

DIVISIÓN DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA



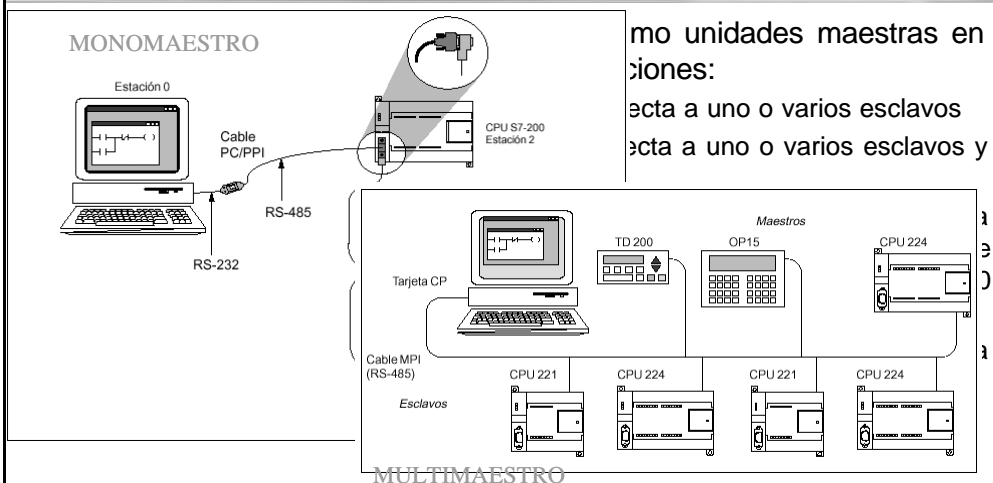
INDICE

- S7-200. Comunicaciones
- Protocolos de comunicación
 - PPI
 - MPI
 - PROFIBUS
 - Protocolos definidos por el usuario (Freeport)
 - USS (Comunicación con variadores)
- Componentes de la red
 - Puerto de comunicación
 - Conectores de bus
- Rendimiento de la red
 - Optimizar el rendimiento de la red

INDICE

- Rotación del testigo
- Enviar mensajes
- Tiempo de rotación del testigo
- Tiempo de rotación del testigo en función del número de estaciones
- NETR y NETW
 - Ejemplo
- Modo Freeport (XMT y RCV)
 - Ejemplo

S7-200. Comunicaciones





Protocolos de comunicación

- Dependiendo de la CPU S7-200 utilizada, la red puede asistir uno o varios de los siguientes protocolos de comunicación:
 - Interface punto a punto (PPI)
 - Interface multipunto (MPI)
 - PROFIBUS
- Los protocolos PPI y MPI se implementan en una red "token ring" (red de anillo con testigo) conforme al estándar Process Field Bus (PROFIBUS) que se describe en la norma europea EN 50170.
- Los tres protocolos se pueden utilizar simultáneamente en una red sin que interfieran entre sí, con la condición de que usen una misma velocidad de transferencia.



Protocolos de comunicación

- Los protocolos prevén dos tipos de dispositivos de red: los maestros y los esclavos.
 - Los maestros pueden enviar una petición a otros dispositivos.
 - En cambio, los esclavos sólo pueden responder a las peticiones de los maestros, sin poder lanzar nunca una petición por su propia cuenta.
- Todos los dispositivos que formen parte de una red deberán tener direcciones unívocas para poder comunicarse entre sí.
 - El ajuste estándar
 - las unidades de programación SIMATIC y los PCs con STEP 7-Micro/WIN 32 es la dirección "0".
 - Los visualizadores de textos TD 200 y los paneles de operador OP3 y OP7 tienen la dirección predeterminada "1".
 - La dirección estándar de los sistemas de automatización es "2".



PPI

- PPI es un protocolo maestro/esclavo
 - Los maestros (otras CPUs, unidades de programación SIMATIC o visualizadores de textos TD 200) envían peticiones a los esclavos y éstos últimos responden
 - Todas las CPUs S7-200 actúan de estaciones esclavas en la red.
 - Estando en modo RUN, algunas CPUs S7-200 pueden actuar de estaciones maestras en la red si se encuentra habilitado el modo maestro PPI en el programa de usuario
 - enviar mensajes a otras CPUs, usando las operaciones Leer de la red (NETR) y Escribir en la red (NETW). Mientras actúa de estación maestra PPI, la CPU S7-200 sigue respondiendo en calidad de esclava a las peticiones de otros maestros.
 - la red no puede comprender más de 32 maestros.



