



PROFIBUS

Comunicaciones Industriales

Índice

TEMA – PROFIBUS

1. - Introducción
 - 1.1 – *Características Generales*
 - 1.2 – *Arquitectura de PROFIBUS*
2. – Nivel Físico - PHY
 - 2.1 – *Generalidades*
 - 2.2 – *Topología*
 - 2.3 – *Método de Transmisión*
3. - Nivel de Enlace - FDL
 - 3.1 – *Protocolo de Transmisión*
 - 3.4 – *Tiempos de PROFIBUS*
 - 3.4 – *Servicios FDL*
4. – Servicios FMA1/2
5. – PROFIBUS FMS
 - 5.1 – *Características FMS*
 - 5.2 – *Elementos FMS*
 - 5.3 – *Servicios FMS*
6. – Servicios FMA7
7. – PROFIBUS DP
 - 7.1 – *Características DP*
 - 7.2 – *Tipos de Dispositivos*
 - 7.3 – *Tipos de Comunicación*

Introducción



- ⇒ Definición y Estándares
- ⇒ Organización de Usuarios de PROFIBUS
- ⇒ Características Generales
- ⇒ Características Técnicas
- ⇒ PROFIBUS & OSI
- ⇒ Arquitecturas PROFIBUS
- ⇒ Niveles de aplicación

Definición y Estándares



- ✓ **PRO**cess **FI**eld **BUS**
- ✓ PROFIBUS es un **estándar** de bus de campo abierto independiente del fabricante
- ✓ Origen Alemán
- ✓ Estándar europeo **EN 50 170**, a partir de la norma alemana **DIN 19 245**
- ✓ Empleado para **interconexión de dispositivos de campo** de entrada/salida simples con PLCs y PCs
- ✓ Amplio rango de aplicaciones en automatización de fabricación, procesos y construcción

Profibus User Organization



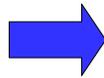
- ✓ Administrar y desarrollar la tecnología PROFIBUS
- ✓ Integrar por fabricantes, usuarios e instituciones investigadoras
- ✓ Certificar y asegurar la calidad de los productos que siguen el estándar
- ✓ Crear un catálogo con todos los productos que soportan este estándar
- ✓ Proponer PROFIBUS como estándar internacional
- ✓ Desarrollar herramientas y equipos de mantenimiento

⇒ www.profibus.com

Características Generales (I)



OBJETIVO de
un bus de campo
como **PROFIBUS**



Interconexión de
dispositivos digitales
de campo o sistemas
de bajas o medias
prestaciones



Sensores, actuadores,
transmisores, PLCs,
Controladores Numéricos,
PCs, interfaces hombre-
máquina, etc.

Características Generales (II)



¿Qué **VENTAJAS**
ofrece
PROFIBUS?

- ✓ Transmite pequeñas cantidades de datos
- ✓ Cubre necesidades de tiempo real
- ✓ Tiene gran compatibilidad electromagnética
- ✓ Número reducido de estaciones
- ✓ Fácil configuración
- ✓ Ampliación o reducción de elementos Plug & Play
- ✓ Bajos costes de conexión y cableado
- ✓ Pseudoconsistente con OSI
- ✓ Permite integrar los dispositivos menos inteligentes
- ✓ Protocolos simples y limitados

Características Generales (III)



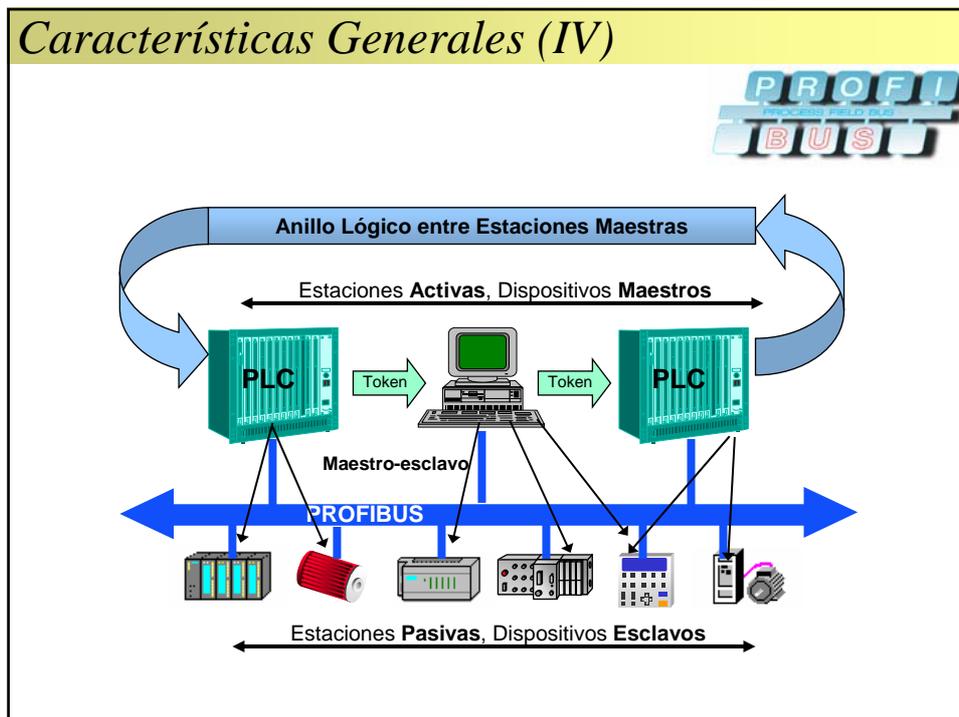
⇒ **Dos tipos de estaciones:**

- ✓ **Maestras (activas):** pueden controlar el bus y transferir mensajes sin una petición remota, si está en posesión del testigo
- ✓ **Esclavas (pasivas):** sólo pueden reconocer mensajes recibidos o transferir datos después de una petición remota

⇒ La **configuración mínima** es una de las siguientes:

- ✓ dos maestras
- ✓ una estación maestra y una esclava

Características Generales (IV)



Características Técnicas (I)

- 
- ⇒ **Topología de red:** bus lineal o en árbol con terminadores
 - ⇒ **Redundancia:** un segundo medio de transmisión es opcional
 - ⇒ **Transmisión:** halfduplex, asíncrona, sincronización por start/stop, sin bit stuffing
 - ⇒ **Acceso al bus:** **híbrido**
 - ✓ **paso de testigo** entre las estaciones **maestras**
 - ✓ **maestro/esclavo** entre **maestra** y **esclavos**
 - ⇒ **Direccionamiento:**
 - ✓ 7 bits (rango de dirección: de 0 a 127)
 - ✓ 127: dirección global para **broadcast y multicast**
 - ✓ **Extensión de las direcciones** para direcciones regionales, direcciones de segmento y direcciones puntos de acceso a servicios (**LSAP**)

Características Técnicas (II)



⇒ Servicios de transferencia:

- ✓ **Acíclicos:** Send Data with/without Acknowledge
Send and Request Data with Reply
- ✓ **Cíclicos** (polling): Cyclic Send and Request Data with Reply

⇒ Longitud de trama:

- ✓ hasta 255 bytes por trama
- ✓ de 0 a 246 octetos de datos de la capa 2 por cada Data Unit sin extensión de dirección

Características Técnicas (III)



⇒ Integridad de los datos:

- ✓ mensajes con distancia Hamming (HD) = 4, detección de deslizamiento de sincronismo, secuencia especial para evitar pérdida y multiplicación de los datos

⇒ Número de estaciones:

- ✓ 32 por segmento
- ✓ hasta 127 usando repetidores

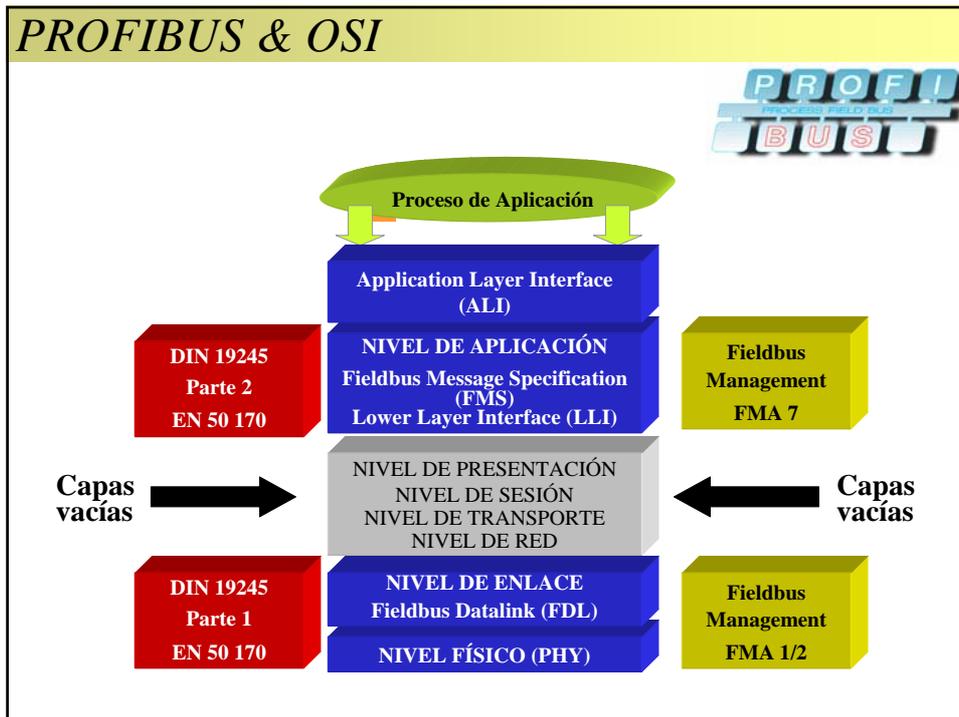
⇒ Velocidad de transmisión: depende del medio.

