

# COMUNICACIONES CON **MODBUS** DEL VARIADOR **GS1**

---



## En este capítulo...

Lista de los parámetros de comunicación . . . . .	5-2
Topología de una red típica MODBUS . . . . .	5-3
Direcciones del variador GS1 (Sólo para leer) . . . . .	5-4
Comunicándose con PLCs DirectLogic . . . . .	5-6
Comunicándose con dispositivos de terceros . . . . .	5-16

## Lista de los parámetros de comunicación

La siguiente lista ofrece un resumen de los parámetros de comunicación del GS1. Para una lista completa de los parámetros del GS1, refiérase al CAPÍTULO 4.

Comunicaciones			
Parámetro del GS1	Descripción	Rango	Valor original
9-00	Dirección de esclavo	1 a 254	1
9-01	Velocidad de transmisión	0: 4800 baud 1: 9600 baud 2: 19200 baud	1
9-02	Protocolo de comunicación	0: Modo MODBUS ASCII, 7 bits de data, no paridad, 2 bits de parar 1: Modo MODBUS ASCII, 7 bits de data, paridad par, 1 bits de parar 2: Modo MODBUS ASCII, 7 bits de data, paridad impar, 1 bits de parar 3: Modo MODBUS RTU, 8 bits de data, no paridad, 2 bits de parar 4: Modo MODBUS RTU, 8 bits de data, paridad par, 1 bits de parar 5: Modo MODBUS RTU, 8 bits de data, paridad impar, 1 bits de parar	0
9-03	Tratamiento de falla en la transmisión	0: Indica falla y continua operando 1: Indica falla y hace RAMPa a parar 2: Indica falla y Para por fricción 3: No indica falla y continua operando	0
9-04	Detección de tiempo de espera de respuesta	0: Desactiva 1: Activa	0
9-05	Duración de tiempo de espera de respuesta	0.1 a 60.0 segundos	0.5
◆ 9-07	Bloqueo de parámetros	0: Todos los parámetros pueden ser configurados y leídos 1: Todos los parámetros son solo para leer	0
9-08	Restablecer valores originales de fábrica	99: Restablece todos los parámetros a los valores originales de fábrica	0
◆ 9-11	Parámetro de Transferencia de Bloque 1	0-00 a 8-01, 9-99	9-99
◆ 9-12	Parámetro de Transferencia de Bloque 2	0-00 a 8-01, 9-99	9-99
◆ 9-13	Parámetro de Transferencia de Bloque 3	0-00 a 8-01, 9-99	9-99
◆ 9-14	Parámetro de Transferencia de Bloque 4	0-00 a 8-01, 9-99	9-99
◆ 9-15	Parámetro de Transferencia de Bloque 5	0-00 a 8-01, 9-99	9-99
◆ 9-16	Parámetro de Transferencia de Bloque 6	0-00 a 8-01, 9-99	9-99
◆ 9-17	Parámetro de Transferencia de Bloque 7	0-00 a 8-01, 9-99	9-99
◆ 9-18	Parámetro de Transferencia de Bloque 8	0-00 a 8-01, 9-99	9-99
◆ 9-19	Parámetro de Transferencia de Bloque 9	0-00 a 8-01, 9-99	9-99
◆ 9-20	Parámetro de Transferencia de Bloque 10	0-00 a 8-01, 9-99	9-99

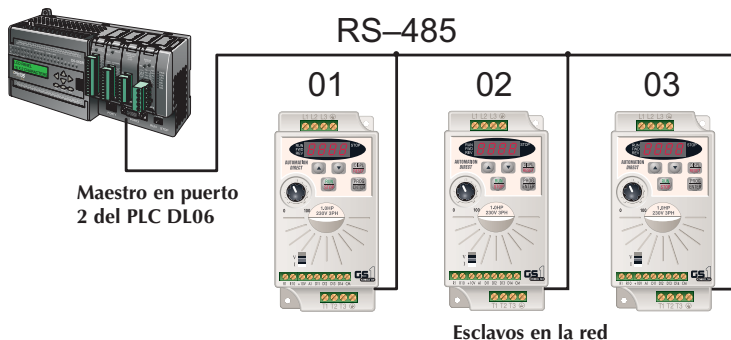
◆ Estos parámetros pueden ser ajustados durante el modo run (FUNCIONAR)

## Resumen de Parámetros de Comunicación (continuación)

Comunicaciones (continuado)			
Parámetro del GS1	Descripción	Rango	Valor original
◆ 9-26	Velocidad de referencia RS485	0.0 a 400.0 Hz	60.0
◆ 9-27	Comando RUN (Partir)	0: Parar                      1:Partir	0
◆ 9-28	Comando de dirección del motor	0: Hacia Delante            1:Reversa	0
◆ 9-29	Falla externa	0: Ninguna falla            1:Falla externa	0
◆ 9-30	Restablecer fallas	0: Ninguna acción        1:Restablecer falla	0
◆ 9-31	Comando de JOG (PULSAR)	0: Parar                      1:Pulsar	0
9-41	Número de Serie GS	1: GS1 2: GS2 3: GS3 4: GS4	##
9-42	Información del modelo del fabricante	0: GS1-10P2 (120V, monofásico, 0.25HP) 1: GS1-10P5 (120V, monofásico, 0.5HP) 2: GS1-20P2 (230V, mono-trifásico, 0.25HP) 3: GS1-20P5 (230V, mono-trifásico, 0.5HP) 4: GS1-21P0 (230V, mono-trifásico, 1HP) 5: GS1-22P0 (230V, 3ph, 2HP)	##

◆ Estos parámetros pueden ser ajustados durante el modo de RUN (Funcionar).

## Topología de una red típica MODBUS



## Memorias de comunicación del variador GS1

El variador de frecuencia Serie GS1 tiene direcciones de memoria que se usan para supervisar el variador de frecuencia. La siguiente lista incluye las direcciones de memoria y definiciones de valores.