MPI

- MPI puede ser un protocolo maestro/maestro, o bien maestro/esclavo
 - CPU S7-300, se establece un enlace maestro/maestro, puesto que todas las CPUs S7-300 son estaciones maestras en la red.
 - CPU S7-200, se establece un enlace maestro/esclavo, ya que las CPUs S7-200 son unidades esclavas.
- El protocolo MPI crea siempre un enlace entre los dos dispositivos intercomunicados.
 - Un enlace es una conexión privada entre los dos dispositivos. Ningún otro maestro puede interferir en un enlace establecido entre dos dispositivos.
 - Un maestro puede establecer un enlace para utilizarlo durante un tiempo breve o indefinido.
- Las CPUs S7-300 y S7-400 pueden leer y escribir datos en las CPUs S7-200, utilizando las operaciones XGET y XPUT

PROFIBUS

- El protocolo PROFIBUS se ha diseñado para la comunicación rápida con unidades periféricas descentralizadas (E/S remotas)
 - Hay numerosos dispositivos PROFIBUS ofrecidos por diversos fabricantes. Dichos dispositivos abarcan desde módulos sencillos de entradas o salidas hasta controladores de motores y sistemas de automatización (autómatas programables).
- PROFIBUS-DP
 - Módulo de expansión EM277
 - Hasta 128 kbytes de lectura y escritura
 - Fácil integración como esclavo PROFIBUS DP

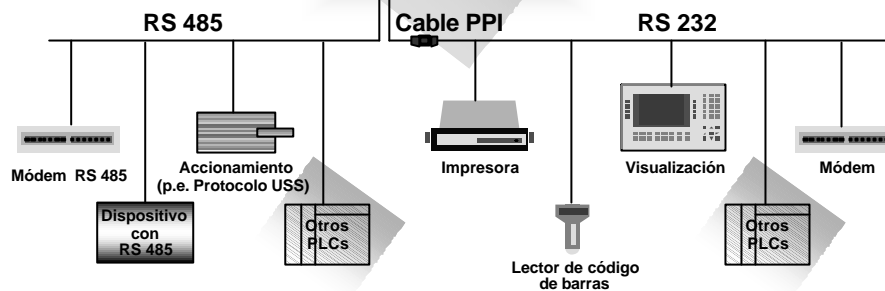


Freeport

- Interfase RS 485 para programación libre
 - Modo Freeport
 - Velocidad binaria hasta 38.4 Kbit/s
 - Sin requerimientos adicionales
 - Conexión a módem para control remoto de la instalación
 - Posibilidades de conexión: impresora, lector de código de barras, PC, PLCs de otras marcas, dispositivos de control y visualización...
- Con el modo Freeport se pueden implementar protocolos de comunicación definidos por el usuario para crear enlaces con numerosos dispositivos inteligentes.

Freeport

Alternativa:
protocolo RS 485 o
RS 232



Freeport

- El programa de usuario controla el funcionamiento del puerto de comunicación utilizando interrupciones de recepción y de transmisión,
- así como las operaciones Transmitir mensaje (XMT) y Recibir mensaje (RCV).
- En modo Freeport, el programa de usuario controla por completo el protocolo de comunicación. El modo Freeport se habilita con la marca SMB30 (puerto 0), estando activo únicamente cuando la CPU está en modo RUN.
- Cuando la CPU retorna a modo STOP, la comunicación Freeport se detiene y el puerto de comunicación vuelve a utilizar el protocolo PPI normal.

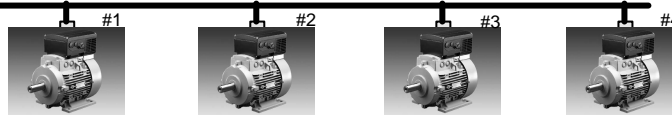
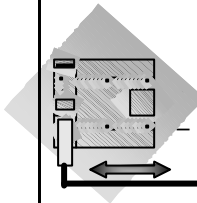
Freeport

■ Protocolo USS

- Control completo del accionamiento
 - drive on, off, quick off
 - dirección
 - velocidad
 - status
 - lee/escribe los parámetros elegidos

Control completo del accionamiento con un sólo cable

Control de varios accionamientos con una sólo CPU



Freeport

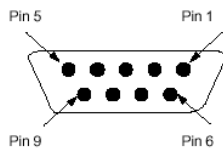
■ Instrucciones de la librería USS:

- USS_INIT
 - habilita e inicializa o deshabilita el puerto para el protocolo USS
- DRV_CTRL
 - control del accionamiento
- READ_PM
 - lee un parámetro del accionamiento
- WRITE_PM
 - escribe un parámetro en el accionamiento



Puerto de comunicación

- Los puertos de comunicación de las CPUs S7-200 son compatibles con el estándar RS-485 mediante un conector D subminiatura de 9 pines conforme al estándar PROFIBUS definido en la norma europea EN 50170



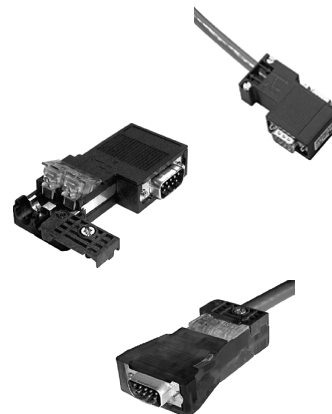
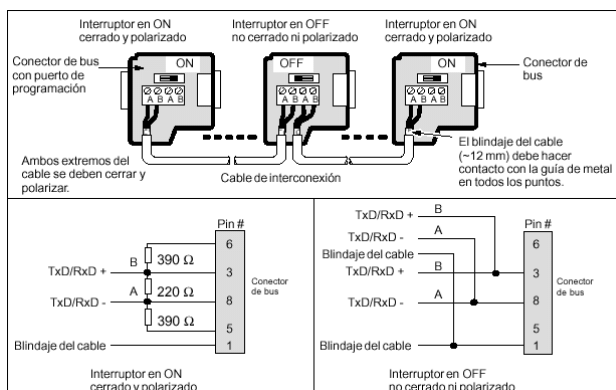
Pin	Denominación PROFIBUS	Puerto 0
1	Blindaje	Hilo lógico
2	Hilo de retorno 24 V	Hilo lógico
3	Señal B RS-485	Señal B RS-485
4	Petición de transmitir	RTS (TTL)
5	Hilo de retorno 5 V	Hilo lógico
6	+5 V	+5 V, 100 Ω resistor en serie
7	+24 V	+24 V
8	Señal A RS-485	Señal A RS-485
9	No aplicable	Selección protocolo de 10 bits (entrada)
Carcasa del enchufe	Blindaje	Tierra