Según la norma RS 485, cable tipo A de par trenzado apantallado:

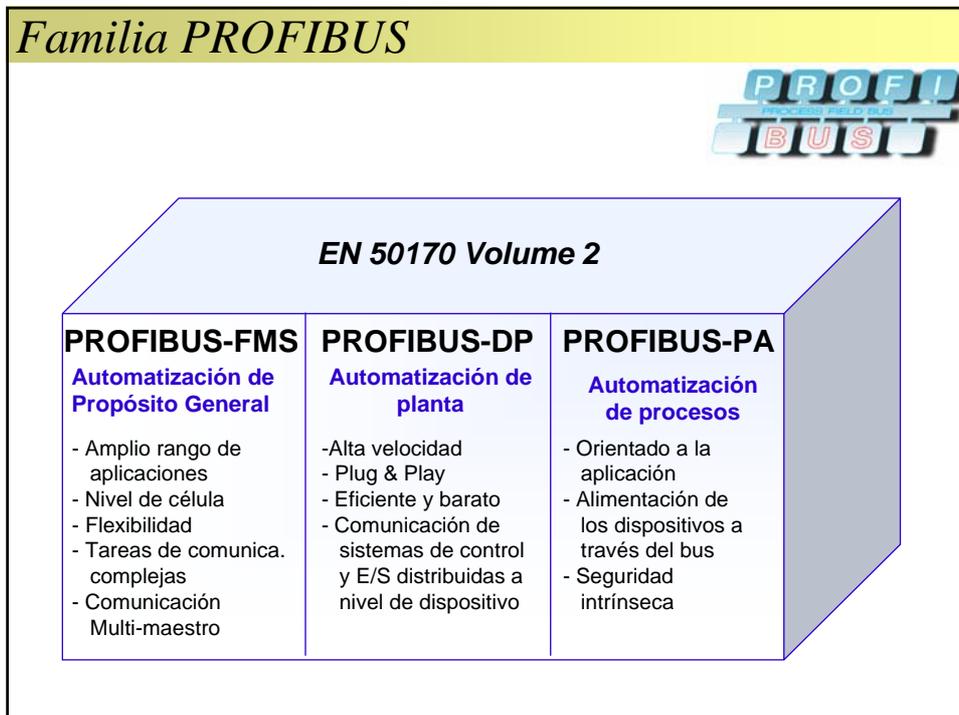
Kbits/s	9,6	19,2	93,75	187,5	500	1.500
Distanc/seg.	≤1200 m	≤1200 m	≤1200 m	≤1000 m	≤ 600m	≤ 200m

PROFIBUS-DP, posterior a la norma, puede alcanzar velocidades de hasta 12Mbit/s para distancias de 100 m

PROFIBUS & OSI



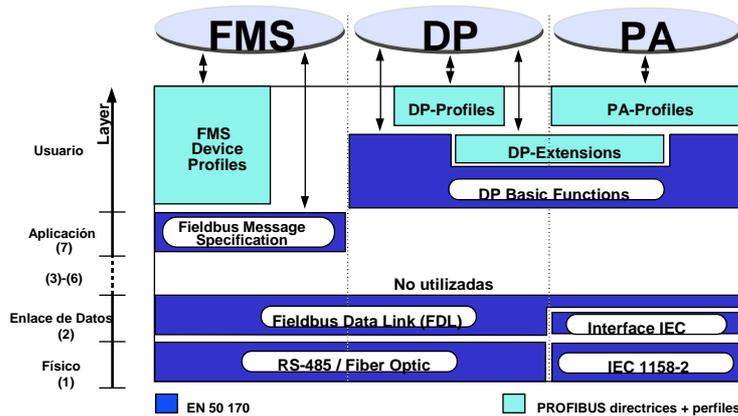
Familia PROFIBUS



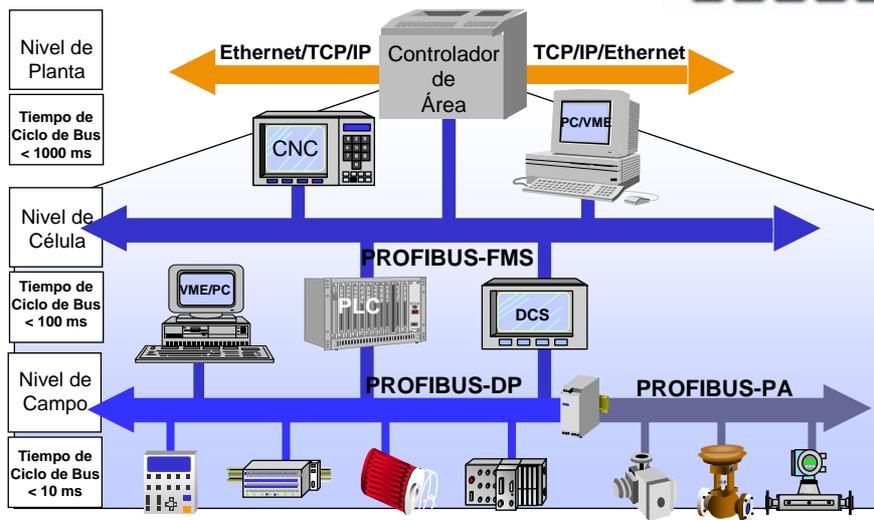
EN 50170 - 2



⇒ El Protocolo PROFIBUS es Acorde con el Modelo de Referencia para Sistemas Abiertos OSI/ISO



Niveles de Aplicación



Nivel Físico – PHY



- ⇒ GENERALIDADES
- ⇒ TOPOLOGÍA
- ⇒ CABLEADO

Generalidades



- ⇒ **Línea Serie RS-485:** línea de transmisión balanceada
- ⇒ **Topología:** Bus lineal con terminadores en ambos extremos
- ⇒ **Medio de transmisión:** Par trenzado y apantallado
- ⇒ **Longitud:** ≤ 1200 m, dependiendo de la velocidad
- ⇒ **Número de estaciones:** 32 (maestros, esclavos y repetidores)
- ⇒ **Velocidad:**
 - 9.6 / 19.2 / 93.75 kbits/s si longitud ≤ 1200 m
 - 187,5 kbit/s si longitud ≤ 1000 m
 - 500 kbit/s si longitud ≤ 600 m
 - 1500 kbit/s si longitud ≤ 200 m
 - 10/12 Mbit/s si longitud ≤ 100 m
- ⇒ **Transceiver chip:** SN75176 A, DS3695 u otros

Topología (I)

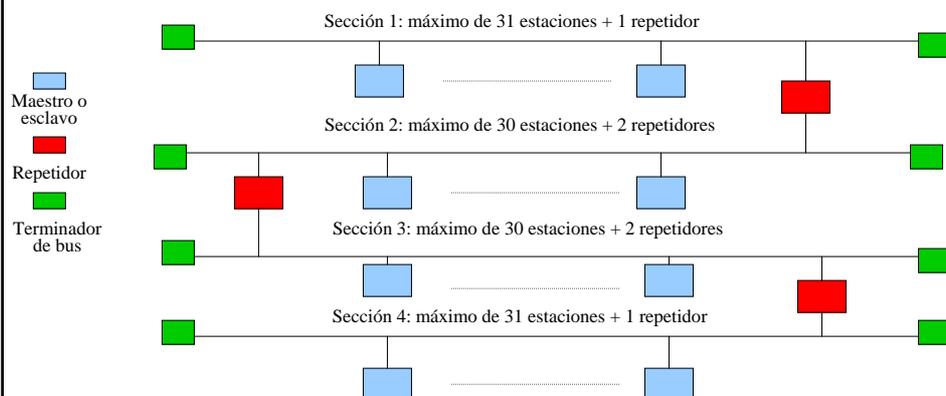


- ⇒ **Repetidores**: La longitud del bus y el nº de estaciones puede ser incrementado mediante el uso de **repetidores** (amplificadores bidireccionales)
- ⇒ **Máximo de 32 estaciones** (maestras, esclavas o repetidores) por segmento de bus
- ⇒ **2 topologías**:
 - ✓ **Lineal**, con un máximo de 3 repetidores entre 2 estaciones
 - ✓ **Árbol**, permite más de 3 repetidores

