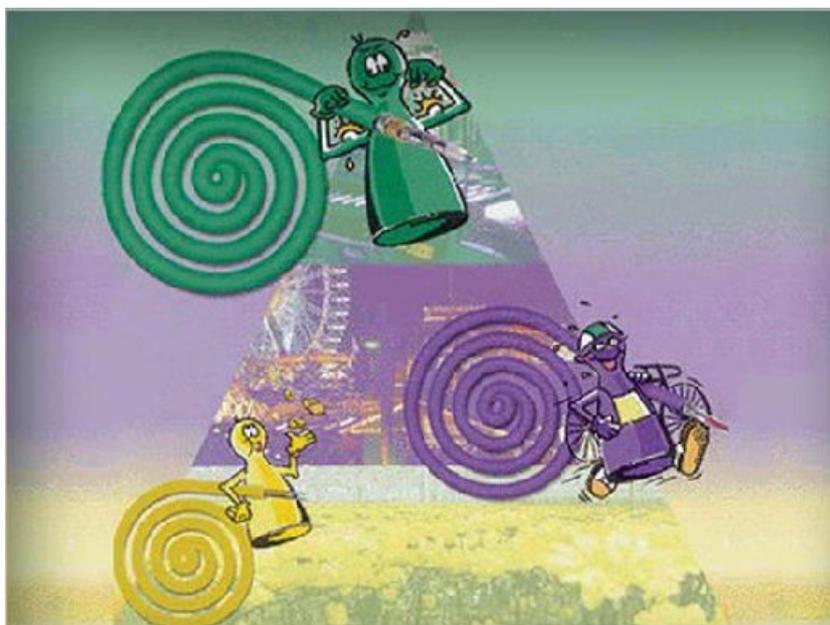


TUTORIAL DE PROFIBUS EN SIMATIC S7



Tutorial de PROFIBUS en Simatic S7

INDICE

1. Redes de comunicación.....	3
1.1. Conceptos fundamentales	3
1.2. Clasificación de las redes	4
2. PROFIBUS	6
2.1. Características generales	6
2.2. Paso de Testigo (Token Bus)	6
2.3. Maestro-Esclavo	7
3. PROFIBUS DP	8
3.1. Periferia distribuida a través del puerto integrado de la CPU	9
3.2. Coherencia de datos.....	13
3.3. Comandos SYNC y FREEZE	17
3.4. Periferia distribuida a través de una CP	20
3.5. Inteligencia distribuida entre CPU's.....	25
3.6. Routing	30
3.7. PG BUS	36
3.8. Configurar los parámetros de la red PROFIBUS.....	37
3.9. Opciones de la red PROFIBUS	39
4. PROFIBUS FDL (SEND/RECEIVE)	41
5. Protocolo S7.....	48
5.1. Comunicación a través de red MPI (Enlaces no configurados).....	48
5.2. Enlaces configurados	51
6. Protocolo FMS.....	53

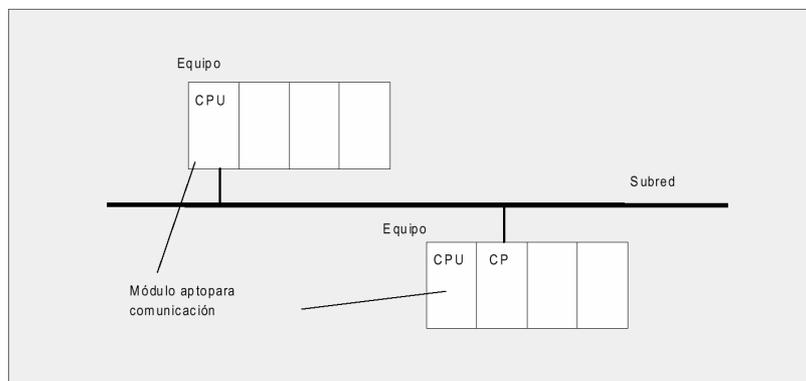
1. Redes de comunicación

1.1. Conceptos fundamentales

Comunicación :

Es la transferencia de datos entre dos interlocutores con diferentes prestaciones y el control de un interlocutor por otro, además de la consulta o interrogación del estado operativo del interlocutor. La comunicación puede establecerse mediante diferentes vías:

- A través del puerto de comunicación integrado en la CPU.
- A través de un procesador de comunicaciones (CP) aparte
- A través del bus de fondo (Bus-K que recorre el bastidor)



Equipo :

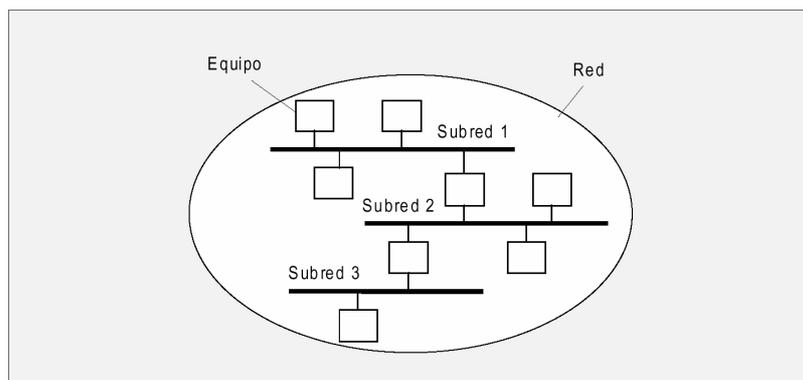
En este contexto se denomina equipo a una unidad, que puede ser un autómata programable, PG, panel operador/sistema, PC, aparato ajeno, etc., conectable a una o varias subredes.

Subred :

Es el conjunto de todos los componentes físicos necesarios para establecer una vía de transmisión de datos, así como el respectivo procedimiento común para el intercambio de datos. Ejemplos de subredes: MPI, PROFIBUS, Industrial Ethernet.

Red :

Conjunto formado por una o varias subredes iguales o diferentes interconectadas.

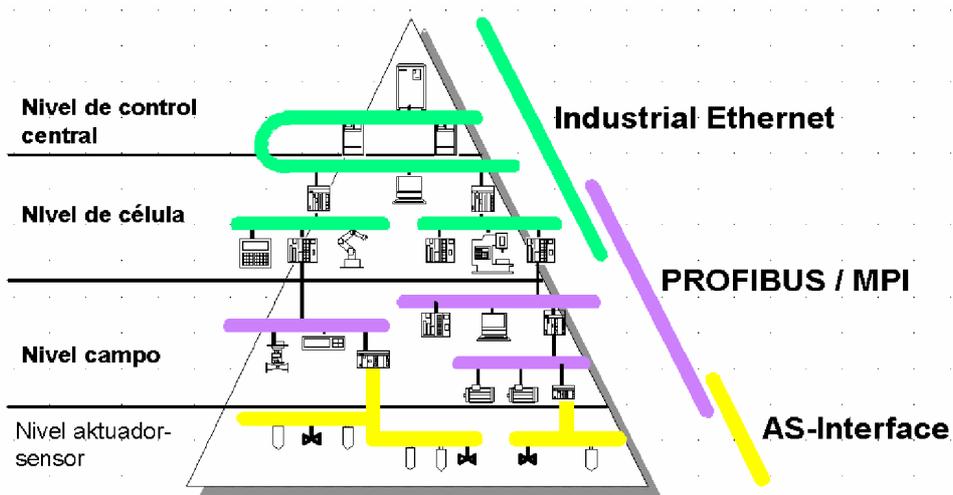


1.2. Clasificación de las redes

Para adaptarse a los diferentes requisitos de automatización, SIMATIC ofrece distintas redes de comunicación:

- 1) Industrial Ethernet
- 2) PROFIBUS / MPI
- 3) Interface AS-i

Existen cuatro niveles de automatización:



Nivel de control central: En este nivel se procesan tareas de generales que conciernen a toda la empresa (funciones de gestión). Entre ellas figuran la memorización de valores del proceso y funciones de procesamiento para optimizar y como analizador, así como su presentación en forma de listados. Los datos necesarios se recolectan y procesan para toda la empresa, con independencia del lugar de emplazamiento. Desde el nivel de control central puede accederse igualmente a otros niveles.

La cantidad de estaciones puede ser superior a 1.000.

Nivel de célula

En el nivel de célula se procesan autónomamente todas las tareas de automatización y optimización. En este nivel están interconectados los autómatas, PCs y los equipos para el funcionamiento y la observación.

Nivel de campo

El nivel de campo es el nexo entre las instalaciones y los autómatas programables. Los dispositivos de campo miden, señalizan y retransmiten a las instalaciones las órdenes recibidas del nivel de célula. En general se transmiten pequeñas cantidades de datos. En este caso es típica una comunicación jerarquizada, es decir varios dispositivos de campo se comunican con un maestro.

Nivel de actuadores-sensores

En este nivel, un maestro se comunica con los actuadores y sensores conectados a su subred. Son característicos aquí tiempos de respuesta rápidos y un número reducido de bits de datos.

	AS-i	PROFIBUS	ETHERNET
Norma	IEC-TG-17B	EN 50170	IEEE 802-3
Velocidad de transmisión	167 Kbits/seg	9,6 Kbits/seg ÷ 12 Mb/seg	10 Mb/seg ÷ 100Mb/seg
Nº de equipos	1 Maestro 31 Esclavos	127 estaciones (32 activas)	1024 estaciones
Medio físico	Cable 2 hilos (Cable amarillo)	2 hilos apantallado, o fibra óptica de vidrio o plástico (Cable morado)	Cable coaxial o par trenzado industrial (ITP) o fibra óptica de vidrio (Cable verde)
Extensión de la red	100m (300 m con repetidores)	10 km apróx. (medio eléctrico) 100 km (fibra óptica)	Depende de muchos factores
Método de acceso al bus (Protocolo)	Maestro/Esclavo	Profibus DP: Maestro/Esclavo FDL: Paso de testigo	CSMA/CD

2. PROFIBUS

2.1. Características generales

PROFIBUS es la red para los niveles de célula y campo. Se utiliza para transmitir cantidades de datos desde pequeñas hasta medias. Físicamente, PROFIBUS es una red eléctrica que puede ser:

- Cable a dos hilos apantallado,
- Red de fibra óptica,
- Red de transmisión inalámbrica mediante infrarrojos.

Velocidad de la red: Desde 9,6 Kbit/s a 12 Mbit/s

Podemos conectar a la red un máximo de 127 estaciones, y de éstas no puede haber más de 32 estaciones activas.

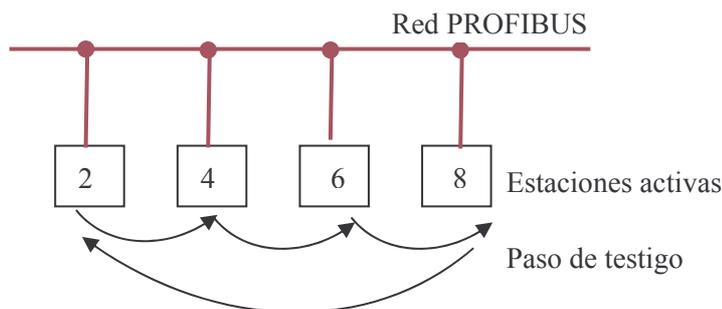
Para una red PROFIBUS dispongo de varios servicios de comunicación:

- Comunicación con PG/OP
- Protocolo S7
- Comunicación con equipos S5 (FDL)
- Comunicación estándar (FMS)
- DP Periferia descentralizada

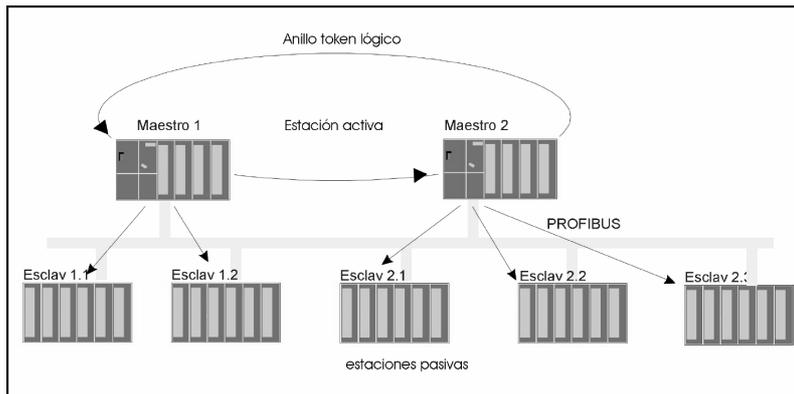
Hay dos métodos de acceso a la red, según queramos acceder a estaciones activas (método de paso de testigo), o a estaciones pasivas (método maestro-esclavo).

2.2. Paso de Testigo (Token Bus)

Las estaciones activas del bus conectadas a PROFIBUS constituyen un anillo lógico de paso de testigo en orden numérico ascendente según su dirección de PROFIBUS. Este orden lógico es independiente de la disposición física de las estaciones. Por anillo paso de testigo se entiende una organización de estaciones (maestros) en la que se pasa un testigo siempre de una estación a la próxima.

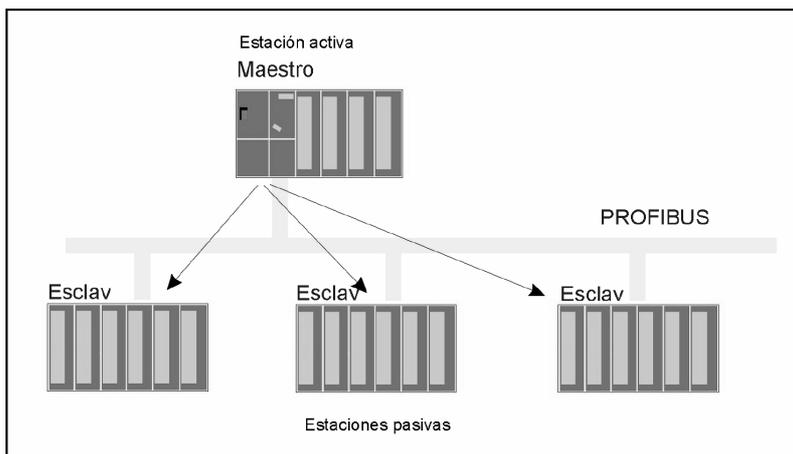


El testigo confiere el derecho a transmitir información por la red. Dicho testigo se retransmite entre las estaciones del bus activas a través de un telegrama de testigo especial. Cuando una estación tiene el testigo, puede enviar telegramas mientras dure el denominado tiempo de retención del testigo (configuración). Una vez transcurrido éste, la estación ya sólo puede enviar un mensaje de alta prioridad. Si una estación no tiene ningún mensaje que enviar, pasa el testigo directamente a la siguiente estación en el anillo lógico. De esto se exceptúa la estación activa con la dirección más alta existente en el bus, también denominada HSA (Highest Station Address); esta estación pasa el testigo exclusivamente a la estación del bus activa con la mínima dirección de bus, para que se cierre nuevamente el anillo de paso de testigo lógico. Este método de acceso es independiente del tipo de red física PROFIBUS utilizada.



2.3. Maestro-Eslavo

Si un anillo lógico consta de una sola estación activa y en el bus hay varias estaciones pasivas, esto es un sistema Maestro-Eslavo.



El método maestro-esclavo permite al maestro (estación activa) que posee entonces el derecho de emisión (tiene el testigo) tener acceso a los esclavos (estaciones pasivas) que tiene asignados. Entonces, el maestro tiene la posibilidad de enviar mensajes a los esclavos o de recibir los procedentes de éstos.

La configuración de PROFIBUS-DP estándar se basa en este método de acceso al bus. Una estación activa (maestro DP) intercambia datos de forma cíclica con las estaciones pasivas (esclavos DP).

3. PROFIBUS DP

PROFIBUS-DP cumple los elevados requisitos de tiempo que se imponen para el intercambio de datos en el sector de la periferia descentralizada y los dispositivos de campo. La configuración DP típica tiene una estructura con un único maestro. La comunicación entre el maestro DP y el esclavo DP se efectúa según el principio maestro-esclavo. El maestro dirige todo el tráfico de datos en el bus, esto significa que los esclavos DP sólo pueden actuar en el bus tras solicitarlo el maestro. A tal efecto, los esclavos DP son activados sucesivamente por el maestro conforme a una lista de llamadas (lista de sondeo). Entre el maestro DP y el esclavo DP se intercambian los datos útiles continuamente (de forma cíclica), sin tener en consideración su contenido. A la periferia (la unidad ET 200) conectada a PROFIBUS como esclavo DP se accede como a cualquier otra unidad periférica situada en el módulo central o de ampliación. Es decir, es posible acceder a los módulos periféricos directamente mediante instrucciones o durante la actualización de la imagen del proceso.

Se puede hacer de 2 formas:

- 1.- A través de una CPU con puerto integrado.
- 2.- Mediante tarjeta de comunicaciones CP o un módulo interface IM.

Velocidad: 12MBd, o si los esclavos no lo permiten, 1,5 MBd.

A un maestro DP (p.ej. CPU) es posible conectar un máx. de 125 estaciones esclavas PROFIBUS DP (según el CPU utilizado).

Requisitos hardware:

S7-200 : Sólo puede ser esclavo de una red DP.

