

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 *DirectLOGIC*® 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 :

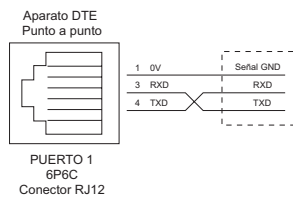
- El estándar de cableado
- El protocolo de comunicaciones
- Los parámetros de comunicaciones

Cada uno de éstos será discutido más detalladamente en relación a los PLCs *DirectLOGIC*.

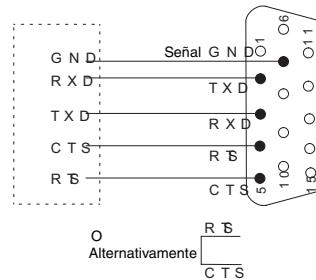
Estándar de cableado

Hay tres estándares de cableado que se pueden utilizar con los PLCs *DirectLOGIC*: 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.



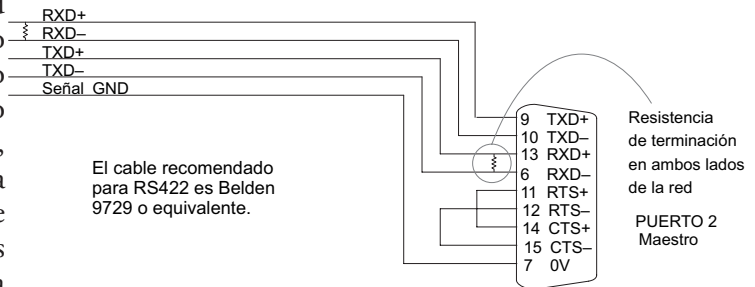
Conexiones del puerto 1



Conexiones del puerto 2

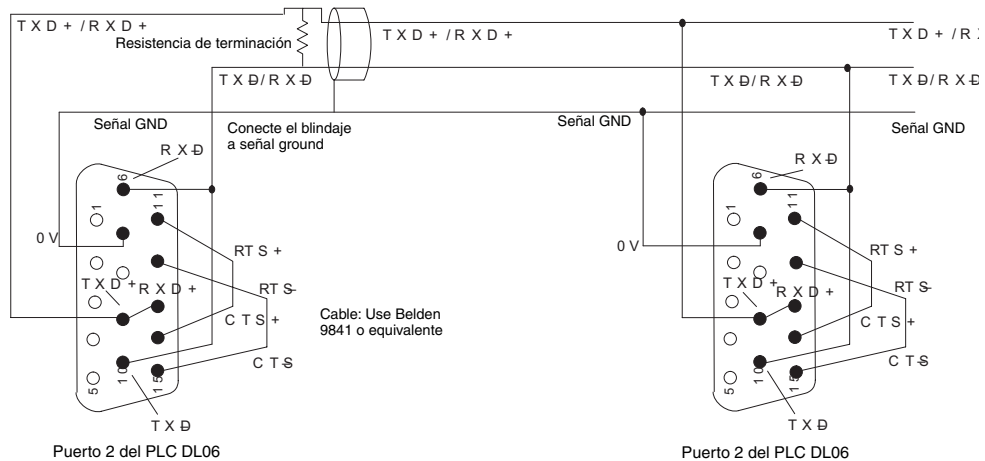
RS-422 es un estándar de múltiples puntos con una distancia práctica de cableado de 1000 metros, o 3280 pies, longitud total máxima. Esto significa que puede comunicarse un maestro solamente, con hasta 10 esclavos,

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).



El cable recomendado para RS422 es Belden 9729 o equivalente.

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
<i>Direct</i> 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.

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.

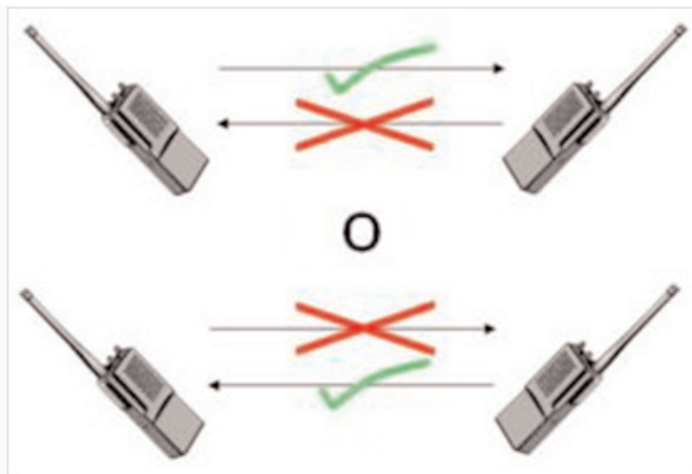


NOTA: El punto a recordar es que cuando hay el mismo tipo de parámetros disponible en el maestro y en el esclavo (es decir velocidad, paridad, bits de parada, etc), los valores de éstos deben ser iguales.

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.

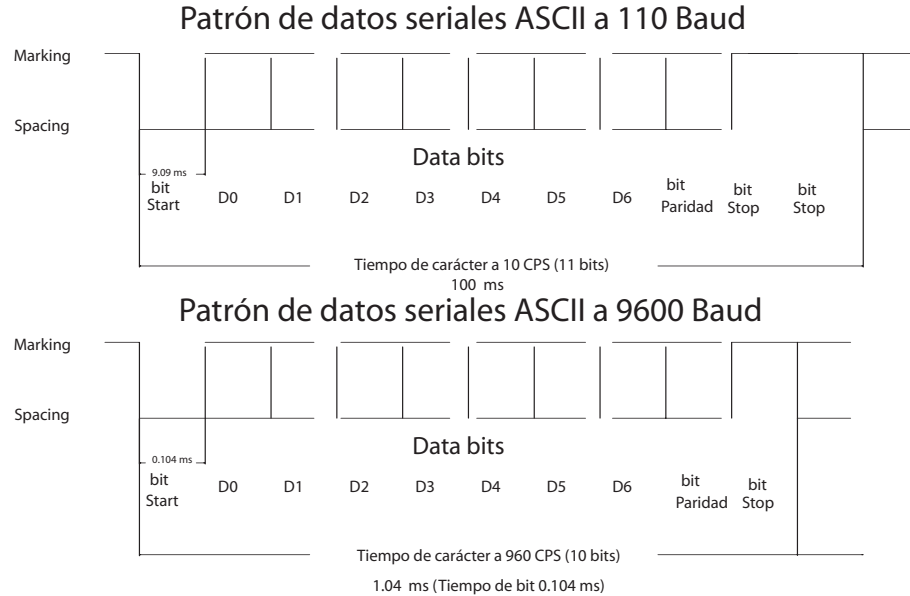


Una simple ilustración de un sistema half duplex.

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:

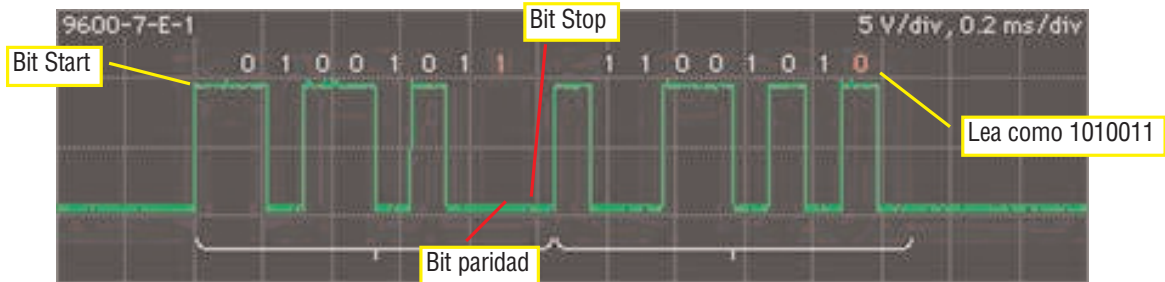


K

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.

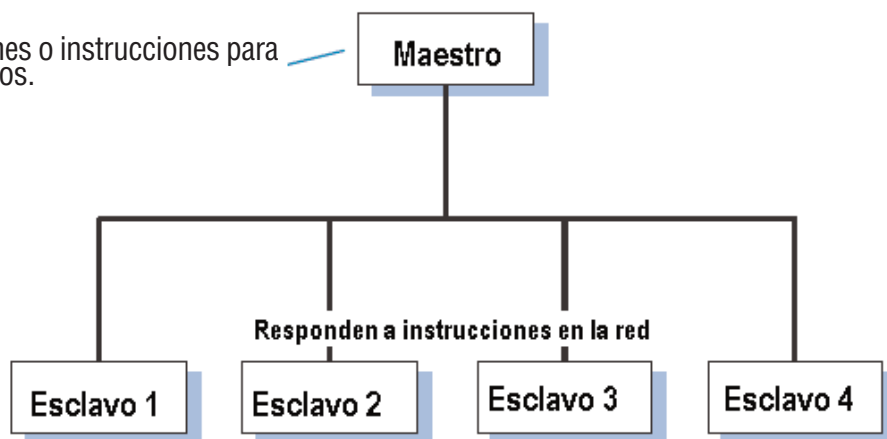
Para utilizar el puerto 2 para comunicaciones K-Sequence usted primero necesita configurar el puerto usando *DirectSOFT* o lógica ladder según lo descrito previamente.

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.

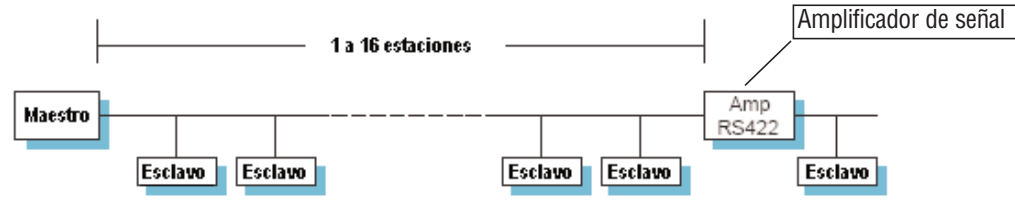
El protocolo *DirectNET* se puede usar para comunicarse a otro PLC o a otros dispositivos que puedan utilizar el protocolo *DirectNET*. El PLC DL06 puede ser usado como maestro en el puerto 2 o como esclavo usando el puerto 1 o el puerto 2.

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.

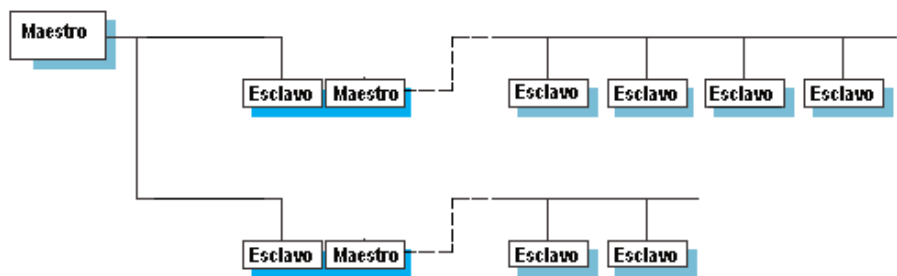
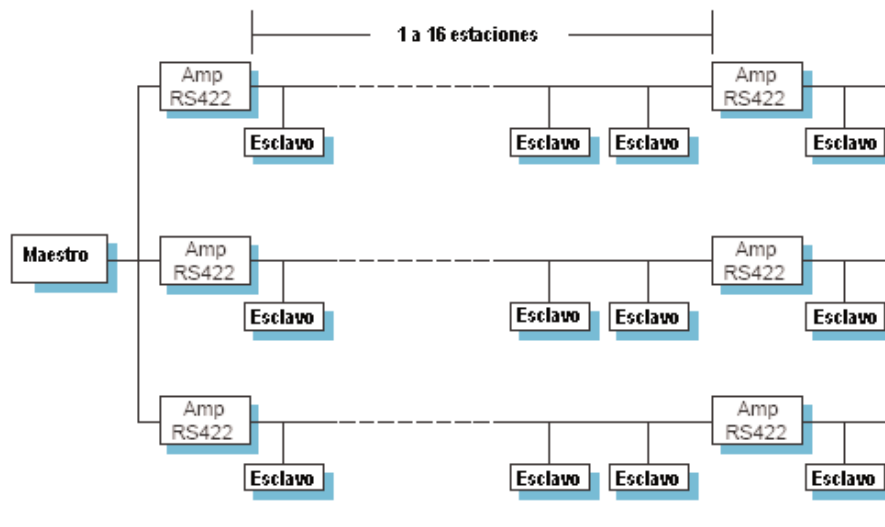
Apéndice K: Introducción a comunicaciones seriales

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.