Topología (II)



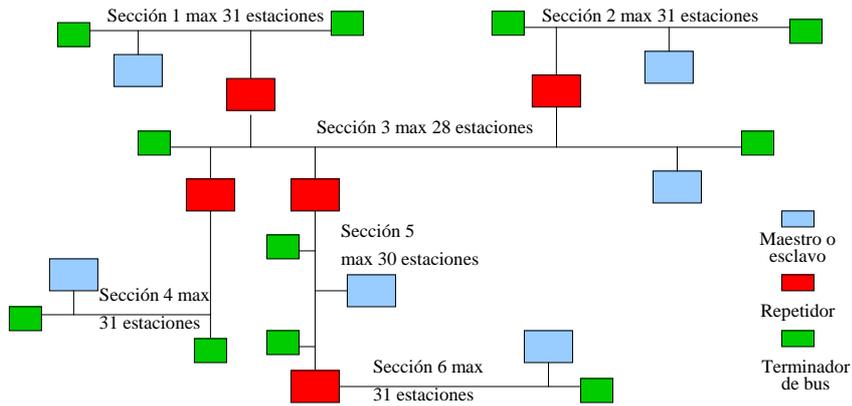
⇒ **LINEAL** (3 repetidores y 122 estaciones, config. máx.)



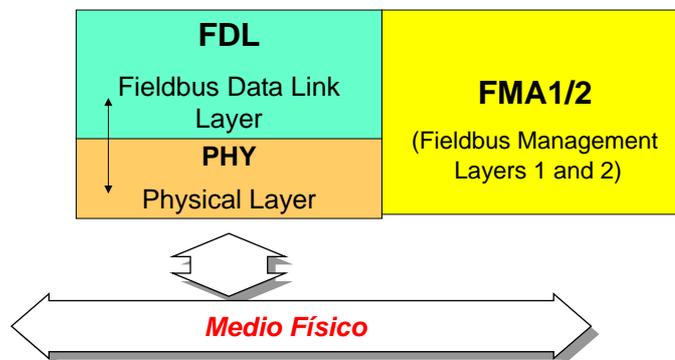
Topología (III)



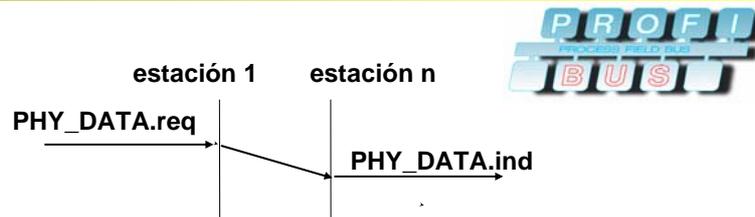
⇒ **ÁRBOL** (127 estaciones, nº máx, y 5 > 3 repetidores)



Interfaz Nivel Físico - Nivel FDL



Interacción y Primitivas



⇒ **2 primitivas:**

PHY_DATA.request (FDL_symbol)

PHY_DATA.indication (FDL_symbol)

⇒ **FDL_symbol** puede tomar los valores:

✓ **ZERO** se corresponde al valor binario "0"

✓ **ONE** se corresponde al valor binario "1"

✓ **SILENCE** deshabilitar el transmisor cuando no se envía ningún FDL symbol

Nivel de Enlace – FDL



⇒ PROTOCOLO DE TRANSMISIÓN

⇒ PASO DEL TESTIGO

⇒ TIEMPOS DE ROTACIÓN DE TESTIGO

⇒ PRIORIDADES

⇒ TIEMPOS DE PROFIBUS

⇒ TRAMAS FDL

⇒ SERVICIOS FDL

Protocolo de Transmisión (I)

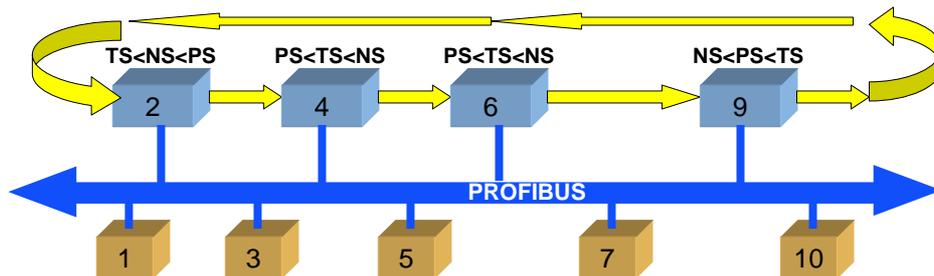


- ✓ El intercambio de mensajes tiene lugar en **ciclos**.

Un **Ciclo de Mensaje (action frame)**, consiste en el envío de una trama por una estación maestra y el reconocimiento o la respuesta, por parte de la maestra o esclava correspondiente

- ✓ Las únicas **excepciones** a este ciclo (ciclo sin reconocimiento) son:
 - × la transmisión de datos sin reconocimiento
 - × la transmisión del testigo
- ✓ Todas las estaciones, excepto la emisora **monitorizan** las peticiones y responden cuando son direccionadas

Recepción del Testigo(II)



- ✓ Cada estación guarda una **lista de estaciones activas (LAS)**. Si una maestra recibe el testigo de una estación que no está marcada en su LAS como su predecesora (PS), no lo aceptará. Si se produce un reintento de la misma PS, la estación asumirá que el anillo lógico ha cambiado y marcará la nueva maestra en la lista como su predecesora

Tiempos de Rotación de Testigo



- ✓ Una vez que una maestra recibe el testigo comienza a contar el **tiempo de rotación real del testigo (Real Rotation Time TRR)**, que acabará cuando se reciba el próximo testigo
- ✓ El tiempo máximo que debería tardar el testigo en volver es el **tiempo de rotación objetivo (Target Rotation Time TTR)**.
 - * Se define en función del número de estaciones maestras que pueda haber, la duración de los ciclos de mensajes de alta prioridad y un margen suficiente para los mensajes de baja prioridad y los posibles reintentos (se parametrizan con este valor todas las estaciones activas)
- ✓ El **tiempo de mantenimiento de testigo (Token Holding Time TTH)** es el tiempo que dispone la maestra para sus tareas, en ese ciclo y es la diferencia de tiempo entre el TTR y el TRR :

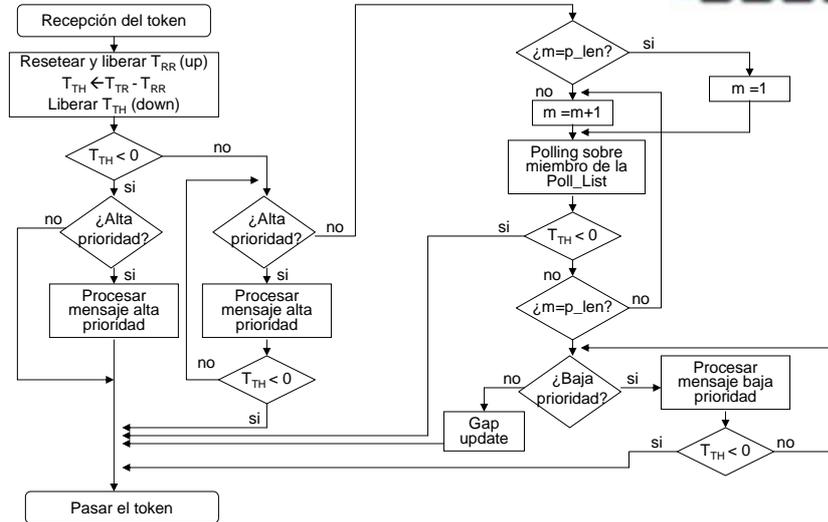
$$T_{TH} = T_{TR} - T_{RR}$$

Prioridades



- ✓ El usuario de la capa FDL (la capa de aplicación) puede elegir para los mensajes **dos prioridades: baja o alta**
- ✓ Cuando una maestra recibe el testigo siempre procesa primero los mensajes de alta prioridad y luego los de baja prioridad
 - * Una vez que un ciclo de mensaje, sea de alta o de baja prioridad, es iniciado, siempre debe concluirse, incluyendo los reintentos si son necesarios
 - * Independientemente del TRR, por cada recepción del testigo, cada maestra debe ejecutar al menos un ciclo de alta prioridad, incluyendo los reintentos en caso de error
- ✓ La prolongación del T_{TH} provocada por algunos de estos casos dará lugar al acortamiento del tiempo para los ciclos de mensajes en la próxima recepción del testigo

