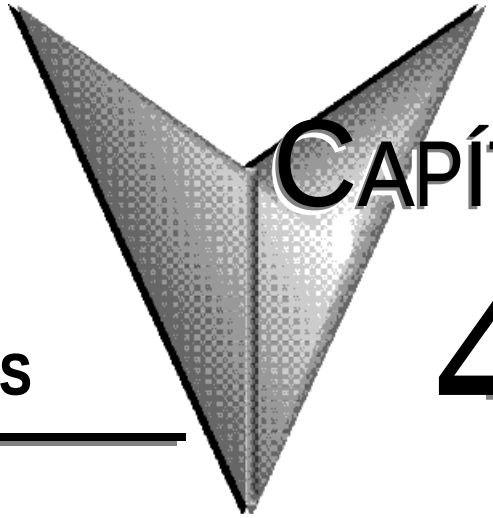
PROGRAMACIÓN LADDER PARA COMUNICACIONES



CAPÍTULO 4

En este capítulo...

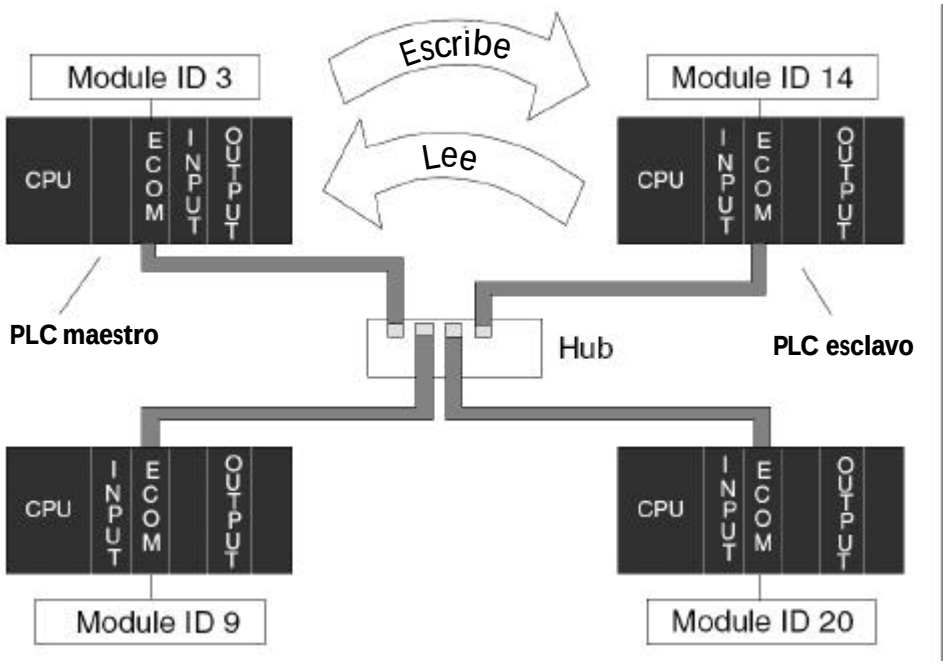
Comunicaciones PLC a PLC

Este capítulo le muestra el desarrollo de un programa de lógica ladder (RLL) para permitir a un PLC comunicarse con otro PLC. Para el programador experimentado de PLCs *Direct*LOGIC, los programas de comunicación presentados en este capítulo serán simples de seguir. Si usted nunca ha programado un PLC *Direct*LOGIC, usted puede necesitar leer y entender el manual de usuario de programación del software *Direct*SOFT32 y el manual de usuario para el PLC que está usando, para información adicional.

NOTA: Los programas descritos en este capítulo no se usan para la comunicación entre una PC y un PLC. Para comunicaciones PC a PLC, vea por favor la documentación del producto para el software de la PC que usted está usando. Si usted está utilizando nuestro software de servidor, el manual que usted necesitará es el manual de usuario de KEPDirect para PLCs.

Cómo se utiliza RLL para comunicaciones

El software de programación *Direct*SOFT32 proporciona instrucciones de lectura y escritura (RX/WX) para comunicación PLC a PLC a través de una red. Las instrucciones lectura y escritura son parte del programa de la lógica ladder que funciona en la CPU de inicio de las transacciones de comunicación, o PLC maestro. Estas instrucciones le dicen a la CPU como enviar un mensaje a través de la red de Ethernet a un PLC que responde, o esclavo. El módulo ECOM es el punto de conexión a la red para cada PLC. Las lecturas y escrituras del PLC que inicia la transacción encuentra su destino por la **identificación del módulo** del módulo ECOM del PLC que responde. Vea el capítulo 2 y 3 para información de como asignar las identificaciones del módulo.



En la figura de arriba, el PLC que inicia la transacción envía el mensaje de lectura o escritura al módulo ECOM del PLC que responde que se identifica como "**módulo ID 14**". El PLC que responde procesa el mensaje. Cualquiera de los PLCs podría iniciar la comunicación con cualquier otro de los PLCs.

Instrucciones de la Red

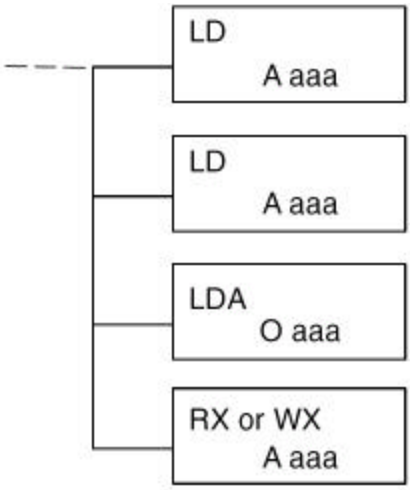
Instrucciones de lectura y escritura RX y WV

Las instrucciones de lectura (RX) y escritura (WX) son usadas por el PLC que inicia las transacciones de comunicación para leer un bloque de datos desde otro PLC a para escribir hasta otro PLC. Para ejecutar estas funciones, las instrucciones deben ser precedidas en el programa ladder por dos instrucciones Load (LD) y una Load Address (LDA).

Las instrucciones LD y LDA cargan parámetros de comunicación en el **acumulador** y el primer y segundo nivel del stack del acumulador. La instrucción RX o WX toma estos parámetros del stack y del acumulador y prepara los datos para ser enviados a través de la red. Si usted necesita saber más sobre la función del acumulador y del stack del acumulador, vea al manual de usuario para su PLC.

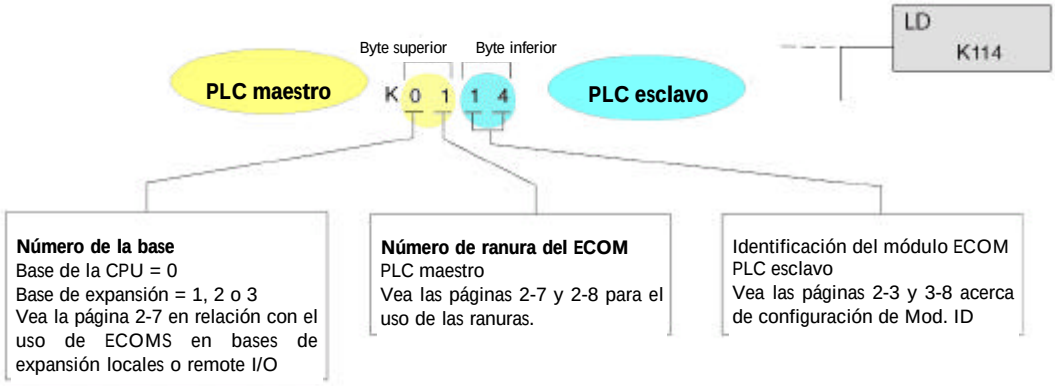
Construyendo una rutina de lectura (RX) o escritura (WX)

Para comunicaciones a través de una red Ethernet, usted construye las instrucciones de lectura (RX) y escritura (WX) en una **rutina** que necesita tener cuatro instrucciones que Ud puede ver en la figura de la derecha. La función de cada una de estas instrucciones es explicada más abajo o en la página siguiente. Ellas debe ser usadas en la secuencia mostrada.



