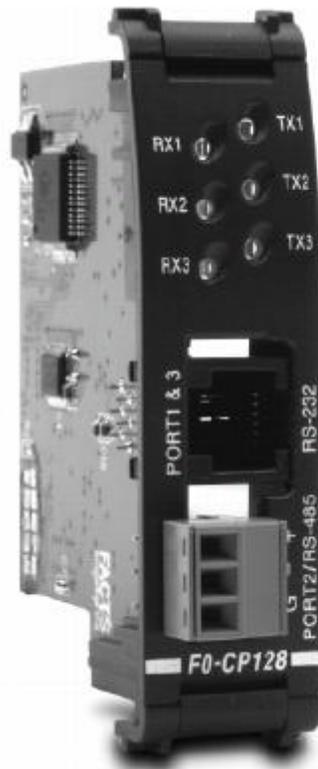


# FACTS engineering

Automationdirect.com™

Manual de Usuario  
del CoProcessor BASIC de tres puertos  
para PLCs *DirectLogic* DL05 y DL6



No. de artículo: F0-CP-M-SP



---

**MARCAS REGISTRADAS**

---

**AUTOMATIONDIRECT.COM** <sup>TM</sup> es una marca registrada de Automationdirect.com.

**COPROCESSOR** <sup>TM</sup> es una marca registrada de FACTS Engineering, Inc.

---

**PROPIEDAD INTELECTUAL (COPYRIGHT)**

---

Copyright 2004, FACTS Engineering Inc., 8049 Photonics Dr, New Port Richey, Florida, 34655, USA.

---

### ADVERTENCIA

---

Gracias por comprar el equipo de automatización de FACTS ENGINEERING. Deseamos que su nuevo equipo de automatización de FACTS ENGINEERING funcione con seguridad. Cualquier persona que instala o aplica este equipo debe leer esta publicación (y cualquiera otra publicación relevante) antes de instalar o de hacer funcionar el equipo.

Para reducir al mínimo el riesgo potencial de problemas de seguridad, usted debe seguir todos los códigos locales y nacionales aplicables que regulen la instalación y la operación de su equipo. Estos códigos varían de área a área y cambian generalmente con el tiempo. Es su responsabilidad determinar qué códigos deben ser seguidos, y verificar que el equipo, la instalación, y la operación esté de acuerdo con la última revisión de estos códigos.

Como mínimo, usted debe seguir todas las secciones aplicables del National Fire Code, National Electrical Code, y los reglamentos de National Electrical Manufacturers Association (NEMA) de Estados Unidos. Otras oficinas gubernamentales reguladoras o locales pueden ayudar a determinar qué códigos y estándares son necesarios para instalación y operación seguras.

Pueden resultar daños al equipo o lesión seria al personal al no seguir todos los reglamentos y estándares aplicables. No garantizamos que los productos descritos en esta publicación son convenientes para su aplicación particular, ni asumimos cualquier responsabilidad del diseño de su producto, instalación, u operación.

Si usted tiene cualquier pregunta referentes la instalación o a la operación de este equipo, o si usted necesita la información adicional, llámenos por favor en 1-800-783-3225.

Este documento se basa en la información disponible a el momento de su publicación. Aunque se han hecho esfuerzos de ser precisos, la información contenida no pretende cubrir todos los detalles o variaciones en hardware y software, ni prever cada contingencia posible en la conexión con la instalación, la operación, y el mantenimiento. Pueden ser descritas características que no están presentes en todos los sistemas de hardware y de software. FACTS ENGINEERING no asume ninguna obligación de aviso a los lectores de este documento con respecto a cambios realizados posteriormente.

FACTS ENGINEERING conserva el derecho de realizar cambios al hardware y al software en cualquier momento, sin aviso previo.

FACTS ENGINEERING no hace ninguna representación o garantía, expresada, implicada, o estatutaria con respecto a, y no asume ninguna responsabilidad de la exactitud, de lo completo, de la suficiencia o de la utilidad de la información contenida adjunto. No se aplicará ninguna garantía de mercadología de aptitud para el propósito.

# HISTORIA DE REVISIONES DEL MANUAL

**Por favor incluya el número y edición del Manual, mostrados abajo, cuando se comunique con Apoyo Técnico en relación a esta publicación.**

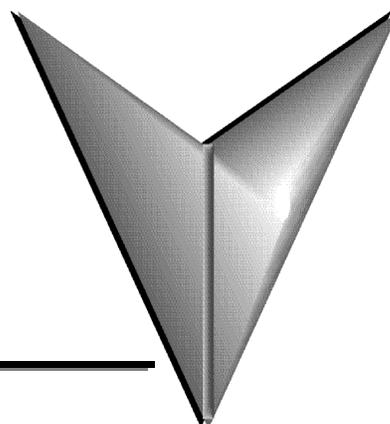
Manual No.: F0-CP-M-SP  
Edición: Primera edición en español  
Fecha de edición: 10/05

Historia de la publicación		
Edición	Fecha	Descripción de cambios
Original	9/2005	Original issue in English
Primera edición en español	10/2005	Traducción del original, por el Ing. Luis Miranda, Miembro del equipo de Apoyo Técnico de Automationdirect



# CONTENIDO

---



## CAPÍTULO 1: INTRODUCCIÓN

Introducción .....	1-2
Propósito de este documento .....	1-3
Quién debe leer este manual .....	1-3
Apoyo técnico .....	1-3
Convenciones usadas .....	1-3
Sincronización con la CPU .....	1-4
Command@ .....	1-5

## CAPÍTULO 2: DECLARACIONES DEL COPROCESSOR

BMOVE .....	2-2
Coma Flotante de IEEE .....	2-2
Numeración octal y tipos de datos para los operandos BMOVE .....	2-3
Operandos BMOVE de DL05 .....	2-3
Operandos BMOVE de DL06 .....	2-3
DPORT .....	2-5
Coma Flotante de IEEE .....	2-5
S06 _ .....	2-8
Coma Flotante de IEEE .....	2-8
Numeración octal y tipos de datos para los operandos S06 .....	2-8
Operandos S06 _ de la CPU DL05 .....	2-9
Operandos S06 _ de la CPU DL06 .....	2-9

## CAPÍTULO 3: Especificaciones

Especificaciones generales del Coprocessor .....	3-2
Descripción del F0-CP128 .....	3-3
Descripción y localización del puente CLEAR ALL del F0-CP128 .....	3-4

Clavijas de los puertos DEL F0-CP128 .....3-5

**APÉNDICE A: Como COMENZAR usando ABM Commander Plus**

Como comenzar usando ABM Commander Plus .....A-2  
Modificando un programa .....A-3  
Archivando un programa (save) .....A-4  
Modo AUTORUN .....A-5  
Borrando un programa .....A-6  
Cancelando el modo AUTORUN .....A-7  
Cambiando el puerto de programación .....A-7

**APÉNDICE B: Búsqueda de problemas**

No es posible comunicarse con el CoProcessor .....B-1

**APÉNDICE C: Diagramas de cableado RS-232 y RS-485**

Estandard RS-232 .....C-2  
Cables de la computadora IBM o compatible .....C-3  
Estandard RS-485 .....C-5  
El blindaje del cable .....C-7  
Cables de conexión y resistencias de terminación .....C-7

**ÍNDICE**

# INTRODUCCIÓN

---



## En este capítulo...

Introducción .....	1-2
Quién debe leer este manual .....	1-3
Apoyo técnico .....	1-3
Convenciones usadas .....	1-3
Sincronización de la CPU .....	1-4
COMMAND@ .....	1-5

# 1 Introducción

Este manual describe detalles específicos del CoProcessor BASIC para los PLCs 05 y 06. Este documento se debe usar para complementar el manual de referencia del usuario de BASIC ampliado (FA-BASIC-M-SP) al programar los módulos de CoProcessor de FACTS Engineering para los PLCs DL05 y DL06

Los módulos CoProcessor se instalan en la ranura de extensión de un PLC D0-05 o en cualesquiera de las cuatro ranuras de un PLC D0-06.

El módulo CoProcessor se comunica a la CPU del PLC DL05 o DL06 usando la instrucción S06 \_, las instrucciones BMOVE, y DPORT. Se usa una interface de alta velocidad de RAM del puerto dual, a través de la barra paralela de la placa trasera del DL05 o DL06, para comunicaciones de CoProcessor al PLC y viceversa. Se pueden transferir hasta 256 bytes por el CoProcessor en un barrido del PLC usando la instrucción BMOVE. No se requiere ninguna lógica ladder de PLC para usar comunicaciones desde o al CoProcessor hasta el PLC. El CoProcessor no ocupa ninguna memoria X o Y del mapa de memoria del PLC.

La lógica ladder del PLC DL05 o DL06 puede generar una interrupción en el CoProcessor con la instrucción ladder WX y la declaración ONPLC del CoProcessor. Además de los 256 bytes que se pueden transferir usando la instrucción BMOVE, se pueden transferir hasta 256 bytes adicionales usando una interrupción ONPLC accionada por WX en un sólo barrido del PLC.

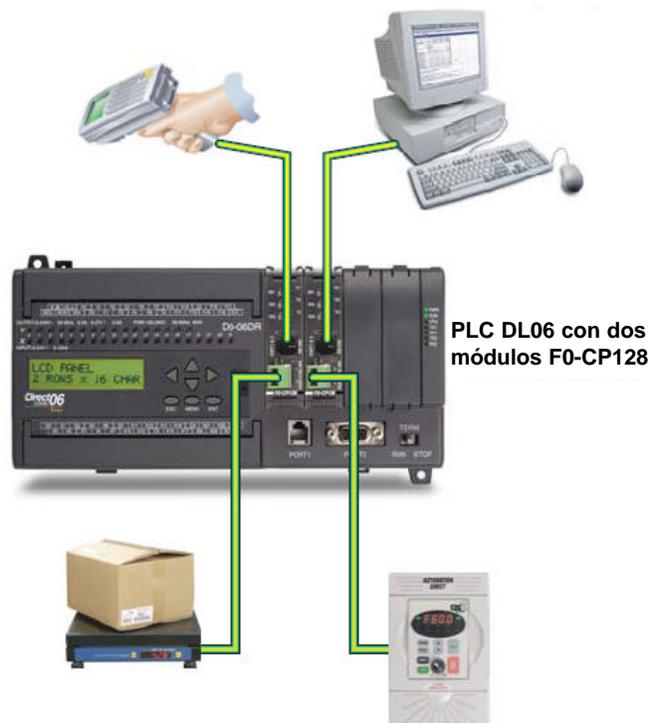
El módulo de CoProcessor se comunica a los dispositivos externos usando puertos seriales ya construidos en el módulo.

Cada módulo ya tiene integrado el program BASIC en ROM. Por lo tanto, solamente es necesario un terminal, tal como *Hyperterminal*, para programar el módulo.

Sin embargo, se entrega un programa llamado **ABM Commander Plus** para facilitar la programación.

Este módulo permite hacer una interface del PLC a lectores de código de barras, terminales de operador, balanzas que tengan comunicación ASCII, variadores de frecuencia, servomotores, compu-tadoras y otros aparatos con comunicación serial, expandiendo las funciones de los PLCs DL05 y DL06.

También permite hacer operaciones matemáticas complejas con números reales, senos, cosenos, tangentes, exponenciales, raíz cuadrada, etc. Y esto permite aliviar la CPU del PLC y por lo tanto reducir el tiempo de barrido del PLC.



## Quién debe leer este manual

Este manual contiene información importante para los que instalen, mantengan, y/o hagan funcionar el modulo Coprocessor F0-CP128.