Procedimientos de Manejo de Mensajes



Tiempos de PROFIBUS



⇒ OBJETIVOS DE LOS TIEMPOS

- ✓ **Monitorizar** la actividad del bus para determinar errores
- ✓ **Sincronizar** las diferentes estaciones

⇒ DEFINICIÓN DE TIEMPOS

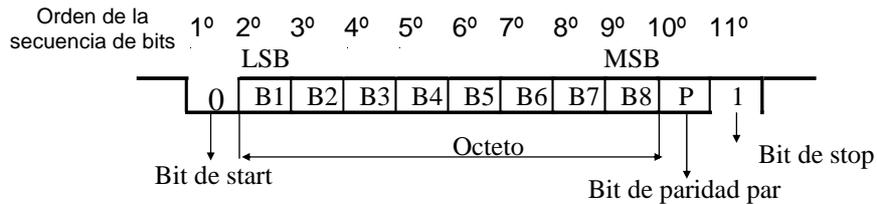
- ✓ **Bit Time:** t_{BIT} (Tiempo de Bit)
- ✓ **Idle Time:** T_{ID} (Tiempo de Estado Ausente o Idle)
- ✓ **Slot Time:** T_{SL}
- ✓ **Time-out:** T_{OUT}
- ✓ **GAP Update Time:** T_{GUD} (Actualización del GAP)
- ✓ **Real Rotation Time:** T_{RR} (Tiempo de Rotación Real)
- ✓ **Target Rotation Time:** T_{TR} (Tiempo de Rotación Objetivo)

Tramas FDL – Carácter de Trama



- ✓ Cada trama consta de un número de **caracteres de trama**, caracteres UART (UC).

Es un carácter start-stop para **transmisión asíncrona**



- ✓ La **sincronización** de bit del receptor siempre empieza con el flanco descendente del **bit de start**.
- ✓ Todos los bits se muestrean en la mitad del tiempo de bit

Tramas FDL – LSAP



⇒ LSAP Link Service Access Point

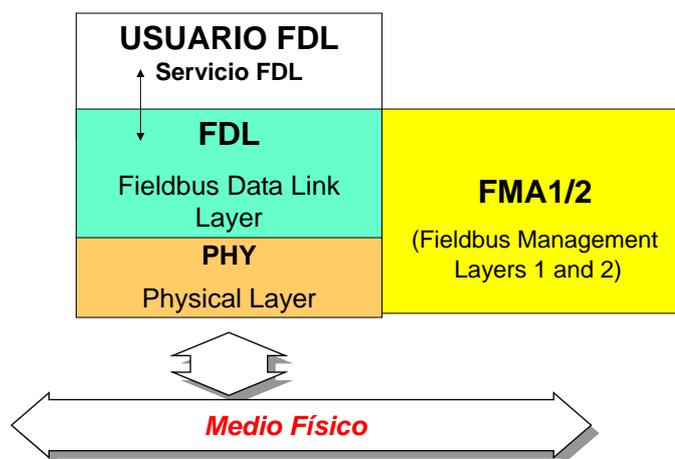
- ✓ En el interfaz Usuario FDL - Nivel FDL los servicios se procesan a través de un LSAP.
- ✓ Puede haber varios LSAP activos a la vez en maestros o esclavos
 - ⇒ Deben transmitirse los LSAP con el mensaje.
 - ✗ **SSAP (Source Service Access Point)** es el LSAP del usuario FDL local y se envía dentro del SAE. Valores de 0 a 62.
 - ✗ **DSAP (Destination Service Access Point)** es el LSAP del usuario FDL remoto y se envía dentro del DAE. Valores de 0 a 63.
 - DSAP = 63 ⇒ dirección de acceso global (sólo en SDA y SDN)
- ✓ En ocasiones, por razones de eficiencia, no se envían los LSAP y todo es procesado en el LSAP por defecto.

Servicios FDL



- ⇒ Interfaz Usuario FDL - Nivel FDL
- ⇒ Introducción
- ⇒ Interacción de primitivas
- ⇒ Parámetros de primitivas
- ⇒ SDA
- ⇒ SDN
- ⇒ SDR
- ⇒ CSRD

Interfaz Usuario FDL - Nivel FDL

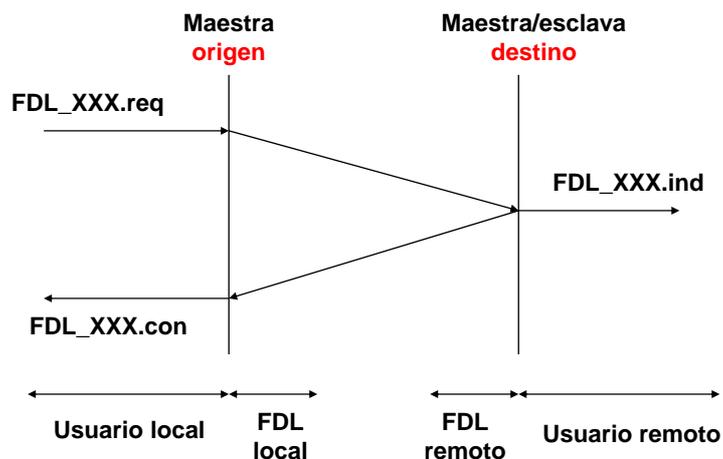


Introducción a FDL



- ✓ Los **servicios de producción** (transferencia de datos) son accesibles a través de la capa 2 (FDL), y los **servicios de gestión**, a través de la capa FMA1/2 asociada a las capas 1 y 2
- ✓ Los servicios se llevan a cabo usando una serie de primitivas (con sus correspondientes parámetros) denotadas por **FDL_XXXX.request**. Tras completar el servicio se devuelve al usuario una primitiva **FDL_XXXX.confirm**. Si ocurre algo inesperado en la estación remota se le indica al usuario por **FDL_XXXX.indication**
- ✓ Hay cuatro servicios de transferencia de datos:
 - × **SDA** Send Data with Acknowledge
 - × **SDN** Send Data with No Acknowledge
 - × **SRD** Send and Request Data with Reply
 - × **CSRD** Cyclic Send and Request Data with Reply

Interacción de Primitivas



Parámetros de Primitivas (I)



- ✓ **SSAP** ⇒ LSAP del usuario local
- ✓ **DSAP** ⇒ LSAP del usuario remoto
 - × Si por razones de eficiencia no se emplean LSAP, los campos anteriores toman el valor NIL y se trabaja con el LSAP por defecto
- ✓ **Rem_add (Remote_address)** ⇒ Dirección FDL de la estación remota
- ✓ **Loc_add (Local_address)** ⇒ Dirección FDL de la estación local
- ✓ **L_sdu (Link_service_data_unit)** ⇒ contiene los datos de usuario que van a ser transferidos por el controlador FDL.