S7-300 :

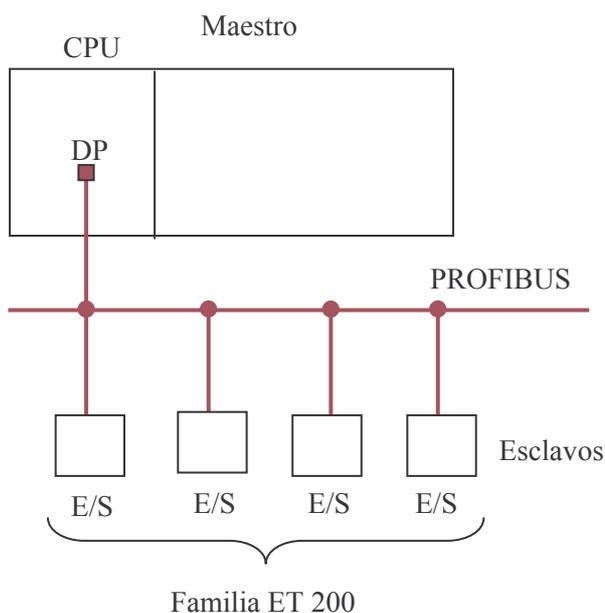
- CPU: 31X-2DP. Puertos: 1 MPI y 1 DP integrado. Por defecto está configurado como maestro, pero puede ser esclavo. Excepto: 318-2DP que sólo puede ser maestro.
- CP 342-5.

S7-400:

- CPU: 41X-DP (Sólo maestro). Si fuese 3DP: 1 puerto MPI, 1 puerto DP, 1 puerto configurable como MPI/DP.
- IM467: Módulo interface equivalente a un puerto DP. Sólo maestro.
- CP 443-5 Extended. Sólo Maestro

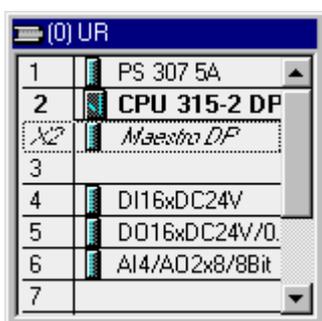
3.1. Periferia distribuida a través del puerto integrado de la CPU

En esta configuración, se intercambian los datos entre el maestro DP y esclavos DP sencillos (módulos de E/S), a través del maestro DP. El maestro DP explora sucesivamente cada esclavo DP configurado en su lista de llamadas (lista de sondeo) dentro del sistema maestro DP, transmitiendo los datos de salida o recibiendo de vuelta sus valores de entrada. Las direcciones E/S son asignadas automáticamente por el sistema de configuración. Esta configuración se denomina también sistema monomaestro, porque aquí hay conectados un solo maestro DP con sus respectivos esclavos DP a una red PROFIBUS-DP.

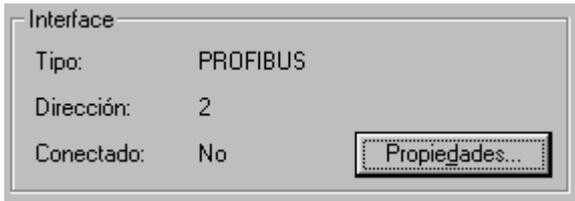


Cuando configuremos la red de este modo, las entradas y salidas de las ETs las verá el maestro como propias, accediendo a ellas con las instrucciones normales de S7.

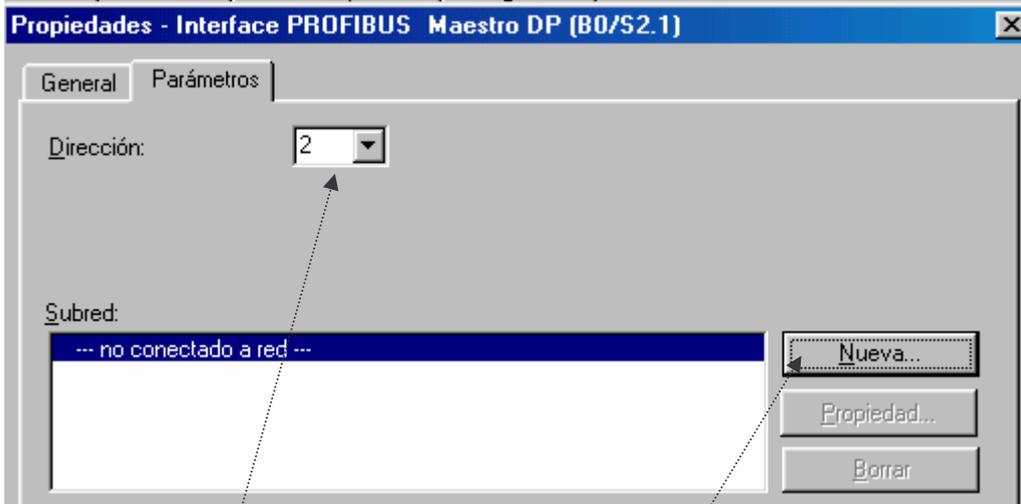
- 1.- Configuramos el HW de nuestro equipo que será el maestro DP.
- 2.- Configuramos el sistema maestro DP :



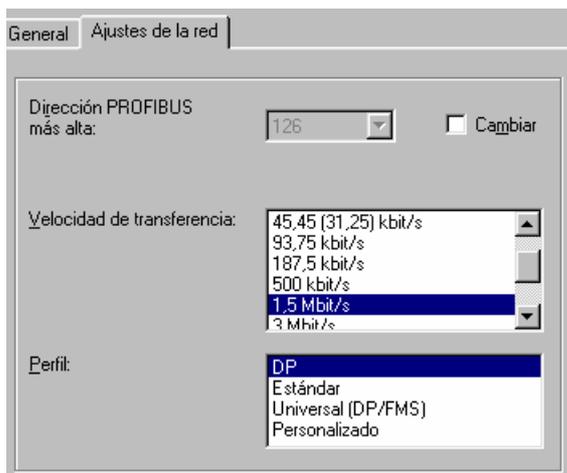
Configuramos las propiedades del maestro DP (doble clic sobre "Maestro DP").



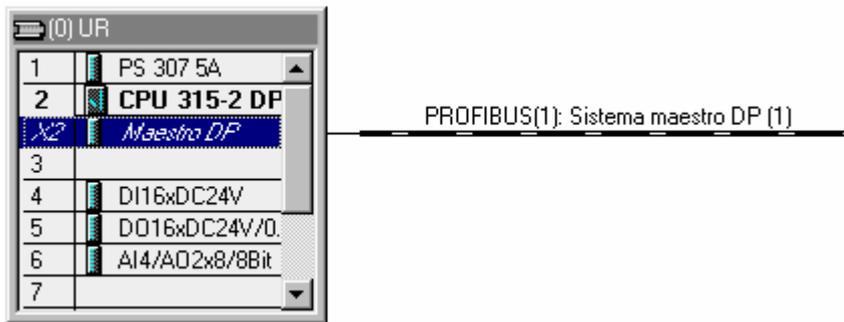
En la pestaña “General” configurar la red. En Interface pulsar sobre botón “Propiedades”.



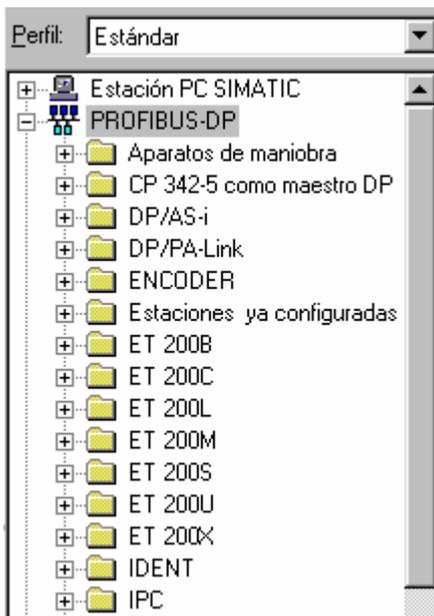
Asignarle una dirección de PROFIBUS y crear una red nueva. En la pestaña “Ajustes de red” de la nueva red, ajustar la velocidad de transferencia y seleccionar como perfil el DP.



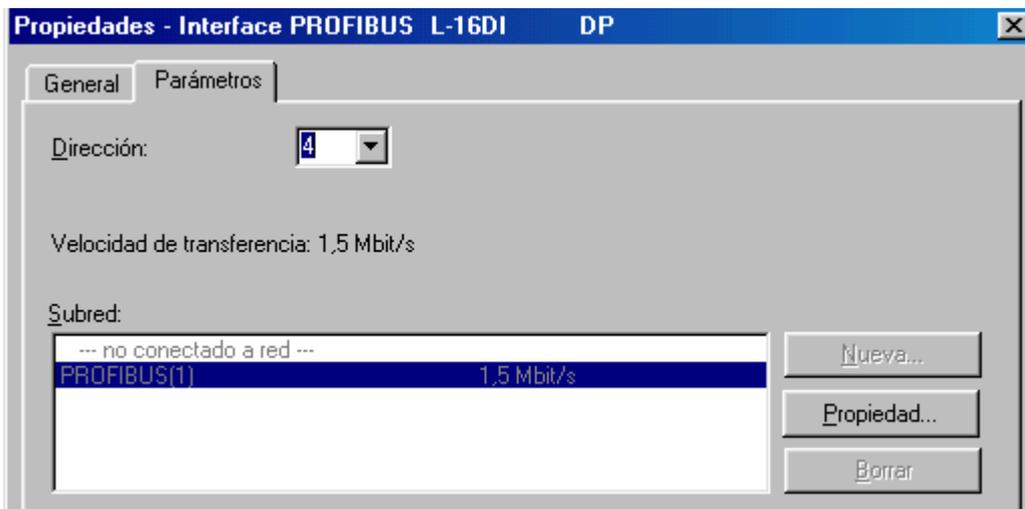
Ya tendremos nuestro sistema maestro DP del que colgarán los esclavos:



3. Para insertar la ET a la red, la seleccionamos en el catálogo de hardware:



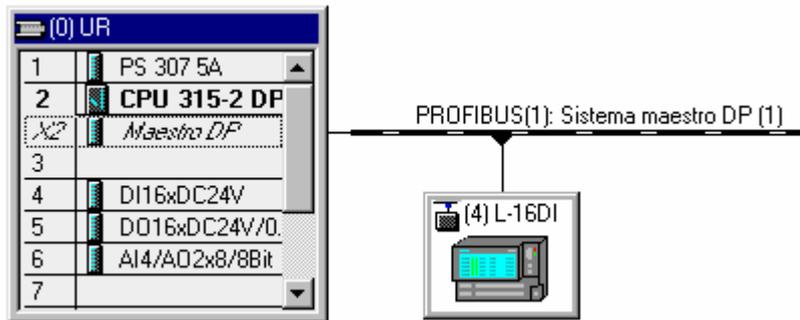
Seleccionamos el Sist. Maestro y hacemos doble clic sobre la ET, o simplemente arrastrar la ET hasta el Sist. Maestro. Entonces habrá que indicarle la dirección PROFIBUS.



Esa dirección será la que le habré puesto a la ET mecánicamente. Atención: Cuando cambiamos la dirección mecánicamente en la ET, y ya esta alimentada, no la detecta, hay que realimentarla: quitarle y ponerle alimentación de nuevo.

La ET se alimenta externamente.

Por fin tendremos nuestra red configurada:



“Guardamos y compilamos” y “Cargamos en modulo”

A partir de ahora las entradas y salidas de la ET formaran parte de la PAE y PAA del maestro. Podemos cambiar el direccionamiento de la ET haciendo doble clic en la vista detallada del HW:

Slot	Módulo / Id...	Referencia	Dirección E	Dirección S	Comentario
0	PS 307 5A	Módulo universal		0	
1	CPU 315-2 DP	Módulo universal	2		

Si meto una dirección mayor que 128, no podré acceder a esas entradas/salidas como parte de la PAE o PAA sino como periferia.

Ejemplo de programa:

```

U  E2.0      //Consulta una de las entradas de la ET
=  A4.0      //y asigno a la salida de mi maestro

U  E0.0      //Consulta una entrada de mi maestro
=  A0.0      //y asigno a una salida de la ET
    
```

3.2. Tipos de ET's

Hay varios tipos de ET's:

- ET 200M
- ET 200S
- ET 200X
- ET 200L
- ET 200B
- ET 200C
- ET 200U

ET 200M :

Coherencia de datos

No hay problema si queremos enviar o recibir paquetes de datos a través de PROFIBUS tamaño byte, palabra o doble palabra. Podemos utilizar las instrucciones de carga (L) y transferencia (T). El problema es si queremos enviar tres bytes o más de 4 bytes a un esclavo que precisa de ellos en un único paquete de datos. El PROFIBUS DP tiene su propio ciclo de bus en el que lee de la periferia, si le mandamos ahora unos bytes y luego otros, no se los mandamos sincronizados.

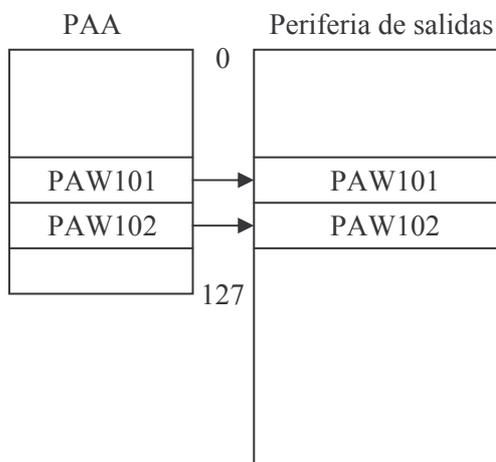
Hay dos tipos de coherencia:

- Unidad: Una unidad a nivel de byte o palabra.
- Total: Paquete de datos direccionado en su totalidad.

Una solución es meter estos datos en la PAA (bytes 0 a 127). La zona de PAA se escribe en la periferia de salida al final de cada ciclo, donde puede ser leída en su totalidad por PROFIBUS.

Ejemplo: Para mandar 8 bytes:

```
L PAW100  
L PAW102
```



Otra solución es utilizar las funciones SFC14 y SFC15 para leer y escribir datos con coherencia.

SFC 14 "DPRD_DAT" : Permite leer datos coherentes de un esclavo DP. Su longitud tiene que ser de tres o más de cuatro bytes; la longitud máxima depende de la CPU.

```
CALL SFC14  
LADDR:=  
RET_VAL:=  
RECORD:=
```

Parámetros de entrada:

- LADDR (WORD): Dirección inicial configurada desde el área de E del módulo esclavo, de donde debe ser leída. Debe tener la misma longitud que la que se ha configurado con STEP 7 para el módulo seleccionado. La dirección se debe indicar en valores hexadecimales, p. ej., para la dirección inicial 100 es: LADDR:=W#16#64.

Parámetros de salida:

- RET_VAL (INT): Si durante la ejecución de la función ocurre un error, este valor de retorno contiene un código de error. Se permite E, A, M, D, L.
- RECORD (ANY): Área de destino para los datos leídos. Se permite E, A, M, D, L. El tipo de datos ANY es un área de memoria indicada con un puntero que es la dirección inicial de BYTE más la longitud en número de BYTES.

Ej.: P#DB1.DBX0.0 BYTE 10 → Señala el área de datos de DB1 (DB0...DB9)
P#A0.0 BYTE 5 → Apunta al área de memoria de salidas AB0...AB4

Por ejemplo, si tengo una ET de 64 salidas digitales con dirección inicial 200:

```
CALL SFC14
  LADDR:=W#16#C8 //Dirección inicial de las salidas de la ET (en la PAE o PEW)
  RET_VAL:=MW100
  RECORD:=P#DB1.DBX0.0 BYTE 8 //Guardo esas salidas en un DB.
```

SFC 15 "DPWR_DAT" : Permite transferir los datos a RECORD de forma coherente con respecto al esclavo DP.

```
CALL SFC15
  LADDR:=
  RECORD:=
  RET_VAL:=
```

Parámetros de entrada:

- LADDR (WORD): Dirección inicial configurada desde el área PAA del módulo en el que debe escribirse. La dirección se debe indicar en valores hexadecimales, p. ej., para la dirección inicial 100 es: LADDR:=W#16#64.
- RECORD (ANY): Área de donde recojo los datos útiles a escribir. Debe tener la misma longitud que la que se ha configurado con STEP 7 para el esclavo seleccionado. Sólo se admite el tipo de datos BYTE.

Parámetros de salida:

- RET_VAL (INT): Si durante la ejecución de la función ocurre un error, este valor de retorno contiene un código de error. Se permite E, A, M, D, L.

Por ejemplo, si tengo una ET de 64 entradas digitales con dirección inicial 200:

```
CALL SFC15
  LADDR:=W#16#C8 //Dirección inicial de las entradas de la ET (en la PAA o PAW)
  RECORD:=P#DB1.DBX0.0 BYTE 8 //De donde cojo los datos para
  RET_VAL:=MW100 //enviar a la ET
```

3.3. Comandos SYNC y FREEZE

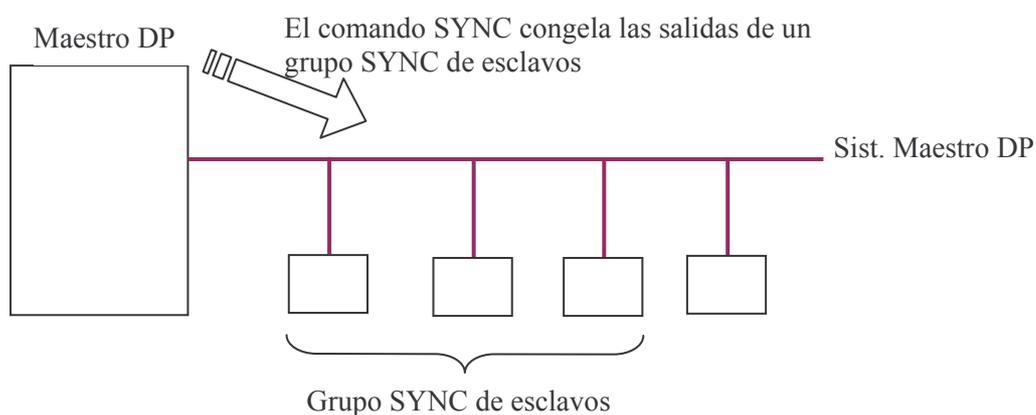
SYNC

Normalmente, el maestro DP transfiere los datos de salida cíclicamente (dentro del ciclo del bus PROFIBUS DP) a las salidas de los esclavos DP.