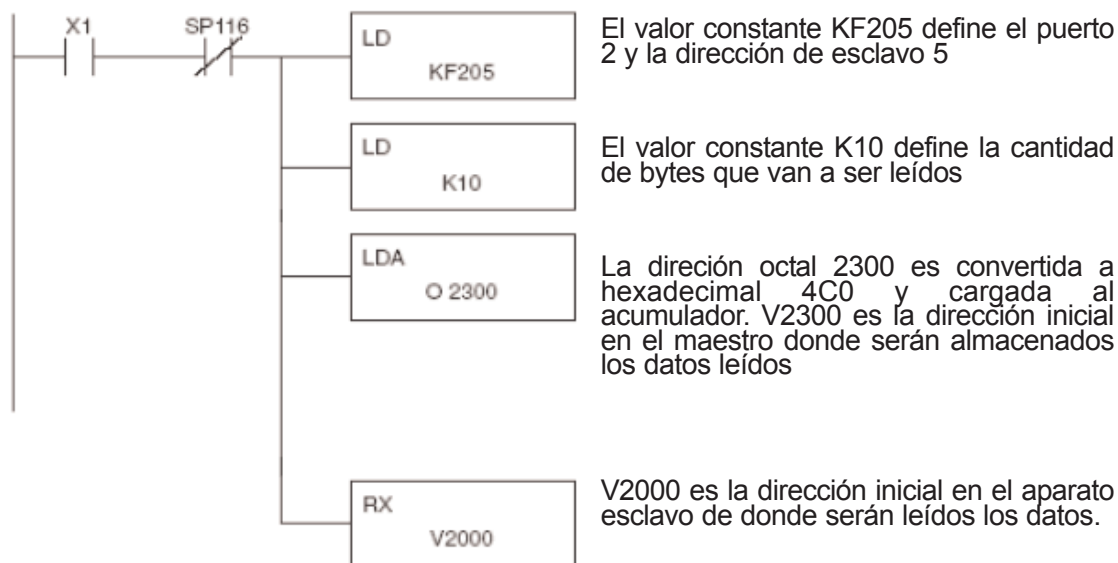
- **PLC como Maestro.** El PLC maestro pide iniciar instrucciones en el programa de RLL para ordenar transferencia de datos.

- **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.

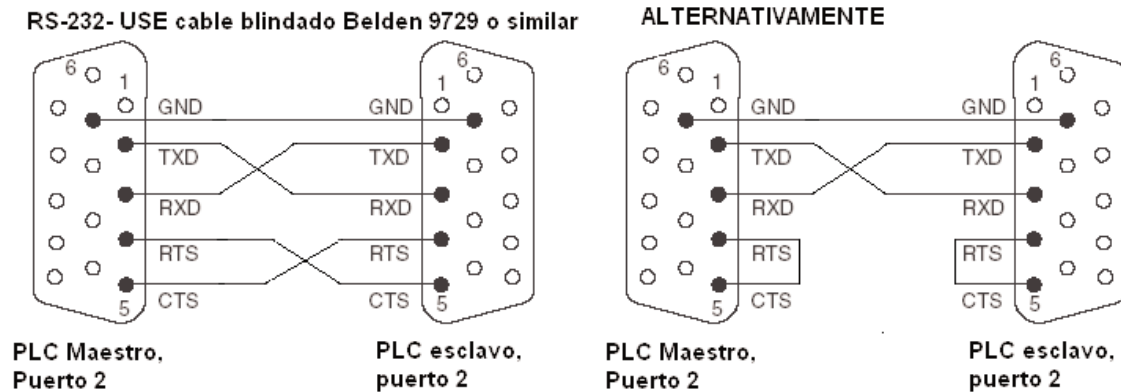
Haga entonces un diagrama de su red o redes. Esto se llama topología de la red.

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.

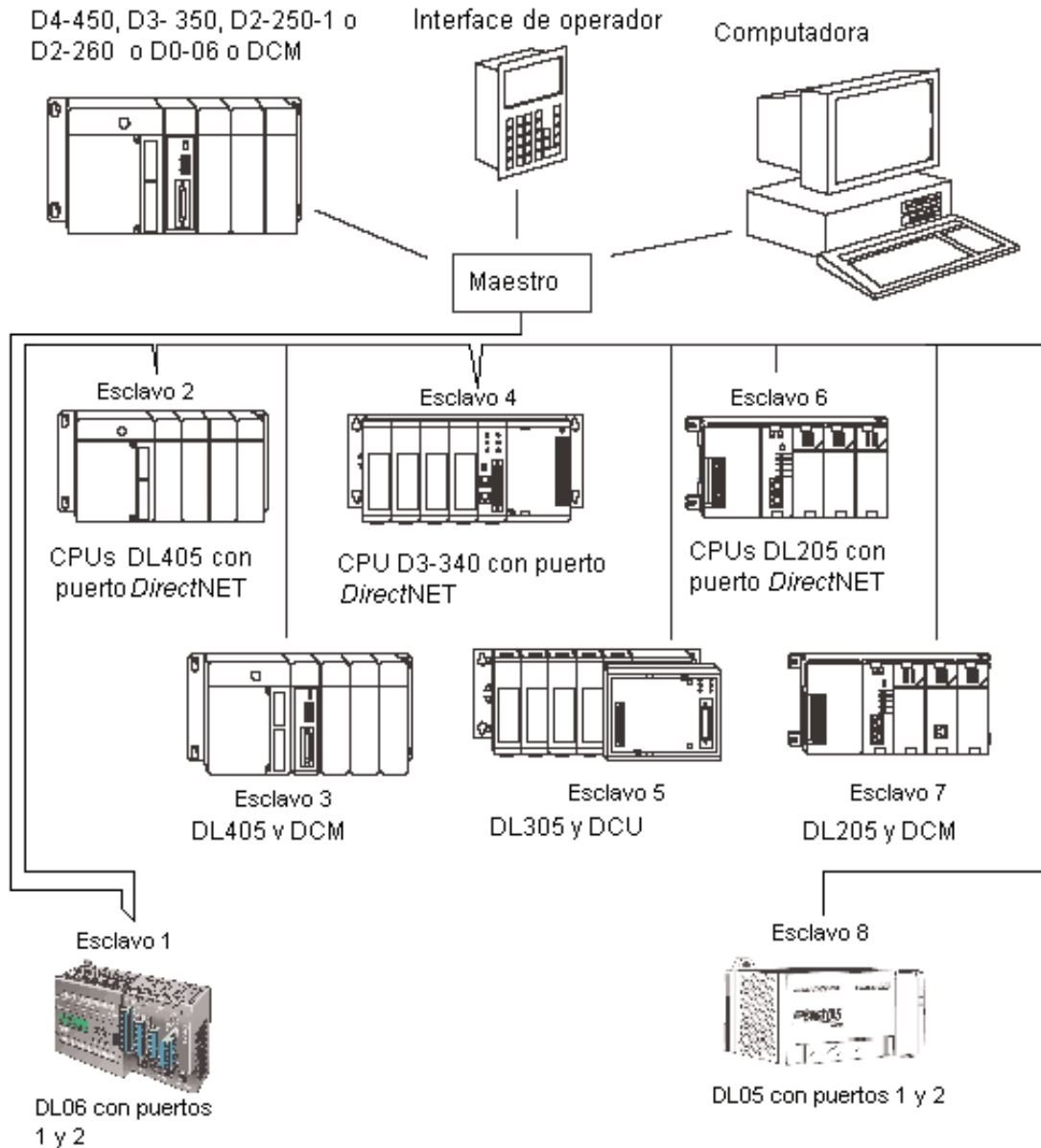
Es importante hacer las terminaciones del cable cuidadosamente, con el conector adecuado, para evitar fallas de comunicación que son difíciles de detectar.

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.



K

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.

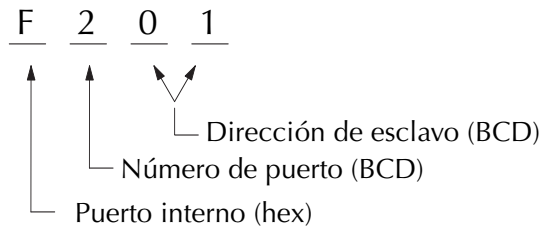
Si usted tiene dos o tres instrucciones de comunicación de red en su programa, usted puede usar bits para enclavarlos, con la ayuda de un bit que muestra el estado del puerto de comunicación libre u ocupado. Veremos más de esto posteriormente.

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.

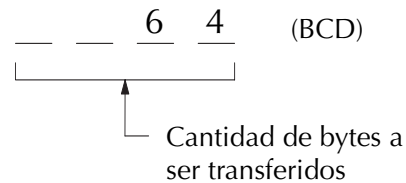
Paso 1: Identifique el puerto del maestro y la dirección del esclavo

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.

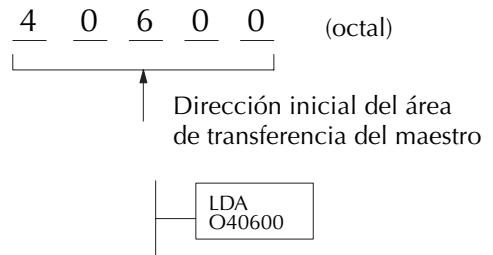
La tabla siguiente muestra los rangos de bytes para los varios tipos de productos de *DirectLOGIC*.

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

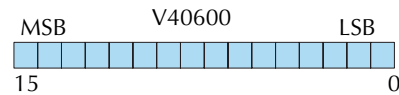


Paso 3: Especifique el área de memoria del maestro

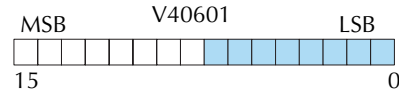
La tercera instrucción en el programa RX o WX es una instrucción LDA. Su propósito es cargar la dirección inicial del área de memoria a ser transferida. Es entrado como un número octal y la instrucción LDA la convierte a hexadecimal y coloca el resultado en el acumulador.



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.



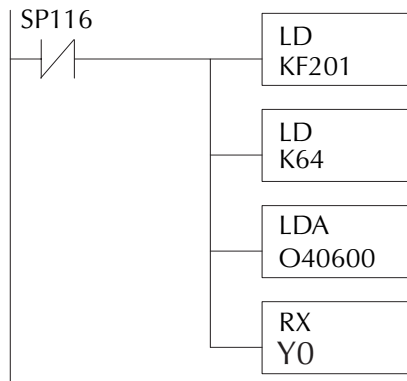
NOTA: Ya que las palabras de memoria V son siempre de 16 bits, usted no siempre puede usar la palabra entera. Por ejemplo, si usted sólo especifica 3 bytes y usted lee las salidas Y del esclavo, usted sólo obtiene 24 bits de datos. En este caso, sólo los 8 bits menos significativos de la última localización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



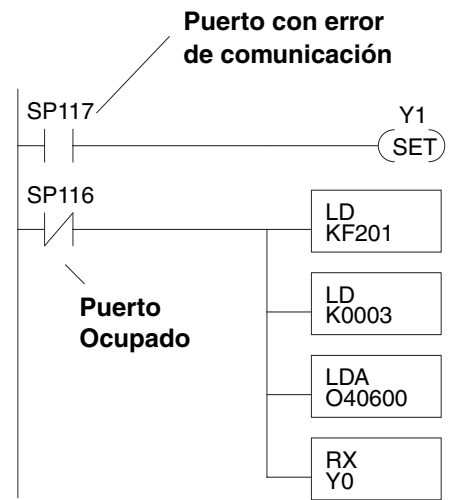
Comunicaciones con un programa ladder

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.



