



Programación PROFIBUS

Comunicaciones Industriales

Índice

TEMA – Programación PROFIBUS

1. - Introducción

- 1.1 – Características Generales
- 1.2 – Arquitectura de PROFIBUS

2. – Programación FDL

- 2.1 – Arquitectura de Programación
- 2.2 – Estructura del Nivel 2
- 2.3 – Primitivas
- 2.4 – Mecanismos de Intercambio
- 2.5 – Servicios FDL
- 2.6 – Estructuras de Datos
- 2.7 – Estructura de Aplicación
- 2.8 – Funciones de Librería

3. - Programación DP

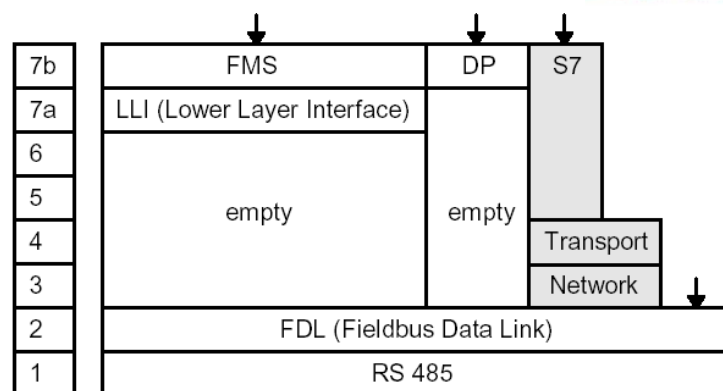
- 3.1 – Arquitectura Básica
- 3.2 – Modos de Funcionamiento
- 3.3 – Tramas de Control
- 3.4 – Programación del Interfaz DP
- 3.5 – Estructuras de Datos
- 3.6 – Estructura de Aplicación
- 3.7 – Funciones de Librería

Introducción



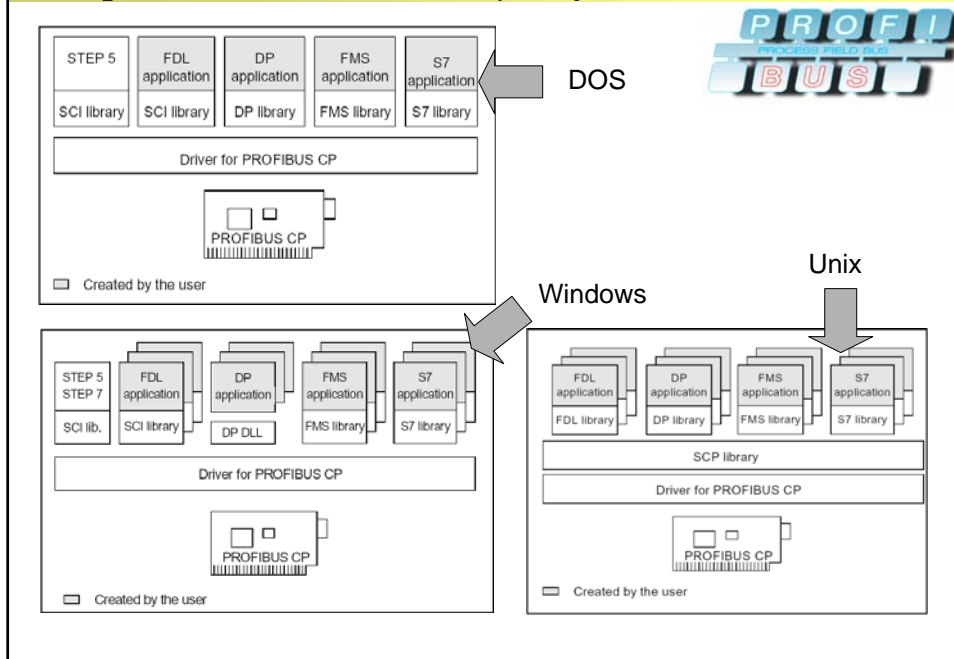
- ⇒ Arquitectura de Programación
- ⇒ Componentes Hardware y Software