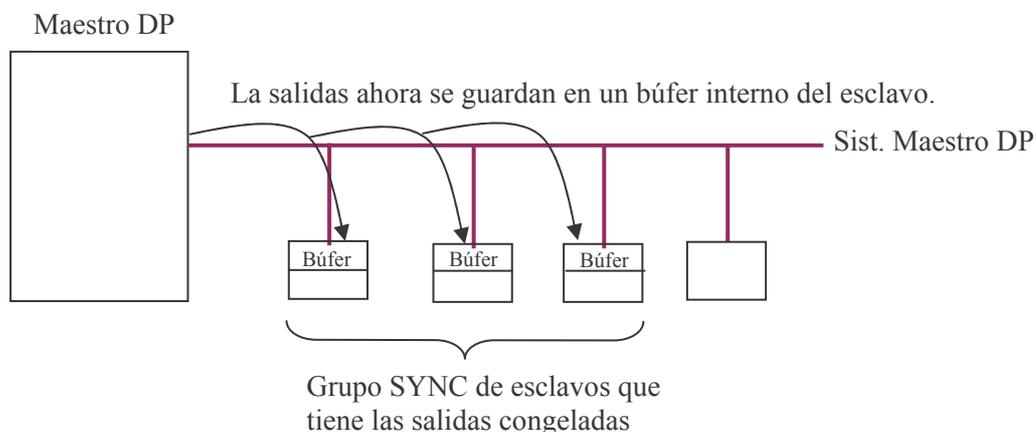
Cuando se quiera que determinados datos de salida (que puedan estar repartidos entre varios esclavos) sean emitidos al proceso exactamente en el mismo instante, se debe enviar el comando de control SYNC al correspondiente maestro DP aplicando la función SFC 11 "DPSYC_FR". Un ejemplo de esto podría ser arrancar una serie de motores en el mismo instante.

Funcionamiento de SYNC:

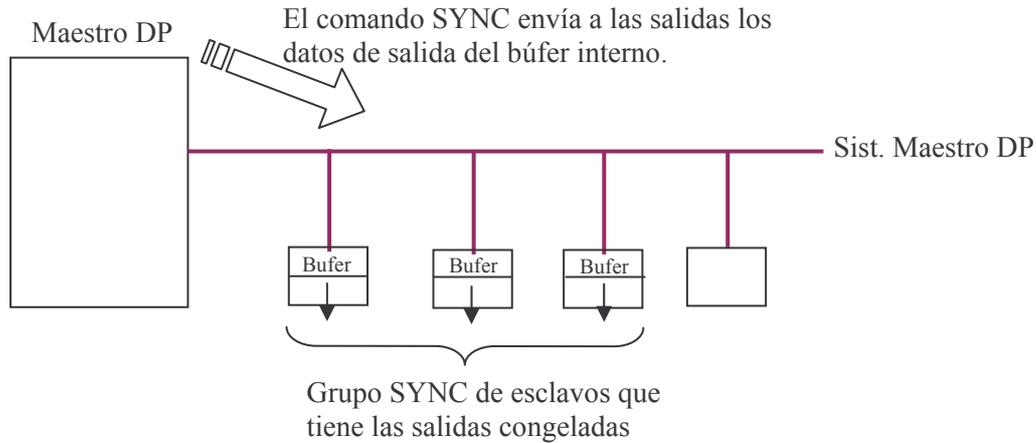
Con el comando de control SYNC los esclavos DP de los grupos especificados cambian al modo SYNC, esto es, el maestro DP transfiere los datos de salida actuales y hace que los esclavos DP afectados congelen las salidas.



En los sucesivos telegramas de respuesta, los esclavos DP guardan los datos de salida en un búfer interno, de tal modo que los valores de las salidas no cambian.



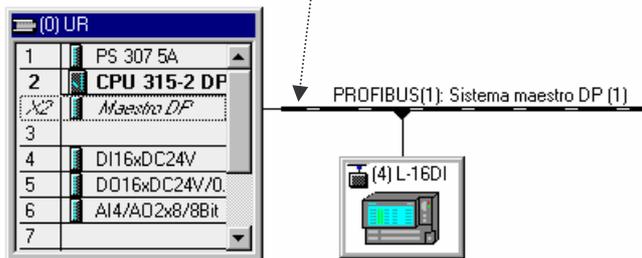
Los esclavos DP de los grupos seleccionados depositan los datos de salida de su búfer interno en las salidas del proceso cada vez que se da el comando SYNC.



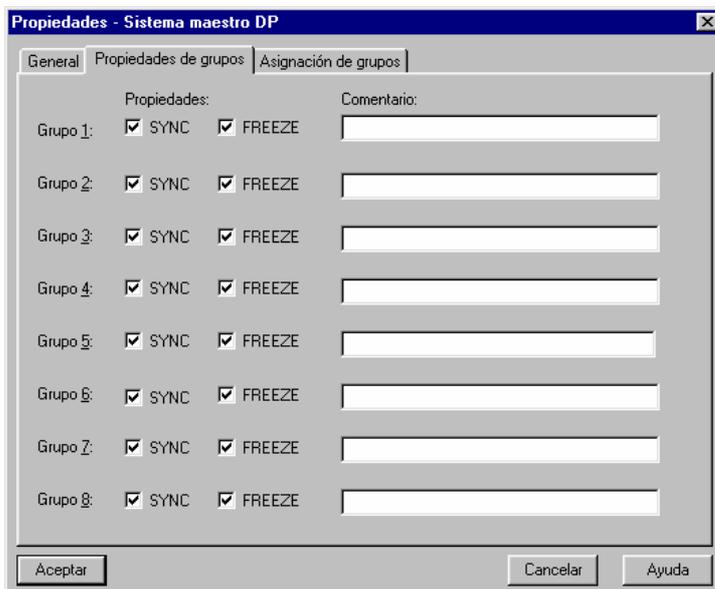
Para que las salidas se vuelvan a actualizar cíclicamente es necesario dar el comando UNSYNC con la SFC 11 "DPSYC_FR".

Procedimiento:

1.- Hay asignar los esclavos DP a grupos SYNC y FREEZE. En HW Config hago doble clic sobre el sistema maestro DP, o botón derecho y selecciono "Propiedades del objeto...".



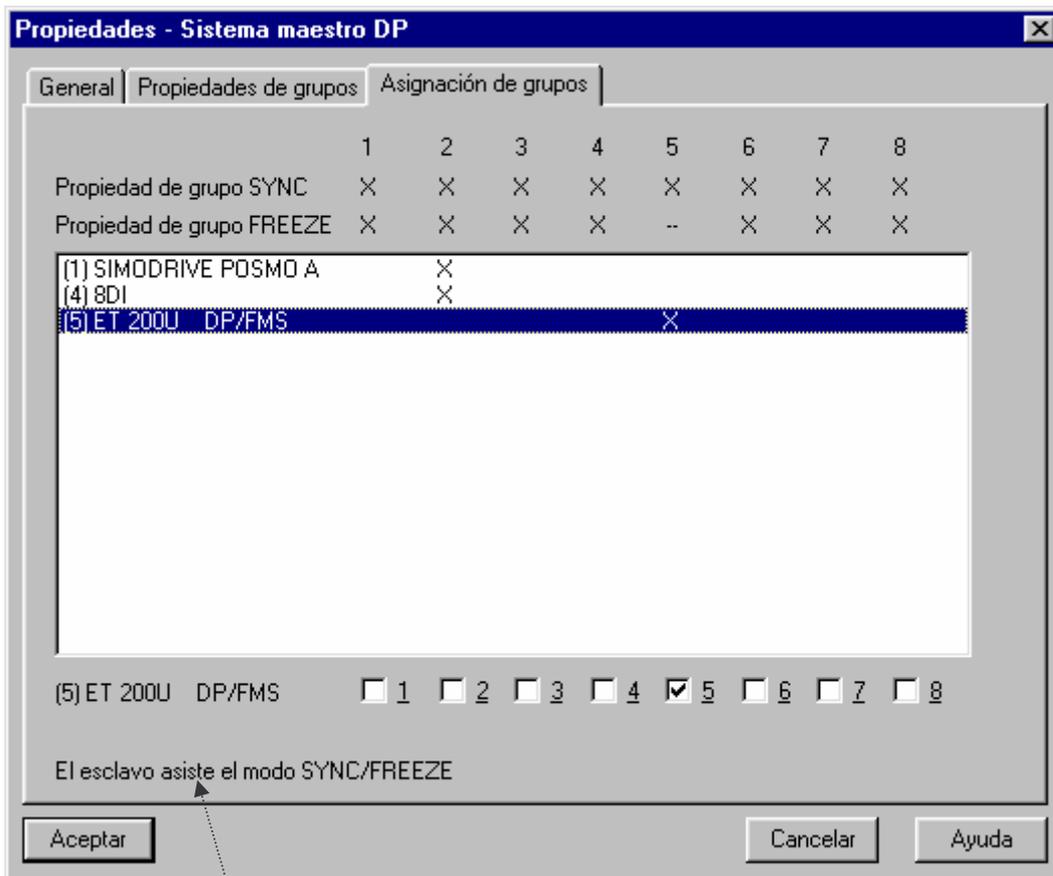
Por cada sistema maestro se pueden formar como máximo 8 grupos. Sin embargo, a cada esclavo DP se le puede asignar solamente un grupo SYNC y un grupo FREEZE.



Activar SYNC y/o FREEZE para el grupo haciendo clic en la casilla de verificación correspondiente.

2.- Seleccione la ficha "Asignación de grupos" donde asignaré los esclavos DP a grupos.

En el siguiente ejemplo tenemos 3 esclavos. Los dos primeros están asignados al grupo 2; el grupo 2 tiene propiedad SYNC y FREEZE. El tercer esclavo pertenece al grupo 5; el grupo 5 tiene propiedad SYNC, pero no FREEZE.

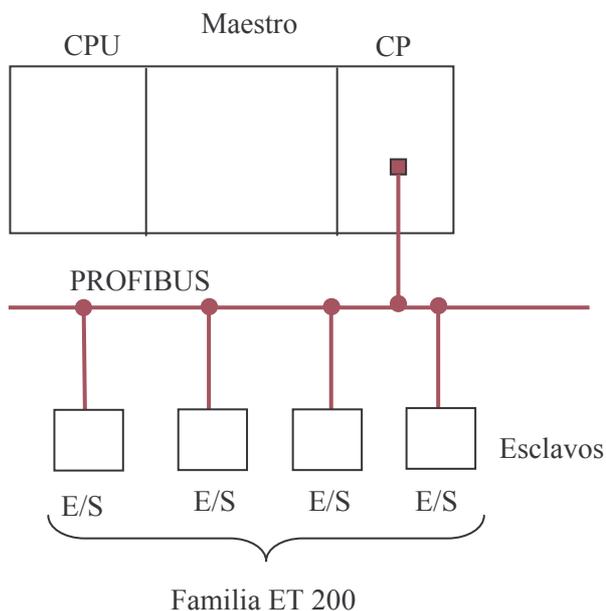


Se puede saber si el esclavo admite los comandos SYNC y FREEZE, pinchando sobre él y leyendo la última línea.

Comando SYNC/FREEZE:

```
CALL "DPSYC_FR"
REQ      :=
LADDR   :=
GROUP   :=
MODE    :=
RET_VAL :=
BUSY    :=
```

3.4. Periferia distribuida a través de una CP



Ahora las E/S de las ET estarán almacenadas en un búfer de entradas o salidas que se encuentra en la CP. El maestro para acceder a ellas tendrá que hacerlo mediante dos funciones: FC1(DP_SEND) y FC2 (DP_RECV).

1.-Configurar el hardware del maestro.



2.- Configuramos un sistema maestro DP en la CP:

Configuramos las propiedades de la CP (doble clic sobre la CP).

- En la pestaña general configurar la red "PROFIBUS".
- Puedo cambiar el direccionamiento de las entradas/salidas de la CP en la pestaña "Direcciones".

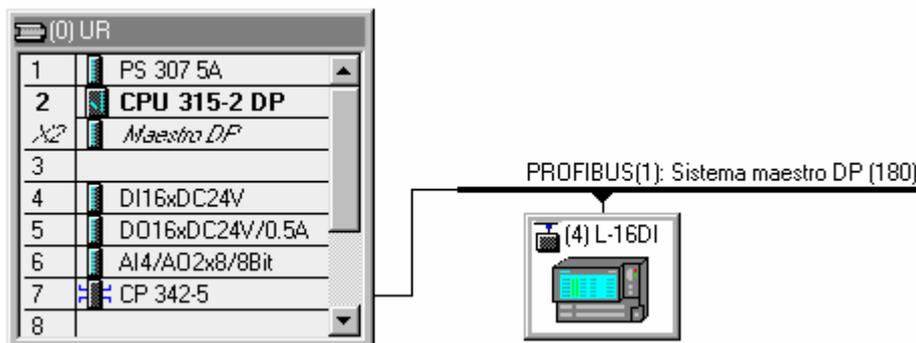
Entradas			
Inicio:	<input type="text" value="304"/>	Long.: 16	<input type="checkbox"/> Estándar
Salidas			
Inicio:	<input type="text" value="304"/>	Long.: 16	<input type="checkbox"/> Estándar

Hay un buffer de entradas en la CP, donde se depositan los datos de la periferia. Este buffer se puede leer mediante la función “DP_RECV”.

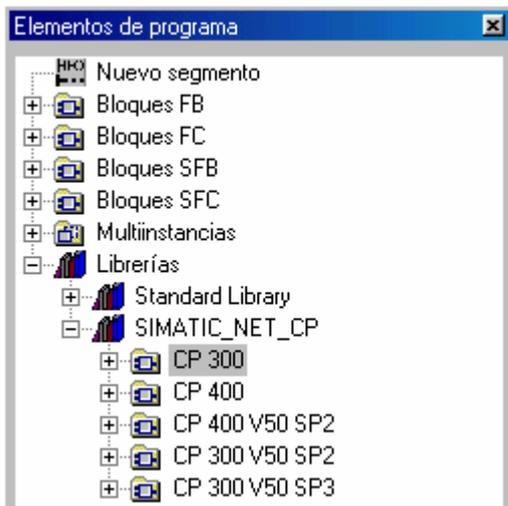
Hay un buffer de salidas en la CP, donde se depositan los datos que a enviado la CPU hacia la periferia. Este buffer se puede escribir mediante la función “DP_SEND”.

En la pestaña “Modo de operación” activar la CP como “Maestro DP”

3. Insertar la ET al sistema maestro DP :



4. El acceso a las entradas salidas se hace mediante dos funciones que se encuentran en el catalogo de “Elementos de programa”: Librerías → SIMATIC_NET_CP



FC1 "DP_SEND" : Transfiere datos (E, M, DB) a la CP, o sea, transfiere los datos de un área de salidas DP determinada al CP PROFIBUS para emitirlo a la periferia descentralizada.

```
CALL "DP_SEND"
  CPLADDR :=
  SEND    :=
  DONE    :=
  ERROR   :=
  STATUS  :=
```

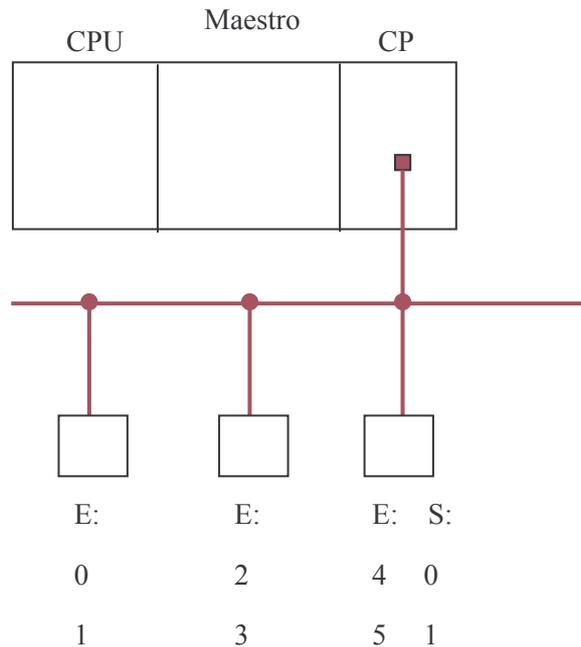
Parámetros de entrada:

- CPLADDR (WORD) :Dirección inicial de la CP. Al configurar el CP con la herramienta de configuración STEP7 se visualiza la dirección inicial del módulo en la tabla de configuración. Introduzca aquí esta dirección en hexadecimal. Ej: W#16#130
- SEND (ANY): Zona de memoria (A, M, DB) de la CPU a transferir a la CP. Se indica la dirección inicial y la longitud.

Ej.: P#E0.0 byte 6

6 bytes será la cantidad de entradas totales en los esclavos.

Advertencia: La longitud de lo que envío al búfer de salida de la CP, debe coincidir con el total de entradas que exista en la periferia. Ejemplo:



En este ejemplo debería enviar 6 bytes.