Cursos de Perfeccionamiento
ISA-UMH © TDOC-2001

15

Conectores de bus



Cursos de Perfeccionamiento
ISA-UMH © TDOC-2001

16



Optimizar el rendimiento de la red

- El rendimiento óptimo de la red se logra utilizando la velocidad de transferencia máxima asistida por todos los dispositivos.
- Si el número de maestros se reduce a un mínimo, aumenta también el rendimiento de la red.
- Cada maestro de la red prolonga el tiempo de procesamiento en la red. Por tanto, dicho tiempo se acortará cuanto menor sea el número de maestros.
- Los siguientes factores influyen también en el rendimiento de la red:
 - Las direcciones elegidas para los maestros y esclavos.
 - El factor de actualización GAP. (Factor de actualización para determinar si existen otros maestros en la red. Huecos)
 - La dirección de estación más alta.



Optimizar el rendimiento de la red

- Las direcciones de los esclavos se pueden ajustar a cualquier valor sin que ello influya en el rendimiento de la red,
 - a menos que los esclavos se encuentren entre los maestros. En este último caso aumentaría también el tiempo de procesamiento de la red como si existieran huecos entre las direcciones de los maestros.
- El factor de actualización GAP le indica a la CPU la frecuencia con la que debe comprobar el hueco de direcciones para determinar si hay otros maestros.
 - Si se elige "1" como factor de actualización GAP, la CPU comprobará el hueco de direcciones cada vez que tenga el testigo en su poder. GAP ↑ reduce el tiempo de procesamiento en la red si hay huecos entre las direcciones de los maestros.
 - Si no existen huecos, el factor de actualización GAP no tendrá efecto alguno en el rendimiento.
 - Si se ajusta un factor de actualización GAP elevado pueden producirse grandes demoras cuando se desee incorporar nuevos maestros a la red, puesto que las direcciones se comprueban con menor frecuencia.
- El factor de actualización GAP se utiliza únicamente cuando una CPU actúa de maestro PPI.

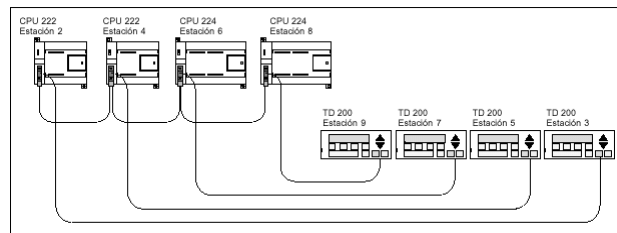
Optimizar el rendimiento de la red

- La dirección de estación más alta es el valor donde un maestro debe buscar a otro.
 - Ajustándose dicho valor se limita el hueco de direcciones que el último maestro (la dirección más alta) debe comprobar en la red.
 - Limitando el tamaño del hueco de direcciones se reduce el tiempo necesario para buscar e incorporar en la red a un nuevo maestro.
 - se deberá ajustar en todos los maestros un mismo valor para la dirección de estación más alta.
 - Dicha dirección debería ser mayor o igual a la dirección más alta de los maestros. El ajuste estándar de la dirección de estación más alta en las CPUs S7-200 es "31".

Rotación del testigo

- En una red con token passing (paso de testigo), la estación que tiene el testigo en su poder es la única que puede iniciar la comunicación.
- Por tanto, un importante factor en una red con token passing es el tiempo de rotación del testigo.
 - Éste es el tiempo que el testigo necesita para recorrer el anillo lógico, o sea, para circular por todos los maestros (token holders) que lo constituyen

En esta configuración, un TD 200 (estación 3) se comunica con una CPU 222 (estación 2), otro TD 200 (estación 5) se comunica con la otra CPU 222 (estación 4), etc. Además, una CPU 224 (estación 6) envía mensajes a las estaciones 2, 4 y 8, y la otra CPU 224 (estación 8) envía mensajes a las estaciones 2, 4 y 6. Esta red comprende seis estaciones maestras (los cuatro TDs 200 y las dos CPUs 224), así como dos estaciones esclavas (las dos CPUs 222).