## Apoyo Técnico

### Apoyo de FACTS Engineering

**Por Teléfono: 1-800-783-3225 en Estados Unidos o**

**Por fax 727-375-5441**

**(Lunes a Viernes, 9:00 a.m.-6:00 p.m. E.T.)**

**En Internet: [www.facts-eng.com](http://www.facts-eng.com)**

### Apoyo de AutomationDirect

**Por Teléfono: 770-844-4200**

**(Lunes a Viernes, 9:00 a.m.-6:00 p.m. E.T.)**

**En Internet: [www.automationdirect.com](http://www.automationdirect.com)**

Nuestros grupos de apoyo técnico trabajarán con usted para contestar sus preguntas. Si no puede encontrar la solución para su aplicación, o si por cualquier otra razón usted necesita ayuda técnica adicional, por favor llame a Apoyo Técnico.

Además, le invitamos a que visite nuestro sitio en Internet, donde puede encontrar información técnica y no técnica sobre nuestros productos y nuestra empresa.

## Convenciones usadas

- argumentos* Los argumentos de una instrucción o declaración son mostrados en itálico
- CTRL-C* El tecleado de un conjunto de teclas será indicado de esta forma.
- RENUMBER Una instrucción o declaración que se use en el programa es mostrada en mayúsculas
- 1085 PRINT1 \$(I) "="
- 1090 NEXT :REM ejemplo de programa
- PRM0
- READY (El programa es mostrado en negrilla arial, 9 puntos).



Cuando vea el icono de la "libreta" en el margen de la izquierda, el párrafo en el lado derecho será una nota especial..

La palabra **NOTA**: en negrilla marcará el inicio del texto.



Cuando vea el icono del "signo de exclamación" en el margen de la izquierda, el párrafo a la derecha será uno de **ADVERTENCIA**. Esta información puede evitar heridas, pérdidas de propiedad, o (en casos extremos) hasta la muerte.

La palabra **ADVERTENCIA**: en negrilla marcará el inicio del texto.

## Asuntos claves de cada capítulo

El inicio de cada capítulo hará una lista de los tópicos principales encontrados en ese capítulo.

INTRODUCCIÓN	CAPÍTULO 1
En este capítulo...	
Descripción del manual	1-2
Introducción al módulo ECOM	1-5

## 1 SINCRONIZACIÓN DE LA CPU

En la energización, el CoProcessor resetea y establece la comunicación con el PLC DL05 o DL06. Luego se ejecuta el modo de operación guardado por el último comando de AUTOSTART.

Vea por favor AUTOSTART en el manual de referencia de BASIC ampliado de FACTS Eng., para información adicional.

El CoProcessor no se resetea cuando la CPU no está en modo RUN. Si se desea, el estado actual de la CPU del PLC puede ser determinado examinando los relevadores especiales SP11-20.

Vea el capítulo 2 (declaraciones de CoProcessor) para una descripción de la declaración S06\_. Vea el manual del usuario de los PLCs DL05 o DL06 para una descripción de los relevadores especiales del PLC.

**Ejemplo**

```
10 IF S06_SP(11) THEN PRINT "Estado corriente forzado"  
20 IF S06_SP(12) THEN PRINT " Estado TERM RUN "  
30 IF S06_SP(13) THEN PRINT "Estado TEST RUN"  
40 IF S06_SP(15) THEN PRINT "Estado TEST PGM"  
50 IF S06_SP(16) THEN PRINT "Estado TERM PGM"  
60IF S06_SP(17) THEN PRINT "Estado de STOP forzado"  
70 IF S06_SP(20) THEN PRINT "Modo PGM"
```

A menudo se usa un bit de estado de relevadores de control o de etapa como un permiso en el programa BASIC. Se usan bits de relevadores de control y estados de etapas para comunicar la información de estado del programa al CoProcessor. Por ejemplo, un relevador de control se puede usar para señalar el comienzo de un informe de un turno de trabajo o simplemente para indicar que la CPU del PLC está funcionando.

**Ejemplo**

```
10 IF S06_C(0) THEN PRINT "CR 0 se energizó"  
20 IF S06_SG(10) THEN PRINT "La etapa 10 está activada"
```

La definición de este comando es diferente de lo mostrado en el manual de referencia de BASIC ampliado de FACTS Eng.

## COMMAND@

**Función** Selecciona el puerto de programación

**Sintaxis** COMMAND@ *puerto*

**Uso** El *puerto* es 1 o 3 y especifica el puerto de programación o de comando. BASIC envía todos los mensajes a y acepta solamente COMMANDos desde el puerto especificado.

El *puerto* por defecto de fábrica para programación o comandos es el puerto 1 funcionando a 9600 Bauds.

Use la declaración SETPORT para cambiar la velocidad en Baud en la energización de cualquiera de los puertos seriales.

Use el COMMAND@ para hacer DEBUG (eliminar errores) de comunicaciones con un dispositivo externo conectado con otro puerto.

El COMMAND @ se puede usar para conmutar por software ambos puertos, mientras que reduce al mínimo la necesidad de intercambio de cables o del uso de cajas de conmutación.

### Ejemplo

Normalmente se tiene el puerto 1 como puerto de programación. Asuma que se ha terminado el programa para una impresión del reportaje de diagnósticos y operaciones en un turno de trabajo, con una impresora conectada al puerto 1. Ahora se desea hacer programar un control regulador de un motor paso a paso. Para comenzar a programar el control del motor paso a paso desde el puerto 3:

> SETPORT 3, 9600 Configura la velocidad en Baud en el puerto 3

> COMMAND@3 el puerto de programación ahora es el puerto 3

Cambie el cable de programación del dispositivo de programación desde el puerto 1 al puerto 3. Se incluye un cable bifurcado para este propósito.

Para ir de nuevo a la programación en el puerto 1, entre COMMAND@1 en la línea de comando.

**1**

# DECLARACIONES DE CoPROCESSOR

---



## En este capítulo...

BMOVE .....	.2-2
Coma Flotante de IEEE .....	.2-2
Numeración octal y tipos de datos para los operandos BMOVE .....	.2-3
Operandos BMOVE de DL05 .....	.2-3
Operandos BMOVE de DL06 .....	.2-3
DPORT .....	.2-5
Coma Flotante de IEEE .....	.2-5
S06 _ .....	.2-8
Coma Flotante de IEEE .....	.2-8
Numeración octal y tipos de datos para los operandos S06 .....	.2-8
Operandos S06 _ de la CPU DL05 .....	.2-9
Operandos S06 _ de la CPU DL06 .....	.2-9

## BMOVE

2

<b>Función</b>	Tener acceso directo a un bloque de memoria de un PLC DL05 o DL06
<b>Sintaxis</b>	BMOVE, <i>dirección</i> , <i>operando inicial (número)</i> , <i>operando final (número)</i> BMOVE, <i>dirección</i> , <i>operando inicial (número)</i> , <i>K (número de bytes)</i>
<b>Vea también</b>	DPORT, ONPLC, y S06 _
<b>Uso</b>	<p>Se pueden leer o escribir hasta 256 bytes de la memoria de un PLC DL05 o DL06 en un barrido, usando BMOVE. La memoria en la CPU del PLC es referenciada usando 11 operandos diferentes especificados con un número octal de dirección.</p> <p>El copiado del bloque comienza en el CoProcessor en la posición de memoria del puerto dual DPORT(0) y en la CPU del PLC en el <i>operando inicial (número)</i>. El copiado del bloque continúa con direcciones de memoria consecutivas hasta e incluyendo el <i>operando final (número)</i>.</p> <p>Alternativamente, se puede especificar el número de bytes a transferir como expresión en paréntesis después de la letra "K ". Si el número de bytes es 0 entonces serán copiados 256 bytes.</p> <p>Use una "R" o "W" en <i>dirección</i> para especificar una acción de lectura o escritura a la memoria del PLC. "R" leerá la memoria del PLC y la copia a la memoria de DPORT. "W" leerá la memoria en el CoProcessor y la copia desde DPORT a la memoria V del PLC.</p> <p>Si el <i>operando inicial</i> u <i>operando final</i> es un tipo de datos de BIT, se usa la dirección entera de memoria que contiene el operando.</p>

### Coma flotante de IEEE

Las variables numéricas en el módulo de CoProcessor se almacenan internamente como valor de coma flotante en el rango de  $\pm 1E-127$  a  $\pm .999999999E+127$ .

La CPU del PLC puede almacenar números como valor BCD, binario, o de coma flotante IEEE en el rango de  $\pm 3,40282E\pm 38$ . Si usted está usando valores de coma flotante IEEE en el PLC y usted desea operar en esos valores, use BMOVE y DPORT del módulo de CoProcessor con la porción R o el S06\_VR.

Numeración octal y tipos de datos para operandos BMOVE

Operandos BMOVE del PLC DL05

2

Descripción	Operando	Cantidad	Número Octal	Tipo de dato	Palabra octal de memoria
Valor corriente de temporizador	T	128	0 -177	BCD	0-177
Valor corriente de contador	CT	128	0- 177	BCD	1000 - 1177
Memoria V	VH	3698	1200-7377	Hexadecimal o BCD o coma flotante	1200-7377
Volátil	VB	128	7400-7577		7400-7577
No Volátil	VR	128	7600-7777		7600-7777
Parámetros de sistema					
Entradas	X	256	0 - 377	Bit	40400-40417
Salidas	Y	256	0 - 377	Bit	40500-40517
Relevadores internos	C	512	0 - 777	Bit	40600-40637
Estado de etapas	SG	256	0 - 377	Bit	41000-41017
Estado de timers	TS	128	0 - 177	Bit	41100-41107
Estado de contadores	CS	128	0 - 177	Bit	41140-41147
Relevadores especiales(solo lectura)	SP	512	0 - 777	Bit	41200-41237

Operandos BMOVE del PLC DL06

Descripción	Operando	Cantidad	Número Octal	Tipo de dato	Palabra octal de memoria
Valor corriente de temporizador	T	256	0 -377	BCD	0 - 377
Valor corriente de contador	CT	128	0- 177	BCD	1000 - 1177
Memoria V	VH	192	400-677	Hexadecimal o BCD o coma flotante	400-677
Volátil		3200	1200-7377		1200-7377
		4096	10000 - 17777		10000 - 17777
No Volátil		128	7400-7577		7400-7577
Parámetros de sistema		64	700 - 777		700 - 777
		128	7600-7777	7600-7777	
		1024	36000 - 37777	36000 - 37777	
Entradas	X	512	0 - 777	Bit	40400-40437
Salidas	Y	512	0 - 777	Bit	40500-40517
Relevadores internos	C	1024	0 - 1777	Bit	40600-40677
Estado de etapas	SG	1024	0 - 1777	Bit	41000-41077
Estado de timers	TS	256	0 - 377	Bit	41100-41117
Estado de contadores	CS	128	0 - 177	Bit	41140-41147
Remote I/O	GX	2048	0 - 3777	Bit	40000-40177
	CY	2048	0 - 3777	Bit	40200-40377
Relevadores especiales(sólo lectura)	SP	512	0 - 777	Bit	41200-41237