### Supervisor de estado 1

Dir. Mem: 2100<sub>H</sub>(48449)

#### Códigos de Errores:

- |                             |   |
|-----------------------------|---|
| 0: No ocurrió falla         | 11: Falla de protección de componentes (HPF)          |
| 1: Sobrecorriente (oc)      | 12: Sobrecorriente durante la aceleración (OCA)       |
| 2: Sobretensión (ov)        | 13: Sobrecorriente durante la desaceleración (Ocd)    |
| 3: Sobrecalentado (oH)      | 14: Sobrecorriente durante régimen estable (Ocd)      |
| 4: Sobrecarga (oL)          | 16: Bajo voltaje (Lv)                                 |
| 5: Sobrecarga 1 (oL1)       | 18: Bloque-Base externo (bb)                          |
| 6: Sobrecarga 2 (oL2)       | 19: Falla de auto ajuste de acel/desaceleración (cFA) |
| 7: Falla externa (EF)       | 20: Código de protección de software (codE)           |
| 8: Falla de la CPU 1 (CF1)  |   |
| 9: Falla de la CPU 2 (CF2)  |   |
| 10: Falla de la CPU 3 (CF3) |   |

### Supervisor de Estado 2

Dir. Mem: 2101<sub>H</sub>(48450)

Dirección de memoria del GS1  
(hexadecimal)

Datos de memoria del GS1 (binario)

2001	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits
	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	
	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1	Valores de Bits (decimal)

Dirección de Memoria 2001		
Bit(s) de Dirección	Valor de Bit(s) Binario (Decimal)	Estado del variador de frecuencia
0 y 1	00 (0)	Operación del variador de frecuencia parada (STOP/PARAR)
	01 (1)	Transición de Correr a Parar
	10 (2)	En espera
	11 (3)	Operación del variador de frecuencia funcionando (RUN/Partir)
2	1 (4)	JOG (PULSAR) activo
3 y 4	00 (0)	Dirección del motor hacia delante (FWD/HACIA DELANTE))
	01 (8)	Transición de REV (REVERSA) a FWD (HACIA DELANTE)
	10 (16)	Transición de FWD (HACIA DELANTE) a REV (REVERSA)
	11 (24)	Dirección del motor en reversa (REV/REVERSA)
5	1 (32)	Origen de frecuencia determinada por interfase de comunicación (P4-00 = 5)
6	1 (64)	Origen de frecuencia determinada por conexión AI (P4-00 = 2, 3, o 4)
7	1 (128)	Origen de operación determinada por interfase de comunicación (P3-00 = 3 or 4)
8	1 (256)	Parámetros están bloqueados (P9-07 = 1)
9 a 15	N/A	Reservado

**Comando de frecuencia F (XXX.X) Dir. Mem: 2102<sub>H</sub>(48451)**

Localización de memoria para el ajuste de la frecuencia del variador de frecuencia.

**Frecuencia de salida H (XXX.X) Dir. Mem: 2103<sub>H</sub>(48452)**

Localización de memoria de la frecuencia de operación corriente presente en las conexiones T1, T2, y T3.

**Corriente de salida A (XXX.X) Dir. Mem: 2104<sub>H</sub>(48453)**

Localización de memoria para la corriente de salida presente en las conexiones T1, T2, y T3.

**Voltaje de la barra de CC d (XXX.X) Dir. Mem: 2105<sub>H</sub>(48454)**

Localización de memoria para el voltaje de la barra CC.

**Voltaje de salida U (XXX.X) Dir. Mem: 2106<sub>H</sub>(48455)**

Localización de memoria para el voltaje de salida presente en las conexiones T1, T2, y T3.

**Velocidad del motor Dir. Mem: 2107<sub>H</sub>(48456)**

Localización de memoria para la velocidad corriente estimada del motor.

**Frecuencia a escala (Palabra baja) Dir. Mem: 2108<sub>H</sub>(48457)**

Localización de memoria para el resultado de la frecuencia de salida x P8-01 (palabra baja).

**Frecuencia a escala (Palabra alta) Dir. Mem: 2109<sub>H</sub>(48458)**

Localización de memoria para el resultado de la frecuencia de salida x P8-01 (palabra alta).

**% de carga del variador Dir. Mem: 210B<sub>H</sub>(48460)**

Localización de memoria para la cantidad de carga en el variador de frecuencia. (Corriente de salida ÷ Corriente nominal para el variador de frecuencia) x 100.

**Versión de firmware Dir. Mem: 2110<sub>H</sub>(48465)**

## Comunicándose con los PLCs *DirectLOGIC*

Los siguientes pasos explican como conectar y comunicarse con los variadores de frecuencia Serie GS1 usando PLCs *DirectLOGIC*.

### Paso 1: Escoja la CPU apropiada

Los variadores de frecuencia Serie GS1 pueden comunicarse con las siguientes CPUs *DirectLOGIC* usando comunicaciones MODBUS.

- DL05      • DL06      • DL250
- DL260    • DL350    • DL450

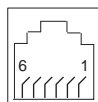
### Paso 2: Haga las conexiones

El puerto de comunicación del variador GS1 puede acomodar una conexión de red RS 485. Un diagrama del puerto de comunicación del variador GS1 está mostrado a la derecha.

Un cable de red RS-485 puede extenderse hasta 1000 metros (o cerca de 4000 pies). Sin embargo, casi todos los PLCs de *DirectLOGIC* requieren un módulo FA-ISOCAN (adaptador de red de RS 232C a RS422/485) para poder hacer este tipo de conexión.

Use el siguiente diagrama de alambrado para conectar el PLC *DirectLOGIC* a un variador de frecuencia Serie GS1 con una interfase RS-485.

