitar la introducción de ruido en el sistema, cualquier montaje del desconectador con llave se debe contener en su propia caja de acero puesta a tierra y se debe mantener la integridad del cable blindado.

Una vez más, para información adicional sobre directivas de CE recomendamos que consiga una copia de nuestro manual de instalación de CE (DA-EU-M). También, si usted está conectado con Internet, usted puede visitar el sitio oficial de la Comisión del EU en: <http://eur-op.eu.int/>

Versiónes alimentadas por corriente continua

Debido a emisiones levemente más altas irradiadas por la C.C., las versiones del DL06 y el funcionamiento de las emisiones difieren para diversos voltajes de fuente de C.C., se deben seguir las estipulaciones siguientes:

- El PLC debe estar dentro de un recinto metálico con una cantidad mínima de orificios.
- Los cables de E/S y las comunicaciones que están saliendo del gabinete deben estar dentro de conducto metálico.



Asuntos específicos al PLC DL06

- El grado entre todos los circuitos en este producto se clasifica como aislamiento básico solamente, como apropiado para las solas condiciones de avería.
- No hay aislación ofrecida entre el PLC y las entradas análogas de este producto.
- Es responsabilidad del diseñador del sistema conectar a tierra un lado de todos los circuitos de control y de energía, y conectar a tierra el blindaje de cables blindados.
- Este equipo debe ser instalado correctamente al seguir las guías en el manual de instalación DA-EU-M del PLC, y a de los estándares de la instalación IEC 1000-5-1, a IEC 1000-5-2 y a IEC 1131-4.
- Es un requisito que todo el equipo del PLC se debe contener en un gabinete de acero protector, que limita el acceso a los operadores usando una cerradura y un interruptor de energía. Si se necesita tener acceso por operadores o personal sin experiencia, el equipo se debe instalar dentro de una cubierta interna o de un gabinete secundario.
- Debe observarse que los requisitos de seguridad de la directiva de maquinaria EN60204-1 indican que todos los circuitos de energía del equipo se deben conectar con transformadores de aislamiento o fuentes de alimentación aisladas, y que un lado de los circuitos de control de CA o de C.C. se debe conectar a tierra.
- Ambas conexiones de entrada de energía al PLC se deben proteger con fusibles por separado usando el tipo de fusibles de 3 amperios T y un supresor de transientes adecuado para limitar sobretensiones de la alimentación.
- Si se llama la atención al usuario colocando una nota en la documentación que si el equipo se utiliza de una manera no especificada por el fabricante, se puede deteriorar la protección proporcionada por el equipo.

J

INTRODUCCIÓN A COMUNICACIONES SERIALES



En este apéndice...

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
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 <i>MODBUS RTU</i>	K-29
Introducción a la comunicación <i>MODBUS RTU</i>	K-29
Convención de direcciones en <i>MODBUS</i>	K-32
Operación de un esclavo en una red <i>MODBUS RTU</i>	K-35
Operación de un maestro en una red <i>MODBUS RTU</i>	K-41
Configuración del puerto como <i>MODBUS RTU</i>	K-45
Operación como maestro usando MRX y MWX	K-50
Ejemplos para redes <i>MODBUS RTU</i>	K-51
Respuesta de excepción en <i>MODBUS</i>	K-52
Comunicación con ASCII non sequence	K-51
Ejemplo para Comunicación ASCII	K-59

Introducción a comunicaciones seriales

Los PLCs *Direct*LOGIC® tienen dos puertos seriales incorporados de comunicación que se pueden usar para comunicarse a otros PLCs o a otros dispositivos seriales. Se entrega aquí una breve introducción a las comunicaciones seriales, para entender las capacidades y las limitaciones de los puertos seriales.

Hay tres componentes importantes a cualquier configuración de comunicación serial :

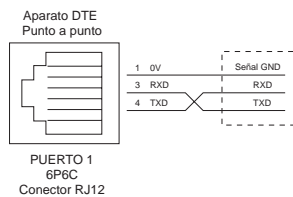
- El estándar de cableado
- El protocolo de comunicaciones
- Los parámetros de comunicaciones

Cada uno de éstos será discutido más detalladamente en relación a los PLCs *Direct*LOGIC.

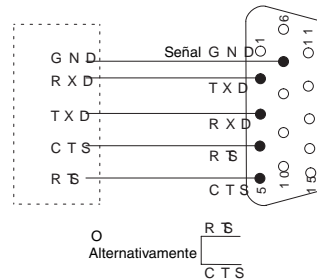
Estándar de cableado

Hay tres estándares de cableado que se pueden utilizar con los PLCs *Direct*LOGIC: RS-232C, RS-422 and RS-485.

RS-232C es un estándar de cableado punto a punto con una distancia práctica de cableado de 15 metros, o 50 pies, máximo. Esto significa que sólo dos dispositivos pueden comunicarse en una red de RS-232c, un sólo dispositivo maestro y un sólo dispositivo esclavo, y la longitud de cable total no debe exceder 50 pies. Se recomienda un cable Belden® 8102, o equivalente, para las redes RS-232c. Un cable Belden 9729 también funcionará.

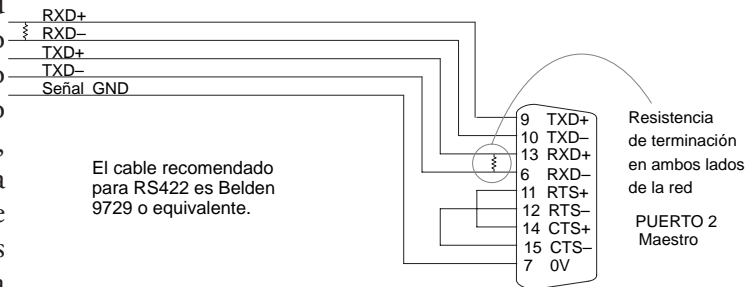


Conexiones del puerto 1

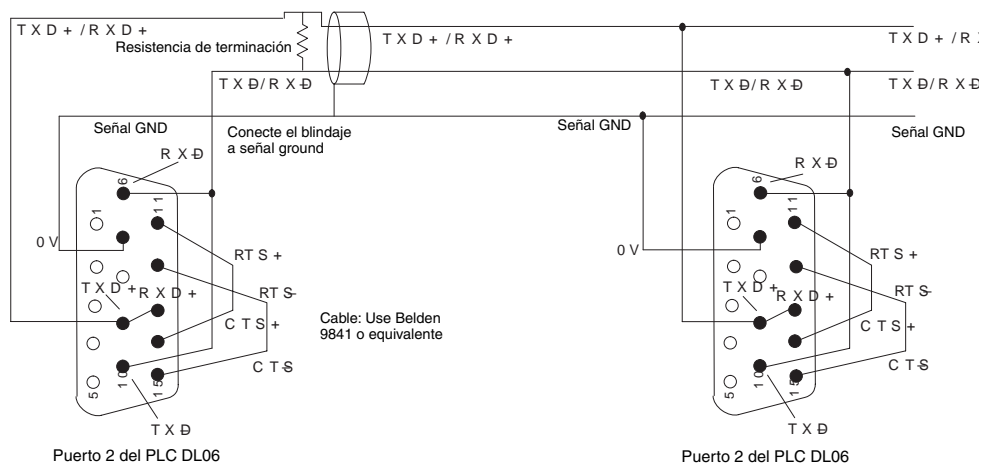


Conexiones del puerto 2

RS-422 es un estándar de múltiples puntos con una distancia práctica de cableado de 1000 metros, o 3280 pies, longitud total máxima. Esto significa que puede comunicarse un maestro solamente, con hasta 10 esclavos, y la distancia total de toda la red entre todos los dispositivos no puede exceder 1000 metros, o 3280 pies. El estándar RS-422 no especifica una topología de la red, pero en la práctica, una topología de conexiones múltiples en serie con el maestro en un extremo es la única manera de hacer la comunicación confiable. Se recomienda el cable Belden® 8102, o equivalente, para las redes RS-422. Use una resistencia de terminación igual en valor a la impedancia característica del cable que es usado (100 Ω for Belden 8102).



RS-485 es un estándar de cableado de múltiples puntos, con una distancia práctica de cableado de 4000 pies máximo. Este estándar de cableado preve la posibilidad de hasta 32 maestros que se comunican con hasta 32 esclavos todos dentro de la distancia máxima de 4000 pies. Observe que mientras que el estándar RS-485 prevé maestros múltiples en la misma red, los PLCs *Direct*LOGIC **no soportan multiple maestros en una red**. El estándar RS-485 no especifica una topología de red, pero en la práctica, una topología de conexiones múltiples en serie con el maestro en un extremo es la única manera de hacer la comunicación confiable. Se recomienda el cable Belden 9841, o equivalente, para las redes RS-485. Use una resistencia de terminación igual en valor a la impedancia característica del cable que sea usado (120 Ω para Belden 9841).



Protocolos de comunicaciones disponibles

Un protocolo de comunicaciones es el 'lenguaje' de los dispositivos en uso de la red para comunicarse uno con otro. Todos los dispositivos en la red deben usar el mismo protocolo de comunicaciones para poder comunicarse entre ellos. Los protocolos disponibles con PLCs *Direct*LOGIC están listados en la siguiente tabla.

Protocolos de comunicaciones							
Protocolo	Maestro	Esclavo	Puerto 1*	Puerto 2	RS-232C	RS-422	RS-485**
K-Sequence	No	Sí	Sí	Sí	Sí	Sí	No
Direct NET	Sí	Sí	Sí	Sí	Sí	Sí	No
MODBUS RTU	Sí	Sí	Sí	Sí	Sí	Sí	Sí
ASCII	Out	In	No	Yes	Sí	Sí	No

* El puerto 1 trabaja solamente como esclavo y es RS-232c, con parámetros fijos de comunicaciones de 9600 kbps, 8 bits de datos, 1 bit de partida, 1 bit de parada, paridad impar y dirección de estación 1. Es un puerto DTE asincrónico, half-duplex y selecciona automáticamente entre los protocolos KSequence, *Direct*NET and MODBUS RTU.

** RS-485 is available on Port 2 for MODBUS RTU protocol only.

El protocolo **K-Sequence** no está disponible para el uso como maestro en el PLC DL06. Por lo tanto, no puede ser usado para el establecimiento de una red entre PLCs, ya que el uso en el PLC DL06 es solamente como esclavo para el software de programación *Direct*SOFT y para algunas interfaces de operador.

Apéndice K: Introducción a comunicaciones seriales

El protocolo *DirectNET* está disponible para uso por un PLC DL06 maestro o esclavo. Ésto, y el hecho de que es el protocolo 'nativo', le hace ideal para la comunicación PLC a PLC en una red de múltiples nodos con las instrucciones RX y WX. Una desventaja de este protocolo es que no puede transmitir datos de estado de solamente un bit. Tiene que ser en grupos de 16 bits.

El protocolo *MODBUS RTU* es un protocolo estándar muy común en la industria, y se puede usar como maestro o esclavo en el PLC DL06 para comunicarse con una amplia variedad de dispositivos industriales que apoyen este protocolo.

ASCII es otro estándar muy común en la industria, y se utiliza comúnmente donde deban ser transferidos datos de caracteres alfanuméricos. Muchos dispositivos de entrada tales como lectores de código de barras y balanzas electrónicas usan el sistema con caracteres ASCII, y también muchos dispositivos de salida aceptan comandos ASCII.

No importa cual estándar o protocolo de cableado se use, hay varios parámetros de comunicaciones que deben seleccionarse para cada dispositivo antes de que ellos puedan establecer la comunicación. Estos parámetros incluyen

Tasa de transmisión en kilobits/s(kbps)	Control de flujo
Bits de datos	Supresión de eco
Paridad	Tiempos de timeouts
Bits de parada	Tiempos de retraso
Dirección del nodo	Formato

Todos estos parámetros pueden no ser necesarios, o no estar disponibles, para su uso. Los parámetros usados dependerán del protocolo que es usado, y si el dispositivo es un maestro o un esclavo.

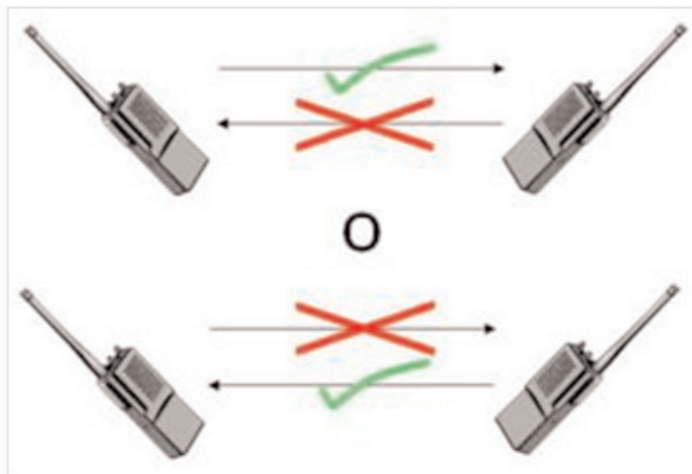


NOTA: El punto a recordar es que cuando hay el mismo tipo de parámetros disponible en el maestro y en el esclavo (es decir velocidad, paridad, bits de parada, etc), los valores de éstos deben ser iguales.

La transmisión serial puede ser del tipo half duplex or full duplex. Una transmisión half duplex puede transmitir en ambas direcciones, pero en una dirección cada vez. Una full duplex puede suceder simultáneamente.

Para hacer una similitud simple, se puede decir que un walkie talkie es un aparato half duplex, porque puede transmitir la voz solamente en una dirección. En contraste, un aparato telefónico es un aparato full duplex, ya que ambas partes pueden conversar simultáneamente.

Los PLCs *DirectLOGIC* pueden transmitir datos solamente en forma half duplex.

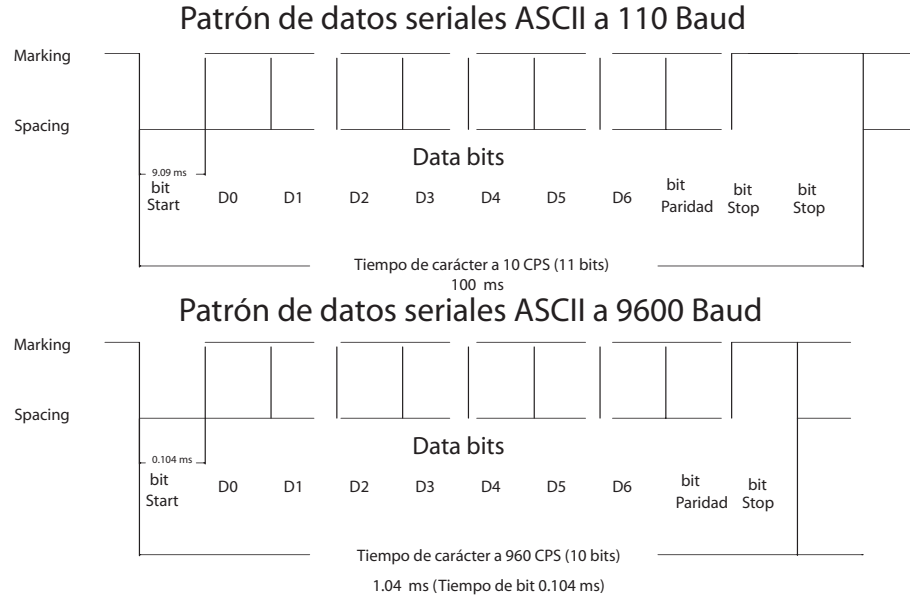


Una simple ilustración de un sistema half duplex.

Conceptos de transmisión de datos seriales

La transmisión de datos usa transmisión de datos digital asincrónica serial. Los caracteres se envían codificados en bits, un bit cada vez, en niveles de señal tales como los definidos en RS-232c.

El código más común para la transmisión de datos asincrónicos es el Código Americano para el Intercambio de Información (ASCII). Según lo utilizado originalmente en el teletypewriter (TTY), es un código de 11 bits con un bit de START y otro de STOP. El patrón básico para este estándar se muestra en la figura siguiente:

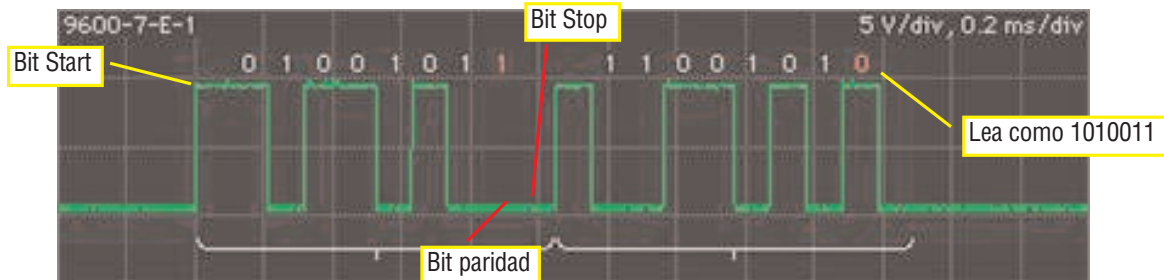


K

Cuando la línea no está transmitiendo ningún dato, está constantemente en MARK o estado 1. El comienzo de un carácter es señalado por el bit START, que lleva la línea a 0 o al estado SPACE por el período de un bit. Los 7 bits que siguen el bit START son los bits de datos de caracteres. Los bits se envían con el bit menos significativo (LSB) primeramente. El código ASCII utiliza 7 bits para generar 128 códigos únicos (Vea el apéndice G). Estos caracteres incluyen letras A a Z, en mayúscula o minúscula, los números 0 a 9, y otros símbolos de puntuación y matemáticos, e incluso otros caracteres de control.

El carácter consiste de 7 bits de datos y un bit de paridad. El estado del bit de paridad depende si se desea paridad par, impar o ninguna.

Después que pasen los datos y un bit de paridad, la línea de transmisión debe pasar a HIGH por 1 o 2 tiempos de bit. Éstos son los bits de parada. La forma de verlos en un osciloscopio es mostrado como ejemplo en la siguiente figura:



Este tren de pulsos no puede ser transmitido a más de 40 a 50 pies (15 metros) con cables de pares torcidos estándares.

Estas señales eran muy usadas con módems, dispositivos que permiten transmitir datos a largas distancias. Un módem es un modulador-demodulador, que transforma los pulsos en tonos de audio y entonces las señales se podrían transmitir en un circuito de teléfono. En el otro lado del dispositivo remoto, el mismo módem demodula las señales de audio y las entrega al equipo remoto como datos digitales.

El estándar de RS232c es una interfase popular que se puede describir eléctricamente por la notación siguiente:

Estado binario del bit de datos	1	0
Condición de la señal	Mark	Space
Función	OFF	ON
Voltaje	-3 a -15 V	+3 a +15 Volt

Tiempo típicos un bit transmitido a 9600 baud: 0,104 ms.

Las señales de transmisión y recepción de datos deben estar entre +3 a +15 V para una condición ON o Space o entre -3 - 15 V para una condición OFF o Mark.

Para transmisión asincrónica, las cinco señales siguientes van entre los dos dispositivos que se están comunicando:

- 1.- TX - Transmite datos desde el maestro al esclavo o modem. Estos son los datos que serán transmitidos.
- 2.- RX Recibe datos - desde el esclavo o modem al maestro. Éstos son los datos recibidos por el maestro desde el esclavo.
- 3.- RTD- Request to send - Pedido de envío desde el dispositivo digital al módem. Esta señal debe ser un 0 (ON) cuando los datos deban ser transmitidos. En una transmisión half duplex, las comunicaciones, se usan para controlar la dirección de la transmisión. Cuando está cambiado a 1, el portador local del módem se apaga, lo que le dice al extremo remoto que puede comenzar a enviar datos. En operaciones full duplex, RTS permanece en el estado 0 (OFF).
- 4.- CTS - Clear to Send - Listo para enviar desde el módem al dispositivo digital. Esta señal es una respuesta a RTS e indica que el módem puede aceptar los datos para la transmisión, En la operación full duplex normalmente está siempre activa y presenta un nivel 0 al dispositivo digital.
- 5.- DCD - Data carrier Detected - Se ha detectado el portador de datos del módem al dispositivo digital. Un 1 en este momento indica que el portador no se está recibiendo. Para los sistemas half duplex, ésta sería la señal de encender el portador local y de comenzar la transmisión. En la operación full duplex indicaría una condición anormal.

Los estándares RS-422 y RS-485 usan otros niveles de voltajes que pueden ser voltajes diferenciales.

Desde el punto de vista del usuario, esto no es importante y esto se presenta aquí solamente para una mejor comprensión de los conceptos básicos.

Comunicación con K-Sequence

El protocolo K-Sequence se puede usar para comunicación con *DirectSOFT*, una interface de operador o cualquier otro dispositivo que pueda 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.

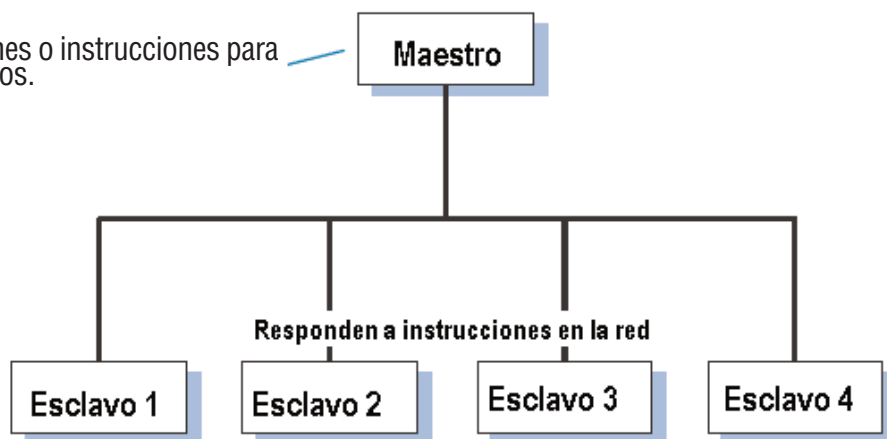
Para utilizar el puerto 2 para comunicaciones K-Sequence usted primero necesita configurar el puerto usando *DirectSOFT* o lógica ladder según lo descrito previamente.

Comunicación con *DirectNET*

K

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.

Ejemplo:
Genera órdenes o instrucciones para transferir datos.



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.

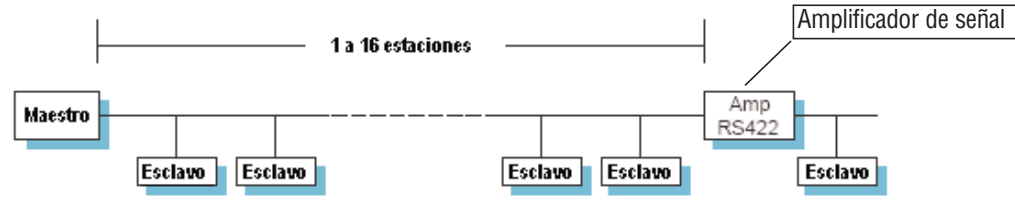
El protocolo *DirectNET* se puede usar para comunicarse a otro PLC o a otros dispositivos que puedan utilizar el protocolo *DirectNET*. El PLC DL06 puede ser usado como maestro en el puerto 2 o como esclavo usando el puerto 1 o el puerto 2.

Muchos paneles de operador 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.

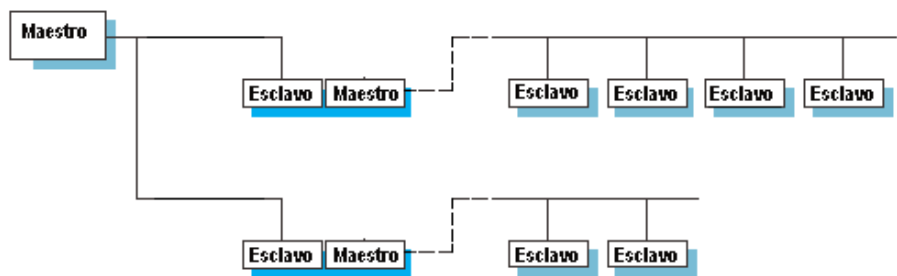
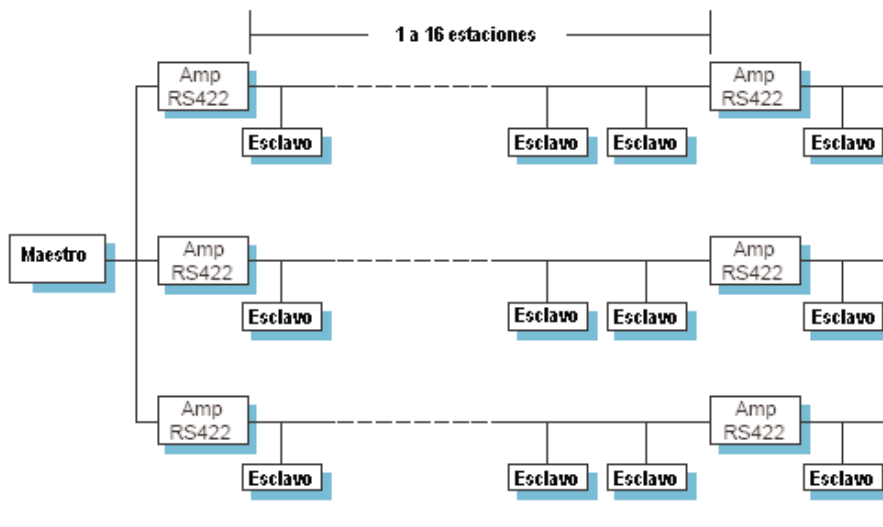
Apéndice K: Introducción a comunicaciones seriales

El esquema siguiente muestra varias combinaciones de maestro y esclavos disponibles con una solución de *DirectNET*.

