SIEMENS

IEC 61158

Working with

PROFIBUS-DP

Device Description
Data Files
GSD

Version: 2.2
Date: April 2003

Liability Exclusion
We have tested the contents of this document regarding agreement with the hardware and software described. Nevertheless, deviations can't be excluded, so that we don't guarantee complete agreement. However, the data in this document is checked periodically. Necessary corrections are included in subsequent editions. We gratefully accept suggestions for improvement.

Subject to technical changes.

Table of Contents

1 The PROFIBUS DP Generic Station Description File

1.1 Preface to GSD Revision 1

The provided information is based on PROFIBUS IEC 61158 part 3 and the additional implementation guideline. The document was defined to our best knowledge, however, in case of any doubt IEC 61158 and the implementation guideline takes precedence.

PROFIBUS DP according to IEC 61158 and PROFIBUS DP/V1 support many possibilities to implement data exchange between bus master and the connected slaves. From the simplest slave that services only a few input/output channels up to the intelligent slave that handles preprocessing tasks, a PROFIBUS DP master can carry out data exchange. For that reason, field devices with PROFIBUS DP connection can be optimally adapted to the respective automation task. In order to cover this large variety safely and conveniently, the bus master (Class 1 master) needs the technical data of the connected field device in the form of a Generic Station Description file (GSD file). The GSD file is to be generated by the field device manufacturer as an ASCII file in the form of an electronic data sheet (for example, MSDOS Editor). In order to describe the technical details of a field device uniformly, a large number of key words were defined that uniquely define a certain attribute of the field device. This ensures, among other things, that different field devices by different manufacturers can exchange data with any master that conforms to standard. An accredited test lab tests the complete standard-conforming performance. Simple field devices can be described with a few key words.