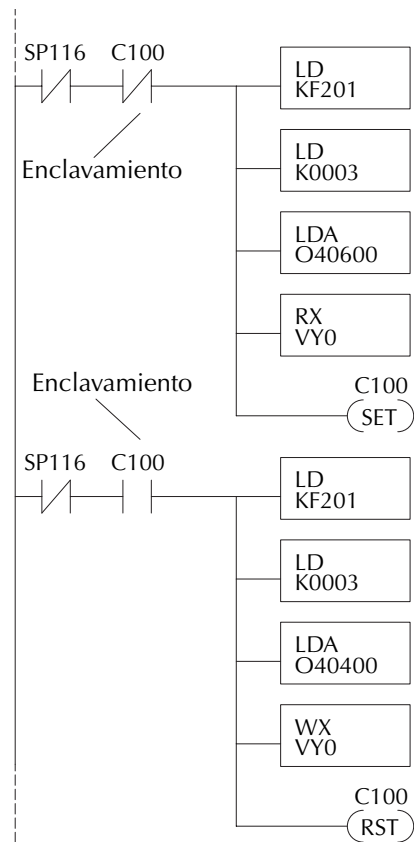
Enclavamientos de lectura y escrituras múltiples

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

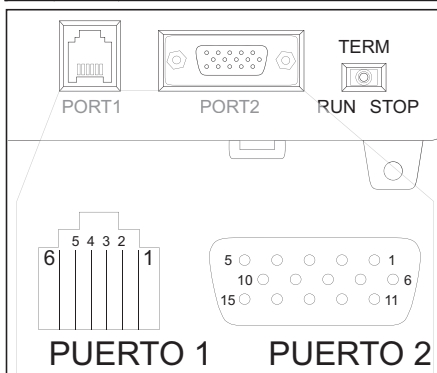
Puerto 1	
Puerto 1	Conecta a HPP, <i>DirectSOFT</i> 5, interfaces de operador , etc.
	6 clavijas, RS232C
	Tasa de comunicación (baud): 9600 (fija)
	Paridad: odd (fija)
	Dirección del nodo: 1 (fija)
	8 data bits
	1 start, 1 stop bit
	Asíncrono, half-duplex, DTE
	Protocolo (auto-select): K-sequence , <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 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 *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	xxxxxxxx (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,

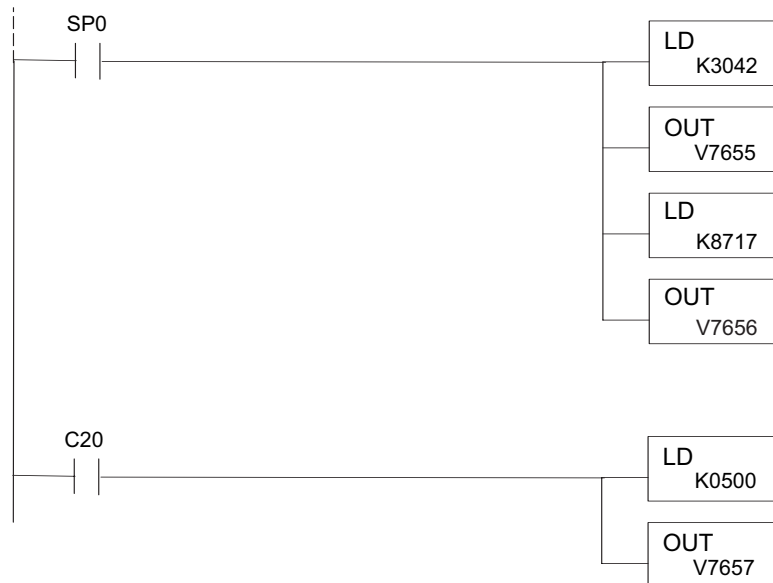
1 Stop bit,

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 table contains the following data:

Element	Status
V7655	3042
V7655	0011000001000010
V7656	8717
V7656	1000011100010111

Annotations with arrows point to the 'Status' column values: 'Formato BCD/HEX' points to '3042' and 'Formato binario' points to '0011000001000010'. On the right is the 'Setup Communication Ports' dialog box. It is configured for 'Port 2'. The 'Protocol' section has 'DirectNET' selected. 'Time-out' is set to 'Base Timeout x 1'. 'RTS on delay time' is 10 ms, and 'RTS off delay time' is 5 ms. 'Station Number' is 23. '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 indicates 'Port 2: 15 Pin'.

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.

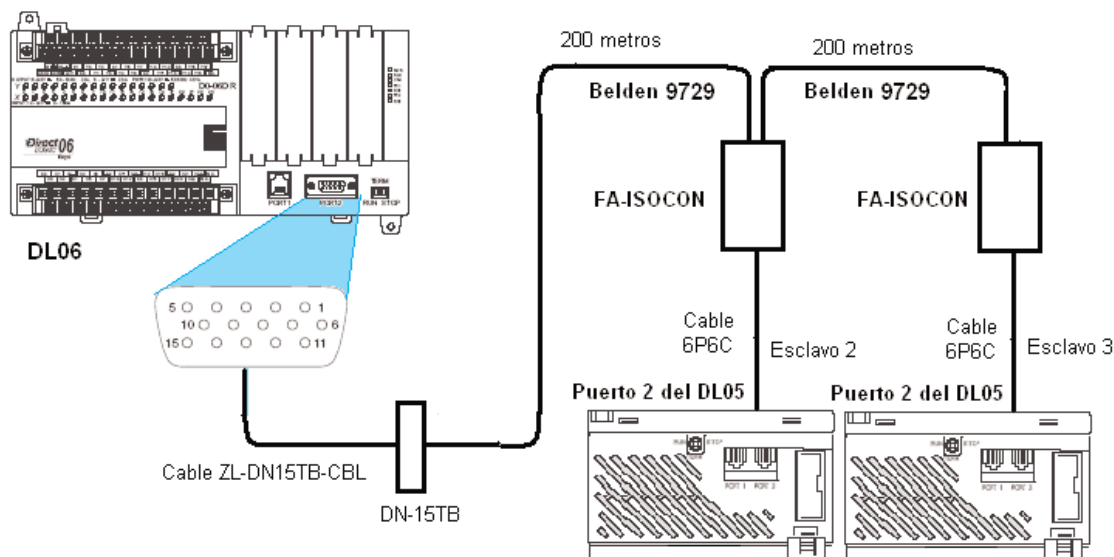
Como ya fue dicho, es necesario:

- 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.

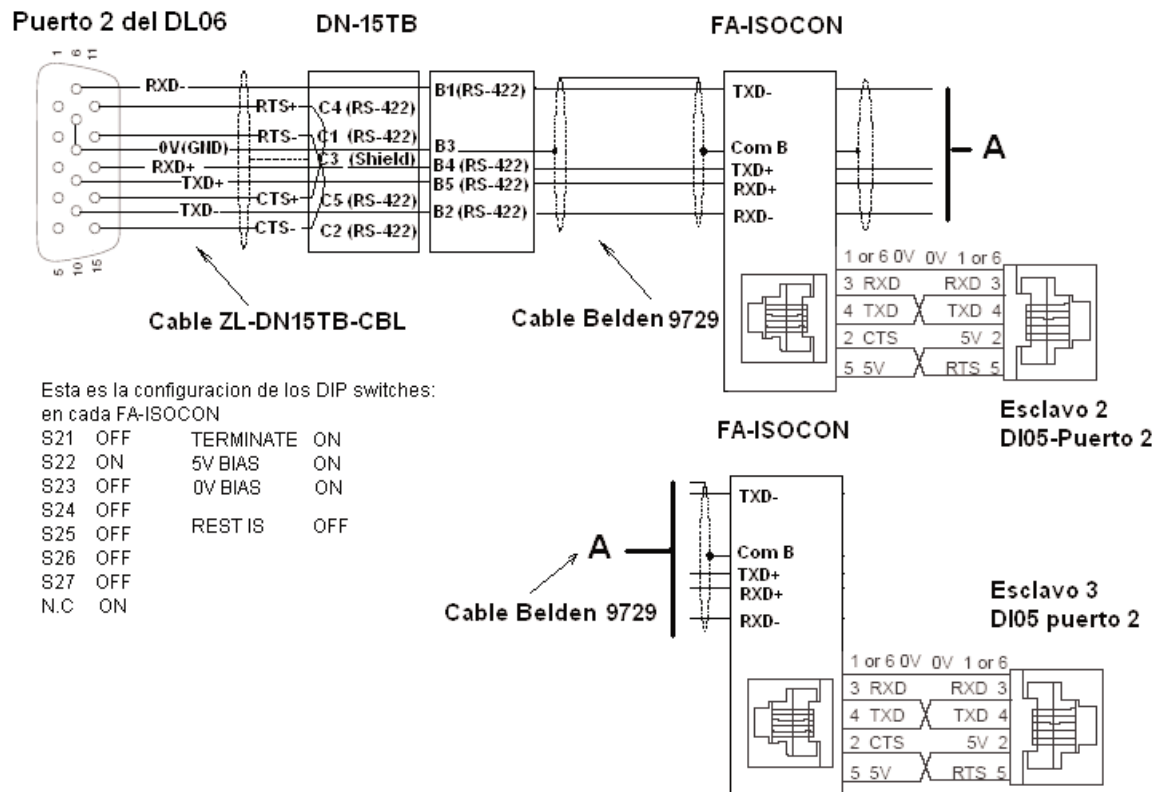
Queremos establecer una red con distancia entre nodos de 200 m a una velocidad de 38400 kbps. Este es el diagrama planeado:



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-ISOCÓN del esclavo 2.

El módulo FA-ISOCÓN 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.

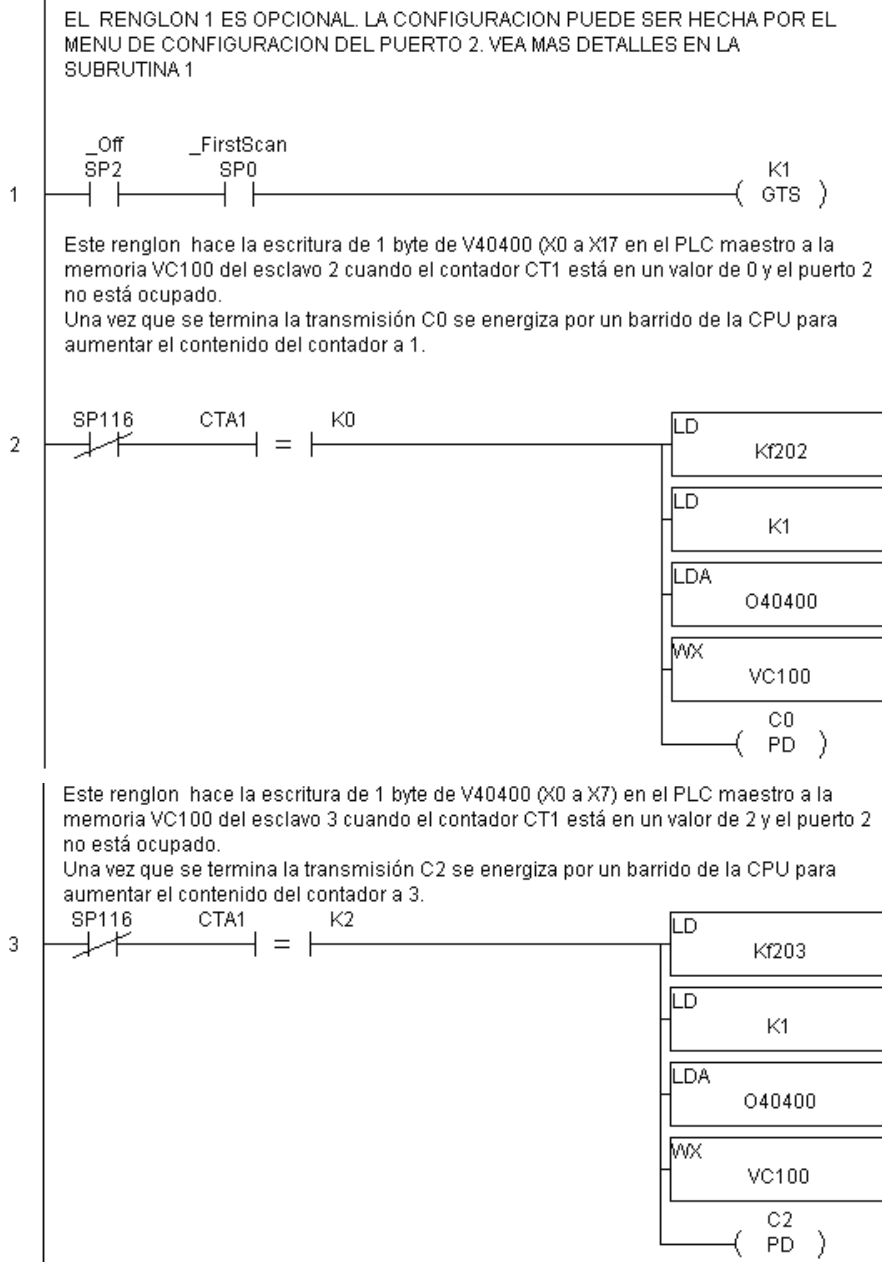
Se usan 4 renglones (rungs) para definir cada escritura y lectura.