- Ejemplo** Cargar una tabla de 6 constantes en la memoria deL PLC que comienza en V2000
- ```
10 REM Carga la tabla de datos en la memoria de puerto dual
20 DPORT(0)=10H
30 DPORT(2)=20H
40 DPORT(4)=25H
50 DPORT(6)=30H
60 DPORT(8)=100H
70 DPORT(10)=9798H
80 REM copia la tabla de datos a la memoria del PLC
90 BMOVE W, VH(2000), K(12)
```
- Ejemplo** Multiplicar un rango de memoria de usuario por un valor constante
- ```
10 BMOVE R, VH(2000), K(32): REM Obtiene los valores desde
20 REM .... el PLC y se multiplican por 2,5
30 FOR DIR = 0 TO 31 STEP 2
40 DPORT(DIR)=DPORT(DIR)*2.5
50 NEXT DIR
60 BMOVE W, VH(2000), K(32): REM Coloca valores de vuelta al PLC
```
- Ejemplo** Obtener la tabla imagen de X (entradas) en el PLC DL06
- ```
10 BMOVE R, X(0), X(20)
```
- Avanzado** Si no se especifica ningún operando entonces el número es la representación hexadecimal de la dirección octal de memoria (7FH = memoria octal 177). BMOVE R, VH(2000), K(10) es igual que BMOVE R, (400H), K(10). De esta forma se simplifican los lazos FOR-NEXT y otros tipos de accesos de memoria "calculados" del PLC.
- Ejemplo** Encontrar todas las direcciones de memoria de usuario en el PLC que tengan una constante predeterminada
- ```
10 K = 1234: REM Valor a ser encontrado
15 REM Busque en las memorias V2000-V7777
20 FOR ÍNDICE=400H TO 1000H STEP 127: REM 2 Bytes/memoria V
30 BMOVE R, (INDEX), K(127)
40 FOR ADDR = 0 TO 125 STEP 2
50 IF DPORT(ADDR)<>K THEN 70
60 PRINT1 "Encontrado en la memoria hexadecimal = ",
62 PRINT1 HEX$(INDEX+ADDR)
70 NEXT ADDR
80 NEXT INDEX
```

## DPORT

<b>Función</b>	Lee o escribe a memorias compartidas con el PLC DL05 o el DL06
<b>Sintaxis</b>	DPORT ( <i>dirección, porción</i> ) = <i>expresión</i> Variable = DPORT ( <i>dirección, porción</i> )
<b>Uso</b>	<p>DPORT (memoria de puerto dual) se utiliza conjuntamente con la declaraciones de interrupción ONPLC y BMOVE (copia de un bloque) para tener acceso a la CPU.</p> <p>El operador DPORT recupera el valor en la dirección de memoria del puerto dual y lo asigna a la <i>variable</i>.</p> <p>La declaración DPORT almacena el valor de la <i>expresión</i> en la dirección de memoria del puerto dual.</p> <p><i>dirección</i> es una expresión de 0 a 516, que selecciona dos bytes de memoria de puerto dual. DPORT recupera o asigna un valor del número entero (0 a 65535) en la <i>dirección</i>.</p> <p><i>porción</i> es opcional y se utiliza para especificar una posición de bit, un nibble (grupo de 4 bits), un byte (grupo de 8 bits), una palabra BCD (2 bytes), o un valor de coma flotante IEEE (4 bytes).</p> <p>Use el sufijo :</p> <p>"<b>B(n)</b>" para especificar una de 16 posiciones de bits, donde n = 0-15.</p> <p>"<b>N(n)</b>" para especificar uno de cuatro nibbles, donde n = 0-3.</p> <p>"<b>H</b>" para especificar el byte mas significativo o use</p> <p>"<b>L</b>" para especificar el byte menos significativo.</p> <p>"<b>B</b>" para especificar una conversión de palabra hexadecimal a BCD.</p> <p>"<b>R</b>" para especificar una conversión de coma flotante BASIC a coma flotante IEEE.</p> <p>Los primeros 256 bytes de memoria de puerto dual, DPORT(0) a DPORT(255), son usados por la declaración BMOVE al leer o escribir datos al PLC.</p> <p>Los 256 bytes siguientes de memoria de puerto dual, DPORT(256) a DPORT(511), se usan conjuntamente con la declaración ONPLC. Este bloque de memoria es obtenido por la CPU usando la instrucción WX. Los últimos 5 bytes de memoria de puerto dual, DPORT(512) a DPORT(516), son bytes de control para WX (vea ONPLC para una descripción completa).</p>

### Coma flotante de IEEE

Las variables numéricas en el módulo de CoProcessor se almacenan internamente como valor de coma flotante en el rango de  $\pm 1E-127$  a  $\pm .999999999E+127$ . La CPU del PLC puede almacenar números como valor BCD, binario, o de coma flotante IEEE en el rango de  $\pm 3,40282E\pm 38$ . Si usted está usando valores de coma flotante IEEE en el PLC y usted desea operar en esos valores, use BMOVE y DPORT del módulo de CoProcessor con la porción R o el S06\_VR.

**Ejemplo** Recuperar un valor BCD de 4 dígitos (0-9999) de la memoria del puerto dual

```
10 REM Se coloca un número BCD en la memoria V2000 con S06_VB
20 S06_VB(2000)=1234
30 REM Se recupera con una instrucción BMOVE
40 BMOVE R, VH(2000), K(2)
50 PRINT1 "Valor BCD en la memoria V2000 =",
52 PRINT1 HEX$(DPORT(0))
```



---

*NOTA: Use DirectSoft DataView y representación BCD/HEX para ver datos BCD en el PLC.*

---

**Ejemplo** Almacenar valores BCD de 8 dígitos (0-99999999) en las memorias V2000 y V2001 usando BMOVE

```
10 DPORT(0) = 1234H: REM Constante colocada en la memoria V2000
20 A = 5678: REM A debe ser un valor BCD en el rango 0 - 9999
30 DPORT(2) = VAL("0"+STR$(A)+"H"): REM igual a DPORT(2, B)=A
40 BMOVE W, VH(2000), VH(2001)
```



---

*NOTA: Use DirectSoft DataView y representación BCD/HEX para ver datos BCD en el PLC.*

---

**Ejemplo** Recuperar un valor entero hexadecimal (0-FFFF<sub>H</sub>/0-65535<sub>d</sub>) de una memoria de puerto dual

```
10 REM Coloca un número Hexadecimal en la memoria V2000
20 S06_VH(2000)=1234
30 REM Se recupera con una instrucción BMOVE
40 BMOVE R, VH(2000), K(2)
50 PRINT1 "Valor del entero en la memoria V2000 =",
52 PRINT1 HEX$(DPORT(0))
```



---

*NOTA: Use DirectSoft DataView y representación BCD/HEX para ver datos BCD en el PLC.*

---

**Ejemplo** Almacenar un valor de coma flotante del PLC y luego recuperar un valor

```
10 REM Escribe un valor real a V1400/1401 y lee un valor real desde V1410/1411
20 DPORT(0, R) = +3.40282E+38
30 BMOVE W, VH(1400), K(4): REM Números reales usan 2 palabras o 4 bytes
40 BMOVE R, VH(1410), K(4): REM Números reales usan 2 palabras o 4 bytes
50 X=DPORT(0, R)
```



---

*NOTA: Use DirectSoft DataView y representación BCD/HEX para ver datos BCD en el PLC.*

---

**Ejemplo**

```

Usar DPORT con el tipo de modificador de la declaración PICK
1000 V=1120H
1010 DPORT(0)=V: PRINT1 "Recupera valores desde DPORT"
1020 PH1. "DPORT(0) =",V, "en hexadecimal"
1030 PRINT1 "Primer nibble =", DPORT(0, N(0)), SPC (5),
1040 PRINT1 "3r nibble =", DPORT(0, N(2))
1050 PRINT1 "DPORT(0) en binario ="; : FOR BT=15 TO 0 STEP -1
1060 IF DPORT(0, B(BT)) THEN PRINT1 "1";ELSE PRINT1 "0";
1070 NEXT BT : PRINT1
1080 PH1. DPORT(0),
1090 PRINT1 "o",V, "tratado como BCD =", DPORT(0, B), "decimal"
1100 HB=DPORT(0, H): REM intercambia los bytes
1110 DPORT(0, H)=DPORT(0, I): DPORT(0, I)=HB
1120 PH1. "Valor con bytes intercambiados =", DPORT(0)
1130 PRINT1: PRINT1 "Asignando bits y nibbles en DPORT"
1140 DPORT(0)=0
1150 FOR BT=0 TO 15
1160 DPORT(0, B(BT))=1
1170 IF BT=8 THEN PRINT1
1180 PH1. DPORT(0), SPC (3),
1190 NEXT: PRINT1
1200 DPORT(0)=0
1210 FOR N=0 TO 3
1220 DPORT(0, N(N))=0FH
1230 PH1. DPORT(0), SPC (3),
1240 NEXT: PRINT1
1250 PRINT1 "ASIGNACIÓN BCD"
1260 DPORT(0, B)=1120
1270 PH1. DPORT(0), "= 1120"
READY
> RUN
    
```

Recupera valores desde DPORT

DPORT(0) = 1120H en hexadecimal

Primer nibble= 0 3r nibble = 1

DPORT(0) en binario = 0001000100100000

1120H o 4384 tratado como BCD = 1120 decimal

Valor con bytes intercambiados = 2011H

Asignando bits y nibbles en DPORT

0001H 0003H 0007H 000FH 001FH 003FH 007FH 00FFH 01FFH 03FFH 07FFH

0FFFH 1FFFH 3FFFH 7FFFH FFFFH 000FH 00FFH 0FFFH FFFFH

ASIGNACIÓN BCD

1120H = 1120

## ONPLC

2

**Función** Interrupción basada en lógica ladder de un flujo de programa BASIC

**Sintaxis** ONPLC *número de línea*

**Vea también** BMOVE, DPORT, y S06 \_

**Uso** ONPLC permite hacer una interrupción a un flujo normal de un programa BASIC en respuesta a peidos hechos por la CPU del PLC.

ONPLC especifica el *número de línea* inicial donde continuará la ejecución de un programa cuando ocurre la interrupción. La interrupción es atrasada hasta que sea completada la ejecución de la declaración BASIC corriente (La ejecución comienza inmediatamente si la declaración corriente es IDLE o DELAY).

Después que se ejecute la declaración RETI, la ejecución se reanuda con la declaración siguiente a la última declaración ejecutada cuando sucedió la interrupción.

La declaración ONPLC solamente activará una sola interrupción BASIC. La interrupciones futuras ONPLC se desactivan hasta que se ejecute otra declaración ONPLC. Normalmente se incluye otra declaración ONPLC en la subrutina de interrupción.

Una declaración ONPLC con un *número de línea* de 0 deshabilita la interrupción ONPLC.

La CPU pasa datos al CoProcessor y causa que ocurra una interrupción usando una instrucción WX del PLC. Pueden ser transferidos hasta 256 bytes de datos con una instrucción WX. Los datos son transferidos a las direcciones del puerto dual del coProcessor DPORT(256) o DPORT(511). La cantidad de bytes transferidos es almacenado en DPORT(516,L)

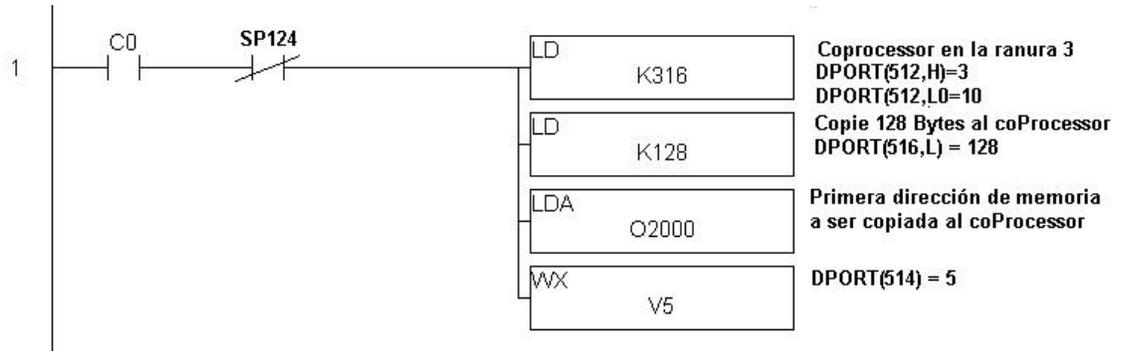
Al ejecutar una instrucción WX en el PLC se activa un relevador especial de comunicación BUSY asociado a la ranura donde está instalado el coProcessor. Una declaración ONPLC desactiva el relevador especial cuando ocurre un GOTO *número de línea*.