Arquitectura de Programación



- ▭ homogeneous expansions
- ↓ user interfaces

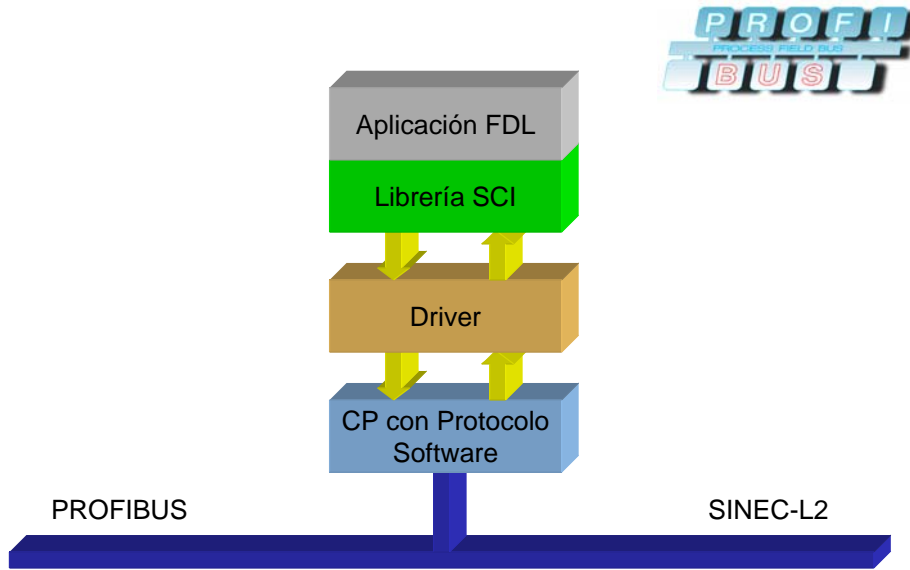
Componentes Hardware y Software



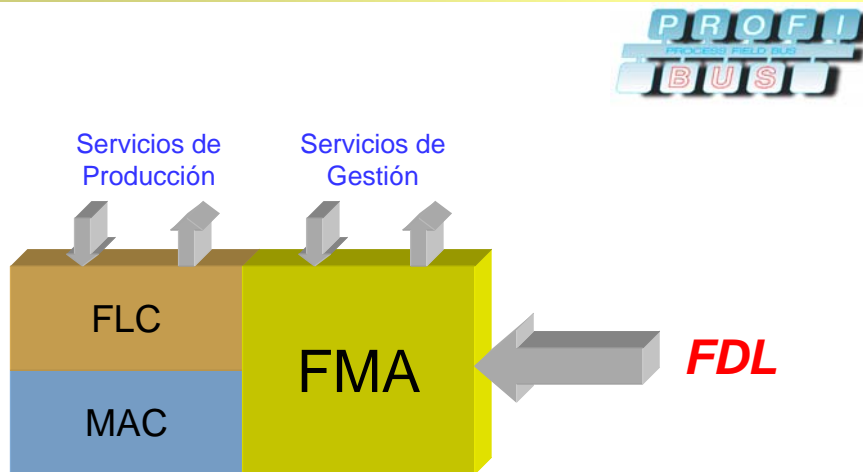
PROFIBUS FDL

- ⇒ Arquitectura de Programación
- ⇒ Estructura del Nivel 2
- ⇒ Primitivas
- ⇒ Mecanismo de Intercambio
- ⇒ Servicios FDL
- ⇒ Estructuras de Datos
- ⇒ Estructura de Aplicación
- ⇒ Funciones de Librería

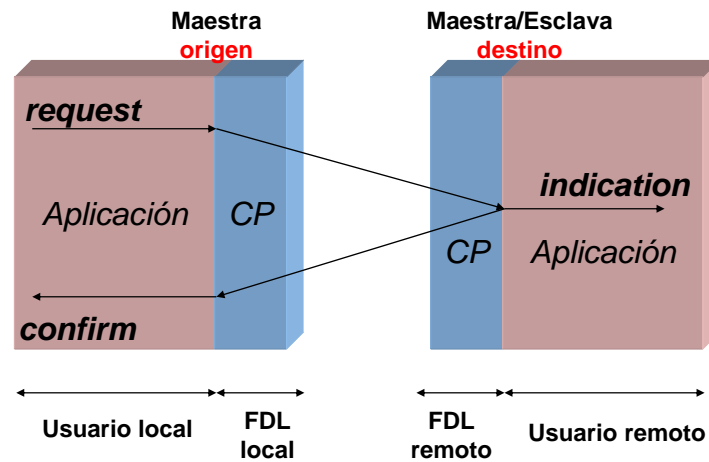
Arquitectura de Programación



Estructura del Nivel 2



Interacción de Primitivas



Mecanismo del Intercambio



⇒ El mecanismo de intercambio de intercambio de entre la aplicación FDL y la CP es el **request block**

| Request Block ID | Tarea del request block | Mecanismo |
|-------------------|----------------------------------------------------------------|-----------|
| request | Trabaja desde la aplicación FDL hacia la CP | Write |
| confirm | Reconocimiento del request desde la CP hacia la aplicación FDL | Read |
| indication | Indicación de un evento desde la CP hacia la aplicación FDL | Read |

Servicios FDL



⇒ Servicios de Producción

- ✓ SDA
- ✓ SDN
- ✓ SRD
- ✓ REPLY_UPDATE_SINGLE
- ✓ REPLY_UPDATE_MULTIPLE

⇒ Servicios de Gestión

- ✓ SAP_ACTIVATE
- ✓ RSAP_ACTIVATE
- ✓ SAP_DEACTIVATE
- ✓ AWAIT_INDICATION
- ✓ WITHDRAW_INDICATION
- ✓ LSAP_STATUS
- ✓ FDL_IDENT
- ✓ FDL_LIFE_LIST_CREATE_LOCAL
- ✓ FDL_LIFE_LIST_CREATE_REMOTE
- ✓ FDL_READ_STATISTIC_CRT
- ✓ FDL_READ_LAS_STATISTIC_CRT
- ✓ FDL_EVENT
- ✓ FDL_READ_VALUE

Mecanismo de Transferencia



⇒ **Requester:** estación que lanza el trabajo de procesado y espera a recibir la confirmación

⇒ **Responder:** estación que recibe la trama de datos desde la estación remota y devuelve una trama de reconocimiento

| Service | Requester | | Responder |
|-------------------------------------|-----------|---------------------------|------------|
| | Request | Confirmation | Indication |
| SDA (Send data with acknowledge) | yes | yes | yes |
| SDN (Send data with no acknowledge) | yes | yes | yes |
| SRD (Send and request data) | yes | yes | yes |
| REPLY_UPDATE_SINGLE | yes | yes | no |
| REPLY_UPDATE_MULTIPLE | yes | yes | no |
| SAP_ACTIVATE | yes | yes | no |
| RSAP_ACTIVATE | yes | yes | no |
| SAP_DEACTIVATE | yes | yes | no |
| AWAIT_INDICATION | yes | Success: no Error: yes | no |
| WITHDRAW_INDICATION | yes | yes | no |
| LSAP_STATUS | yes | yes | no |
| FDL_IDENT | yes | yes | no |
| FDL_LIFE_LIST_CREATE_REMOTE | yes | yes | no |
| FDL_LIFE_LIST_CREATE_LOCAL | yes | yes | no |
| FDL_READ_STATISTIC_COUNTER | yes | yes | no |
| FDL_EVENT | no | no | yes |
| FDL_READ_VALUE | yes | yes | no □ |