Conexión esclavo serial



Conexión esclavo paralela



Se puede establecer más de una red, todas asociadas a un PLC. El PLC DL06 puede ser maestro y esclavo a mismo tiempo usando 2 redes, obviamente.

Los tres tipos de redes se pueden combinar para resolver muchas aplicaciones diferentes. Aunque las configuraciones se pueden combinar en una aplicación, cada red debe permanecer independiente. La estación maestra de una red no puede solicitar los datos directamente de estaciones de un esclavo en otra red. Esto no significa que usted no puede obtener los datos de estas redes, usted puede pero se requiere más de un aparato de comunicación para esa estación esclava o puede llevar los datos a un PLC intermediario.

¿Qué programa de comunicaciones se deben usar?

Hay dos opciones de tipos de programas de comunicaciones, las instrucciones RLL o programas de protocolo *DirectNET*. La configuración de red determina el tipo del programa de comunicaciones requerido.

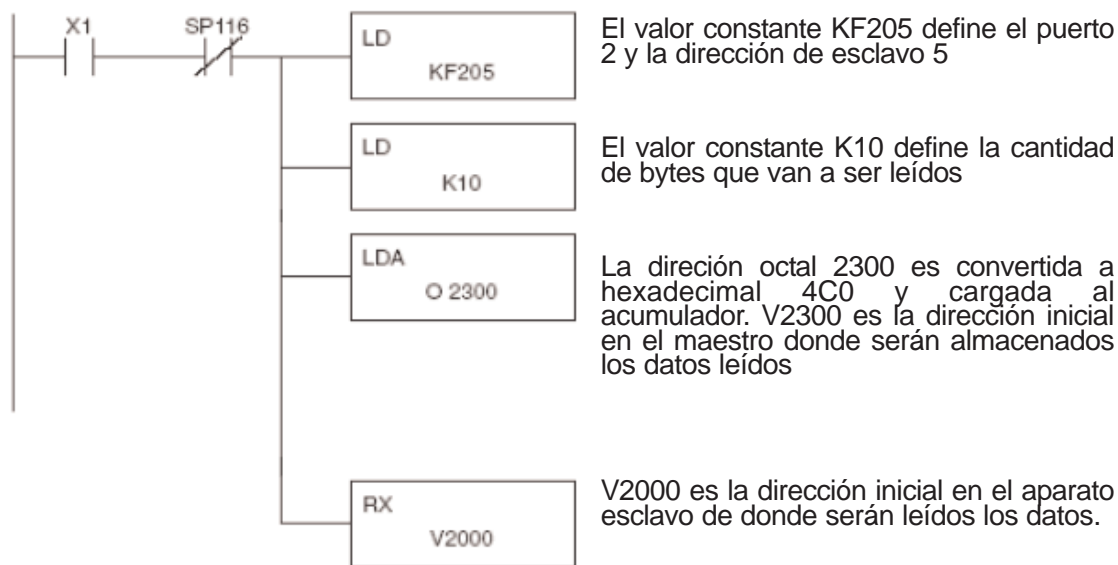
- **PLC como Maestro.** El PLC maestro pide iniciar instrucciones en el programa de RLL para ordenar transferencia de datos.

- **Peer to peer:** Ambos maestros igualmente requieren las instrucciones en el programa de RLL. Ya que ambas estaciones contienen instrucciones de la red, cualquier estación puede iniciar un pedido para datos.

- **Computadoras como Maestro** o un panel de operador debe ejecutar los programas de comunicaciones que pueden publicar a la red las órdenes con el protocolo *DirectNET*. Usted tiene que usar un driver. (ése vino o con su paquete de software de anfitrión o panel de operador), o usted tendrá que crear uno.

Programa

Se usan instrucciones para describir e iniciar la operación que procesará el maestro. Aquí está un ejemplo de un programa sencillo de comunicaciones de RLL. Mas adelante se dan las reglas de uso.



¿Cómo se puede crear una red?

Usted puede crear fácilmente una red *DirectNET* siguiendo cuatro pasos sencillos.

- 1) Diseñe la red de acuerdo a su aplicación
- 2) Seleccione los cables y parámetros de comunicación
- 3) Cree el programa en el aparato maestro
- 4) Haga funcionar la red configurando el puerto y active el programa del PLC para modo RUN.

A continuación se proporcionan explicaciones y ejemplos.

La primera etapa para determinar la configuración es analizar su aplicación. Como todas las cosas, hay normalmente un grupo lógico que le guiará en la propia dirección. Si se tienen varios sistemas, Ud. debe decidir que estaciones serán maestras y cuales serán esclavos.

Determine como planea usar la información que será transferida. Esto puede ayudarlo a diseñar su red. Por ejemplo, está tratando de obtener datos entre varias máquinas o está tratando de obtener datos a ser usados en datos de producción o en hojas de cálculo?

Aunque las tres configuraciones de red mostradas en la página K-8 pueden ser combinadas en una aplicación, cada red debe ser independiente. La estación maestra de una red no puede solicitar datos directamente de estaciones esclavo de otras redes. Esto no significa que no se pueden obtener datos de aquellas redes, sólo se necesita que haya una interfase con aquel esclavo.

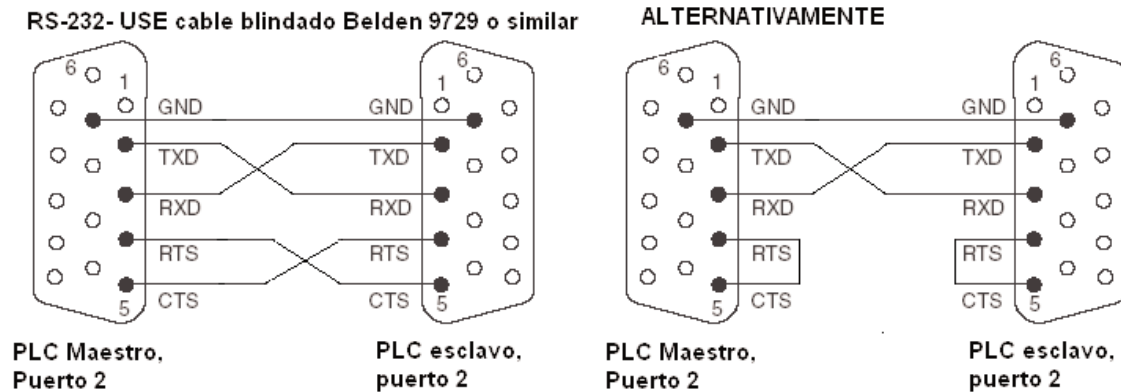
Haga entonces un diagrama de su red o redes. Esto se llama topología de la red.

La próxima etapa es seleccionar los cables. *DirectNET* permite usar los modos RS-232C o RS422. Debe usar RS-422 si usa multinodos, o si la distancia entre aparatos es mayor a 50 pies.

Hay varios tipos de cables que pueden funcionar, pero no todos los cables funcionan. En general, seleccione un cable blindado, con pares trenzados de alambre de sección 24 AWG, una impedancia de 100 Ohm a 1 MHz, y una capacidad de 60 pF por metro o menos.

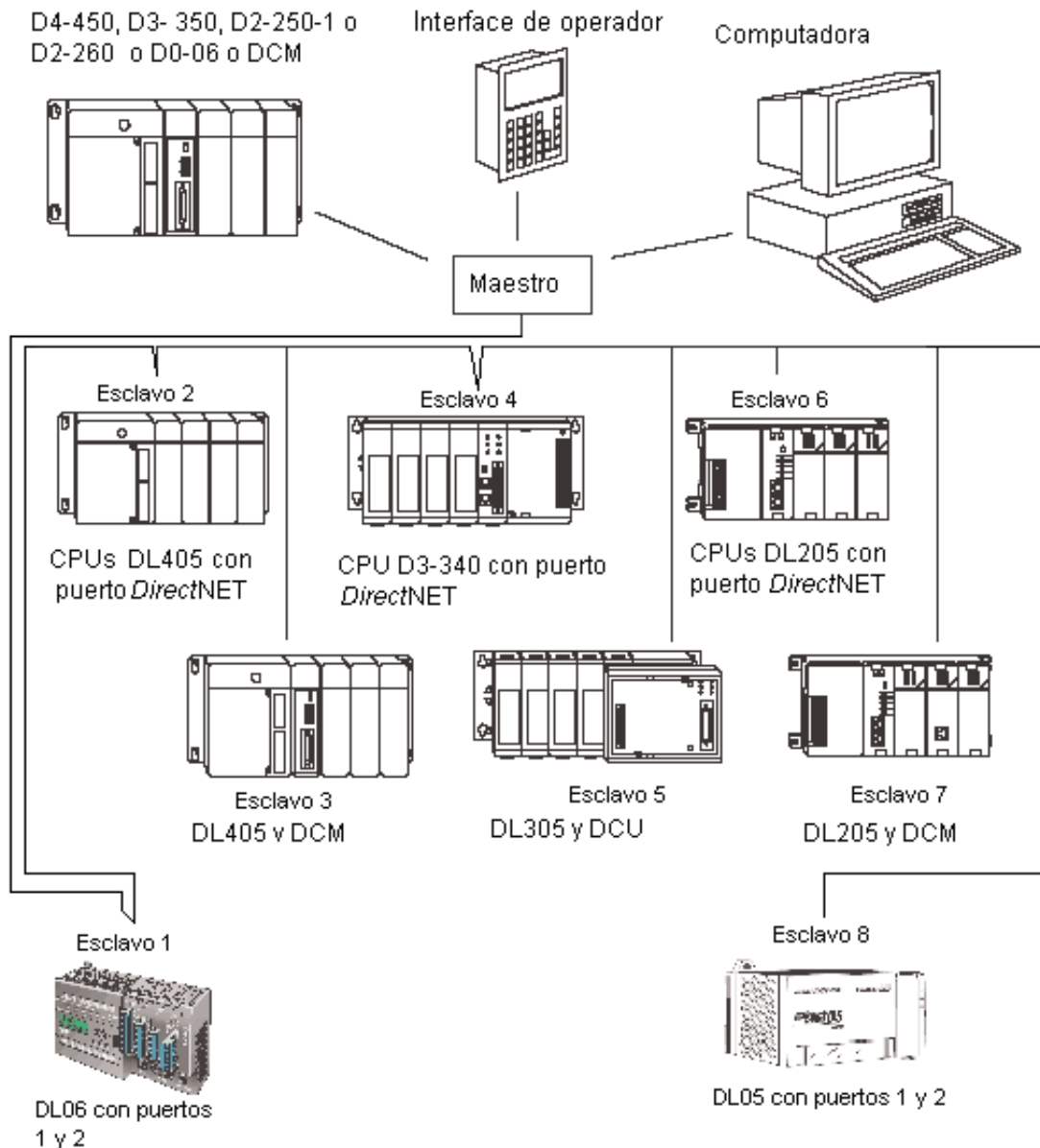
Es importante hacer las terminaciones del cable cuidadosamente, con el conector adecuado, para evitar fallas de comunicación que son difíciles de detectar.

Si la comunicación es punto a punto o peer to peer, la conexión de un cable típico entre puertos 2 es mostrada abajo, entre dos PLCs DL06, con RS-232c.



Si la comunicación es multinodos, se pueden usar hasta 16 nodos sin usar un amplificador de señal RS-422. Los mejores amplificadores son regenerativos y tratan de reducir cualquier señal de ruido que esté presente.

Algunos amplificadores no son regenerativos y amplifican la señal y al mismo tiempo el ruido. Se muestra a continuación un diagrama, como ejemplo.



Es importante que se agreguen resistencias de terminación al final de cada línea en una red RS422. Esto ayuda a reducir errores de transmisión de datos. Deben usarse valores que correspondan a la misma impedancia del cable. Por ejemplo, un cable de 22 AWG con 4,5 trenzas por metro tiene una impedancia típica de 120 Ohm.

Hay dos métodos de conectar las resistencias:

- línea a línea: requiere una resistencia en cada final de línea.
- línea a tierra: Conecte la resistencia entre una línea y tierra. Requiere dos resistencias pero es mejor ya que se mejora el rechazo a ruidos. En este caso, la suma total debe ser un valor que corresponda a la impedancia.

Una vez preparados los cables, se puede hacer el programa.

Operación de un esclavo en la red *DirectNET*

Esta sección describe cómo otros aparatos en una red pueden comunicar con un puerto de CPU que usted ha configurado como un *DirectNET* esclavo (DL06 esclavo).

El maestro de *DirectNET* usa las direcciones normales de entradas y salidas de PLCs *DirectLogic* para conseguir acceso al sistema y a la CPU DL06. No es necesaria ninguna lógica ladder para permitir la operación de esclavo de *DirectNET*, tanto en el puerto 1 como en el puerto 2. El puerto 1 tiene siempre una configuración fija; sin embargo, el puerto 2 puede ser configurado para varias tasas de transmisión, varias direcciones de nodos, etc., que deben equivaler a las del maestro.

Operación de un maestro en la red *DirectNET*

La red *DirectNet* debe tener un maestro y puede tener múltiples esclavos.

Esta sección describe cómo el DL06 se puede comunicar en una red *DirectNET* como un maestro.

El maestro es el único miembro de la red que puede iniciar ordenes en la red. Esta sección le enseña cómo diseñar la lógica ladder requerida para la red con la operación del maestro.

Cuando se usa el PLC DL06 como la estación maestra con *DirectNet* se usan instrucciones sencillas RLL para iniciar los pedidos de transferencia de datos. Solamente el puerto 2 puede ser maestro en el PLC DL06.

Para una operación como maestro de la red, usted necesitará simplemente agregar algunos renglones usando las instrucciones de comunicación de red RX y/o WX.

La instrucción WX inicia las operaciones de escritura en la red y la instrucción RX inicia las operaciones de lectura en la red. Antes de ejecutar las órdenes de WX o RX, es necesario cargar los datos relacionados a la operación de lectura o escritura en el stack del acumulador de la CPU. Cuando se ejecuta la instrucción WX o RX, se usa la información en el stack combinado con datos en el bloque de la instrucción para definir completamente la tarea, que va al puerto.

Para utilizar el puerto 2 con *DirectNET* usted debe primero configurar el puerto usando *DirectSOFT* o lógica ladder según lo descrito más adelante.

Para operación como esclavo de la red, nada debe ser hecho. El puerto 2 funcionará como un esclavo a menos que sean ejecutadas instrucciones de comunicación de red por el programa de lógica ladder en el PLC.

Si usted usa más de una instrucción, solamente es ejecutada una instrucción en un momento dado.

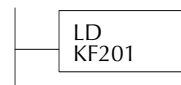
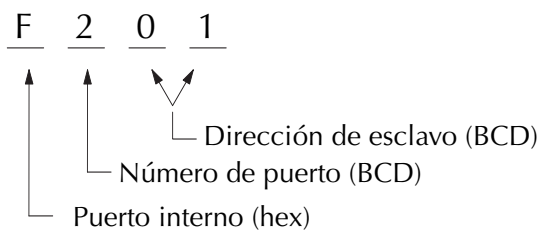
Si usted tiene dos o tres instrucciones de comunicación de red en su programa, usted puede usar bits para enclavarlos, con la ayuda de un bit que muestra el estado del puerto de comunicación libre u ocupado. Veremos más de esto posteriormente.

Si usted está utilizando muchas instrucciones de comunicaciones de red, un contador o un shift register será una manera más conveniente de enclavar las instrucciones.

El procedimiento paso a paso siguiente le proporcionará información necesaria para instalar su programa para recibir datos desde un esclavo de la red.

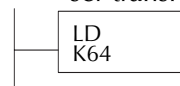
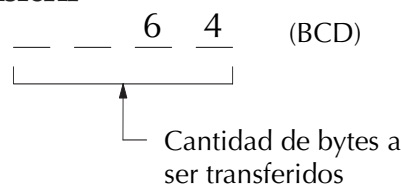
Paso 1: Identifique el puerto del maestro y la dirección del esclavo

La primera instrucción LD identifica el número del puerto de comunicación en el maestro (DL06) de la red y la dirección de la estación auxiliar. Esta instrucción puede direccionar hasta 90 esclavos *DirectNET*. El formato de la palabra se muestra a la derecha. El "F2" en el byte superior indica el uso del puerto derecho del PLC DL06, puerto 2. El byte más bajo contiene el número de la dirección del esclavo en BCD (01 a 99).



Paso 2: Defina la cantidad de Bytes a transferir

La segunda instrucción LD determina la cantidad de bytes que serán transferidos entre el maestro y el esclavo en la instrucción subsecuente WX o RX. El valor estará en formato BCD, a partir 1 hasta 128 bytes.



La cantidad 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 localizaciones de memoria V pueden ser obtenidas en incrementos de 2 bytes.

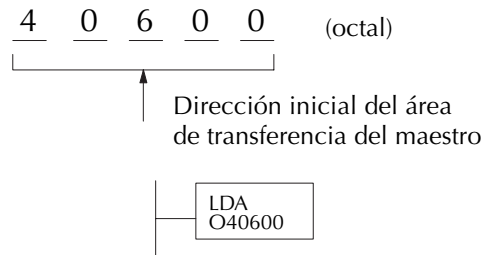
La tabla siguiente muestra los rangos de bytes para los varios tipos de productos de *DirectLOGIC*.

Memoria del DL05 / 06 / 205 / 350 / 405	Bits por unidad	Bytes
Memoria V	16	2
Valor corriente deTemp. / Contador	16	2
Entradas (X, SP)	8	1
salidass (Y, C, Etapas, Bits de Temp./Contadores)	8	1
Memoria de Scratch Pad	8	1
Relevadores especiales	8	1

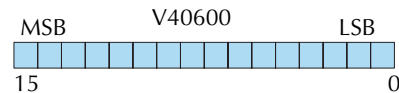


Paso 3: Especifique el área de memoria del maestro

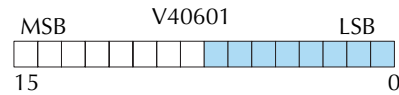
La tercera instrucción en el programa RX o WX es una instrucción LDA. Su propósito es cargar la dirección inicial del área de memoria a ser transferida. Es entrado como un número octal y la instrucción LDA la convierte a hexadecimal y coloca el resultado en el acumulador.



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



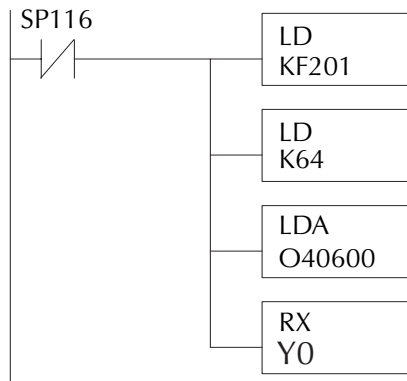
NOTA: Ya que las palabras de memoria V son siempre de 16 bits, usted no siempre puede usar la palabra entera. Por ejemplo, si usted sólo especifica 3 bytes y usted lee las salidas Y del esclavo, usted sólo obtiene 24 bits de datos. En este caso, sólo los 8 bits menos significativos de la última localización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



Paso 4: Especifique el área de 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.



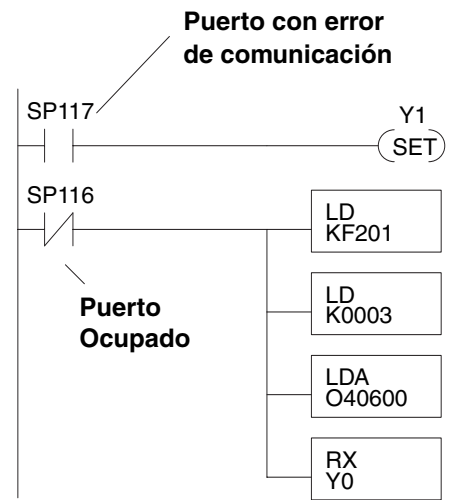
Comunicaciones con un programa ladder

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 él.

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 escritura o lectura a 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.



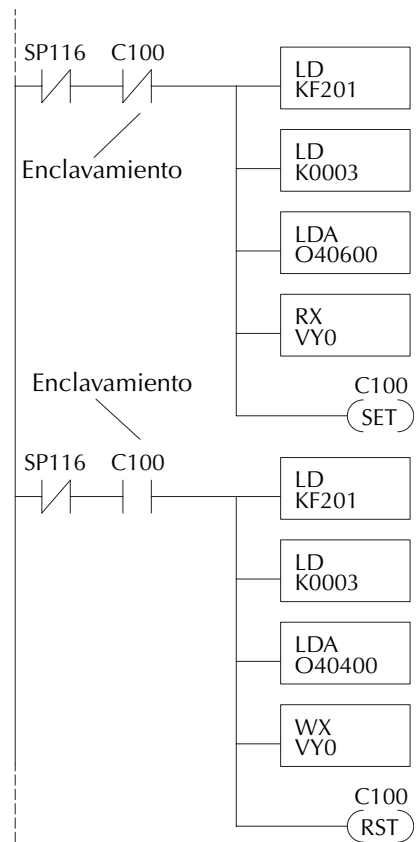
Enclavamientos de lectura y escrituras múltiples

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, 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.



Configuración del puerto como *DirectNET*

Esta sección describe como configurar el puerto del PLC DL06 como maestro para una red *DirectNET*.

Hay dos formas de configurar el puerto 2 para ser maestro. El puerto 1 puede ser solamente esclavo y no necesita configuración.

- a) Uno es usando *DirectSOFT*, el método más simple.
- b) La otra forma es usando lógica ladder; este tiene la ventaja de que ésto queda grabado en la memoria Flash del PLC, junto con el código. Al usar *DirectSOFT*, los parámetros se quedan en memoria RAM pueden ser borrados, por ejemplo, haciendo "Initialize scratchpad" o si el PLC queda sin energía por más de 4 días y sin batería.

Estos son los datos principales de los 2 puertos:

Especificaciones del puerto 2 del PLC DL06

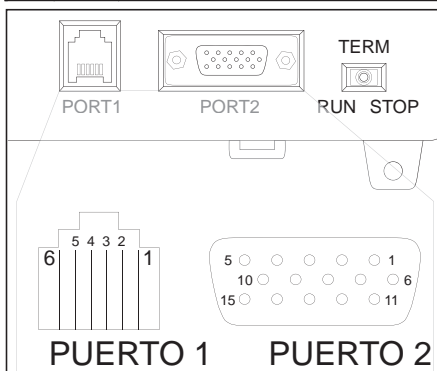
Puerto 1	
Puerto 1	Conecta a HPP, <i>DirectSOFT</i> 5, interfaces de operador , etc.
	6 clavijas, RS232C
	Tasa de comunicación (baud): 9600 (fija)
	Paridad: odd (fija)
	Dirección del nodo: 1 (fija)
	8 data bits
	1 start, 1 stop bit
	Asíncrono, half-duplex, DTE
	Protocolo (auto-select): K-sequence , <i>DirectNET</i> MODBUS todos esclavo solamente

Puerto 2	
Puerto 2	Conecta a HPP, <i>DirectSOFT</i> , o interfaces de operador , etc.
	Puerto multifunción, RS232C, RS422, RS485
	Tasa de comunicación (baud): 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
	Paridad: odd (por defecto), even, none
	Dirección del nodo: 1 (por defecto)
	8 data bits
	1 start, 1 stop bit
	Asíncrono, half-duplex, DTE
	Protocolo (auto-select): K-sequence (esclavo solamente), <i>DirectNET</i> (maestro o esclavo), MODBUS (maestro o esclavo), non-sequence/print/ASCII in/out

Clavijas del puerto 2

Descripciones del Puerto 1		
1	0V	0 Volt (-) (GND)
2	5V	Fuente de 5 Volt (+)
3	RXD	Receive data (RS-232C)
4	TXD	Transmit data (RS-232C)
5	5V	Fuente de 5 Volt (+)
6	0V	0 Volt (-) (GND)

Descripciones del Puerto 2		
1	5V	Fuente de 5 Volt (+)
2	TXD	Transmit data (RS-232C)
3	RXD	Receive data (RS-232C)
4	RTS	Ready to send (RS-232C)
5	CTS	Clear to send (RS232C)
6	RXD-	Receive data (-) (RS-422/485)
7	0V	0 Volt (-) (GND)
8	0V	0 Volt (-) (GND)
9	TXD+	Transmit data (+) (RS-422/485)
10	TXD-	Transmit data (-) (RS-422/485)
11	RTS+	Ready to send (+) (RS-422/485)
12	RTS-	Ready to send (-) (RS-422/485)
13	RXD+	Receive data (+) (RS-422/485)
14	CTS+	Clear to send (+) (RS-422/485)
15	CTS-	Clear to send (-) (RS-422/485)



Configuración del puerto 2 como *DirectNET* con *DirectSOFT*

En *DirectSOFT*, escoja el menú PLC, luego Setup, luego "Setup Second. Comm Port". Siga las instrucciones que aparecen en la página 4-11 de este manual.