Relevadores especiales (SP) de comunicación de datos				
Ranura	1	2	3	4
BUSY	120	122	124	126
ERROR	121	123	125	127

Después de una interrupción ONPLC, DPORT (513,L) vuelve a 1 si la interrupción fue causada por una instrucción WX. Vuelve a 0 si la interrupción fuera causada por una instrucción RX (La instrucción RX no es permitida actualmente por el coProcessor).

La primera ranura en un PLC DL05 es la ranura 1. La instrucción WX solamente se ejecuta cuando el PLC está en modo RUN.

**Ejemplo** Escribir una memoria V al coProcessor usando WX.



2

El byte más significativo del valor el LD en el ejemplo de arriba contiene el número de la ranura donde está el CoProcessor (1-4). El byte menos significativo contiene un código BCD desde 1 a 90, que es escrito al coProcessor en DPORT(512,L). Este valor puede ser usado como se necesite en el programa de la aplicación y no afecta la ejecución de la instrucción WX. El valor escrito estará en el segundo registro del stack cuando se ejecute la instrucción WX.

El primer registro del stack contiene la dirección fuente de la memoria V de datos en el PLC. Esto es definido en la segunda instrucción LD en el programa de arriba. El conteo de bytes se almacena en el coProcessor en DPORT(516,L).

El acumulador contiene la dirección fuente de la memoria V de datos en el PLC. Esto es especificado con la instrucción LDA en el ejemplo. Pueden ser transferidos hasta 256 bytes o 128 registros consecutivos de memoria V al coProcessor con una instrucción WX. Los datos se almacenan en el coProcessor comenzando en DPORT(256).

La dirección usada con la instrucción WX puede ser cualquiera. Esta dirección es convertida desde octal a hexadecimal y es almacenada en el coProcessor en DPORT(514).

En el ejemplo, una bobina C0 se activa para iniciar una interrupción ONPLC. SP124 se usa para prevenir que otra instrucción WX se ejecute cuando el CoProcessor esté ocupado.

LD K310 identifica el coProcessor en la ranura 3 y almacena el valor 10 en DPORT(512,L). LD K128 especifica que serán escritos 128 bytes. LDA O2000 especifica la dirección de memoria del PLC. Los datos serán copiados desde la memoria V2000 hasta V2077 al rango DPORT(256) hasta DPORT(383). La instrucción WX V5 activa el revelador BUSY SP124, escribe los datos y almacena 5 en DPORT(514).

10 ONPLC 100

20 IDLE : GOTO 10: REM No hace nada esperando por la interrupción.

100 REM Inicio de la subrutina de interrupción - imprime los datos

130 PRINT "Códigos de identificación de datos"

140 PRINT DPORT(512,L), SPC (2), DPORT(514)

150 FOR K=0 TO DPORT(516,L) -1 STEP 2

160 PRINT "Palabra dual port", K/2+1,"=", DPORT(256+K): NEXT K

175 ONPLC 100

180 RETI

### S06\_

2

<b>Función</b>	Tener acceso directo a la memoria del PLC
<b>Sintaxis</b>	$S06 = \text{operando (número)} = \text{expresión}$ $\text{variable} = S06\_ \text{operando (número)}$
<b>Abreviatura</b>	<b>S.</b> <i>operando (número)</i>
<b>Ver también</b>	BMOVE, DPORT y ONPLC
<b>Uso</b>	<p>Se puede tener acceso directo a la memoria del PLC en cada barrido usando 12 operandos diferentes especificados con un <i>número</i> octal de dirección.</p> <p>La declaración <b>S06_</b> copia el valor de la <i>expresión</i> en la dirección de memoria del PLC especificada por el <i>operando(número)</i>. Si la dirección de memoria es luego escrita por el programa ladder del PLC, la declaración S06_ será anulada.</p> <p>El operador <b>S06_</b> copia el valor de dirección de memoria del PLC especificada por el <i>operando(número)</i> en una variable numérica.</p> <p>Los valores <b>S06_</b> serán tipos de datos BCD (<b>VB</b>), HEXadecimal (<b>VH</b>), BIT (<b>X</b>, <b>Y</b>, <b>C</b>, etc.) o coma flotante IEEE (<b>VR</b>) dependiendo del operando usado. Los operandos discretos tales como puntos de entradas o salidas y relevadores de control operan en bits y devuelven valores lógicos. Los valores corrientes de temporizadores (<b>T</b>) y contadores (<b>CT</b>) están en BCD.</p> <p>La tabla de la próxima página especifica los tipos de datos y el sistema octal para cada uno de los operandos <b>S06_</b> (se muestra en los ejemplos el uso de operandos típicos VB y VH).</p>

### Coma flotante de IEEE

Las variables numéricas en el módulo de CoProcessor se almacenan internamente como valor de coma flotante en el rango de  $\pm 1E-127$  a  $\pm .99999999E+127$ .

La CPU del PLC puede almacenar números como valor BCD, binario, o de coma flotante IEEE en el rango de  $\pm 3,40282E\pm 38$ . Si usted está usando valores de coma flotante IEEE en el PLC y usted desea operar en esos valores, use BMOVE y DPORT del módulo de CoProcessor con la porción R o el S06\_VR.

Numeración octal y tipos de datos para operandos de S06\_

Operandos de S0-6\_ para el PLC DL05

2

Descripción	Operando	Cantidad	Numero Octal	Tipo de dato	Palabra octal de memoria
Valor corriente de temporizador	T	128	0 - 177	BCD	0-177
Valor corriente de contador	CT	128	0- 177	BCD	1000 - 1177
Memoria V	VH	3698	1200-7377	Hexadecimal o BCD o coma flotante	1200-7377
Volátil	VB	128	7400-7577		7400-7577
N Volátil	VR	128	7600-7777		7600-7777
Parámetros de sistema					
Entradas	X	256	0 - 377	Bit	40400-40417
Salidas	Y	256	0 - 377	Bit	40500-40517
Relevadores internos	C	512	0 - 777	Bit	40600-40637
Estado de etapas	SG	256	0 - 377	Bit	41000-41017
Estado de timers	TS	128	0 - 177	Bit	41100-41107
Estado de contadores	CS	128	0 - 177	Bit	41140-41147
Relevadores especiales(sólo lectura)	SP	512	0 - 777	Bit	41200-41237

Operandos de S0-6\_ para el PLC DL06

Descripción	Operando	Cantidad	Numero Octal	Tipo de dato	Palabra octal de memoria
Valor corriente de temporizador	T	256	0 - 377	BCD	0 - 377
Valor corriente de contador	CT	128	0- 177	BCD	1000 - 1177
Memoria V	VH	192	400-677	Hexadecimal o BCD o coma flotante	400-677
Volátil	VB	3200	1200-7377		1200-7377
	VR	4096	10000 - 17777		10000 - 17777
N Volátil		128	7400-7577		7400-7577
Parámetros de sistema		64	700 - 777		700 - 777
		128	7600-7777		7600-7777
		1024	36000 - 37777		36000 - 37777
Entradas	X	512	0 - 777	Bit	40400-40437
Salidas	Y	512	0 - 777	Bit	40500-40517
Relevadores internos	C	1024	0 - 1777	Bit	40600-40677
Estado de etapas	SG	1024	0 - 1777	Bit	41000-41077
Estado de timers	TS	256	0 - 377	Bit	41100-41117
Estado de contadores	CS	128	0 - 177	Bit	41140-41147
Remote I/O	GX	2048	0 - 3777	Bit	40000-40177
	CY	2048	0 - 3777	Bit	40200-40377
Relevadores especiales(sólo lectura)	SP	512	0 - 777	Bit	41200-41237

**Ejemplo**

```
Usando operandos del tipo de datos de bit :
10 REM Muestra el estado en la entrada X4
20 IF S06_X(4) THEN PRINT1 "ON" ELSE PRINT1 "OFF"

10 REM Hace ON el relevador interno C400
20 S06_C(400) = 1
```

```
10 REM Hace que Y23=OFF si CT2 está ON y X17 está OFF
20 IF S06_CS(2).AND.NOT(S06_X(17)) THEN S06_Y(23) = 0
```

**Ejemplo**

```
Usando operandos del tipo de datos BCD :
10 REM Muestra el estado corriente de CNT CT10 y TMR F T0
20 PRINT1 "Contador 10 =", S06_CT(10)
30 PRINT1 "Temporizador T0 =", S06_T(0)/100
```

```
10 REM divide el conteo corriente de CNT C7 por 2
20 S06_CT(7) = S06_CT(7)/2
```

```
10 REM El valor de la entrada análoga está en la memoria V2000
20 REM La memoria V2010 tiene un valor para una salida análoga
30 REM La salida análoga es proporcional a la entrada análoga
35 ESCALA=0.5: OFFSET=100
40 AOUT = S06_VB(2000) * ESCALA - OFFSET
50 REM Límite del rango del valor análogo de salida (0-4095)
60 IF AOUT < 0 THEN AOUT = 0
70 IF AOUT > 4095 THEN AOUT = 4095
80 S06_VB(2010) = AOUT
```

**Ejemplo**

```
Usando tipo de datos hexadecimal:
10 REM Muestra el tiempo corriente de barrido
20 PRINT1 "tiempo actual de barrido =", S06_VH(7775)
```

**Avanzado**

Se muestra la enumeración de memorias V para cada operando en la tabla de la página anterior. Se pueden usar los operandos VH y VB para tener acceso a cualquier porción de la memoria V.

Para mostrar el conteo corriente de CNT C0  
> P. S.VB(1000)

Para mostrar el estado de las primeras 16 entradas, X0 - X17  
> P. S.VH(40400)

**S06\_** sin operando permite direccionamiento hexadecimal de memoria. La dirección hexadecimal de memoria es igual a la dirección octal. **S06\_VH(2000)** es igual que **S06\_(400H)**. Esta característica es útil para lazos FOR-NEXT y otros tipos de accesos de memoria "calculados" del PLC.

# ESPECIFICACIONES

---



## En este capítulo...

Especificaciones generales del Coprocessor .....	3.2
Descripción del F0-CP128 .....	3.3
Descripción y localización del puente CLEAR ALL del F0-CP128 ...	3.4
Clavijas de los puertos DEL F0-CP128 .....	3.5
Clavijas del bifurcador de puertos .....	3.6

## Especificaciones generales

3

Lugar de montaje	- Cualquier ranura de módulo opcional
Consumo de energía	150 mA @ máximo a 5 VCC (suministrado por la base 05 o 06)
Ambiente de operación	- 0 a 60° C (32 a 140 grados Fahrenheit) - Humedad relativa de 5 a 95% (sin condensación)
Procesador	Cygnal C8051F123
Velocidad de clock	100 MHz, hasta 100 MIPS
Memoria de usuario	Total 128K (64K de datos, 64K de programa ) - no volátil
Conectores físicos	- 1 enchufe de 6 conductores RJ12 (puerto 1 y puerto 3 RS-232) - 1 Bloque de terminales removibles con tres conexiones (Puerto 2 RS-485)
Indicador LED	- TXD1, RXD1, TXD2, RXD2, RTS1/TXD3, CTS1/RXD3
Puerto 1	- RS-232 - Máximo 512000 Bauds
Puerto 2	- RS-485 - Máximo 512000 Bauds
Puerto 3	- RS-232 (localizado en el mismo enchufe RJ12 que el Puerto1. El Puerto 3 usa las clavijas RTS1/CTS1 en ese enchufe. Si Ud usa esas señales para otro propósito (por ejemplo handshaking de hardware en el puerto 1) el puerto 3 no puede ser usado. - Máximo 115200 Bauds
Características adicionales	- reloj de calendario respaldado por batería - El módulo puede ser programado por el puerto 1 o el puerto 3

Los siguientes **accesorios** vienen juntos cuando se compra el módulo F0-CP128 :

- Un CD con el programa **ABM Commander Plus**. Este programa funciona con Windows 98/ME/2000/XP. Permite hacer EDITS (o modificaciones) de programas BASIC para mejorar aplicaciones industriales. Ejemplos de programas para establecer comunicaciones como MODBUS maestro y esclavo.
- Un cable de 7 pies con 2 conectores RJ12
- Un adaptador D-sub de 9 clavijas hembra para RJ12 (para adaptar a un puerto serial de una computadora PC)
- Un cable bifurcador para el puerto 1 y 3.