Puerto de comunicación serial RJ-12



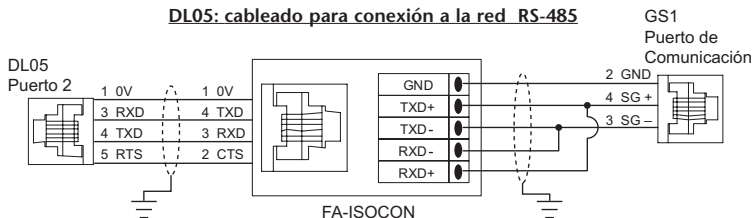
Interfase RS-485

- 1: +17V
- 2: GND
- 3: SG-
- 4: SG+

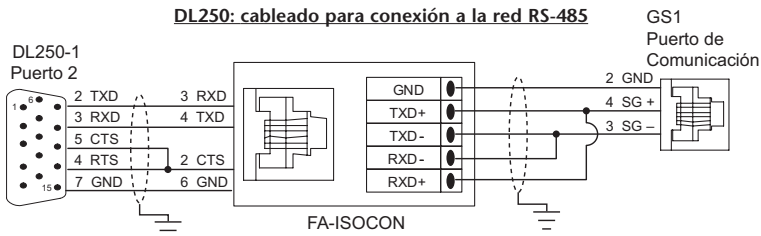


*Nota: Si Ud. usa un módulo FA-ISOCAN en su conexión, asegúrese que los puentes estén configurados para comunicación del tipo RS485.*

#### DL05: cableado para conexión a la red RS-485

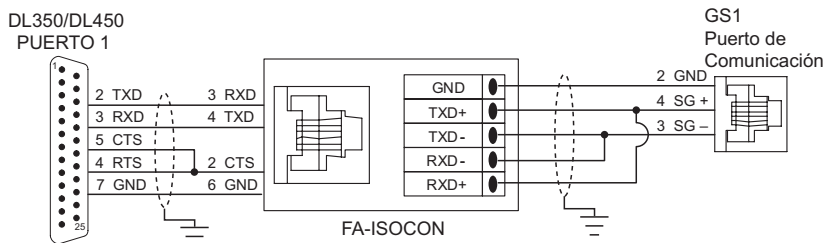


#### DL250-1: cableado para conexión a la red RS-485



Continúa en la próxima pagina

**DL350/DL450: RS-485; Alambrado para conexión a la red RS-485**



**Paso 3: Configure los parámetros del variador de frecuencia**

Los siguientes parámetros necesitan ser configurados según se muestra para poder comunicarse adecuadamente.

- 3-00: 3 or 4 – La operación es determinada por la interfase RS485. La tecla STOP (PARAR) está activada (3) o desactivada (4).
- 4-00: 5 – La frecuencia es determinada por la interfase de comunicación RS485.
- 9.00: xx – La dirección de comunicación en el rango 1-254 (única para cada dispositivo, vea P9.00)
- 9.01: 1 – 9600 baud de velocidad de transmisión de datos.
- 9.02: 5 – Modo MODBUS RTU <8 bits de datos, paridad impar, 1 bitio de parar.>



*Nota: La lista previa de configuración de parámetros es lo mínimo requerido para comunicarse con un PLC **DirectLOGIC**. Puede haber otros parámetros que necesiten ser configurados para satisfacer las necesidades de su aplicación.*

**Step 4: Configure las CPUs *DirectLOGIC***

La CPU **DirectLOGIC** debe ser configurada para comunicarse con los variadores de frecuencia Serie GS1. Esta configuración incluye configurar el puerto de comunicación y añadir instrucciones a su programa de lógica.

La configuración de todos los CPUs **DirectLOGIC** es muy similar. Sin embargo, puede haber algunas diferencias sutiles entre CPUs. Refiérase al Manual del usuario apropiado de la CPU para información específica de su CPU **DirectLOGIC**.

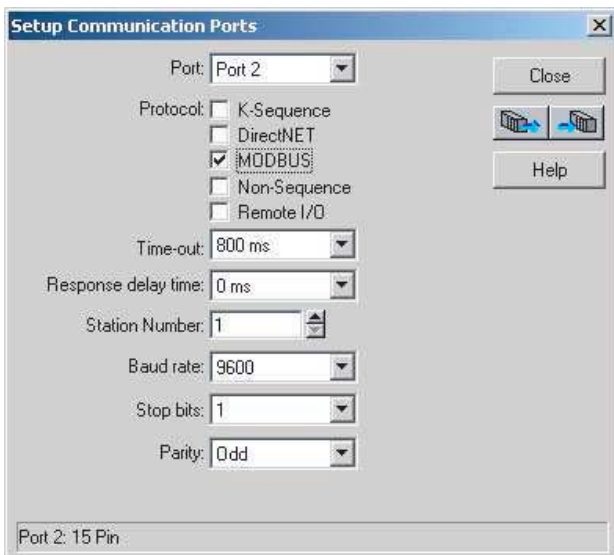


*Nota: Para instrucciones sobre la configuración del puerto en la CPU para MODBUS específico, refiérase al Manual de usuario apropiado de la CPU.*

### Configuración del puerto de la CPU *Direct*LOGIC como MODBUS

El siguiente ejemplo de configuración es específico para la CPU DL250-1. Refiérase al Manual de usuario apropiado de la CPU para información específica de su CPU *Direct*LOGIC.

- En *Direct*SOFT, escoja el menú "PLC", luego "Setup", luego "Secondary Comm Port".
- De la lista de **Port (puerto)**, escoja "Port 2".
- Para el **Protocol** (Protocolo), seleccione "MODBUS".



- En la lista de **Timeout (Tiempo de espera para respuesta)**, seleccione "800 ms".
- El valor **Response Delay Time (Demora de Tiempo para responder)** debe ser "0 ms".
- El **Station Number (Número de estación)** debe ser ajustado a "1" para hacer que el CPU DL250-1 sea el maestro en MODBUS.



*Nota: Las instrucciones de red del DL250-1 usadas en modo maestro darán acceso solamente a esclavos 1 al 90. Cada esclavo debe tener un número único.*

- El **Baud Rate (Margen de Baud)** debe ser ajustado a "9600".
- En la lista de **Stop Bits**, escoja "1".
- En la lista de **Parity (Paridad)**, escoja "Impar".



**Programación de la transmisión MODBUS en PLCs DirectLOGIC**

La configuración para todas las CPUs *DirectLOGIC* es muy similar. Sin embargo, puede haber algunas diferencias sutiles entre las CPUs. Refiérase al Manual de usuario apropiado de la CPU para información específica de su CPU *DirectLOGIC*.