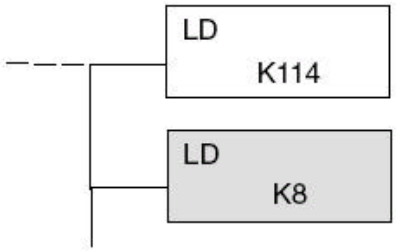
La primera instrucción LD acepta una constante o una variable. Use una "K" para designar el número como constante. Use una "V" si usted está entrando la dirección de un registro o memoria. El contenido de esa memoria realiza la misma función que la constante mostrada abajo. Por ejemplo, usted podría utilizar V2000 en lugar de K114. Si el contenido de V2000 es el número "114," la función sería igual. Usar una variable permite cambio de parámetros cuando el programa está funcionando. Se recomienda, sin embargo, usar una constante cuando sea posible.

La primera instrucción LD

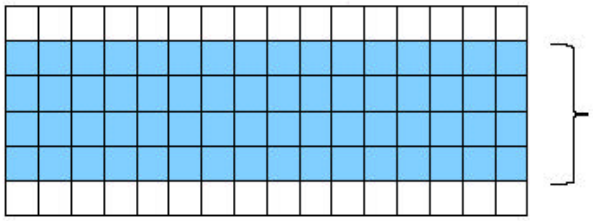


La segunda instrucción LD

La segunda instrucción LD determina la longitud del bloque de datos que se transmitirá durante la transacción de lectura. Esta instrucción también aceptará dos tipos de datos. Use una "K" para designar el número como constante. Use una "V" si usted está entrando la dirección de un registro o memoria. Para datos de memoria de palabra, usted debe utilizar un múltiplo de dos bytes entre 2 y 128. Para datos de memoria de bit, usted puede usar cualquier múltiplo de un byte entre 1 y 128. Para más información sobre la dirección de memoria de palabras y de bit, vea la página 4-6.



Memoria del PLC

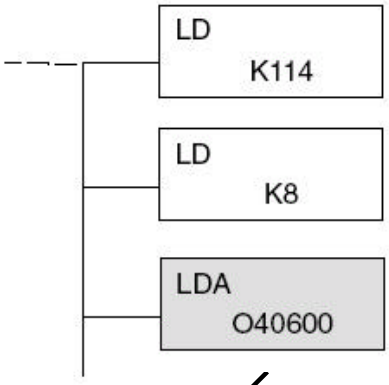


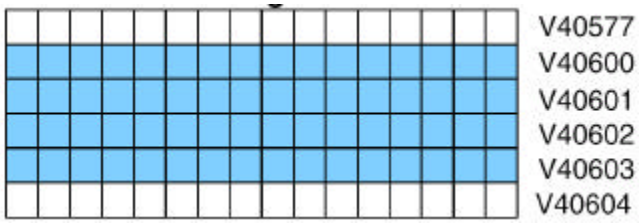
La instrucción LDA

La instrucción LDA especifica la dirección de memoria V del registro inicial de la memoria en el PLC maestro. El bloque de datos que se transmitirá comenzará en esta dirección y se extenderá por el número de bytes especificados en la instrucción precedente LD. El prefijo "O" indica que esto es un número octal. Substituya simplemente la letra "O" por el "V" en la designación de la memoria V. Por ejemplo, V40600 se convierte en O40600.

Las instrucciones de lectura copian el bloque de datos de la memoria del PLC que responde a la memoria del PLC maestro.

Las instrucciones de escritura copian el bloque de datos de la memoria del PLC que inicia la transacción (maestro) a la memoria del PLC que responde.

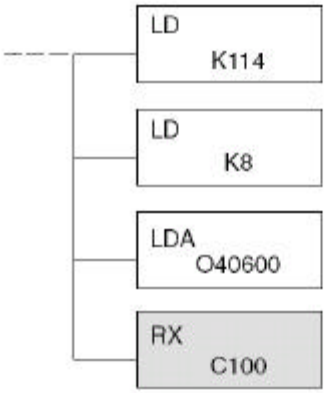


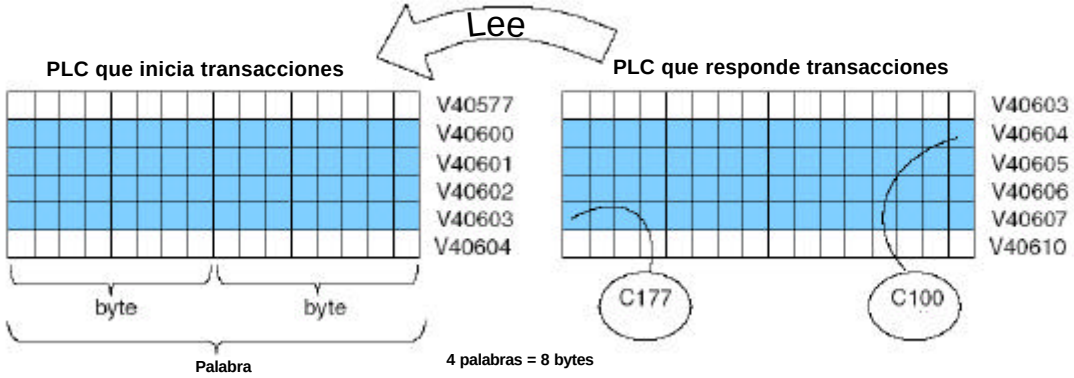


La instrucción de lectura (RX) especifica la posición de memoria *que se leerá* en el PLC que responde.

Se lee un bloque de datos que comienza en la posición de memoria especificada y se extiende por el número de bytes especificados en la segunda instrucción LD.

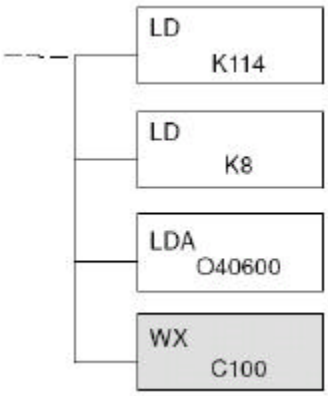
En este ejemplo, el bloque de datos de ocho bytes que comienza en C100 y que termina en C177 en el PLC que responde, o el esclavo, se lee (copia), en la memoria del PLC maestro, que comienza en V40600.

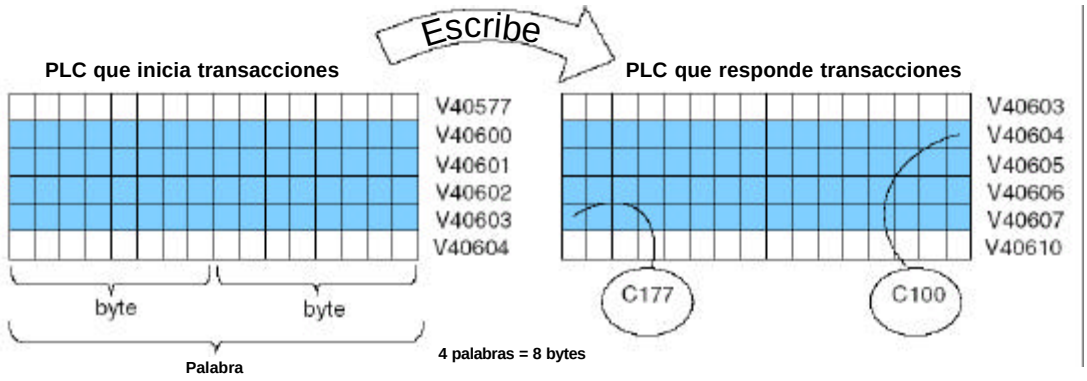




La instrucción de escritura (WX)

La instrucción de escritura (WX) especifica la posición de memoria que se escribirá al PLC que responde. Se escribe un bloque de datos que comienza en la posición de memoria especificada y se extiende por el número de bytes especificados en la segunda instrucción LD. En el ejemplo, el bloque de datos de 8 bytes que comienza en V40600 y que termina en V40603 en el PLC maestro, se escribe (se copia) en la memoria del PLC que responde, que comienza en C100 y que termina en C177.





Dirección de los diversos tipos de memoria

Ciertos tipos de datos son intrínsecamente de 16 bits de longitud, por ejemplo, los valores corrientes de temporizadores y contadores. Otros tipos de datos son 1 bit de longitud, por ejemplo: entradas y salidas discretas. Los datos de longitud de palabra y de longitud de bit son relacionados a una memoria de palabra, también conocida como memoria V, que permite que usted trate cualquiera de los diversos tipos de memoria como palabras de 16 bits.