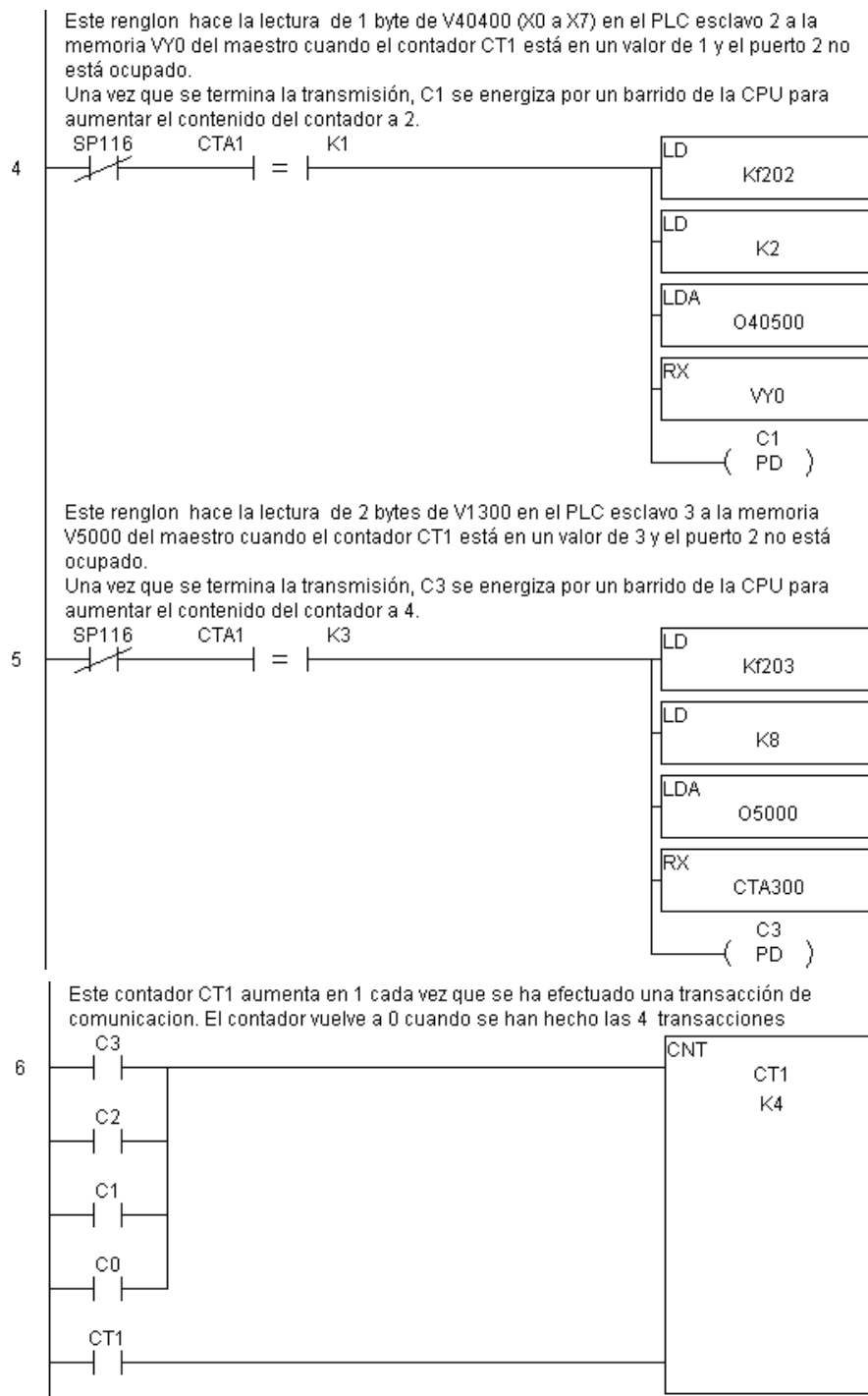
Adicionalmente se ha colocado un contador para medir la eficiencia de transmisión. Para eso se cuentan las veces que se han terminado las cuatro transacciones de comunicación.

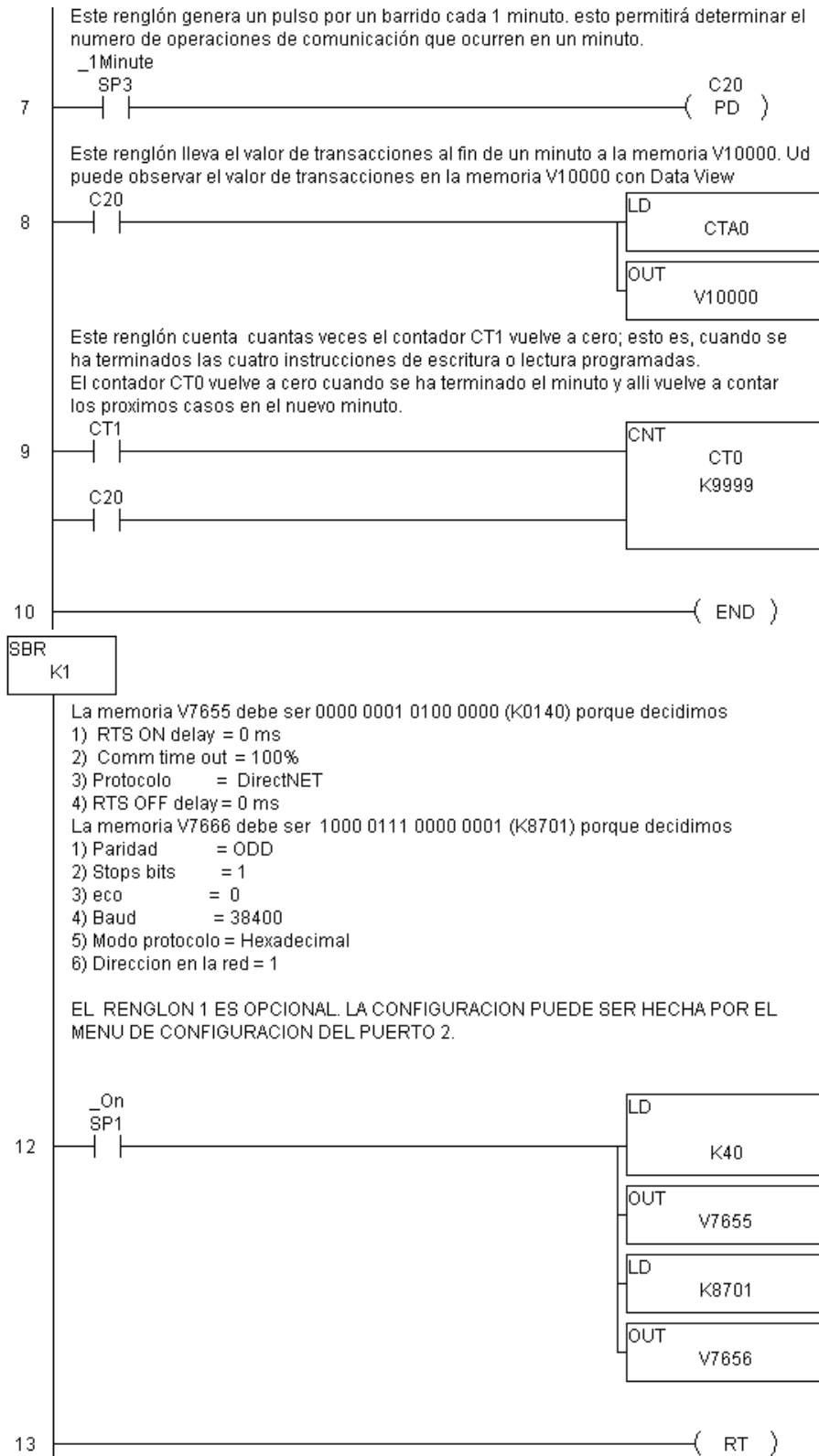
En este caso se llega a 386 transacciones por minuto, es decir, hay 6,43 actualizaciones por segundo y desde allí se puede determinar que se actualizan los datos cada 155 ms.

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:



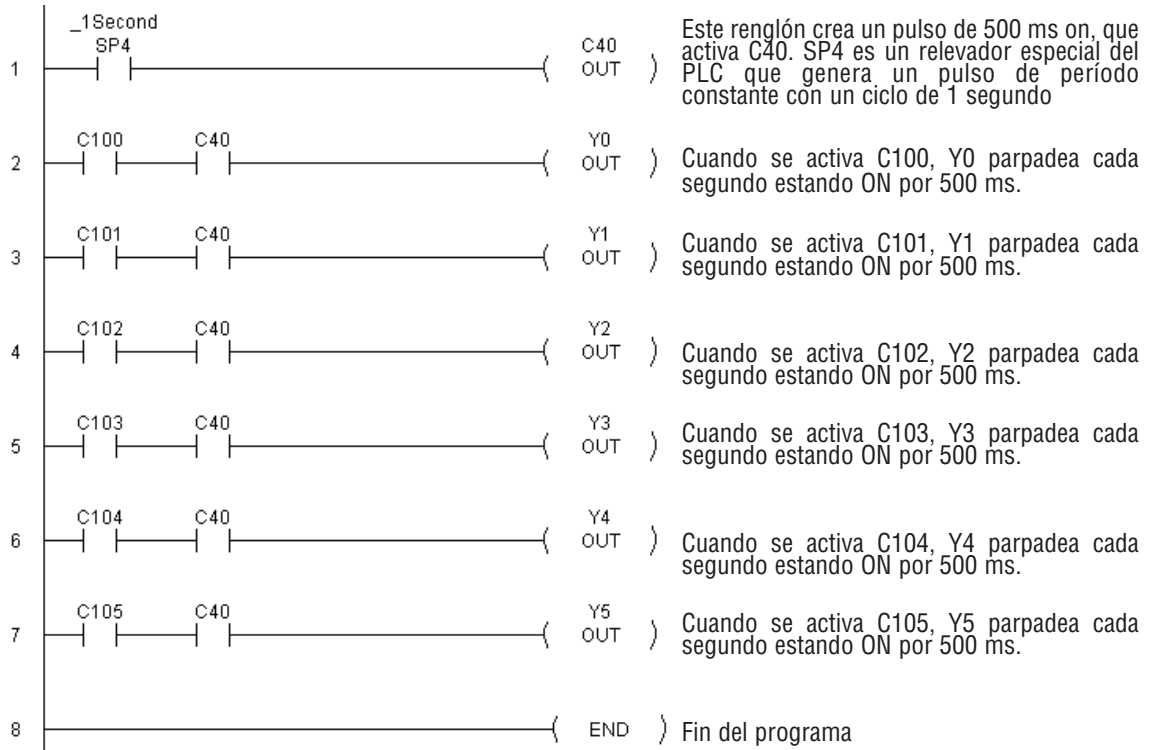




K

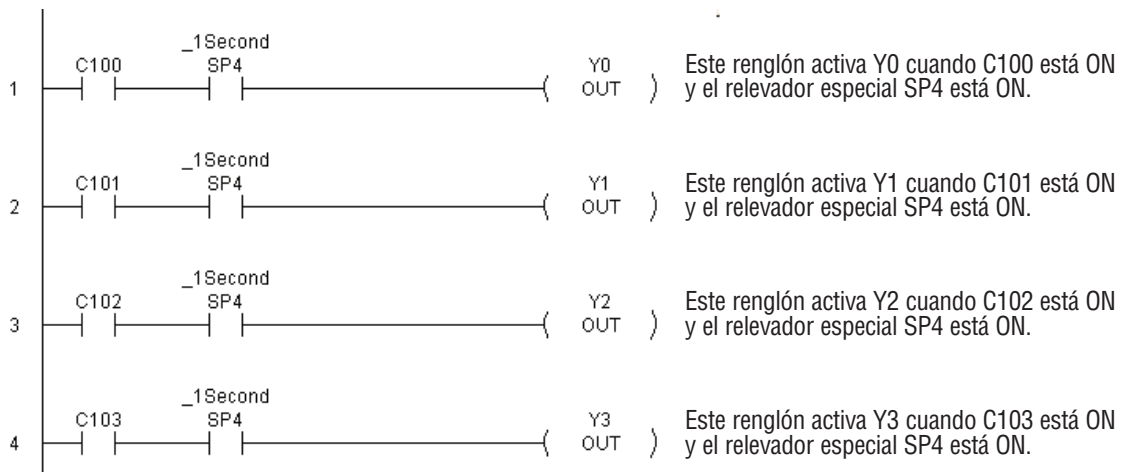
C2. Creación del programa en el esclavo 2.