## Descripción del módulo F0-CP128

Este módulo CoProcessor es compatible con la familia DL05/06 y ofrece 128K de memoria permanente, tres puertos seriales, reloj de calendario en tiempo real respaldado por batería, matemáticas de coma flotante, y el intérprete BASIC ampliado de FACTS Eng.

La arquitectura del microprocesador ejecuta el 70% de las instrucciones de los procesadores en 1 o 2 ciclos de clock del sistema. Un lazo enclavado genera un reloj interno del sistema de 100 MHZ para hasta 100 MIPS de ejecución. La velocidad de ejecución del módulo BASIC es aproximadamente 10 veces más rápida que las generaciones anteriores. Los programas interpretados BASIC, que son fáciles de mantener y desarrollar, pueden ahora funcionar a velocidades comparables a programas anteriores compilados en BASIC, al lenguaje Assembler o en C.

Los 128K bytes de memoria permanente permiten almacenaje y ejecución de múltiples programas, expansión de la memoria de los PLCs DL05 y DL06, y almacenaje y recuperación retentiva de datos. La memoria es mantenida por una batería que permite mantener el programa por 10 años sin energía al PLC.

El puerto 1 es una interface serial RS-232 de alto rendimiento con hasta 512000 Baud máximos, completamente configurable.

El puerto 2 es una interface serial RS-485 de alto rendimiento con hasta 512000 Baud máximos, completamente configurable.

El puerto 3 es una interface serial RS-232 de alto rendimiento con hasta 115000 Baud máximos, completamente configurable.

Los tres puertos tienen un buffer de entrada o memoria intermediaria de 255 caracteres para comunicación simultánea con tres o más dispositivos externos.

El reloj de calendario en tiempo real, respaldado por batería, mantiene la hora y fecha en que ocurren las interrupciones de energía. Las interrupciones BASIC basadas en tiempo pueden ser programadas hasta 5 milisegundos.

La operaciones de coma flotante solucionan fórmulas complejas con 8 dígitos significativos.

El intérprete ampliado BASIC de FACTS Eng. tiene muchas características y declaraciones que simplifican la programación orientada al control.

Programa desde el puerto 1 o desde el puerto 3 (COMMAND@)

Instrucción flexible de manipulación de bits (BITS y PICK)

Interrupciones por los puertos seriales y por tiempo (ONPORT y ONTIME)

Control extenso del puerto serial (SETPORT, SETINPUT, PRINT, INPUT, INPLEN, INLEN)

Instrucciones extensas de manipulación de strings (MID\$, LEFT\$, RIGHT\$, REVERSE\$, ASC, CHR\$, LCASE\$, UCASE\$, STR\$, VAL, HEX\$, OCTHEX\$, DATE\$, TIME\$)

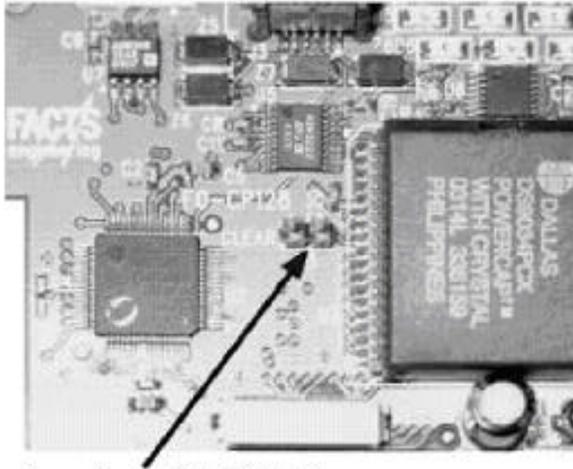
Herramientas de DEBUG (Eliminación de errores) (TRACE, STOP, CONT)

Encadenamiento de programa (GOPRM)

Declaraciones y estructuras de control comunes a la mayoría de los programas BASIC.

## Descripción y localización del puente CLEAR ALL de F0-CP128

3



Puente de CLEAR ALL

### CLEAR ALL

EL puente CLEAR ALL mostrado en la figura, habilita o deshabilita un reset del modo AUTOSTART.

La colocación del puente en un poste permite el modo AUTOSTART. Después de la energización inicial, el módulo usará los últimos parámetros almacenados del AUTOSTART. Éste es el ajuste de fábrica por defecto y la posición normal del modo RUN.

La colocación del puente en ambos postes inhabilita AUTOSTART y borra todo.

Esto se hace normalmente solamente si ha fallado el resto de las medidas para comunicarse con el CoProcessor .

Ésta es también la única manera de quitar una declaración de seguridad LOCKOUT. Cuando el CoProcessor se energiza con el puente CLEAR ALL instalado en ambos postes, el COMMAND está en el puerto 1 y la tasa de transferencia del puerto 1 es 9600 Baud.

---

**ADVERTENCIA:** Instalando el puente CLEAR ALL borrará el programa 0, todas las variables almacenadas, cancela un COMMAND@3, remueve la protección LOCKOUT y borra la información almacenada de AUTOSTART.

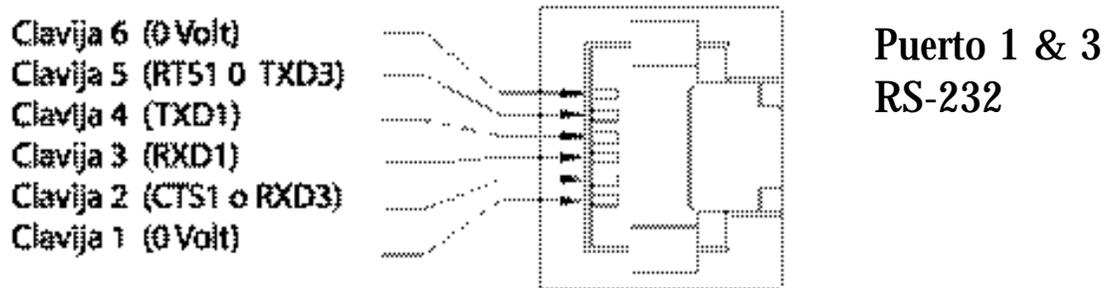
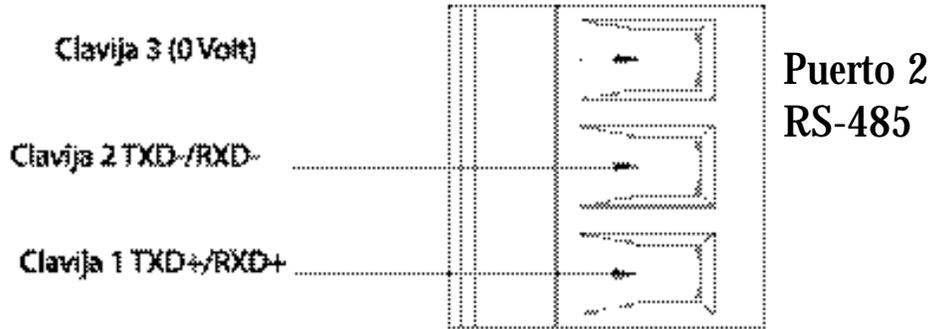
---



## Clavijas de los puertos del módulo FO-CP128

Se suministra un cable modular de enchufe RS-232 y un adaptador modular de 9 clavijas con el módulo para una conexión fácil a un puerto serial de una computadora PC. Si usted tiene una PC tal como una computadora portátil que no tenga un puerto RS-232, use por favor un adaptador USB- RS232 (USB-RS232) de **AUTOMATIONDIRECT**.

**3**



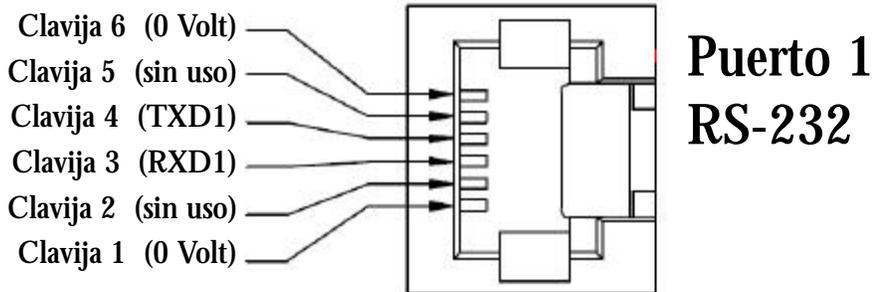
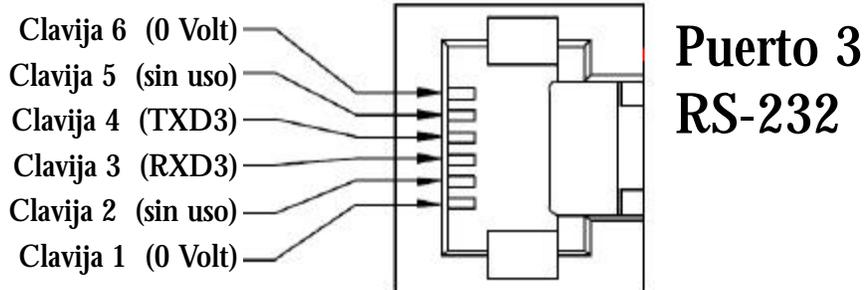
El conector RS-232 del módulo con el cable incluido es compatible con todos los conectores modulares de PLCs. También se enchufa directamente en FA-15HD (adaptador HD DSUB de 15 clavijas para PLCs), FA-CABKIT (adaptadores de fines generales RS-232 incluyendo los módems y conectores DB-25) y el convertidor aislador FA-ISOCAN (RS-232 a RS-485) de **AUTOMATIONDIRECT**.

Para programación se puede usar también el cable D2-DSCBL de **AUTOMATION DIRECT**.

## Clavijas del cable bifurcador

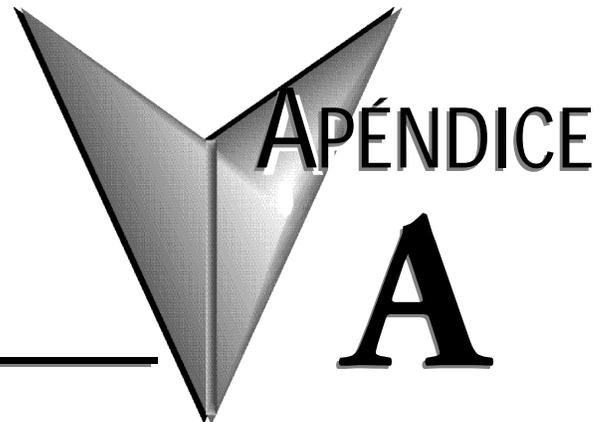
Si RTS1 y CTS1 no están siendo usados, conecte el cable bifurcador del puerto 1 y del puerto 3, mostrado abajo, con el puerto de RS-232 del módulo. Esto suministra una conexión mas fácil de cables RS-232 a ambos puertos 1 y 3.