El siguiente programa de escalera muestra algunos ejemplos de cómo controlar el variador de frecuencia a través de la red MODBUS RTU. El variador de frecuencia debe ser configurado y probado para comunicaciones antes de ser conectado a una carga.



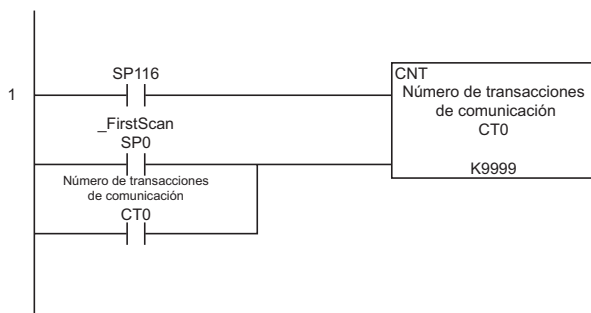
**ADVERTENCIA: Nunca debe conectarse un variador de frecuencia a un motor hasta que el programa de comunicación aplicable haya sido probado.**



*Nota: Este programa es ofrecido solamente con el propósito de ilustración y no pretende ser usado en una aplicación verdadera.*

En varias de las aplicaciones de variadores de frecuencia, la interferencia electromagnética puede a veces causar errores de comunicación frecuentes, de corta duración. A menos que el ambiente de la aplicación sea perfecto, ocasionalmente ocurrirán errores de comunicación. Para poder distinguir entre estos errores no-fatales y un error de comunicación genuino, tal vez quiera usar las instrucciones según se muestran en los renglones 1 al 4. El renglón 1 supervisa el número de veces que el PLC trata de comunicarse con el variador de frecuencia. Cuando la comunicación del PLC tiene éxito, el SP116 contará y el SP117 no contará. Cuando el conteo llegue a 9999, el contador vuelve a cero y reanuda el conteo.

DirectSOFT

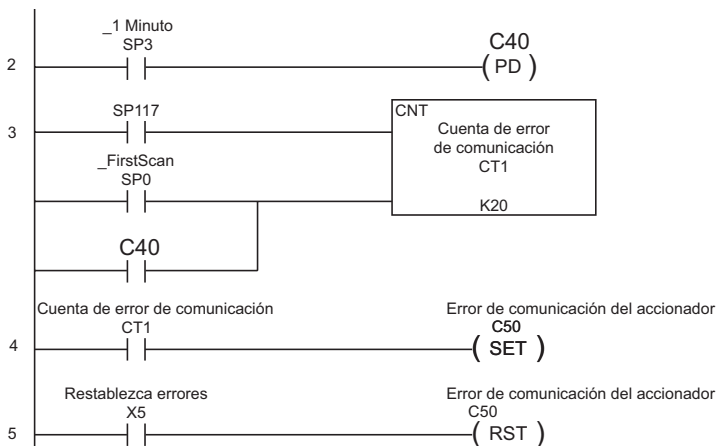


*Nota: SP116 y SP117 son relevadores especiales en la CPU DirectLOGIC que supervisan las comunicaciones del PLC. SP116 estará encendido cuando el puerto 2 se esté comunicando con otro dispositivo. SP117 estará encendido cuando el puerto 2 ha encontrado un error de comunicación.*

(Continúa en la próxima pagina)

**Programación de la transmisión MODBUS en PLCs DirectLOGIC(continuación)**

Los renglones 2 al 5 supervisan las veces que el PLC falla al comunicarse con el variador de frecuencia. Estas instrucciones colocan ON el bit C50 (para usarse como alarma o para apagar) basado en el número de veces que el bit SP117 está activo en un minuto. En este ejemplo el bit C50 será ajustado si el número de errores en un minuto excede 20



**Transferencia en bloque**

Hay un grupo de parámetros de transferencia en bloque disponibles en el variador de frecuencia GS1 (P9.11 a P9.20). Este bloque de parámetros contiguo puede ser usado para "agrupar" parámetros misceláneos a través del variador de frecuencia. Esto le permite transferir estos parámetros misceláneos en un bloque en vez de tener que usar comandos múltiples de WX o RX.

Por ejemplo: Si necesita cambiar la compensación de deslizamiento (P2-01), tiempo de aceleración (P1-01), tiempo de desaceleración (P1-02), y multi-velocidad (P5-01), esto típicamente tomaría tres comandos WX distintos porque los parámetros no son contiguos. Si usted ajusta P9-11 a P2-01, P9-12 a P1-01, P9-13 a P1-02, y P9-14 a P5-01, entonces todos estos parámetros pueden ser controlados usando solamente un comando WX.

El renglón 6 escribe los valores desde V2000 a V2023 a los parámetros del variador de frecuencia P9-11 a P9-20. En el bloque WX, el valor es V4413. 4413 es un numero octal como todas las direcciones en los PLCs **DirectLOGIC**. Si convierte el octal 4413 a hexa, obtiene 90B. 90B es la dirección del parámetro P9-11.

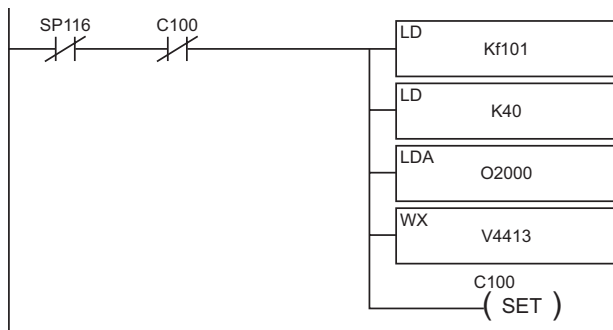


*Nota: Refiérase a su Manual del usuario del PLC para más información específica sobre direcciones del MODBUS y conversión de direcciones.*

(Continúa en la próxima pagina)

**Programación de la transmisión MODBUS en PLCs DirectLOGIC(continuación)**

Si solo desea controlar el partir/parar y la referencia de velocidad del variador de frecuencia, simplemente cambie el segundo comando LD de este renglón al comando K4 y WX al V4432. Entonces V2000 será su localización de referencia de velocidad y V2001 será su localización de partir/parar.