Estructuras de Datos de los Servicios de Producción



```
typedef struct
{
    rb2_header_type    rb2_header;
    struct application_block application_block;
    UBYTE reserved [12];
    UBYTE reference [2];
    UBYTE user_data_1 [260];
    UBYTE user_data_2 [260];
} fdi_rb;
```

```
typedef struct
{
    UWORD reserved [2];
    UBYTE length;
    UWORD user;
    UBYTE rb_type;
    UBYTE priority;
    UBYTE reserved_1;
    UWORD reserved_2;
    UBYTE subsystem;
    UBYTE opcode;
    UWORD response;
    UWORD fill_length_1;
    UBYTE reserved_3;
    UWORD seg_length_1;
    UWORD offset_1;
    UWORD reserved_4;
    UWORD fill_length_2;
    UBYTE reserved_5;
    UWORD seg_length_2;
    UWORD offset_2;
    UWORD reserved_6;
} rb2_header_type;
```

```
struct application_block
{
    UBYTE opcode;
    UBYTE subsystem;
    UWORD id;
    struct service service;
    struct remote_address loc_add;
    UBYTE ssap;
    UBYTE dsap;
    struct remote_address rem_add;
    enum service_class serv_class;
    struct link_service_data_unit receive_l_sdu;
    UBYTE reserved_1;
    UBYTE reserved;
    struct link_service_data_unit send_l_sdu;
    enum link_status l_status;
    UWORD reserved_2 [2];
};

struct service
{
    enum service_code code;
};

struct remote_address
{
    UBYTE station;
    UBYTE segment;
};

struct link_service_data_unit
{
    void far * buffer_ptr;
    UBYTE length;
};
```

Estructuras de Datos de los Servicios de Producción



| request | sda | sdn | srd | reply_update |
|----------------------|-----|-----|-----|--------------|
| length | m | m | m | m |
| user | x | x | x | x |
| rb_type | m | m | m | m |
| priority | m | m | m | m |
| subsystem | m | m | m | m |
| opcode | m | m | m | m |
| response | x | x | x | x |
| fill_length_1 | m | m | m | m |
| seg_length_1 | m | m | m | m |
| offset_1 | m | m | m | m |
| fill_length_2 | m | m | m | m |
| seg_length_2 | m | m | m | m |
| offset_2 | x | x | m | x |
| opcode | m | m | m | m |
| subsystem | x | x | x | x |
| id | x | x | x | x |
| service.code | m | m | m | m |
| loc_add.station | x | o | x | x |
| loc_add.segment | x | x | x | x |
| ssap | m | m | m | m |
| dsap | m | m | m | m |
| rem_add.station | m | m | m | m |
| rem_add.segment | x | x | x | x |
| serv_class | m | m | m | m |
| receive_l_sdu.length | x | x | m | x |
| send_l_sdu.length | m | m | m | m |
| l_status | x | x | x | x |
| user_data_1 | m | m | m | m |
| user_data_2 | x | x | m | x |

Request block header

Application block

User data 1
User data 2

Estructuras de Datos de los Servicios de Producción

| confirm | sda | sdn | srd | reply_update |
|----------------------|-----|-----|-----|--------------|
| length | r | r | r | r |
| user | x | x | x | x |
| rb_type | r | r | r | r |
| priority | r | r | r | r |
| subsystem | r | r | r | r |
| opcode | r | r | r | r |
| response | r | r | r | r |
| fill_length_1 | r | r | r | r |
| seg_length_1 | x | x | x | x |
| offset_1 | r | r | r | r |
| fill_length_2 | x | x | r | x |
| seg_length_2 | x | x | r | x |
| offset_2 | x | x | r | r |
| opcode | r | r | r | r |
| subsystem | x | x | x | x |
| id | x | x | x | x |
| service.code | r | r | r | r |
| loc_add.station | x | o | x | x |
| loc_add.segment | x | x | x | x |
| ssap | r | r | r | r |
| dsap | r | r | r | r |
| rem_add.station | r | r | r | r |
| rem_add.segment | x | x | x | x |
| serv_class | r | r | r | x |
| receive_1.sdu.length | x | x | r | x |
| send_1.sdu.length | r | r | r | r |
| l_status | r | r | r | r |
| user_data_1 | x | x | x | x |
| user_data_2 | x | x | r | x |



Request
block
header

Application
block

User data 1
User data 2

Estructuras de Datos de los Servicios de Producción

| indication | sda | sdn | srd | sdn_broadcast |
|----------------------|-----|-----|-----|---------------|
| length | r | r | r | r |
| user | x | x | x | x |
| rb_type | r | r | r | r |
| priority | r | r | r | r |
| subsystem | r | r | r | r |
| opcode | r | r | r | r |
| response | r | r | r | r |
| fill_length_1 | r | r | r | r |
| seg_length_1 | x | x | x | x |
| offset_1 | r | r | r | r |
| fill_length_2 | x | x | x | x |
| seg_length_2 | x | x | x | x |
| offset_2 | x | x | x | x |
| opcode | r | r | r | r |
| subsystem | x | x | x | x |
| id | x | x | x | x |
| service.code | r | r | r | r |
| loc_add.station | x | o | x | x |
| loc_add.segment | x | x | x | x |
| ssap | r | r | r | r |
| dsap | r | r | r | r |
| rem_add.station | r | r | r | r |
| rem_add.segment | x | x | x | x |
| serv_class | r | r | r | r |
| receive_1.sdu.length | r | r | r | r |
| send_1.sdu.length | x | x | x | x |
| l_status | x | x | r | x |
| user_data_1 | r | r | r | r |
| user_data_2 | x | x | x | x |