3



# COMO COMENZAR

---

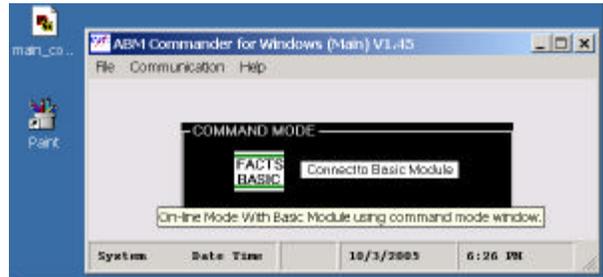
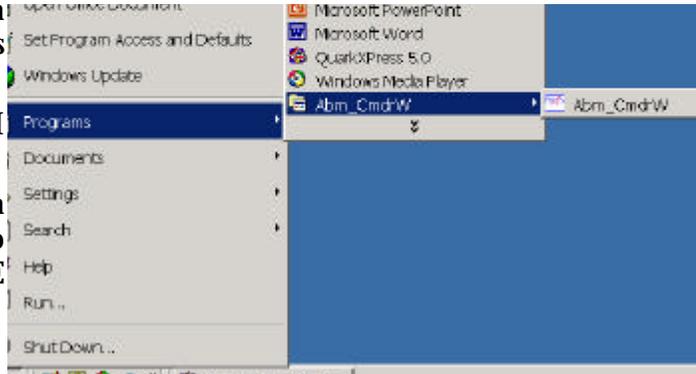


## En este apéndice:

Operación inicial usando ABM Commander Plus . . . . .	A-2
Creando o modificando un programa . . . . .	A-3
Archivando un programa (save) . . . . .	A-4
Modo AUTORUN . . . . .	A-5
Borrando un programa . . . . .	A-6
Cancelando el modo AUTORUN . . . . .	A-6
Cambiando el puerto de programación . . . . .	A-7

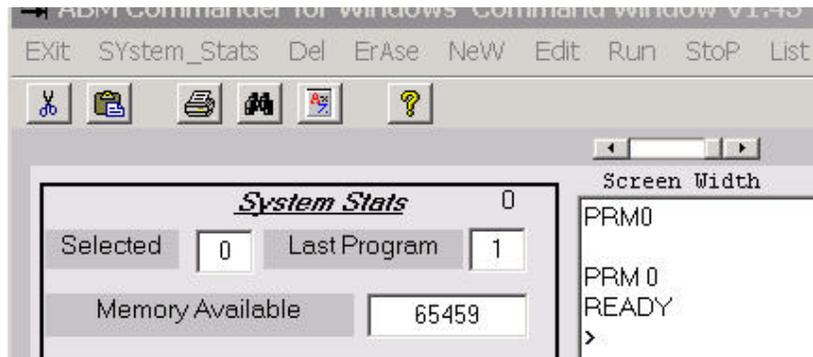
# A Operación inicial del módulo usando ABM COMMANDER PLUS

1. Haga funcionar el programa ABM Commander para Windows desde **Programs** en la PC .
2. Repase las instrucciones de ABM Commander para Windows.
3. Conecte el cable de la computadora con el módulo CoProcessor. Vea el APÉNDICE C para diagramas eléctricos.
4. Energice al PLC.
5. Seleccione el menú "**Communication**" y luego seleccione el puerto en "**Parameters(Port)**".
6. Seleccione el puerto serial de la PC que usted está usando. Haga clic en el botón "**Defaults**". Los ajustes de la comunicación ahora son 9600, 8, ninguno, 1, ninguno. Haga clic en el botón "**Apply**".
5. Haga clic en el "**COMMAND MODE> Connect to Basic Module**" ( con el módulo BASIC" desde la ventana principal. Haga clic en "**System Stats**" del menú del MODO de COMANDO.
6. El módulo ahora responderá con el aviso siguiente.



PRM0  
READY

> ("**>**" este carácter indica que BASIC está en modo de COMANDO)



Si usted no recibe este mensaje, siga por favor el procedimiento de búsqueda de problemas en el APÉNDICE B.

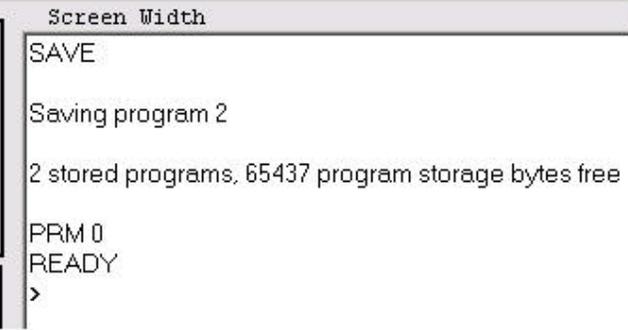
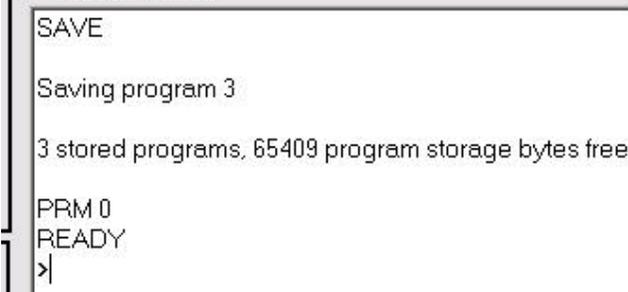
7. El CoProcessor BASIC está listo ahora para programación en línea, para supervisar valores o para la transferencia directa del programa desde el PC hasta el módulo y desde el módulo hasta el PC.

## Creando o modificando un programa (EDIT)

**A**

Acción del usuario	Lo que muestra la pantalla
<p>Haga clic en <b>'Auto'</b> en la barra de menú. Seleccione Modo 0, programa 0, y haga clic en <b>OK</b></p>	<pre> AUTOSTART 0,0  Mode = 0, Edit Program = 0 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600)  &gt;                      </pre>
<p>Entre lo siguiente en el campo "Command line"                      10 p. &lt; ENTER &gt;                      65535 p. &lt; ENTER&gt;</p>	<pre> &gt;10p. &gt;65535p. &gt;                     </pre>
<p>Haga clic en <b>'Reset'</b> en la barra de menú de ABM Commander. Apagando y encendiendo el PLC también hará reset del CoProcessor BASIC .</p>	<pre> RESET  FACTS Extended BASIC Plus DL 05/06 PLCs Warp Drive CoProcessor Version 1.1/HS (c)Copyright FACTS Engineering, Inc. 1988 - 2005  AUTOSTART Mode, Program, Baud Mode = 0, Edit Program = 0 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600)  1 stored programs, 65459 program storage bytes free  PRM 0 READY &gt;                     </pre>
<p>Haga clic en <b>'List'</b> de la barra de menú. Observe que el modo cero usa la velocidad almacenada en Baud. El programa en el buffer, PROGRAM 0, se conserva en la memoria durante una pérdida de energía en el modo cero.</p>	<pre> &gt;list 10 PRINT1 65535 PRINT1  PRM 0 READY &gt;                     </pre>

## A Archivando un programa (saving)

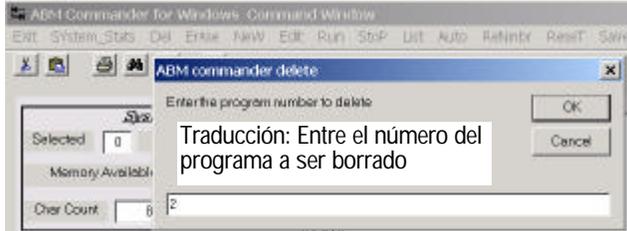
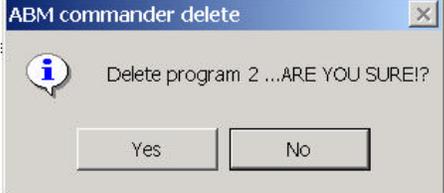
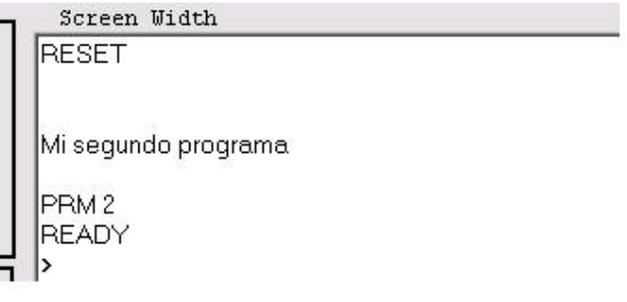
Acción del usuario	Lo que muestra la pantalla
Haga clic en <b>NeW</b> en la barra de menú.	
Entre lo siguiente en el campo "Command line" 10 p. "Mi programa 1"< ENTER>	<pre>NEW &gt; 10 p. "Mi programa 1" &gt;</pre>
Haga clic en <b>Save</b> en la barra de menú de ABM Commander.   <p><i>Nota: El módulo F0-CP128 es entregado por la fábrica con un programa de diagnóstico en PRM 1 (programa 1), de modo que el primer programa archivado será PRM2</i></p>	
Entre lo siguiente en el campo "Command line" 10 p. "Mi segundo programa "< ENTER>	<pre>PRM 0 READY &gt;10 p. "Mi segundo programa" &gt;</pre>
Haga clic en <b>Save</b> en la barra de menú de ABM Commander.	

# MODO AUTORUN.

**A**

Acción del usuario	Lo que muestra la pantalla
<p>Haga clic en <b>'Auto'</b> en la barra de menú.                      Seleccione Modo 1, Programa 2,</p>	
<p>Haga clic en <b>OK</b>. Esto especifica que el CoProcessor hará funcionar el programa 2 después de un Reset</p>	<pre>&gt;AUTOSTART 1,2 Mode = 1, RUN (CLEAR) Program = 2 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600) &gt;</pre>
<p>Haga clic en <b>'Reset'</b> en la barra de menú.                       Apagando y encendiendo el PLC también hace un reset del CoProcessor BASIC</p>	
<p>Haga clic en <b>'Sel'</b> en la barra de menú.                      Haga clic en el botón <b>'Program 0'</b> y luego haga <b>OK</b>.</p>	
<p>Haga clic en <b>'List'</b> en la barra de menú de ABM Commander.                      Confirme de que el programa en el buffer de modificación (PRM0) está aún presente.</p>	

## A BORRANDO UN PROGRAMA.

Acción del usuario	Lo que muestra la pantalla
<p>Haga clic en <b>Del</b> en la barra de menú. Entre 2 en el campo indicado.</p>	
<p>Haga clic en <b>OK</b> y acepte <b>Yes</b> en el cuadro de diálogo para dar una confirmación</p>	
<p>Esto es lo que resulta después de la acción anterior</p>	<pre data-bbox="792 909 1419 1081">&gt;DELPRM2 2 stored programs, 65431 program storage bytes free &gt;</pre>
<p>Haga clic en <b>Reset</b> en la barra de menú. Apagando y encendiendo el PLC también hará reset del CoProcessor BASIC .</p>	

## CANCELANDO EL MODO AUTORUN.

Acción del usuario	Lo que muestra la pantalla
<p>Haga clic en <b>Auto</b> en la barra de menú. Seleccione Modo 0, programa 0 y haga clic en <b>OK</b>. Esto especifica que el CoProcessor va a partir en modo Edit después de un reset</p>	<pre data-bbox="792 1579 1419 1841">AUTOSTART 0,0 Mode = 0, Edit Program = 0 Port 1 Baud = 9600 Programming (Port 2 = 9600) (Port 3 = 9600) &gt;</pre>

## CAMBIANDO EL PUERTO DE PROGRAMACIÓN.

**A**

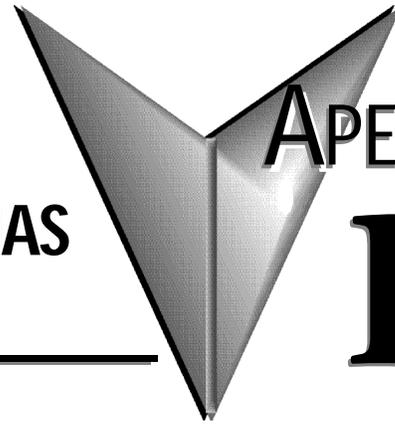
Al comunicarse con dos o tres dispositivos externos, usted puede cambiar el puerto de programación del puerto 1 al puerto 2 o aún al puerto 3. Será necesario un convertidor RS-232 a RS-485 para permitir una conexión RS-232 de una computadora PC al puerto 2. Cambiar el puerto del comando se hace según lo mostrado abajo.