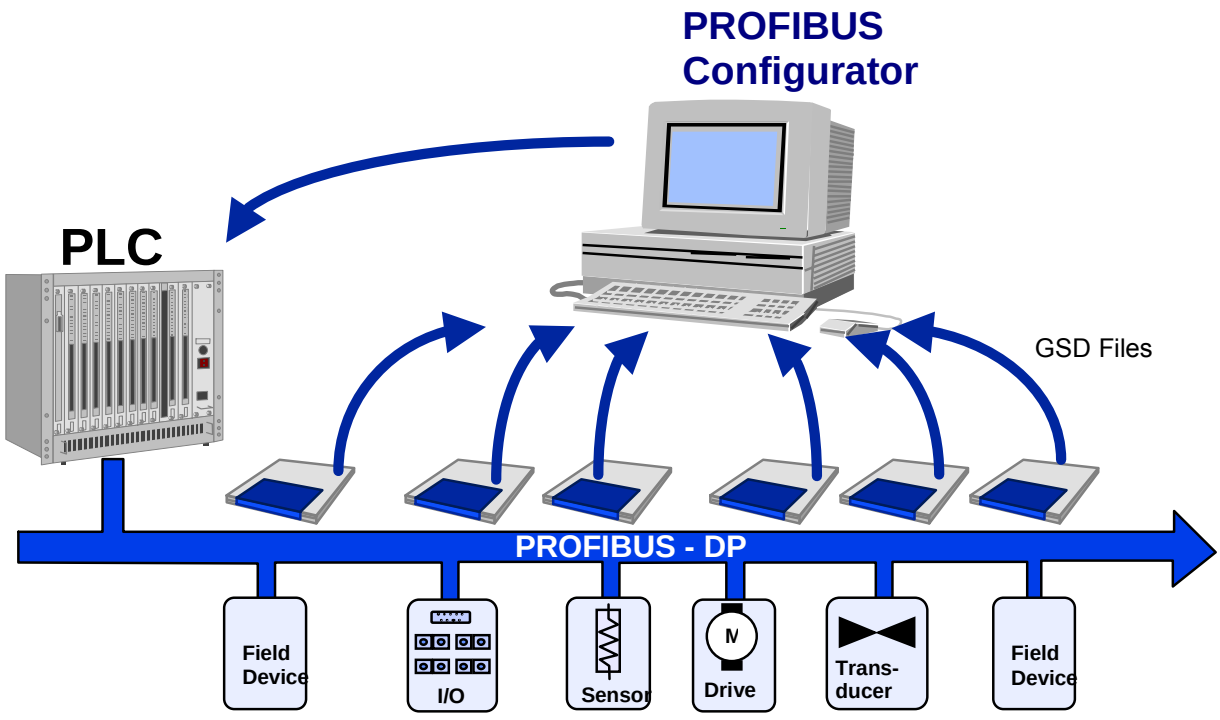


Figure 1-1: The Meaning of the Generic Station Description File

While reading this manual you may come across different revision numbers. During the years it became necessary to describe the functional extensions of field devices. Therefore you´ll find a brief overview on the different revisions:

- **Revision 1:** describes the general keywords, dedicated to relatively simple field devices which are supporting the cyclic data exchange.
- **Revision 2:** describes some syntactical changes as well as additional transmission rates and the necessary reaction timings.
- **Revision 3:** describes the keywords of the acyclic data exchange acc to DP-V1. Furthermore new physical interfaces and the requirements of PROFIBUS PA are explained.
- **Revision 4:** the following functionality is described:

  Isochron Mode,

  Data Exchange with Broadcast, -

  Subscriber_Supp,

  F-Parameter,

  Extended Parameterization

  Extended Diagnostic Description,

  Automatically SlotNumber mapping, Subsystems

  HMD ...(HART Master Devices), Extended Description of MaxTsdr for Optimizing

Essentially, the following data is included in a GSD file:

- The supported transmission rates
- The length of the input/output data to be exchanged
- The meaning of the diagnostic data, and possibly of the user parameters
- Type of field device (compact station, modular station)
- Text assignments for symbolic configuring
- The supported services (sync/freeze mode, …)

1.2 Who needs a GSD File?

Every Class 1 master and all field devices with slave functionality have to be described by the manufacturer with a GSD file.

1.3 Who does what with the GSD File?

Configuring tools for the PROFIBUS DP master that is to be configured interpret the content of the GSD files of the slaves, and from it, generate a master parameter set for the Class 1 master that handles the user data traffic. In part, Class 2 master functionalities are integrated, in order to load configuring data to the Class 1 master. A Class 2 master needs the GSD files of a Class 1 master in order to recognize, for example, in which form the configuring data can be loaded to the Class 1 master. If a Class 1 master supports the services Upload and Download, the configuring data can be loaded to the Class 1 master online, and existing configuring data can be changed online (refer to Figure 1-1).
Based on the information from the GSD files, the Class 1 master recognizes the following: the degree of expansion of the bus, which services the respective slave supports, and in which form the data is to be exchanged.

1.4 How does the Configuring Tool Process the GSD Files?

GSD files are needed during configuring and commissioning. Every manufacturer of a PROFIBUS DP Class 1 master makes a configuring tool available for configuring the Class 1

master that knows the internal data structure of the Class 1 master, and of the host system. When configuring a system, the GSD files that are needed respectively are to be made known to the configuring tool. Usually, this is done by copying the GSD files to the hard disk of the PC (the exact path indication is provided in the description of the configuring tool). When a system is configured, the configuring tool interprets the data of the GSD file for the field device that was selected. In addition, validity checks are performed so that the configuring data is structured logically correct.
At the end of configuring, the user can select in what form the compiled configuring data is to be transferred to the Class 1 master (usually on a diskette, Flash-EPROM, or online). When commissioning the system, the interpretation of the GSD file can provide information regarding errors that might occur.

1.5 Where Does the User Obtain the GSD Files?

The manufacturer supplies the GSD files for the respective field device, together with the respective product. Some manufacturers include GSD files with the configuring tool. GSD files that are not included in the configuring tool can be obtained as follows:

- through the Internet address of the PROFIBUS Trade Organization (PNO) (address: http://www.profibus.com

1.6 How Can a GSD File be Created?

GSD files are created as ASCII files with an ASCII Editor by describing each feature of the field device with a standardized key word. To make it easy for everyone the PROFIBUS Nutzerorganisation provides a GSD-editor free of charge which can be downloaded from http://www.profibus.com.