Request
block
header

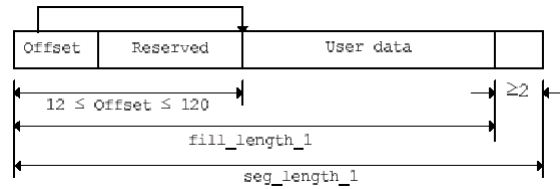
Application
block

User data 1
User data 2

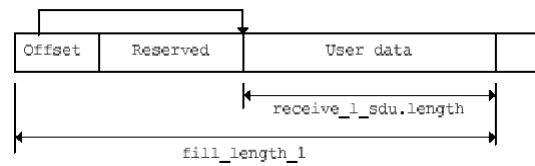
Estructuras de Datos de los Buffers de Datos



⇒ Buffer de Envío



⇒ Buffer de Recepción



Valores de Status - Confirm



- ok = Positive acknowledgment, service executed.
- rr = Negative acknowledgment, resources of the CP (remote) not available.
- ue = Negative acknowledgment, FDL application/FDL interface error (remote)
- rs = Service or rem_add not activated on SAP (remote)
- ls = Service not activated on SAP (local)
- na = No or no plausible reaction from station (remote)
- ds = CP (local) not in logical token ring or disconnected from the bus.
- iv = Invalid parameters in the request.

SDA

- ok = Transfer of the data from the CP completed
- ls = Service not activated on the SAP (local)
- ds = CP not in the logical token ring or disconnected from the bus
- iv = Invalid parameters in the request

SDN

SRD

- ue = Negative acknowledgment, FDL application/FDL interface error (remote).
- rs = Service or rem_add not activated on SAP (remote).
- ls = Service not activated on SAP (local).
- na = No or no plausible reaction from station (remote).
- ds = CP not in logical token ring or disconnected from the bus.
- iv = Invalid parameters in the request.
- dl = Response data low exist. Positive acknowledgment for transmitted data.
- dh = Response data high exist. Positive acknowledgment for transmitted data.
- nr = Negative acknowledgment. Response data not available on CP (remote). Positive acknowledgment for transmitted data.
- rdl = Response data low exist. Negative acknowledgment for transmitted data since CP resources (remote) are not available.
- rdh = Response data high exist. Negative acknowledgment for transmitted data since CP resources (remote) are not available.
- rr = Negative acknowledgment. CP resources (remote) and response data (remote) are not available.

Estructuras de Datos de los Servicios de Gestión



- ⇒ Utilizan la misma estructura de **Request Block** que los Servicios de Producción
- ⇒ Utilizan los **bloque de datos de usuario** para introducir la información estructurada correspondiente al **Servicio de Gestión** que realizan

| Service | Structure used |
|---------------|-----------------|
| sap_activate | fdl_sap |
| rsap_activate | fdl_sap |
| Others | No significance |

Servicios de Petición

Servicios de Respuesta

| Service | Structure used |
|-----------------------------|---------------------|
| fdl_read_value | Bus_parameter_block |
| fdl_event | Event_indication |
| lsap_status | Byte buffer |
| fdl_life_list_create_local | Byte buffer |
| fdl_life_list_create_remote | Byte buffer |
| fdl_ident | Byte buffer |
| fdl_read_statistic_ctr | Statistic_ctr_list |
| fdl_read_las_statistic_ctr | Byte buffer |
| await_indication | Byte buffer |
| others | No significance |

Estructuras de Datos de los Servicios de Gestión



```

struct bus_parameter_block
{
    UBYTE          hsa;
    UBYTE          ts;
    enum           station_type station_type;
    enum           baud_rate   baud_rate;
    enum           medium_red;
    UWORD         retry_ctr;
    UBYTE         default_sap;
    UBYTE         network_connection_sap;
    UWORD         tsl;
    UWORD         tqui;
    UWORD         tset;
    UWORD         min_tsdrr;
    UWORD         max_tsdrr;
    d_word        ttr;
    UBYTE         g;
    boolean       in_ring_desired;
    enum           physical_layer physical_layer;
    struct         ident        ident;
};
    
```

```

struct ident
{
    UBYTE          reserved_header[8];
    UBYTE          ident[202];
    UBYTE          response_frame_length;
};
    
```

```

struct event_indication
{
    struct event_ctr time_out;
    struct event_ctr not_syn;
    struct event_ctr uart_error;
    struct event_ctr out_of_ring;
    struct event_ctr sdn_not_indicated;
    struct event_ctr duplicate_address;
    struct event_ctr hardware_error;
    struct event_ctr mac_error;
};
    
```

```

struct fdl_sap
{
    UWORD          user_id;
    UBYTE          max_l_sdu_length;
    UBYTE          access_sap;
    UBYTE          access_station;
    UBYTE          access_segment;
    UBYTE          max_l_sdu_length;
    UBYTE          sda;
    UBYTE          sdn;
    UBYTE          srd;
    UBYTE          csrd;
    void           far *rup_l_sdu_ptr_low;
    void           far *rup_l_sdu_ptr_high;
};
    
```