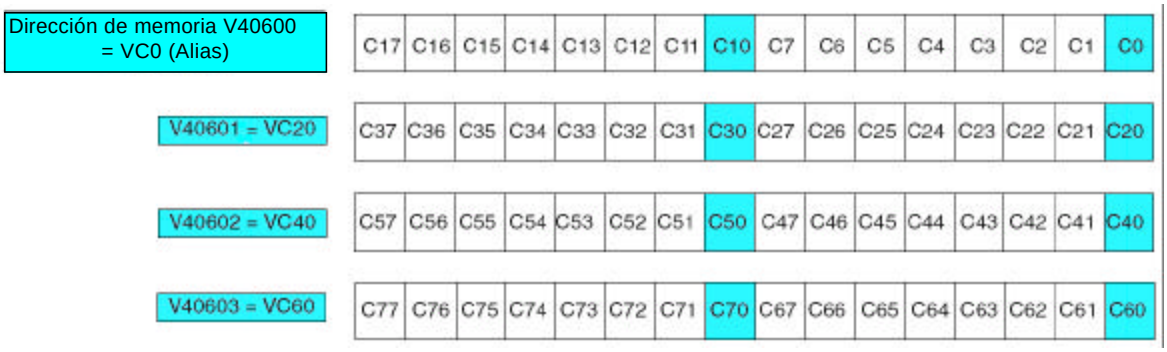
La memoria de bits

La memoria de bits se puede tratar en instrucciones de lectura y escritura por el nombre del primer bit de cualquier byte. Si la segunda instrucción LD contiene la constante K8, serán transmitidos ocho bytes. Si usted usa C0 en la instrucción RX o WX, usted transmitirá los ocho bytes de C0 hasta C77.

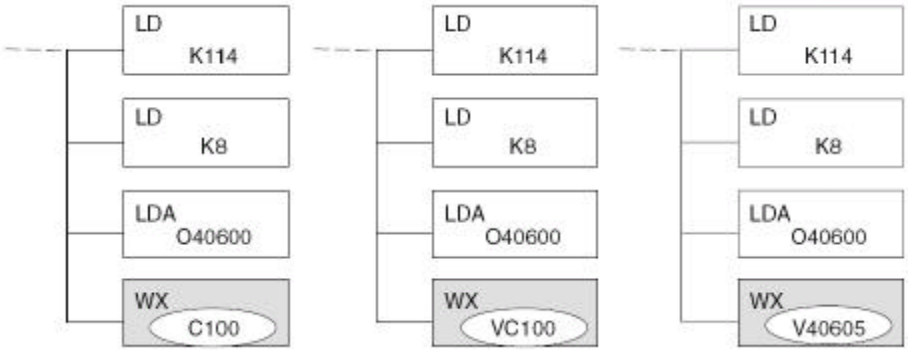
Memoria de palabra y el otro nombre (alias)

En el ejemplo de abajo, V40600 es la designación de la memoria V para los dieciséis bits de C0 hasta C17. Los alias son un substituto conveniente para las designaciones de la memoria V, y puede ser utilizado alternativamente en instrucciones de lectura y escritura. VC0 es el alias para V40600. Cualquiera de estas nomenclaturas corresponden a los mismos 16 bits.

El alias es simplemente el nombre del primer bit en un grupo de dieciséis bits, agregando V como prefijo. Por ejemplo, VC0 representa los 16 bits que comienzan con la memoria C0, memoria de palabra, de bit y los alias todos utilizan el sistema de numeración octal.



Las siguientes rutinas de escritura son todas equivalentes. *Direct*SOFT32 le da la flexibilidad de identificar el área de memoria del PLC que responde de tres maneras diferentes, según se muestra abajo.



*Direct*SOFT32 es flexible

Usted puede tratar diferentes tipos de datos de cualquier convención disponible mostrada en las tablas que siguen. El bloque más grande de datos que pueden ser enviados en sola operación de lectura y escritura es 128 bytes. El bloque más pequeño de datos es un byte para los tipos de memoria de bit y dos bytes, o una palabra para tipos de memoria de palabra. El sistema de numeración octal se utiliza para todas las direcciones en estas tablas.

CPU DL05

CPU DL05			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V177	TA0 - TA177
Valores corrientes de contador	Ninguno	V1000 - V1177	CTA0 - CTA177
Palabras de datos del usuario	Ninguno	V1200 - V7377	Ninguno
Puntos de entradas (vea la nota 1)	X0 - X377	V40400 - V4417	VX0- VX360
Puntos de salidas (vea la nota 1)	Y0 - Y377	V40500 - V4517	VY0- VY360
Relevadores de control	C0 - C777	V40600 - V4617	VC0- VC760
Relevadores especiales	SP0 - SP777	V41200 - V41237	VSP0 - VSP760
Bits de estado de temporizadores	T0 - T177	V41100- V41107	VT0 - VT160
Bits de estado de contadores	CT0 -C T177	V41140- V41147	VCT0 - VCT160
Etapas	S0 - S377	V41000 - V41017	VS0 - VS360

Nota 1 - *Los sistemas DL05 se limitan a 8 entradas discretas y a 6 salidas discretas con el actual hardware disponible, pero existen direcciones de 256 puntos.*

CPU DL06

CPU DL06			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V377	TA0 - TA377
Valores corrientes de contador	Ninguno	V1000 - 1177	CTA0 - CTA177
Palabras de datos del usuario	Ninguno	V400 - V677 V1200 - V7377 V10000 - V17777	Ninguno
Puntos de entradas (vea la nota 1)	X0 - X777	V40400 - V4437	VX0- VX760
Puntos de salidas (vea la nota 1)	Y0 - Y777	V40500 - V4537	VY0- VY760
Relevadores de control	C0 - C1777	V40600 - V4677	VC0- VC760
Relevadores especiales	SP0 - SP777	V41200 - V41237	VSP0 - VSP1760
Bits de estado de temporizadores	T0 - T377	V41100- V41117	VT0 - VT160
Bits de estado de contadores	CT0 -C T177	V41140- V41147	VCT0 - VCT160
Etapas	S0 - S1777	V41000 - V41077	VS0 - VS1760
Remote I/O	GX0 - GX3777 GY0-GY3777	V40000 - V40177 V40200 - V40377	VGX0 - VGX3760 VGY0 - VGY3760

Nota 1 - *Los sistemas DL06 se limitan a 20 entradas discretas y a 16 salidas discretas con el actual hardware disponible, pero existen direcciones de 512 puntos.*

NOTA: La CPU -D2 230 no apoya los módulos ECOM.

CPU D2-240			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V177	TA0 - TA177
Valores corrientes de contador	Ninguno	V1000 - 1177	CTA0 - CTA177
Palabras de datos del usuario	Ninguno	V02000 - V3777 V4000 - V4377	Ninguno
Puntos de entradas	X0 - X477	V40400 - V40423	VX0- VX460
Puntos de salidas	Y0 - Y477	V40500 - V523	VY0- VY460
Relevadores de control	C0 - C377	V40600 - V40617	VC0- VC360
Relevadores especiales	SP0 - SP137 SP540 - SP617	V41200 - V41205 V41226 - V41230	VSP0 - VSP120 VSP540 - VSP600
Bits de estado de temporizadores	T0 - T177	V41100- V41107	VT0 - VT160
Bits de estado de contadores	CT0 -C T177	V41140- V41147	VCT0 - VCT160
Etapas	S0 - S777	V41000 - V41037	VS0 - VS760

CPU D2-250-1			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V377	TA0 - TA377
Valores corrientes de contador	Ninguno	V1000 - 1377	CTA0 - CTA377
Palabras de datos del usuario	Ninguno	V1400 - V7377 V10000 - V17777	Ninguno
Puntos de entradas	X0 - X777	V40400 - V40437	VX0- VX760
Puntos de salidas	Y0 - Y777	V40500 - V40537	VY0- VY760
Relevadores de control	C0 - C1777	V40600 - V40677	VC0- VC1760
Relevadores especiales	SP0 - SP777	V41200 - V41237	VSP0 - VSP760
Bits de estado de temporizadores	T0 - T177	V41100- V41107	VT0 - VT360
Bits de estado de contadores	CT0 -C T177	V41140- V41147	VCT0 - VCT160
Etapas	S0 - S777	V41000 - V41077	VS0 - VS1760