Configuración del puerto 2 como *DirectNET* con lógica Ladder

El puerto 2 en el DL06 se puede también configurar para usar el protocolo *DirectNET* usando lógica ladder dentro del programa del PLC. También, los parámetros de las comunicaciones se pueden configurar como los parámetros en el otro aparato con el cual el PLC se comunicará.

Note que los parametros del puerto 2 nunca se almacenan al disco con *DirectSOFT* de modo que si usted está usando el puerto 2, con excepción de la configuración por defecto, es una buena idea incluir la configuración en el programa ladder.

Para configurar el puerto 2 en lógica ladder se deben escribir valores apropiados a V7655 (palabra 1) y a V7656 (palabra 2) para especificar la configuración del puerto. Luego escriba K0500 a V7657 (palabra 3) para pedir a la CPU que acepte los valores.

Una vez que la CPU vea K0500 en V7657, verificará los parámetros de comunicación que se han seleccionado y después cambiará el valor en V7657 según los resultados de esta prueba.

Si éstos son válidos, la CPU cambiará el valor en V7657 a 0A00 ('A' para aceptado). Si había un error en los valores, la CPU cambiará el valor en V7657 a 0E00 ('E' por error).



NOTA: Sugerencia. En vez de construir las palabras de configuración manualmente desde las tablas, use *DirectSOFT* para configurar el puerto como desee y use *Dataview* para ver las palabras en V7655 y V7656 como BCD/HEX. Luego use estos números en el código de configuración.

Los datos que se escriben a las palabras de configuración tienen dos formatos. El formato que se usa que depende si es K-Sequence, *DirectNET*, MODBUS RTU (método 1) o ASCII (método 2).

Observe que es la configuración por defecto del puerto 2 es:

Detección automática entre protocolos K-Sequence, *DirectNET*, and MODBUS RTU

Timeout = Base Timeout x 1 (800 ms)

RTS on delay time = 0 ms

RTS off delay time = 0ms

Número del nodo= 1

Baud rate = 19200

Bits de Stop = 1

Parity = impar

Formato = Hexadecimal

Supresión de eco = RS-422/485 (4-wire) o RS-232C

Las palabras mencionadas V7655 y V7656 se configuran según las tablas en las páginas siguientes.

Configuración del Puerto 2 usando *DirectNET*

La palabra V7655 está formada de los siguientes grupos de bits:

0yyy 0TTT mmmm mxxx

Siendo yyy = RTS ON-delay

TTT= Time out de comunicación

mmmm= tipo de protocolo

xxx = TRS off-delay

V7655 (Palabra 1)	RTS On-delay	Timeout (% de timeout)	Protocolo	RTS Off-delay
0yyy 0ttt mmmm mxx	yyy	TTT	mmmmm	xxx
	000 = 0 ms	000 = 100%	10000 = K-Sequence	000 = 0ms
	001 = 2 ms	001 = 120%	01000 = <i>DirectNET</i>	001 = 2ms
	010 = 5 ms	010 = 150%	00100 = MODBUS RTU	010 = 5ms
	011 = 10 ms	011 = 200%		011 = 10ms
	100 = 20 ms	100 = 500%		100 = 20ms
	101 = 50 ms	101 = 1000%		101 = 50ms
	110 = 100 ms	110 = 2000%		110 = 100ms
	111 = 500 ms	111 = 5000%		111 = 500ms

La palabra V7656 para *DirectNET* está formada de los siguientes grupos de bits:

pps0 ebbb xaaa aaaa

Siendo pp = Paridad

s = Bits de Stop


e = Supresión de eco

bbb = Tasa de Baud

x = Modo del protocolo

aaaaaa = Dirección del esclavo

V7656 (Palabra 2)	Paridad	Bits de Stop	Supresión de eco	Tasa de Baud
<i>DirectNET</i>				
pps0 ebbb xaaa aaaa	pp	s	e	bbb
	00 = Ninguna	0 = 1 bit	0 = 232 or 422	000 = 300
	10 = Odd	1 = 2 bits	1 = 485, 2 wire	001 = 600
	11 = Even			010 = 1200
				011 = 2400
				100 = 4800
				101 = 9600
				110 = 19200
				111 = 38400

V7656 (Palabra 2) continuada	Protocolo	Dirección del nodo
<i>DirectNET</i>	(<i>DirectNET</i>)	xaaaaaaaa (<i>Modbus RTU DirectNET</i>)
pps0 ebbb xaaa aaaa	x	aaaaaaaa (K-Seq. & MODBUS RTU)
	0 = Hex	K-Sequence: 1-90
	1 = ASCII	1-90
		MODBUS: 1-247
		 Use el número como hexadecimal

K

La palabra V7657 debe ser cargada con el valor K0500 con ladder, para que los valores en las otras 2 palabras sean verificados y cargados

Ejemplo: Para configurar el puerto 2 para protocolo *DirectNET* con lo siguiente:

RTS On-delay of 10ms,

Base timeout x1,

RTS Off-delay of 5ms,

Paridad Odd,

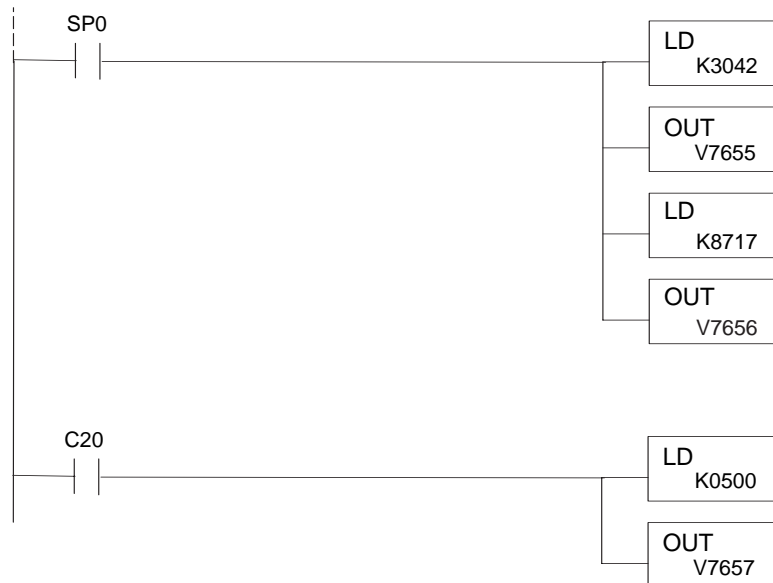
1 Stop bit,

Supresión de eco para RS232-C/RS422, 1

384000 baud,

Número de nodo 23 (hexadecimal 17)

Se usaría la lógica ladder mostrada en la figura siguiente (C20 se ejecuta después de SP0) :



Explicación de 0yyy 0TTT mmmm mxxx

0yyy - 0011 - RTS On-delay de 10ms,

0TTT - 0000 - Base timeout x1,

mmmm - 01000 -

xxx - 010 - RTS Off-delay de 5ms,

Ésto es, 0001 0000 0100 0010 equivalente a 3 0 4 2

Explicación de pps0 ebbb xaaa aaaa

pp - 10 Paridad Odd,

s0 - 00 1 Stop bit,

e - 0 Supresión de eco para RS232-C/RS422, 1

bbb - 111 38,400 baud,

x - 0 Modo de protocolo

aaa aaaa - 001 0111 Número de nodo 17 hexadecimal, que corresponde a 23 decimal

Ésto es, 1000 0111 0001 0111 equivalente a 8 7 1 7

Una forma de hacer esta verificación de los números a cargar en las palabras 1 y 2 es usar *DirectSOFT* y *Data View*, como sigue:

The image shows two windows from the DirectSOFT software. On the left is the 'Data1' window, which displays a table with two columns: 'Element' and 'Status'. The first two rows are highlighted. The first row shows 'V7655' in the Element column and '3042' in the Status column. The second row shows 'V7655' in the Element column and '0011000001000010' in the Status column. A callout box labeled 'Formato BCD/HEX' points to the '3042' value, and another callout box labeled 'Formato binario' points to the binary string '0011000001000010'. On the right is the 'Setup Communication Ports' dialog box. It is configured for 'Port 2'. The 'Protocol' section has 'DirectNET' selected. The 'Time-out' is set to 'Base Timeout x 1'. The 'RTS on delay time' is 10 ms, and the 'RTS off delay time' is 5 ms. The 'Station Number' is 23. The 'Baud rate' is 38400, 'Stop bits' is 1, and 'Parity' is Odd. The 'Format' is set to Hex. The 'Echo Suppression' section has 'RS-422/485 (4-wire)' selected. The status bar at the bottom of the dialog box indicates 'Port 2: 15 Pin'.

Element	Status
V7655	3042
V7655	0011000001000010
V7656	8717
V7656	1000011100010111

Ejemplo para Redes *DirectNET*

1. Conexión de un DL 06 como maestro a 2 PLCs DL05 como esclavos que están separados por 200 metros cada uno, a una velocidad de transmisión de 38400 kbps

El PLC maestro tiene un programa simple que hará que:

Las salidas Y0 a Y5 del esclavo 2 se cierren a una frecuencia de 1 Hz siendo 500 ms cerrados y 500 ms abiertos cuando las correspondientes entradas X0 a X5 en el esclavo 2 se cierren.

Al mismo tiempo las entradas X0 a X5 del maestro estén cerradas, y las salidas Y0 a Y5 del esclavo 3 se cierren a una frecuencia de 2 Hz siendo 1 segundo cerrado y 1 segundo abierto cuando las correspondientes entradas X0 a X5 en el esclavo 3 se cierren, y

Al mismo tiempo las entradas X10 a X15 del maestro estén cerradas para el esclavo 3 y confirme al maestro que al menos una de las salidas se ha cerrado en los bits C100 a C107

Este programa no tiene ninguna aplicación práctica, pero es un ejercicio para definir las características de una conexión en red.

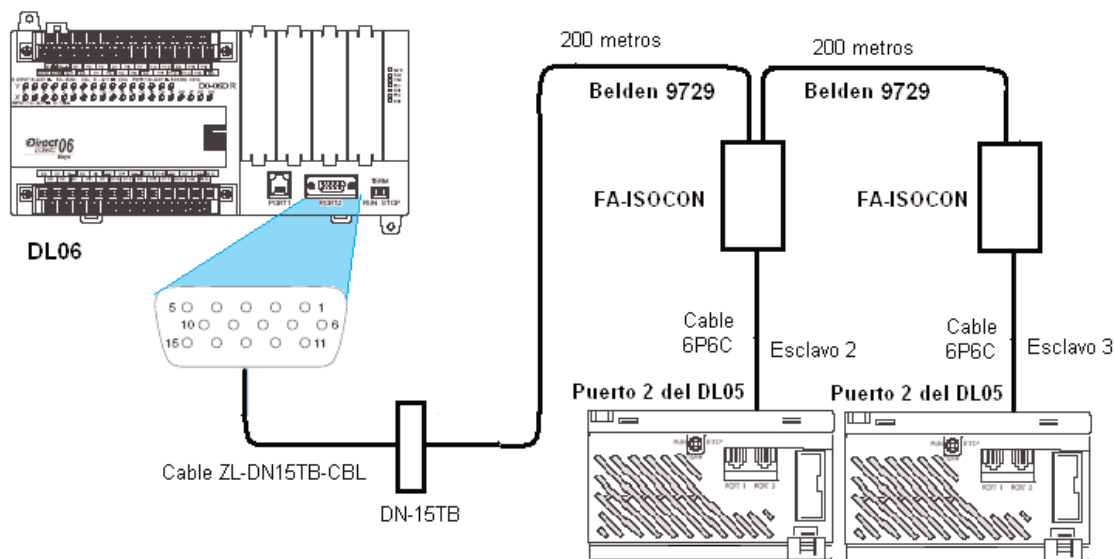
Como ya fue dicho, es necesario:

- a. Diseñar la red de acuerdo a su aplicación
- b. Seleccionar los cables y parámetros de comunicación
- c. Crear el programa en el aparato maestro y los esclavos si fuera necesario.

a. Diseñar la red de acuerdo a la aplicación.

Usaremos el sistema RS-422. Para eso usaremos el puerto 2 del PLC DL06, que ya tiene este sistema incorporado. Los PLCs DL05 solo tienen un sistema RS-232 y para adaptar esto usaremos el módulo FA-ISOCON.

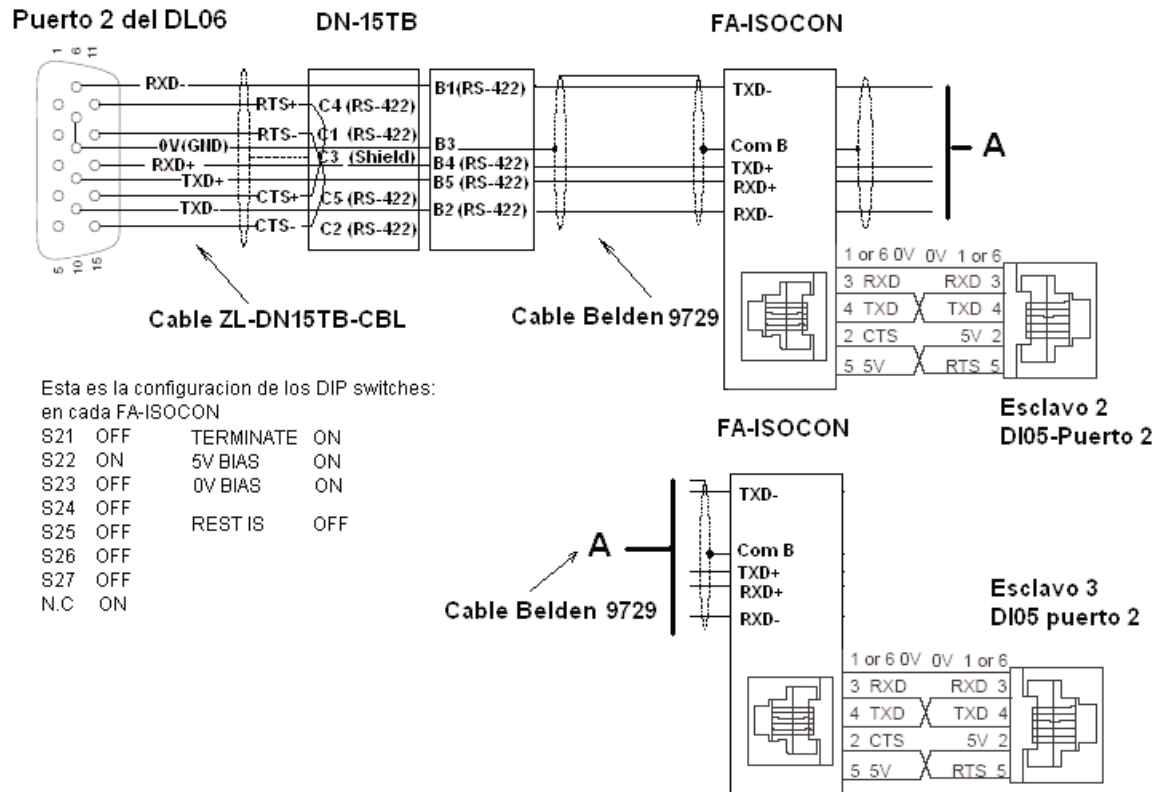
Queremos establecer una red con distancia entre nodos de 200 m a una velocidad de 38400 kbps. Este es el diagrama planeado:



b. Seleccionar los cables y parámetros de comunicación:

Los cables se muestran en el diagrama de arriba y las conexiones están mostradas en el siguiente diagrama. Note que esta es una posibilidad de solución. Es posible que Ud prepare un cable conectado directamente entre el puerto 2 del DL06 y el módulo FA-ISOCON del esclavo 2.

El módulo FA-ISOCON ya trae incorporado un cable de un pie de longitud para conectar este módulo con el puerto 2 del PLC DL05.



Los parámetros de comunicación pueden ser configurados de la forma normal de hacer la configuración de los puertos con *DirectSOFT*; pero más adelante, en la sección del programa de los PLCs, haremos una configuración por diagrama ladder. El PLC maestro tiene la dirección 1 y los esclavos seán 2 y 3, aunque se puede seleccionar de 2 a 90. C1.

Creación del programa en el aparato maestro

Se muestra a continuación la lógica ladder que ejecuta esta función.

El renglón (rung) 1 lleva a la subrutina 1 que muestra la forma alternativa de configurar el puerto 2 maestro para este caso. Note que usamos 38400 baud.

Se usa un contador para hacer que solamente sea hecha una transacción de transferencia de datos en cada instante.

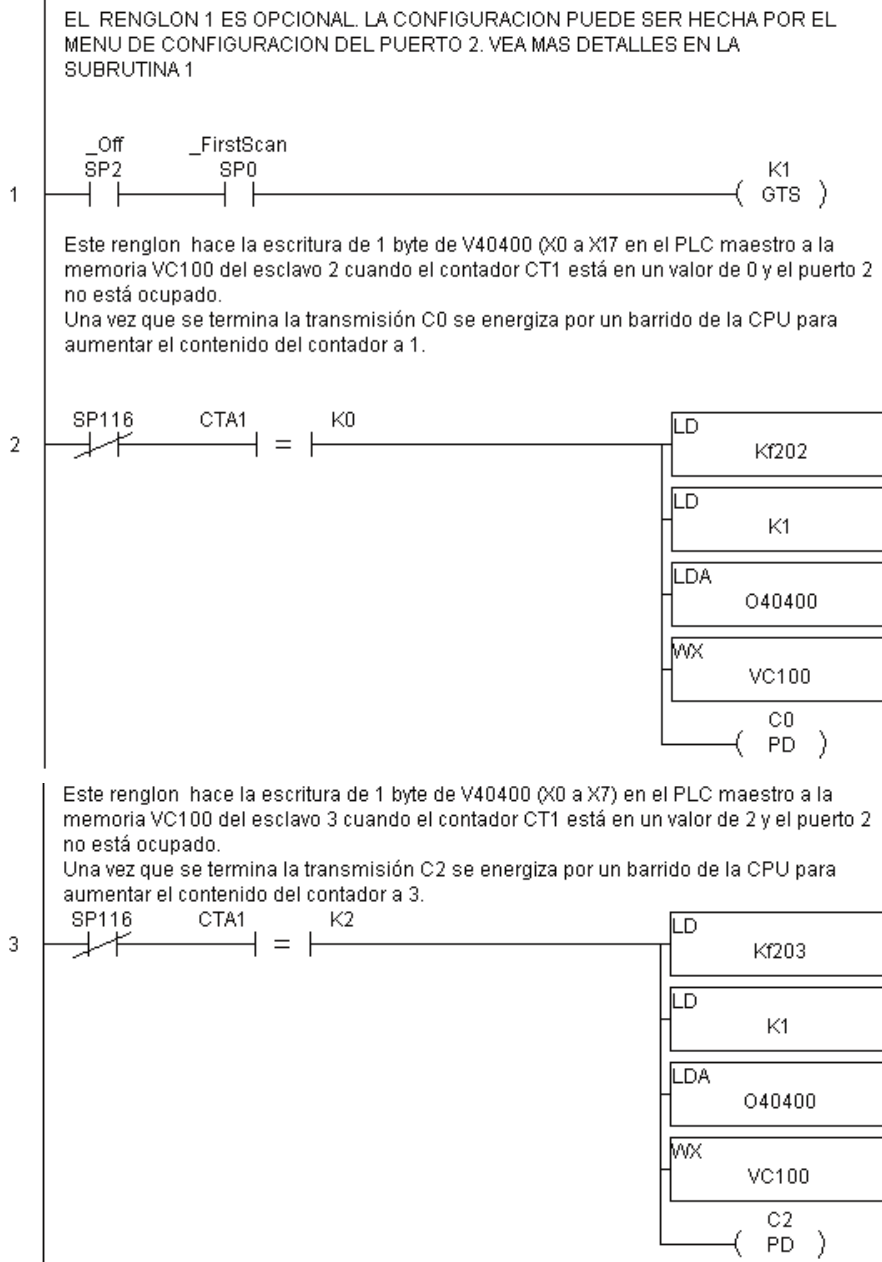
Se usan 4 renglones (rungs) para definir cada escritura y lectura.

Adicionalmente se ha colocado un contador para medir la eficiencia de transmisión. Para eso se cuentan las veces que se han terminado las cuatro transacciones de comunicación.

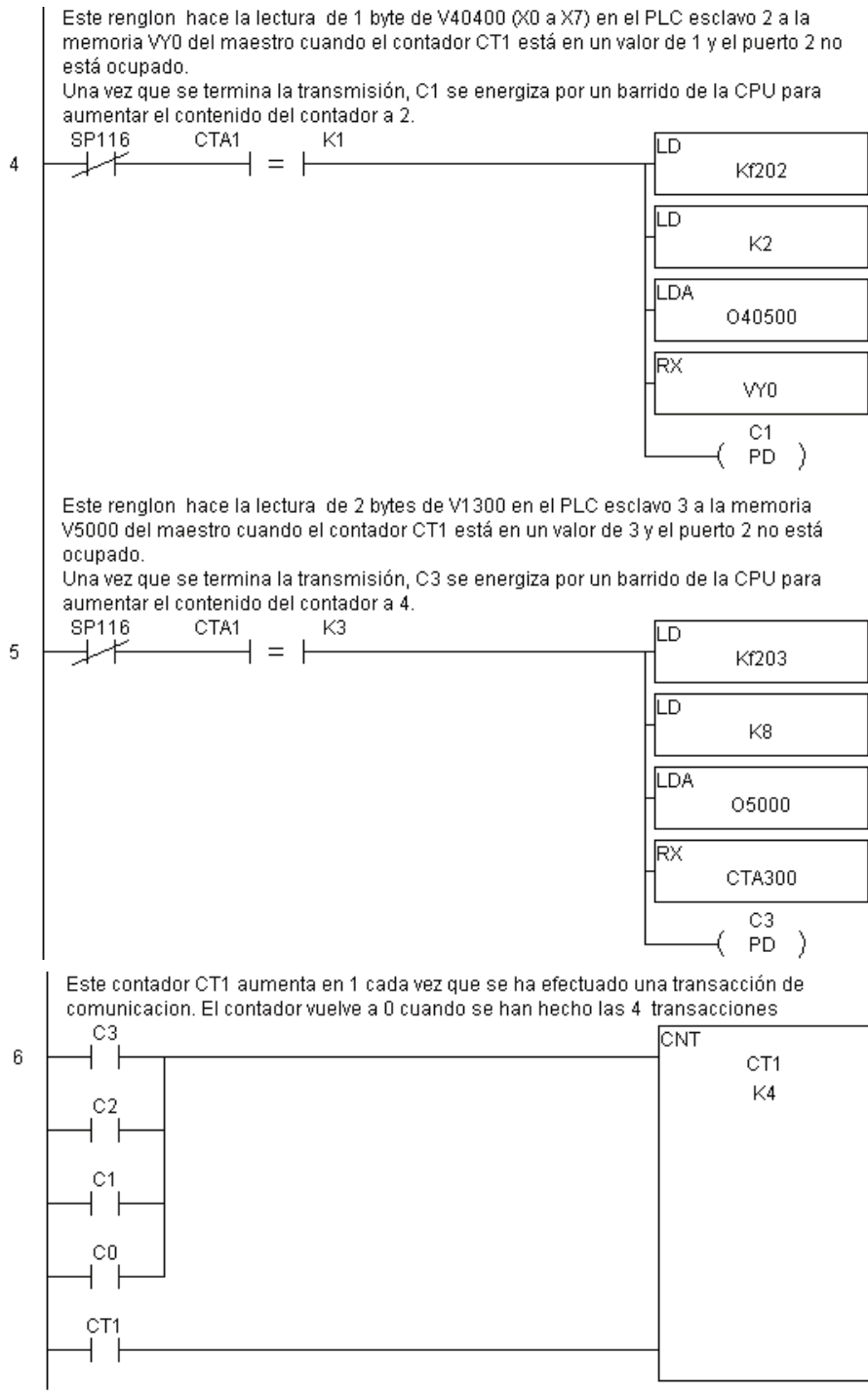
En este caso se llega a 386 transacciones por minuto, es decir, hay 6,43 actualizaciones por segundo y desde allí se puede determinar que se actualizan los datos cada 155 ms.