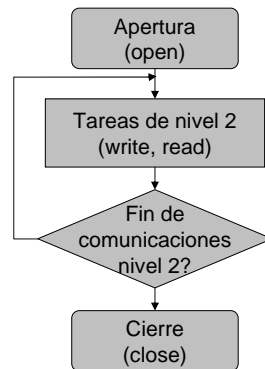
```

struct event_ctr
{
    UWORD          threshold;
    UWORD          counter;
};
    
```

```

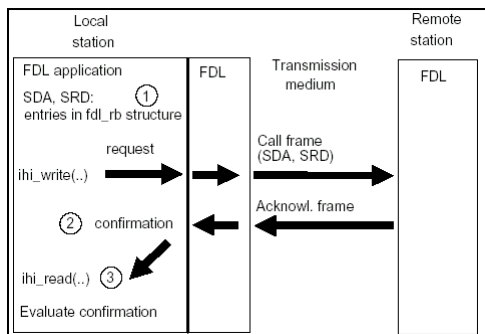
struct statistic_ctr_list
{
    UWORD          invalid_start_delimiter_ctr;
    UWORD          invalid_fcb_fcv_ctr;
    UWORD          invalid_token_ctr;
    UWORD          collision_ctr;
    UWORD          wrong_fcs_or_ed_ctr;
    UWORD          frame_error_ctr;
    UWORD          char_error_ctr;
    UWORD          retry_ctr;
    d_word         start_delimiter_ctr;
    d_word         stop_receive_ctr;
    d_word         send_confirmed_ctr;
    d_word         send_sdn_ctr;
};
    
```

Estructura Básica de una Aplicación FDL

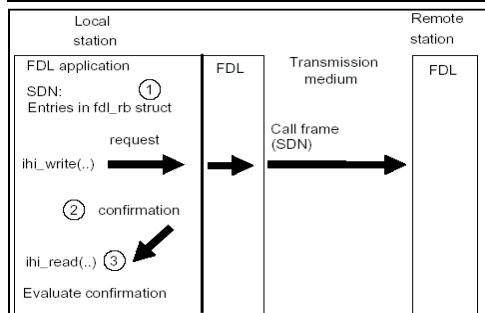


| Interfaz IHI | Funciones de Librería C para FDL | Interfaz SCP |
|---------------------|-----------------------------------------------|----------------------|
| ihi_open_dev | Función de conexión para la aplicación FDL | SCP_open |
| ihi_write | Envío de datos al interfaz | SCP_send |
| ihi_read | Recepción de datos desde el interfaz | SCP_receive |
| ihi_close | Función de desconexión para la aplicación FDL | SCP_close |
| | Búsqueda de identificadores de error | SCP_get_errno |

Envío de Tramas de Datos

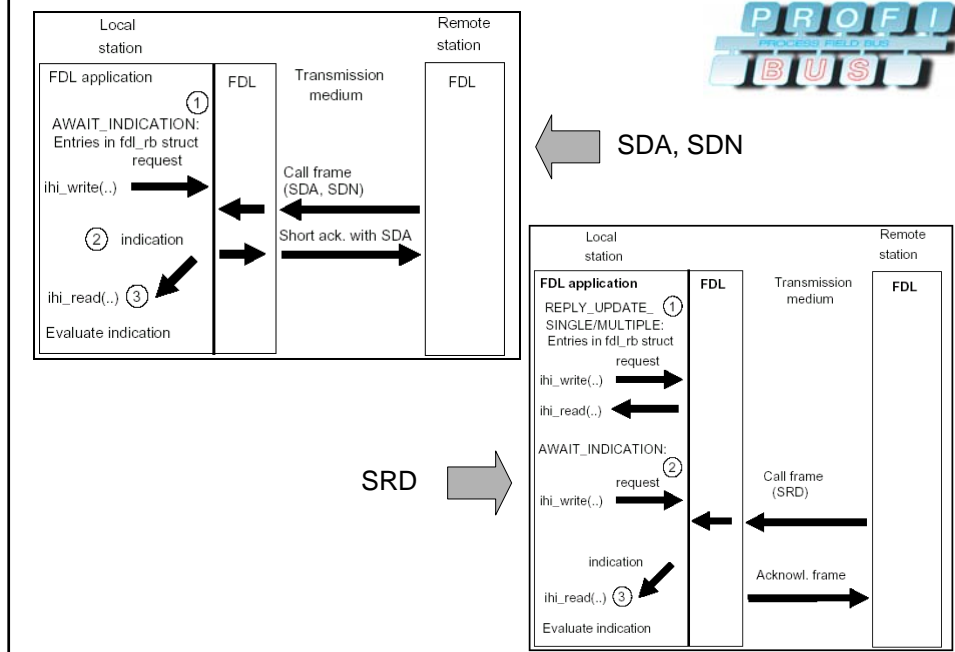


← SDA, SRD



← SDN

Recepción de Tramas de Datos



Librerías FDL Bajo Windows

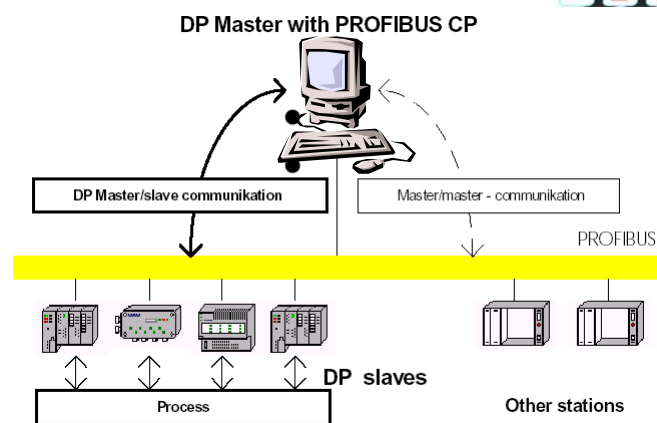
| Ficheros | Nombre | Significado |
|------------------------------|--------------|-----------------------------------------------|
| Librería estática FDL | s7onlinx.lib | Funciones FDL |
| Fichero include FDL | fdl_rb.h | Declaraciones y constantes específicas de FDL |