Parámetros de primitivas (II)



- ✓ **Serv_class (Service_class)** ⇒ define la prioridad de los datos
 - × **High priority (high)**: mensajes temporales críticos, como alarmas y datos de sincronización y coordinación
 - × **Low priority (low)**: datos menos urgentes; datos de proceso, diagnóstico o programa
- ✓ **L_status (Link_status)** ⇒ indica el éxito o fracaso de una primitiva de request o si existen o no errores temporales o permanentes.
- ✓ **Update_status** ⇒ especifica si los datos han sido pasados o no al controlador FDL
- ✓ **Transmit** ⇒ indica si la actualización (Update) se transmite una vez o varias
- ✓ **Poll_list**

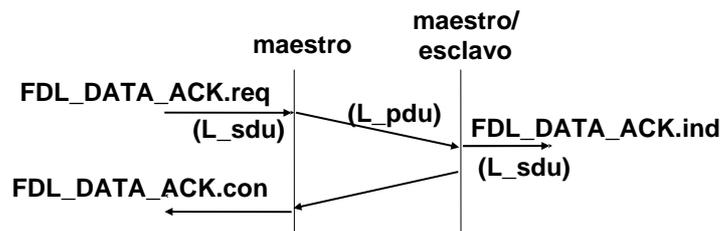
SDA



⇒ SDA (Send Data with Acknowledgement)

- ✓ Permite al usuario en una estación maestra enviar datos a una única estación remota y recibir inmediatamente la confirmación
- ✓ En la estación remota se entrega el `L_sdu` al usuario remoto, si se ha recibido sin errores
- ✓ Al usuario local la capa FDL le envía confirmación de la recepción de los datos, si ha recibido reconocimiento de la estación remota. Si dentro del Slot Time no se recibe la confirmación, se realizará el número indicado de reintentos, tras los cuales se remitirá un reconocimiento negativo al usuario local
- ✓ Entre la transferencia de los datos y la recepción de su reconocimiento ningún otro tráfico tiene lugar en el bus

SDA (II)



3 primitivas

`FDL_DATA_ACK.req(SSAP, DSAP, Rem_Add, L_Sdu, Serv_Class)`

`FDL_DATA_ACK.ind(SSAP, DSAP, Loc_Add, Rem_Add, L_Sdu, Serv_Class)`

`FDL_DATA_ACK.con(SSAP, DSAP, Rem_Add, Serv_Class, L_Status)`

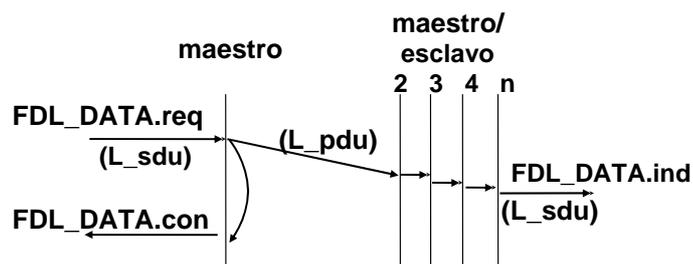
SDN



⇒ SDN (Send Data with No Acknowledgement)

- ✓ Además de enviar datos a una única estación, permite enviar a todas las estaciones (**broadcast**) o a un grupo de ellas (**multicast**)
- ✓ Para estos dos casos los bits 1 a 7 de la dirección destino deben estar a 1 (**dirección global 127**). El grupo de estaciones para multicast se identifica por una dirección de LSAP
- ✓ El usuario local recibe confirmación de la transferencia de los datos, pero no de si se han recibido correctamente. No hay reintentos
- ✓ Una vez que los datos son enviados alcanzan todos los usuarios remotos a la vez, pero sólo los controladores FDL que identifiquen su propia dirección y que hayan recibido sin error lo pasarán a su usuario remoto

SDN (II)



3 primitivas

FDL_DATA.req(SSAP, DSAP, Rem_Add, L_Sdu, Serv_Class)

FDL_DATA.ind(SSAP, DSAP, Loc_Add, Rem_Add, L_Sdu, Serv_Class)

FDL_DATA.con(SSAP, DSAP, Rem_Add, Serv_Class, L_Status)

SRD (I)



⇒ SRD (Send and Request Data with Reply)

- ✓ Permite transferir datos a una única estación remota y al mismo tiempo solicitar datos que el usuario remoto había dejado disponibles previamente. La transferencia de datos, en este caso, es opcional
- ✓ Tan pronto como se recibe la trama sin error, se transmiten los datos solicitados
- ✓ El usuario local recibe los datos pedidos o una indicación de que no estaban disponibles (ambos casos suponen la confirmación de la recepción de los datos transferidos) o una confirmación de la recepción de los datos transmitidos

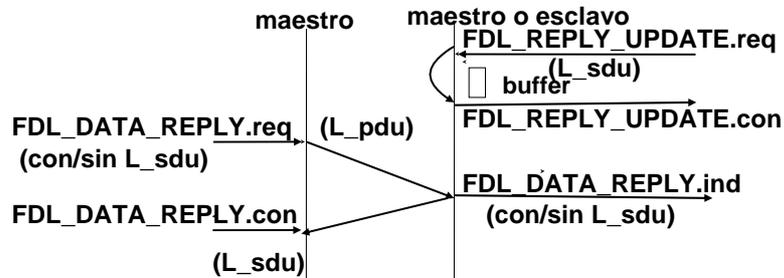
SRD (II)



⇒ SRD (Send and Request Data with Reply)

- ✓ Si ocurre un error durante la transferencia, la capa FDL del usuario local repite tanto los datos como la petición
- ✓ Entre la transmisión original y la respuesta ningún otro tráfico tiene lugar en el bus

SRD (III)



5 primitivas

FDL_DATA_REPLY.req(SSAP, DSAP, Rem_Add, L_Sdu, Serv_Class)

FDL_DATA_REPLY.ind(SSAP, DSAP, Loc_Add, Rem_Add, L_Sdu, Serv_Class, Update_Status)

FDL_DATA_REPLY.con(mismos parámetros que req + L_Status)

FDL_REPLY_UPDATE.req(SSAP, L_Sdu, Serv_Class, Transmit)

FDL_REPLY_UPDATE.con(SSAP, Serv_Class, L_Status)

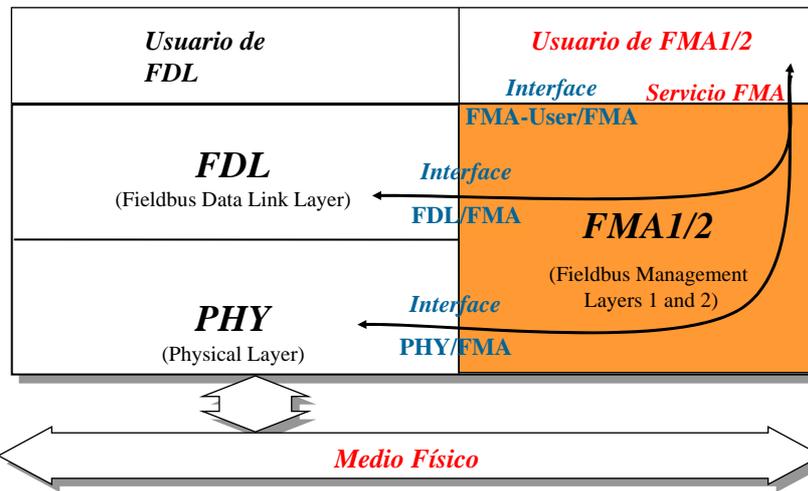
CSRD (I)



⇒ CSRD (Cyclic Send and Request Data with Reply)

- ✓ La funcionalidad de este servicio es la misma que la del SRD
- ✓ Las acciones se realizan cíclicamente con cada estación de las incluidas en la lista **Poll List** definida por el usuario local.
 - ✗ En la lista también se deben indicar el número y la secuencia de las transferencias y peticiones.
 - ✗ En la lista pueden aparecer repetidas las direcciones FDL, para permitir ordenar las estaciones convenientemente
- ✓ Después de cada transferencia y petición se espera una respuesta o confirmación inmediata
- ✓ Durante un CSRD se permiten los servicios acíclicos SDA, SDN y SRD

Servicios FMA1/2



Servicios de la Capa FMA1/2



- ⇒ Se llevan a cabo por medio de primitivas del tipo **FMA1/2_XXXX.xxxx**
- ⇒ Los servicios pueden ser **opcionales (optional)** u **obligatorios (mandatory)**, y referirse a las capas **locales o remotas**
- ⇒ **Servicios FMA1/2:**
 - ✓ **Reset FMA1/2** (l, m): resetea (equivalente al power on) las capas PHY, FDL Y FMA1/2. El usuario recibe confirmación
 - ✓ **Set Value FMA1/2** (l, o): permite asignar nuevos valores a las variables de las capas 1 y 2. El usuario recibe confirmación
 - ✓ **Read Value FMA1/2** (l, o): permite leer variables de las capas 1 y 2. La respuesta contiene los valores de las variables especificadas
 - ✓ **Event FMA1/2** (l y r, m): la capa informa al usuario de sucesos y errores en las capas 1 y 2

Servicios de la Capa FMA1/2



⇒ Servicios FMA1/2:

- ✓ **Ident FMA1/2** (l y r, o): en una esclava identifica la versión software y hardware de las capas FDL y FMA1/2. En una maestra puede hacer lo mismo con respecto a una estación remota
- ✓ **LSAP Status FMA1/2** (l y r, o): permite obtener información sobre los LSAP locales y remotos (en estación pasiva, sólo de los locales)
- ✓ **Live List FMA1/2** (r, o): proporciona al usuario en la estación maestra una lista actualizada de las estaciones funcionales en el bus
- ✓ **SAP Activate FMA1/2** (l, o): permite habilitar y activar un LSAP local. El usuario recibe confirmación
- ✓ **RSAP Activate FMA1/2** (l, o): realiza la misma función que el anterior pero para los servicios con respuesta (SRD y CSRD)
- ✓ **SAP Deactivate FMA1/2** (l, o): permite desactivar un LSAP local. El usuario recibe confirmación