Parámetros de salida:

- DONE (BOOL) : Si esta a 1, indica que se han aceptado nuevos datos.
- ERROR (BOOL): Si esta a 1, se ha producido error. El código de error se indica en el parámetro STATUS.
- STATUS (WORD): Código de estado.

FC1 "DP_RECV" : Recibe datos a través del CP PROFIBUS, o sea, recibe los datos del proceso de la periferia descentralizada al igual que una información de estado en un área de entrada DP determinada.

```
CALL "DP_RECV"
  CPLADDR :=
  RECV    :=
  NDR     :=
  ERROR   :=
  STATUS  :=
  DPSTATUS :=
```

Parámetros de entrada:

- CPLADDR (WORD) :Dirección inicial de la CP. Al configurar el CP con la herramienta de configuración “HW Config” se visualiza la dirección inicial del módulo en la tabla de configuración. Introduzca aquí esta dirección en hexadecimal. Ej: W#16#130
- RECV (ANY): Zona de memoria (A, M, DB) de la CPU donde deposito las entradas leídas en la CP. Se indica la dirección inicial y la longitud.

Ej.: P#A4.0 byte 2

2 bytes será la cantidad de salidas totales en los esclavos.

Advertencia: La longitud de lo que recibo en el búfer de entradas de la CP, debe coincidir con el total de salidas que existan en la periferia.

Parámetros de salida:

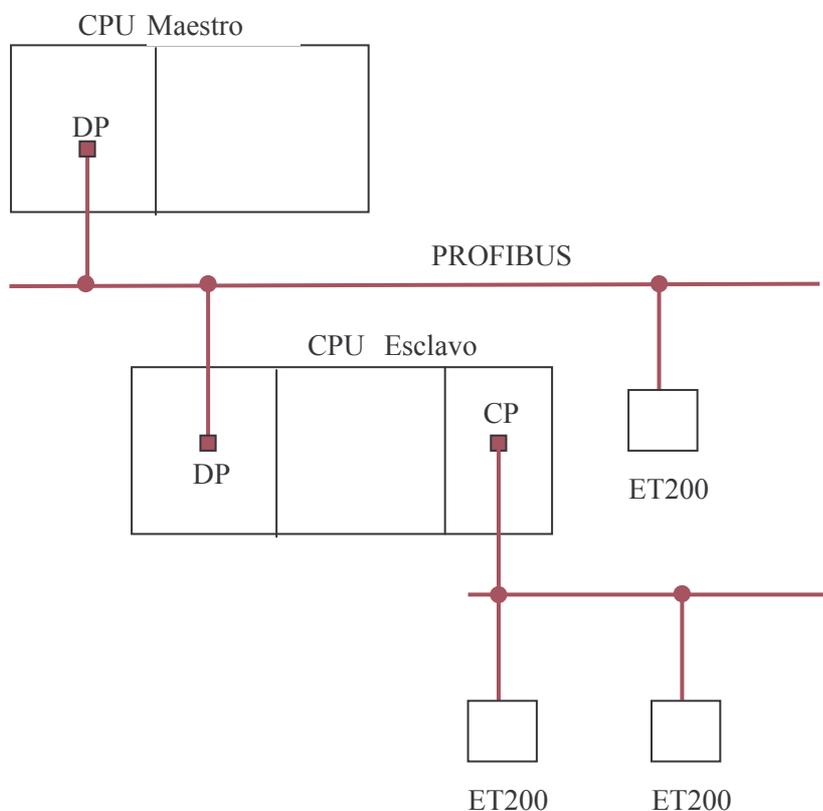
- NDR (BOOL) : Si esta a 1, indica que se han aceptado nuevos datos.
- ERROR (BOOL): Si esta a 1, se ha producido error. El código de error se indica en el parámetro STATUS.
- STATUS (WORD): Código de estado.
- DPSTATUS (BYTE): Código de estado DP.

3.5. Inteligencia distribuida entre CPU's

Las tareas de automatización pueden desglosarse en tareas parciales, que son controladas por un sistema de automatización de orden superior. Estas tareas parciales de control, ejecutables de forma autónoma y eficiente, son procesadas previamente en una CPU. Dicha CPU se puede establecer como esclavo DP inteligente.

En las configuraciones con esclavos DP inteligentes, como p.ej. una CPU 315-2DP, el maestro DP no accede a los módulos de E/S de la CPU esclava, sino sólo a la zona de operandos de la CPU esclava; es decir, que dicha zona de operandos no deberá ser ocupada para módulos E/S reales de la CPU esclava. Esta asignación debe efectuarse durante la configuración del esclavo.

A tal efecto, el maestro DP puede direccionarse dentro de un sistema monomaestro, con lo que los datos siguen intercambiándose según el principio maestro-esclavo (MS) cíclico.

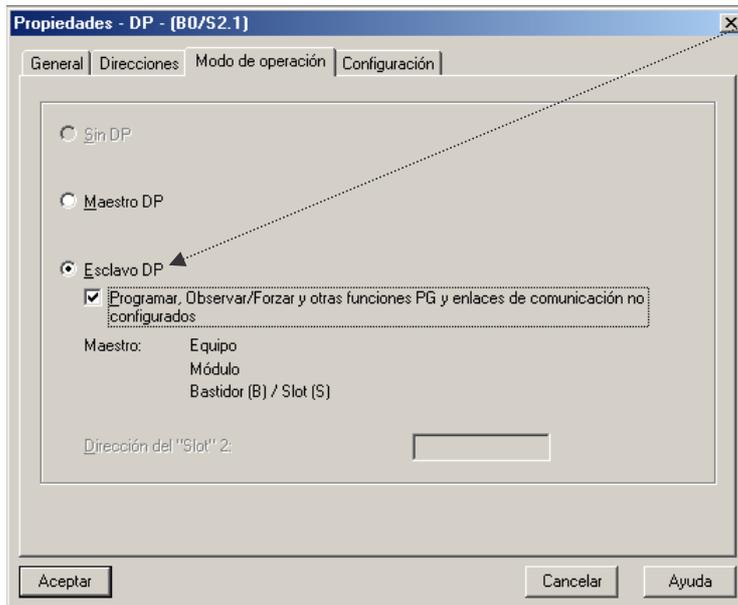


Como CPU maestro podemos tener: S7-400 o S7-300.

Como CPU esclavo: S7-300 o S7-200

Excepciones: la CPU318 y la 315-2AFF00 (es antigua) no pueden ser maestros.

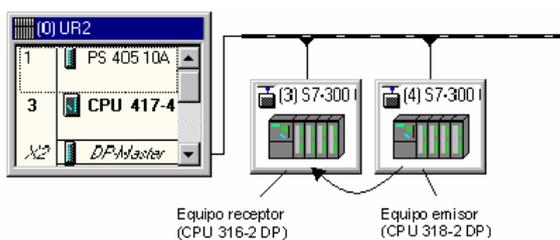
1. Configuro el HW de la CPU esclavo. Hago doble clic sobre el puerto DP integrado y lo conecto a una red PROFIBUS DP. En la pestaña “Modo de operación” establezco que la CPU debe operar como Esclavo (inteligente).



Activaré la casilla “Programar y Observar/Forzar...” si va a ejecutar funciones vía PROFIBUS-DP como programar (p. ej. cargar), test (observar/forzar) o comunicación a través de enlaces no configurados (p. ej. con SFC 65, SFC 66)

2. En el caso de un esclavo DP inteligente, el maestro DP no accede a las entradas y salidas conectadas del esclavo, sino a un área de transferencia en el espacio de direccionamiento de entradas/salidas de la CPU. Según esto deberé configurar unas áreas de E/S para el intercambio de datos entre el esclavo DP inteligente y el maestro DP asignado. Lo haré en la pestaña “Configuración”. Se elegirá el modo Maestro/Esclavo "ME".

El modo de Comunicación Directa “CD” es para el caso de comunicación entre esclavos inteligentes. Ejemplo:



Ejemplo de configuración de la tabla:

Modo		local: Esclavo			
		E/S	Dirección	Imagen del proceso	Diagnóstico
1	ME	E	100	---	1022
2	ME	S	20	OB1-PA	1022
3					

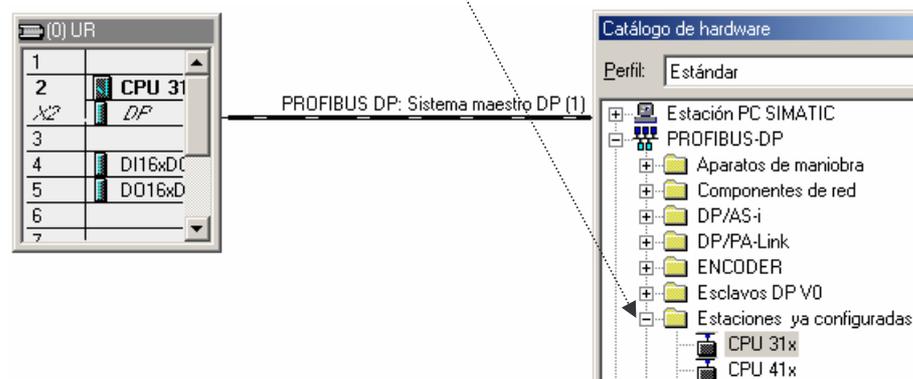
Interlocutor PROFIBUS-DP								
Dir...	E/S	Dirección	Ima...	OB de alar...	Longitud	Unidad	Coherencia	Comentario
					2	Byte	Long. total	
					1	Word	Unidad	

Se ha creado un área de entradas en la CPU Esclavo donde se recepcionarán los datos que envíe el Maestro: E100 (periferia del esclavo) de longitud 2 bytes coherencia total. Los datos a transferir al maestro se depositan en el área de salidas 20 (PAA del esclavo) de longitud 1 byte coherencia unidad.

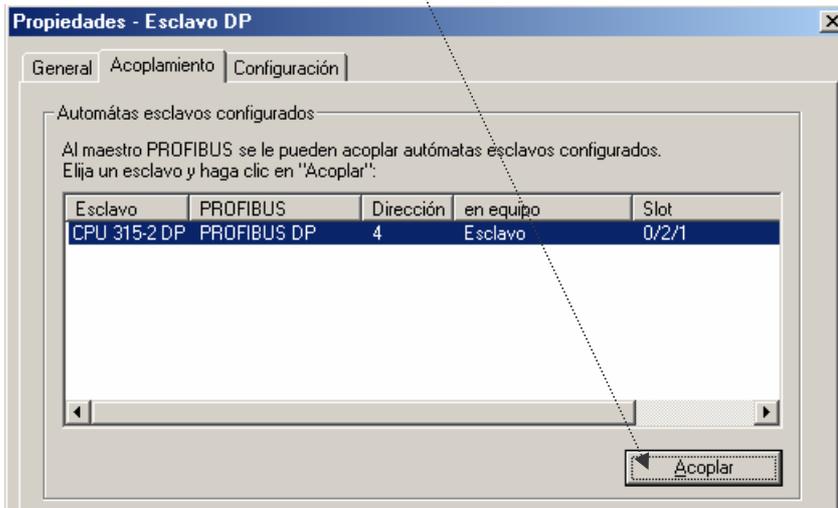
3. Ahora toca configurar el HW del equipo Maestro. Haciendo doble clic en el puerto integrado, conectamos la CPU a la red PROFIBUS DP ya creada. En la pestaña “Modo de operación” configuro la CPU como “Maestro”. Aceptamos y con el botón derecho sobre el puerto integrado selecciono: “Insertar sistema maestro”:



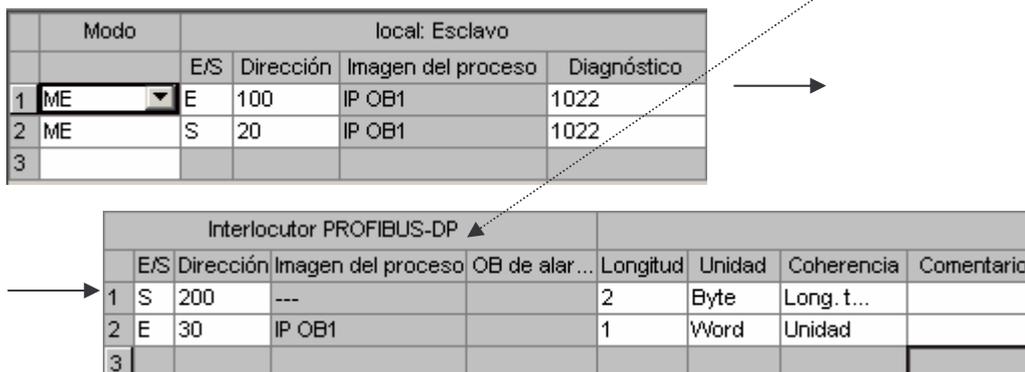
Del sistema maestro DP deberá colgar la CPU Esclavo. Para ello vamos al Catalogo de Hardware → Carpeta “PROFIBUS DP” → Carpeta “Estaciones ya configuradas” → Y aquí seleccionamos la CPU correspondiente.



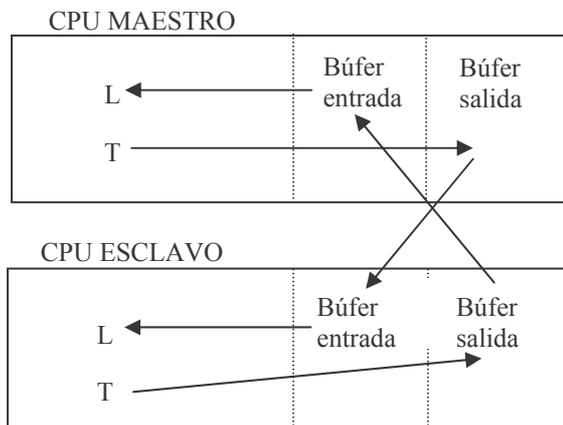
Nos saldrá la siguiente ventana, Pestaña “Acoplamiento”, donde aparecerán todos los esclavos previamente configurados. Seleccionamos aquel que queremos insertar en el sistema maestro DP y pinchamos en el botón “Acoplar”.



A continuación vamos a la pestaña “Configuración”, donde rellenamos la tabla para establecer las áreas de intercambio de datos de entradas y salidas del equipo maestro:



En el ejemplo anterior hemos definido un área (búfer) de salidas de datos de dirección 200 que se enviarán al área de entradas del esclavo. Y asimismo definimos en el maestro un área de entradas de datos donde se depositarán las salidas del esclavo.



Por ejemplo: Queremos mandar datos del esclavo al maestro.

Para ello enviamos los datos al búfer de salida del esclavo mediante una transferencia T. Automáticamente los datos depositados en el búfer de salida del esclavo se transfieren al búfer de entrada del maestro, de donde pueden ser leídos mediante una carga L.

4. Crearíamos los programas en el Maestro y el Esclavo. En nuestro ejemplo:

OB1 Maestro:

```
L   MW77
T   PAW200

L   PEW30
T   DB1.DBW0
```

OB1 Esclavo:

```
L   DB5.DBW0
T   PAW20

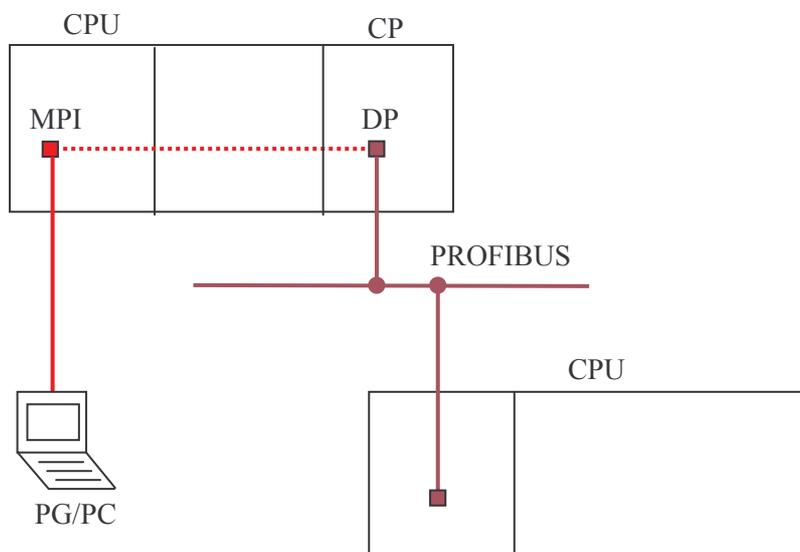
L   PEW100
T   MW34
```

Crear también el OB82 (OB de alarma de diagnóstico) y el OB86 (OB de fallo de la periferia) para que las CPU's no se vayan a STOP.

5. Transferir el hardware y programas de los dos equipos.

3.6. Routing

Se puede acceder a la red PROFIBUS a través de la red MPI. Se pueden hacer hasta 2 pasarelas.