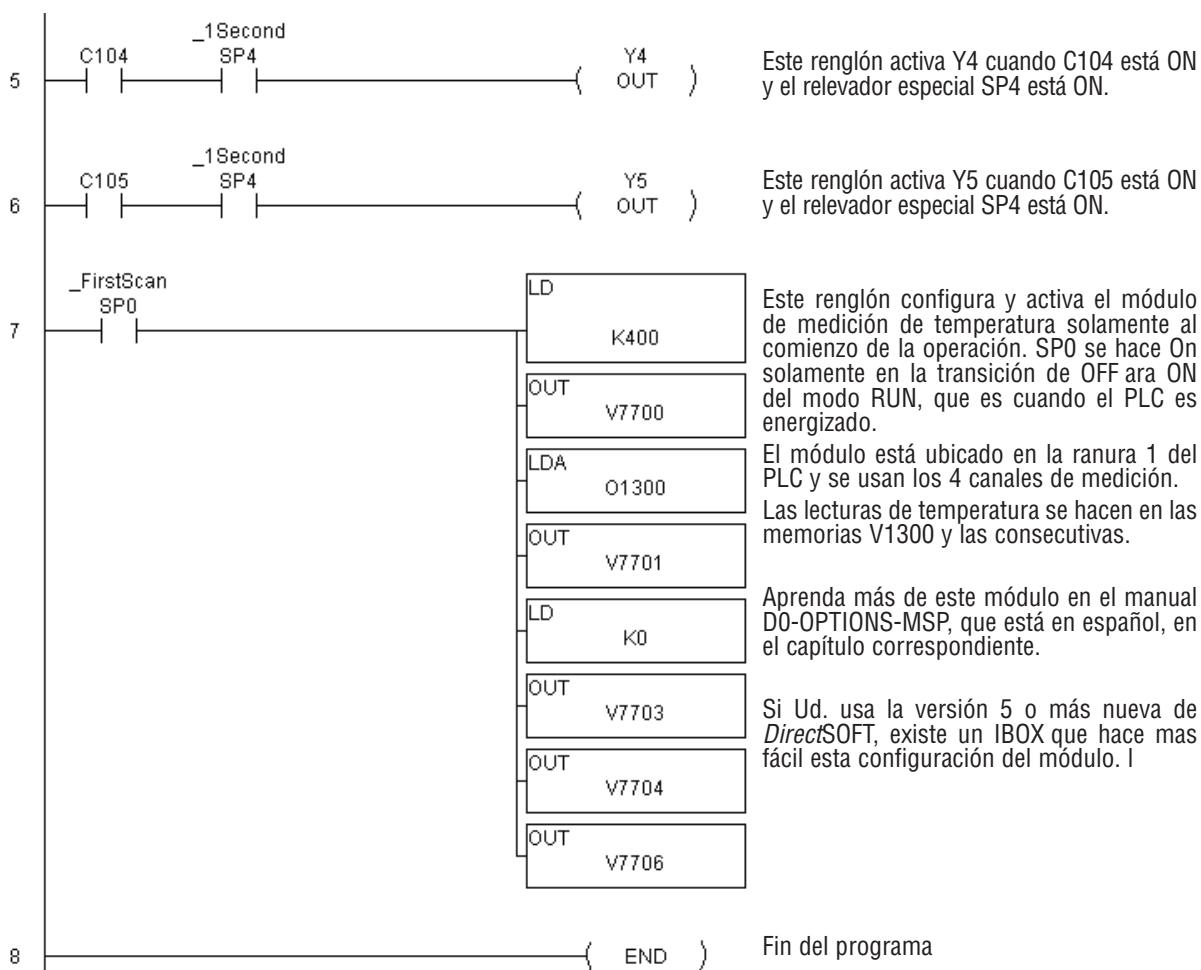
Note que la generación del pulso de 500 ms se hace en el propio PLC esclavo. La lectura de los bits Y0 a Y5 se hace en el maestro con la palabra V40500 y no es necesaria ninguna lógica de comunicación en el esclavo 2.



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

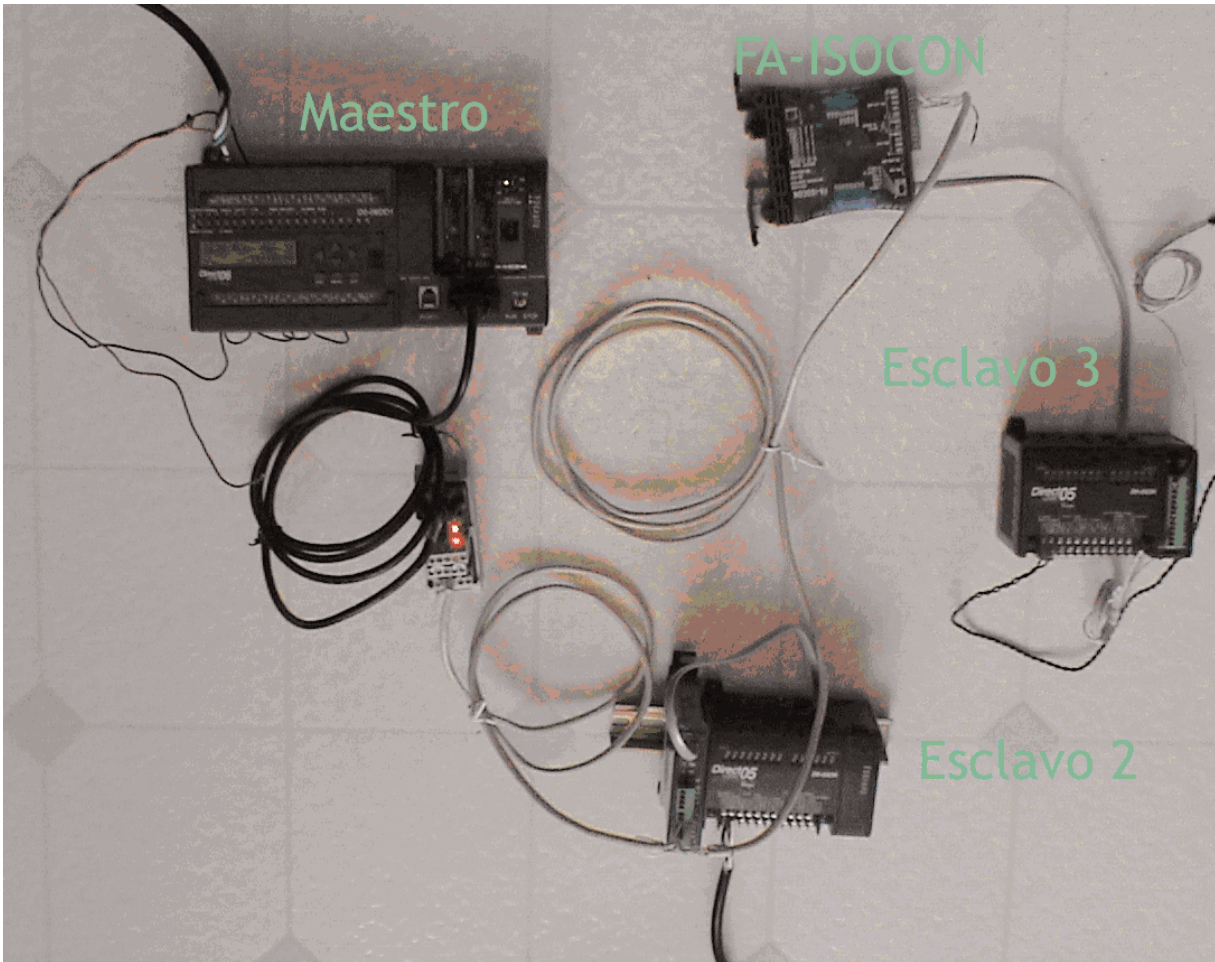
Despues de programar los PLCs, deben configurarse los puertos 2 del PLC DL06 así como también los puertos 2 de los PLCs DL05 a 19200 kbps y también los parámetros adicionales.

Los DIP switches de los adaptadores FA-ISOCON 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-ISOCON. 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.

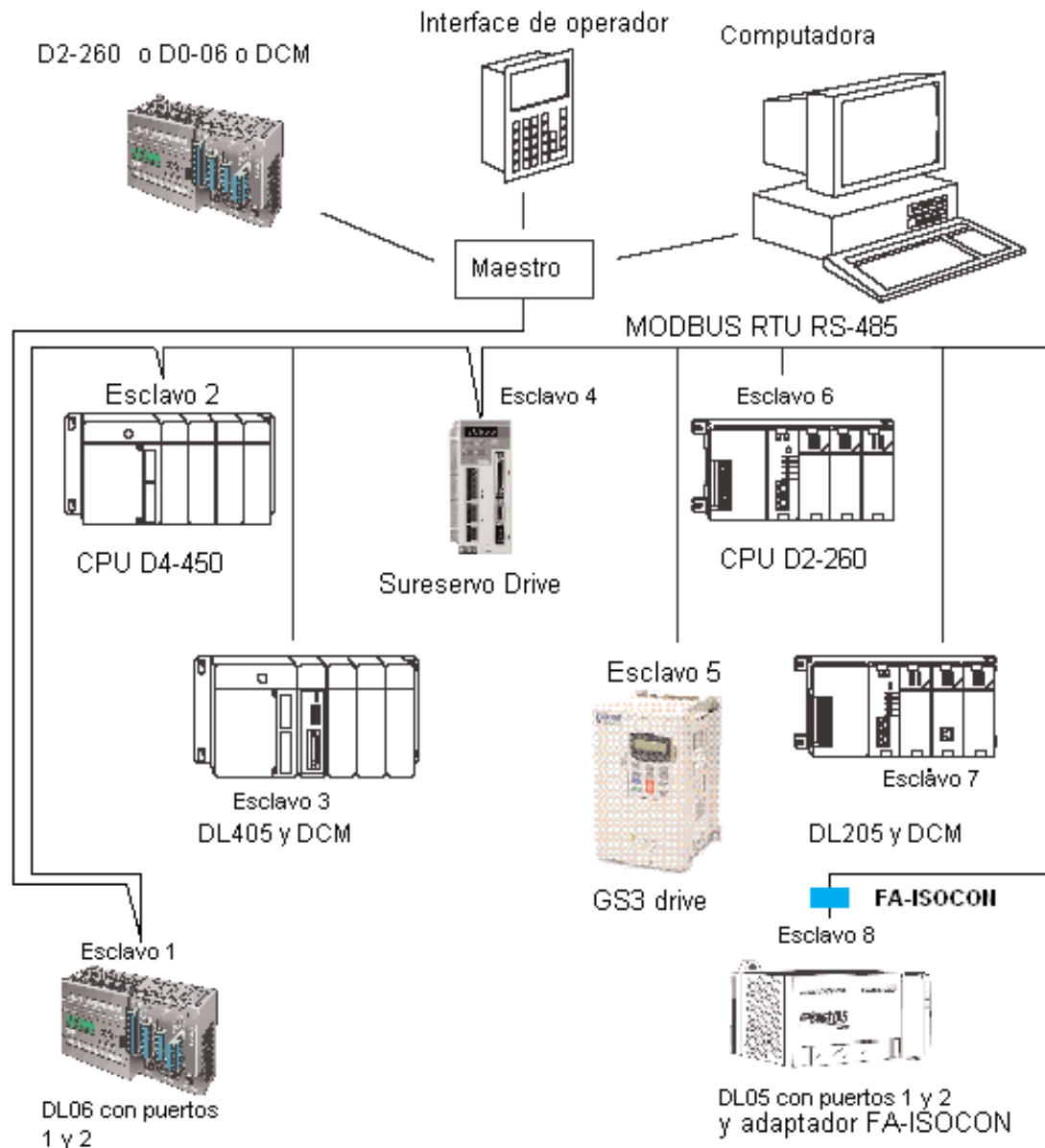


K

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

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 em 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.