1.7 How Can a GSD file be Checked for Correctness?

After the GSD file has been created, it can be checked internally from the GSD-editor.

2 The Structure of a GSD File

A GSD file exists once if it is configured independent of language (*.gsd). If it is generated in a certain language, it may exist more often. One GSD file is then to be used per language, where only the parameters of the type Visible String may differ. The language-related GSD files differ in the last letter of the extension (*.gs?).

Default (independent of language):	?=d
German	?=g
English	?=e
French	?=f
Italian	?=i
Portuguese	?=p
Spanish	?=s

Tabel 1-1: Language-Dependent GSD Files

Example of a GSD Name:
Abc_0111.gsd (this means the following:
Abc_ = 4 characters free to choose
0111 = Ident number 011 assigned by the PNO, always 4 characters
.gs**d** = **d**efault. Language-independent GSD file)

2.1 Preface to GSD Revision 2

The development of the PROFIBUS product range also entails enhancements as regards the device properties and features to be described in the GSD files. These developments, in particular the introduction of PROFIBUS-PA and the associated new transmission rates, are the reason for the extension to the GSD file for the present GSD Revision 2.

A primary aim of the revision was to improve the readability of the formal description of the GSD file. The individual rules in this clause have been numbered in order to enable better referencing. Rules that left room for interpretation have been made more precise. Rules that unnecessarily limited the format of the GSD file and thus made it more difficult to create and read GSD files have been relaxed.

The changes to the informal description of the keywords since GSD Revision 1 essentially boil down to the addition of the keywords for the new transmission rates.

In the formal description the following changes have been made since GSD Revision 1.

- Description of continuation lines
- Description of a beginning of a GSD line
- Description of white spaces in octet strings
- Description of white spaces in User definitions
- Support of new transmission rates
- MaxTsdr for new transmission rates
- Trdy for new transmission rates (only relevant for the master)
- Tqui for new transmission rates (only relevant for the master)
- Tset for new transmission rates (only relevant for the master)
- Tsdi for new transmission rates (only relevant for the master)
- Subfamily_Name description
- Change of the reference number to Unsigned16
- Last_Bit limited to 495
- Extension of Unit-Def-Items
- Value range to {0<=Bit<=495}
- Ext-Module-Prm-Len description
- Is concluded with EndExtUserPrmData
- Replacement of the previous module definition
- Replacement of the previous GSD description
- User_Prm_Data_Def has been deleted

2.2 General PROFIBUS DP Slave Key Words in the GSD File (Revision 1 and 2)

Each line starts with one of the key words below. The key words described below are standardized, and are to be used only in the described designation. Key words that are company-specific can be defined, and have to be interpreted that way. These self-defined key words are not to be read by configuring tools of other companies. A PROFIBUS DP GSD file always starts with the key word **#Profibus_DP**.
The type ID specified for the keyword refers to the parameter with the same name. Regarding the parameters, the following differentiation is made:

- Mandatory (M) (absolutely required)
- Optional (O) (possible in addition)
- Optional with default = 0 if not present (D); (the key words marked with *) should always be specified because of the better readability of the GSD file, even if the default setting is 0.
- At least one of the group (G) matches the corresponding transmission rate.

GSD_Revision: (M starting with GSD_Revision 1)
Version ID of the GSD file format
Type: Unsigned8
Example: GSD_Revision= 1

Vendor_Name: (M)
Vendor Name.
Type: Visible String (32)
Example: Vendor_Name= "Corp_ABC & Co"
Model_Name: (M)
Manufacturer Name (Controller Type) of the DP device. This name is indicated in the configuring tool.
Type: Visible String (32)
Example: Model_Name= "Modular I/O Station"

Revision: (M)
Version of the DP device.
Type: Visible String (32)
Example: Revision= "Version 01"

Revision_Number: (O starting with GSD_Revision 1)
Version ID . The value of the Revision_Number has to agree with the value of the Revision_Number in the slave-specific diagnosis if provided.
Type: Unsigned8 (1 bis 63)
Example: Revision_Number= 05