PROFIBUS DP



- ⇒ CARACTERISTICAS
- ⇒ ACCESO AL MEDIO
- ⇒ TIPOS DE DISPOSITIVOS
- ⇒ TIPOS DE COMUNICACION
- ⇒ DESCRIPCION DE DISPOSITIVOS
- ⇒ DESARROLLO DE APLICACIONES

Características PROFIBUS-DP



Requerimientos	Características
<ul style="list-style-type: none">• Tiempo de reacción corto• Operación Monomaestro o Multimaestro• Protocolo simple, con interfaz de comunicación de bajo coste• Excelente diagnóstico• Interfaz de usuario simple• Uso del cableado existente	<ul style="list-style-type: none">• Cambio de más de 1000 Entradas y Salidas con 32 dispositivos en menos de 10 ms.• Método de acceso híbrido• Funcionalidad reducida, solución con ASIC sin microprocesador• Varios diagnósticos en maestro y esclavo• Conjunto básico de parámetros y datos de configuración• Misma tecnología de transmisión en todas las aplicaciones

Tipos de Dispositivos



⇒ Maestro Clase 1

- ✓ controlador central que intercambia datos con los dispositivos de I/O conectados (esclavos)
- ✓ determina la velocidad
- ✓ maneja el Testigo (Token)
- ✓ son permitidos varios maestros de clase 1 en una configuración
- ✓ los dispositivos típicos maestros DP clase 1 son PLC, PC

⇒ Maestro Clase 2

- ✓ herramienta de diagnosis y arranque, normalmente herramienta de configuración
- ✓ también puede controlar esclavos

⇒ Estación Esclava

- ✓ estación pasiva que reconoce mensajes o contesta a peticiones

Tipos de Comunicación



⇒ Dos tipos de comunicación

- ✓ uno a uno
- ✓ uno a muchos (Multicast)

⇒ Pueden darse comunicaciones

- ✓ entre M-DP (clase 1 o 2) y E-DP
- ✓ entre M-DP (clase 2) y M-DP (clase 1)

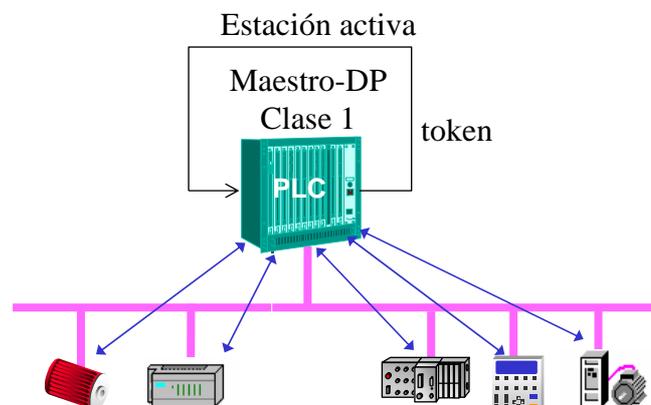
⇒ Iniciadores de comunicación

- ✓ el iniciador de comunicación maestro-esclavo es siempre el M-DP
- ✓ el iniciador de la comunicación M-M es siempre el M-DP clase 2
- ✓ no está definida la comunicación M-DPs de la misma clase

Acceso al Medio (I)

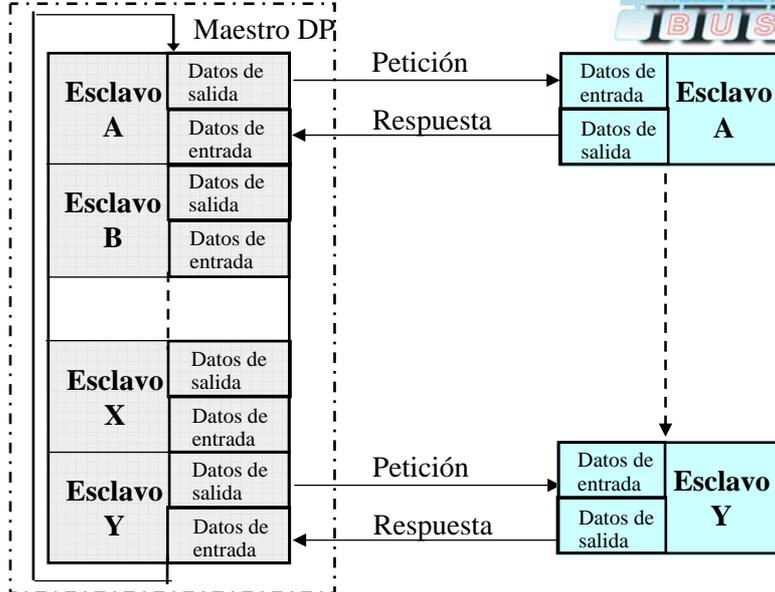


⇒ Protocolo de Transmisión y Acceso al Medio Sistema Monomaestro



Acceso al Medio (II)

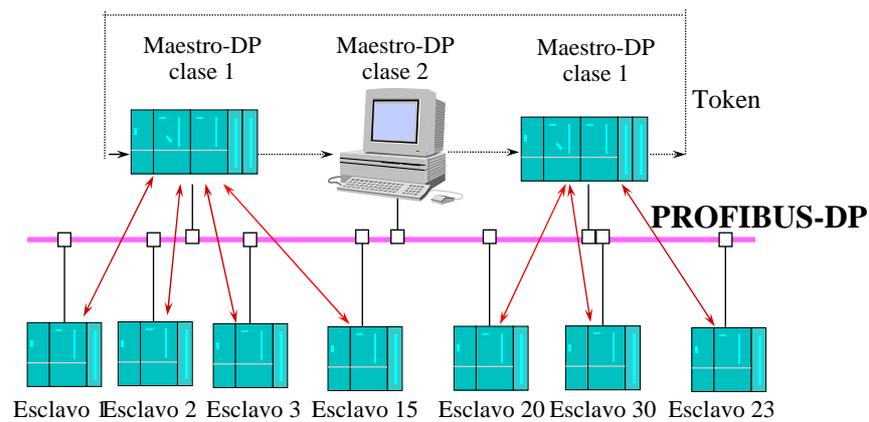
⇒ Procesado Lista de Sondeo



Acceso al Medio (III)

⇒ Protocolo de Transmisión y Acceso al Medio

Sistema Multimaestro



Tipos de Dispositivos (I)



⇒ Maestros DP - Clase 1

- ✓ El maestro DP clase 1 intercambia datos con los esclavos DP cíclicamente. Ejecuta las tareas usando las siguientes funciones del protocolo:
 - * **Set_Prm and Chk_Cfg.** Para la fase de arranque. Para transmitir parámetros a los esclavos-DP. El número de bytes de datos de E/S con un esclavo DP se define durante la configuración
 - * **Data_Exchange.** Realiza el intercambio cíclico de datos de E/S con el esclavo DP asignado
 - * **Slave_Diag.** Lee información de diagnóstico del esclavo DP durante el arranque o durante el intercambio cíclico de datos
 - * **Global_Control.** El maestro DP usa comandos de control para informar a los esclavos DP de estados de operación. Los comandos de control pueden enviarse a un esclavo individual o a un grupo específico de esclavos DP.