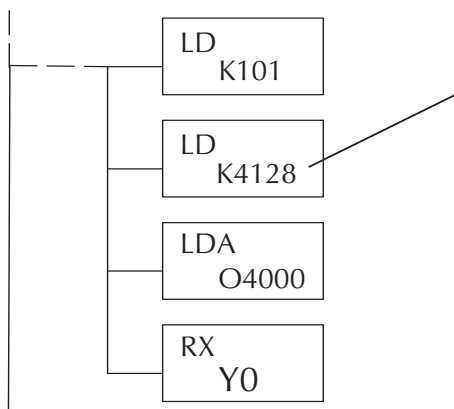
Tipo de de datos discretos				
Tipo de memoria del DL06	Rango del PLC (Octal)	(Modo 484)	Dirección (Modo 584/984)	Tipo datos MODBUS
Entradas globales (GX)	GX0-GX1746	1001 - 1999	10001 - 10999	Entrada
	GX1747-GX3777	---	11000 - 12048	Entrada
Entradas (X)	X0 - X1777	---	12049 - 13072	Entrada
Relevadores 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

K

* 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 instrucción 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
TA10 = 8 decimal

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

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

Ejemplo 4: C54 con el modo 584/984

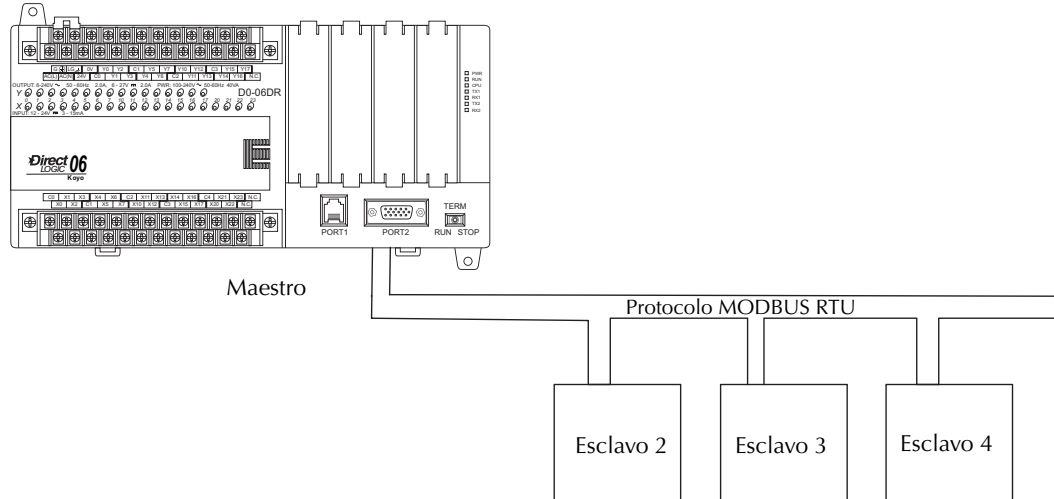
Encuentre la dirección MODBUS para C54. Dirección del PLC (Dec.)+Dir. inicial+ modo

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

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

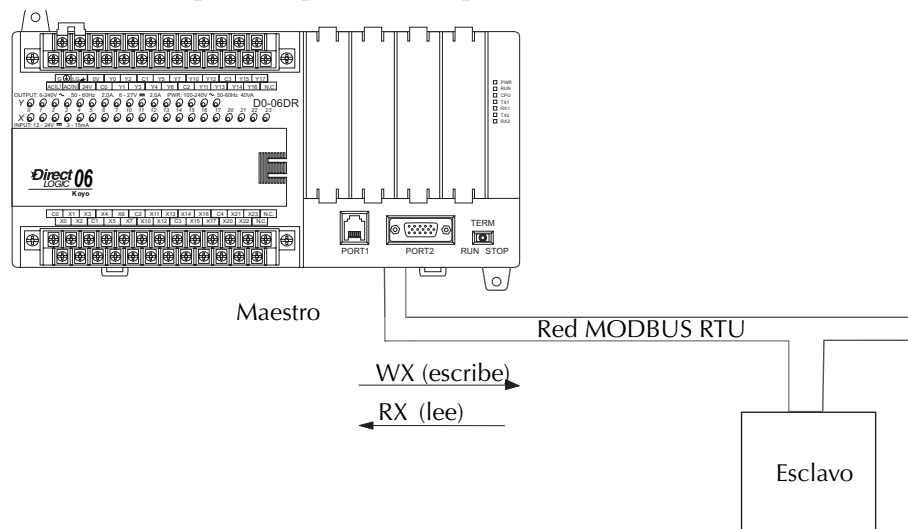
Operación del maestro en una red MODBUS RTU

Esta sección describe cómo el PLC DL06 puede comunicarse en una red de MODBUS como maestro (también es válido para *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.

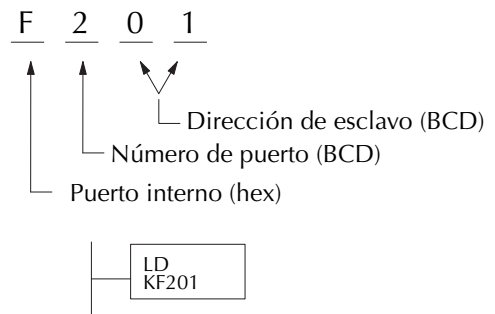


Paso 1: Identifique el número del puerto maestro y el número de cada esclavo

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).

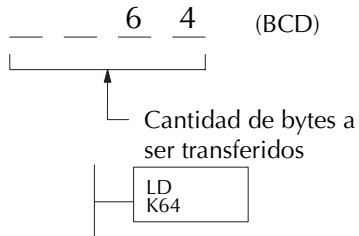


Paso 2: Cargue el no. de bytes a transferir:

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.

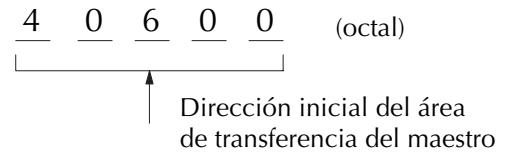
La siguiente tabla muestra los rangos de bytes para los varios tipos de productos *DirectLOGIC™*.



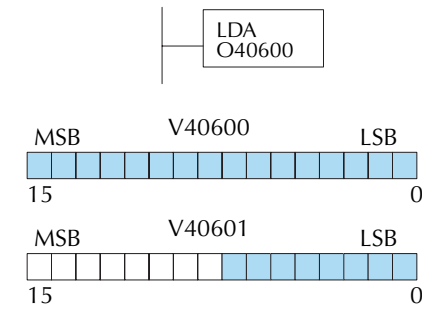
Memoria DL05 / 06 / 205 / 350 / 405	Bits por unidad	Bytes
Memoria V del valor corriente de temporizador o contador	16 16	2 2
Entradas (X, SP)	8	1
salidas (Y, C, estapas, bits de T/CT)	8	1
Memoria Scratch Pad	8	1
Estado de diagnóstico	8	1

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.

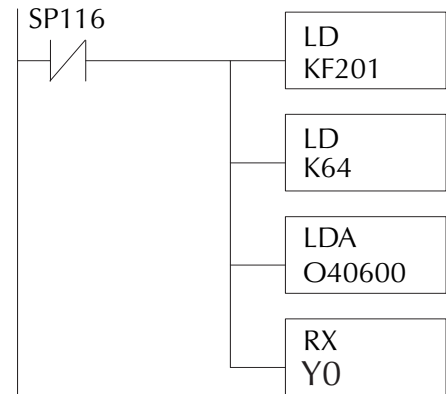


Paso 4: Especifique la memoria del esclavo

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

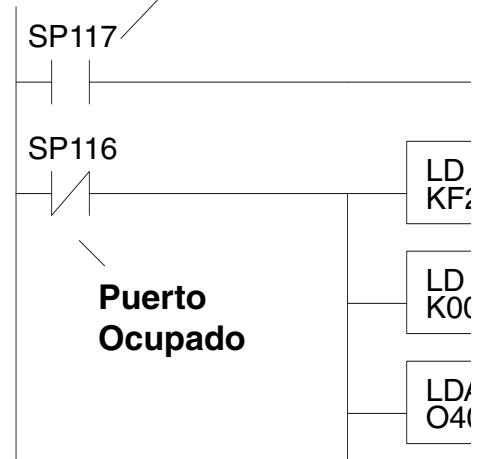
Si usted usa varias instrucciones WX y RX en el programa ladder, usted tiene que enclavar las rutinas para asegurarse que todas las rutinas se ejecutan completas.

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.

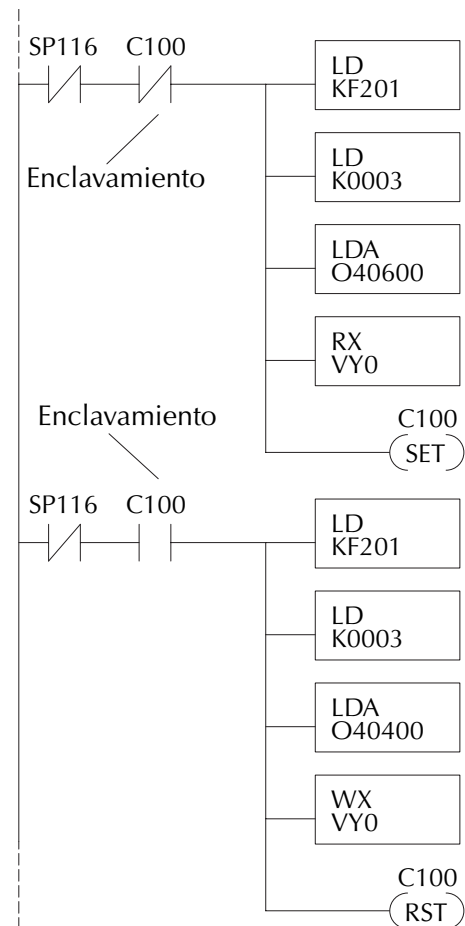
En el ejemplo a la derecha después que se ejecuta la instrucción RX, C100 se activa. Cuando el puerto ha terminado la tarea de comunicación, la segunda rutina se ejecuta y C100 es desactiva.

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



Puerto Ocupado



Configuración del puerto 2 del PLC DL06 para MODBUS

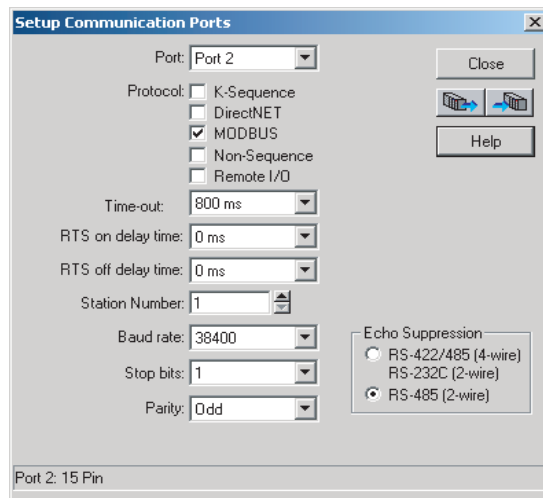
Esta sección describe cómo configurar el puerto 2 para establecimiento de una red con MODBUS RTU, usando *DirectSOFT*. Esto le permitirá conectar el sistema del PLC DL06 directamente con redes de MODBUS usando este protocolo. Los maestros MODBUS en la red deben ser capaces de generar comandos MODBUS 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: 1-90
	1 = ASCII	1-90
		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