CPU D2-260			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V377	TA0 - TA377
Valores corrientes de contador	Ninguno	V1000 - 1377	CTA0 - CTA377
Palabras de datos del usuario	Ninguno	V1400 - V777 V1400 - V7377 V10000 - V37777	Ninguno
Puntos de entradas	X0 - X1777	V40400 - V40477	VX0- VX1760
Puntos de salidas	Y0 - Y1777	V40500 - V40577	VY0- VY1760
Relevadores de control	C0 - C3777	V40600 - V40777	VC0- VC3760
Relevadores especiales	SP0 - SP137 SP320 - SP777	V41200 - V41205 V41215 - V41234	VSP0 - VSP120 VSP320 -VSP700
Bits de estado de temporizadores	T0 - T377	V41100- V41117	VT0 - VT360
Bits de estado de contadores	CT0 -C T377	V41140- V41157	VCT0 - VCT160
Etapas	S0 - S1777	V41000 - V41077	VS0 - VS1760
Remote I/O	GX0-GX3777 Gy0-GY3777	V40000 - V40177 V40200 - V40377	VGX0 - VGX3760 VGY0 - VGY3760

CPU D4-430			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V177	TA0 - TA177
Valores corrientes de contador	Ninguno	V1000 - 1177	CTA0 - CTA177
Palabras de datos del usuario	Ninguno	V1400 - V7377	Ninguno
Puntos de entradas	X0 - X477	V40400 - V40423	VX0- VX460
Puntos de salidas	Y0 - Y477	V40500 - V40523	VY0- VY460
Relevadores de control	C0 - C737	V40600 - V40635	VC0- VC720
Relevadores especiales	SP0 - SP137 SP320 - SP777	V41200 - V41205 V41215 - V41230	VSP0 - VSP120 VSP320 -VSP600
Bits de estado de temporizadores	T0 - T177	V41100- V41107	VT0 - VT160
Bits de estado de contadores	CT0 -C T177	V41140- V41107	VCT0 - VCT160
Etapas	S0 - S577	V41000 - V41027	VS0 - VS560
Remote I/O	GX0-GX777	V40000 - V40037	VGX0 - VGX760

CPU D4-440			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V377	TA0 - TA377
Valores corrientes de contador	Ninguno	V1000 - 1177	CTA0 - CTA177
Palabras de datos del usuario	Ninguno	V1400 - V7377 V10000 - V17777	Ninguno
Puntos de entradas	X0 - X477	V40400 - V40423	VX0- VX460
Puntos de salidas	Y0 - Y477	V40500 - V40523	VY0- VY460
Relevadores de control	C0 - C1777	V40600 - V40677	VC0- VC1760
Relevadores especiales	SP0 - SP137 SP320 - SP777	V41200 - V41205 V41215 - V41234	VSP0 - VSP120 VSP320 -VSP700
Bits de estado de temporizadores	T0 - T177	V41100- V41117	VT0 - VT360
Bits de estado de contadores	CT0 -C T177	V41140- V41147	VCT0 - VCT160
Etapas	S0 - S577	V41000 - V41077	VS0 - VS1760
Remote I/O	GX0-GX1777	V40000 - V40077	VGX0 - VGX1760

CPU D4-450			
Tipos de Datos	**Memoria de Bit**	**Memoria de Palabra**	**Alias**
Valores corrientes de temporizador	Ninguno	V0 - V377	TA0 - TA377
Valores corrientes de contador	Ninguno	V1000 - 1377	CTA0 - CTA377
Palabras de datos del usuario	Ninguno	V1400 - V7377 V10000 - V37777	Ninguno
Puntos de entradas	X0 - X1777	V40400 - V40477	VX0- VX1760
Puntos de salidas	Y0 - Y1777	V40500 - V40577	VY0- VY1760
Relevadores de control	C0 - C3777	V40600 - V40777	VC0- VC3760
Relevadores especiales	SP0 - SP137 SP320 - SP717	V41200 - V41205 V41215 - V41234	VSP0 - VSP120 VSP320 -VSP700
Bits de estado de temporizadores	T0 - T377	V41100- V41157	VT0 - VT360
Bits de estado de contadores	CT0 -C T177	V41140- V41157	VCT0 - VCT360
Etapas	S0 - S1777	V41000 - V41177	VS0 - VS1760
Remote I/O	GX0-GX3777 GY0-GY3777	V40000 - V40177 V40200 - V40377	VGX0 - VGX3760 VGY0 - VGY3760

Relevadores especiales para comunicaciones

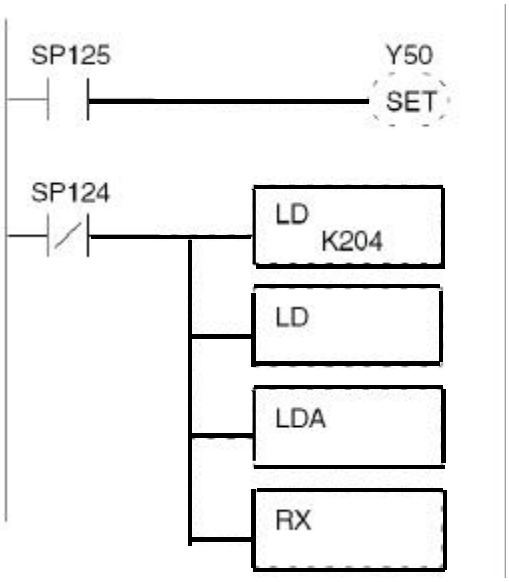
Los PLCs DirectLOGIC proporcionan contactos internos (bits) para supervisar el estado de comunicaciones. Los contactos internos se llaman Special Relays (hay otros relevadores especiales usados para otros propósitos). Hay dos relevadores especiales para cada ranura en la base que aceptará el módulo de ECOM. Los dos relevadores realizan las funciones siguientes:

- **Communication busy** (Comunicación ocupada) - Este bit se enciende cuando el módulo de comunicación está ocupado transmitiendo o recibiendo datos. Usted debe usar este bit, o el contacto del relevador, para evitar el sobreescribir las instrucciones de lectura o de escritura (RX/WX).

- **Error de Comunicación** - Este bit es encendido cuando ocurrió un error en la última comunicación RX o WX. Este error se limpia automáticamente (resetea el bit a cero) cuando se ejecuta otra instrucción RX o WX.

Por ejemplo, los relevadores especiales SP124 y SP125 corresponden a un módulo de ECOM en la ranura 3 de la base del PLC.

Se usa el relevador especial SP125 en el ejemplo para energizar la salida Y50, que indica que ha ocurrido un error de comunicación. Este relevador especial debe aparecer anteriormente en el programa a la instrucción RX o WX porque se hace cero cuando se ejecuta la instrucción de lectura o escritura subsecuente.

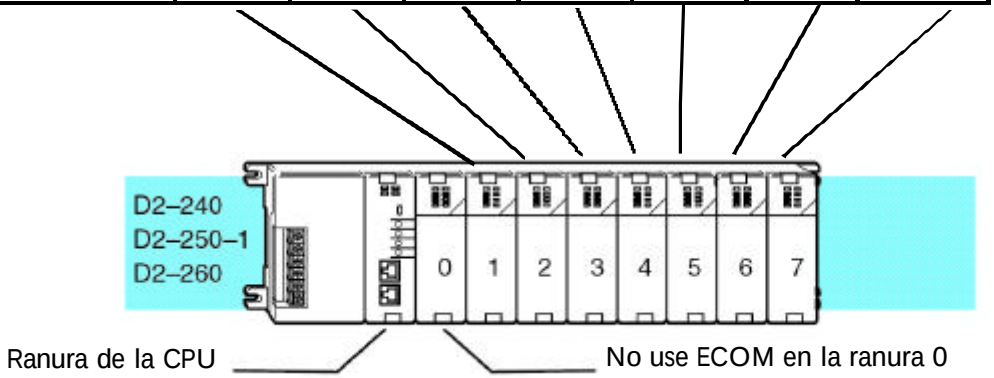


El relevador especial SP124 indica que el módulo ECOM está ocupado. Cuando SP124 está encendido, el contacto normalmente cerrado se abre para evitar ejecutar otra instrucción RX o WX hasta que la última se haya completado. **Se debe usar** el bit **busy** apropiado como contacto NC en cada reglón de la instrucción RX/WX en el programa.