Ident_Number: (M)
Identifies the device type of the DP device. Each field device is characterized by an Ident number allocated by the PROFIBUS Trade Organization (PNO) which establishes a unique reference to the GSD file, and thus to the technical data of the field device. Field device variants that can be described with **one** GSD file, may use the same Ident number (for example, modular devices). Data exchange with a field device is possible only if the PROFIBUS DP device identifies itself clearly with the Ident number of the field device during system power-up (parameter assignment message).
Type: Unsigned16
Example: Ident_Number=0x00A2

Protocol_Ident: (M)
Protocol used for the DP devices.
Type: Unsigned8
0: PROFIBUS-DP,
16 to 255: manufacturer-specific
Example: Protocol_Ident= 0 ; here, a PROFIBUS DP device is described

Station_Type: (M)
DP device type.
Type: Unsigned8
0: DP Slave,
1: DP Master (Class 1)
Example: Stations_Type= 0 ; here, a PROFIBUS DP slave is described

FMS_supp: (D) [1)]
This device is a FMS/DP mixed device.
Type: Boolean (1: True)
Example: FMS_supp= 0 ; it is a pure DP device

Hardware_Release: (M)
Hardware release of the DP device.
Type: Visible String (32)
Example: Hardware_Release= "Hardware Release HW= 0.1"

Software_Release (M)
Software release of the DP device.
Type: Visible String (32)
Example: Software_Release= "Software Release SW= 1.01"

9.6_supp: (G)
The DP device supports the transmission rate 9.6 kBaud.
Type: Boolean (1: True)
Example: 9.6_supp= 1 ; the device supports the specified transmission rate

19.2_supp: (G)
The DP device supports the transmission rate 19.2 kBaud.
Type: Boolean (1: True)
Example: 19.2_supp= 1 ; the device supports the specified transmission rate

31.25_supp: (G)
The DP device supports the transmission rate 31.25 kBaud.
Type: Boolean (1: True)
Example: 31.25_supp= 1 ; the device supports the specified transmission rate

45.45_supp: (G)
The DP device supports the transmission rate 45.45 kBaud.
Type: Boolean (1: True)
Example: 45.45_supp= 1 ; the device supports the specified transmission rate

93.75_supp: (G)
The DP device supports the transmission rate 93.75 kBaud.
Type: Boolean (1: True)
Example: 93.75_supp= 1 ; the device supports the specified transmission rate

[1)] Although this key word is not mandatory, this detail should always be defined because of the easier readability of a GSD.

187.5_supp: (G)
The DP device supports the transmission rate 187.5 kBaud.
Type: Boolean (1: True)
Example: 187.5_supp= 1 ; the device supports the specified transmission rate

500_supp: (G)
The DP device supports the transmission rate 500 kBaud.
Type: Boolean (1: True)
Example: 500_supp= 1 ; the device supports the specified transmission rate

1.5M_supp: (G)
The DP device supports the transmission rate 1.5 MBaud.
Type: Boolean (1: True)
Example: 1.5M_supp= 1 ; the device supports the specified transmission rate

3M_supp: (G)
The DP device supports the transmission rate 3 MBaud.
Type: Boolean (1: True)
Example: 3M_supp= 1 ; the device supports the specified transmission rate

6M_supp: (G)
The DP device supports the transmission rate 6 MBaud.
Type: Boolean (1: True)
Example: 6M_supp= 1 ; the device supports the specified transmission rate

12M_supp: (G)
The DP device supports the transmission rate 12 MBaud.
Type: Boolean (1: True)
Example: 12M_supp= 1 ; the device supports the specified transmission rate

Important note: To speed up the complete system the max TSDR can be optimized. Please refer to the corresponding ASIC descriptions on www.ad.siemens.de/csi_e/pb-doc
These informations are necessary when Isochronous mode is used (see GSD-Revision 4).