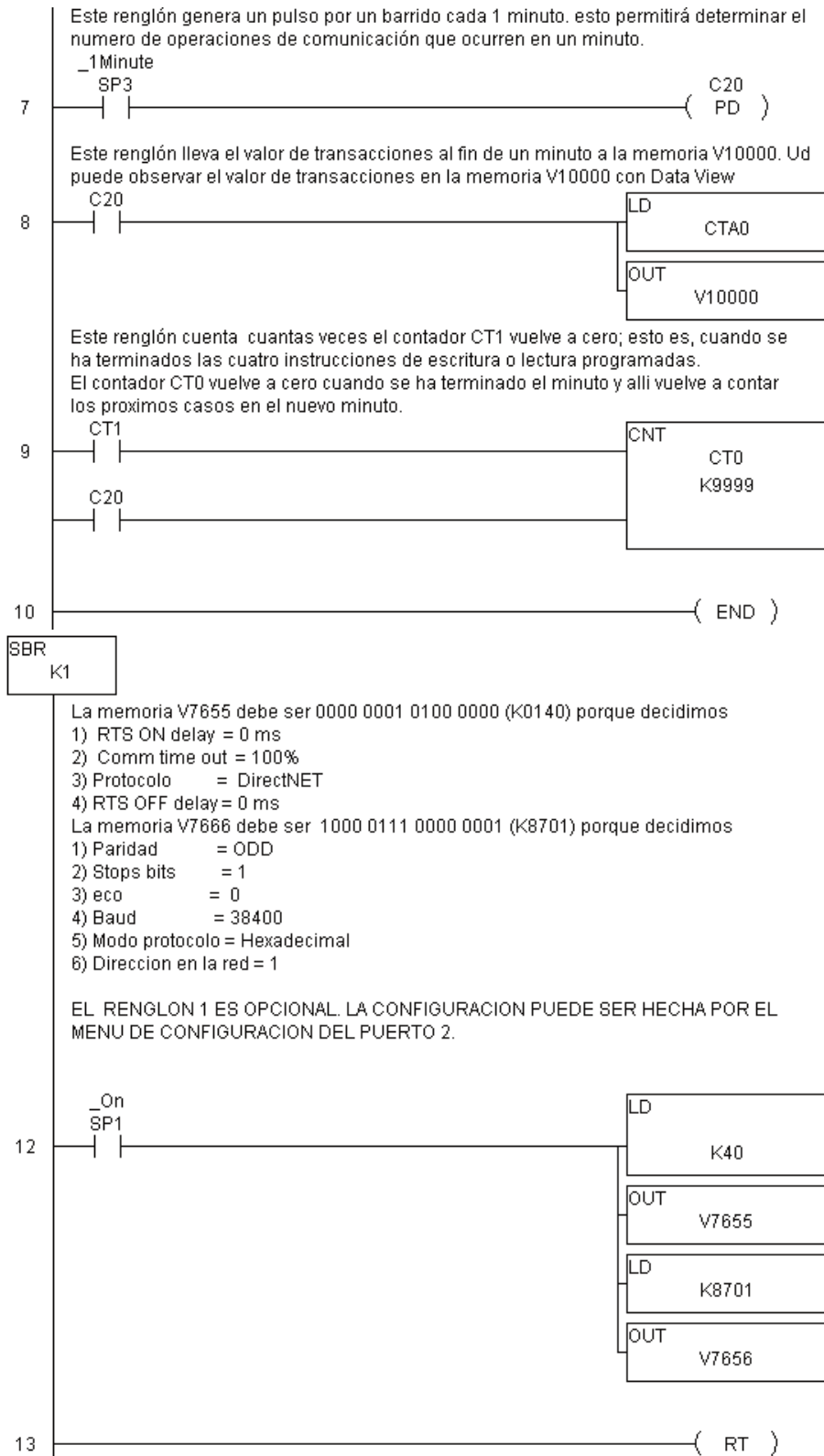
El tiempo de barrido del PLC maestro es de 3 ms. Ud ve entonces que la transmisión se realiza en más tiempo que un barrido

Observe los comentarios en el diagrama a continuación:



K

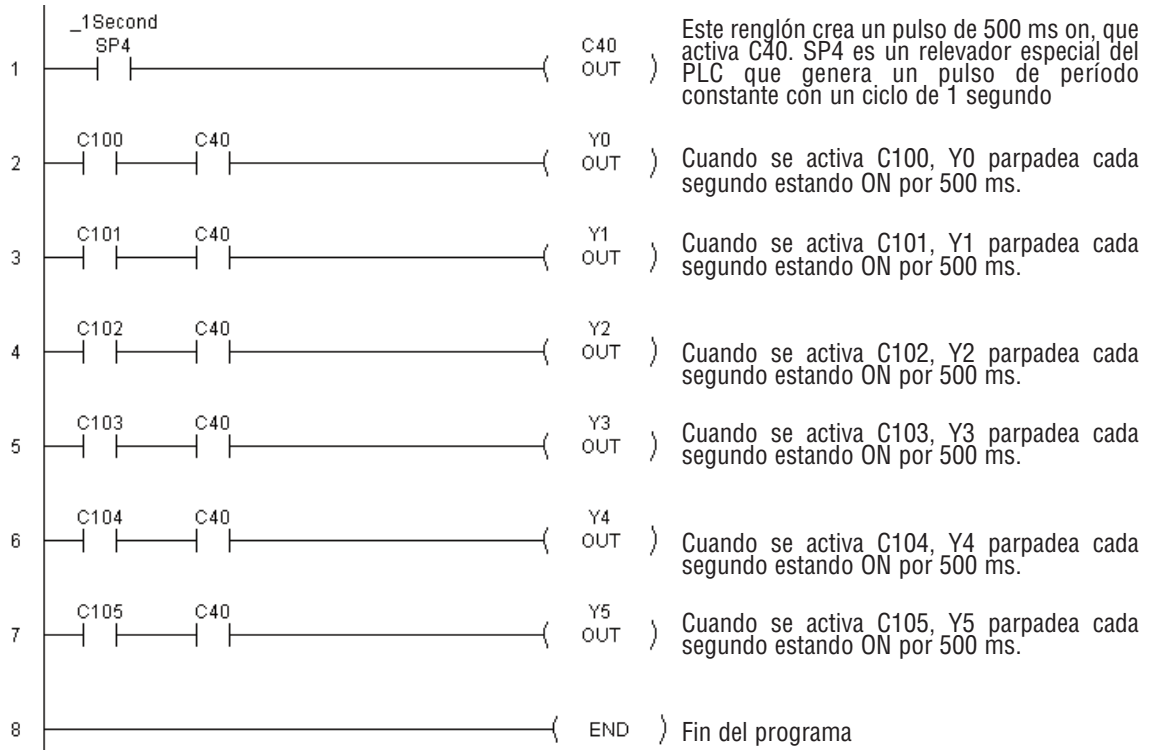




K

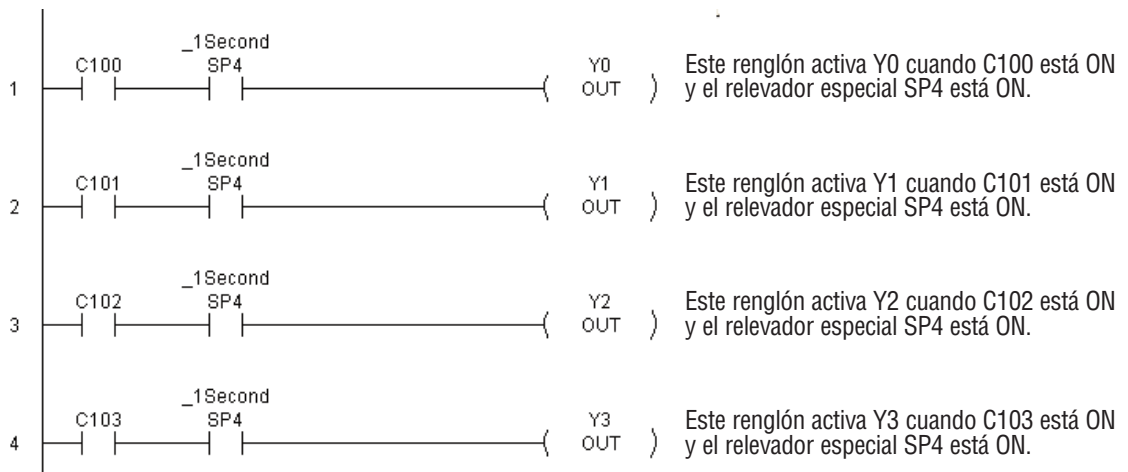
C2. Creación del programa en el esclavo 2.

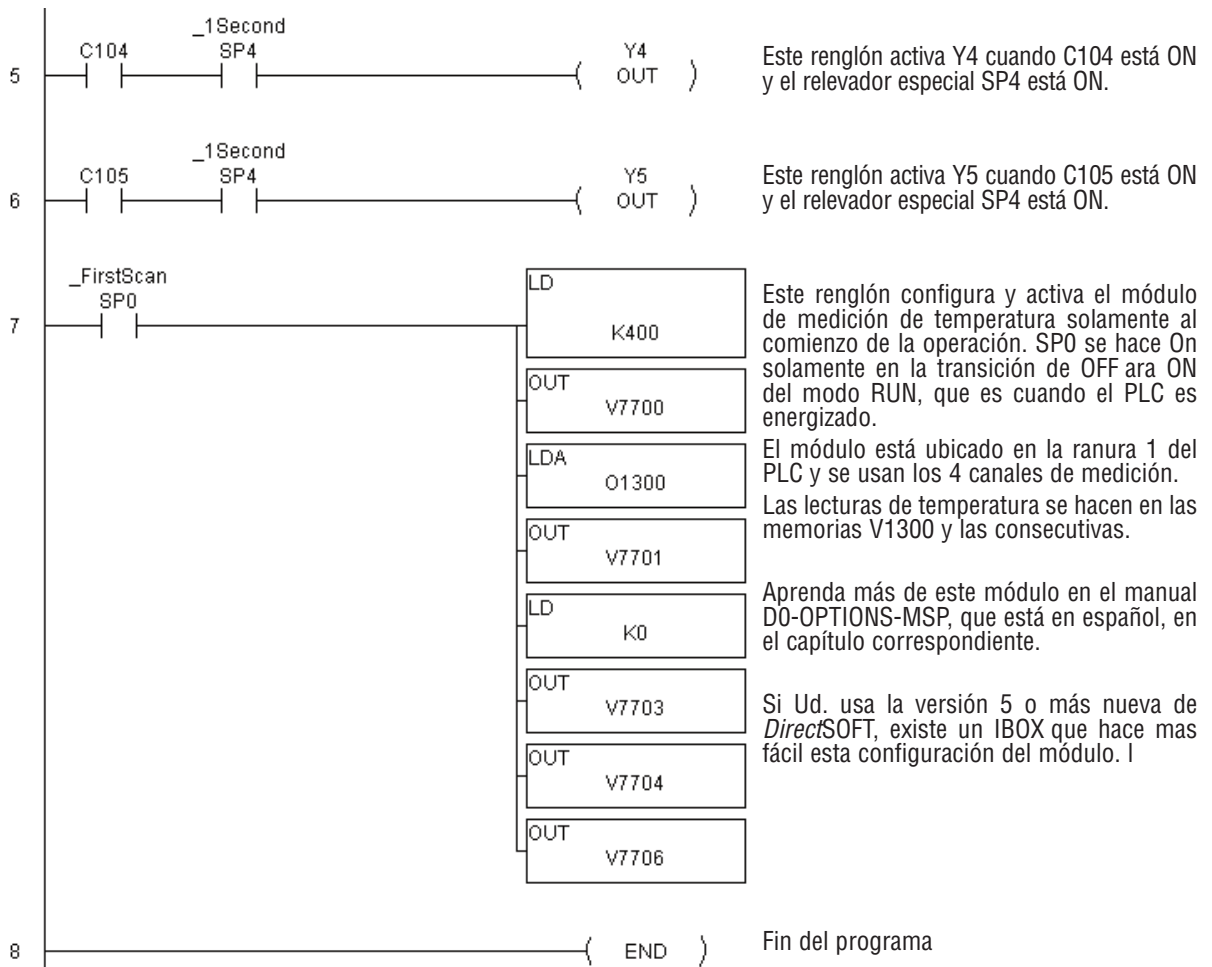
Note que la generación del pulso de 500 ms se hace en el propio PLC esclavo. La lectura de los bits Y0 a Y5 se hace en el maestro con la palabra V40500 y no es necesaria ninguna lógica de comunicación en el esclavo 2.



C3. Creación de la lógica en el esclavo 3.

Este PLC tiene un módulo de medición de temperaturas por RTDs. Por eso se ha agregado un renglón de configuración de la medición. Las lecturas se hacen en las memorias V1300 a V1303 y la lectura del primer canal se coloca en V5000 en el PLC maestro.





d) Haga funcionar la red configurando el puerto y active el programa para modo RUN

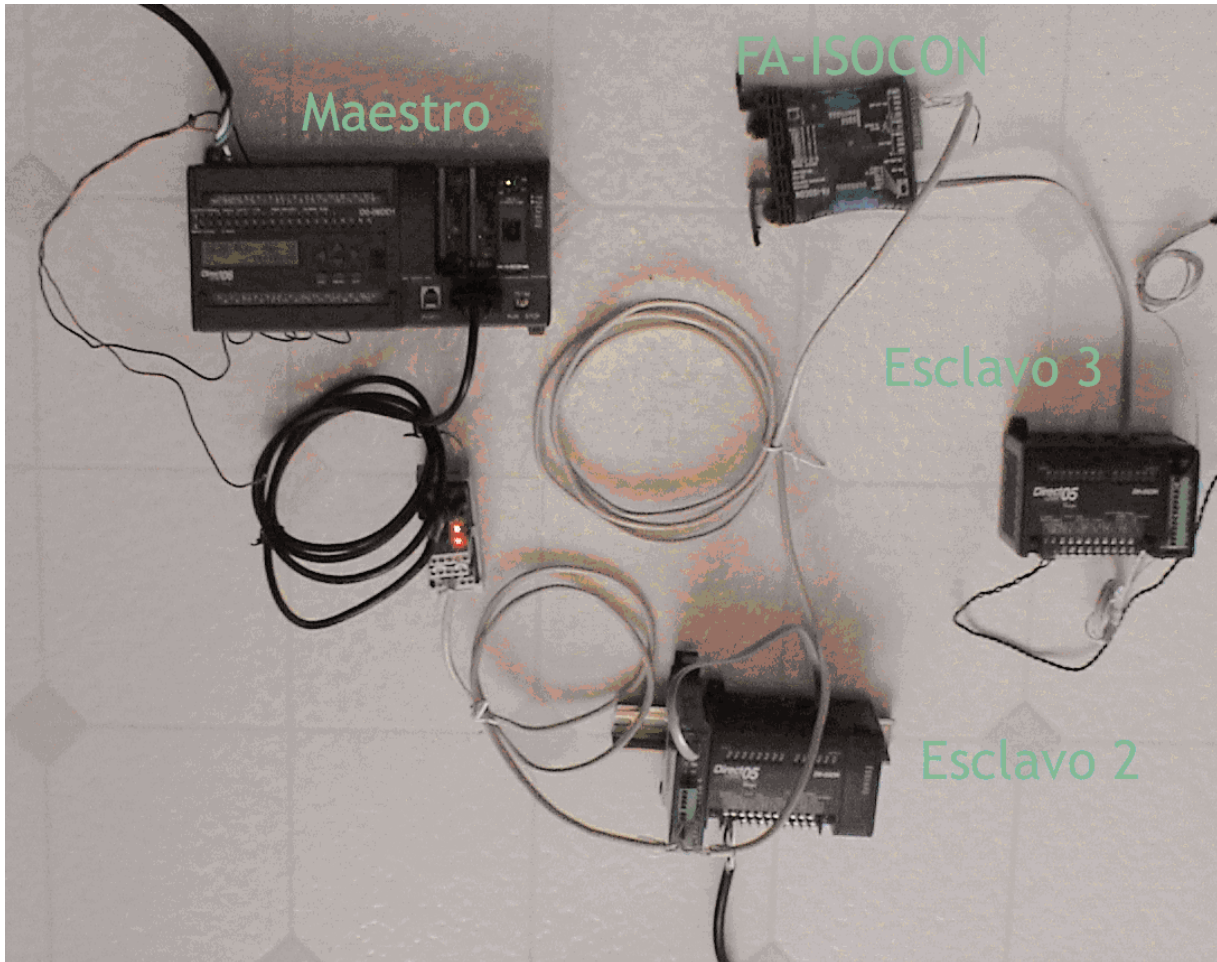
Despues de programar los PLCs, deben configurarse los puertos 2 del PLC DL06 así como también los puertos 2 de los PLCs DL05 a 19200 kbps y también los parámetros adicionales.

Los DIP switches de los adaptadores FA-ISOCAN deben ser colocados de acuerdo a lo mostrado en la página K-22.

Se establecerá la comunicación al colocar los PLC en modo RUN, dejando los switches de modo en TERM,

Ud. podrá ver que los LEDs TX2 y RX2 del PLC DL06 parpadean (localizados en la esquina derecha del PLC) , lo que muestra que las señales de transmisión están actuando. Estos LEDs se pueden ver también en los adaptadores FA-ISOCAN. Los LEDs de transmisión son útiles en la búsqueda de problemas de comunicación.

Vea en la foto de la página siguiente una fotografia con un sistema de prueba para este tipo de comunicación.

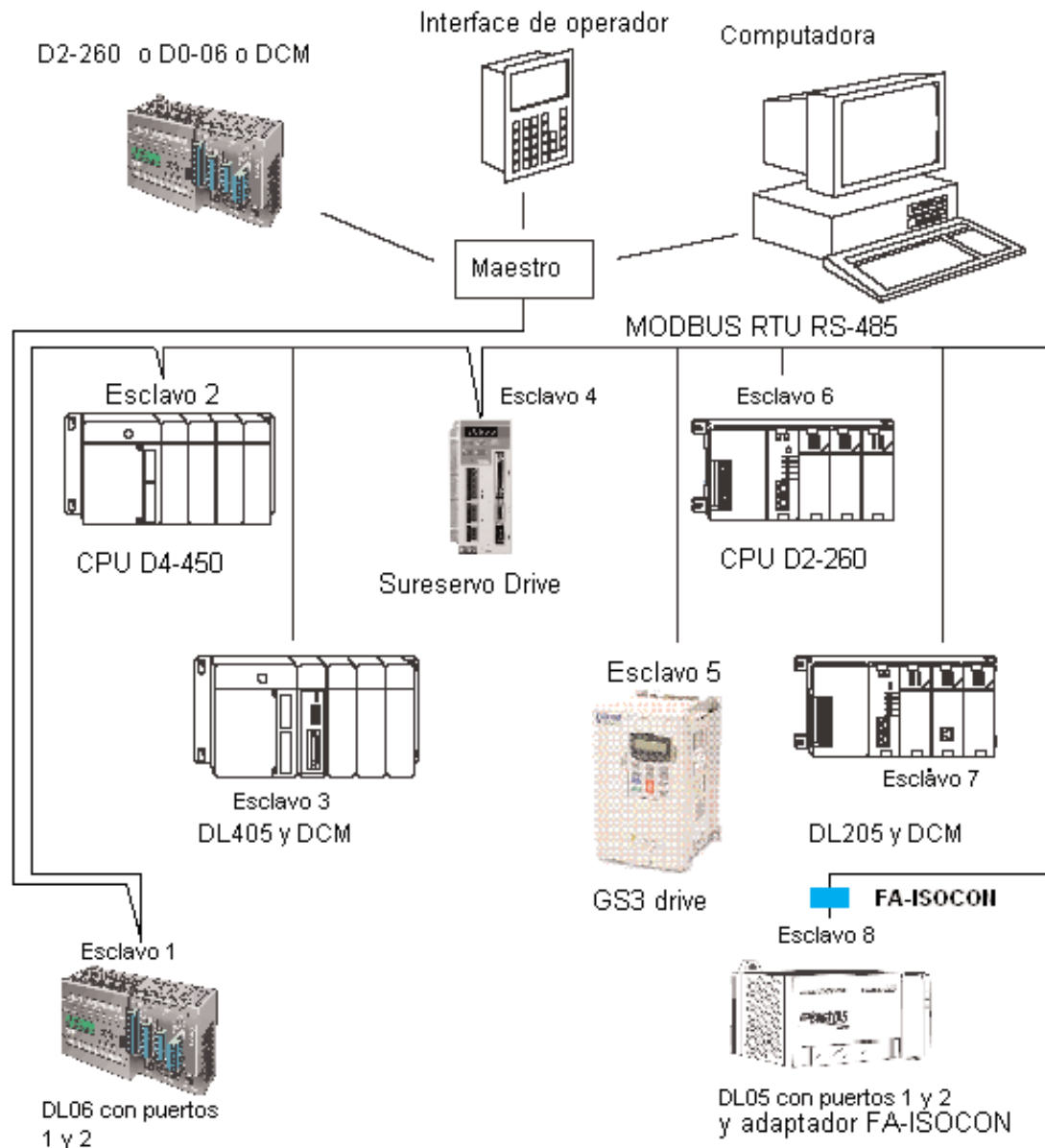


K

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. Se puede utilizar el estándar RS-485 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 un ejemplo a continuación:



Introducción al protocolo MODBUS RTU

El protocolo MODBUS RTU (Remote Terminal Unit) es una estructura de mensaje usada para establecer comunicaciones maestras a esclavos entre aparatos inteligentes. Cuando un maestro MODBUS envía un mensaje a un esclavo MODBUS, el mensaje contiene la dirección del esclavo, una función, los datos y una cifra de verificación de que el mensaje fue correctamente transmitido llamado checksum. La respuesta del esclavo contiene campos confirmando la requisición del maestro, cualquier dato pedido y el campo de verificación de error.

Una estructura de mensaje consiste de los siguientes campos:

Dirección del esclavo **Función** **Datos** **Checksum**

El campo de **dirección** de un mensaje contiene ocho bits. Las direcciones válidas de esclavos están en el rango 0-247 decimal (la dirección 0 es la transmisión “broadcast”, es decir, una transmisión a todos los esclavos) tanto los aparatos individuales esclavos se colocan en el rango de 1-247 decimal. El maestro especifica un esclavo colocando la dirección del esclavo en el campo de dirección del Mensaje. Cuando el esclavo responde, coloca su propia dirección en el campo de dirección para identificar al maestro qué esclavo está respondiendo.

El campo de **función** es un mensaje que contiene ocho bits. Los códigos de función válidos están en el rango de 1-255 decimal. El código de función instruye al esclavo hacer una clase de acción. Algunos ejemplos son leer el estado de un grupo de entradas discretas; leer los datos en un grupo de memorias; escribir el estado a una bobina de salida o a un grupo de memorias o leer el estado de diagnóstico de un esclavo. En la nomenclatura de MODBUS a las memorias se le dicen registros.

Cuando un esclavo responde al maestro, usa el campo de código de función para indicar que tiene una respuesta normal o que ha ocurrido un tipo de error. En una respuesta normal, el esclavo entrega como eco el mismo código de función original. Si hay una condición de error, el esclavo responde como eco el código de función original con el bit más significativo colocado como un valor 1.

El campo de **datos** es construido usando conjunto de 2 dígitos hexadecimales en el rango de 00 hasta FF. De acuerdo al modo de transmisión serial de la red, estos dígitos pueden ser un par de caracteres ASCII o caracteres RTU. El campo de datos también contiene información adicional que los esclavos usan para ejecutar la acción definida por el código de función. Esto puede incluir direcciones internas, cantidad de items a ser transferidos, etc.

El campo de datos de una respuesta desde un esclavo a un maestro contiene los datos pedidos si no ocurre un error. Si ocurre un error, el campo contiene un código de excepción que el maestro usa para determinar la próxima acción a ser tomada. Puede ser que no exista el campo de datos en ciertos tipos de mensajes.

El campo de **CHECKSUM** (examen de verificación de error) es usado para verificación de errores. Las redes seriales normales usan dos tipos de verificación de error.

Verificación de Parity (paridad) EVEN or ODD (par o impar, en inglés, respectivamente) y este valor totaliza la cantidad de bits colocados como 1 en el campo de datos y coloca el tipo de paridad en 0 o 1 representando un total par o impar de bits lógicos 1.

Cyclical Redundancy Check CRC (Verificación de redundancia) verifica el mensaje completo y es aplicado sin importar el método de paridad usado. El campo CRC se consiste de 2 bytes,

creando un valor binario de 16 bits. El CRC es calculado en el aparato transmisor y es recalculado y comparado por el aparato receptor.

Los caracteres de verificación y el mensaje son generados en el aparato maestro y aplicados al mensaje antes de la transmisión. El aparato esclavo verifica cada carácter y la estructura de mensaje durante la recepción.

Para conocer mas detalles del protocolo MODBUS, vea el documento de Gould MODBUS (P1-MBUS-300. B) "MODBUS Protocol Reference Guide". Si está disponible una versión más reciente, verifique esta situación antes de ordenar la documentación MODBUS.

Posibilidades de conexión de un PLC DL06 con MODBUS RTU

K

Como se ha establecido anteriormente, el PLC DL06 puede ser maestro o esclavo en una red MODBUS RTU.

Siendo el PLC DL06 un maestro en la red MODBUS RTU, se puede usar un sinnúmero de esclavos. Entre otros:

Otros PLC *Direct*LOGIC, como PLCs DL05 y DL06, CPUs D2-250-1 y D2-260, D3-350, D4-450 y todos los que usen el módulo DCM.