Routing con módulos S7 a través de redes MPI, PROFIBUS y Ethernet Industrial

Requisitos para la función de Routing:

Requisitos de Software:

Redes MPI y PROFIBUS:

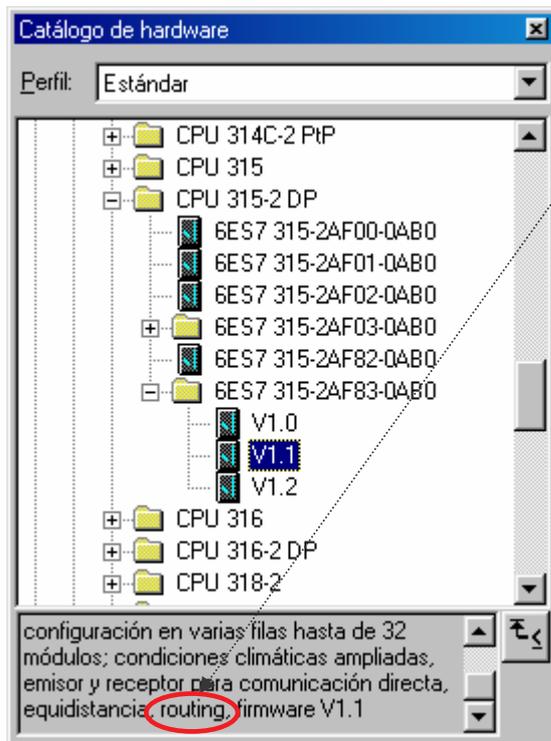
- Al menos STEP7 V5.0+SP1, pero se recomienda **STEP7 V5.0+SP3**.
- Para las tarjetas Sofnet PROFIBUS (**CP 5411 (ISA)**, **CP 5511 (PCMCIA)**, **CP 5611 (PCI)**, **CP 5611 (on board)**) no se necesitan drivers adicionales, ya se suministran con STEP7.
- Para las tarjetas Hardnet PROFIBUS (**CP 5412(A2) (ISA)**, **CP 5613/5614 (PCI)**) son imprescindibles los drivers del SIMATIC NET CD de Noviembre del 99.

Redes Ethernet Industrial:

- Para STEP7 V5.0+SP1 y SOFTNET IE [PG or S7] V3.1 para conexiones de Ethernet Industrial también se recomienda **STEP7 V5.0+SP3** y **SOFTNET IE [PG or S7] V3.2**.
- Para las tarjetas Sofnet IE (**CP 1411 (ISA)**, **CP 1511 (PCMCIA)**, **3COM**) son indispensables los drivers que vienen en el CD SIMATIC NET de Noviembre 99.
- Para las tarjetas Hardnet-IE (**CP 1413 (ISA)**, **CP 1613 (PCI)**) son indispensables los drivers que vienen en el CD SIMATIC NET de Noviembre 99.
- Para implementar un TS Adapter:
TeleService S7 V5.0.

Requisitos de hardware:

Se necesitan módulos compatibles con el routing. Acerca de los módulos que son compatibles con el routing véase el catálogo de hardware de la configuración hardware de STEP7.



Además se necesita una de las CP's mencionadas más arriba en "Requisitos de software" o el TS Adapter V5.0 o un a PC Adapter V5.0.

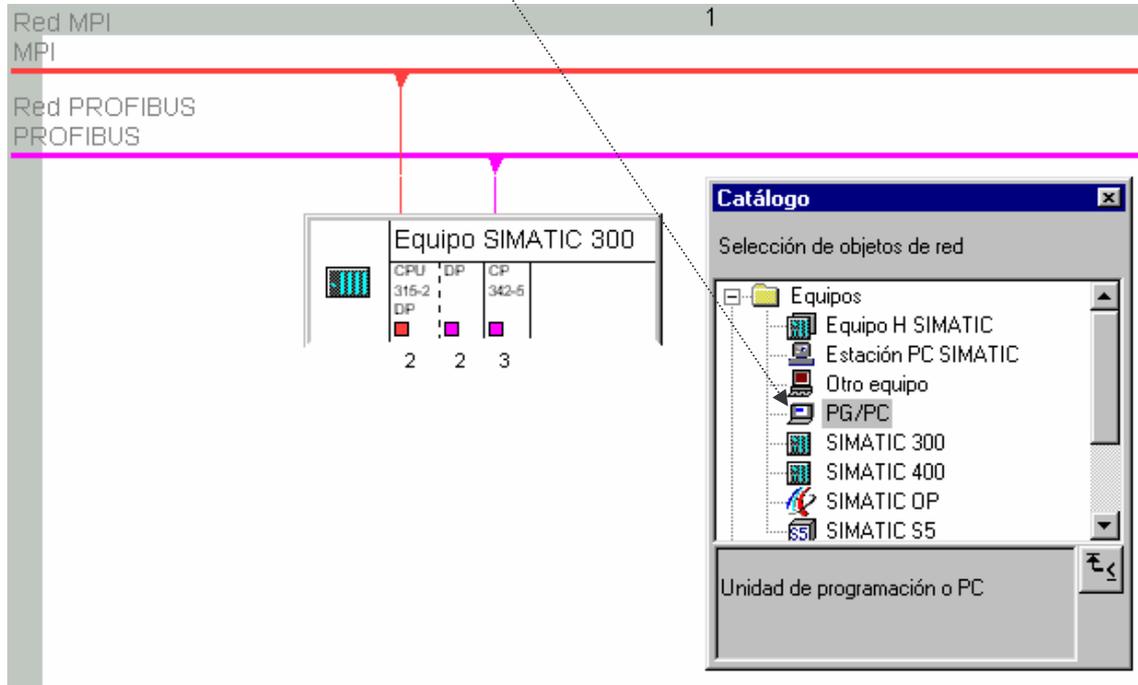
Observación:

El último módulo para ser direccionado a través de las funciones de routing no tiene por qué ser compatible con el routing.

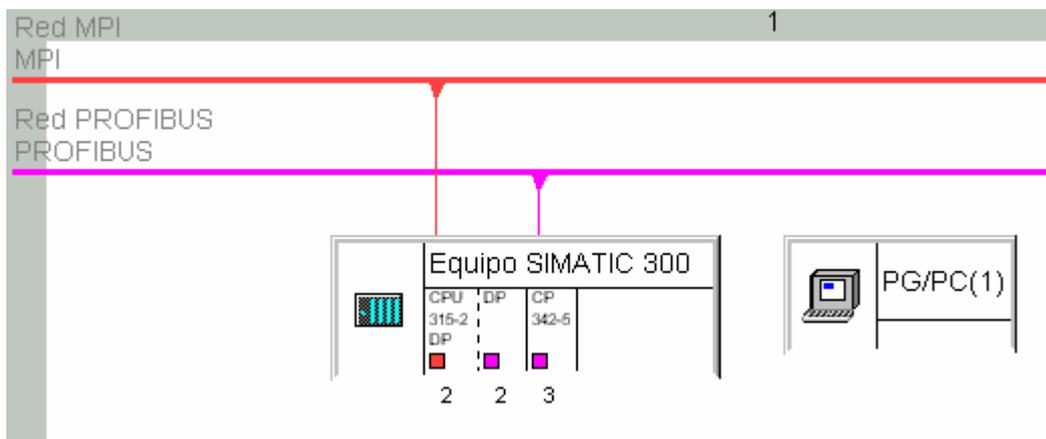
Procedimiento de configuración:

Todas las estaciones que están entre los dispositivos de inicio y final se deben configurar en un **único** proyecto STEP7. Desde el configurar de redes NetPro:

1. Añadir una estación del tipo "PG/PC".



2. Configurar todas las conexiones de red físicamente presentes (MPI, PROFIBUS, Industrial Ethernet) en el NETPRO o en el HW Config. Esto también se hará para las redes que no están directamente en el camino de ruta, pero que están conectadas a las estaciones de ruta. Esto **no** significa que se deban conectar a la red todos los interfaces disponibles. Sólo hay que configurar las conexiones de comunicación que estén físicamente presentes.

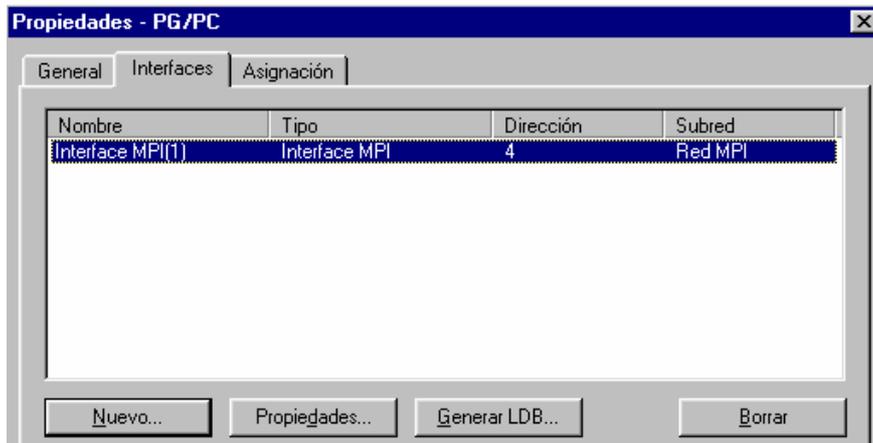


3. Abrir las Propiedades de la estación "PG/PC" haciendo doble clic sobre el dibujo de la PG/PC.

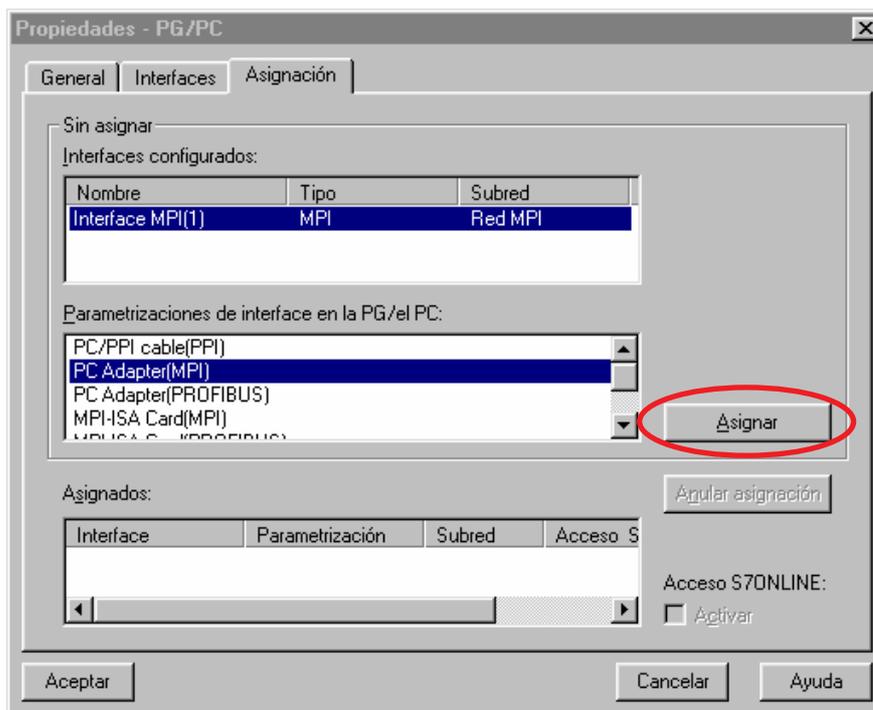
4. Accede a la pestaña "**Interfaces**" y crea un nuevo interface (botón "Nuevo"). Escoge el tipo de interface de la lista que se visualiza y confirma con "Aceptar".
5. En la ventana que se ha abierto escoge la red física a la que está conectada tu PG/PC. Si todavía no has configurado una red, entonces deberás hacerlo ahora.

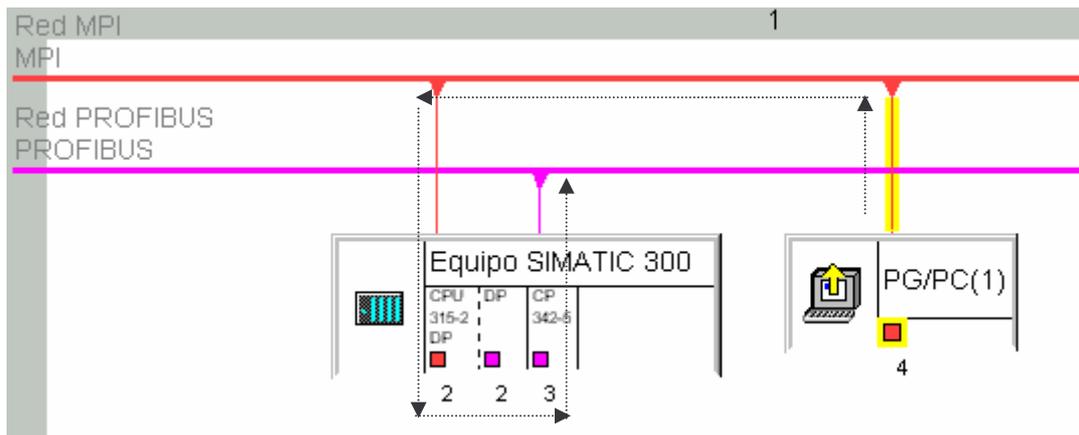
Advertencia:

La estación "PG/PC" **tiene que tener la misma dirección** que la que está establecida en el programa "Ajustar interface PG/PC" (esto también es de aplicación para la CP 1413 y CP 1613).



6. Accede a la pestaña "**Asignación**" y asigna el punto de entrada de routing a tu PG/PC. Primero, en el "Interfaces configurados" escoge el usuario y luego en la segunda ventana especifica la CP o el PC Adapter con el ID de la red asociada, a través de la cual tú quieres acceder online.

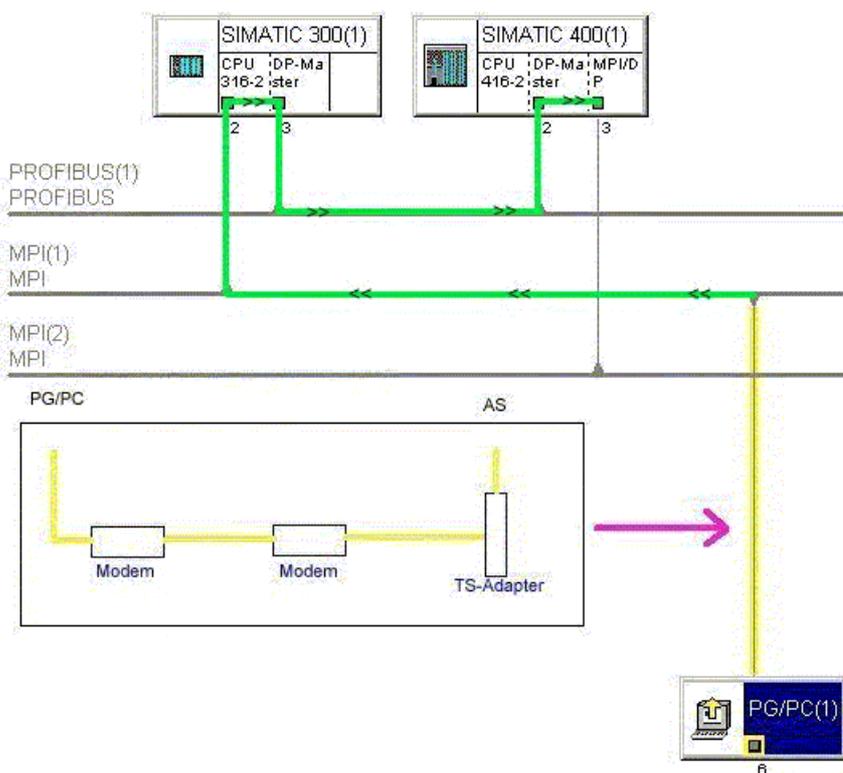




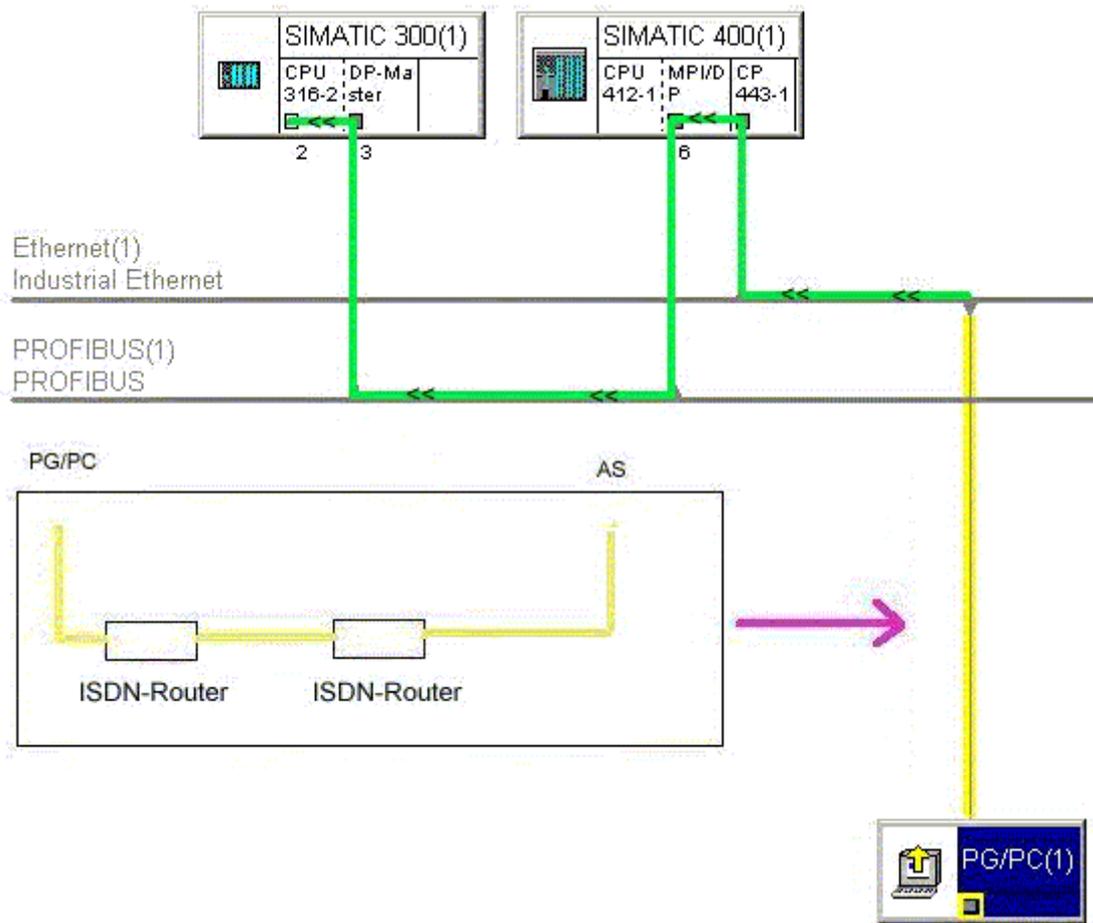
7. Guarda y compila la configuración de red.
8. Carga la configuración en la CPU.
9. Establece una conexión online a través del proyecto offline.
No se puede hacer esto a través de "**Estaciones accesibles**".