Enviar mensajes

- Para que un maestro pueda enviar un mensaje deberá tener el testigo en su poder.
- Ejemplo:
 - cuando la estación 3 tiene el testigo en su poder, envía una petición a la estación 2 y pasa el testigo a la estación 5.
 - La estación 5 envía una petición a la estación 4 y pasa el testigo a la estación 6.
 - La estación 6 envía un mensaje a las estaciones 2, 4 u 8 y pasa el testigo a la estación 7.
 - Este proceso de enviar un mensaje y pasar el testigo continúa por el anillo lógico de la estación 3 a la estación 5, a la estación 6, a la estación 7, a la estación 8, a la estación 9 y de allí retorna finalmente a la estación 3.
 - El testigo debe recorrer todo el anillo lógico para que un maestro pueda enviar una petición de información. En un anillo lógico compuesto por seis estaciones que envían una petición para leer o escribir un valor de doble palabra (cuatro bytes de datos) cada vez que tienen el testigo en su poder, el tiempo de rotación del mismo será de unos 900 milisegundos a una velocidad de transferencia de 9.600 bit/s.



Tiempo de rotación del testigo

- Depende del tiempo que cada estación tiene el testigo en su poder.
 - El tiempo de rotación del testigo en redes S7-200 multimaestro se puede determinar sumando los tiempos de posesión del testigo por parte de cada maestro.
- Calcular el tiempo aproximado de rotación del testigo, dando por supuesto que:
 - Cada estación envía una petición cuando tiene el testigo en su poder.
 - La petición es una operación de lectura o de escritura a direcciones consecutivas de datos.
 - No hay conflictos de acceso al único búfer de comunicación de la CPU.
 - Ninguna CPU tiene un tiempo de ciclo superior a aprox. 10 ms.

Tiempo de rotación del testigo

Tiempo de posesión del testigo (T_{pos}) = (tiempo necesario 128 + n caráct. datos) — 11 bits/caráct. — 1/vel. transf.

Tiempo de rotación del testigo (T_{rot}) = T_{pos} del maestro 1 + T_{pos} del maestro 2 + ... + T_{pos} del maestro m

siendo n el número de caracteres de datos (bytes)
y m el número de maestros

Conforme al ejemplo indicado arriba, el tiempo de rotación se calcula de la siguiente forma si el tiempo de posesión del testigo es igual en los seis maestros:

T (tiempo de posesión del testigo) = (128 + 4 caracteres) — 11 bits/carácter — 1/9.600 "bit times"/s
= 151,25ms/maestro

T (tiempo de rotación del testigo) = 151,25 ms/maestro — 6 maestros
= 907.5 ms

(Un "bit time" equivale a la duración de un período de señal).

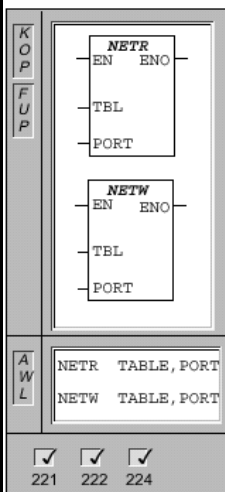
Tiempo rotación testigo en función del nº estac

Bytes transferidos por estación a 9,6 kbit/s	Número de estaciones (indicaciones de tiempo en segundos)									
	2 estaciones	3 estaciones	4 estaciones	5 estaciones	6 estaciones	7 estaciones	8 estaciones	9 estaciones	10 estaciones	
1	0,30	0,44	0,59	0,74	0,89	1,03	1,18	1,33	1,48	
2	0,30	0,45	0,60	0,74	0,89	1,04	1,19	1,34	1,49	
3	0,30	0,45	0,60	0,75	0,90	1,05	1,20	1,35	1,50	
4	0,30	0,45	0,61	0,76	0,91	1,06	1,21	1,36	1,51	
5	0,30	0,46	0,61	0,76	0,91	1,07	1,22	1,37	1,52	
6	0,31	0,46	0,61	0,77	0,92	1,07	1,23	1,38	1,54	
7	0,31	0,46	0,62	0,77	0,93	1,08	1,24	1,39	1,55	
8	0,31	0,47	0,62	0,78	0,94	1,09	1,25	1,40	1,56	
9	0,31	0,47	0,63	0,78	0,94	1,10	1,26	1,41	1,57	
10	0,32	0,47	0,63	0,79	0,95	1,11	1,27	1,42	1,58	
11	0,32	0,48	0,64	0,80	0,96	1,11	1,27	1,43	1,59	
12	0,32	0,48	0,64	0,80	0,96	1,12	1,28	1,44	1,60	
13	0,32	0,48	0,65	0,81	0,97	1,13	1,29	1,45	1,62	
14	0,33	0,49	0,65	0,81	0,98	1,14	1,30	1,46	1,63	
15	0,33	0,49	0,66	0,82	0,98	1,15	1,31	1,47	1,64	
16	0,33	0,50	0,66	0,83	0,99	1,16	1,32	1,49	1,65	