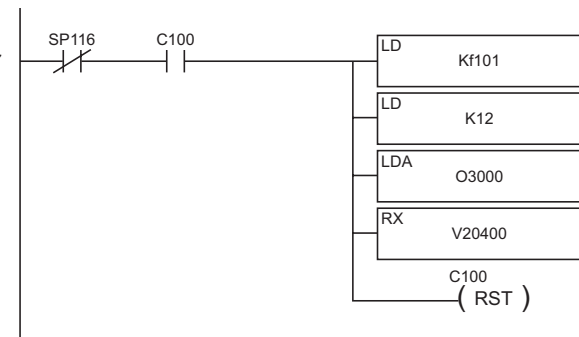


El renglón 7 se usa para leer el estado del variador GS1. Estas instrucciones leen los valores de la direcciones de memoria del GS1, 2100 a 2105, y coloca los valores en las direcciones de memoria del PLC, V3000 a V3005.

Note que el número en el bloque RX - V20400. V20400 es un número octal así como son todas las referencias de direcciones en los PLCs **DirectLOGIC**. El octal 20400 convertido a hexadecimal resulta 2100 - la primera dirección de memoria para el variador de frecuencia GS1.



*Nota: Refiérase a su manual de usuario del PLC para información más específica sobre las direcciones MODBUS y conversiones de direcciones.*



### Programación alternativa del PLC para la red MODBUS

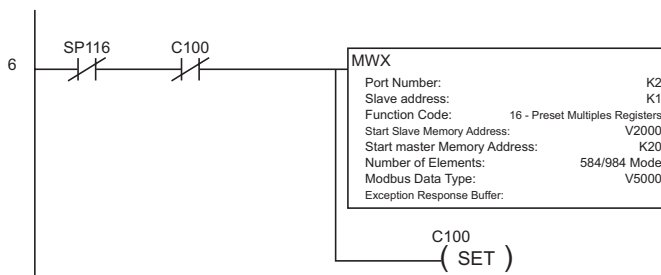
Los comandos para LEER y ESCRIBIR para las CPUs DL260 y DL06 son distintos a las de otras CPUs **DirectLOGIC**. Los renglones 6 y 7 se muestran en la siguiente figura de acuerdo a como se relacionan a los CPUs DL260 y DL06.

El renglón 6 escribe los valores del V2000 a V2023 a los parámetros del variador de frecuencia 9.11 a 9.30. En el marco del MWX, el valor del inicio de la dirección esclava es 42316. 42316 es un número decimal del MODBUS. Para convertir el decimal 42316 a hexa, primero tiene que restar 40001, y entonces convertir el restante a hexa (90B). 90B es la dirección para el parámetro 9.11.



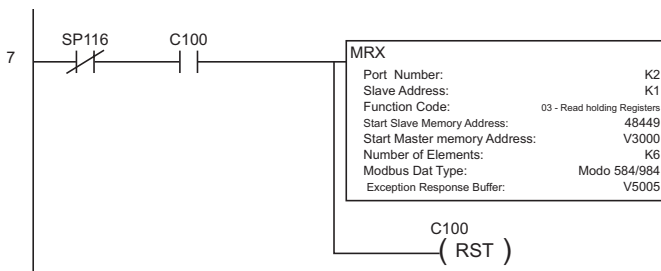
*Nota: Refiérase a su Manual del Usuario del PLC para información mas específica de las direcciones del MODBUS y conversiones de direcciones.*

Si solamente quiere controlar el partir/parar y la referencia de velocidad del variador de frecuencia, simplemente cambie el número de elementos a K2 y la dirección de la memoria esclava a 42331. Entonces V2000 será su localización de referencia de velocidad y V2001 será su localización de partir/parar.



El renglón 7 se usa para leer el estado del variador de frecuencia GS1. Estas instrucciones leen los valores de las direcciones de estado del GS1, 2100 a 2105, y coloca los valores en las direcciones de memoria del PLC, V3000 a V3005.

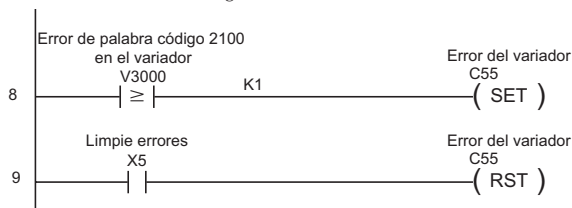
El inicio de la dirección de memoria esclava en el bloque MRX es 48449. 48449 es un número decimal de MODBUS. Para convertir el decimal 48449 a hexa, debe primero restar 40001, y luego convertir el restante a hexa (2100). 2100 es la dirección para el supervisor de estado del GS1.



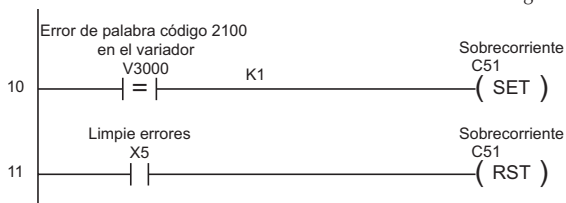
(Continuado en la siguiente pagina)

**Programación de la transmisión MODBUS en PLCs DirectLOGIC (continuación)**

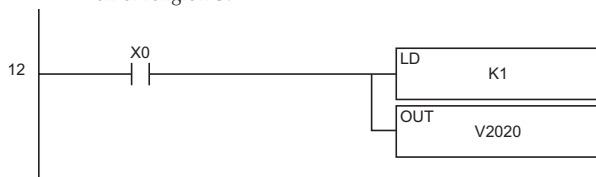
El renglón 8 se usa para colocar ON el bit C55 si el variador de frecuencia tiene un error. El renglón 9 vuelve a OFF el bit C55.