Configuraciones de ejemplo:

Routing con modem y TS Adapter

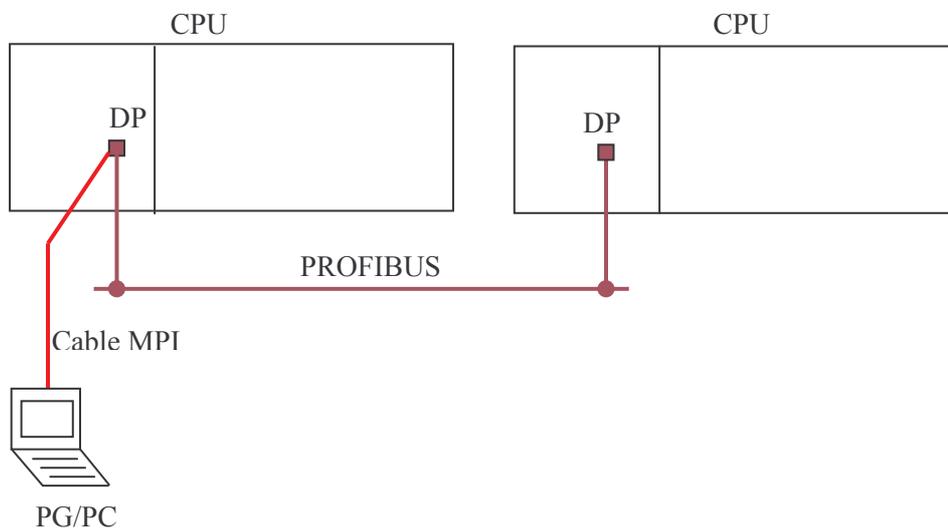


Routing con ISDN Router



3.7. PG BUS

Consiste en conectar la PG a la red PROFIBUS para acceder a los distintos equipos existentes en dicha red.



1. Configurar la red PROFIBUS entre los dos equipos.
2. Cambiar el interface de la PG a PROFIBUS en "Ajustar interface PG/PC". A la PG asignarle la dirección PROFIBUS 0.
3. El cable MPI de la PG se conecta al puerto integrado de la CPU o al puerto de la CP.
4. Ahora metiéndonos online en un proyecto podremos acceder a las todas CPU's de la red integrantes de la red PROFIBUS.

Observaciones:

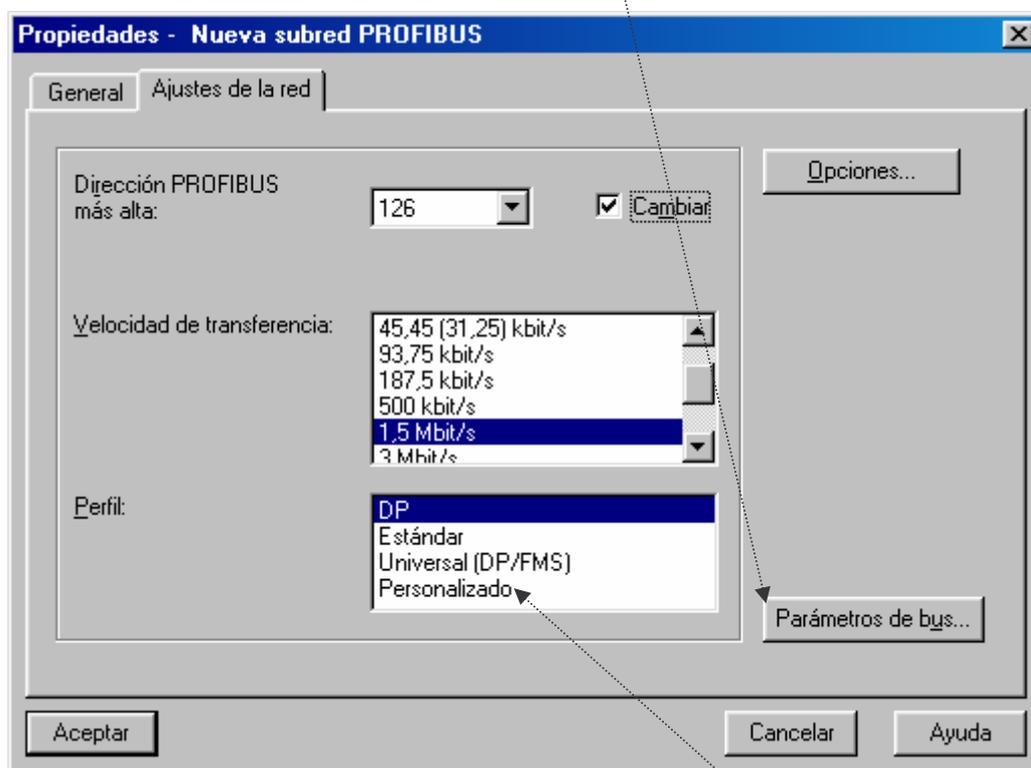
- Si se hace PGBUS a través de las CP's, si se hace el acceso online mediante "Mostrar estaciones accesibles", no se accederá al contenido de las CPU's, sino al de las CP's
- Recoger el hardware de los equipos a través de PROFIBUS no es demasiado seguro, es aconsejable hacerlo a través del cable MPI.

3.8. Configurar los parámetros de la red PROFIBUS

Podemos definir las propiedades de la red PROFIBUS.

- Dirección de PROFIBUS más alta (HSA – High Station Active Address): Es la dirección más alta de las estaciones activas de la red PROFIBUS. La dirección más alta de PROFIBUS es necesaria para pasar el token o testigo, es decir, para pasar el permiso de transmisión entre las estaciones activas. Sólo afecta a las estaciones PROFIBUS activas, como por ejemplo un maestro DP. Por consiguiente, los esclavos DP pasivos pueden tener asignadas direcciones PROFIBUS de 1 a 125, aunque la dirección más alta se haya ajustado p. ej. a 15.

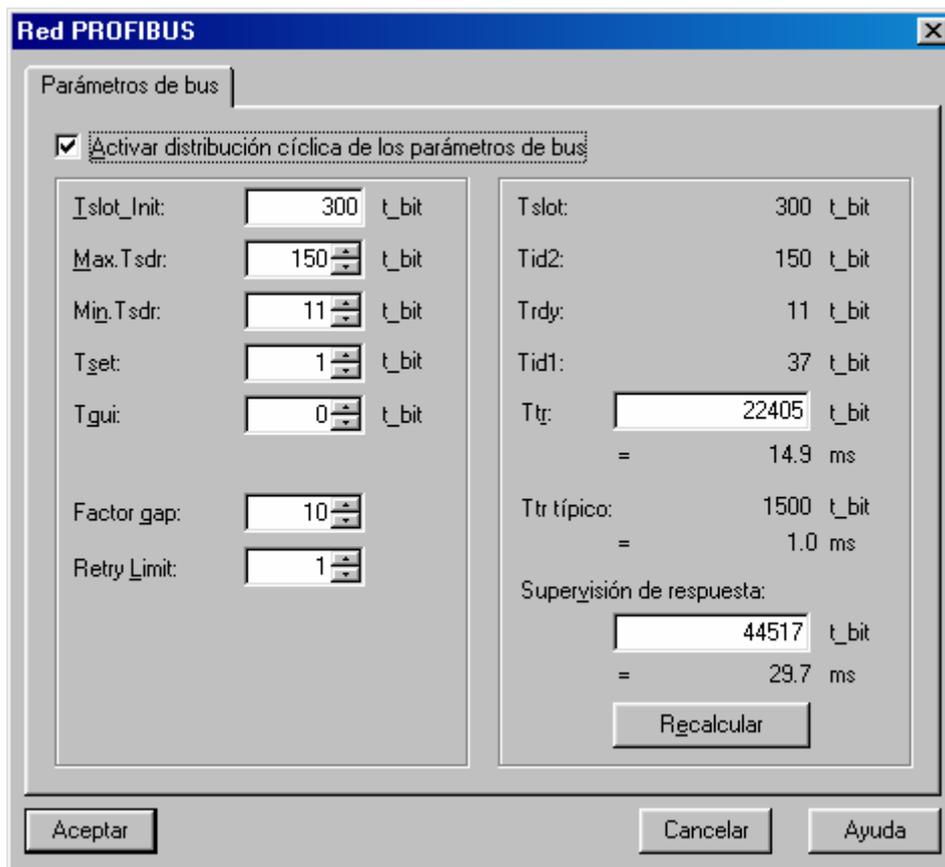
Podemos ajustar los parámetros de la red PROFIBUS:



Para poder cambiar estos parámetros deberemos elegir el perfil “Personalizado”. Además, todos los parámetros de bus deberán ser iguales en todos los equipos de la red.

GAP: Es un telegrama de consulta a una estación. Este telegrama se envía a una dirección PROFIBUS para ver si existe estación en esa dirección.

LAS (List of Active Stations) : Lista de las estaciones activas de la red generada cada X ciclos.



Todos los parámetros vienen definidos en la unidad **t_bit**, que es el tiempo que se tarda en transmitir un bit; este tiempo depende de la velocidad de la red.

- **Factor gap** (0÷99): Indica cada cuantos ciclos de bus se actualiza la LAS. Un factor gap bajo tiene el inconveniente de que se pierde tiempo de bus al mandar telegramas GAP, pero por el contrario la red es más dinámica ya que detecta con rapidez la incorporación de nuevos equipos.

TSDR (Time Station Delay Response): Tiempo de respuesta de una estación ante un mensaje que se le envía. No existe un tiempo fijo, depende de la estación. Se define un margen de respuesta: un TSDR mínimo y un TSDR máximo.

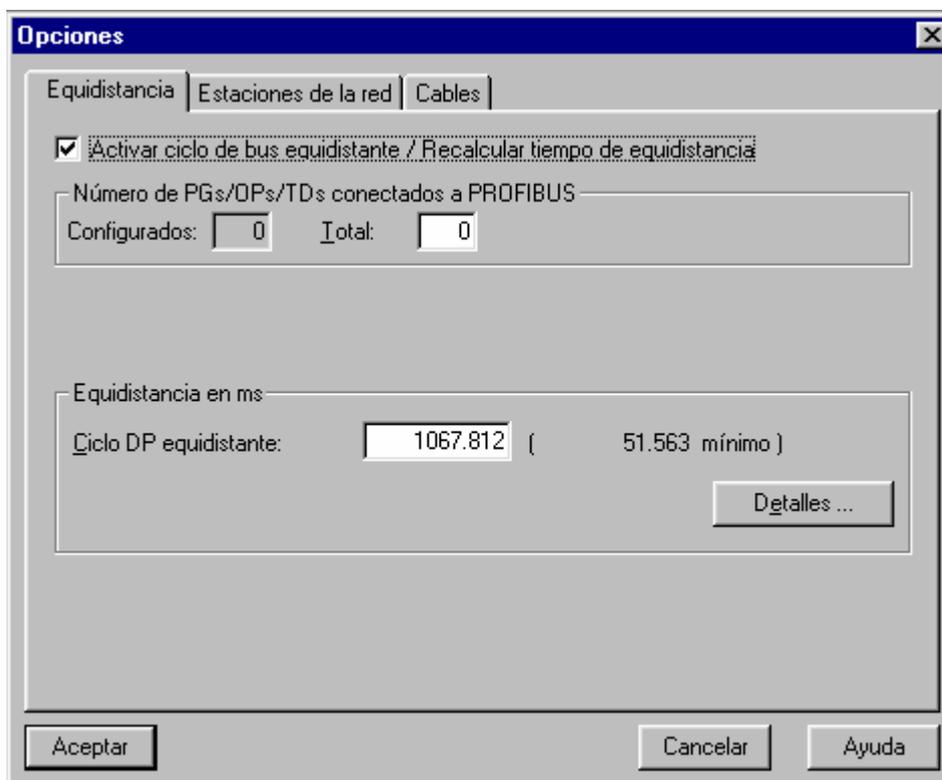
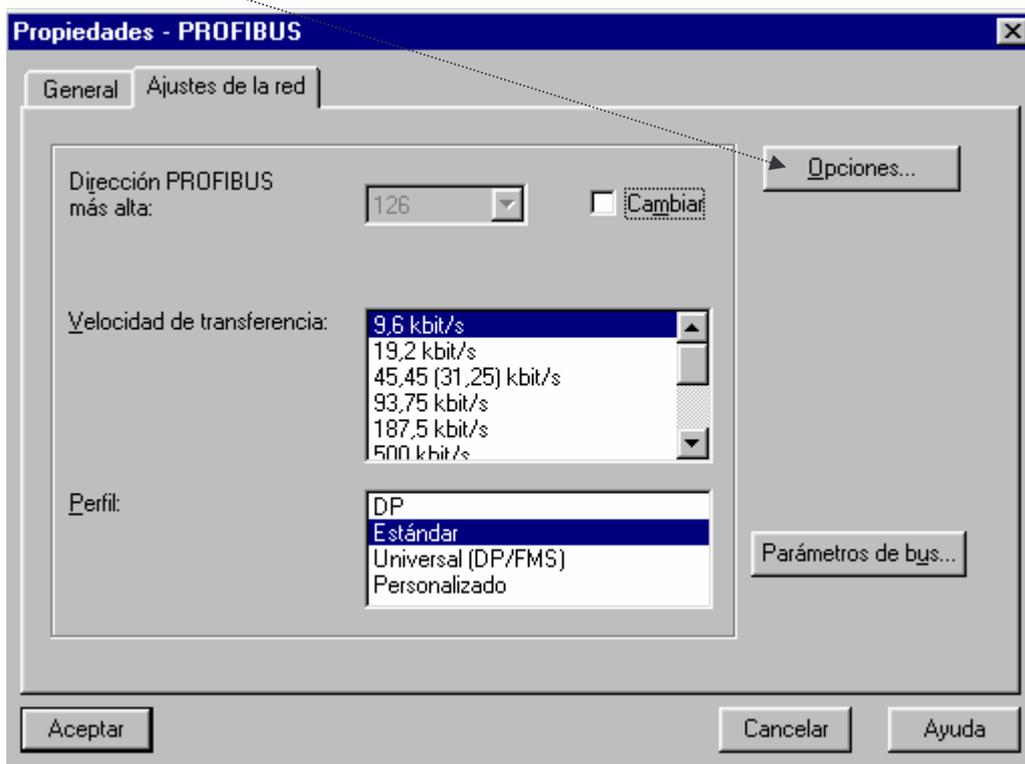
- **Tsdr mín.** : Determina el tiempo que debe transcurrir como mínimo para que la estación pueda responder
- **Tsdr máx.** : Determina el tiempo tras el cual tiene que haber respondido la estación

Si el tiempo de respuesta supera al Tsdr máx. aún espera el tiempo **Tslot** que es el máximo tiempo que espera el emisor a que responda su estación. Si se supera el Tslot, el emisor vuelve a enviar el telegrama tantas veces como indique el parámetro **Retry Limit**.

- **Ttr**: Tiempo máximo de rotación del testigo. Establece el tiempo máximo para que el testigo vuelva a una estación activa (maestro DP)
- **Tset** (Tiempo de preparación): Tiempo que transcurre entre la recepción de un telegrama en una estación y la reacción de la dicha estación a ese telegrama.

3.9. Opciones de la red PROFIBUS

En ventana de propiedades de PROFIBUS, pestaña “Ajustes de la red”, seleccionar el botón “Opciones”:



□ ***Pestaña “Equidistancia”:***

Equidistancia es la propiedad de PROFIBUS DP que garantiza ciclos equidistantes, o sea, ciclos de idéntica duración. Cuando los ciclos de bus son equidistantes, el maestro DP inicia el ciclo de bus DP siempre una vez transcurrido el mismo período de tiempo, lo que significa que desde la perspectiva de los esclavos conectados, éstos también reciben sus datos del maestro a intervalos de idéntica duración.

La equidistancia se puede ajustar sólo para redes PROFIBUS con un solo maestro DP (clase 1). Los maestros DP clase 1 son maestros DP que acceden cíclicamente a los esclavos DP que tienen asignados – es decir, no pueden ser PGs/PCs.

Activar ciclo de bus equidistante.

Tiempo de equidistancia en ms, STEP 7 le propone un valor calculado en base a la configuración que se haya hecho en NetPro.

□ ***Pestaña “Estaciones de las red”:***

Para calcular los tiempos de bus puede establecer una configuración de red que difiera de la configuración de red creada con STEP 7. La configuración de red se puede ajustar para los perfiles "Estándar" y "Universal (DP/FMS)".

□ ***Pestaña Cables:***

En esta ficha se pueden introducir datos acerca de la estructura de la red PROFIBUS. Dichos datos se tendrán en cuenta al calcular los parámetros de bus de STEP 7 si activa la casilla de verificación "Considerar la configuración de los cables".

4. PROFIBUS FDL (SEND/RECEIVE)

FDL (Fieldbus Data Link) ofrece servicios de funciones compatibles con S5 para transmitir datos por la subred PROFIBUS. Esta transmisión de datos se hace únicamente entre equipos inteligentes (CPU's o PC's).