Terminator I/O con un controlador T1K-MODBUS, que es un producto de AUTOMATION DIRECT que permite tener un sistema modular de entradas y salidas discretas y análogas.

Esclavos de PLCs MODICON que usen protocolo MODBUS RTU.

Aparatos esclavos de cualquier marca que usen este protocolo, por ejemplo, variadores de frecuencia, relevadores de protección tales como los de GE-Multilin, instrumentos, aparatos SCADA, en fin, existen cientos de fabricantes que usan este protocolo como medio de comunicación.

Siendo el PLC DL06 un esclavo en la red MODBUS, se puede usar un sinnúmero de maestros. Entre otros:

Otros PLC *Direct*LOGIC, como PLCs DL05 y DL06, CPUs D2-250-1 y D2-260, D3-350, D4-450 y todos los que usen el módulo DCM.

Las interfases de operador que usen este protocolo. Entre ellos podemos nombrar los paneles C-more y C-more micro, los programas de PCs que trabajen con MODBUS RTU, tales como Wonderware, Lookout, Lookout Direct, LabView de National Instruments, y varias docenas de otros productos. Adicionalmente se están agregando cada día más de estos programas.

- PLCs MODICON maestros que usen protocolo MODBUS RTU.
- PLCs de otras marcas que puedan ser maestros de una red MODBUS RTU.
- Aparatos maestros de cualquier marca que usen este protocolo.

Es posible combinar maestros y esclavos de la misma forma que se hace con *Direct*NET, es decir, pueden usarse redes dentro de otra red. Aunque las configuraciones se pueden combinar en una aplicación, cada red permanece independiente. La estación maestra de una red no puede solicitar los datos directamente de estaciones de un esclavo en otra red. Esto no significa que usted no puede obtener los datos de estas redes, usted puede. Se requiere más de un aparato de comunicación para esa estación esclava.

El puerto multi-función del PLC DL06 le da la opción de usar especificaciones RS-232C, RS-422 o RS-485 con el protocolo MODBUS RTU, de la misma forma que *Direct*NET.

Convención de direcciones en MODBUS

El sistema MODBUS usa convenciones de direcciones diferentes que las direcciones de los PLCs *Direct*LOGIC. Con *Direct*NET, la convención de dirección es única. Otros PLCs *Direct*LOGIC deben hacer un cruzamiento a las direcciones de MODBUS de acuerdo a las tablas de ejemplo a continuación.

Por ejemplo, digamos que usamos un PLC DL06 como maestro MODBUS y otro como esclavo, y queremos leer dirección V1200 en el esclavo, esto corresponde a la dirección MODBUS 40641. Esto es, si Ud quiere llevar esta información al PLC DL06, que está en la red como maestro, debe leer al maestro desde la dirección MODBUS 40641. El contenido de esta memoria puede ser colocado, por ejemplo, en la memoria V3000. Si Ud quiere llevar otra información desde el PLC DL06 en V4000, por ejemplo, para el esclavo a la dirección V1400, debe escribir desde el maestro el contenido de la dirección V4000 para la dirección MODBUS 40769, que corresponde a V1400 en el esclavo.

Vea que para cada bit de entrada del PLC DL06 hay también una correspondencia a una dirección MODBUS. Por ejemplo, si Ud quiere leer el estado del bit X17 desde un PLC *Direct*LOGIC esclavo, la dirección de MODBUS es 2063. Sin embargo, para leer el bit de entrada existe una función determinada. Vea más datos en las próximas páginas.

	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
X160	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	40407
X200	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	40410
X220	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	40411
X240	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	40412
X260	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	40413
X300	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	40414
X320	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	40415
X340	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	40416
X360	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	40417
X400	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	40420
X420	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	40421
X440	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	40422
X460	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	40423
X500	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	40424
X520	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	40425
X540	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	40426
X560	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	40427
X600	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447	40430
X620	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	40431
X640	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479	40432
X660	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	40433
X700	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511	40434
X720	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527	40435
X740	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543	40436

Vea que para cada bit de salida del PLC DL06 hay una correspondencia a una dirección MODBUS. Por ejemplo, si Ud quiere leer el estado del bit Y25 desde un PLC *Direct*LOGIC esclavo, la dirección de MODBUS es 2070. Sin embargo, para leer el bit de salida existe una función determinada. Vea más datos en las próximas páginas.

065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079	2080
081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095	2096
097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111	2112
113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128
129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143	2144
145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159	2160
161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175	2176
177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192
193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207	2208
209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223	2224
225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239	2240
241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255	2256
257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271	2272
273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287	2288
289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303	2304
305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320
321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336
337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352
353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368
369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384
385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399	2400
401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415	2416
417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431	2432
433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447	2448
449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463	2464
465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479	2480
481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495	2496
497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511	2512
513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527	2528
529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543	2544
545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559	2560



Apéndice K: Introducción a comunicaciones seriales

Vea que para cada memoria V de usuario del PLC DL06 hay una correspondencia a una dirección MODBUS. Por ejemplo, si Ud quiere leer el estado de la memoria V1300 desde un PLC *DirectLOGIC* esclavo, la dirección de MODBUS es 704. Sin embargo, para escribir o leer una memoria en el PLC existe una función determinada.

Memoria V	Memoria V	Memoria V	Memoria V	Memoria V	Memoria V
1200	640	1250	680	1320	640
1201	641	1251	681	1321	641
1202	642	1252	682	1322	642
1203	643	1253	683	1323	643
1204	644	1254	684	1324	644
1205	645	1255	685	1325	645
1206	646	1256	686	1326	646
1207	647	1257	687	1327	647
1210	648	1260	688	1330	648
1211	649	1261	689	1331	649
1212	650	1262	690	1332	650
1213	651	1263	691	1333	651
1214	652	1264	692	1334	652
1215	653	1265	693	1335	653
1216	654	1266	694	1336	654
1217	655	1267	695	1337	655
1220	656	1270	696	1340	656
1221	657	1271	697	1341	657
1222	658	1272	698	1342	658
1223	659	1273	699	1343	659
1224	660	1274	700	1344	660
1225	661	1275	701	1345	661
1226	662	1276	702	1346	662
1227	663	1277	703	1347	663
1230	664	1300	704	1350	664
1231	665	1301	705	1351	665
1232	666	1302	706	1352	666
1233	667	1303	707	1353	667
1234	668	1304	708	1354	668
1235	669	1305	709	1355	669
1236	670	1306	710	1356	670
1237	671	1307	711	1357	671
1240	672	1310	712	1360	672
1241	673	1311	713	1361	673
1242	674	1312	714	1362	674
1243	675	1313	715	1363	675
1244	676	1314	716	1364	676
1245	677	1315	717	1365	677
1246	678	1316	718	1366	678
1247	679	1317	719	1367	679

Y así existe una equivalencia para cada tipo de elementos en el PLC DL06.

Vea más datos en las próximas páginas.



Nota. Hay disponible 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)

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.

K

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

Recordemos que el sistema MODBUS usa convenciones de direcciones diferentes que las direcciones de los PLCs *Direct*LOGIC. Con *Direct*NET la convención de dirección es única y es la usada en los PLCs *Direct*LOGIC.

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

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.

Coil 2064

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

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.

Input Reg. 8

Valores corrientes de T (V)	128	V0 – V177	0 - 127	Input Register
-----------------------------	-----	-----------	---------	----------------

Ejemplo 4: Relevador de control C54

Encuentre la dirección de MODBUS para el relevador 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.

Coil 3116

Relevadores de control (C)	512	C0 – C77	3072 – 3583	Coil
----------------------------	-----	----------	-------------	------

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 para 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". Un asunto 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 inaccesibles. La ecuación real 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.

- 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 relación usada para cada grupo de datos.

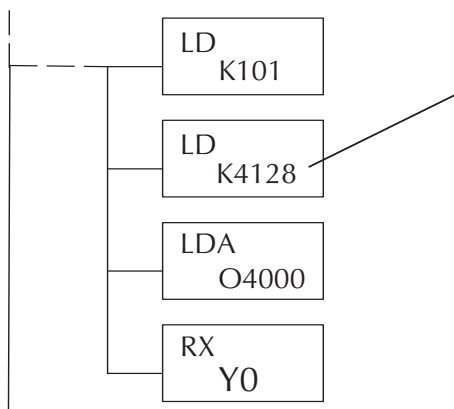
Tipo de de datos discretos				
Tipo de memoria del DL06	Rango del PLC (Octal)	(Modo 484)	Dirección (Modo 584/984)	Tipo datos MODBUS
Entradas globales (GX)	GX0-GX1746	1001 - 1999	10001 - 10999	Entrada
	GX1747-GX3777	---	11000 - 12048	Entrada
Entradas (X)	X0 - X1777	---	12049 - 13072	Entrada
Relevadores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

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 máxima posible 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).



Nota 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.



Nota 2. Hay disponible 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

- Encuentre el valor corriente en la tabla. $TA10 = 8$ decimal
- Convierta T10 a decimal (8). $8 + 3001 =$ **3009**
- 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

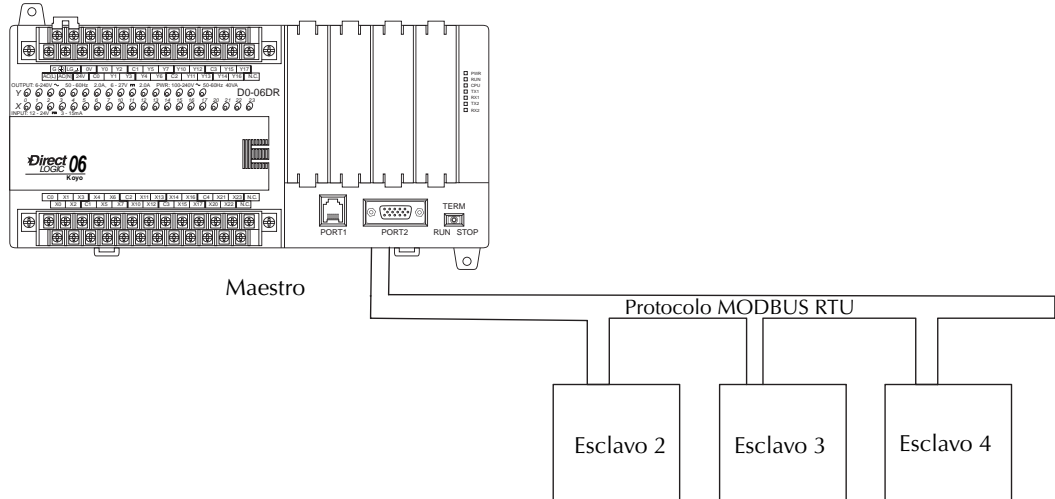
Encuentre la dirección MODBUS para C54. Dirección del PLC (Dec.)+Dir. inicial+ modo

- 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

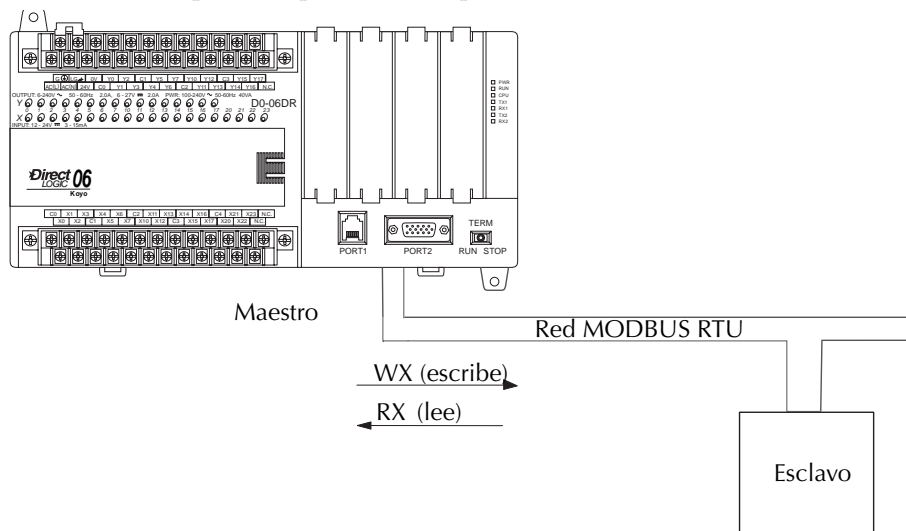
Operación del maestro en una red MODBUS RTU

Esta sección describe cómo el PLC DL06 puede comunicarse en una red de MODBUS como maestro (también es válido para *Modbus RTU 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.



K

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.

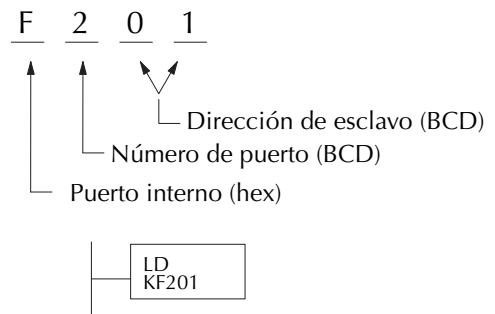


Paso 1: Identifique el número del puerto maestro y el número de cada esclavo

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.

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).

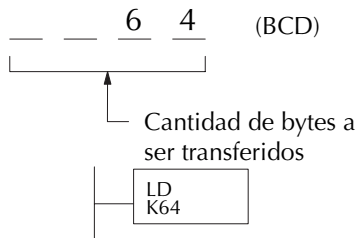


Paso 2: Cargue el no. de bytes a transferir:

La segunda instrucción LD determina la cantidad 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.

La cantidad 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.

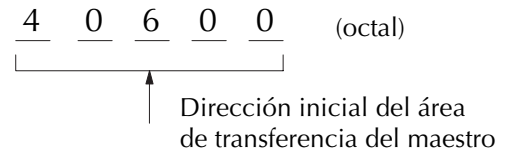
La siguiente tabla muestra los rangos de bytes para los varios tipos de productos *DirectLOGIC™*.



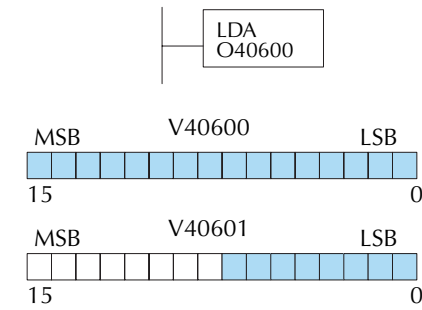
Memoria DL05 / 06 / 205 / 350 / 405	Bits por unidad	Bytes
Memoria V del valor corriente de temporizador o contador	16 16	2 2
Entradas (X, SP)	8	1
salidas (Y, C, estapas, bits de T/CT)	8	1
Memoria Scratch Pad	8	1
Estado de diagnóstico	8	1

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 la cantidad de bytes previamente especificado al área de memoria en la dirección inicial especificada en la instrucción LDA. Recuerde que debe cruzar la dirección de memoria del PLC a la dirección n MODBUS RTU.



Para una instrucción RX, la CPU DL06 lee la cantidad 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, se modificarán sólo los 8 bits menos significativos de la última dirección de palabra. Los restantes 8 bits no son afectados.

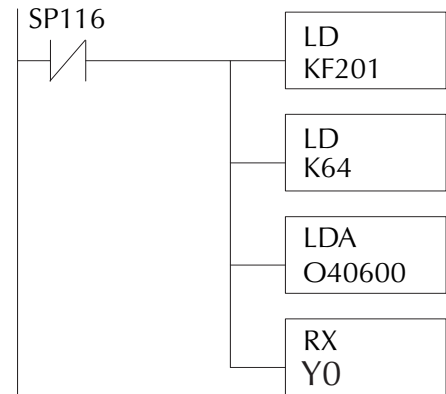


Paso 4: Especifique la memoria del esclavo

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 MODBUS DL405, DL205, o DL06 – especifique la dirección correspondiente en la instrucción WX y RX como la dirección nativa del esclavo.

Recuerde que debe cruzar la dirección de memoria del PLC a la dirección en MODBUS RTU.



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

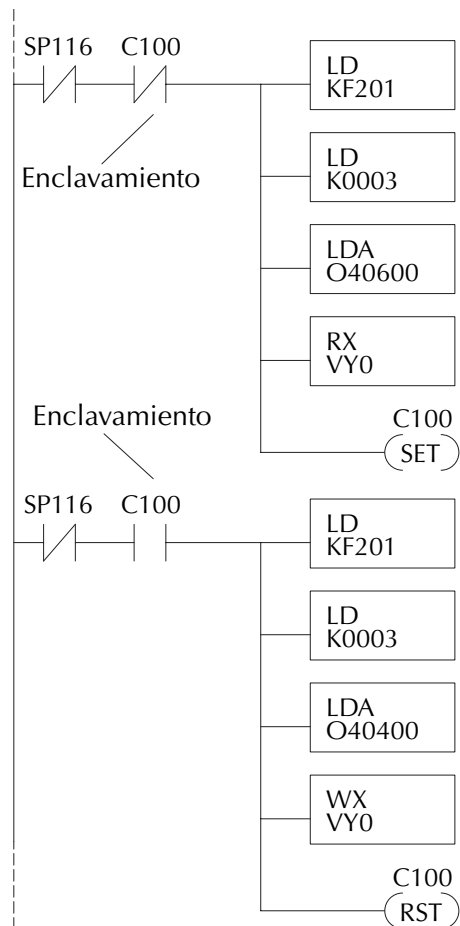
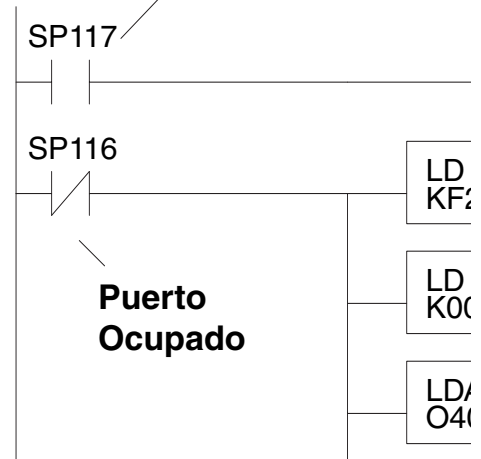
Si usted usa varias instrucciones WX y RX en el programa ladder, usted tiene que enclavar las rutinas para asegurarse que todas las rutinas se ejecutan completas.

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.

Puerto con error de comunicaci



Configuración del puerto 2 del PLC DL06 para MODBUS

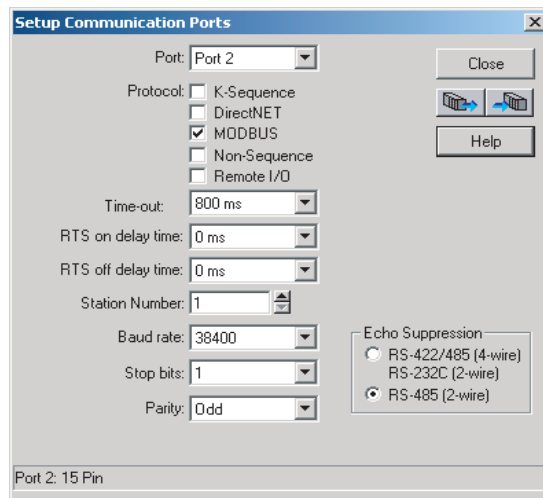
Esta sección describe cómo configurar el puerto 2 para establecimiento de una red con MODBUS RTU, usando *DirectSOFT*. Esto le permitirá conectar el sistema del PLC DL06 directamente con redes de MODBUS usando este protocolo. Los maestros MODBUS en la red deben ser capaces de generar comandos MODBUS para leer o de escribir los datos apropiados.

Configuración del puerto 2 como protocolo MODBUS RTU

En *DirectSOFT*, haga clic en menú PLC, luego SETUP, y por último "Set up Sec. Comm Port".

- **Port:** Elija "Puerto 2" desde la lista que aparece al hacer clic en la flecha para abajo.
- **Protocol:** Acepte el cuadro a la izquierda de "MODBUS" haciendo clic en él (o use 56 AUX. en el programador D2-HPP, y seleccione "MBUS"), y entonces usted verá el diálogo como en la figura adyacente.
- **Timeout:** Cantidad de tiempo que el puerto esperará después de que envíe un mensaje para obtener una respuesta antes de registrar un error.
- **RTS ON / OFF Delay Time:** RTS ON Delay Time especifica el tiempo que el PLC DL06 espera para enviar datos después de que haya activado la línea de señales RTS. El RTS OFF Delay Time especifica el tiempo que el PLC DL06 espera para desactivar la línea de señales RTS después de que se hayan enviado datos. *Al usar el PLC DL06 en una red multinodo, se debe configurar el RTS ON Delay Time a 5ms o más y el RTS OFF Delay Time a 2ms o más. Si usted encuentra problemas, se puede aumentar el tiempo.*
- **Station Number:** (Dirección de nodo): Para hacer que el PLC sea un maestro MODBUS, coloque aquí un "1". El rango posible de direcciones para direcciones de esclavo MODBUS es a partir de 1 a 247, pero las instrucciones de red DL06 usadas en el modo maestro tienen acceso solamente a los esclavos 1 a 99. Cada esclavo debe tener un número único. En la energización, el puerto es automáticamente un esclavo, a menos que y hasta que el DL06 ejecuta las instrucciones de red de la lógica ladder que usen el puerto como maestro. Después de eso, el puerto invierte de nuevo a modo esclavo hasta que la lógica ladder use el puerto otra vez.
- **Baud Rate:** Las velocidades disponibles incluyen 300, 600, 1200, 2400, 4800, 9600, 19200, y 38400 bauds. Escoja una velocidad más alta inicialmente, y bájela la velocidad si usted experimenta errores de datos o problemas de ruido en la red. Importante: Usted debe configurar la velocidad de todos los nodos en la red al mismo valor. Vea el manual de cada producto esclavo para esos detalles.
- **Stop Bits:** Seleccione entre 1 o 2 bits de parada para uso en el protocolo.
- **Parity:** Seleccione paridad none, even, o odd.
- **Echo Suppression:** Seleccione la configuración apropiada del cableado usada en el puerto 2.

Luego haga clic en el botón de archivar al PLC para enviar la configuración al PLC y cierre haciendo clic en Close.



K

Configuración del puerto 2 como Modbus RTU con lógica Ladder

El puerto 2 en el DL06 se puede también configurar para usar el protocolo *Modbus RTU* usando lógica ladder dentro del programa del PLC. También, los parámetros de comunicación se pueden configurar como los parámetros en el otro aparato con el cual el PLC se comunicará.

Note que los parámetros del puerto 2 nunca se almacenan al disco con *DirectSOFT* de modo que si usted está usando el puerto 2, con excepción de la configuración por defecto, es una buena idea incluir la configuración en el programa ladder.

Para configurar el puerto 2 en lógica ladder se deben escribir valores apropiados a V7655 (palabra 1) y a V7656 (palabra 2) para especificar la configuración del puerto. Luego escriba K0500 a V7657 (palabra 3) para pedir a la CPU que acepte los valores.

Una vez que la CPU vea K0500 en V7657, verificará los parámetros de comunicación que se han seleccionado y después cambiará el valor en V7657 según los resultados de esta prueba.

Si éstos son válidos, la CPU cambiará el valor en V7657 a 0A00 ('A' para aceptado). Si había un error en los valores, la CPU cambiará el valor en V7657 a 0E00 ('E' por error).



NOTA: Sugerencia. En vez de construir las palabras de configuración manualmente desde las tablas, use *DirectSOFT* para configurar el puerto como desee y use *Dataview* para ver las palabras en V7655 y V7656 como BCD/HEX. Luego use estos números en el código de configuración.

Los datos que se escriben a las palabras de configuración tienen dos formatos. El formato que se usa que depende si es K-Sequence, *DirectNET*, MODBUS RTU (método 1) o ASCII (método 2).

Observe que es la configuración por defecto del puerto 2 es:

Detección automática entre protocolos K-Sequence, *DirectNET*, and MODBUS RTU

Timeout = Base Timeout x 1 (800 ms)

RTS on delay time = 0 ms

RTS off delay time = 0ms

Número del nodo= 1

Baud rate = 19200

Bits de Stop = 1

Parity = impar

Formato = Hexadecimal

Supresión de eco = RS-422/485 (4-wire) o RS-232C

Las palabras mencionadas V7655 y V7656 se configuran según las tablas en las páginas siguientes.