MaxTsdr_9.6: (G) (Value= 60)
This is the time that a responder needs as a maximum at a transmission rate of 9.6 kBaud to respond to a request message.
Type: Unsigned16
Time base: bit time
Input: MaxTsdr_9.6= 60

MaxTsdr_19.2: (G) (Value= 60)
This is the time that a responder needs as a maximum at a transmission rate of 19.2 kBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

MaxTsdr_31.25: (G) (Value= 60)
This is the time that a responder needs as a maximum at a transmission rate of 31.25 kBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

MaxTsdr_45.45: (G) (Value= 60)
This is the time that a responder needs as a maximum at a transmission rate of 45.45 kBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

MaxTsdr_93.75: (G) (Value= 60)
This is the time that a responder needs as a maximum at a transmission rate of 93.75 kBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

MaxTsdr_187.5: (G) (Value= 60)
This is the time that a responder needs as a maximum at a transmission rate of 187.5 kBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

MaxTsdr_500: (G) (Value= 100)
This is the time that a responder needs as a maximum at a transmission rate of 500 kBaud to respond to a request message. (refer to DIN 19245 Part 1\4.91 Section 4.1.7).
Type: Unsigned16
Time base: bit time

MaxTsdr_1.5M: (G) (Value= 150)
This is the time that a responder needs as a maximum at a transmission rate of 1.5 MBaud to respond to a request message (refer to DIN 19245 Part 1\4.91 Section 4.1.7).
Type: Unsigned16
Time base: bit time

MaxTsdr_3M: (G) (Value= 250)
This is the time that a responder needs as a maximum at a transmission rate of 3 MBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

MaxTsdr_6M: (G) (Value= 450)
This is the time that a responder needs as a maximum at a transmission rate of 6 MBaud to respond to a request message..
Type: Unsigned16
Time base: bit time

MaxTsdr_12M: (G) (Value= 800)
This is the time that a responder needs as a maximum at a transmission rate of 12 MBaud to respond to a request message.
Type: Unsigned16
Time base: bit time

Redundancy: (D)
This value indicates whether a device supports redundant transmission or not. Type: Boolean
0: No, 1: Redundancy supported.
Example: Redundancy= 0

Repeater_Ctrl_Sig: (D) [2)]
Here, the level of the connector signal CNTR-P is specified. *A look to chapter 3.1.2 "Additional keywords for different physical interfaces" may also be useful*
.
Type: Unsigned8
0: Not connected, 1: RS485, 2:TTL
Example: Repeater_Ctrl_Sig= 2

24V_Pins: (D) [2)]
Here, the meaning of the connector signals M24V and P24V is specified. *A look to chapter 3.1.2 "Additional keywords for different physical interfaces" may also be useful.*

Type: Unsigned8
0: Not connected, 1:Input, 2:Output
Example: 24V_Pins= 0

Implementation_Type: (O starting with GSD_Revision 1) [2)]
Here, a description is provided of what standard implementation is used in the DP slave; for example, standard software solution, controller solution, or ASIC solution. The name is specifed by the manufacturer of the standard solution. From this detail, configuring tools can already perform checks.
Type: Visible String (32)
Example: Implementation_Type= "SPC3 solution" or "Software solution"; when using the key word SPC3, the configuring tool COM PROFIBUS locks the first User_Prm_Byte for the user.

Bitmap_Device: (O starting with GSD_Revision 1)
Here, the file name (*.DIB) of the bitmap file is specified in DIB format (70*40 pixel (width*height) 16 colors), which normally contains the symbolic representation of the device. Depending on the configuring tool used, the bit map that is used is to be copied either to a certain directory, or the exact path is to be indicated -including the bitmap- prior to being used. Regarding this, read the description of the configuring tool used.
Type: Visible String (8)
Example: Bitmap_Device= "OK_state"

Bitmap_Diag: (O starting with GSD_Revision 1)
Here, the file name (*.DIB) of the bitmap file is specified in DIB format (70*40 pixel (width*height) 16 colors), which contains the symbolic representation of the device if there is a diagnosis. Depending on the configuring tool used, the bit map that is used is to be copied either to a certain directory, or the exact path is to be indicated -including the bitmap- prior to being used. Regarding this, read the description of the configuring tool used.
Type: Visible String (8)
Example: Bitmap_Diag= "Diag_sta"

Bitmap_SF: (O starting with GSD_Revision 1)
Here, the file name (*.DIB) of the bitmap file is specified in DIB format (70*40 pixel (width*height) 16 colors), which contains the symbolic representation of the device in special operating modes. The meaning is manufacturer-specific. Depending on the configuring tool used, the bit map that is used is to be copied either to a certain directory, or the exact path is to be indicated -including the bitmap- prior to being used. Regarding this, read the description of the configuring tool used.