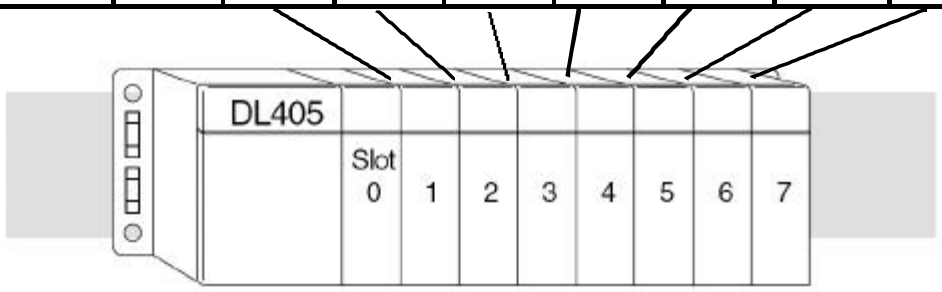
Relevadores especiales del DL05	
Base de CPU	Ranura de opción
Comunicación “busy”	SP120
Error de comunicación	SP121

Relevadores especiales del DL06				
Base de CPU	Ranura 1	Ranura 2	Ranura 3	Ranura 4
Comunicación “busy”	SP120	SP122	SP124	SP126
Error de comunicación	SP121	SP123	SP125	SP127

Relevadores especiales de CPUs DL240, DL250-1 y DL260							
Base de CPU	Ranura 1	Ranura 2	Ranura 3	Ranura 4	Ranura 5	Ranura 6	Ranura 7
Comunicación “busy”	SP122	SP124	SP126	SP130	SP132	SP134	SP136
Error de comunicación	SP123	SP125	SP127	SP131	SP133	SP135	SP137



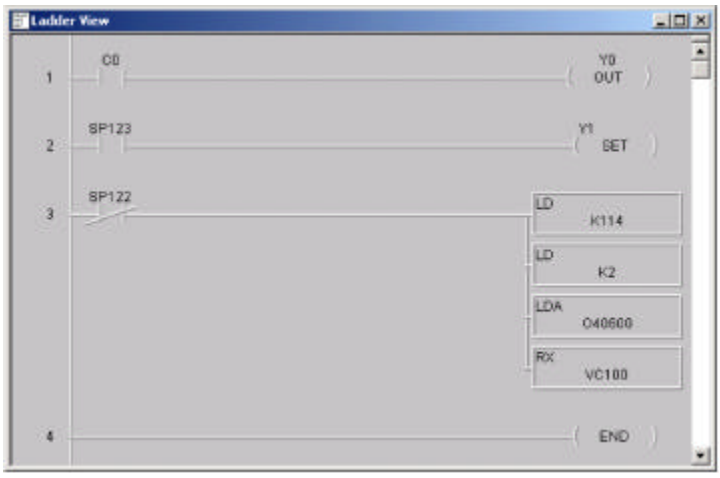
Relevadores especiales de CPUs D4-430 y D4-440								
Base de CPU	Ranura 0	Ranura 1	Ranura 2	Ranura 3	Ranura 4	Ranura 5	Ranura 6	Ranura 7
Comunicación “busy”	SP120	SP122	SP124	SP126	SP130	SP132	SP134	SP136
Error de comunicación	SP121	SP123	SP125	SP127	SP131	SP133	SP135	SP137



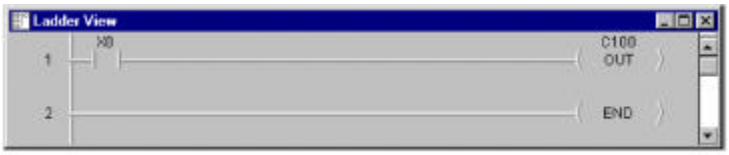
Relevadores especiales de la CPU D4-450								
Base de CPU	Ranura 0	Ranura 1	Ranura 2	Ranura 3	Ranura 4	Ranura 5	Ranura 6	Ranura 7
Comunicación “busy”	SP120	SP122	SP124	SP126	SP130	SP132	SP134	SP136
Error de comunicación	SP121	SP123	SP125	SP127	SP131	SP133	SP135	SP137
Base 1 de expansión	Ranura 1	Ranura 2	Ranura 3	Ranura 4	Ranura 5	Ranura 6	Ranura 7	Ranura 7
Comunicación “busy”	SP140	SP142	SP144	SP146	SP150	SP152	SP154	SP156
Error de comunicación	SP141	SP143	SP145	SP147	SP151	SP153	SP155	SP157
Base 2 de expansión	Ranura 1	Ranura 2	Ranura 3	Ranura 4	Ranura 5	Ranura 6	Ranura 7	Ranura 7
Comunicación “busy”	SP160	SP162	SP164	SP166	SP170	SP172	SP174	SP176
Error de comunicación	SP161	SP163	SP165	SP167	SP171	SP173	SP175	SP177
Base 3 de expansión	Ranura 1	Ranura 2	Ranura 3	Ranura 4	Ranura 5	Ranura 6	Ranura 7	Ranura 7
Comunicación “busy”	SP200	SP202	SP204	SP206	SP210	SP212	SP214	SP216
Error de comunicación	SP201	SP203	SP205	SP207	SP211	SP213	SP215	SP217

Programa con una instrucción de lectura

La pantalla **Ladder View** mostrada abajo es la pantalla de desarrollo de programa en el software de programación *Direct*SOFT32. Este programa de cuatro renglones se explica detalladamente en la página 4-14. Esto es un programa completo aunque su función es muy limitada y solo sirve para entregar el concepto. Hay también un programa de dos renglones que funciona en el PLC que responde, y también se explica en la página 4-14.

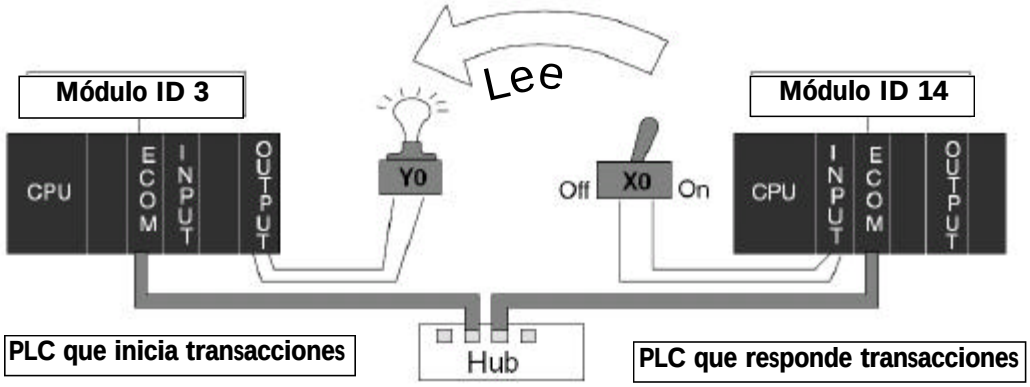


Programa para el PLC que inicia las transacciones



Programa para el PLC que responde las transacciones

Cuando se activa la entrada del interruptor al PLC que responde (esclavo) (en la transición

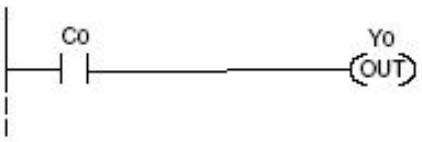


de OFF a ON), el bit C0 en el PLC que inicia las transiciones (maestro) se va de OFF para ON El programa en el PLC maestro hace que Y0 se encienda en respuesta al bit C0.

En estos programas de ejemplo, el PLC que inicia y el PLC que responde las transacciones

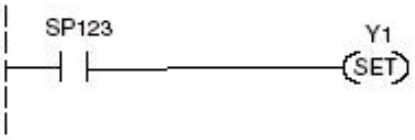
Renglón 1

En nuestro ejemplo, el contacto normalmente abierto rotulado C0 es un relevador interno de control. Cuando C0 está encendido, se energiza la salida discreta Y0.



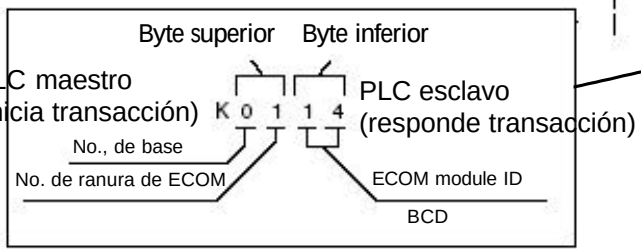
Renglón 2

El segundo renglón usa un relevador especial para identificar un error de comunicación. En el ejemplo, SP123 está encendido si está presente un error de comunicación en la ranura uno. Use relevadores especiales diferentes si su módulo de ECOM está en una ranura diferente (vea la página 4-11 y 4-12). Utilizamos SP123 para encender una luz de indicador conectada con una salida discreta.