Configuración del Puerto 2 usando Modbus RTU

La palabra V7655 está formada de los siguientes grupos de bits:

0yyy 0TTT mmmm mxxx

Siendo yyy = RTS ON-delay

TTT= Time out de comunicación

mmmmm= tipo de protocolo

xxx = RTS off-delay

V7655 (Palabra 1)	RTS On-delay	Timeout (% de timeout)	Protocolo	RTS Off-delay
0yyy 0ttt mmmm mxx	yyy	TTT	mmmmm	xxx
	000 = 0 ms	000 = 100%	10000 = K-Sequence	000 = 0ms
	001 = 2 ms	001 = 120%	01000 = DirecNET	001 = 2ms
	010 = 5 ms	010 = 150%	00100 = MODBUS RTU	010 = 5ms
	011 = 10 ms	011 = 200%		011 = 10ms
	100 = 20 ms	100 = 500%		100 = 20ms
	101 = 50 ms	101 = 1000%		101 = 50ms
	110 = 100 ms	110 = 2000%		110 = 100ms
	111 = 500 ms	111 = 5000%		111 = 500ms

La palabra V7656 para Modbus RTU está formada de los siguientes grupos de bits:

pps0 ebbb xaaa aaaa

Siendo pp = Paridad

s = Bits de Stop

e = Supresión de eco

bbb = Tasa de Baud

x = Modo del protocolo

aaaaaaa = Dirección del esclavo

V7656 (Palabra 2)	Paridad	Bits de Stop	Supresión de eco	Tasa de Baud
<i>DirecNET</i>				
pps0 ebbb xaaa aaaa	pp	s	e	bbb
	00 = Ninguna	0 = 1 bit	0 = 232 or 422	000 = 300
	10 = Odd	1 = 2 bits	1 = 485, 2 wire	001 = 600
	11 = Even			010 = 1200
				011 = 2400
				100 = 4800
				101 = 9600
				110 = 19200
				111 = 38400

V7656 (Palabra 2) continuada	Protocolo	Dirección del nodo
MODBUS RTU	MODBUS RTU	xxxxxx (Modbus RTU Direct)
pps0 ebbb xaaa aaaa	x	_aaaaaa (K-Seq. & MODBUS RTU)
	0 = Hex	K-Sequence: 1-99
	1 = ASCII	1-99
		MODBUS: 1-247
		<i>Use el número como hexadecimal</i>

La palabra V7657 debe ser cargada con el valor K0500 con ladder, para que los valores en las otras 2 palabras sean verificados y cargados

Ejemplo:

Para configurar el puerto 2 para protocolo MODBUS RTU con lo siguiente:

RTS On-delay of 5 ms,

Base timeout x1,

RTS Off-delay of 5 ms,

Paridad Odd,

1 Stop bit,

Supresión de eco para RS232-C/RS422,

19200 Baud,

Número de nodo 5

Explicación de 0yyy 0TTT mmmm mxxx

0yyy - 0010 - RTS On-delay de 5ms,

0TTT - 0000 - Base timeout x1,

mmmmm - 000100 - MODBUS

xxx - 010 - RTS Off-delay de 5ms,

Ésto es, 0010 0000 0010 0010 equivalente a 2 0 2 2

Explicación de pps0 ebbb xaaa aaaa

pp - 10 Paridad Odd,

s0 - 00 1 Stop bit,

e - 0 Supresión de eco para RS232-C/RS422, 1

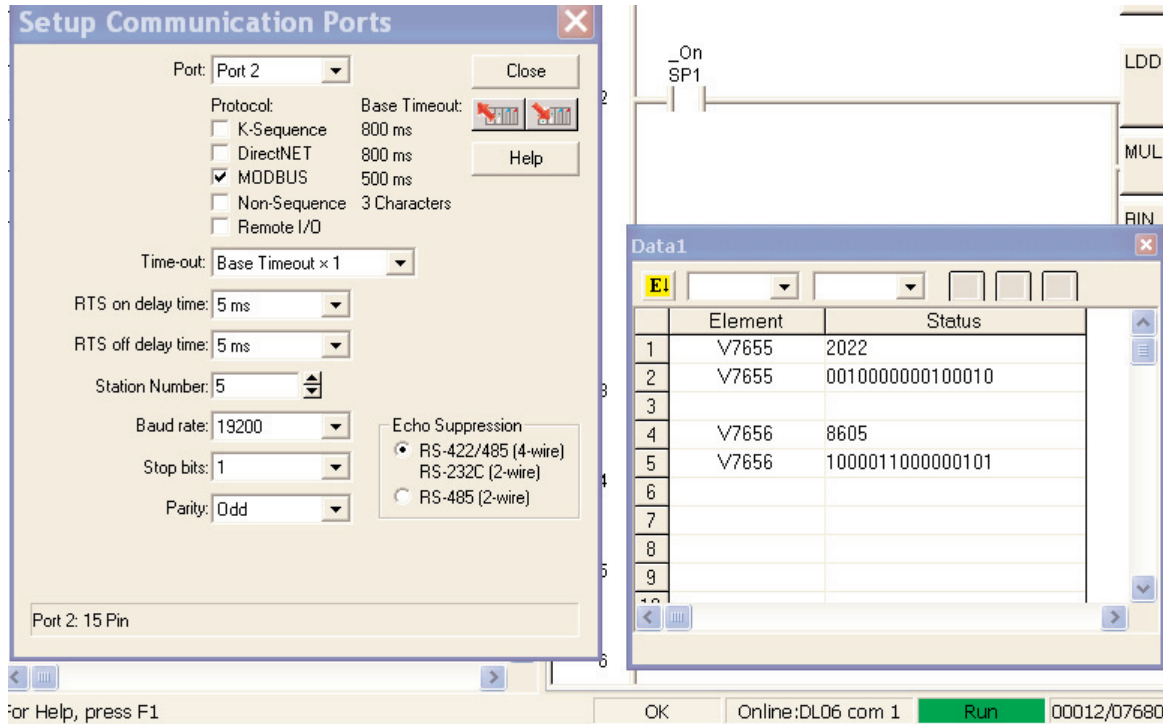
bbb - 110 19,200 baud,

x - 0 Modo de protocolo

aaa aaaa - 000 0101 Número de nodo 5 hexadecimal, que corresponde a 5 decimal

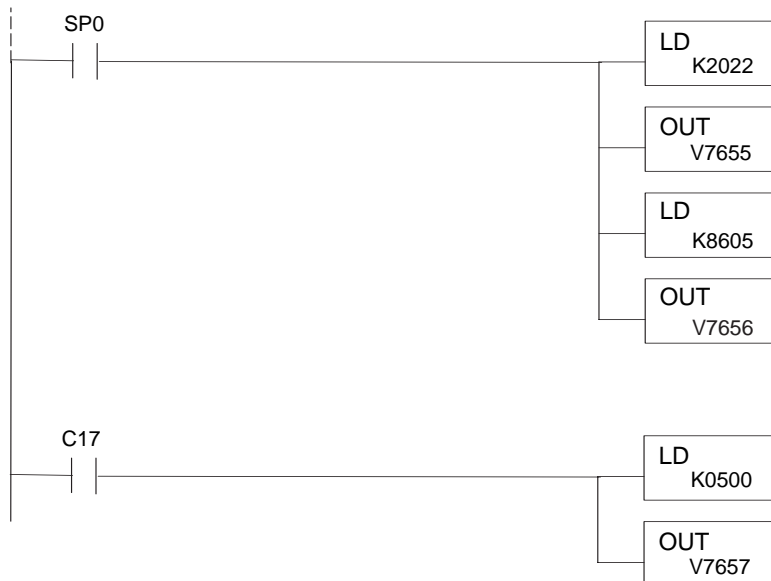
Ésto es, 1000 0110 0000 0101 equivalente a 8 6 0 5

Una forma de hacer esta verificación de los números a cargar en las palabras 1 y 2 es usar *DirectSOFT* y *Data View*, como sigue:



K

Se usaría la lógica ladder mostrada en la figura siguiente (C17 se ejecuta después de SP0) :



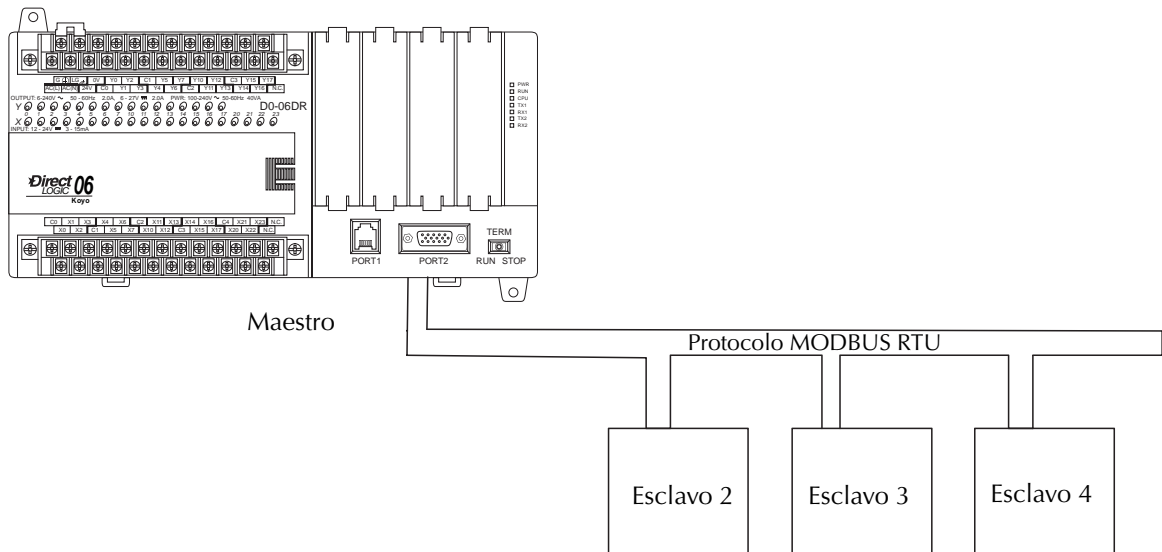
La configuración del puerto con *DirectSOFT* o con ladder se hace de la misma forma para usar las instrucciones RX /WX o MRX /MWX.

Operación como maestro usando 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 de comunicaciones a la red. Esta sección le enseña cómo diseñar la lógica necesaria 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

Hay más explicaciones sobre este uso de estas instrucciones en el capítulo 4 y en el capítulo 5.

Ejemplos de MRX/MWX en *DirectSOFT*

Vea un ejemplo en el capítulo 5, después de la instrucción MWX. También hay otro ejemplo en el capítulo 11, en la página 11-55.

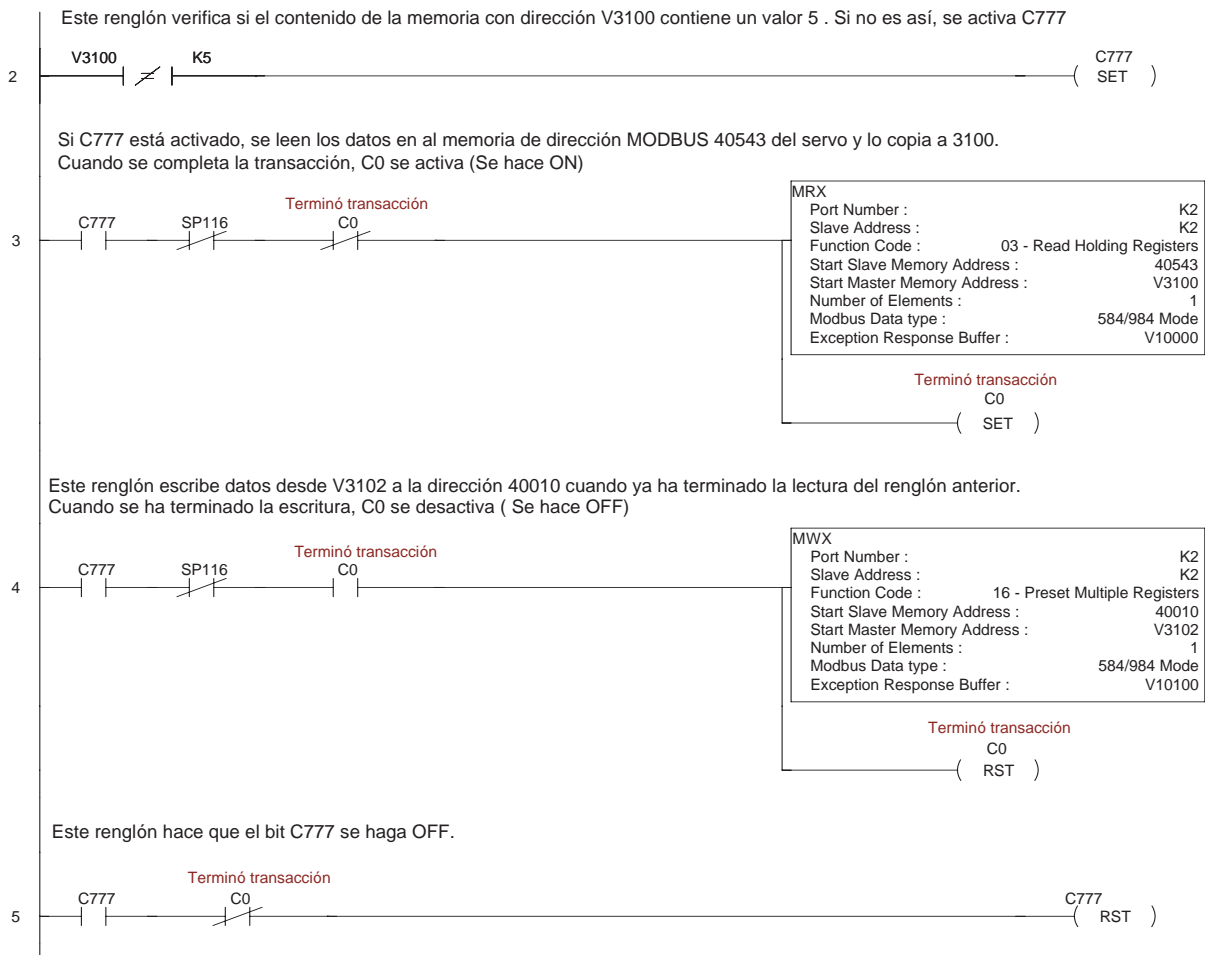
El puerto 2 del DL06 tiene dos contactos de relevadores especiales asociados a él (véa el apéndice D para relevadores especiales). Uno indica el "puerto ocupado" (SP116) y el otro indica "Error de comunicación del puerto" (SP117).

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 el bit de error vuelve a OFF 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.

Vea a continuación una instrucción de comunicación de lectura simple, como ejemplo para leer datos del accionamiento esclavo de un servo motor *Sureservo*, que es uno de los productos de **AUTOMATION DIRECT**.



Enclavamientos múltiples de lectura y escritura

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

En el ejemplo anterior, C0 es el bit de enclavamiento.

Ud. también puede usar un contador para hacer el enclavamiento de varias operaciones de comunicación, o una instrucción shift register, como fue hecho en el ejemplo para *DirectNET*.

Si está usando programación por etapas, se 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 ellas sea activa 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.

Respuesta de excepción (Exception response)

Cuando el PLC maestro pide una transacción a un esclavo, se espera una respuesta normal. Pero pueden suceder otros eventos a partir de la petición de transmisión de datos.

- Si el esclavo recibe la petición y no hay error, responde con una respuesta normal.
- Si el esclavo no recibe la petición debido a un error de comunicación, no hay una respuesta al maestro. En ese caso el maestro puede generar una condición de timeout.
- Si el esclavo recibe la petición pero detecta un error de comunicación, no hay una respuesta al maestro. En ese caso el maestro puede generar una condición de timeout.
- Si el esclavo recibe la petición y no hay error, pero puede ser que no sepa como responder con una respuesta normal, por ejemplo, se trata de leer a un registro que no exista en el esclavo, el esclavo devuelve una respuesta de excepción (exception response) informando al maestro cual es el tipo del error.

En una respuesta normal, colocando el mismo código de función generado por el maestro en la parte de la respuesta que corresponde al código de función. Todos los códigos de función tienen el bit más significativo como 0. En una respuesta de excepción este bit se hace 1. Esto hace que el código en una respuesta de excepción sea más alto que cualquier código de función normal, el cual llega solamente hasta 50 hexadecimal.

Es posible ver esta respuesta de excepción en las instrucciones MRX y MWX. Vea en el ejemplo de la página anterior que se han dejado memorias disponibles en el PLC maestro para poder almacenar los datos de la respuesta de excepción en las memorias V10000 y V10100. De hecho, la respuesta de excepción ocupa más de 16 bits, como veremos a continuación.

Los códigos de error definidos en el manual de referencia de MODBUS- rev. J, son los siguientes;

- 01 - Función no reconocida - El código recibido por el esclavo no tiene una acción permitida por el esclavo.
- 02 - Dirección de datos no reconocida - La dirección entregada por el maestro no es una dirección permitida en el esclavo.

- 03 - Valor de dato no reconocido - El valor en el campo de datos del maestro no es un valor permitido en el esclavo
- 04 - Falla del aparato esclavo - Ocurrió un error que no puede ser corregido durante la acción de la petición
- 05 - Reconocido - El esclavo ha aceptado la petición y la está procesando, pero va a tomar un tiempo de un período grande. esta respuesta es enviada para evitar un error de timeout en el maestro.
- 06 - Aparato esclavo ocupado- El esclavo está procesando un comando de programa de alta duración. El maestro podría retransmitir el mensaje mas adelante cuando el esclavo esté libre
- 07- Negative acknowledge - El esclavo no puede ejecutar la función del programa recibido en la petición.
- 08 - Error de paridad de memoria - El esclavo trató de leer una memoria extendida, pero detectó un error de paridad en esa memoria.

Por ahora, explicaremos cómo interpretar el código de error de estos datos. El tema de respuesta de excepción huye del ámbito de este manual y el lector puede consultar otras obras para poder conocer más sobre este asunto.

La instrucción de leer, MRX, instruye al esclavo a entregar al maestro datos relacionados al esclavo 2. Si el esclavo 2 está configurado en la red y es el único esclavo, todo debe funcionar bien. Digamos, que por un error de operación, el esclavo fue cambiado a 4.

Ésto generaría una respuesta por excepción. Los datos serían almacenados en V10000 y las memorias consecutivas.

Digamos que V10000 contenga 8302. Este valor viene en formato con bytes intercambiados. Sería más fácil visualizar ésto como 02 83. 02 es la dirección del nodo. 83 es el código de la función (03) con el bit más significativo como 1 (las respuestas de excepción tienen siempre el bit más significativo como 1).

V10001 podría contener, por ejemplo, C002. Intercambiando bytes, sería 02 C0. 02 es el código de error real. C0 es el primer byte del CRC. Podemos no hacer caso de estos datos así como los datos en V10002.

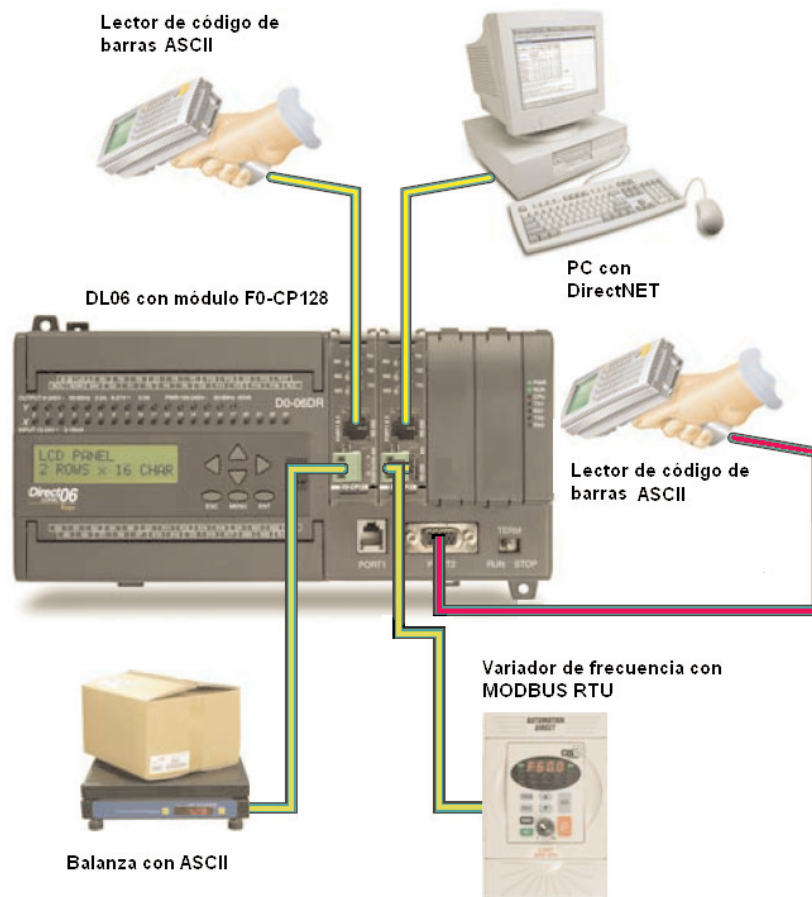
Observe que el código de error 02 dice la dirección de datos **no es reconocida**.

Esto estaría correcto puesto que estamos pidiendo la dirección 40101 de Modbus y ésta no existe actualmente en el proyecto Modbus que estamos corriendo.

Comunicación con el protocolo Non-sequence (ASCII)

Se puede usar el protocolo Non-sequence para comunicación con cualquier dispositivo que utilice el intercambio de informaciones con caracteres ASCII. Esta forma de transmisión es una de las primeras formas de comunicación, es muy común y es probablemente la cosa más cercana a un protocolo estándar "industrial" en existencia. El PLC DL06 puede ser un esclavo y puede ser un maestro Non-sequence en el puerto 2. Se pueden utilizar los estándares RS 232-c, RS-422 y RS-485 en el puerto 2 para el protocolo Non-sequence.

La forma de conectar el maestro o los esclavos es muy similar a la forma hecha con *DirectNET* en el caso de RS-232. Vea un ejemplo de aplicación a continuación,



Hay varios ejemplos de código ladder comenzando en la página 5-213 del capítulo 5.

El conjunto de instrucciones para manejar caracteres ASCII es relativamente simple, lo que implica que puede hacer muchas funciones simples. Si necesita funciones más poderosas, por favor considere usar el módulo F0-CP128, que es un módulo con un microprocesador incorporado, con un sistema poderoso de instrucciones en BASIC, lo que permite aliviar las funciones de la CPU del PLC DL06, y puede comunicarse a velocidades de comunicación más altas. Vea más detalles en el manual en español de este módulo, el que puede bajar desde nuestro sitio de Internet www.automationdirect.com.

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 RTS se ha hecho ON.

- **RTS Off Delay Time:** tiempo que espera el PLC DL06 después de mandar datos para hacer OFF la señal RTS.

- **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 la tasa de Baud del aparato 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.

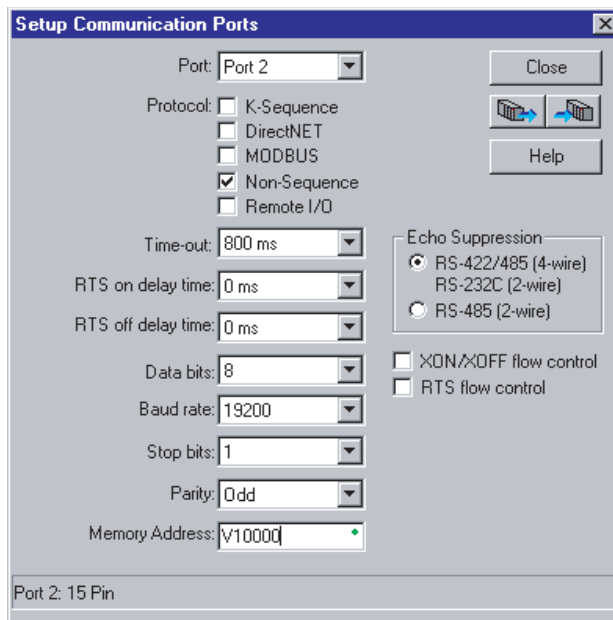
- **Echo Suppression:** Seleccione el botón de radio adecuado basado en la configuración usada en el puerto 2 (RS-232C, RS-422 o RS-485).

- **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.



Configuración del puerto 2 como Non-sequence con lógica Ladder

El puerto 2 en el DL06 se puede también configurar para usar el protocolo *Non-sequence* usando lógica ladder dentro del programa del PLC. También, los parámetros de comunicación se pueden configurar como los parámetros en el otro aparato con el cual el PLC se comunicará.

Note que los parámetros del puerto 2 nunca se almacenan al disco con *DirectSOFT* de modo que si usted está usando el puerto 2, con excepción de la configuración por defecto, es una buena idea incluir la configuración en el programa ladder.

Para configurar el puerto 2 en lógica ladder se deben escribir valores apropiados a V7655 (palabra 1) y a V7656 (palabra 2) para especificar la configuración del puerto. Luego escriba K0500 a V7657 (palabra 3) para pedir a la CPU que acepte los valores.

Una vez que la CPU vea K0500 en V7657, verificará los parámetros de comunicación que se han seleccionado y después cambiará el valor en V7657 según los resultados de esta prueba.

Si éstos son válidos, la CPU cambiará el valor en V7657 a 0A00 ('A' para aceptado). Si había un error en los valores, la CPU cambiará el valor en V7657 a 0E00 ('E' por error).



NOTA: Sugerencia. En vez de construir las palabras de configuración manualmente desde las tablas, use *DirectSOFT* para configurar el puerto como desee y use *Dataview* para ver las palabras en V7655 y V7656 como BCD/HEX. Luego use estos números en el código de configuración.