Type: Visible String (8)

[2)] Although this key word is not mandatory, this detail should always be defined because of easier readability.

Example: Bitmap_SF= "SF_state"

Freeze_Mode_supp: (D)[1])

The DP device supports the freeze mode. During power-up, the parameter assignment message specifies whether the slave is to support the freeze mode. The freeze mode is activated with a global control message and causes the inputs of the slave to be “frozen“ in the momentary state. DP slaves that support the freeze mode have to ensure that in the next data cycle after the freeze control command, the values of the inputs that were frozen last are transmitted to the bus.
Type: Boolean (1: True)
Example: Freeze_Mode= 1 ;Freeze Mode is supported in the slave

Sync_Mode_supp: (D) [1)]
The DP device supports the sync mode. During power-up, the parameter assignment message specifies whether the slave is to support the sync mode. The sync mode is activated with a global control message and causes the slave to keep the outputs in the momentary state.
Type: Boolean (1: True)
Example: Sync_Mode= 1 ;Sync-Mode is supported in the slave

Field devices that support the sync/freeze mode can be combined into groups.

[1] Should always be specified

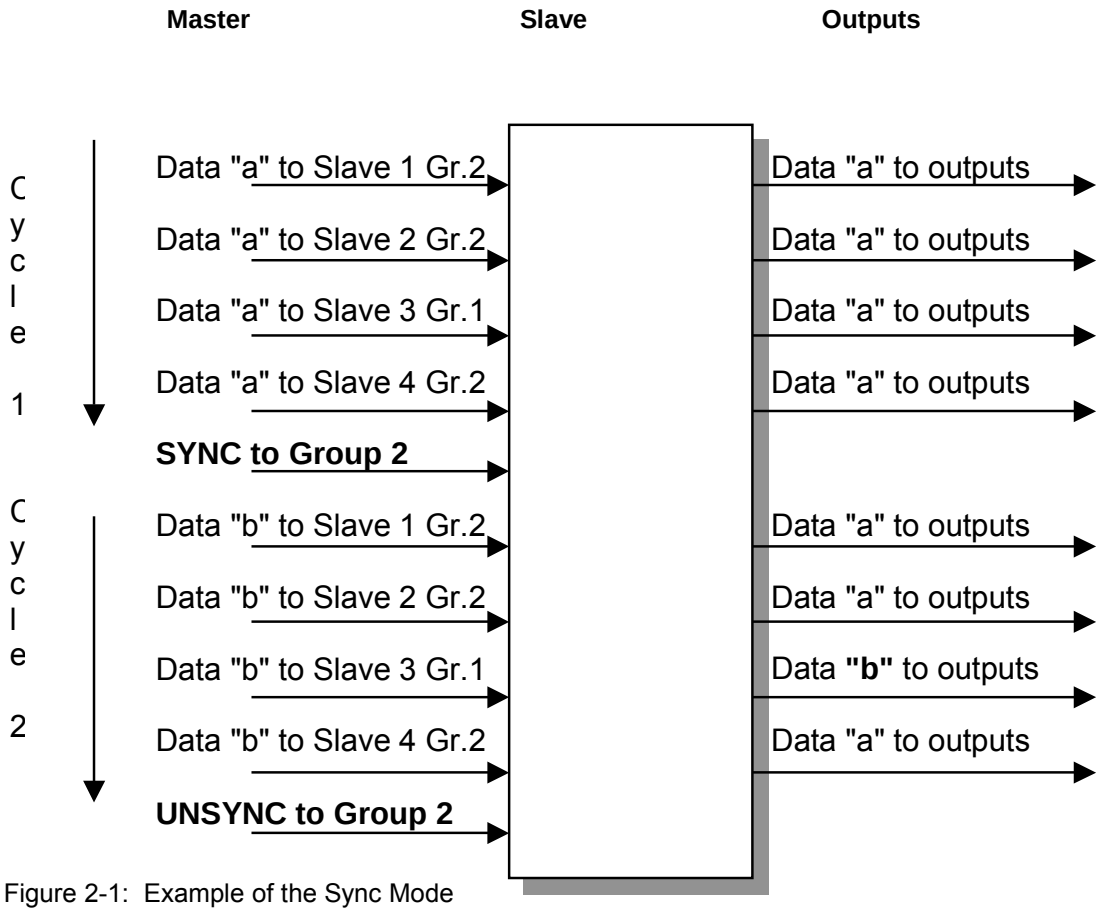


Figure 2-1: Example of the Sync Mode

In the next bus cycle after the UNSYNC command, the outputs are updated again.

Auto_Baud_supp: (D)[2]
The DP device supports the automatic transmission rate recognition. It automatically sets itself to the transmission rate specified by the master.
Type: Boolean (1: True)
Example: Auto_Baud_supp= 1 ; the function is supported

Set_Slave_Add_supp: (D) [2]
The DP device supports the function Set_Slave_Add for setting the slave address via the PROFIBUS.
Type: Boolean (1: True)
Example: Set_Slave_Add_supp= 1 ; the function is supported

[2] should always be specified

Fail_Safe: (D starting with GSD_Revision 1)
Here it is specified whether the DP slave accepts a data message without data instead of a data message with data = 0 in the CLEAR mode of the DP master (Class 1). As a matter of standard, the PROFIBUS DP master sets the outputs to zero if it is in the CLEAR mode. Here, the user can specify the pre assignments of the outputs.
Type: Boolean (1: True)
Example: Fail_Safe=1 ; means the slave accepts a data message without data in the
; Clear mode

Max_Diag_Data_Len: (M starting with GSD_Revision 1) [2]