Tiempo rotación testigo en función del nº estac

Bytes transferidos por estación a 187,5 kbit/s	Número de estaciones (indicaciones de tiempo en milisegundos)									
	2 estaciones	3 estaciones	4 estaciones	5 estaciones	6 estaciones	7 estaciones	8 estaciones	9 estaciones	10 estaciones	
1	8,68	13,02	17,37	21,71	26,05	30,39	34,73	39,07	43,41	
2	8,80	13,20	17,60	22,00	26,40	30,80	35,20	39,60	44,00	
3	8,92	13,38	17,83	22,29	26,75	31,21	35,67	40,13	44,59	
4	9,03	13,55	18,07	22,59	27,10	31,62	36,14	40,66	45,17	
5	9,15	13,73	18,30	22,88	27,46	32,03	36,61	41,18	45,76	
6	9,27	13,90	18,54	23,17	27,81	32,44	37,08	41,71	46,35	
7	9,39	14,08	18,77	23,47	28,16	32,85	37,55	42,24	46,93	
8	9,50	14,26	19,01	23,76	28,51	33,26	38,02	42,77	47,52	
9	9,62	14,43	19,24	24,05	28,86	33,67	38,49	43,30	48,11	
10	9,74	14,61	19,48	24,35	29,22	34,09	38,95	43,82	48,69	
11	9,86	14,78	19,71	24,64	29,57	34,50	39,42	44,35	49,28	
12	9,97	14,96	19,95	24,93	29,92	34,91	39,89	44,88	49,87	
13	10,09	15,14	20,18	25,23	30,27	35,32	40,36	45,41	50,45	
14	10,21	15,31	20,42	25,52	30,62	35,73	40,83	45,84	51,04	
15	10,33	15,49	20,65	25,81	30,98	36,14	41,30	46,46	51,63	

NETR y NETW



- La operación **Leer de la red** inicia una comunicación para registrar datos de una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL).
- La operación **Escribir en la red** inicia una comunicación para escribir datos en una estación remota a través del puerto indicado (PORT), según se define en la tabla (TBL).
- 16 BYTES y sólo 8 operaciones simultáneas

Entradas/salidas	Operandos	Tipos de datos
TBL	I, Q, M, S, V, VB, MB, *VD, *AC, *LD	BYTE
PORT	constante	BYTE

NETR y NETW

Desplazamiento de byte	7	0
1	Dirección de la estación remota	
2	Puntero al área de datos	
3	en la	
4	estación remota	
5	(I, Q, M, o V)	
6	Longitud de datos	
7	Byte de datos 0	
8	Byte de datos	
	↓	
22	Byte de datos 15	

D Concluida (operación ejecutada): 0 = no ejecutada 1 = ejecutada
 A Activa (operación en cola de espera): 0 = no activa 1 = activa
 E Error (operación incluye un error): 0 = sin error 1 = error

Dirección de la estación remota: dirección del PLC a cuyos datos se desea acceder.

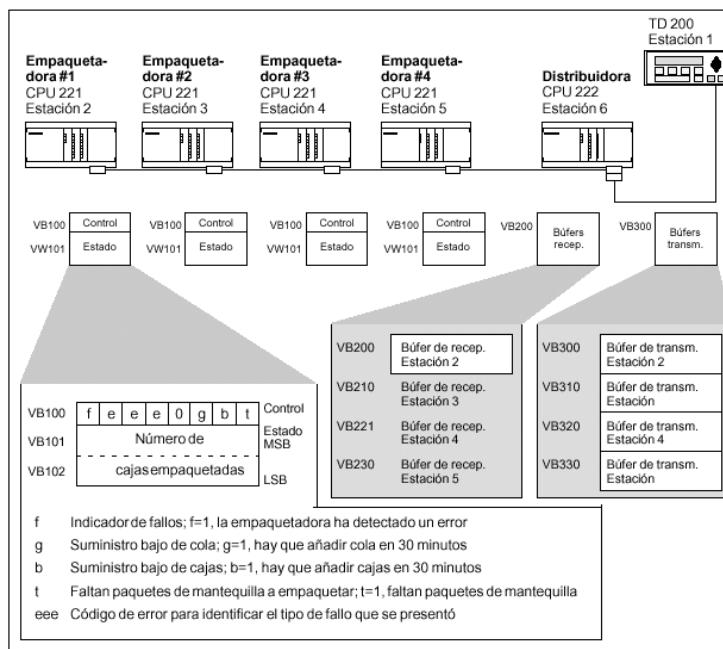
Puntero al área de datos en la estación remota: puntero indirecto a los datos que se desea acceder.

Longitud de datos: número de bytes de datos a acceder en la estación remota (1 a 16 bytes)

Área de datos a recibir o transferir: 1 a 16 bytes reservados para los datos, como se describe a continuación:

Para NETR, esta área de datos es donde se almacenarán los valores leídos de la estación remota después de la operación NETR.
 Para NETW, esta área de datos es donde se almacenarán los valores a enviar a la estación remota después de la operación NETW.