Los datos que se escriben a las palabras de configuración tienen dos formatos. El formato que se usa que depende si es K-Sequence, *DirectNET*, MODBUS RTU (método 1) o ASCII (método 2).

Observe que es la configuración por defecto del puerto 2 es:

Detección automática entre protocolos K-Sequence, *DirectNET*, and MODBUS RTU

Timeout = Base Timeout x 1 (800 ms)

RTS on delay time = 0 ms

RTS off delay time = 0ms

Número del nodo= 1

Baud rate = 19200

Bits de Stop = 1

Parity = impar

Formato = Hexadecimal

Supresión de eco = RS-422/485 (4-wire) o RS-232C

Las palabras mencionadas V7655 y V7656 se configuran según las tablas en las páginas siguientes.

Configuración del puerto 2 como Non-sequence

Palabra 1	RTS On-delay	Timeout (en% del timeout normal)	Protocolo	RTS Off-delay
0yyy 0ttt mmmm mxx	yyy	ttt	mmmmm	xxx
DL05/06: V7655	000 = 0ms	000 = 100%	00010 = Non-Sequence	000 = 0ms
	001 = 2ms	001 = 120%		001 = 2ms
	010 = 5ms	010 = 150%		010 = 5ms
	011 = 10ms	011 = 200%		011 = 10ms
	100 = 20ms	100 = 500%		100 = 20ms
	101 = 50ms	101 = 1000%		101 = 50ms
	110 = 100ms	110 = 2000%		110 = 100ms
	111 = 500ms	111 = 5000%		111 = 500ms

Palabra 2	Paridad	Stop Bits	Supresión de eco (válido solamente para DL06)	Tasa de Baud	Modo de Protocolo	
pps0 ebbb xaaa aaaa	pp	s	e	bbb	01110000 = No Control de flujo	
DL05/06: V7656	00 = Ninguna	0 = 1 bit	0 = RS-232C, RS-422 o RS-485 (4 wire)	000 = 300	01110001 = Control de flujo Xon/Xoff	
	10 = Odd	1 = 2 bits	1 = RS-485 (2 wire)	001 = 600	01110010 = Control de flujo RTS	
	11 = Even				010 = 1200	01110011 = Control de flujo Xon/Xoff y RTS
					011 = 2400	
					100 = 4800	
					101 = 9600	
					110 = 19200	
111 = 38400						

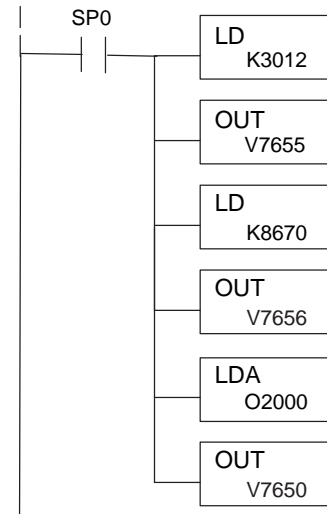
Palabra 3	Dirección de Memoria para datos
DL05/06: V7650	Valor hexadecimal de la dirección de memoria V temporaria para almacenar los datos ASCII que vienen al PLC. Configure este parámetro en una dirección de memoria V que tenga bastantes posiciones de memoria consecutivas libremente para almacenar la secuencia de caracteres más larga que vendrá al PLC.



Ejemplo:

Configure el puerto 2 para para comunicaciones Non-sequence (ASCII) con lo siguiente:

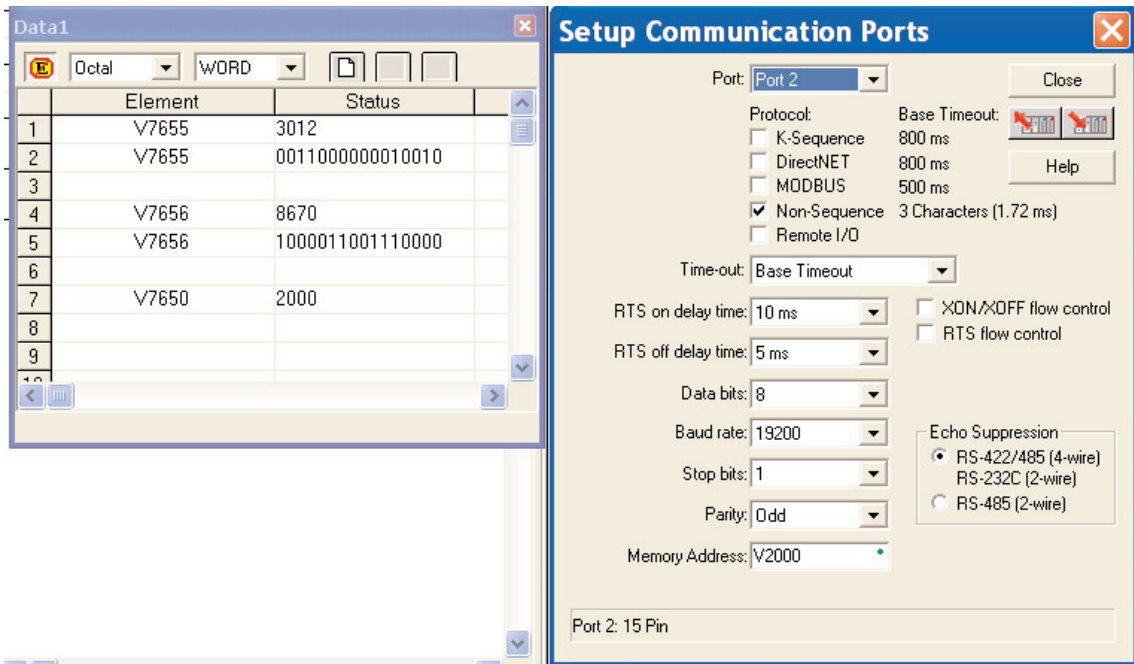
- RTS On-delay de 10ms,
- Base timeout x1,
- RTS Off-delay de 5ms,
- Paridad Odd,
- 1 bit de Stop,
- Supresión de eco para RS232-C/RS422,
- 19,200 Baud,
- 8 bits de datos,
- Memoria V intermediaria comenzando en V2000 y sin control de flujo



Se usaría la lógica ladder mostrada en la figura adyacente.

El lector puede verificar que los números colocados en el código ladder corresponden a la configuración del ejemplo.

Una forma de hacer esta verificación de los números a cargar en las palabras 1, 2 y 3 es usar *DirectSOFT* y *Data View*, como sigue:



Note que V7655 y V7656 son mostrados en *Data View* como formato BCD/HEX y binario, respectivamente, desde arriba para abajo.

La memoria V2000 es mostrada en formato octal.

Ejemplo para comunicación ASCII

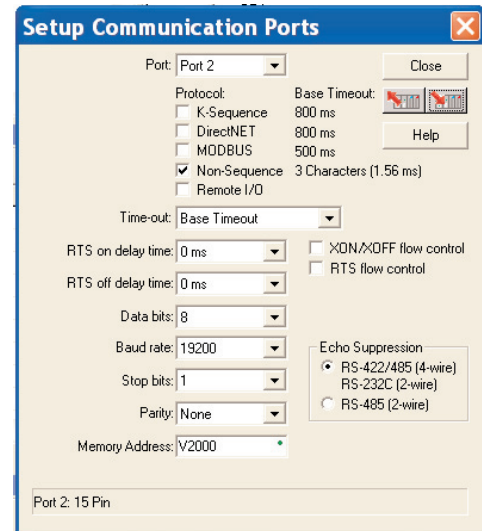
Un ejemplo simple puede ser lo siguiente;

Imaginemos que el PLC DL06 mide temperaturas de un horno con un módulo de entradas analógicas cuyo rango en grados Fahrenheit es 32 a 600 grados.

Es necesario en este ejemplo registrar en un archivo que acepte comandos ASCII, tal como una impresora serial, un visor que acepte comandos ASCII o aún la función Hyperterminal de Windows, en el momento en que la temperatura pasa de 490 grados, como alarma de alta temperatura y de 530 grados, como alarma de temperatura muy alta., con la fecha y hora del evento.

Para eso podemos usar el programa que se muestra en una de las próximas páginas; más detalles sobre la programación son dadas en la parte derecha del diagrama.

El cable usado entre el puerto 2 del PLC y la computadora PC es en D2-DSCBL-1. La configuración del puerto 2 del PLC es la que se muestra en la figura adyacente:



Hyperterminal es un programa que viene ya en Windows y que permite conectarse a otras computadoras, o a servicios On line, por ejemplo, a través de la comunicación del puerto serial de la PC, usando caracteres ASCII. Hyperterminal es una herramienta fácil de usar pero no tiene muchas funciones.

Para aprender mas de Hyperterminal, lea las intrucciones que vienen en la ayuda de Windows.

Para configurar un PC con Hyperterminal, por ejemplo, para prueba antes de conectar a una impresora, siga los pasos siguientes:

Haga clic en **Iniciar (Start)** y luego **Accesorios** en Windows y seleccione **Hyperterminal**.

Aparece la figura adyacente. Coloque un nombre tal como



“Conexión a DL06” en el campo Name. Luego haga clic en OK.

Defina cual es el puerto serial de la computadora con el cuadro de diálogo como mostrado en la figura adyacente:

Aparecerá un diálogo de definición de la configuración del puerto de la computadora. En este ejemplo, es COM2.

Luego coloque los valores de configuración adecuados y luego pase al paso siguiente de configuración de la velocidad de transmisión.



Apéndice K: Introducción a comunicaciones seriales

Los mismos valores deben ser colocados en el diálogo del puerto 2 del PLC DL06. Vea la figura adyacente.

Seleccione el protocolo Non sequence en este puerto, usando *DirectSOFT* o el código mostrado en la página K-57.

Estas acciones dejan al sistema listo para probar el programa.

Coloque el programa de la página siguiente en el PLC, coloque el PLC en modo RUN y simule la subida de temperatura.

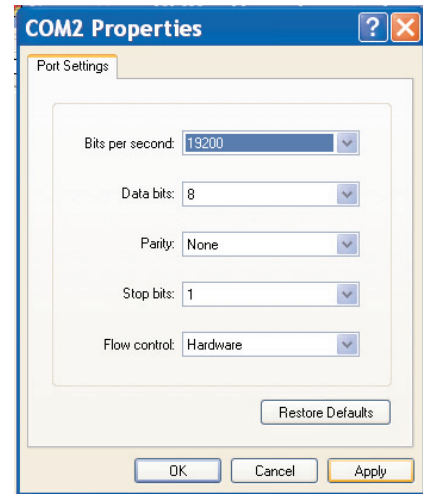
Se puede simular la subida de temperatura si usa Data View y no tiene aún configurado el módulo analógico o, si ya está instalado, puede inyectar una corriente tal que corresponda al valor de alarma deseado.

Note que la escala a valores de ingeniería son tales que 0 - 4095 en el valor de entrada corresponde a 32 hasta 600 grados F.

Por lo tanto, 489 grados F en V7000 corresponden a un valor de 3300 en V5000 y para llegar a 490 se debe colocar 3302 en V5000.

De la misma forma, 529 grados F en V7000 corresponden a un valor de 3590 en V5000 y para llegar a 530 se debe colocar 3591 en V5000.

La figura siguiente muestra las etapas para generar reportajes en Hyperterminal, como se muestra en la página K-62.



	Element	Status	Edits	
1	V5000	3592	3300	Primera etapa
2	V7000	489		
3				Segunda etapa
4	V5000	3592	3302	
5	V7000	490		Tercera etapa
6				
7	V5000	3592	3590	Cuarta etapa
8	V7000	529		
9				
10	V5000	3592	3592	
11	V7000	530		
12				
13				
14				
15				
16				
17				

Etapas 1- EL PLC tiene un valor de temperatura en V7000 de 489 grados

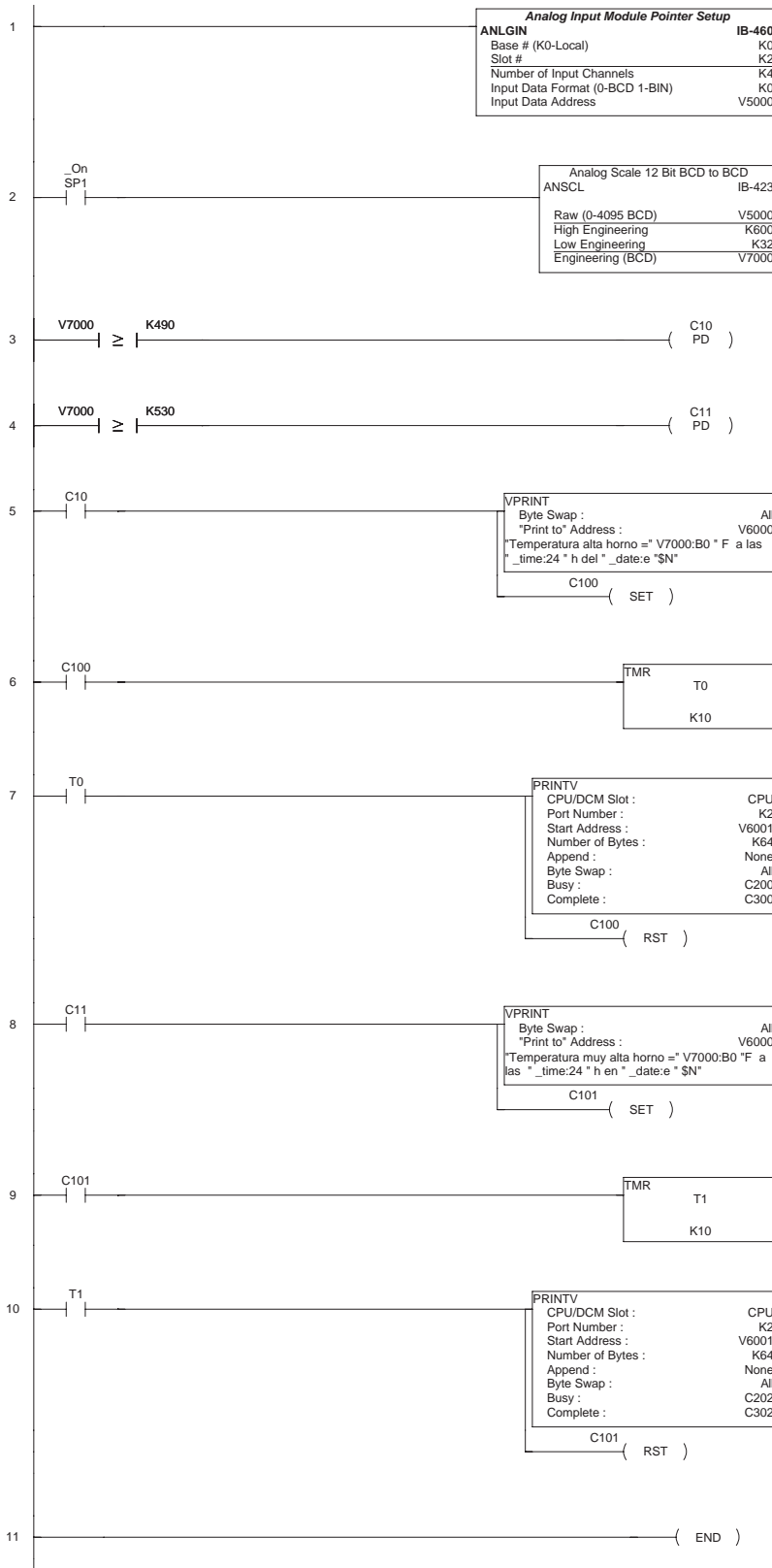
Etapas 2- EL PLC tiene un valor de temperatura en V7000 de 490 grados... El programa genera la primera línea

Etapas 3- EL PLC tiene un valor de temperatura en V7000 de 529 grados

Etapas 4- EL PLC tiene un valor de temperatura en V7000 de 490 grados... El programa genera la segunda línea

Note que el programa genera el comando solamente en la transición de OFF para ON de C10 o C11.

Apéndice K: Introducción a comunicaciones seriales



Este renglón configura el módulo de señales analógicas de entradas en la ranura 2, asignando las memorias V5000 hasta V5003 para tener el valor bruto entre 0- 4095

Este renglón coloca el valor bruto en valores de grados F. El rango es 32 hasta 600 grados F.

C10 es un bit que se hace ON por un barrido del PLC cuando la temperatura pasa de 489 a un valor superior

C11 es un bit que se hace ON por un barrido del PLC cuando la temperatura pasa de 529 a un valor superior

La instrucción VPRINT coloca el texto ASCII en la memoria V6000 y adyacentes. C100 se activa.

C100 inicia el temporizador T0, para contar 1 segundo

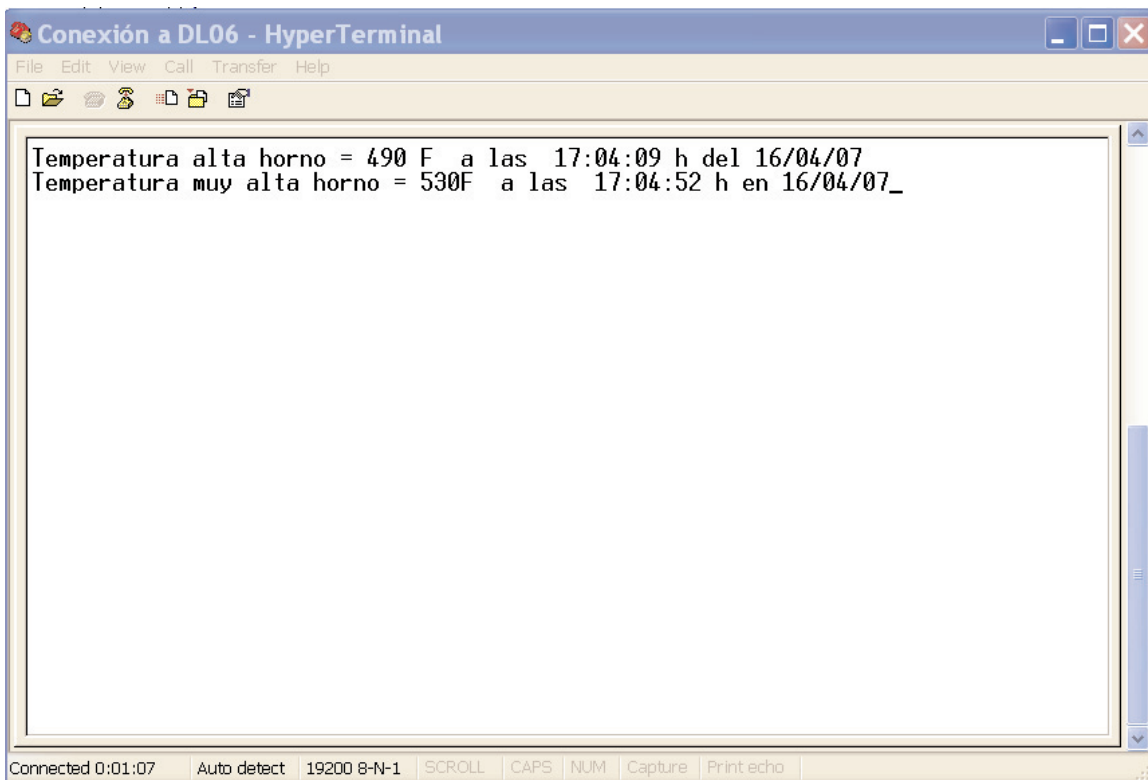
Al final de 1 segundo, se activa T0 lo que inicia la transmisión del contenido en V6000 y adyacentes para el aparato receptor. C100 se hace OFF

La instrucción VPRINT coloca el texto ASCII en la memoria V6000 y adyacentes. Re-escribe datos sobre lo anterior. C101 se activa.

C101 inicia el temporizador T1, para contar 1 segundo

Al final de 1 segundo, se activa T1 lo que inicia la transmisión del contenido en V6000 y adyacentes para el aparato receptor. C101 se hace OFF

K



K

Note lo siguiente:

- La transmisión de datos solamente ocurre en la transición de 489 para 490 grados , en un caso y de 529 para 530 en otro caso.
- Los dos eventos no suceden simultáneamente.
- La información puede ser enviada directamente a una impresora.
- Esta herramienta de Windows también puede ser usada para otros tipos de transferencia de datos.

Recuerde que ésto es solamente una simulación. Normalmente sucederá una alarma solamente en un minuto dado.

Un ejemplo de la instrucción AIN se encuentra el el capítulo 5.

A

Acceso a funciones AUX con <i>DirectSOFT</i>	A-3	Atendiendo los aparatos periféricos	3-21
Acceso a funciones AUX con programador 1	A-3	Aterramiento equipotencial	J-6
Activando generador de Ramp/Soak	8-63	Aterramiento interno del gabinete	J-5
Actualización de la hora y fecha,	3-15	AUX 2* - Operaciones de programa ladder	A-4
Actualizando hora y fecha, relevadores especiales, memorias dedicadas	3-21	AUX 21 Verifique Programa	A-4
Administrando texto ASCII	5-208	AUX 22 Cambie Referencia	A-4
Aislamiento de una red de comunicación	J-7	AUX 23 Suprimir partes del programa	A-4
Ajustando el Bias	8-11	AUX 24 Borre el programa	A-4
Alarma de tasa de cambio	8-39	AUX 3* - Operaciones de memoria V	A-4
Alarmas de desvío	8-38	AUX 31 Limpia la memoria V completa	A-4
Algunos pasos de verificación rápidos	9-8	AUX 4* - Configuración de entradas y salidas	A-4
Aliases o apodos del PLC DL06	3-31	AUX 41 Muestra la configuración de E/S	A-4
Apéndice A: Funciones auxiliares		AUX 5* - Configuración de la CPU	A-5
Apéndice B: Códigos de error del PLC DL06		AUX 51 Modifique el nombre del programa	A-5
Apéndice C: Tiempos de ejecución de cada instrucción		AUX 53 Tiempo de barrido del PLC	A-5
Apéndice D: Relevadores especiales		AUX 54 Initialize Scratchpad	A-5
Apéndice E: Entradas de alta velocidad y salidas de tren de pulsos		AUX 55 Configure el watchdog Timer	A-5
Apéndice F: Memorias del PLC		AUX 56 CPU Network Address	A-5
Apéndice G: Tabla ASCII		AUX 57 Configure rangos retentivos	A-6
Apéndice H: Pesos de los componentes		AUX 58 Pruebe las operaciones	A-6
Apéndice I: Sistemas numéricos		AUX 59 Bit Override	A-6
Apéndice J: Directivas de la Unión Europea (CE)		AUX 5B Configuración HSIO	A-7
Apéndice K: Introducción a comunicaciones seriales		AUX 5D Seleccione modo de barrido del PLC	A-7
Apoyo técnico, como obtenerlo	1-2	AUX 6* - Configuración del D2-HPP	A-8
Aprobación de sitios peligrosos Clase 1, Div 2	2-4	AUX 61 Muestre el no. de versión	A-8
Aprobaciones de agencias reguladoras	2-9	AUX 62 Zumbador activado o desactivado	A-8
Áreas de memoria transferibles	A-8	AUX 65 Diagnósticos	A-8
Asignando contadores en etapas	6-6	AUX 7* - Operaciones con EEPROM	A-8
Asuntos claves en cada capítulo	1-3	AUX 71 CPU al EEPROM del programador	A-8
Asuntos específicos al PLC DL06	J-9	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

Indice

AUX 82 Destrahe la CPU	A-10	Códigos de error del PLC DL06	B-2
AUX 83 Trabe la CPU	A-10	Códigos de errores del programa	9-5
B			
Bibliografía de control PID	8-81	Colocación de fecha y/o hora	5-198
BLOCK (BLK)	7-25	Colocación de texto desde memoria en LCD	5-203
Borre un programa existente	3-9	Colocación directa de texto en LCD	5-200
Buffer de excepción response MWX	5-205	Colocando datos de memorias V en LCD	5-198
Búsqueda de fallas de ruido	9-10	Comienzo rápido	1-6
Búsqueda de problemas de entradas y salidas	9-8	Como cambiar modo PID desde un programa	8-55
Búsqueda de problemas del modo 10	E-23	Como hacer que el tambor vuelva al comienzo	6-11
Búsqueda de problemas en el modo 20	E-37	Comparación booleana	5-7
Búsqueda de problemas para el Modo 30	E-62	Compensación de desvíos de datos	8-29
C			
Cables blindados dentro de gabinetes	J-7	Comunicación con periféricos y forzar E/S	3-14
Cables de señales análogas y RS232	J-7	Comunicación con ASCII non sequence	K-51
Cables multinodo	J-7	Comunicación con <i>DirectNET</i>	K-7
Cálculo de valores prefijados	E-13	Comunicación con K-sequence	K-7
Cambiando fecha y hora en visor LCD	10-14	Comunicación con MODBUS RTU	K-29
Cambiando los datos del acumulador	5-53	Comunicación de la barra de la CPU	3-21
Cambiando mensajes originales de pantalla	10-25	Comunicación por la barra de la CPU	3-15
Cambiando modos en el PLC DL06	3-7	Comunicaciones desde un programa ladder	4-60
Cambie el valor de posición	E-49	Comunicaciones y cables blindados	J-6
Características de la instrucción de etapas	7-6	Concepto de E/S drenadoras/ surtidoras	2-15
Características del PLC DL06	1-4	Concepto de transmisión de datos seriales	K-5
Características principales de la CPU DL06	4-2	Concepto de instalación con PLC's	11-18
Características principales	8-2	Concepto del terminales "comunes"	2-16
Cargas de salidas de estado sólido	2-17	Conceptos de procesamiento paralelo	7-19
Causas posibles de problemas en el PLC	9-8	Condiciones de cambio de modo PID	8-56
Circuitos combinación	5-7	Conectando aparatos de interfase de operación	2-14
Circuitos dedicados de E/S de alta velocidad	E-3	Conectando aparatos de programación	2-14, 3-5
Códigos de caracteres ASCII	10-27	Conectando E/S a aparatos de estado sólido	2-17
Códigos de error del perfil trapezoidal	E-62	Conectando a redes MODBUS y <i>DirectNET</i>	4-48
Códigos de error del PLC DL06	9-4	Conexión del PLC a un PC con <i>DirectSOFT</i>	11-5
		Conexión de entradas de un accionamiento	E-40
		Conexión de salidas de un encoder	E-25
		Configuración INT temporizada / entrada X	E-66
		Configuración de la salida de control (CV)	8-33
		Configuración de la variable de proceso (PV)	8-29