PROFIBUS DP

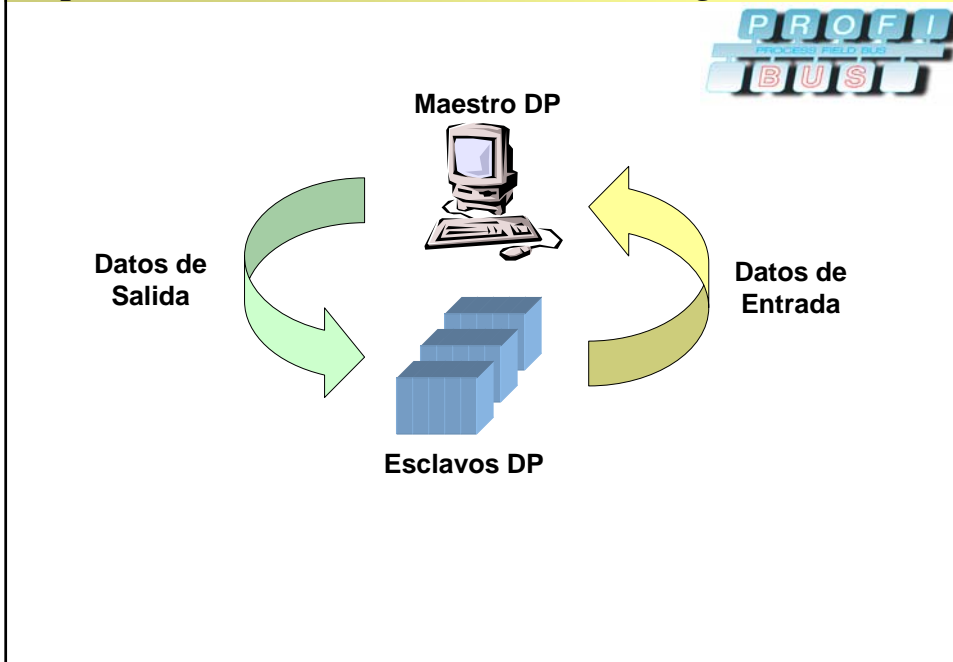


- ⇒ Arquitectura Básica
- ⇒ Modos de Funcionamiento
- ⇒ Tramas de Control
- ⇒ Programación de Interfaz DP
- ⇒ Estructuras de Datos
- ⇒ Estructura de Aplicación
- ⇒ Funciones de Librería