Código de error	Definición
0	Sin error
1	Error de timeout; no responde la estación remota
2	Error de recepción; error de paridad, de ajuste o de suma de verificación en la respuesta
3	Error offline; colisiones causadas por direcciones de estación repetidas o fallo del hardware
4	Error de desbordamiento en la cola de espera; se han activado más de ocho cuadros NETR/NETW
5	Violación de protocolo; se intentó ejecutar NETR/NETW sin habilitar PPI+ en SMB30
6	Parámetro no válido; la tabla NETR/NETW contiene un valor no válido
7	Sin recursos; la estación remota está ocupada (secuencia de cargar en PG o de cargar en CPU en curso)
8	Error de capa 7; violación de aplicación de protocolo
9	Error de mensaje; dirección de datos errónea o longitud de datos incorrecta
A-F	No utilizado; (reservado para futuras aplicaciones)

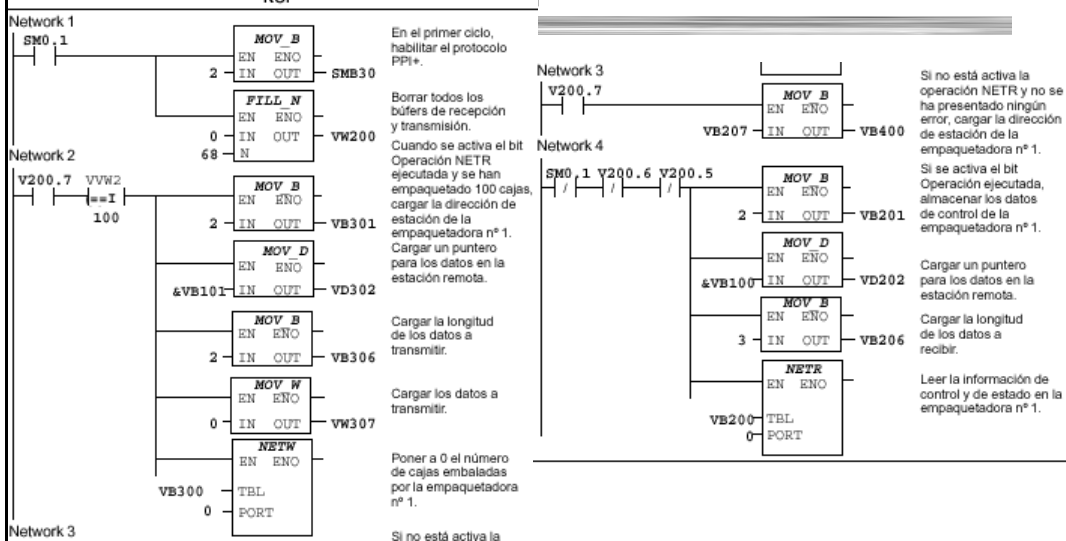


se están
de las
cada
e
oras y
para

Ejemplo

- los búfers de recepción y transmisión para acceder a los datos de la estación 2 (en VB200 y VB300, respectivamente).
- La CPU 224 utiliza la operación NETR para leer continuamente el control y el estado de cada una de las empaquetadoras. Cada vez que una empaquetadora ha embalado 100 cajas, la máquina distribuidora lo registra y envía un mensaje para borrar la palabra de estado utilizando una operación NETW.
- Ejemplo:
 - El programa para leer el byte de control, el número de cajas embaladas y para poner a 0 el número de cajas embalado por una sola empaquetadora (empaquetadora nº 1).

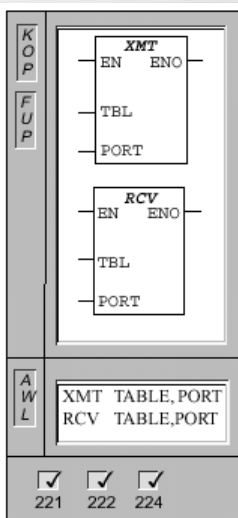
Ejemplo



Modo Freeport (XMT y RCV)

- Freeport
 - El programa de usuario puede controlar el puerto serie de la CPU.
 - el programa KOP controla el puerto de comunicación utilizando:
 - interrupciones de recepción y de transmisión
 - Transmitir mensaje (XMT) y Recibir mensaje (RCV).
 - Las marcas especiales SMB30 (para el puerto 0) y SMB130 (para el puerto 1, si la CPU dispone de dos puertos) se utilizan para elegir la velocidad de transferencia y la paridad.
 - En el caso más simple se puede enviar un mensaje a la impresora o a la pantalla, la conexión a un lector de código de barras, una báscula o una soldadora.

Modo Freeport (XMT y RCV)



Transmitir mensaje, Recibir mensaje

- La operación **Transmitir mensaje(XMT)** activa la transmisión del búfer de datos (TBL). La primera entrada del búfer indica cuántos bytes se han de transmitir. PORT indica el puerto de programación por donde se va a transmitir.
- La operación **Recibir mensaje(RCV)** inicia o finaliza la función Recibir mensaje. Para el cuadro Recibir mensaje es preciso indicar una condición inicial y final. Los mensajes que se hayan recibido a través del puerto indicado (PORT) se almacenan en el búfer de datos (TBL). La primera entrada del búfer indica el número de bytes que se han recibido.