Indice

Diagr. de cableado de E/S del D0-06DR-D	2-42	Ejemplo 2: Perfil trapezoidal automático con interrupción externa	E-50
Diagrama de cableado del modo 30	E-40	Ejemplo 3: Contador con valor inicial	E-21
Diagrama de cableado del modo 10	E-8	Ejemplo 3: Contadores en cuadratura	E-34
Diagramas de cableado para cada modo HSIO	E-3	Ejemplo 3: Perfil trapezoidal automático con búsqueda de "Home"	E-53
Diagramas de puertos de comunicación serial	3-4	Ejemplo 3: Uso de comparación	11-50
Diagramas eléctricos y especificaciones	2-26	Ejemplo 4: Perfil trapezoidal paso a paso	E-56
Diez pasos para obtener un buen control	8-16	Ejemplo 4: Uso de una interfase de operador	11-51
Dimensiones de los PLCs	2-6	Ejemplo 5: Uso de E/S análogas	11-53
Dirección inicial de datos de valores prefijados	E-11	Ejemplo 6: Uso de comunicaciones seriales	11-55
Dirección remota del valor de referencia (SP)	8-29	Ejemplo 7: Uso de lazo de control PID	11-58
Direcciones de memoria de códigos de error	9-3	Ejemplo de búsqueda con AFIND	5-214
Direcciones de memoria V para las áreas de memoria discretas	3-25	Ejemplo de edición durante modo RUN	9-15
Direcciones discretas y de palabra	3-25	Ejemplo de CMPV	5-217
Directivas de la Unión Europea (EU)	J-2	Ejemplo de contador UDC con contactos de comparación	5-50
Diseñando diagramas de transición de estados	7-3	Ejemplo de contador usando contactos de comparación	5-46
Disponibilidad de las características HSIO	E-2	Ejemplo de contador usando el bit de estado	5-46
Disposición del panel & distancias mínimas	2-7	Ejemplo de data Label	5-188
E			
Efecto del modo del PLC sobre los modos de control PID	8-56	Ejemplo de feedforward	8-73
Ejecución de un programa	3-22	Ejemplo de FilterBV	8-59
Ejemplo 1 modo 40: Interrupción externa	E-67	Ejemplo de instrucción AFIND combinado con instrucción AEX	5-215
Ejemplo 1 modo 50 : Captura de pulso	E-72	Ejemplo de instrucción Fault	5-186
Ejemplo 1: Contador en cuadratura con interrupción	E-30	Ejemplo de longitud variable con AIN	5-214
Ejemplo 1: Contador sin valor prefijado	E-16	Ejemplo de MLS/MLR	5-182
Ejemplo 1: Perfil trapezoidal automático sin interrupción externa	E-48	Ejemplo de MWX	5-203
Ejemplo 1: Triturador de minerales con E/S discretas	11-21	Ejemplo de MWX	5-206
Ejemplo 2 modo 40: Interrupción por tiempo	E-68	Ejemplo de programa de control PWM	8-71
Ejemplo 2: Contador con valores prefijados	E-18	Ejemplo de programa de etapas: Abridor de un portón de garaje	7-10
Ejemplo 2: Contador incremental y decremental con entradas normales	E-32	Ejemplo de programa de etapas: Controlador de lámpara con flip flop	7-8
		Ejemplo programa de interrupción externa	5-185
		Ejemplo de programa con control PID	8-74

Ejemplo de Ramp/Soak con <i>DirectSOFT</i>	8-65	Entre el período de muestreo en PID	8-27
Ejemplo de SWAPBX	5-225	Enumeración de los puntos de E/S	2-12
Ejemplo de temporizador acumulador usando contactos de comparación	5-43	Error de cálculo Overflow o underflow	8-40
Ejemplo de temporizador con contactos de comparación	5-41	Error de programación de alarma en PID	8-40
Ejemplo de uso de contadores	11-50	Errores de programación de Ramp/Soak	8-64
Ejemplo de uso de temporizador acumulador con bits de estado	5-43	Errores fatales del PLC	9-2
Ejemplo de uso de temporizador con los bits de estado	5-41	Errores no fatales del PLC	9-2
Ejemplo de VPRINT combinado con la instrucción PRINTV	5-222	Escogiendo formato unipolar o bipolar	8-28
Ejemplo del contador de etapas usando el bit de estado	5-48	Escribiendo a las salidas	3-17, 3-20
Ejemplo del control de portón de garaje	7-10	Escribiendo el control del modo 10	E-15
Ejemplo modo 60 : Entradas filtradas	E-75	Escribiendo el control del modo 20	E-29
Ejemplo para Comunicación ASCII	K-59	Escribiendo salidas a módulos de especialidad	3-17
Ejemplo para redes <i>DirectNET</i>	K-21	Escribiendo Texto ASCII	5-207
Ejemplos de longitud fija de AIN	5-210	Especificaciones ambientales	2-9
Ejemplos para redes MODBUS RTU	K-51	Especificaciones de la CPU	3-3
Ejercicios para el nuevo programador	11-13	Especificaciones del perfil de movimiento	E-41
El cálculo del complemento de 2	I-10	Esquema de bloque de la instrucción de tambor	6-8
El control feedforward	8-72	Esquema de bloques funcional	E-64
El filtro analógico del PLC DL06	8-57	Estableciendo el tamaño de la tabla de lazos y localización	8-20
El fin de bloque (BEND)	7-25	Estado al energizar de la memorias de tambor	6-9
El salto o JUMP (JMP)	7-22	Estado de bits	10-13
El sistema operativo de la CPU	3-12	Estado de entradas HSIO	D-4
El Tambor de eventos (EDRUM)	6-14	Estado del acumulador	D-3
El tambor de eventos con máscara de palabra en las salidas (MDRMW)	6-21	Estándares de cableado	K-2
Elementos en paralelo	5-6	Estrategias de cableado del sistema	2-13
Eliminación de la acción proporcional, integral o derivativa	8-12	Etapas (o Stage) (SG)	7-21
Enclavamientos múltiples de lectura y escritura	4-67	Etapas,	
Entendiendo los seis modos HSIO	E-4	Agregue una lámpara para iluminar el garaje por un corto tiempo	7-12
Entradas de parámetros de control PID	8-34	Agregue una parada de emergencia	7-14
		Etapas de convergencia (CV) y el salto de convergencia (CVJMP)	7-23
		Etapas inicial (ISG)	7-22
		Etapas de convergencia (CV)	7-19
		Etapas iniciales	7-5

Indice

Examinando las ranuras con LCD	10-8	IBOXES	
Explicación del panel frontal del PLC DL06	2-4	BCD al cuadrado (SQUARE)	5-265
F		Binario al cuadrado(SQUAREB)	5-266
Familiarización con <i>DirectSOFT</i>	11-4	Circuito Push On/Push Off o flipflop (PONOFF)	5-253
Filtro de la variable de proceso PV	8-57	Configuración de CTRIO (CTRIO)	5-330
Filtros de alimentación en CA	J-5	Configuración ECOM100 (ECOM100)	5-272
Formatos de datos de parámetros de lazos	8-28	Configuración de e-mail ECOM100 (ECEMSUP)I	5-286
Forzando puntos de entradas y salidas	9-16	Configuración de IP en ECOM100 (ECIPSUP)	5-290
Forzar con el bit override	9-19	Configuración de módulo Analógico Combo (ANLGCMB)	5-232
Forzar normal con acceso directo	9-18	Configuración de módulo Analógico de entradas (ANLGIN)	5-234
Fuente del mensaje del visor LCD	10-26	Configuración de módulo Analógico de salidas (ANLGOUT)	5-236
Funciones AUX del programador	3-9	Configuración de Restore Default E-mail ECOM100 (ECEMRDS)	5-281
Funciones de control PID en el PLC DL06	8-2	CTRIO Add Entry to End of Preset Table (CTRADPT)IB-1005	5-332
Funciones del conmutador de modo	3-6	CTRIO Clear Preset Table (CTRCLRT)IB-1007	5-335
Funciones lógicas de entradas y salidas HSIO	E-41	CTRIO Configuration (CTRIO)	5-330
G		CTRIO Edit Preset Table Entry (CTREDPT)	5-338
Gabinetes para el PLC DL06	2-6, J-4	CTRIO Edit Preset Table Entry and Reload (CTREDRL)I	5-342
Generador de Ramp/Soak	8-60	CTRIO Initialize Preset Table (CTRINPT)	5-346
Glosario de lazos de control PID	8-79	CTRIO Initialize Preset Table (CTRINTR)	5-350
Glosario de términos del PLC	2-25	CTRIO Load Profile (CTRLDPR)1	5-354
Guía rápida de selección de E/S	1-5	CTRIO Read Error (CTRRDER)	5-357
H		CTRIO Run to Limit Mode (CTRRTL)M	5-359
Histéresis de alarmas de PV	8-39	CTRIO Run to Position Mode (CTRRTPM)I	5-362
Historia de errores en el visor LCD	10-20	CTRIO Velocity Mode (CTRVELO)	5-365
Indicaciones de error de programación en tabla Ramp/Soak (Addr + 35)	8-26		
I			
IBOXES			
Alarma alta/baja-BCD (HILOAL)	5-244		
Alarma alta/baja-inaria (HILOALB)	5-246		
BCD a Real (BCDTOR)	5-256		

IBOXES

CTRIO Write File to ROM (CTRWFTR)	5-368
Doble BCD a Real con coma implicada (BCDTORD)	5-257
ECOM100 Disable DHCP (ECDHCPD)	5-274
ECOM100 Enable DHCP (ECDHCPE)	5-276
ECOM100 Query DHCP Setting (ECDHCPQ)	5-278
ECOM100 Read Gateway Address (ECRDGWA)	5-294
ECOM100 Read IP Address (ECDIP)	5-296
ECOM100 Read Module ID (ECRDMID)	5-298
ECOM100 Read Module Name (ECRDNAM)	5-300
ECOM100 Read Subnet Mask (ECRDSNM)	5-302
ECOM100 RX Network Read (ECRX)	5-316
ECOM100 Write Gateway Address (ECWRGWA)	5-302
ECOM100 Write IP Address (ECWRIP)	5-304
ECOM100 Write Module ID (ECWRMID)	5-310
ECOM100 Write Name (ECWRNAM)	5-312
ECOM100 Write Subnet Mask (ECWRSNM)	5-314
ECOM100 WX Network Write (ECWX)	5-319
Enviar E-mail ECOM100 (ECEMAIL)	5-280
Escala de 12 Bit BCD a BCD (ANSCL)	5-238
Escala de 12 Bit Binario a Binario (ANSCLB)	5-239
Escriba Descripción ECOM100 (ECWRDES)	5-304
Filtro - BCD (FILTER)	5-240
Filtro - Binario (FILTERB)	5-242
Flipflop (PONOFF)	5-253

IBOXES

Función Math - BCD (MATHBCD)	5-258
Función Math - Binaria (MATHBIN)	5-260
Función Math - Real (MATHR)	5-262
Lea Descripción ECOM100 (ECRDDES)	5-292
Move Double Word (MOVED)	5-255
Move Single Word (MOVEW)	5-254
NETCFG Network Configuration (NETCFG)	5-322
Network RX Read (NETRX)	5-324
Network WX Write (NETWX)	5-327
One Shot (ONESHOT) I	5-252
Real a BCD con coma implicada y redondeo (RTOBCD)	5-263
Real to Double BCD con coma implicada y redondeo (RTOBCDD)	5-264
Square Real (SQUARER)	5-267
Suma de números BCD (SUMBCD)	5-268
Suma de números Binarios (SUMBIN)	5-269
Suma de números Reales (SUMR)	5-270
Temporizador Off Delay (OFFDTMR)	5-248
Temporizador On Delay (ONDTMR)	5-250
Indicaciones de error de del control PID	8-6
Indicador CPU	9-7
Indicador PWR	9-6
Indicador RUN	9-7
Indicadores de bit override	9-19
Indicadores LED de la CPU	9-6
Información de configuración de la CPU	3-5
Información de memoria del PLC para el visor LCD	10-22
Inicializando la memoria del sistema	3-9
Inicializando salidas del tambor	6-11
Instrucción LCD del PLC DL06	10-26
Instrucción END	5-5
Instrucciones aritméticas	5-86

Indice

Instrucciones ASCII	5-207	L	
Instrucciones booleanas inmediatas	5-9		
Instrucciones de acción inmediata	5-32	La indicación de estado del PLC	3-6
Instrucciones de comparación booleanas	5-26	La instrucción DRUM	6-12
Instrucciones de control de la CPU	5-173	La pantalla Stage View en <i>DirectSOFT</i>	7-18
Instrucciones de control de programa	5-175	La tabla Ramp/Soak en control PID	8-61
Instrucciones de conversión de números (Acumulador)	5-127	La técnica de transición del flujo de potencia	7-18
Instrucciones de fecha y hora	5-171	Las entradas de control del tambor	6-10
Instrucciones de funciones transcendentales	5-118	Las etapas (Datos tipo S)	3-28
Instrucciones de interrupción	5-183	Las instrucciones de salto JMP, SET y RESET	7-7
Instrucciones de mensajes	5-186	Las interrupciones y programa en HSIO	E-65
Instrucciones de MODBUS RTU	5-201	Lazos PID de acción directa e inversa	8-14
Instrucciones de operación con bits	5-120	Leyendo entradas	3-14, 3-20
Instrucciones de RLLPLUS (Etapas)	7-21	Leyendo Texto ASCII	5-207
Instrucciones de tablas	5-141	Limitación de la ganancia derivativa	8-35
Instrucciones de temporizadores, contadores y Shift Register	5-39	Límites de aislación del PLC	2-13
Instrucciones especiales	9-12	Límites del valor de referencia (SP)	8-29
Instrucciones generales de montaje	2-6	Llamada de bloque (BCALL)	7-25
Instrucciones lógicas (Acumulador)	5-69	Localización de la tabla Ramp/Soak	8-26
Instrucciones RLL del DL06	5-1	Los indicadores de bit tabla Ramp/Soak	8-63
Interconexión a las entradas de conteo	E-8	Los lazos en cascada en el PLC DL06	8-68
Introducción a comunicaciones seriales	K-2	Los relevadores especiales (Datos tipo SP)	3-28
Introducción a estados de procesos	7-3	M	
Introducción a MODBUS RTU	K-29	Mantenimiento del sistema de hardware	9-2
Introducción a la programación por etapas	7-2	Mantenimiento normal	9-2
Introducción a lazos en cascada	8-67	Manual de instalación DA-EU-M	J-4
Introducción a Ramp/Soak	8-60	Manuales complementarios	1-2
Introducción a sistemas numéricos	J-2	Mapa de bits de control del estado de etapas	3-34
Introducción al capítulo 11	10-2	Mapa de bits de entradas X o salidas Y	3-33
Introducción al control PID con el PLC DL06	8-6	Mapa de bits de estado de contadores	3-38
Introducción al visor LCD del DL06	10-2	Mapa de bits de estado de temporizadores	3-38
		Mapa de bits de relevadores de control	3-36
		Mapa de bits de Remote I/O	3-39
		Mapa de memoria del DL06	3-32
		Mapa de memoria	3-25

Mejorando el tiempo de respuesta	3-19	Modos de funcionamiento del control PID	8-14
Memoria de palabra (Datos tipo V)	3-28	N	
Memoria de selección de velocidad o de perfil	E-43	Navegación por el menú del visor LCD	10-5
Memoria del sistema DL06	3-39	Necesidad de diagramas de estado	7-3
Memoria reservada del visor LCD	10-23	Nemotécnicos de la instrucción DRUM con el programador D2-HPP	6-16
Memoria V	3-24	No Salto (NJMP)	7-22
Memorias de bits	C-2	Número de Elementos MWX	5-205
Memorias de datos y memorias para configuración del PLC	I-3	Número en representación Gray	I-7
Memorias del PLC DL06	F-2	Números BCD	3-24
Menú 2, M2:SYSTEM CFG. LCD	10-8	Números hexadecimales	3-24
Menú 3, M3:MONITOR, LCD	10-10	O	
Menu 4, M4 : CALENDAR R/W, LCD	10-14	Operación como maestro con MRX y MWX	K-50
Menu 5, M5 : PASSWORD R/W, LCD	10-17	Operación de la CPU	3-12
Menú 6, M6 : ERR HISTORY, LCD	10-20	Operación de perfil trapezoidal automático	E-47
Menu 7, M7 : LCD TEST&SET, LCD	10-21	Operación de esclavo en una red <i>DirectNET</i>	K-12
Métodos de cableado con salidas a relevador	2-19	Operación de esclavo en una red MODBUS	K-35
Métodos de cableado de entradas CC	2-22	Operación de maestro en una red MODBUS	K-41
Métodos de cableado de E/S HSIO	2-24	Operación del control PID	8-9
Métodos de cableado de salidas CC	2-23	Operación del perfil de velocidad	E-59
Métodos de programación del PLC	1-4	Operación del perfil trapezoidal paso a paso	E-55
Modificaciones del programa durante el modo RUN	9-14	Operaciones de carga y copia con el acumulador y Stack	5-52
Modifique diagramas de bloque y de estado	7-12	Organización del programa	7-15
Modo 10: Contador	E-7	P	
Modo 20: Contador incremental/decremental	E-24	Palabra de supervisión de modo/alarmas	8-11, 8-25
Modo 30: Salidas de tren de pulsos	E-38	Parada de emergencia	2-3
Modo 40: Interrupciones de alta velocidad	E-64	Parada normal del sistema	2-3
Modo 50: Entrada de captura de pulso	E-69	Parámetros de constante de tiempo del filtro de entrada en HSIO	E-73
Modo 60: Entradas discretas con filtro	E-73	Parámetros de interrupción por tiempo	E-66
Modo de operación durante la energización del PLC	3-7	Parámetros de sincronía de captura de pulso	E-69
Modo de velocidad del algoritmo PID	8-12		
Modo normal de fábrica	E-5		
Modo Program	3-13		
Modo Run	3-13		

Indice

Parámetros sincronía de interrupción externa	E-66	Programación de entradas y salidas análogas con PLCs DL	11-15
Parámetros del sistema y de datos originales de fábrica (tipo de datos V)	3-29	Programador portátil D2-HPP	1-5
Partida de una máquina y búsquedas de fallas del programa	9-11	Programando el control Ramp/Soak en lógica ladder	8-65
Pasos para escribir un programa por etapas	7-9	Prolongando la vida útil de los contactos de un relevador	2-21
Pasos para diseñar un sistema	1-10	Propósito de este manual	1-2
Pautas básicas de instalación de EMC	J-4	Propósito de las funciones auxiliares	A-2
Perfil trapezoidal automático	E-43	Protección de fusibles para la alimentación de energía al PLC	2-10
Perfil trapezoidal paso a paso	E-44	Protección de Reset Windup	8-10
Planeando las rutas de cableado	2-11	Protección por fusibles de los circuitos de entradas y de salidas	2-12
Planee con la seguridad en mente	2-2	Protocolos de comunicaciones disponibles	K-3
Preguntas y respuestas acerca de la programación por etapas	7-27	Prueba de lazo abierto	8-42
Preguntas y respuestas sobre el PLC DL06	1-12	Puntos de entradas (Datos tipo X)	3-26
Prioridad de la pantalla LCD	10-4	Puntos de salidas (Datos tipo Y)	3-26
Probando el perfil de ejemplo de Ramp/Soak	8-66		
Problemas de comunicaciones	9-7	Q	
Problemas eléctricos de ruido	9-10	Que es memoria V	11-2
Procedimiento de sintonía automática	8-47	Qué hacen los bits de etapas	7-6
Procedimiento de sintonía manual	8-44		
Procedimientos alternativos de sintonía manual	8-46	R	
Proceso de 2 estados en etapas	7-3	Ramas en paralelo que se unen en serie	5-7
Proceso de cuatro estados en etapas	7-8	Ramp/Soak	8-41
Procesos de convergencia en etapas	7-19	Rangos de direcciones de esclavo MRX 202	5-
Procesos paralelos en etapas	7-19	Rangos de direcciones de esclavo MWX	5-205
Programa Ejemplo: Perfil de velocidad HSIOE-60		Rangos de direcciones de memoria del maestro MWX	5-205
Programa ejemplo para colocar el mensaje de pantalla por defecto	10-25	Rangos de números de punto flotante	I-7
Programa ejemplo: alarma con datos embutidos de la memoria V	10-29	Recursos del PLC	3-23
Programa ejemplo: Alarma mostrada con la fecha y hora que ocurrió	10-28	Reducción del ruido eléctrico	9-10
Programa ejemplo: Texto de alarma con datos embutidos de la memoria V	10-30	Reglas de uso de la etapa de convergencia	7-20
Programa asociado a un lazo del control PID	8-74		
Programación con <i>DirectSOFT</i> en Windows	1-4		

Relevador de supervisión de comunicaciones en cada ranura del PLC	D-4	Secuencias de salidas en etapas	6-3
Relevador de supervisión de comunicación	D-4	Seguridad general	J-3
Relevador especial de salida de pulsos HSIO	D-4	Selección de modo de errores	8-35
Relevadores de control (Datos tipo C)	3-26	Selección de modo de operación HSIO	E-4
Relevadores de estado de la CPU	D-2	Selección de tipo de perfil	E-45
Relevadores de partida y de pulsos de tiempo definido	D-2	Sensores de entrada de estado sólido	2-17
Relevadores especiales de comparación	E-12	Señal en cuadratura del encoder	E-25
Relevadores especiales (SP) que corresponden a códigos de error	9-3	Síntoma: El contador cuenta en la dirección errada	E-37
Relevadores especiales de detección del contador 1 en el Modo 10	D-5	Síntoma: El contador cuenta hacia arriba y abajo pero no hace reset	E-37
Relevadores especiales de detección del contador 2 en el Modo 10	D-6	Síntoma: El contador cuenta pero los valores prefijados no funcionan	E-23
Relevadores especiales del PLC DL06	D-2	Síntoma: El contador cuenta pero no vuelve a cero	E-23
Relevadores especiales para módulos opcionales	D-4	Síntoma: El contador no cuenta	E-23, E-27
Relevadores especiales y valores prefijados	E-9	Síntoma: El motor de paso a paso no gira	E-62
Relevadores especiales y valores prefijados del modo 20	E-27	Síntoma: El motor gira en la dirección equivocada	E-63
Renglones simples	5-5	Sintonía automática de lazo abierto	8-48
Representación de una tabla del tambor	6-3	Sintonía automática de lazo cerrado	8-49
Representación del complemento de 2	I-9	Sintonizando lazos de control PID	8-42
Resolviendo algoritmos de lazos PID	3-16	Sintonizando lazos en cascada	8-69
Resolviendo programas de aplicación	3-16	Sistema de numeración octal	3-25
Respuesta de entradas y salidas normal	3-18	Sistema decimal	I-2
Respuesta de excepción en MODBUS	K-52	Sistema numérico BCD (Binary coded decimal)	I-4
Respuesta máxima de entradas y salidas	3-18	Sistema numérico binario	I-3
Resumen de las particularidades de los PLCs		Sistema numérico hexadecimal	I-5
<i>Direct</i> LOGIC	11-15	Sistema numérico octal	I-2
Retiro del bloque de terminales	2-5	Sistema numérico real de punto flotante	I-5
		Sistemas numéricos del PLC	3-23
S		Soluciones de control de movimiento	E-2
Salidas en el medio del renglón	5-6	Stack booleano	5-8
Salto de convergencia (CVJMP)	7-20	Sufijos de formato de datos para datos de memoria V mostrados en el visor	10-22
Secuencia de teclado en el programador usada para probar un punto de salida	9-9	Sufijos de formatos de datos para datos embutidos de memoria V	5-199

Verificación de sintaxis	9-11
Versiones alimentadas por corriente continua	J-8