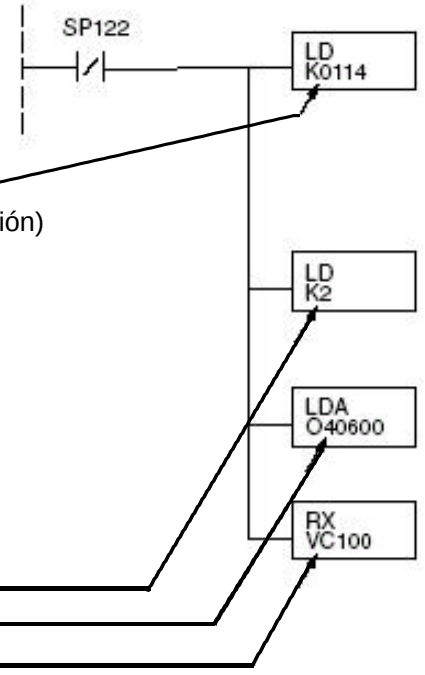


Renglón 3

El relevador especial rotulado SP122 está encendido



cuando el ECOM en la ranura 1 está ocupado transmitiendo o recibiendo datos. La instrucción de lectura puede durar más de un barrido del PLC para completarse. Use este relevador especial para evitar el sobreescribir la instrucción de lectura anterior con cada barrido del PLC.



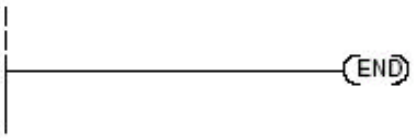
Número de bytes que se transferirán. Máximo = 128 bytes.

Dirección inicial en el PLC maestro, como número octal.

Dirección inicial en el PLC que responde.

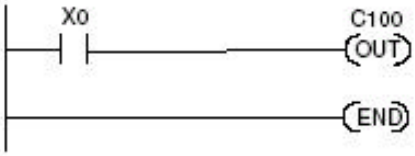
Renglón 4

Todos los PLCs *Direct*LOGIC usan una instrucción END en el renglón final para identificar el cuerpo principal del programa.



Programa para el PLC que responde

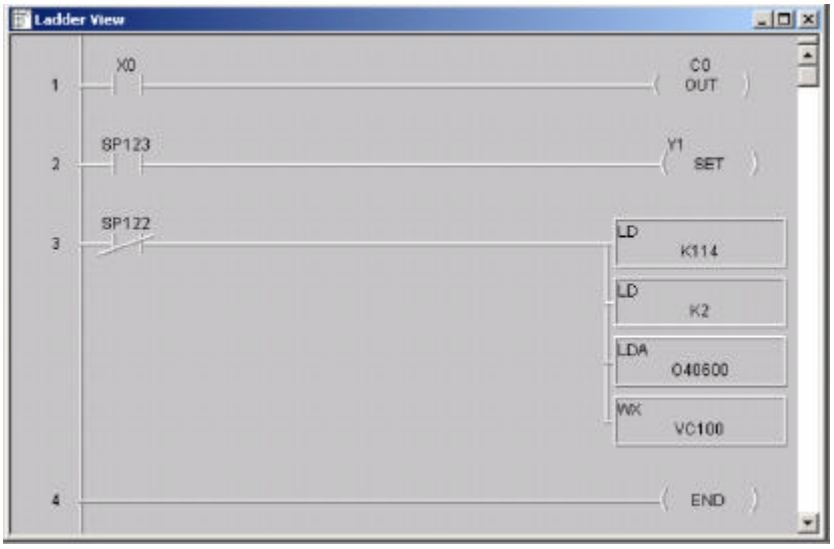
Este programa de 2 renglones reside en la CPU del PLC que responde. Su función es simplemente usar el contacto X0 para activar el relevador interno de control, C100.



Programa con una instrucción de escritura

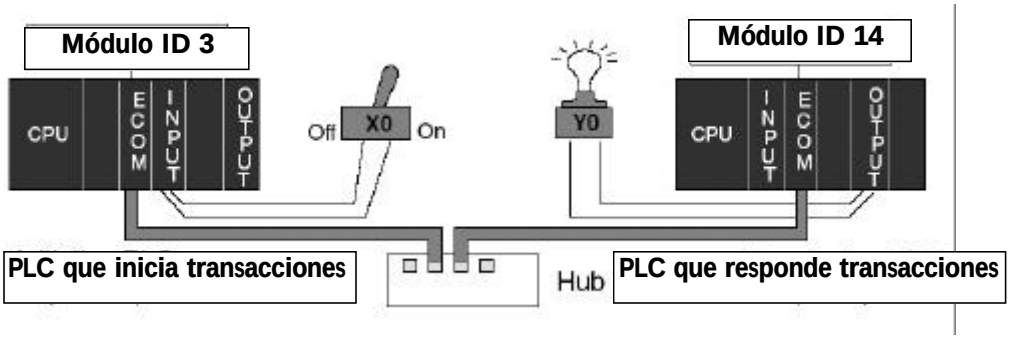
La pantalla Ladder View mostrada abajo abajo es la pantalla de desarrollo de programa en el software de programación *Direct*SOFT32. Este programa de cuatro renglones se explica detalladamente en la página 4-16. Esto es un programa completo aunque su función es muy limitada y solo sirve para entregar el concepto. Hay también un programa de dos renglones que funciona en el PLC que responde, y también se explica en la página 4-16.

Programa para el PLC que inicia las transacciones



Programa para el PLC que responde las transacciones

Cuando se activa la entrada del interruptor al PLC ***que inicia las transiciones*** (maestro) (en la transición de OFF a ON), el bit C100 en el PLC ***que responde*** (esclavo) se va de OFF para ON El programa en el PLC esclavo hace que Y0 se encienda en respuesta al bit C100.



En estos programas de ejemplo, el PLC que inicia y el PLC que responde las transacciones deben estar en el **modo RUN**.

Renglón 1

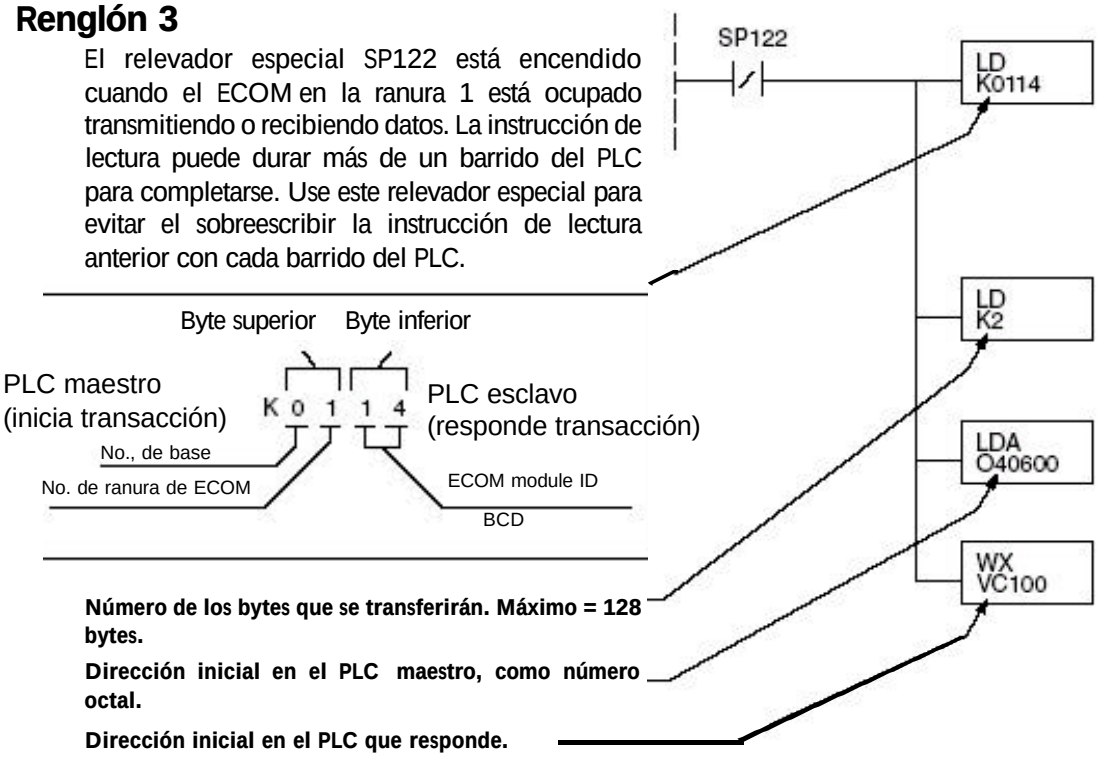
En nuestro ejemplo, el contacto normalmente abierto etiquetado C0 es un relevador interno de control. Cuando C0 está encendido, se energiza la salida discreta Y0.