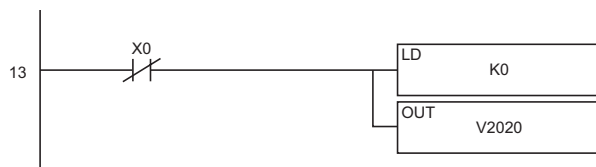
El renglón 10 se usa para colocar ON el bit C51 si el variador de frecuencia tiene un error específico. En este ejemplo C51 será colocado ON si el variador de frecuencia tiene un error de sobrecorriente. El renglón 11 coloca OFF el bit C51.



El renglón 12 carga un valor 1 dentro de los parámetros del variador de frecuencia P9-27. Esta es la señal de partir. V2020 es el 17mo registro de la memoria-V en el bloque de 20 en el que está escribiendo en las instrucciones WX en el renglón 5.



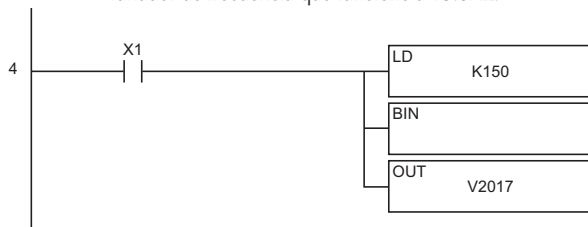
El renglón 13 carga un valor de 0 dentro del parámetro del variador de frecuencia P9-27. Esta es la señal de parar. V2020 es el 17mo registro de la memoria-V en el bloque de 20 que está siendo escrito en la instrucción WX en el renglón 5.



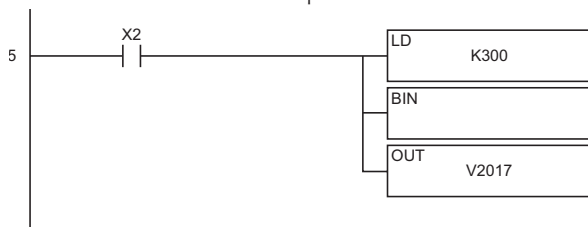
(Continúa en la siguiente pagina)

### Programación de la transmisión MODBUS en PLCs DirectLOGIC (continuación)

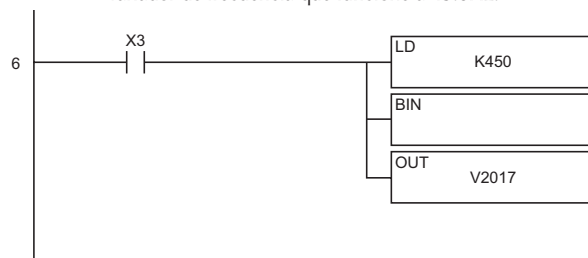
El renglón 14 carga el valor decimal de 150 dentro del parámetro del variador de frecuencia P9-26. La instrucción BIN convierte BCD/HEXA a decimal. Esto le dice al variador de frecuencia que funcione a 150.0Hz.



El renglón 15 carga el valor decimal 300 dentro del parámetro del variador de frecuencia P9-26. La instrucción BIN convierte BCD/HEXA a decimal. Esto le dice al variador de frecuencia que funcione a 30.0Hz.



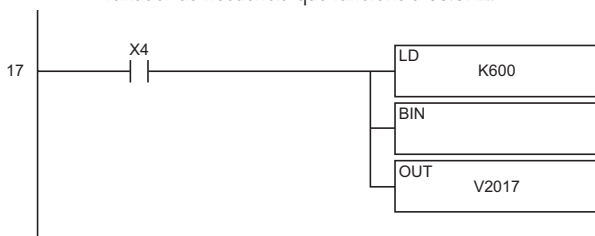
El renglón 16 carga el valor decimal 450 dentro del parámetro del variador de frecuencia P9-26. La instrucción BIN convierte BCD/HEXA a decimal. Esto le dice al variador de frecuencia que funcione a 45.0Hz.



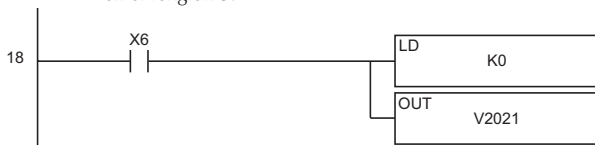
(Continúa en la próxima página)

**Programación de la transmisión MODBUS en PLCs DirectLOGIC (continuación)**

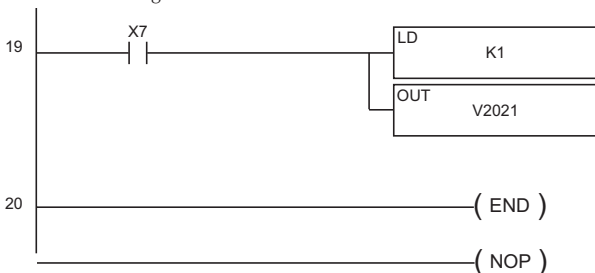
El renglón 17 carga el valor decimal 600 dentro del parámetro del variador de frecuencia P9-26. La instrucción BIN convierte BCD/HEXA a decimal. Esto le dice al variador de frecuencia que funcione a 60.0Hz.



El renglón 18 configura el movimiento del variador de frecuencia a giro hacia adelante cargando un valor 0 dentro del parámetro P9-28. V2021 es el 18vo registro de memoria-V en el bloque de 20 el cual se le está escribiendo en la instrucción WX en el renglón 5.



El renglón 19 configura el movimiento del variador de frecuencia a giro inverso cargando un valor de 1 dentro del parámetro P9-28. V2021 es el 18vo registro de memoria-V en el bloque de 20 el cual se le está escribiendo en la instrucción WX en el renglón 5.



## Comunicándose con dispositivos de terceros

El puerto de comunicación del variador GS1 puede conectarse a una red RS485, como esclavo de la red. El diagrama del puerto de comunicación del variador GS1 puede ser visto en la figura a la derecha. Un cable de red RS-485 puede cubrir hasta 1000 metros (o hasta aprox. 4000 pies).

La dirección de esclavo del variador de frecuencia Serie GS1 está especificada en P9-00. El dispositivo de terceros controla cada uno de los variadores de frecuencia de acuerdo con su dirección de esclavo.