Tipos de Dispositivos (II)



⇒ Maestros DP - Clase 2

- ✓ Los maestros DP clase 2 son dispositivos como, unidades de programación y dispositivos diagnóstico y mantenimiento del bus. Además de las funciones descritas para los maestros clase 1 soportan las siguientes funciones:
 - * **RD_Inp and RD_Outp.** Permite leer datos de E/S de los esclavos DP
 - * **Get_Cfg.** Permite leer los datos de configuración actuales de un esclavo DP
 - * **Set_Slave_Add.** Permite al maestro DP asignar una nueva dirección a un esclavo DP, siempre que el esclavo soporte este método de fijar la dirección
 - * Adicionalmente, los maestros DP clase 2 soportan funciones para la comunicación con maestros DP clase 1

Tipos de Dispositivos (III)



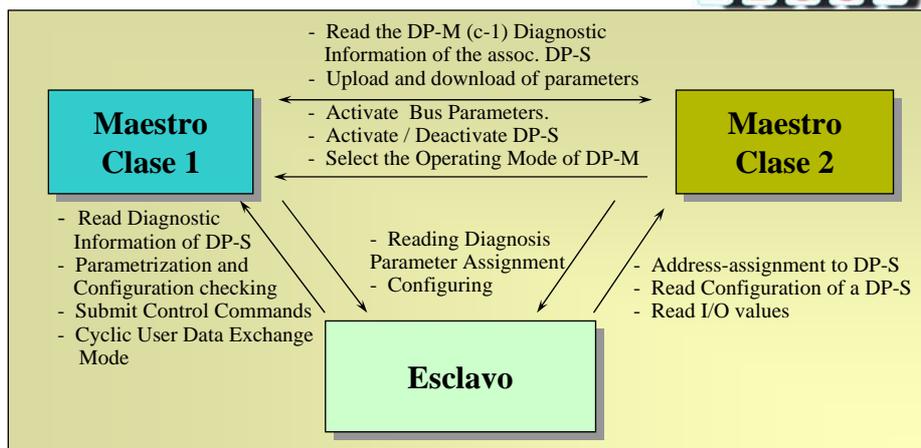
⇒ Esclavos DP

- ✓ Un esclavo DP sólo intercambia datos de usuario con el maestro DP cuando el maestro DP ha cargado los parámetros y la configuración. Un esclavo DP tiene permitido interrumpir la comunicación para procesar información de diagnóstico local e interrumpir al proceso en el maestro DP.

Funciones PROFIBUS-DP (IV)



⇒ Comunicación Maestro-Maestro, Maestro-Esclavo

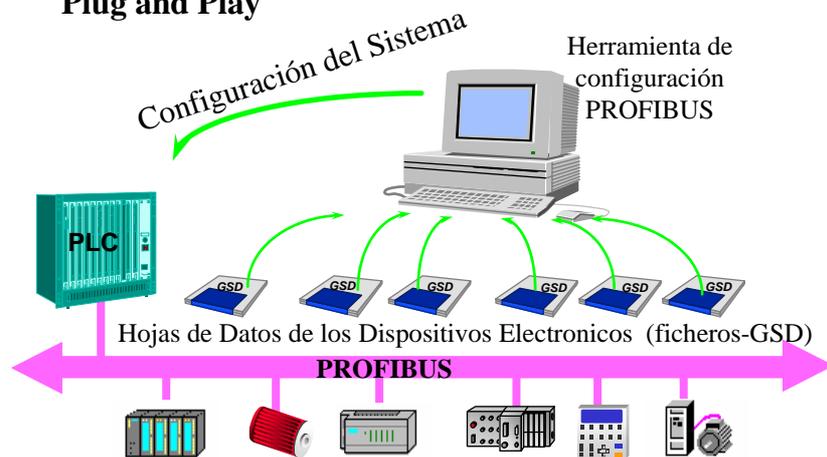


- ✓ Un dispositivo puede soportar múltiples funciones, clase 1 y clase 2, clase 1 y esclavo

Descripción del Dispositivo

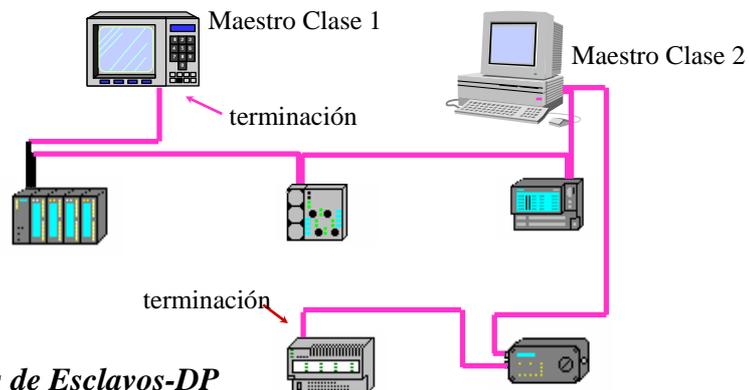
⇒ Permite una Configuración Abierta

Plug and Play



Desarrollo de Aplicaciones

⇒ Tipos de Dispositivos en una Aplicación

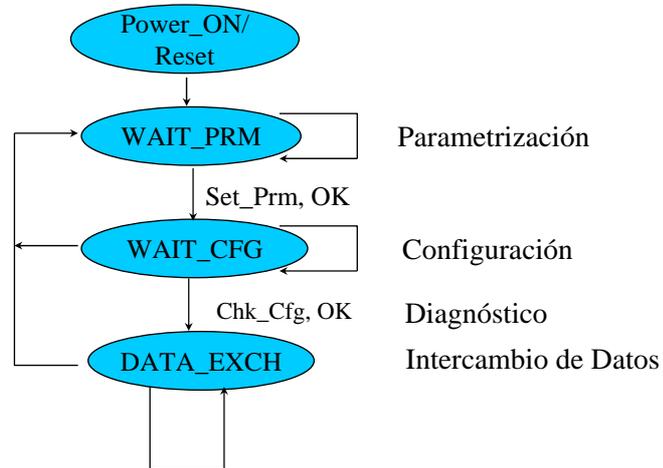


Tipos de Esclavos-DP

- ✓ Compactos
- ✓ Modulares
- ✓ Inteligentes

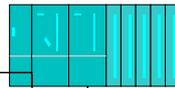
Interacciones Maestro-Esclavo (I)

- Máquina de estados del esclavo



Interacciones Maestro-Esclavo (II)

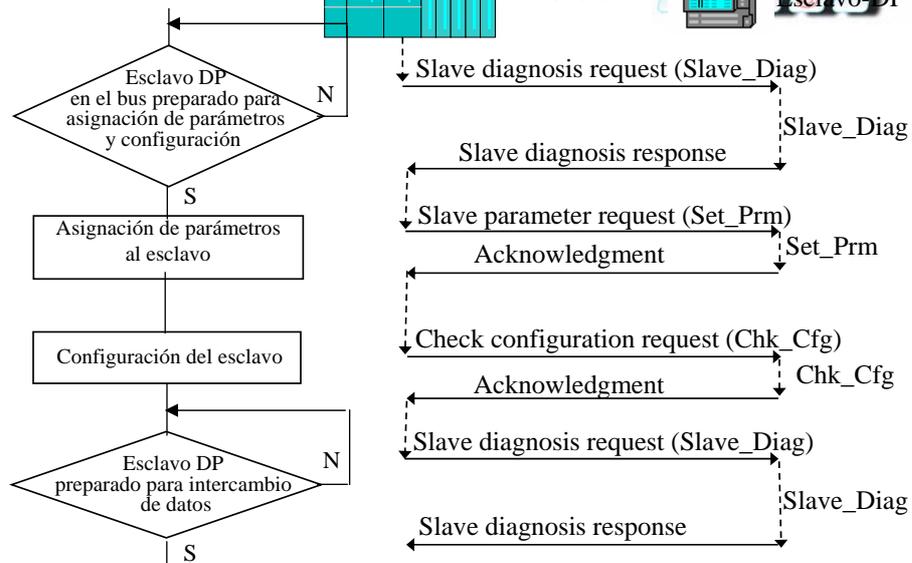
Fase de inicialización



Maestro-DP

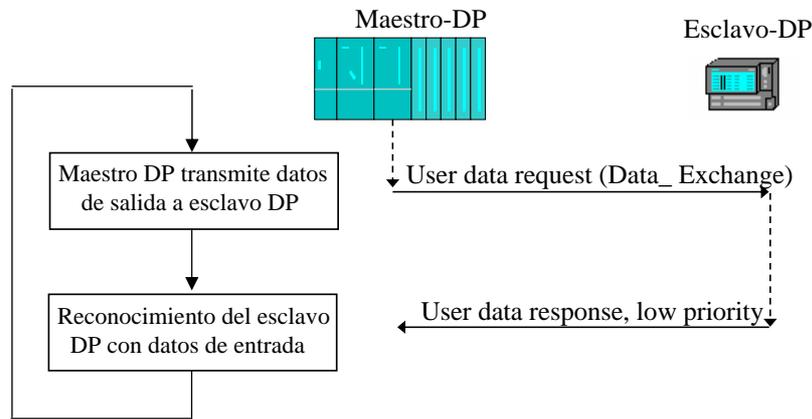


Esclavo-DP



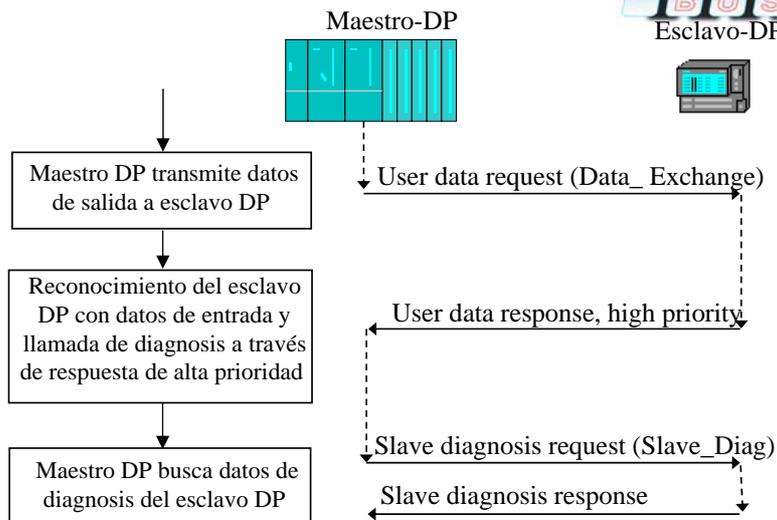
Interacciones Maestro-Esclavo (III)