Acción del usuario	Lo que muestra la pantalla
Haga clic en el campo ' <b>Port Select</b> ' en la barra de menú (a la izquierda inferior de la ventana de comando) para seleccionar el botón de radio "Port 3".	No hay cambio
En el campo ' <b>Port Select</b> ' haga clic en el botón Command Port (ABM). Haga clic en el diálogo de confirmación.	No hay cambio....
Mueva el cable desde el puerto 1 al puerto 3 y luego haga clic en ' <b>OK</b> ' en el cuadro de diálogo para aceptar el cambio.	No hay cambio
Seleccione ' <b>System Stats</b> ' desde la barra de menú.	>



# BÚSQUEDA DE PROBLEMAS

---



## APENDICE

# B

### En este apéndice:

No es posible comunicarse con el CoProcessor .....B-1

## No es posible comunicarse con el CoProcessor

**B**

1. Si el LED RXD del puerto 1 destella cuando se entran datos en el terminal entonces vaya al paso 2. Si no destella el LED use una caja de desbloqueo RS-232 para determinar si el problema está en el cable o la computadora.
2. Apague el PLC, quite el módulo desde la ranura y coloque el puente "**CLEAR ALL**" en ambos postes (vea la página 3-4)



---

**ADVERTENCIA:** Instalando el puente "**CLEAR ALL**" borrará el programa 0, todos los datos almacenados, cancela un **COMMAND@3**, BORRA el **LOCKOUT** y borra la información almacenada en **AUTOSTART**.

---

3. Haga funcionar el ABM Commander para Windows.
4. Repase la instrucciones del ABM Commander para Windows.
5. Conecte el cable de la computadora con el módulo de CoProcessor. Vea el APÉNDICE C para los diagramas eléctricos.
6. Energice al PLC.
7. Seleccione el menú "**Communication**" y luego seleccione "**Parameters(Port)**".
8. Seleccione el puerto serial de la PC que usted está usando. Haga clic en el botón "**Defaults**". Las configuraciones de comunicación ahora son 9600, 8, ninguno, 1, ninguno. Haga clic en el botón "**Apply**".
9. Seleccione "**COMMAND MODE> Connect to BASIC module**" desde la ventana principal. Seleccione "**S~~Y~~stem\_Stats**" del menú del **COMMAND MODE**.
10. El módulo ahora responderá con un aviso **READY**.
11. Teclee el comando siguiente y presione **ENTER**.
12. Apague y retire el módulo. Coloque el puente "**CLEAR ALL**" en un solo poste.
11. Instale el módulo y energice el PLC; el módulo ahora responderá con el mensaje.

```
> AUTOSTART 0.0
```

```
FACTSExtendedBASIC.
```

```
READY
```

```
>
```

(El carácter ">" indica que BASIC está en el modo de COMANDO)

# DIAGRAMAS DE CABLEADO RS-232 Y RS-485

---



## En este apéndice:

Estándar RS-232 .....	C-2
Cables de la computadora IBM o compatible .....	C-3
Estándar RS-485 .....	C-5
El blindaje del cable .....	C-7
Cables de conexión y resistencias de terminación .....	C-7

## Estándar RS-232

### C

RS-232-C (RS-232) es un estándar de interface de la Electronics Industries Association (EIA). El estándar nombra y define 20 señales de comunicación, asignadas a clavijas en un conector de 25 clavijas. Las cinco clavijas no asignadas pueden llevar señales que no sean las normales requeridas por cualquier sistema individual.

Cada señal se transmite como corriente eléctrica positiva o negativa entre 3 y 15 Volt (generalmente 12 Volt). La señal asignada a cada clavija fluye en una dirección solamente. La salida de las señales, por ejemplo, de una computadora debe entrar a un terminal y viceversa.

Las señales RS-232 son transmitidas por un cable de interface serial que puede tener hasta 25 conductores. Puesto que no se requiere la mayoría de las señales para comunicación simple, los cables tienen normalmente únicamente 2 o 3 alambres. Hay instalados puentes, a menudo en uno o ambos conectores, según lo mostrado en los diagramas siguientes, para asegurarse de que las señales de control de flujo estén satisfechas.

Las señales fluyen entre dos tipos puertos de interface, el equipo de comunicación de datos (DCE) y equipo terminal de datos (DTE). Los nombres de las clavijas son iguales para el equipo DCE y el DTE, sin embargo, la dirección del flujo de la señal es invertida.

### Nomenclaturas de las clavijas RS-232 DTE y DCE y flujo de señal

Clavija	Abreviatura	Nombre	Dirección de señal		Descripción
			DCE	DTE	
1	FG	Chasis	Ninguna	Ninguna	
2	TXD	Transmit data	Entrada	Salida	Ruta de datos de salida DTE
3	RXD	Receive data	Salida	Entrada	Ruta de datos de salida DCE
4	RTS	Request to send	Entrada	Salida	DTE tiene datos para transmitir
5	CTS	Clear to send	Salida	Entrada	DTE puede transmitir datos
6	DSR	Data set ready	Salida	Entrada	DCE tiene datos para transmitir
7	SG	Signal ground	Entrada	Salida	0 Volt
8	DCD	Data carrier detect	Salida	Entrada	El modem tiene portadora
20	DTR	Data terminal ready	Entrada	Salida	DCE puede transmitir datos
22	RI	Ring indicator	Salida	Entrada	

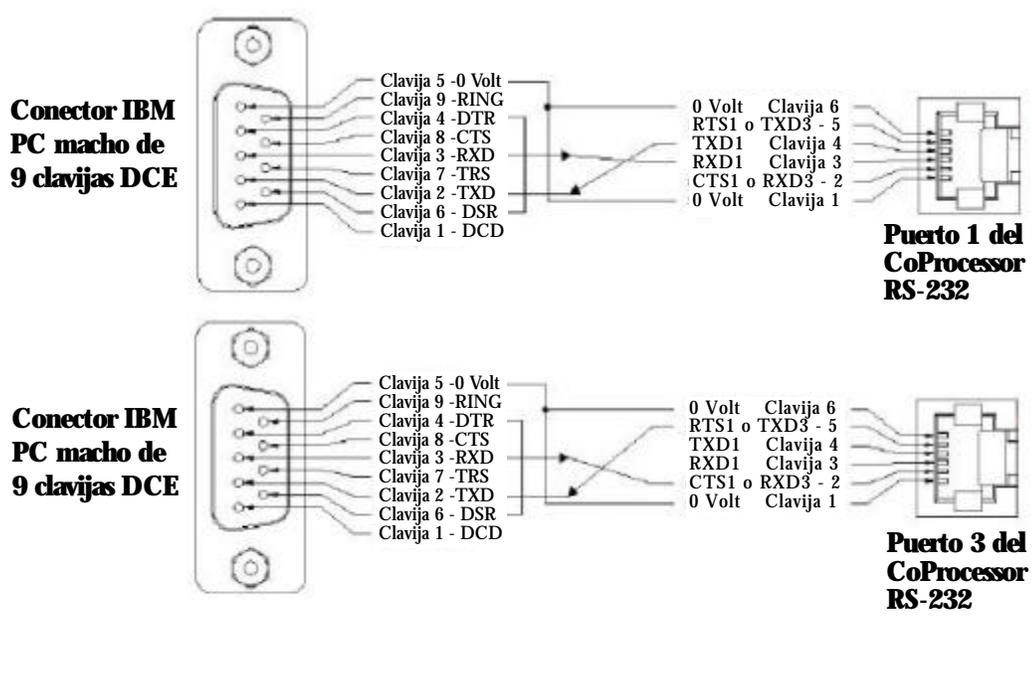
## Cables de la computadora PC (compatible con IBM)

El F0-CP128 incluye un cable RS-232 y un adaptador D-SUB de 9 clavijas para interconectar a las PC. Se incluye también un cable bifurcador del puerto 1 y del puerto 3 para permitir que la PC sea conectada con el puerto 1 o el puerto 3 del CoProcessor.

El juego de cables FA-CABKIT de AUTOMATIONDIRECT proporciona un cable RS-232 y adaptadores de interfaces a la mayoría de los dispositivos RS-232 incluyendo un adaptador para una PC. Esta es una manera rápida y fácil de hacer una conexión RS-232 entre el CoProcessor y un dispositivo externo. Si usted desea un cable blindado o necesita una longitud de cable diferente, use los diagramas eléctricos de las páginas siguientes para hacer un cable de interface.

La mayoría de las computadoras portátiles más nuevas no tienen un puerto RS-232. Para interconectar estas computadoras portátiles al CoProcessor, usted necesitará un adaptador USB RS-232 (número de artículo USB-RS232 de AUTOMATIONDIRECT).

El cable y el adaptador RS-232 incluido con el F0-CP128 se enchufa en el cable del adaptador de la PC, USB-RS232.



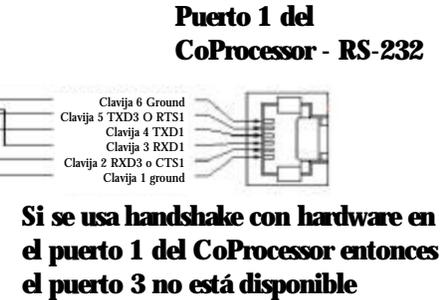
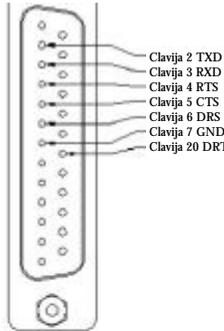
## Identificando un puerto de comunicación como DCE O DTE

Con un puerto RS-232 desconocido, con energía, mida el voltaje de C.C. entre la clavija 2 y 0 Volt (clavija 7) y la clavija 3 y 0 Volt. Si la clavija más negativa es 2 entonces el puerto es DTE. Si la clavija más negativa 3 entonces el puerto es conexión DCE. Una conexión errada de la clavija 2 y 3 no dañarán la interface.

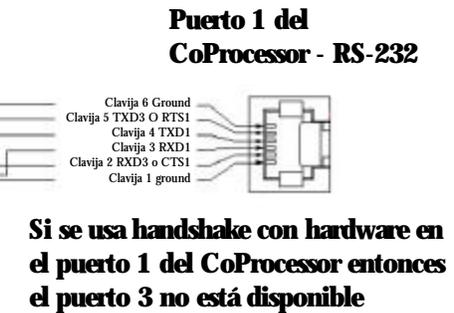
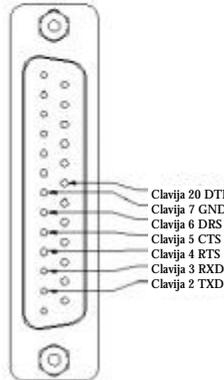
**C**

### RS-232 con hardware de “handshake”

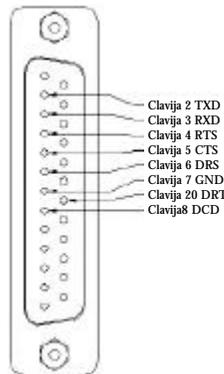
**Modem u otro aparato DCE usando handshake con hardware**  
**Conector hembra típico de 25 clavijas**



**Modem u otro aparato DCE usando handshake con hardware**  
**Conector macho típico de 25 clavijas**



**Aparato DCE usando handshake con hardware**  
**Conector hembra típico de 25 clavijas**



## ESTÁNDAR RS-485

Los transmisores-receptores RS-485 en el CoProcessor son compatibles con las señales RS-485.

RS-485 es una versión mejorada de EIA RS-422a y ofrece drivers de triple estado de corriente más alta internamente protegidos contra conflictos en la barra de comunicación causadas por múltiples drivers en la misma línea. Los drivers RS-485 también soportarán voltajes más altos en sus salidas cuando están inhabilitados (estado de alta impedancia). RS-485 es especificado para sistemas de múltiples transmisores y receptores así como aplicaciones de un y múltiples nodos. El estándar RS-485 permite hasta 32 drivers y receptores en la misma línea de transmisión.