Here, the maximum length of the diagnostic information (Diag_Data) is specified. At least, the 6 octets of the system diagnosis have to be always specified. This key word should always be indicated so that the bus master can optimize its memory location.
Type: Unsigned8 (6 - 244)
Example: Max_Diag_Data_Len= 10 ; the field device supplies 4 user diagnoses

Example: Max_Diag_Data_Len= 78 ; The field device supplies 70 device related
; user diagnostics + 6 Bytes
; standard diagnostics + 2 header bytes

Content in diagnosis telegram:

Bytes 1 - 6 Standard Diagnosis
Byte 7 **0011 1111** (Headerbyte 1) ; (63 bytes User diagnosis part I, including Byte 7,
; Headerbyte

Bytes 8 - 69 User diags (part I) ; --> 62 bytes user diag (user specific)
Byte 70 **0000 1001** (Headerbyte 2) ; 9 bytes User diags part II, including byte 70
; Headerbyte
Byte 71 - 78 User diags (part II) ; - -> 8 bytes user diag (user specific)

Max_User_Prm_Data_Len: (O starting with GSD_Revision 1)
Here, the maximum length of the User_Prm_Data is specified. The length of the transferred user parameters can have the specified maximum **or less**. They can also exist of User_Prm_Data and Ext_Module_Prm_Data.
The definition of this key word **excludes** the evaluation of User_Prm_Data_Len.
Type: Unsigned8 (0 - 237)
Example: Max_User_Prm_Data_Len= 120 ; as a maximum, 120 user parameters are
; possible from the field device

Modul_Offset: (D starting with GSD_Revision 1)
Here, the slot number is specified that is to appear as the first slot number in the configuring tool at configuring (is used to improve representation). In the case of modular devices, manufacturers sometimes designate as modules such units that the PROFIBUS DP can't address directly (such as PROFIBUS interface, power supply, CPU).
Type: Unsigned8
Example: Module_Offset=3 ; representation of the I/O modules starts withOffset 3

Slave_Family: (M starting with GSD_Revision 1)
In order to be able to find the individual slaves more easily when configuring a plant, the slaves are combined into families. The slave families are visualized for the user with the configuring tool. With the key word Slave_family, the DP slave is assigned to a function class. The family name is structured hierarchically. In addition to the main family, subfamilies can be formed that are attached with "@". A maximum of 3 subfamilies can be defined. Assignment to a slave family

facilitates finding a GSD file when configuring, since configuring tools file the stored GSD files according to the Slave_Family.