No es un protocolo estándar, sólo se pueden comunicar equipos de Siemens.

SIMATIC S7 ofrece funciones de comunicación para emitir y recibir datos a través de enlaces FDL, usando los bloques estándar de FDL: AG_SEND (FC5) y AG_RECV (FC6) a través de la red PROFIBUS. De este modo se pueden mandar de una sola vez 240 bytes máx. También podemos utilizar las funciones AGL_SEND (FC50) y AGL_RECV (FC60) con las que podremos mandar 8 kbytes máximo.

La comunicación a través del interface SEND/RECEIVE se efectúa exclusivamente mediante CP's.

Requisitos de hardware:

S7-300 :

- CP 342-5 y CP343-5

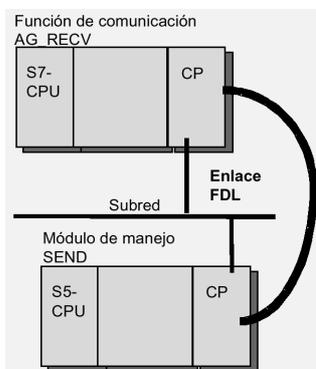
S7-400:

- CP 343-5 BASIC Y EXTENDED

Enlaces:

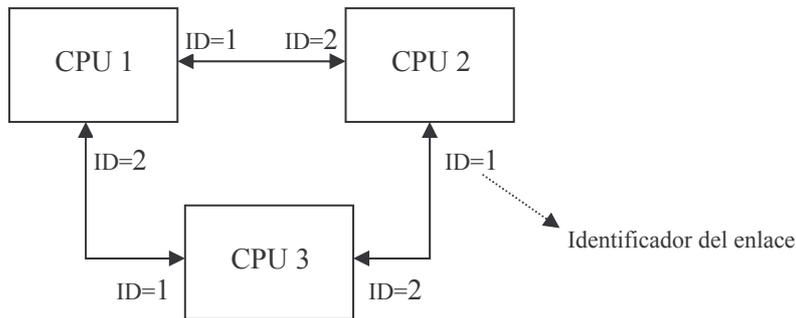
Un enlace constituye la correspondencia lógica entre dos interlocutores para ejecutar servicios de comunicación.

Cada enlace tiene dos puntos finales (situados en la respectiva CPU o CP), que incluyen las informaciones necesarias para direccionar el interlocutor, así como otros atributos para establecer el enlace.



Los enlaces son bidireccionales, creamos uno para comunicar dos equipos.

Si tenemos n equipos, tendremos n-1 enlaces:



Recursos de enlace: Cada enlace precisa recursos en los equipos implicados para el punto final (por ej. CPU) o para el punto de transición (p.ej. CP). El número de recursos de enlace depende del tipo de CPU/CP. Si están ocupados todos los recursos de enlace de un interlocutor no puede establecerse ningún nuevo enlace.

Tipos de enlaces:

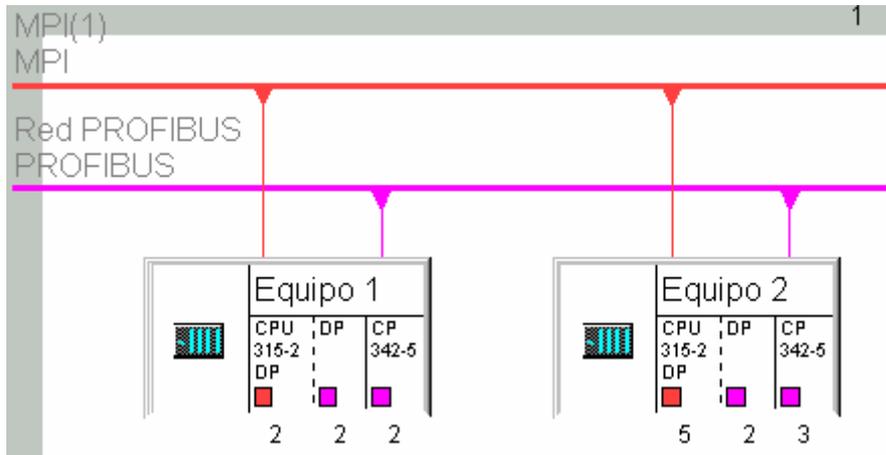
Enlaces			
Tipo de enlace	Configurado (vía tabla de enlaces)		no configurado
Establecimiento/ disolución del enlace	Estático	dinámico (sólo en M7-300/400)	dinámico

- **Enlaces configurados:** Esta clase de enlace se configura mediante STEP 7 (en la Tabla de Enlaces). Se asigna a cada extremo de enlace un ID local; éste se precisa para parametrizar las funciones de comunicación. El ID local referencia un área de datos que contiene, entre otros, las informaciones de dirección propias y las del interlocutor. Estos enlaces requieren recursos de la CPU. Ejemplo: Enlaces FMS Y FDL.
- **Enlaces no configurados:** Por regla general, este tipo de enlaces no se configuran en STEP 7 a través de la Tabla de enlaces. Dichos enlaces se especifican sólo mediante la dirección de destino y se establecen implícitamente al solicitar la función de comunicación, disolviéndose eventualmente tras acabar la transmisión de los datos. Ejemplo: Enlaces S7.
- **Estático:** Se utilizan enlaces estáticos cuando al configurar una instalación hay suficientes recursos de enlaces disponibles en los diferentes equipos y no es necesario liberar ninguno. Por otro lado, al planificar no es necesario considerar el establecimiento y disolución del enlace, de tiempo crítico. Una vez establecidos, los enlaces estáticos permanecen de forma duradera.
- **Dinámico** Los enlaces dinámicos se utilizan para intercambiar sucesivamente datos con diferentes interlocutores o para aprovechar con más efectividad los recursos de enlace disponibles. El establecimiento y disolución propiamente dicho del enlace no se efectúa al arrancar el equipo, sino sólo cuando lo solicita explícitamente la aplicación desde el programa.

El enlace FDL es configurado y estático. El número de enlaces máximo es de 16.

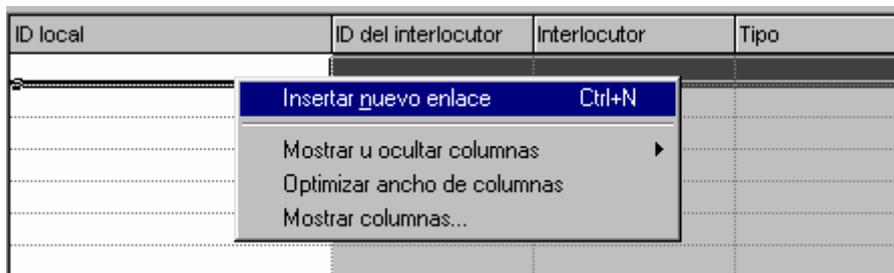
Pasos para configurar una red FDL:

1. Configuro el hardware de los equipos a comunicar. Y en el NetPro configuro la red PROFIBUS.

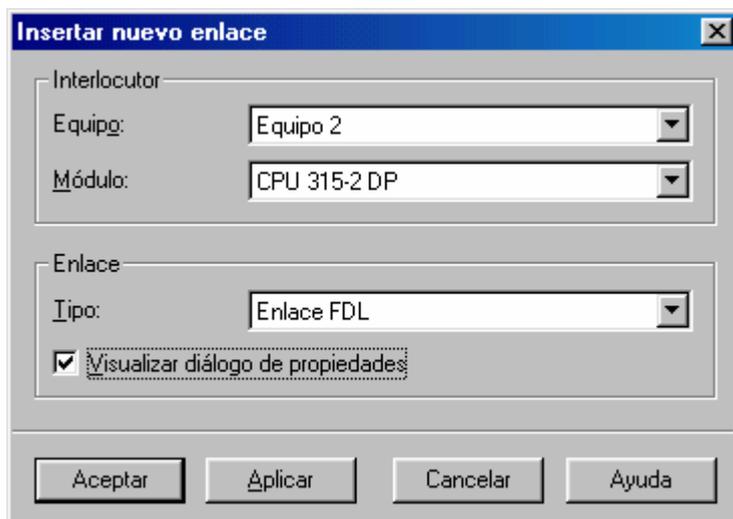


Debo conectar las CP's a una red PROFIBUS. El perfil de la red debe ser “Estándar”.

2. El segundo paso es configurar los enlaces. Los enlaces se configuran en la CPU. Haciendo clic sobre una CPU me aparecerá la Tabla de enlaces. Haciendo clic sobre la primera línea y botón derecho del ratón, “Insertar nuevo enlace”.



Seleccionar el equipo y módulo interlocutor. Como tipo de enlace seleccionar “FDL”:



Activar “Visualizar dialogo de propiedades” para que al pulsar “Aceptar” me aparezca la ventana de propiedades del enlace:



- ID local: Se trata del ID del enlace FDL (máx. 16 enlaces).
- Nombre del enlace : Nombre asignado al enlace. Identifica al enlace FDL.

Parámetros de bloque:

- ID: Número de ID local
- LADDR: Dirección inicial del módulo (de la CP) en hexadecimal. La dirección es la del HW Config.



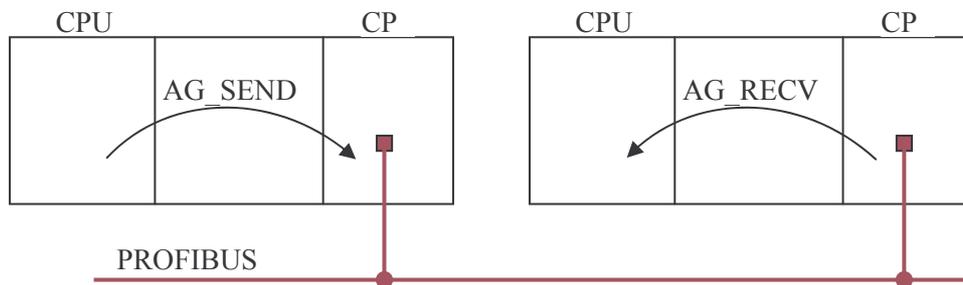
“Guardar y compilar” y



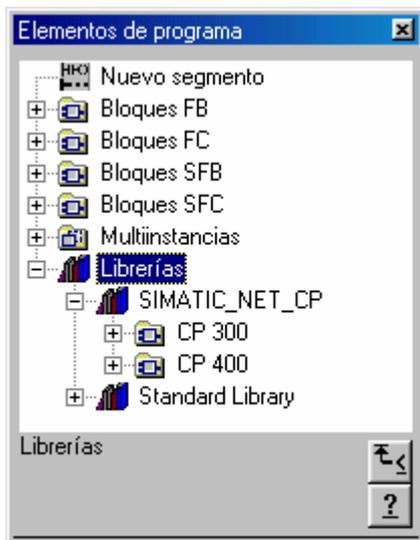
“Cargar equipos seleccionados”

3. Programación:

La comunicación se realiza a través de la CP, pero los datos los envío o recibo desde la CPU mediante las funciones FC5 "AG_SEND" y FC6 "AG_RECV".



Estas funciones están en 'Elementos de programa' → "Librerías" → "SIMATIC_NET_CP" → "CP300" o "CP400".



Ejemplo: Voy a mandar datos del Equipo 1 y recibirlos por el Equipo 2.

OBI Equipo 1:

```
CALL "AG_SEND"  
ACT    :=  
ID     :=  
LADDR :=  
SEND  :=  
LEN   :=  
DONE  :=  
ERROR :=  
STATUS :=
```

Parámetros de entrada:

- ACT (BOOL) : Si ACT = 1 se envían los bytes.
Si ACT = 0 se actualizan los códigos de condición de estado DONE, ERROR y STAT
- ID (INT) : Número del enlace FDL (1÷16) por el que voy a enviar.
- LADDR (WORD) : Dirección inicial de la CP por la que voy a enviar. Al configurar el CP con la herramienta de configuración “HW Config” se visualiza la dirección inicial del módulo en la tabla de configuración. Introduzca aquí esta dirección en hexadecimal.
Ej: W#16#130
- SEND (ANY) : Dirección y longitud de los datos que voy a enviar.

El área de datos puede ser la PAE, PAA, marcas o bloques de datos.

Ej.: P#E0.0 BYTE 2 //Mandaría EB0 y EB1

P#DB1.DBX0.0 BYTE 40 //Mandaría los primeros 40 bytes del DB1

- LEN (INT): Número de bytes (1 hasta 240 máx.) que enviaré del area de datos indicada en el parámetro SEND. Este número debe ser menor que la longitud de datos indicada con SEND.

Parámetros de salida:

- DONE (BOOL): Si DONE=1, significa que la petición de envío ha terminado con o sin errores.
- ERROR (BOOL): Si ERROR=1, se ha producido un error
- STATUS (WORD): Código de estado.

Envío terminado sin error: DONE=1, ERROR=0 y STATUS=0000

En nuestro ejemplo:

```
CALL "AG_SEND"  
ACT := 1  
ID := 1  
LADDR := W#16#100  
SEND := P#E0.0 BYTE 2  
LEN := 2  
DONE := M0.0  
ERROR := M0.1  
STATUS:= MW2
```

OBI Equipo 2:

Recogeré los datos mediante la función FC6 "AG_RECV". No hay que activar la recepción, siempre estará alerta para recibir los datos.

```
CALL "AG_RECV"  
ID      :=  
LADDR  :=  
RECV   :=  
NDR    :=  
ERROR  :=  
STATUS:=  
LEN    :=
```

Parámetros de entrada:

- ID (INT) : Indica el número del enlace FDL (1÷16).
- LADDR (WORD) : Dirección inicial de la CP por la que recibo, en hexadecimal.
- RECV (ANY) : Area de memoria donde almaceno los datos que recibo. El área de datos puede ser la PAE, PAA, marcas o bloques de datos.

Parámetros de salida:

- LEN (INT): Indica el número de bytes (1÷240) que han sido recibidos por la CP en el área de datos FDL.
- NDR (BOOL): Si NDR=1, indica si se aceptaron nuevos datos.
- ERROR (BOOL): Si ERROR=1, se ha producido un error
- STATUS (WORD): Código de estado.

En nuestro ejemplo:

```
CALL "AG_RECV"  
ID      := 1  
LADDR  := W#16#100  
RECV   := P#A4.0 BYTE 2  
NDR    := M0.0  
ERROR  := M0.1  
STATUS:= MW2  
LEN    := MW4
```

5. Protocolo S7

Cada CPU S7 lleva integradas una serie de funciones S7 (SFB's y SFC's) para la transferencia de datos entre ellas. Estas funciones son independientes del tipo de red, pueden utilizarse para MPI, PROFIBUS e Ethernet.

Gracias a la repetición automática de los telegramas incompletos o erróneos en PROFIBUS y MPI, se consigue una elevada seguridad de los datos.

Tipos:

5.1. Comunicación a través de red MPI (Enlaces no configurados)

Se usan unas SFCs para comunicaciones a través de enlaces S7 no configurados. Los enlaces hacia los interlocutores se establecen dinámicamente al solicitar la SFC. Por ello se requiere en cada interlocutor un recurso de enlace libre. Máx. 76 bytes.

Función SFC68 "X_PUT": Esta SFC permite a una CPU escribir datos en otra CPU sin que ésta última deba tener ninguna SFC, ya que esta funcionalidad la realiza el sistema operativo.

```
CALL  "X_PUT"  
REQ   :=  
CONT  :=  
DEST_ID :=  
VAR_ADDR :=  
SD     :=  
RET_VAL :=  
BUSY   :=
```

Parámetros de entrada:

- REQ (BOOL): La escritura de datos se activa llamando la SFC con REQ=1 (cambio de nivel). Si no existe enlace, se crea antes de enviar los datos. Seguidamente se llama repetidamente a la SFC hasta que se señalice, con BUSY=0, que la escritura de datos ha finalizado. Si se vuelve a llamar a la función con REQ=1 y todavía no se han enviado los datos, está nueva petición se desestima.
- CONT (BOOL):

Si CONT=0 en la primera llamada: El enlace se rompe después de enviar los datos. Con esto aseguramos de sólo ocupar los recursos de enlaces que son actualmente necesarios.

Si CONT=1 en la primera llamada: El enlace permanece tras finalizar la transferencia de los datos. Esto lo utilizaré para el intercambio de datos cíclicos entre dos estaciones. De todos modos, para romper el enlace dispongo de la función SFC 69 "X_ABORT".
- DEST_ID (WORD): Contiene la dirección MPI de la CPU donde se va a escribir.

- VAR_ADDR (ANY): Indica el área de datos de la CPU donde se va escribir. Es necesario elegir un tipo de datos soportado por el CPU receptora.
- SD (ANY): Indica el área de datos de la CPU emisora que se van a enviar. Se permiten los tipos de datos siguientes: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME, así como arrays de los tipos de datos mencionados con excepción de BOOL.

Hay que tener en cuenta que la longitud del área de datos y el tipo de datos de la CPU emisora deben coincidir con que la longitud del área de datos y el tipo de datos de la CPU receptora.

Parámetros de salida:

- RET_VAL (INT): Si durante la ejecución de la función aparece un error, este valor de retorno incluye su código de error asociado.
- BUSY (BOOL):
 BUSY=1: La emisión aún no ha finalizado.
 BUSY=0: La emisión ha finalizado o no hay ninguna emisión activa.

Si la CPU que envía los datos pasa a STOP: Se rompe el enlace.

Si la CPU donde se escriben los datos pasa a STOP: No pasa nada, los datos emitidos (que estarán en un búfer) se escriben de todos modos.