C

## Comunicación RS-485

El CoProcessor tiene una interface de comunicaciones RS-485 en el puerto 2. La cancelación de eco RS-485 es automática. Se puede conectar un dispositivo RS-232 con este puerto con un convertidor RS-232 a RS-485 (número de artículo de AUTOMATIONDIRECT: FA-ISOCAN).

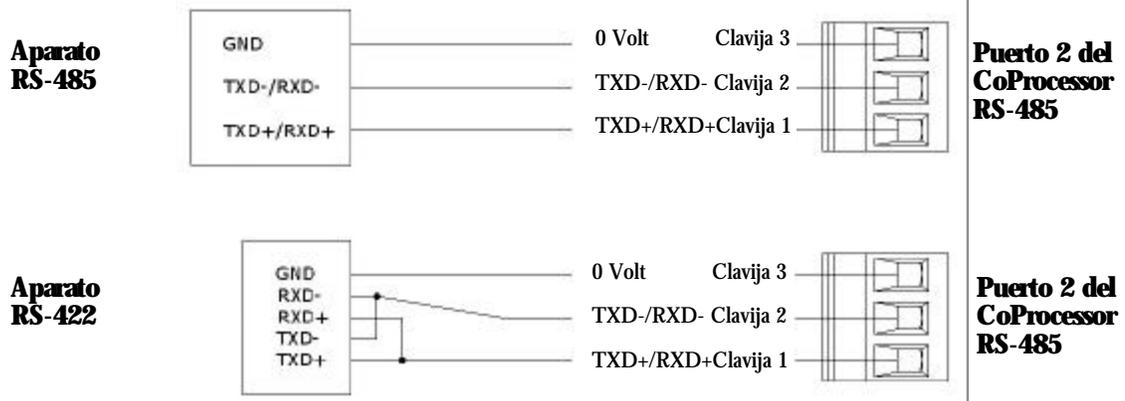
Para activar los transmisores RS-485 solamente cuando se usa la declaración PRINT, use SETPORT para seleccionar el modo "M" de multinodo. Use la opción de multinodo cuando el CoProcessor es un esclavo en una configuración maestro/esclavo o cuando se necesita una configuración "peer-to-peer" o punto a punto.

Para dejar los transmisores RS-485 ON aún cuando no se use PRINT, use SETPORT para seleccionar el modo "P" punto a punto. Use la opción punto a punto cuando el CoProcessor tiene una configuración de un solo maestro o es un maestro y esclavo único. Esta configuración proporciona una gran inmunidad contra ruido porque los drivers RS-485 siguen activados y evitan que el ruido sea recibido por los dispositivos esclavos en la red.

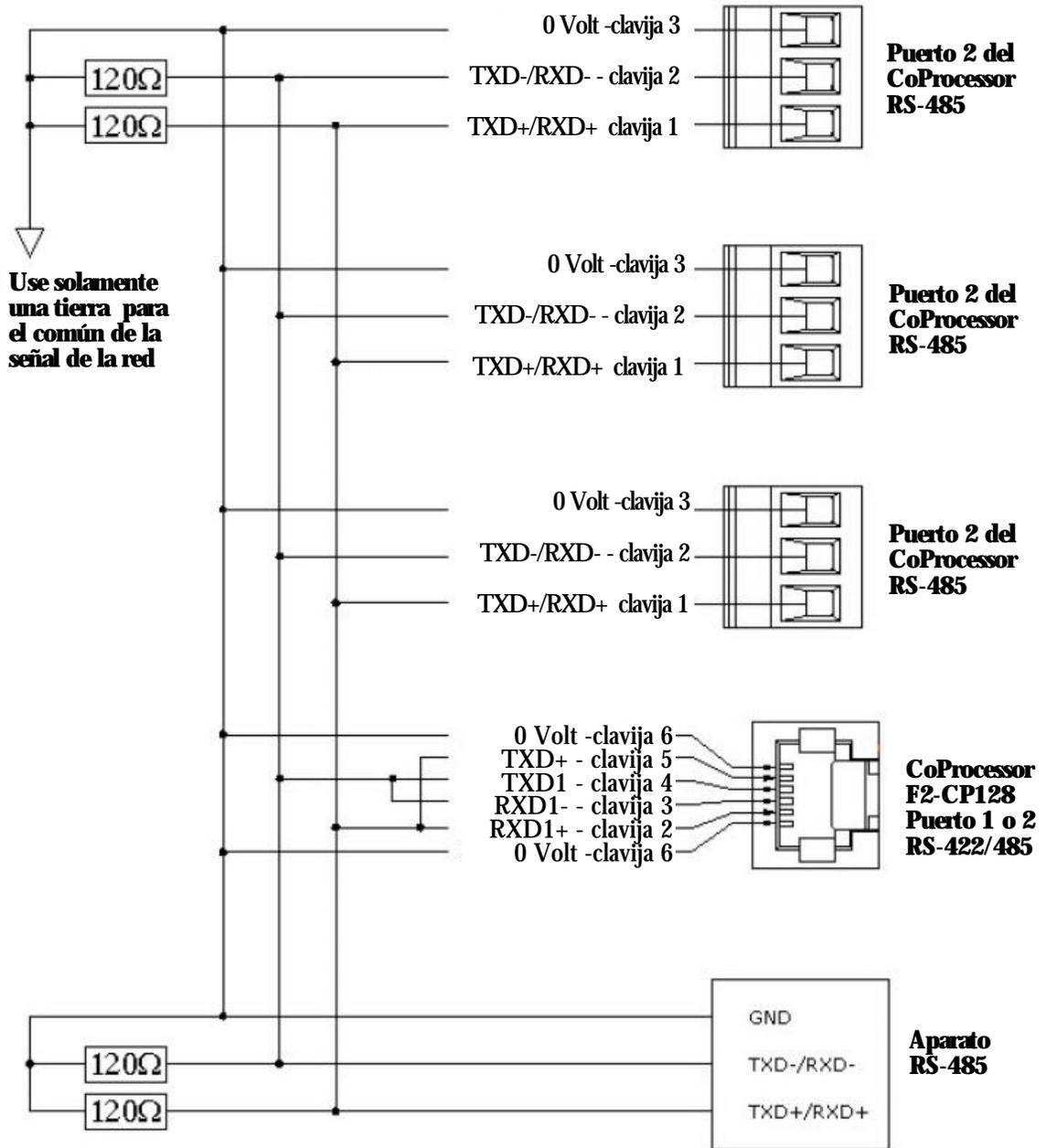
**Ejemplo:** Configurar el puerto 1 para 9600 Bauds, ninguna paridad, palabra de 8 bits, 1 bit de parada, handshake por software XON/XOFF y un modo de múltiples nodos RS-485.

SETPORT 1, 9600, N, 8, 1, S, M

## Diagrama de cables para conexión punto a punto Rs-485



**C**



## El blindaje del cable

El blindaje mejora la inmunidad de ruido (protección del campo magnético). Es importante aterrar el blindaje en el extremo del receptor solamente. Poner a tierra el extremo del receptor proporciona menos atenuación de la señal de alta frecuencia y un mejor rechazo de señales indeseadas. Poner a tierra ambos extremos del blindaje causará que circulen corrientes inducidas por el campo magnético a través de tierra. Puede entonces aparecer ruido en las líneas de datos debido a inducciones. Si se usa el blindaje del cable como conductor de tierra del sistema entonces que coloca una resistencia de 100 Ohm en serie con el blindaje y la conexión de tierra reducirá las corrientes de aterramiento que producen ruido.

C

## Cables de conexión y resistencias de terminación de línea

Se recomienda usar un cables con un par trenzado más la conexión de tierra para redes de dos hilos RS-485. Se requiere una terminación apropiada de línea equilibrada de transmisión para prevenir errores de datos. Un alambre típico sólido de sección 22AWG con cubierta plástica de 0,06 pulgadas, torcida 4,5 veces por pie tiene una impedancia característica de cerca de 120 Ohm. En este caso la selección de las cuatro resistencias de terminación es de 120 Ohm (dos resistencias de 120 Ohm en paralelo en cada línea es 60 Ohm).

Se prefiere una resistencia de terminación de línea a tierra en vez de una resistencia, a menudo mostrada, de línea a línea, de 120 Ohm. En aplicaciones de redes ruidosas o cables de gran longitud, la mejor capacidad del rechazo de modo común de la línea a tierra es particularmente importante. En redes de multinodos, la línea se debe terminar en las extremidades solamente según lo mostrado en los dos diagramas anteriores. Al contrario, agregando resistencias de terminación intermedias cargará la línea.

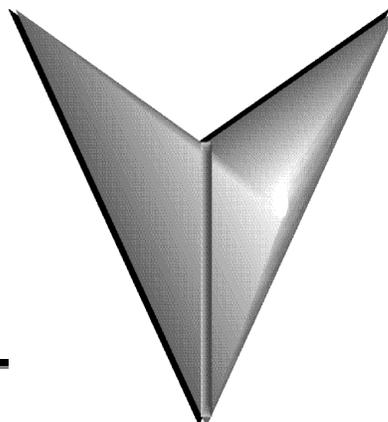
Algunos dispositivos RS-485 no tienen una conexión de tierra. Estos dispositivos tienen una conexión positiva y una negativa RS-485, pero ninguna conexión de tierra.

En este caso, se debe usar un aislador de red tal como el FA-ISOCAN de AUTOMATIONDIRECT para eliminar la conexión de tierra en el CoProcessor. El FA-ISOCAN proporciona aislación entre el puerto del CoProcessor RS-232 y la red RS-485. El puerto RS-485 en el CoProcessor no se usa con el FA-ISOCAN.



# ÍNDICE

---



<b>A</b>		<b>E</b>	
Accesorios	3-2	El blindaje del cable	C-7
APÉNDICE A: Usando ABM Commander		Especificaciones generales del Coprocessor	3-2
APÉNDICE B: Búsqueda de problemas		Estandar RS-232	C-2
APÉNDICE C: Diagramas de cableado		Estandar RS-485	C-5
Apoyo técnico	1-3	<b>I</b>	
Archivando un programa (save)	A-4	Introducción	1-2
<b>B</b>		<b>M</b>	
Blindaje del cable	C-7	Modificando un programa	A-3
BMOVE	2-2	Modo autorun	A-5
Borrando un programa	A-6	<b>N</b>	
<b>C</b>		No es posible comunicarse con el CoProcessor	B-1
Cables de conexión y resistencias de terminación	C-7	Numeración octal y tipos de datos para los operandos BMOVE	2-3
Cables de la computadora IBM o compatible	C-3	Numeración octal y tipos de datos para los operandos S06	2-8
Cambiando el puerto de programación	A-7	<b>O</b>	
Cancelando el modo autorun	A-7	ONPLC	2-8
CAPÍTULO 1: Introducción		Operación inicial usando ABM Commander Plus	A-2
Capítulo 2: Declaraciones de coProcesor		Operandos BMOVE de DL05	2-3
CAPÍTULO 3: Especificaciones		Operandos BMOVE de DL06	2-3
Clavijas de los puertos del F0-CP128	3-5	Operandos S06 _ de la CPU DL05	2-9
Coma Flotante de IEEE	2-2, 2-5, 2-8	Operandos S06 _ de la CPU DL06	2-9
Convenciones usadas	1-4	<b>P</b>	
<b>D</b>		Propósito de este documento	1-3
Descripción del F0-CP128	3-3	Puente CLEAR ALL	3-4
Descripción y localización del puente CLEAR del F0-CP128	3-4		
DPORT	2-5		

## Índice

---

### Q

Quién debe leer este manual 1-3

### S

Sincronización de la CPU 1-4

S06 \_ 2-8