Intercambio de datos (Data Exchange) I



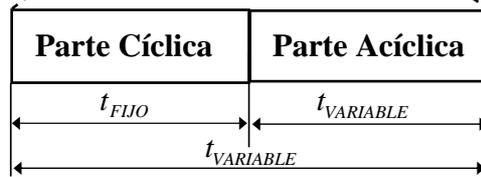
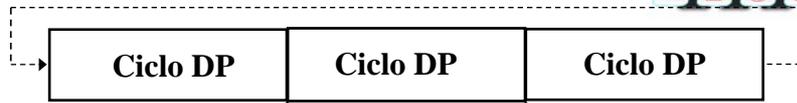
Interacciones Maestro-Esclavo (IV)

Intercambio de datos (Data Exchange) II



PROFIBUS-DP – V2 (I)

Ciclo Profibus-DP variable

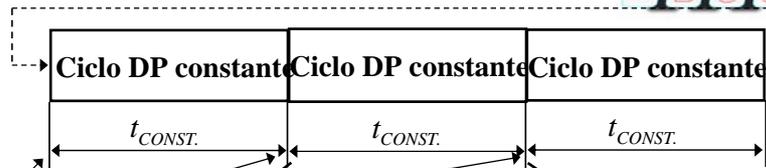


- Intercambio de datos de E/S

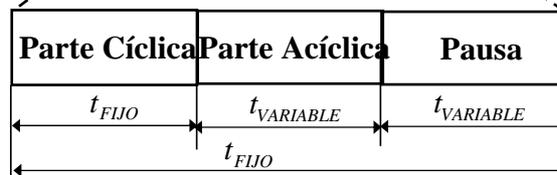
- Servicios de manejo del bus
- Inicialización de los esclavos DP
- Funciones de diagnóstico y alarmas
- Lectura y escritura de datos no cíclicos
- Comunicación PG/TD/OP
- Repetición de telegramas en caso de fallos

PROFIBUS-DP – V2 (II)

Ciclo Profibus-DP constante

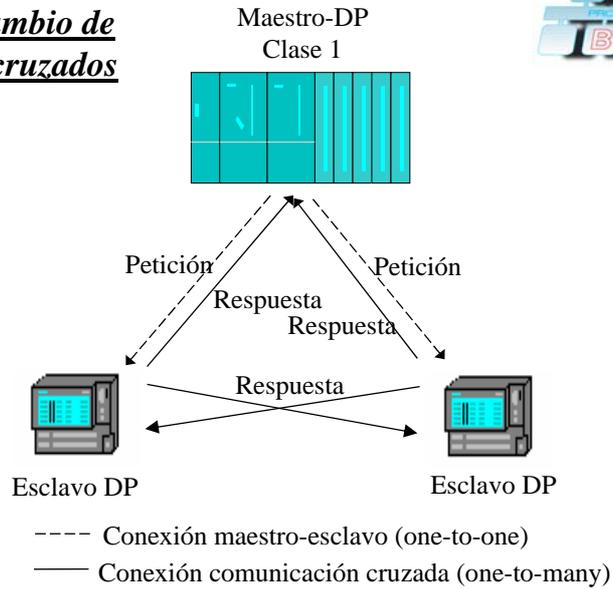


Comienzo de ciclo DP constante



PROFIBUS-DP – V2 (III)

Intercambio de datos cruzados



PROFIBUS-DP – V2 (IV)

Sincronización



Sync_Mode_supp=1 sincronización de datos de salida
p.e. arrancar accionamientos en el mismo momento

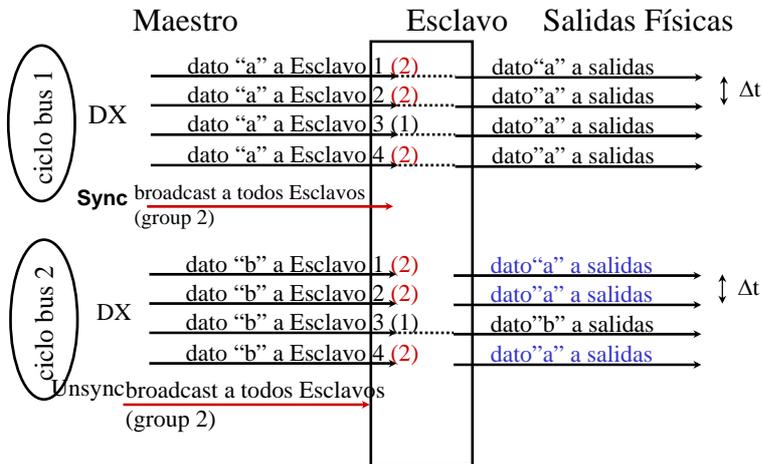
Freeze_Mode_supp=1 sincronización de datos de entrada
p.e. lanzar foto de agrupación de entradas

- ✓ Sync y Freeze son dirigidos desde la aplicación
- ✓ El interfaz a la aplicación debe ofrecer la capacidad al usuario para activar estos comandos bajo demanda
- ✓ Los comandos Sync y Freeze están relacionados con grupos

PROFIBUS-DP – V2 (V)



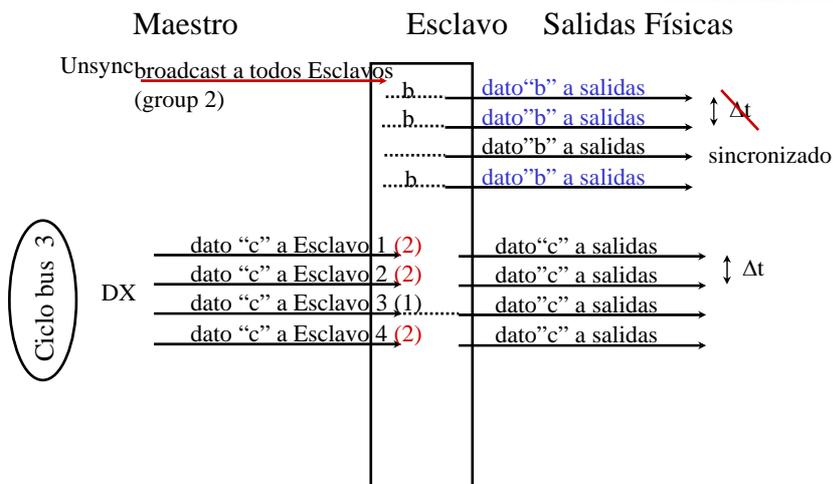
Sync



PROFIBUS-DP – V2 (VI)



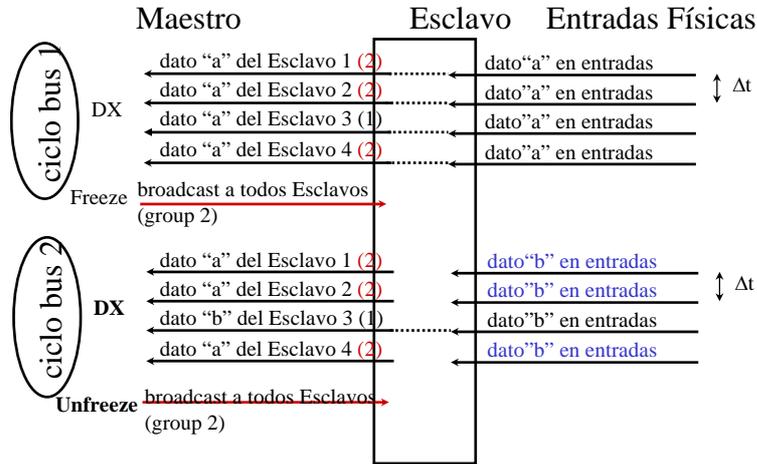
Sync



PROFIBUS-DP – V2 (VII)



Freeze



PROFIBUS-DP – V2 (VIII)



Freeze