Entradas/salidas	Operandos	Tipos de datos
TABLE	VB, IB, QB, MB, SB, SMB, *VD, *AC, *LD	BYTE
PORT	Constante (0)	BYTE

Modo Freeport (XMT y RCV)

■ Inicializar el modo Freeport

Puerto 0	Puerto 1	Descripción
Formato de SMB30	Formato de SMB130	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB ↑ LSB ↓ </div> <div style="display: flex; align-items: center; margin-top: 5px;"> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">p</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">p</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">d</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">b</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">b</div> <div style="border: 1px solid black; padding: 2px; margin-right: 5px;">m</div> <div style="border: 1px solid black; padding: 2px;">m</div> </div> <p>Byte de control del modo Freeport</p>
SM30.6 y SM30.7	SM130.6 y SM130.7	<p>pp Selección de paridad</p> <p>00 = sin paridad</p> <p>01 = paridad par</p> <p>10 = sin paridad</p> <p>11 = paridad impar</p>
SM30.5	SM130.5	<p>d Bits por carácter</p> <p>0 = 8 bits por carácter</p> <p>1 = 7 bits por carácter</p>
SM30.2 a SM30.4	SM130.2 a SM130.4	<p>bbb Velocidad de transferencia</p> <p>000 = 38.400 bit/s</p> <p>001 = 19.200 bit/s</p> <p>010 = 9.600 bit/s</p> <p>011 = 4.800 bit/s</p> <p>100 = 2.400 bit/s</p> <p>101 = 1.200 bit/s</p> <p>110 = 600 bit/s</p> <p>111 = 300 bit/s</p>
SM30.0 y SM30.1	SM130.0 y SM130.1	<p>mm Selección de protocolo</p> <p>00 = Protocolo de interface punto a punto (PPI/modo esclavo)</p> <p>01 = Protocolo Freeport</p> <p>10 = PPI/modo maestro</p> <p>11 = Reservado (estándar: PPI/modo esclavo)</p>

Cursos de Perfeccionamiento
ISA-UMH © TDOC-2001

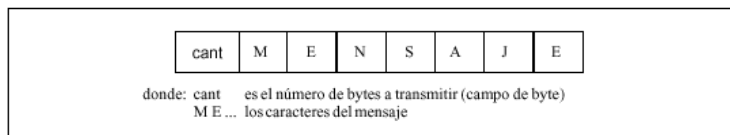
33

Nota: Se genera un bit de parada para todas las configuraciones.

Modo Freeport (XMT y RCV)

■ Utilizar la operación XMT para transmitir datos

- Con la operación XMT se puede enviar un búfer de uno o más caracteres (hasta un máximo de 255).
- Una vez transmitido el ultimo carácter del búfer, se genera una interrupción (evento de interrupción 9 para el puerto 0 y evento de interrupción 26 para el puerto 1), si una rutina de interrupción se ha asociado al evento Transmisión finalizada.
- También es posible transmitir datos sin utilizar interrupciones (p.ej. enviar un mensaje a una impresora), vigilando SM4.5 ó SM4.6 hasta que finalice la transmisión.



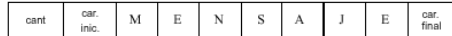
Cursos de Perfeccionamiento
ISA-UMH © TDOC-2001

34

Modo Freeport (XMT y RCV)

■ Utilizar la operación RCV para recibir datos

Puerto 0	Puerto 1	Descripción
SMB86	SMB186	<div style="display: flex; justify-content: space-between; align-items: center;"> MSB 7 LSB 0 </div> <div style="text-align: center; border: 1px solid black; padding: 2px; margin: 5px 0;"> n r e 0 0 t c p </div> <p>Byte de estado de recepción de mensajes</p> <p>n: 1 = El usuario ha inhibido la recepción de mensajes</p> <p>r: 1 = Recepción de mensajes terminada: error en parámetros de entrada o falta condición inicial o final</p> <p>e: 1 = Carácter final recibido</p> <p>t: 1 = Recepción de mensajes terminada: ha transcurrido la temporización</p> <p>c: 1 = Recepción de mensajes terminada: se ha excedido el número máximo de caracteres</p> <p>p: 1 = Recepción de mensajes terminada debido a un error de paridad</p>



Modo Freeport (XMT y RCV)

■ Recibir datos mediante interrupciones de caracteres

- controlado por interrupciones de caracteres.
- Cada carácter recibido genera una interrupción. El carácter recibido se deposita en SMB2 y el estado de la paridad (si se ha habilitado) se deposita en SM3.0.
- Ello sucede inmediatamente antes de ejecutarse la rutina de interrupción asociada al evento Recibir carácter.

Puerto 0	Puerto 1	Descripción
SMB90 SMB91	SMB190 SMB191	Tiempo de línea de inactividad en milisegundos. El primer carácter recibido una vez transcurrido el tiempo de línea de inactividad es el comienzo del nuevo mensaje. SM90 (o SM190) es el byte más significativo y SM91 (o SM191) es el byte menos significativo.
SMB92 SMB93	SMB192 SMB193	Vigilancia de tiempo del temporizador entre caracteres/mensajes en milisegundos. Si se excede el tiempo, se finaliza la recepción de mensajes. SM92 (o SM192) es el byte más significativo y SM93 (o SM193) es el byte menos significativo.
SMB94	SMB194	Número máximo de caracteres a recibir (1 a 255 bytes). Nota: Este margen debe ajustarse al tamaño máximo esperado para el búfer, incluso si no se utiliza la finalización de mensajes por el contaje de caracteres.