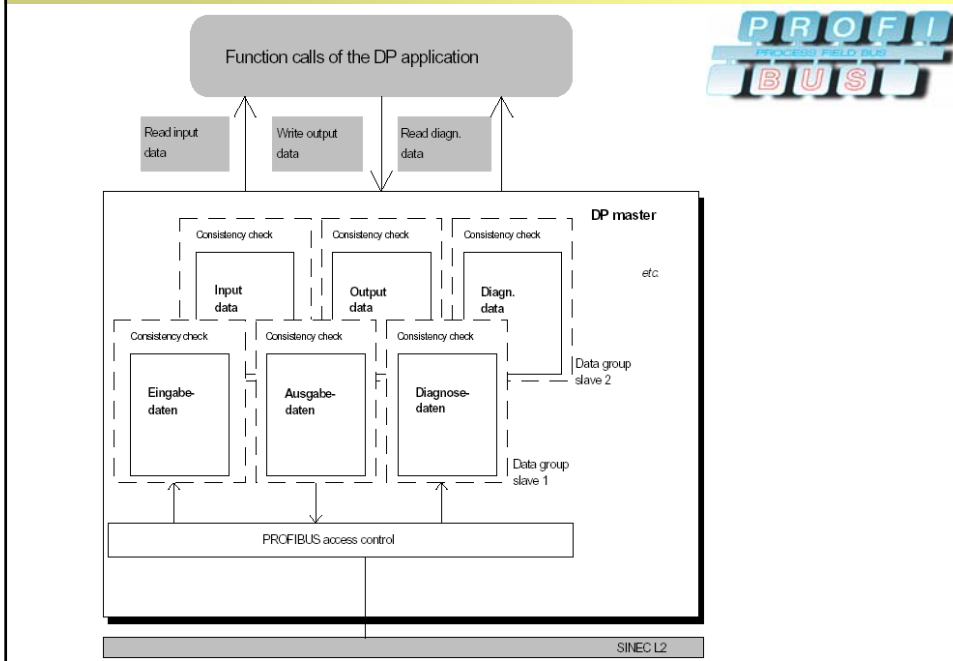
Arquitectura Básica



Representación del Modo de Polling



Áreas de Datos del Maestro DP



Modos de Funcionamiento del Maestro DP



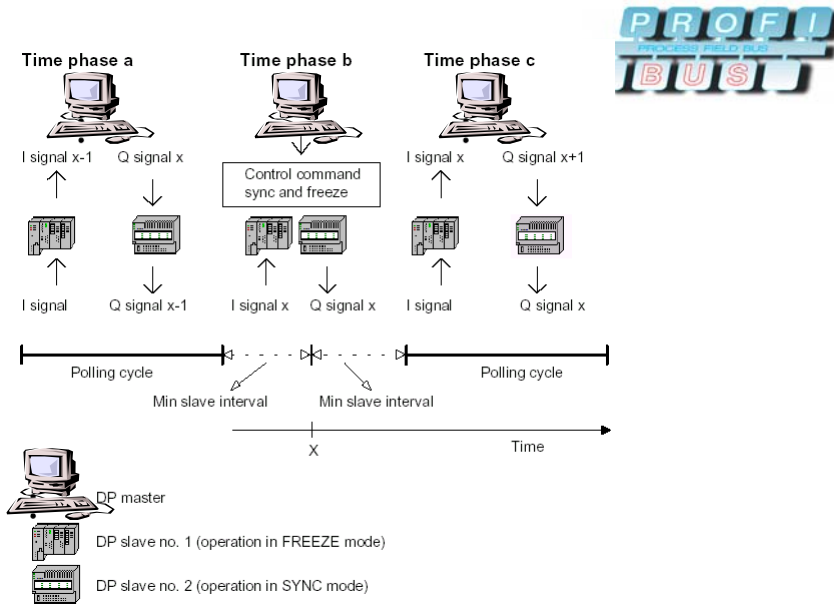
| Mode | Meaning |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OFFLINE | There is no communication whatsoever between the DP master and the DP slaves. This is the initial status of the DP master. |
| STOP | There is also no communication between the DP master and DP slaves in this mode. In contrast to the OFFLINE mode, a DP diagnostic station (DP master Class 2) can read out diagnostic information of the DP master. |
| CLEAR | In this mode, the master assigns parameters to and configures all DP slaves entered in the database and activated. Following this, the cyclic data exchange between the DP master and DP slaves begins. In the CLEAR mode, the value 0h is sent to all slaves with process output, i.e. the process output is deactivated. The input data of the slaves are known and can be read out. |
| OPERATE | The cyclic data transfer to the DP slaves takes place in the OPERATE mode. This is the productive phase. In this mode, the DP slaves are addressed one after the other by the DP master. The call frame contains the current output data and the corresponding response frame contains the current input data. |

Tramas de Control



- ⇒ **SYNC:** Se inmoviliza el estado de las salidas
- ⇒ **UNSYNC:** Se cancela el comando sync para salidas
- ⇒ **FREEZE:** Se lee el estado de las entradas y se inmoviliza
- ⇒ **UNFREEZE:** Se cancela el comando freeze para entradas
- ⇒ **CLEAR:** Se resetean todas las salidas

Tramas de Control Cíclicas



Funciones de Programación del Interface DP

Initialization Functions

| | |
|------------|---------------------------------------------------------|
| dpn_init() | Log on a DP application at the DP programming interface |
| dpn_wd() | Activate monitoring of the DP application |

Database Functions

| | |
|---------------------|------------------------------------------------|
| dpn_read_bus_par() | Read out the bus parameters from the database |
| dpn_load_bus_par() | Modify DP-specific parts of the bus parameters |
| dpn_read_slv_par() | Read slave parameters |
| dpn_set_slv_state() | Activate/deactivate a slave of the database |
| dpn_read_cfg() | Read out the DP configuration |

Diagnostic Functions

| | |
|---------------------|------------------------------------|
| dpn_slv_diag() | Request diagnostic data of a slave |
| dpn_read_sys_info() | Read out system information |

Data transfer Functions

| | |
|-----------------|------------------------------------------|
| dpn_out_slv() | Send output data to a single slave |
| dpn_out_slv_m() | Send output data to several slaves |
| dpn_in_slv() | Read input data from a single slave |
| dpn_in_slv_m() | Read input data from several slaves |
| dpn_read_slv() | Read local output data of a single slave |

Control Functions

| | |
|-------------------|----------------------------------------|
| dpn_set_mode() | Set the DP mode |
| dpn_get_mode() | Read out the current DP mode |
| dpn_global_ctrl() | Send control commands to a slave group |

Close Functions

| | |
|-------------|----------------------------------------------------------|
| dpn_reset() | Log off a DP application at the DP programming interface |
|-------------|----------------------------------------------------------|

Llamadas y Valores de Retorno de las Funciones del Interface DP



Formato de las Llamadas

```
Return value = dpn_...(struct dpn_interface far * ptr);  
or  
Return value = dpn_...(struct dpn_interface_m far * ptr);
```

Valores de Retorno y Error Code

| Return value of the function call | Meaning |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DPN_NO_ERROR | The function was processed. The return parameters of the dpn_interface structure (dpn_interface_m) are valid. |
| Not DPN_NO_ERROR | The processing of the function was aborted due to an error. The return parameters of the dpn_interface structure (dpn_interface_m) are not valid. The return parameter of the function contains a detailed error ID. |

Estructuras de Datos



```
struct dpn_interface  
{  
struct REFERENCE reference; /* Reference of applic. */  
unsigned char stat_nr; /* Station number */  
unsigned char length; /* Length */  
unsigned short error_code; /* Error identifier */  
unsigned char slv_state; /* Status of the DP slave */  
unsigned char sys_state; /* Status of the DP master */  
unsigned char sys_event; /* Event messages */  
unsigned char user_data[255]; /* Data field */  
};  
  
struct REFERENCE  
{  
unsigned char board_select;  
unsigned char access;  
};
```

```
struct dpn_interface_s  
{  
struct REFERENCE reference;  
unsigned char stat_nr;  
unsigned char length;  
unsigned short int error_code;  
unsigned char slv_state;  
unsigned char sys_state;  
unsigned char sys_event;  
unsigned char user_data[DPN_SINGLE_SIZE];  
};  
  
struct dpn_interface_m  
{  
struct dpn_interface_s dpn_if_single[DPN_MULTIPLE_SIZE];  
};
```