Función SFC67 "X_GET": Esta SFC permite a una CPU leer datos de otra CPU sin que ésta última deba tener ninguna SFC, ya que esta funcionalidad la realiza el sistema operativo.

```
CALL  "X_GET"
REQ   :=
CONT  :=
DEST_ID :=
VAR_ADDR :=
RET_VAL :=
BUSY  :=
RD    :=
```

- REQ (BOOL): La lectura de datos se activa llamando la SFC con REQ=1. Si no existe enlace, se crea.
- CONT (BOOL):

Si CONT=0 en la primera llamada: El enlace se rompe después de leer los datos. Con esto aseguramos de sólo ocupar los recursos de enlaces que son actualmente necesarios.

Si CONT=1 en la primera llamada: El enlace permanece tras finalizar la lectura de los datos. Esto lo utilizaré para el intercambio de datos cíclicos entre dos estaciones. De todos modos, para romper el enlace dispongo de la función SFC 74 "I_ABORT".

- DEST_ID (WORD): Contiene la dirección MPI de la CPU de la que se va a leer.
- VAR_ADDR (ANY): Indica el área de datos de la CPU de donde se va a leer. Es necesario elegir un tipo de datos soportado por el CPU receptora.

Parámetros de salida:

- RET_VAL (INT): Si durante la ejecución de la función aparece un error, este valor de retorno incluye su código de error asociado. Si no aparece ningún error, RET_VAL incluye la longitud en bytes del paquete de datos que se ha leído.
- BUSY (BOOL):
BUSY=1: La recepción aún no ha finalizado.
BUSY=0: La recepción ha finalizado o no hay ninguna emisión activa.
- RD (ANY): Indica el área de datos de la CPU receptora donde se van a almacenar los datos leídos. Se permiten los tipos de datos siguientes: BOOL, BYTE, CHAR, WORD, INT, DWORD, DINT, REAL, DATE, TOD, TIME, S5_TIME, DATE_AND_TIME, así como arrays de los tipos de datos mencionados con excepción de BOOL.

Hay que tener en cuenta que la longitud del área de datos y el tipo de datos de la CPU receptora deben coincidir con que la longitud del área de datos y el tipo de datos de la CPU emisora.

Para que haya coherencia de datos en la recepción, el área de datos RD donde se almacenan los datos recibidos se debe evaluar antes de hacer una nueva recepción en esa área.

Si la CPU donde se escriben los datos recibidos pasa a STOP: Se rompe el enlace. Los datos recibidos se perderán o no según el tipo de rearranque: Si "Rearranque" (no en los S7-300), no se pierden los datos, se escriben en el área RD. Si "Rearranque completo (caliente)" o "Rearranque frío", se pierden los datos.

Si la CPU de donde se leen los datos pasa a STOP: No pasa nada, los datos son leíbles aún estando la CPU en STOP.

Ej:

```
CALL  "X_PUT"  
  REQ      := E0.0           //Activo la transferencia con E0.0  
  CONT     := FALSE  
  DEST_ID  := W#16#4         //Dirección MPI=4 de la CPU de destino  
  VAR_ADDR:= P#A4.0 BYTE 1  //Dónde guardo los datos en la CPU de destino  
  SD       := P#E1.0 BYTE 1 //Datos a enviar  
  RET_VAL  := MW0  
  BUSY     := A5.0
```

5.2. Enlaces configurados

Los SFBs para comunicaciones permiten realizar una transmisión de datos acusada vía enlaces S7 configurados. Dichos enlaces se crean con STEP 7.

Los SFBs para comunicaciones sólo pueden aplicarse en los CPUs de la familia S7-400. Las funciones PUT/GET permiten leer o inscribir datos de un S7-300 desde un S7-400. Estas funciones permiten transmitir datos hasta a 64 Kbytes vía las subredes MPI, PROFIBUS e Industrial Ethernet.

La comunicación es posible exclusivamente dentro de un proyecto S7. Los interlocutores accesibles deberán estar conectados a una misma red MPI, PROFIBUS o ETHERNET.

Funciones:

SFB14 (GET) : Lee por programa datos del programa de aplicación del interlocutor.

SFB15 (PUT) : Escribe por programa datos en el programa de aplicación del interlocutor.

SFB 15 "PUT": Escribe datos en una CPU remota cuando hay un flanco positivo en la entrada de control REQ .

```
CALL  "PUT" , DB15
      REQ   :=
      ID    :=
      DONE  :=
      ERROR :=
      STATUS:=
      ADDR_1:=
      ADDR_2:=
      ADDR_3:=
      ADDR_4:=
      SD_1  :=
      SD_2  :=
      SD_3  :=
      SD_4  :=
```

Parámetros de entrada:

- REQ (BOOL) Parámetro de petición, activa el intercambio de datos con flanco positivo.
- ID (WORD) Parámetro de direccionamiento ID

Parámetros de salida:

- DONE (BOOL) Parámetro de estado
 - 0: Petición aún no iniciada o en curso.
 - 1: Petición realizada sin errores
- ERROR (BOOL)
 - 0: STATUS TIENE VALOR 0000H (sin error)

1: STATUS informa sobre el error

- STATUS (WORD)

Parámetros de entrada y salida:

- ADDR_1, ADDR_2, ADDR_3 y/o ADDR_4 (ANY) : Puntero señala a las áreas de la CPU interlocutora en la que se debe escribir.
- SD_1, SD_2, SD_3 y/o SD_4 (ANY): Puntero señala a las áreas de la CPU en las que están los datos a enviar.

Nota:

Si el puntero ANY accede a un DB, este DB se debe especificar siempre (p. ej., : p# DB10.DBX5.0 Byte 10).

Ej.: Crear un enlace tipo S7 en una CPU 400.

```
CALL PUT, DB15
  REQ:= E0.0 //Cuando flanco positivo de E0.0, transfiero los datos
  ID:=
  DONE:= M0.0
  STATUS:= MW10
  ADDR_1:= P#A0.0 BYTE 1 //Escribo en las salidas de la CPU interlocutora
  ADDR_2:=
  ADDR_3:=
  ADDR_4:=
  SD_1:= P#E0.0 BYTE 1 //Datos a mandar.
  SD_2:=
  SD_3:=
  SD_4:=
```

SFB 14 "GET": Permite leer datos de una CPU remota.

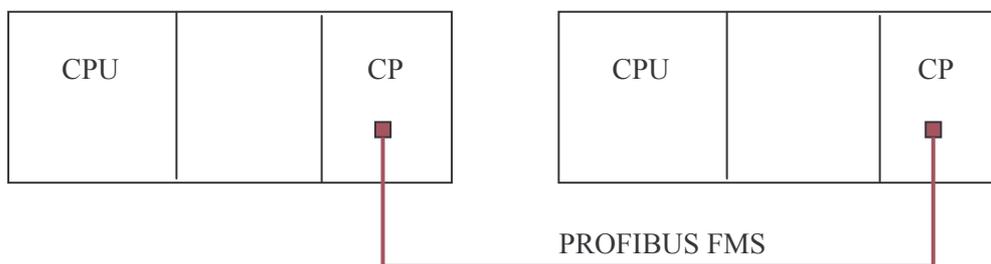
```
CALL "GET" , DB14
  REQ :=
  ID :=
  NDR :=
  ERROR :=
  STATUS:=
  ADDR_1:=
  ADDR_2:=
  ADDR_3:=
  ADDR_4:=
  RD_1 :=
  RD_2 :=
  RD_3 :=
  RD_4 :=
```

6. Protocolo FMS

PROFIBUS-FMS (Fieldbus Message Specification) ofrece servicios para la transferencia de datos estructurados (variables FMS). Hasta 237 bytes. El servicio FMS puede clasificarse en el nivel 7 del modelo de referencia ISO. Permite así una comunicación abierta con equipos no Siemens.

Propiedades

Los servicios FMS permiten leer, inscribir y notificar las variables FMS a través de enlaces FMS. El interlocutor confirma la recepción de los datos mediante un acuse de usuario, es decir la aplicación en el interlocutor remoto ha recibido correctamente los datos. La transmisión de los datos usando el servicio FMS se realiza exclusivamente a través de una red PROFIBUS.



Requisitos hardware:

S7-300:

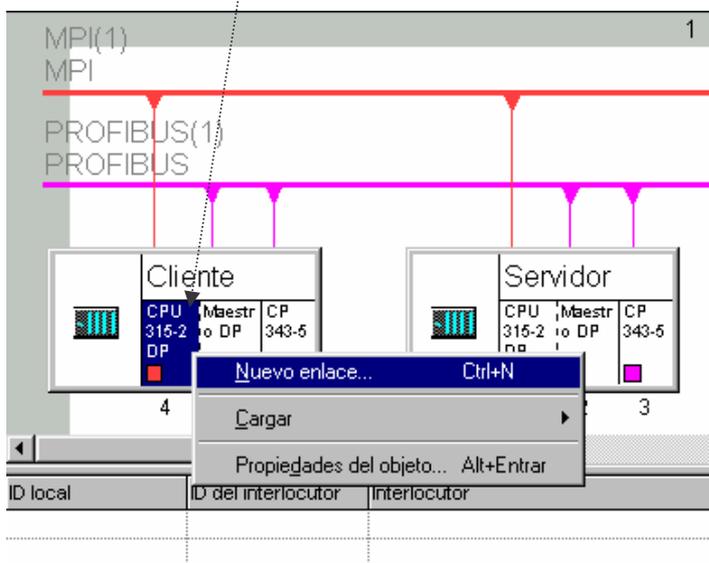
- CP343-5

S7-400:

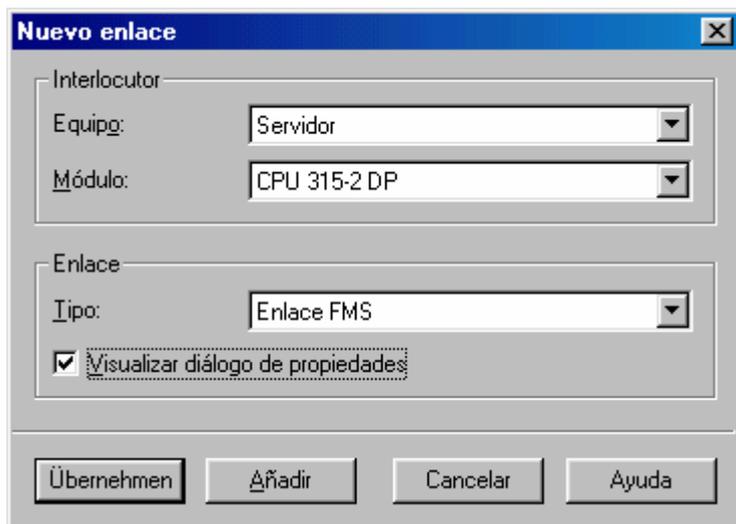
- CP443-5 Basic

1.- Configuramos el HW de los 2 equipos, el Cliente y Servidor. Las 2 CPs se deben conectar a una red PROFIBUS con perfil “Estándar”.

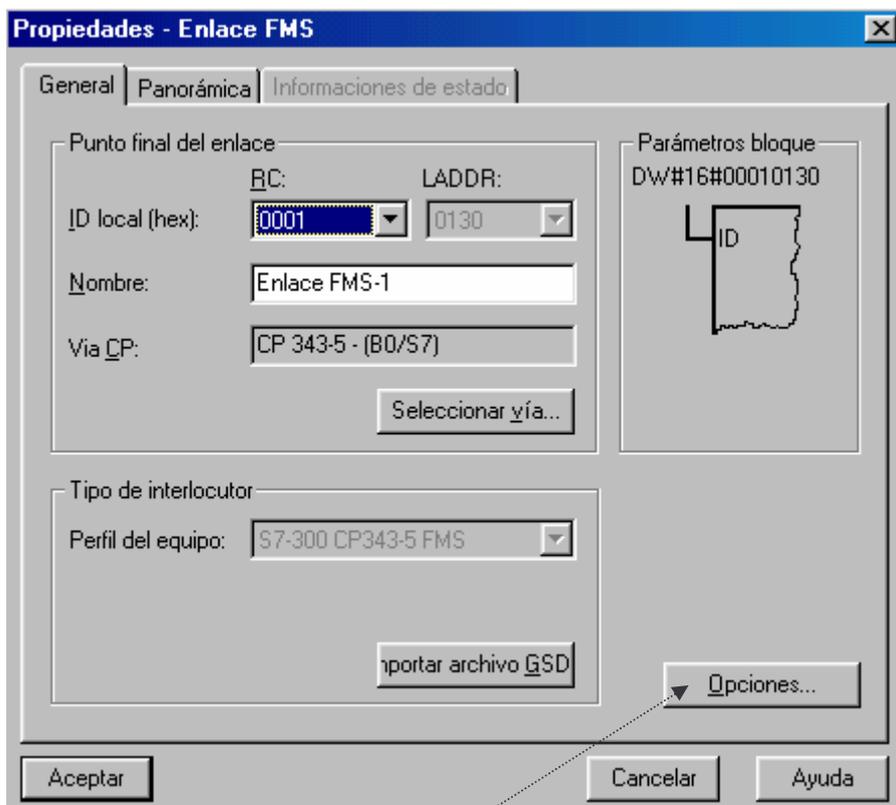
2. Configuro el cliente. En el NetPro creo los enlaces entre las CPU. Hago clic con el botón derecho sobre la CPU y creo un enlace nuevo:



Configuro el enlace. Selecciono el equipo interlocutor y tipo de enlace FMS:

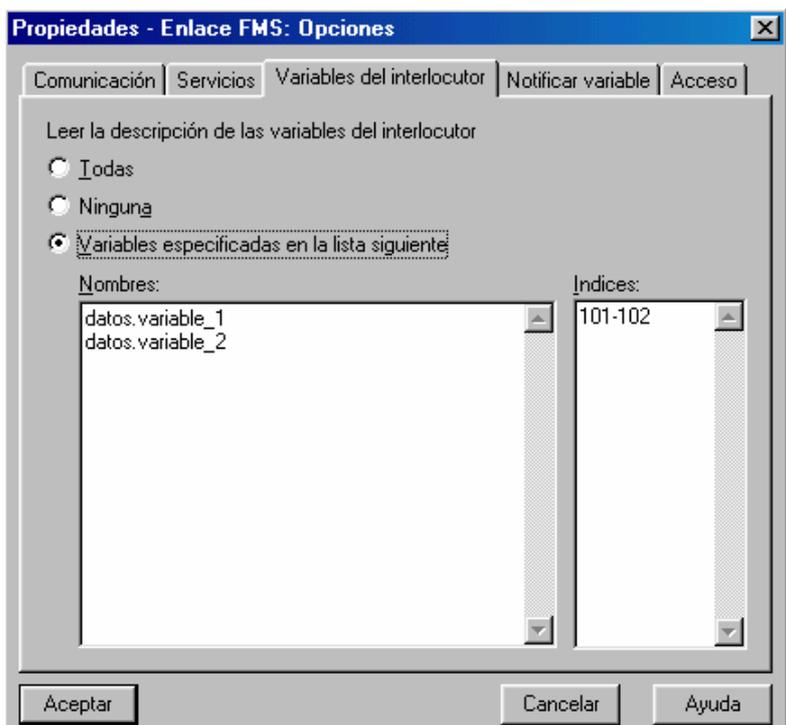


Las propiedades del enlace FMS:



Hago clic sobre las opciones

En las variables del interlocutor indico las variables FMS:



Creo el DB1 en el cliente:

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	indice_1	STRING[5]	'101'	
+8.0	indice_2	STRING[5]	'102'	
=16.0		END_STRUCT		

3.- En el servidor creo un DB :

Dirección	Nombre	Tipo	Valor inicial	Comentario
0.0		STRUCT		
+0.0	variable_1	BYTE	B#16#0	
+1.0	variable_2	BYTE	B#16#0	
=2.0		END_STRUCT		

En la tabla de símbolos:

	S	V	M	C	Símbolo	Dirección	Tipo de datos	Comentario
1				X	datos	DB 1	DB 1	
2								

Cortar Ctrl+X
 Copiar Ctrl+C
 Pegar Ctrl+V
 Borrar Supr
 Insertar símbolo
Propiedades especiales del objeto

Manejo y visualización...
Comunicaciones...
 Mensajes...
 Supervisión...

Propiedades de comunicación - símbolo

Usar símbolo como variable de comunicación

General | Atributos FMS | Estructura

Acceso simbólico

a toda la variable Índice EMS:

al primer nivel estructura bloque datos Índice base FMS:

 Núm. índices reservados:

Variable accesible para interlocutor:

Nombre variable:	Índice FMS	Subíndice F...	Tipo S7	Tipo FMS
datos	201		STRUCT	STRUCT
variable_1	201	1	BYTE	Unsigned8
variable_2	201	2	BYTE	Unsigned8
datos.variable_1	101		BYTE	Unsigned8
datos.variable_2	102		BYTE	Unsigned8

Programa servidor:

```
L EB 0
T "datos".variable_1 //variable que utiliza el cliente para leer

L "datos".variable_2 //variable que utiliza el cliente para escribir
T AB 14
```

Programa cliente:

Función FB3 READ de Librerías → SIMATIC_NET_CP

```
CALL FB3 , DB3
  REQ :=E0.0 //Petición de lectura
  ID :=DW#16#10130 //Identificación del enlace
  VAR_1 :="indices".indice_1 //Variable de comunicación a leer
  RD_1 :=P#A 4.0 BYTE 2 //dirección donde guardar dicha lectura
  NDR :=M100.0 //Nueva lectura
  ERROR :=M100.1 //error durante el proceso de lectura
  STATUS:=MW102 //código de estado
```