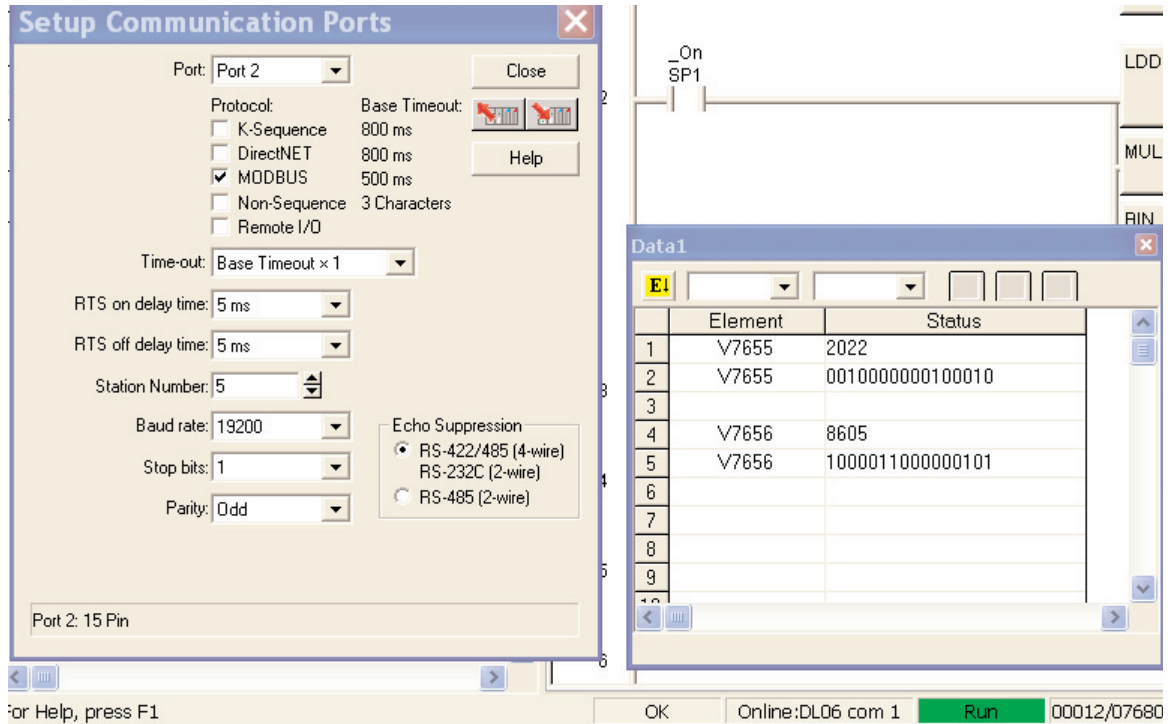
bbb - 110 19,200 baud,

x - 0 Modo de protocolo

aaa aaaa - 000 0101 Número de nodo 5 hexadecimal, que corresponde a 5 decimal

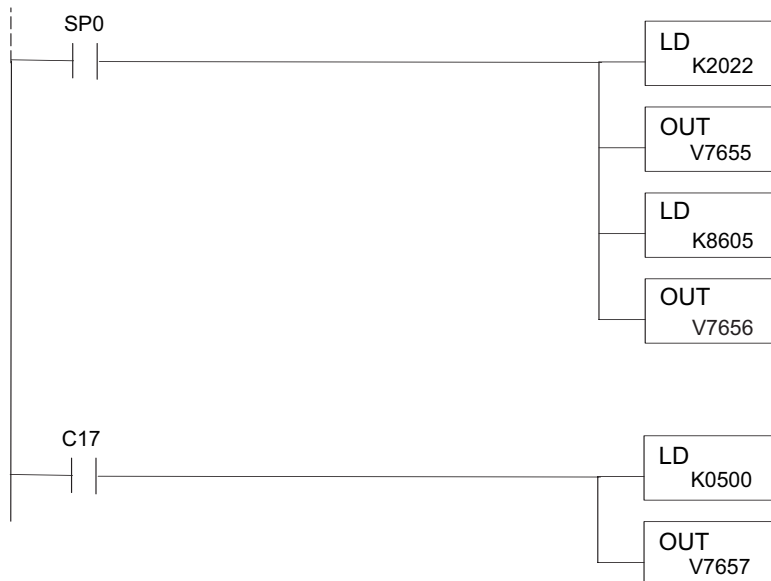
Ésto es, 1000 0110 0000 0101 equivalente a 8 6 0 5

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) :



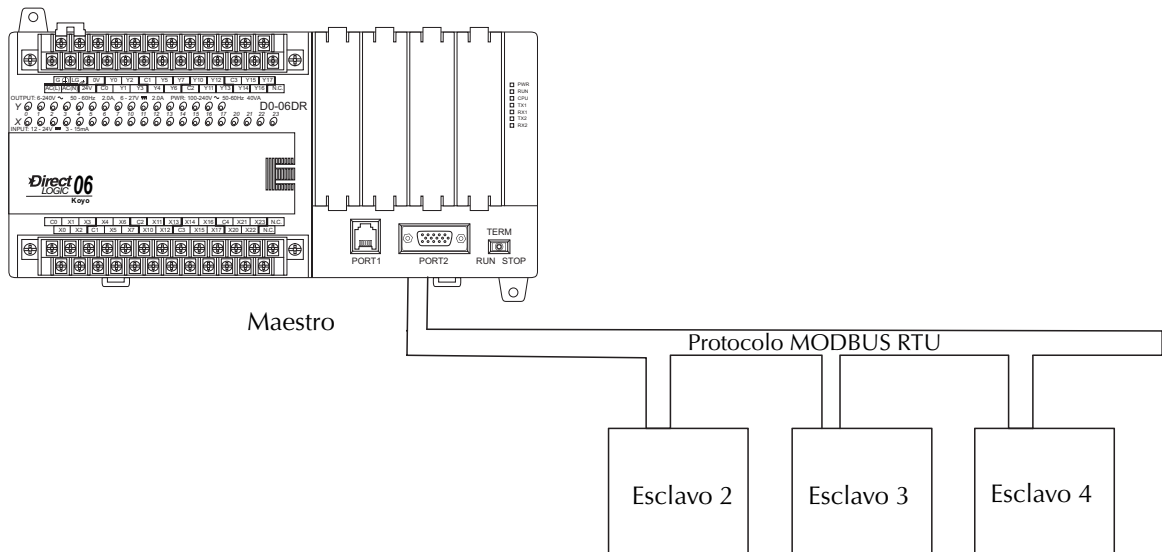
La configuración del puerto con *DirectSOFT* o con ladder se hace de la misma forma para usar las instrucciones RX /WX o MRX /MWX.

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.

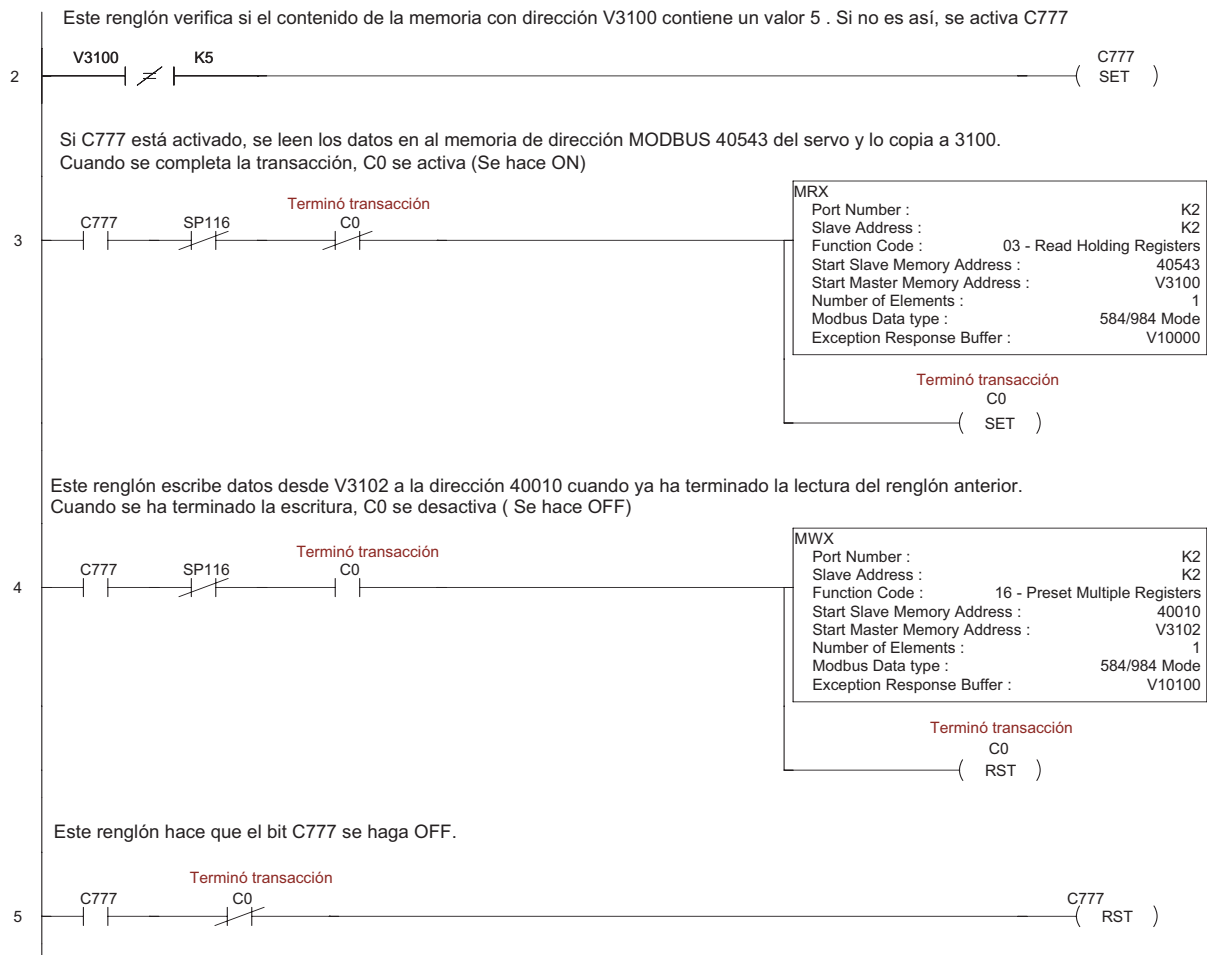
El puerto 2 del DL06 tiene dos contactos de relevadores especiales asociados a él (véa el apéndice D para relevadores especiales). Uno indica el "puerto ocupado" (SP116) y el otro indica "Error de comunicación del puerto" (SP117).

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.

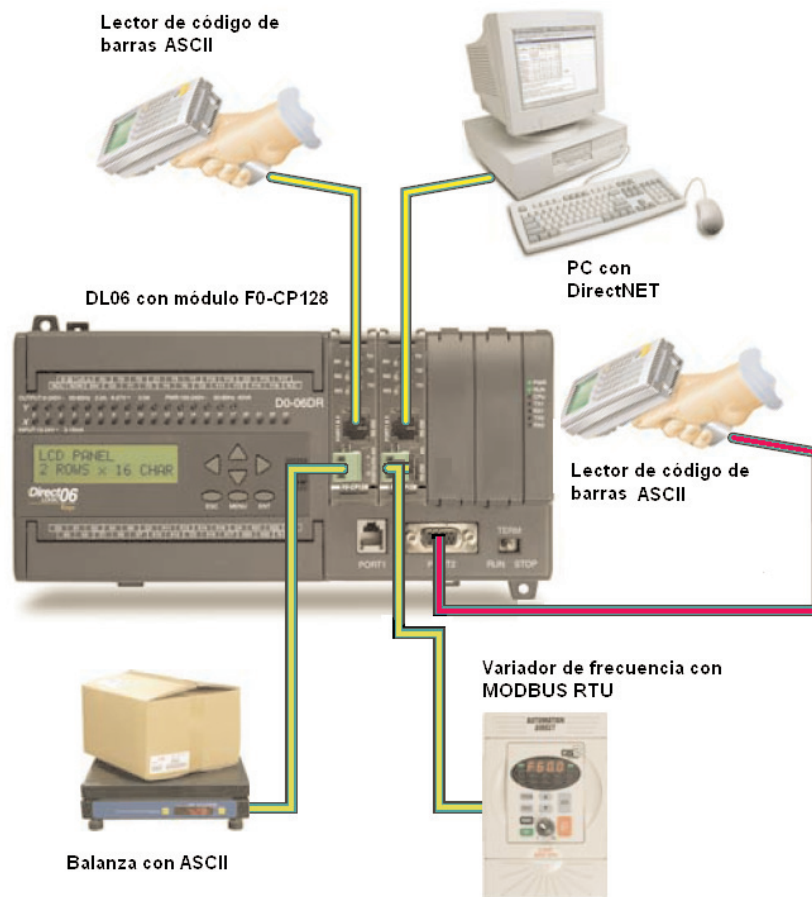
Observe que el código de error 02 dice la dirección de datos **no es reconocida**.

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.

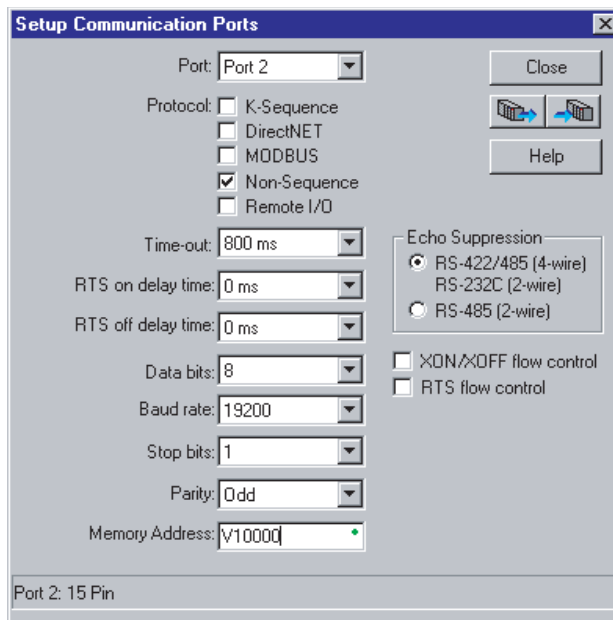
- **Echo Suppression:** Seleccione el botón de radio adecuado basado en la configuración usada en el puerto 2 (RS-232C, RS-422 o RS-485).

- **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.