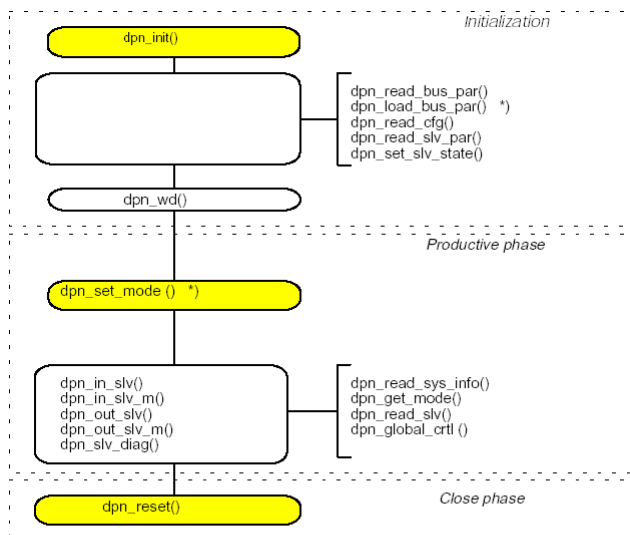
Asignación de Parámetros



| Function Calls | reference | stat_nr | length | error code | slv_state | sys_state | sys_event | user data |
|---------------------|-----------|---------|--------|------------|-----------|-----------|-----------|-----------|
| dpn_init() | C/R | - | C | R | - | - | R | C |
| dpn_wd() | C | - | C | R | - | R | R | C |
| dpn_read_bus_par() | C | - | C/R | R | - | R | R | R |
| dpn_load_bus_par() | C | - | C | R | - | R | R | C |
| dpn_read_slv_par() | C | C | C/R | R | C/R | R | R | R |
| dpn_set_slv_state() | C | C | C | R | R | R | R | C |
| dpn_read_cfg() | C | - | C/R | R | - | R | R | R |
| dpn_slv_diag() | C | C | C/R | R | R | R | R | R |
| dpn_read_sys_info() | C | - | C/R | R | - | R | R | R |
| dpn_out_slv() | C | C | C | R | R | R | R | C |
| dpn_out_slv_m() | C | C | C | R | R | R | R | C |
| dpn_in_slv() | C | C | C/R | R | R | R | R | R |
| dpn_in_slv_m() | C | C | R | R | R | R | R | R |
| dpn_read_slv() | C | C | C/R | R | R | R | R | R |
| dpn_set_mode() | C | - | - | R | - | C/R | R | - |
| dpn_get_mode() | C | - | - | R | - | R | R | - |
| dpn_global_ctrl() | C | C | C | R | C | R | R | C |
| dpn_reset | C | - | - | R | - | - | - | - |

C: El parámetro se utiliza en la llamada
R: El parámetro aparece en la respuesta

Estructura de una Aplicación DP



*) only permitted for central DP application

Constantes de Acceso a los Elementos de Comunicación

Acceso de la Aplicación – reference.access

| Possible Entries | Meaning |
|----------------------|---------------------------------------------------------------------|
| DPN_ROLE_CENTRAL | The DP application that is logging on is a central application. |
| DPN_ROLE_NOT_CENTRAL | The DP application that is logging on is not a central application. |
| DPN_SYS_CENTRAL | One of the DP applications is a central application. |
| DPN_SYS_NOT_CENTRAL | None of the DP applications is a central application. |

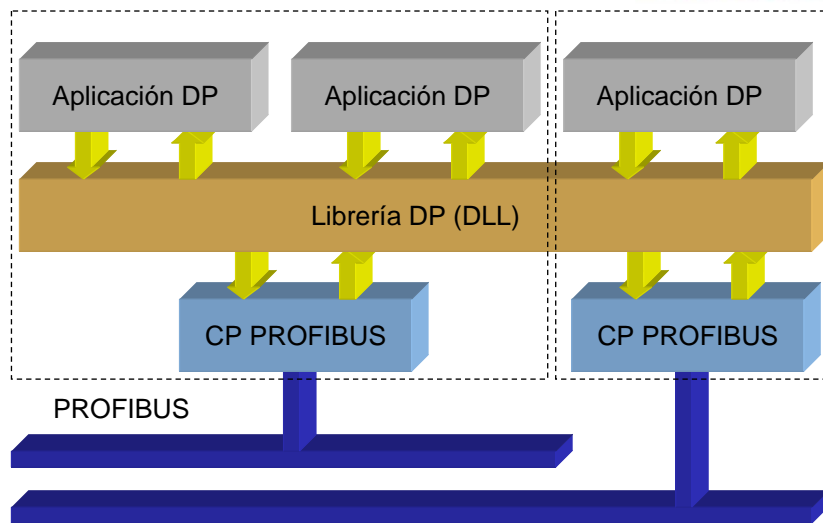
Acceso a los Esclavos

| Possible Entries | Meaning |
|--------------------|----------------------------------|
| DPN_SLV_NO_ACCESS | No access requested |
| DPN_SLV_READ | Only read input data |
| DPN_SLV_WRITE_READ | Send output data/read input data |

Inicialización DP

| Parameter | Possible Entries | Comment |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------|
| reference.board_select | 1 - 2 | Number of the CP |
| reference.access | (DPN_SYS_CENTRAL) (DPN_ROLE_CENTRAL) (DPN_SYS_CENTRAL) (DPN_ROLE_NOT_CENTRAL) (DPN_SYS_NOT_CENTRAL) (DPN_ROLE_NOT_CENTRAL) | Type and environment of the DP application |
| user_data[n] where n=0 to 125 | DPN_SLV_WRITE_READ DPN_SLV_READ DPN_SLV_NO_ACCESS | Right of access to DP slaves |

Estructura en Multitarea



Librerías DP Bajo Windows



| <i>Ficheros</i> | <i>Nombre</i> | <i>Significado</i> |
|-----------------------------|---------------|----------------------------------------------|
| Librería dinámica DP | dplib.dll | Funciones DP en DLL |
| Librería estática DP | dplib.lib | Prototipos de funciones DP |
| Fichero include DP | dpn_user.h | Declaraciones y constantes específicas de DP |