El variador de frecuencia Serie GS1 puede ser configurado para comunicarse en una red MODBUS común usando los siguientes modos de transmisión: ASCII o RTU. Usando el parámetro de Protocolo de Comunicación (P9-02), puede seleccionarse el modo deseado, bits de datos, paridad y bits de parar. El modo y parámetros de la comunicación serial deben ser los mismos para todos los dispositivos en una red MODBUS.

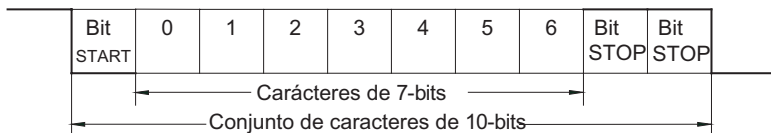




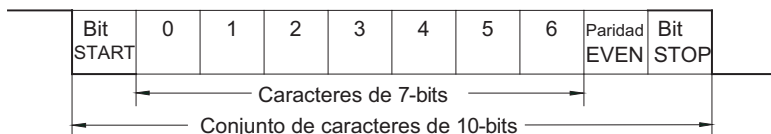
## Formato de Datos

### Modo ASCII: Marco de caracteres de 10-bits (Para caracteres de 7-bits):

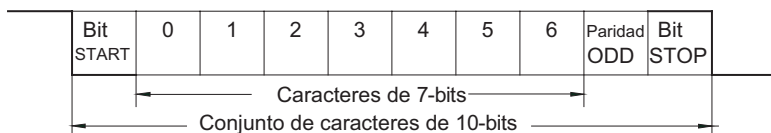
9-02 = 0 (7 bits de datos, ninguna paridad, 2 bits de parar)



9.02 = 01 (7 bits de datos, paridad par, 1 bits de parar)

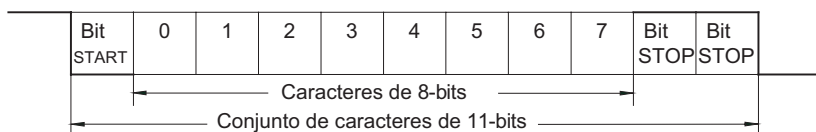


9-02 = 2 (7 bits de datos, paridad impar, 1 stop bit)

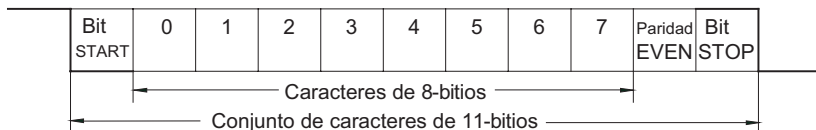


### Modo RTU: Marco de caracteres de 11-bits (Para caracteres de 8-bits):

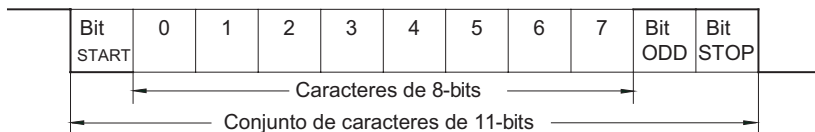
9-02 = 3 (8 bits de datos, ninguna paridad, 2 bits de parar)



9-02 = 4 (8 bits de datos, paridad par, 1 bits de parar)



9-02 = 5 (8 bits de datos, paridad impar, 1 bits de parar



### Protocolo de comunicación MODBUS

#### Modo ASCII:

STX	Carácter de partida : (3AH)
ADR 1	Dirección de comunicación: dirección de 8-bits consiste de 2 códigos ASCII
ADR 0	
CMD 1	
CMD 0	
DATOS (n-1)	Contenidos de datos: n x 8-bits de datos consiste de 2n códigos ASCII. n[]25 máximo de 50 códigos ASCII
.....	
DATOS 0	
LRC CHK 1	LRC check sum: check sum de 8-bits consiste de 2 códigos ASCII
LRC CHK 0	
END 1	Caracteres FINAL: END 1=CR (0DH), END 0 =LF (0AH)
END-0	

#### Modo RTU:

START	Un intervalo silencioso de más de 10 ms
ADR	Dirección de comunicación: Dirección de 8-bits
CMD	
DATOS (n-1)	Contenido de data: n x 8 bits de datos, n = 25
.....	
DATOS 0	
CRC CHK Bajo	CRC check sum: check sum de 16-bits consiste de 2 caracteres de 8-bits
CRC CHK Alto	
FINAL	Un intervalo silencioso de más de 10 ms

#### ADR (Dirección de esclavo)

Las direcciones de esclavo válidas están en el rango de 0 a 254. La dirección de comunicación igual a 0 significa transmisión a todos los variadores de frecuencia (AMD), en este caso, el AMD no contestará ningún mensaje al dispositivo maestro.

Por ejemplo, comunicación al AMD con una dirección 16 decimal:

Modo ASCII: (ADR 1, ADR 0)='1','0' => '1'=31H, '0'=30H

Modo RTU: (DIR)=10H

**CMD (Código de comando) y DATA (caracteres de datos)**

El formato de los caracteres de datos depende del código de comando. Los códigos de comandos disponibles se describen de la siguiente forma: Código de Comando: 03<sub>H</sub>, lea n palabras. El valor máximo de N es 20. Por ejemplo, leyendo 2 palabras continuas de la dirección inicial 2120<sub>H</sub> de AMD con la dirección 01<sub>H</sub>.