K

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



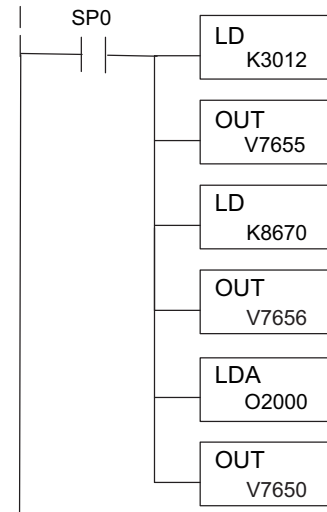
Palabra 2	Paridad	Stop Bits	Supresión de eco (válido solamente para DL06)	Tasa de Baud	Modo de Protocolo	
pps0 ebbb xaaa aaaa	pp	s	e	bbb	01110000 = No Control de flujo	
DL05/06: V7656	00 = Ninguna	0 = 1 bit	0 = RS-232C, RS-422 o RS-485 (4 wire)	000 = 300	01110001 = Control de flujo Xon/Xoff	
	10 = Odd	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:

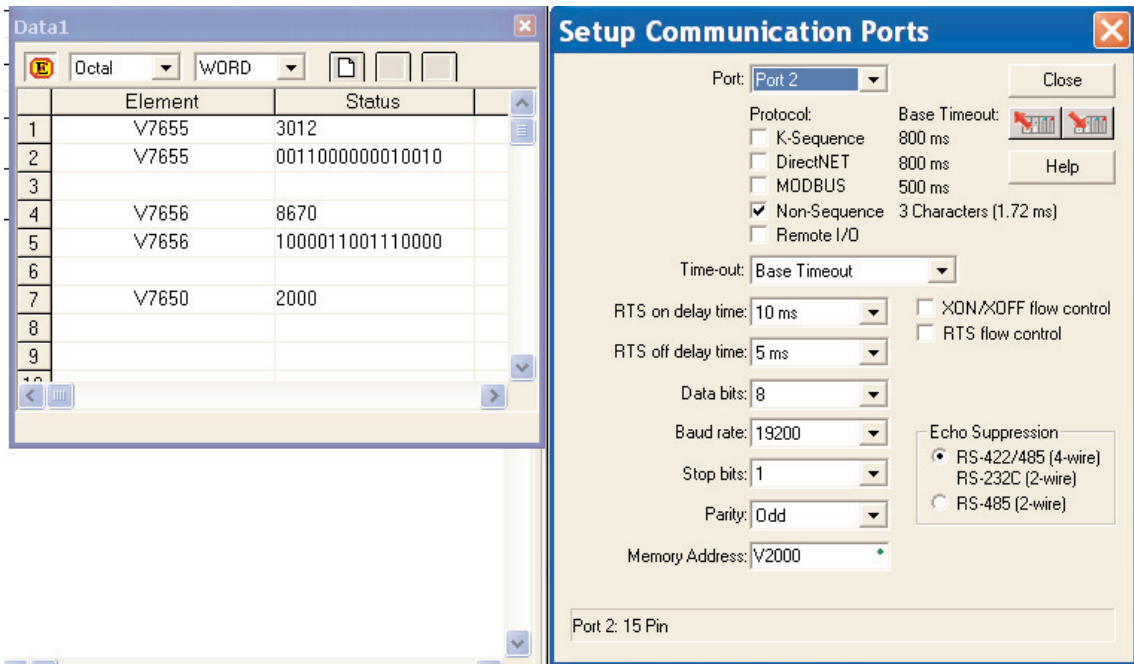
- RTS On-delay de 10ms,
- Base timeout x1,
- RTS Off-delay de 5ms,
- Paridad Odd,
- 1 bit de Stop,
- Supresión de eco para RS232-C/RS422,
- 19,200 Baud,
- 8 bits de datos,
- Memoria V intermediaria comenzando en V2000 y sin control de flujo



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

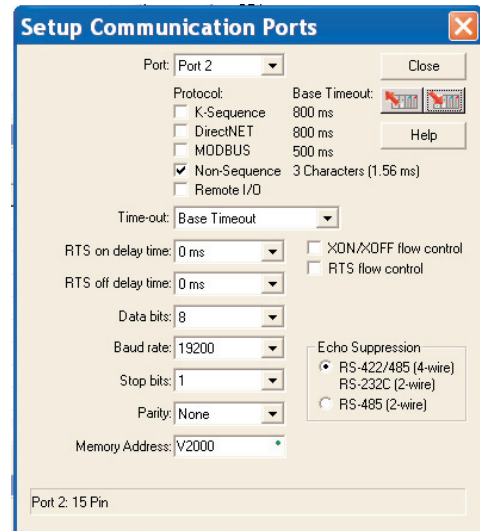
Un ejemplo simple puede ser lo siguiente;

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.

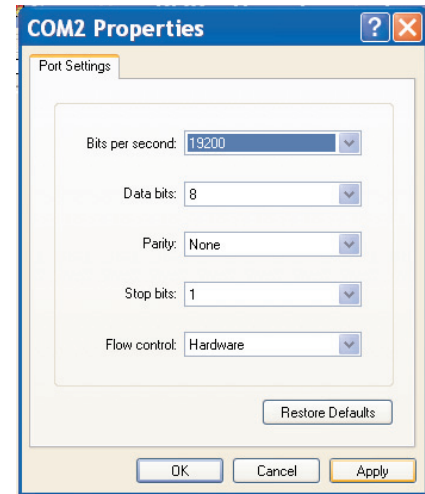
Se puede simular la subida de temperatura si usa Data View y no tiene aún configurado el módulo analógico o, si ya está instalado, puede inyectar una corriente tal que corresponda al valor de alarma deseado.

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

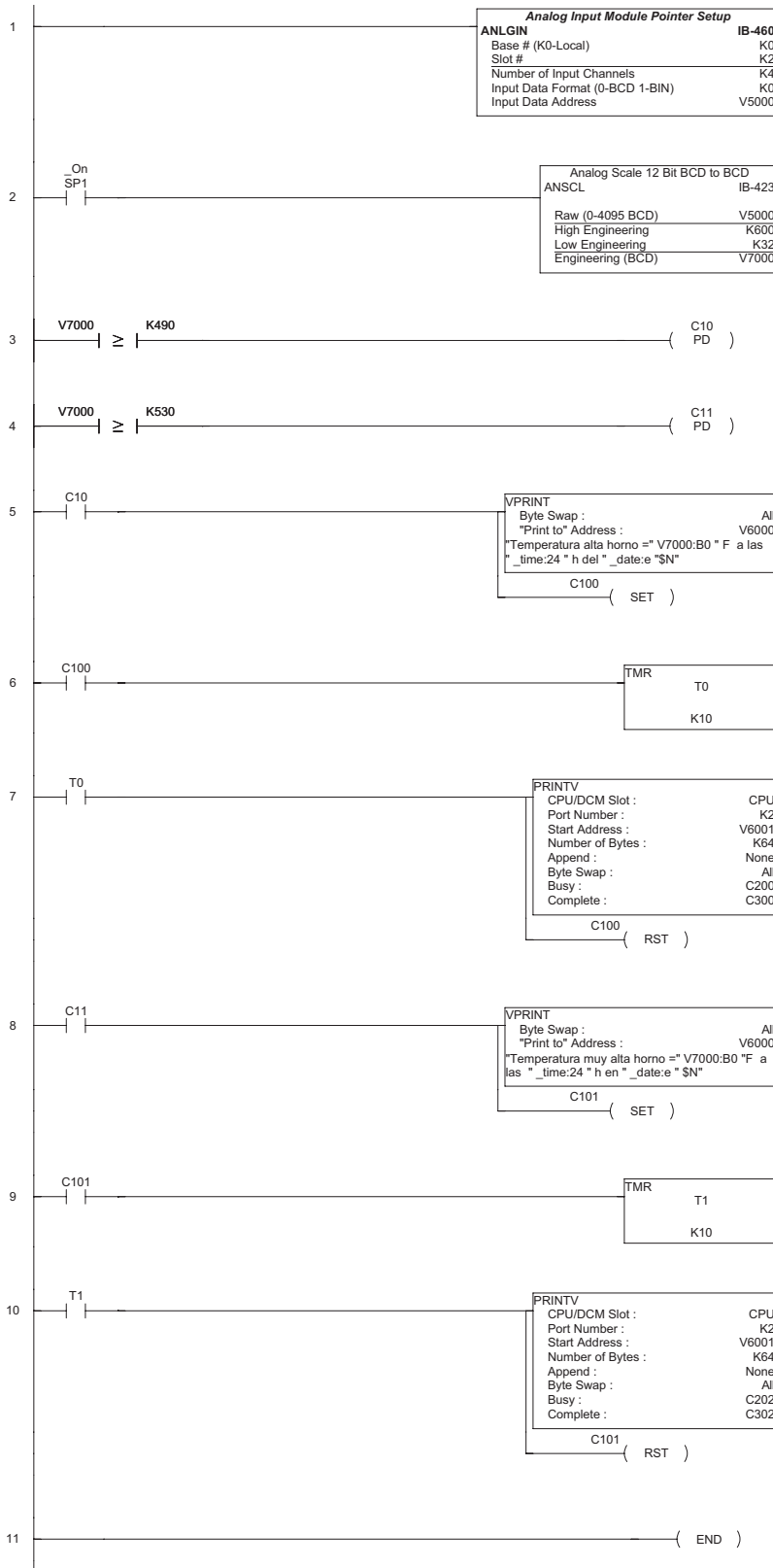
Etapas 2- EL PLC tiene un valor de temperatura en V7000 de 490 grados... El programa genera la primera línea

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

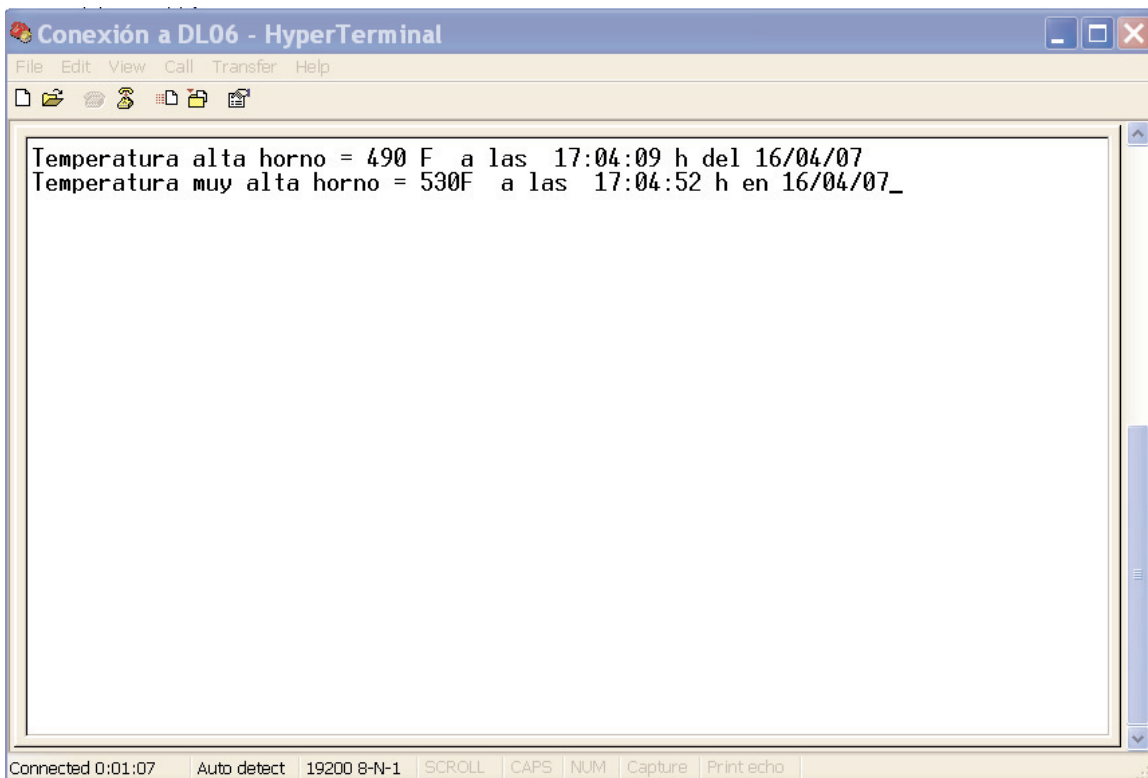
Al final de 1 segundo, se activa T0 lo que inicia la transmisión del contenido en V6000 y adyacentes para el aparato receptor. C100 se hace OFF

La instrucción VPRINT coloca el texto ASCII en la memoria V6000 y adyacentes. Re-escibe 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.