Renglón 2

El segundo renglón usa un relevador especial para identificar un error de comunicación. En el ejemplo, SP123 se hace ON si está presente un error de comunicación para la ranura uno. Use relevadores especiales diferentes si su módulo ECOM está en una ranura diferente (vea la página 4-11 y 4-12). Usamos SP123 para encender una luz indicadora conectada con una salida discreta.

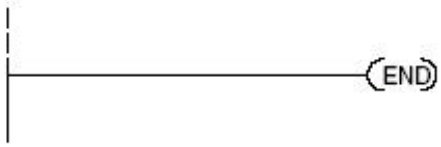
Renglón 3

El relevador especial SP122 está encendido cuando el ECOM en la ranura 1 está ocupado transmitiendo o recibiendo datos. La instrucción de lectura puede durar más de un barrido del PLC para completarse. Use este relevador especial para evitar el sobreescribir la instrucción de lectura anterior con cada barrido del PLC.



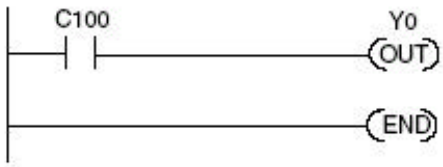
Renglón 4

Todos los PLCs *Direct*LOGIC usan una instrucción END en el renglón final para identificar el cuerpo principal del programa.



Programa para el PLC que responde

Este programa de 2 renglones reside en la CPU del PLC que responde. Su función es simplemente usar el contacto X0 para activar el relevador interno del control, C100.



Integrando instrucciones múltiples de lectura y escritura

Varias instrucciones de lectura y escritura necesitan tener enclavamientos para ejecutar uno por vez, porque se puede procesar solamente una instrucción RX/WX por barrido de la CPU. Usando enclavamientos, se procesa una instrucción RX/WX en cada barrido hasta que se han ejecutado todas las instrucciones RX/WX. Después de la última instrucción, comienza otra vez la secuencia en la primera instrucción RX/WX.

Sin enclavamientos, las instrucciones RX/WX serían ejecutadas en una orden imprevisible, y algunas se podrán ejecutar muchas veces antes de que otras se ejecuten una vez. Los enclavamientos sirven para ***abrir*** (desconectar o deshabilitar) los circuitos para todas las instrucciones de lectura y escritura excepto la que se debe procesar en el barrido corriente de la CPU.

Mostramos aquí dos métodos para crear los enclavamientos necesarios para ordenar instrucciones de lectura y escritura múltiples:

* Relevador de control Internos en secuencia
* Shift Registers
* Contadores

Le mostraremos el desarrollo de los enclavamientos usando estos métodos. Los dos ejemplos mostrados realizan la misma función. Solamente son diferentes los enclavamientos.

NOTA: Para entender completamente el material en esta sección, usted primero debe entender los programas de ejemplo en las páginas 4-13 y 4-15, y también el material en la sección de las instrucciones de red, comenzando en la página 4-3.

El segmento siguiente de programa usa tres instrucciones RX/WX (dos instrucciones de escritura y una instrucción de lectura). Usted puede desarrollar su propio programa que incorpora cualquiera de las estrategias de control de enclavamineto y que amplía el número de enclavamientos para acomodar el número de las instrucciones RX/WX en su programa.

Relevadores de enclavamiento

Es fácil ver la función de los relevadores de enclavamiento si construimos una tabla de verdad primero.

Tabla de verdad	C52	C51	C50
Primer RX/WX	0	0	0
Segundo RX/WX	0	0	1
Tercero RX/WX	0	1	0
Cuarto RX/WX	0	1	1
Quinto RX/WX	1	0	0
Sexto RX/WX	1	0	1
Séptimo RX/WX	1	1	0
Octavo RX/WX	1	1	1

En la parte superior de la tabla de verdad mostramos relevadores internos de control que estamos considerando para nuestra estrategia de secuenciamiento. Hemos utilizado C50 hasta C52 para nuestra tabla, pero cualquier contacto que no se use para otros propósitos en su programa trabajará del mismo modo. Abajo al lado izquierdo de la tabla, enumeramos el número de las instrucciones RX/WX que podemos desear usar en nuestro programa de RLL.

Los tres contactos en esta tabla de verdad acomodarán ocho instrucciones. Nuestro programa tiene solamente tres instrucciones RX/WX así que necesitamos solamente utilizar dos contactos (vea porqué en la página 4-18). Usaremos C50 y C51. Un contacto adicional (C53) nos daría 32 combinaciones puesto que el número de combinaciones se amplía como la potencia de 2.

Nuestras tres instrucciones de RX/WX se pueden secuenciar por los dos contactos C50 y los contactos de C51. Dos bits proporcionan cuatro estados binarios diferentes:

* ambos apagados
* C50 encendido y C51 apagado
* C50 apagado y C51 encendido
* ambos encendidos

Necesitamos solamente utilizar tres de los cuatro estados binarios (en un círculo) puesto que tenemos que hacer una secuencia de solamente tres instrucciones RX/WX.

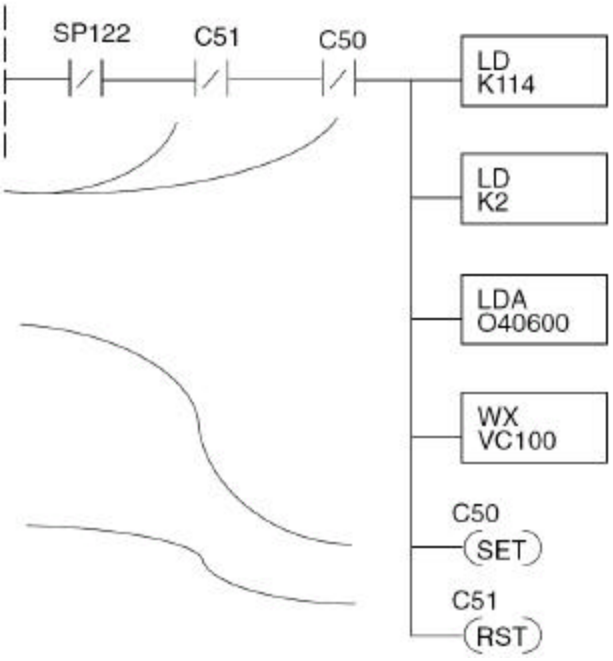
Tabla de verdad	C52	C51	C50
Primer RX/WX	0	0	0
Segundo RX/WX	0	0	1
Tercero RX/WX	0	1	0
Cuarto RX/WX	0	1	1
Quinto RX/WX	1	0	0
Sexto RX/WX	1	0	1
Séptimo RX/WX	1	1	0
Octavo RX/WX	1	1	1

Primera Instrucción de RX/WX

C50 y C51 son contactos de enclavamiento. Están cerrados normalmente en este renglón para permitir flujo de energía a la primera instrucción WX. Ambos bits están apagados, correspondiendo a la primera fila de la tabla de verdad.

Después de que se ejecute la instrucción de WX se configura C50 (activado) que abre el contacto en este renglón y cierra el contacto C50 en el renglón siguiente.

C51 es un reset (se apaga), lo que deja el contacto C51 cerrado para el próximo renglón.

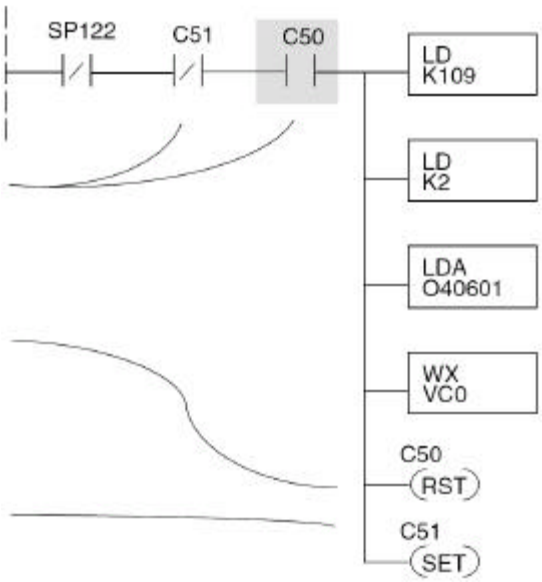


Segunda instrucción RX/WX

C50 está normalmente abierto y C51 está normalmente cerrado. Para que este renglón sea ejecutado, el bit C50 debe estar encendido y el bit C51 debe estar apagado, correspondiendo a la segunda fila de la tabla de verdad. C50 fue encendido en el renglón anterior. C51 fue apagado en el renglón anterior.