**Modo ASCII:**

Mensaje de Comando	
STX	':
ADR 1 ADR 0	'0'
	'1'
CMD 1 CMD 0	'0'
	'3'
Dirección de comienzo de datos	'2'
	'1'
	'0'
	'2'
Número de datos (Conteo por palabra)	'0'
	'0'
	'0'
	'2'
LRC CHK 1 LRC CHK 0	'D'
	'7'
END 1 END 0	CR
	LF

Mensaje de Respuesta	
STX	':
ADR 1 ADR 0	'0'
	'1'
CMD 1 CMD 0	'0'
	'3'
Número de datos (Conteo por byte)	'0'
	'4'
Contenido de dirección de datos inicial 2102H	'1'
	'7'
	'7'
	'0'
Contenido de dirección de datos 2103H	'0'
	'0'
	'0'
	'0'
LRC CHK 1 LRC CHK 0	'7'
	'1'
END 1 END 0	CR
	LF

**Modo RTU:**

Mensaje de Comando	
ADR	01H
CMD	03H
Dirección de comienzo de datos	21H
	02H
Número de datos (Conteo por palabra)	00H
	02H
CRC CHK Bajo CRC CHK Alto	6FH
	F7H

Mensaje de Respuesta	
ADR	01H
CMD	03H
Número de datos (Conteo por byte)	04H
	'0'
Contenido de dirección de datos 2102H	17H
	70H
Contenido de dirección de datos 2103H	00H
	02H
CRC CHK Bajo CRC CHK Alto	FEH
	5CH

Código de comando: 06<sub>H</sub>, escribe 1 palabra

Por ejemplo, escribiendo 6000 (1770<sub>H</sub>) a la dirección 0100<sub>H</sub> de AMD con dirección 01<sub>H</sub>.

### Modo ASCII:

Mensaje de Comando	
STX	':
ADR 1 ADR 0	'0'
	'1'
CMD 1 CMD 0	'0'
	'6'
Dirección de Datos	'0'
	'1'
	'0'
	'0'
	'1'
	'7'
	'0'
LRC CHK 1 LRC CHK 0	'7'
	'1'
END 1 END 0	CR
	LF

Mensaje de Respuesta	
STX	':
ADR 1 ADR 0	'0'
	'1'
CMD 1 CMD 0	'0'
	'6'
Dirección de datos	'0'
	'1'
	'0'
	'0'
Contenido de datos	'1'
	'7'
	'7'
	'0'
LRC CHK 1 LRC CHK 0	'7'
	'1'
END 1 END 0	CR
	LF

### Modo RTU:

Este es un ejemplo de cómo usar el código de función 16 para escribir a registros múltiples.

Mensaje de Comando	
ADR	01H
CMD	10H
Comienzo de dirección de datos	20H
	00H
Número de datos (Conteo por byte)	04H
Contenido de dirección de datos 2000H	00H
	02H
Contenido de dirección de datos 2001H	02H
	58H
CRC CHK Bajo CRC CHK Alto	CBH
	34H

Mensaje de Respuesta	
ADR	01H
CMD	10H
Comienzo de dirección de datos	20H
	00H
Número de datos (Conteo por palabra)	00H
	02H
CRC CHK Bajo CRC CHK Alto	4AH
	08H

CHK (check sum)

**Modo ASCII:**

El LRC (Longitudinal Redundancy Check ) se calcula sumando, módulo 256, los valores de bytes desde ADR1 hasta el último carácter de datos y luego calculando la representación hexadecimal de la negación del complemento de 2 de la suma. Por ejemplo, leyendo 1 palabra de la dirección 0401<sub>H</sub> del variador de frecuencia con dirección 01<sub>H</sub>.

Mensaje de Comando	
STX	':'
ADR 1	'0'
ADR 0	'1'
CMD 1	'0'
CMD 0	'3'
Comienzo de dirección de datos	'0'
	'4'
	'0'
	'1'
Número de datos (Conteo por palabra)	'0'
	'0'
	'0'
	'1'
LRC CHK 1	'F'
LRC CHK 0	'6'
END 1	CR
END 0	LF

01<sub>H</sub>+03<sub>H</sub>+04<sub>H</sub>+01<sub>H</sub>+00<sub>H</sub>+01<sub>H</sub>=0A<sub>H</sub>,  
la negación del complemento de 2 de 0A<sub>H</sub>  
es F6<sub>H</sub>.

**Modo RTU:**

Mensaje de Respuesta	
ADR	01H
CMD	03H
Comienzo de dirección de datos	21H
	02H
Número de datos (Conteo por palabra)	00H
	02H
CRC CHK Bajo	6FH
CRC CHK Alto	F7H

CRC (Verificación de Redundancia Cíclica) es calculado por los siguientes pasos:

Paso 1: Cargue un registro de 16-bits (llamado registro CRC) con FFFF<sub>H</sub>.

Paso 2: Haga la instrucción Or Exclusivo el primer byte de 8-bits del mensaje de comando con el byte de orden baja del registro de CRC de 16-bits, poniendo el resultado en el registro CRC.

Paso 3: Haga shift el registro CRC un bit hacia la derecha con llenado de cero MSB. Extraiga y examine el LSB.

Paso 4: Si el LSB del registro CRC es 0, repita el paso 3, haga otro OR Exclusivo el registro CRC con el valor polinomio de A001<sub>H</sub>.

Paso 5: Repita pasos 3 y 4 hasta que se hayan hecho ocho shifts. Cuando se complete esto, se habrá procesado un byte de 8-bits completo.

Paso 6: Repita los pasos del 2 al 5 para el próximo byte de 8-bits del mensaje de comando.

Continúe haciendo esto hasta que todos los bytes hayan sido procesados. El contenido final del registro CRC son los valores del CRC.



*Nota: Cuando se transmite el valor de CRC en el mensaje, los bytes superiores e inferiores del valor del CRC deben ser intercambiados, por ejemplo, el byte de orden inferior se transmitirá primero.*

El siguiente es un ejemplo para generar CRC usando lenguaje C. La función toma dos argumentos:

Unsigned char\* data ← Un puntero al buffer de mensaje

Unsigned char length ← La cantidad de bytes en el buffer de mensaje

La función retorna el valor de CRC como un tipo entero sin signo.

```
Unsigned Int crc_chk(unsigned char* data, unsigned char length){
    int j;
    unsigned int reg_crc=0xFFFF;
    while(length--){
        reg_crc ^= *data++;
        for(j=0;j<8;j++){
            if(reg_crc & 0x01){ /* LSB(b0)=1 */
                reg_crc=(reg_crc>>1) ^ 0xA001;
            }else{
                reg_crc=reg_crc >>1;
            }
        }
    }
    return reg_crc;
}
```