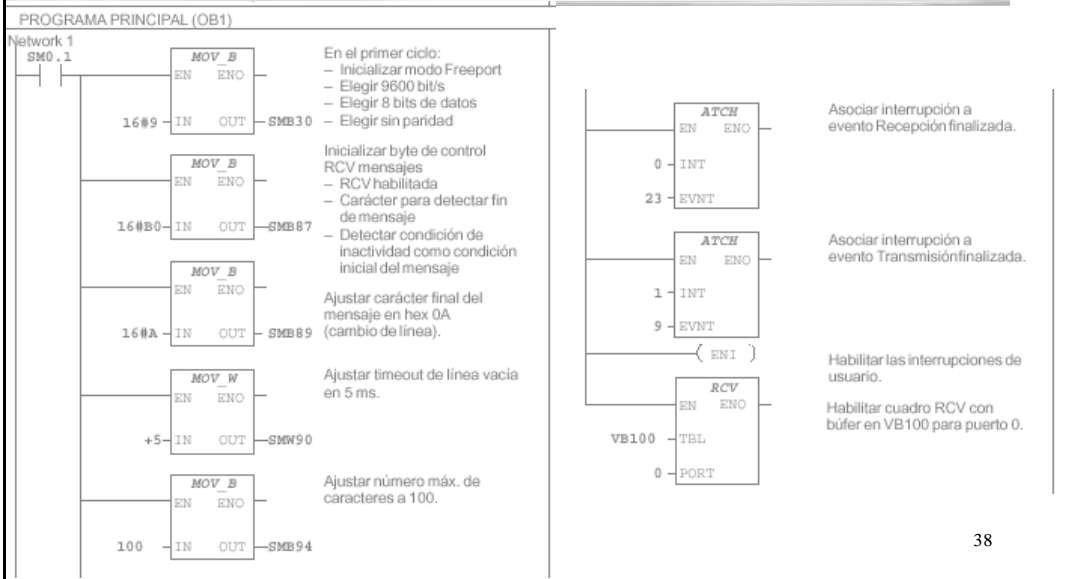


Modo Freeport (XMT y RCV)

Puerto 0	Puerto 1	Descripción
SMB87	SMB187	<div style="text-align: center;"> </div> <p>Byte de control de recepción de mensajes</p> <p>n: 0 = Inhabilita la función de recibir mensajes. 1 = Habilitada la función de recibir mensajes. El bit para habilitar/inhabilitar la recepción de mensajes se comprueba cada vez que se ejecuta la operación RCV.</p> <p>x: 0 = Ignorar SMB88 o SMB188. 1 = Utilizar el valor de SMB88 o de SMB188 para detectar el comienzo del mensaje.</p> <p>y: 0 = Ignorar SMB89 o SMB189. 1 = Utilizar el valor de SMB89 o de SMB189 para detectar el fin del mensaje.</p> <p>z: 0 = Ignorar SMW90 o SMB190. 1 = Utilizar el valor de SMW90 para detectar una condición de inactividad.</p> <p>m: 0 = Utilizar el temporizador como temporizador entre caracteres. 1 = Utilizar el temporizador como temporizador de mensajes.</p> <p>t: 0 = Ignorar SMW92 o SMW192. 1 = Finalizar la recepción si se excede el periodo de tiempo indicado en SMW92 o SMW192.</p> <p>Las marcas del byte de control de interrupción de mensajes se utilizan para definir los criterios con los que se identifica el mensaje. Se definen los criterios tanto de comienzo como de final del mensaje. Para determinar el comienzo de un mensaje, uno de los dos juegos de criterios de comienzo de mensaje combinados lógicamente mediante Y deberán ser verdaderos y deberán ocurrir en secuencia (inactividad seguida de un comienzo de carácter o condición BREAK seguida de un comienzo de carácter). Para determinar el fin de un mensaje, los criterios habilitados se combinan mediante O. A continuación se indican las ecuaciones de comienzo y de final:</p>
		<p>Comienzo del mensaje = $ll * sc + bk * sc$</p> <p>Final del mensaje = $ec + tmr + \text{contaje máximo de caracteres alcanzado}$</p> <p>Programar los criterios de comienzo de mensaje para:</p> <ol style="list-style-type: none"> Detección de inactividad: $ll=1, sc=0, bk=0, SMW90>0$ Detección de carácter inicial: $ll=0, sc=1, bk=0, SMW90$ no es relevante Detección BREAK: $ll=0, sc=1, bk=1, SMW90$ no es relevante Cualquier respuesta a una petición: $ll=1, sc=0, bk=0, SMW90=0$ (El temporizador de mensajes se puede utilizar para terminar la recepción si no hay respuesta). BREAK y carácter inicial: $ll=0, sc=1, bk=1, SMW90$ no es relevante Inactividad y carácter inicial: $ll=1, sc=1, bk=0, SMW90 > 0$ Inactividad y carácter inicial (no válido): $ll=1, sc=1, bk=0, SMW90=0$ <p>Nota: La recepción se finalizará automáticamente si ocurre un error de desbordamiento o de paridad (si se han habilitado).</p>
SMB88	SMB188	Carácter de comienzo del mensaje.
SMB89	SMB189	Carácter de fin del mensaje.



Ejemplo



Ejemplo