Después de que se ejecute la instrucción de WX se apaga o desactiva C50, que abre el contacto C50 en este renglón y lo cierra en el próximo renglón.

C51 es SET (activado), lo que cierra el contacto normalmente abierto C51 en el próximo renglón

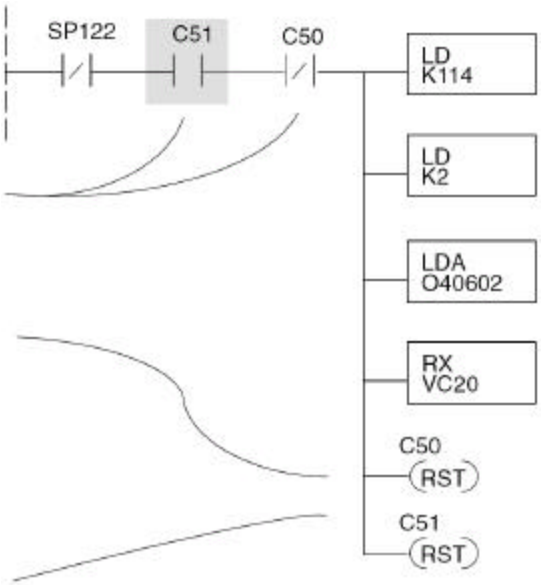


Tercera instrucción RX/WX

En este último renglón, C50 está normalmente cerrado y C51 está normalmente abierto. Para que este renglón sea ejecutado, el bit C50 debe estar apagado y el bit C51 debe estar encendido, correspondiendo a la tercera fila de la tabla de verdad. C51 fue apagado en el renglón anterior.

Después de que se ejecute la instrucción de RX, se desactiva C50, lo que abre el contacto C50 en este renglón y permite que se cierre en preparación para repetir el primer renglón de comunicación en el barrido siguiente de la CPU (página 4-18).

C51 también SE REAJUSTA, lo que permite que el contacto C51 se cierre en preparación para repetir el primer renglón de comunicación en el barrido siguiente de la CPU (página 4-18).



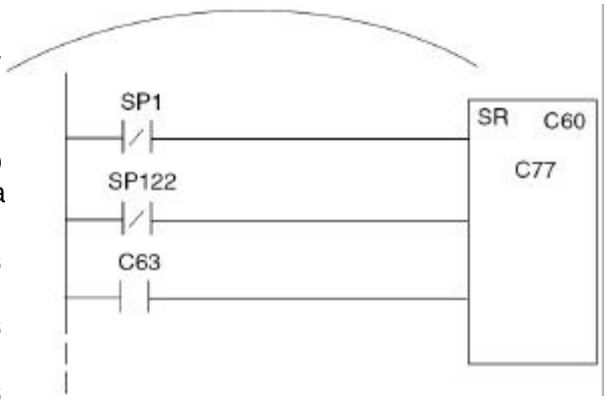
Volviendo a la primera instrucción RX/WX

Volviendo al final de la tercera instrucción de RX/WX, completamos un ciclo volviendo a la línea superior de la tabla de verdad en la página 4-18. C50 y C51 están apagados, y el barrido siguiente de la CPU ejecuta la instrucción.

El shift register

El shift register se puede usar para crear enclavamientos, como alternativa de usar los relevadores de control. Para una explicación completa de la función del shift register, vea el manual de usuario para su PLC. Si usted tiene más de algunas instrucciones RX/WX, podría ser inconveniente usar los relevadores de control. El shift register permite que un solo contacto sea utilizado en cada renglón de comunicación como enclavamiento.

La entrada de datos al shift register (SR) es el relevador especial SP1. SP1 es el bit "siempre encendido". Combinado con un contacto normalmente cerrado envía ceros a la entrada de datos del shift register. La entrada de clock al shift register es SP122, el bit de comunicación "ocupado". Cada vez que una de las instrucciones RX/WX se ejecuta, el shift register mueve el bit de datos un lugar. C63 se utiliza en este ejemplo para resetear la palabra del shift register a ceros.



C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	C63	C62	C61	C60

Shift register después del primer barrido

C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	C63	C62	C61	**C60**

Shift register después del segundo barrido

C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	C63	C62	**C61**	C60

Shift register después del primer RX/WX

C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	C63	**C62**	C61	C60

Shift register después del segundo RX/WX

C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	**C63**	C62	C61	C60

Shift register después del tercer RX/WX

C77	C76	C75	C74	C73	C72	C71	C70	C67	C66	C65	C64	C63	C62	C61	C60

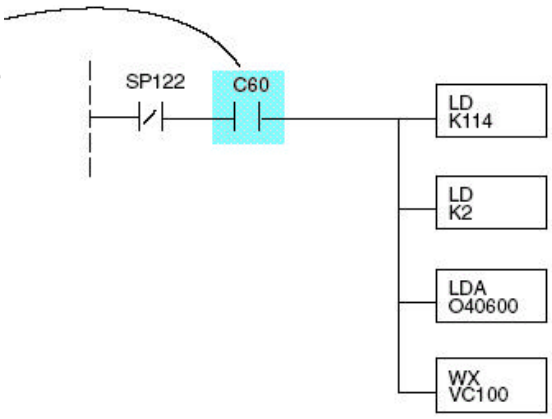
Shift register después del tercer RX/WX más un barrido

Comparación “igual a”

La instrucción de comparación “igual” detecta cuando el shift register se resetea a cero. Cuando esa condición es verdad el bit C60 es SET por este renglón. El bit C60 se convierte en el bit más alto cambiado por el shift register hasta que cada instrucción RX/WX se ejecuta alternadamente.

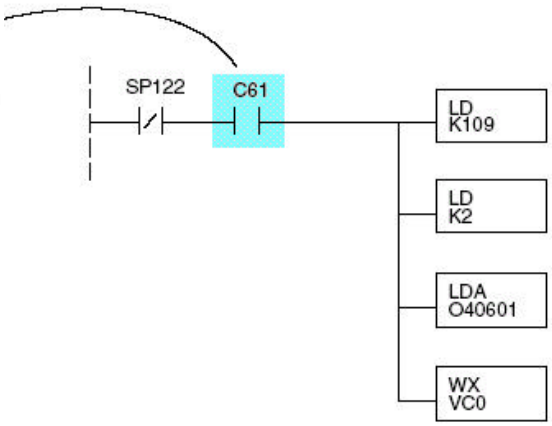
Primera instrucción de RX/WX

C60 es el contacto de enclavamiento. Es activado por el renglón de comparación igual que precede éste.



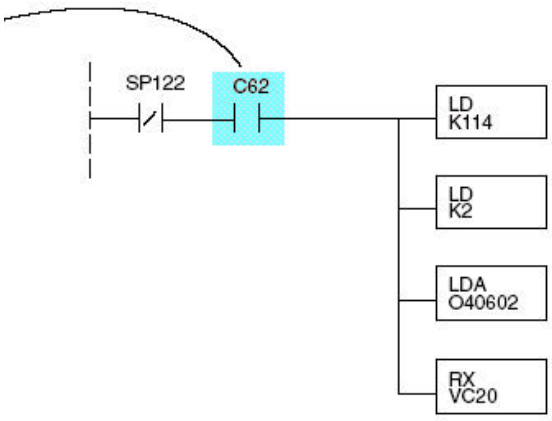
Segunda instrucción de RX/WX

C61 es el contacto de enclavamiento. Es cambiado a ON por los pasos de comando del shift register en un renglón precedente.



Tercera instrucción de RX/WX

C62 es el contacto de enclavamiento. Es cambiado a ON por los pasos de comando del shift register en un renglón precedente.



Después de que se ejecute este renglón, el shift register desplaza el bit superior de C62 a C63 en el barrido siguiente de la CPU. C63 resetea el shift register a ceros, la comparación “igual”configura el bit C60 como ON y la CPU ejecuta la primera instrucción de RX/WX.

La preferencia del traductor es usar un contador, de tal modo que cada vez que el bit busy se activa, el contador aumenta en 1. Al final del conteo, el contador se resetea para comenzar el ciclo nuevamente.

Entonces cada instrucción RX/WX tendrá un enclavamiento con una comparación. como es mostrado en el ejemplo de abajo, para el caso de 5 instrucciones.