Example: Slave_Family=3@Digital@24V
The following main families are specified:
0: General (no assignment to the following categories possible)
1: Drives
2: Switching devices
3: I/Os
4: Valves
5: Controllers
6: HMI (MMI)
7: Encoders
8: NC/RC
9: Gateways
10: PLCs
11: Ident systems
12 PA (new in Revision 2)
13-255: reserved
Type: Unsigned8
Example: Slave_Family=7 ; the GSD file is stored under the category Encoders

User_Prm_Data_Len: (D)
Here, the length of the user-specific parameters (User_Prm_Data) is specified. When this keyword is defined and no Max_User_Prm_Data_Len is defined the user parameters have to have exactly that specified length. Please note that some ASICs need user-specific data.
Type: Unsigned8
Example: User_Prm_Data_Len= 5

User_Prm_Data: (O)
Type: Octet String
Meaning: Manufacturer-specific field. Provides the default value for User_Prm_Data. If this parameter is used, its length has to agree with the User_Prm_Data_Len.
Example: User_Prm_Data= 0x00,0x10,0xdf,0x00,0x23

Min_Slave_Intervall: (M)
This time specifies the minimum interval between two poll cycles for the DP device.
Type: Unsigned16
Time base: 100 µs
Example: Min_Slave_Intervall= 10 ; corresponds to a poll cycle of 1ms

The maximum time for the Min_Slave_Intervall at the transmission rates is:
up to 1500 kbit/s max. 20 (2 ms)
at 12 000 kbit/s max. 6 (0.6 ms)

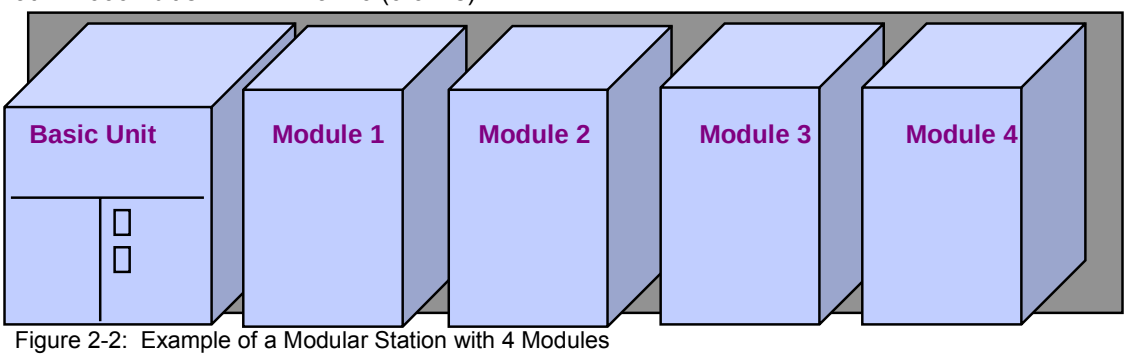


Figure 2-2: Example of a Modular Station with 4 Modules

Modular_Station: (D) [3]
Here it is specified whether the DP device is a modular station. Modular stations can be created from several modules. A list of the different modules that can be used in the field device is to be specified in the GSD file. A module is either a physical unit (refer to Figure 2-1) or a logical unit. When configuring, the configuring engineer can symbolically select the modules defined in the GSD file, and thus set up the modular station.
Type: Boolean (0: compact device, 1: modular device)

Max_Module: (M if Modular_Station)
Here, the maximum number of the modules is specified that can be inserted in the described device. The list of modules provided in the GSD file may be much longer.
Type: Unsigned8
Example: Max_Module= 4 ; 4 modules can be inserted

Max_Input_Len: (M if Modular_Station)
Here, the maximum length of the input data of a modular station is specified in bytes.
Type: Unsigned8
Example: Max_Input_Len= 100

Max_Output_Len: (M if Modular_Station)
Here, the maximum length of the output data of a modular station is specified in bytes.
Type: Unsigned8
Example: Max_Output_Len= 100

Max_Data_Len: (M if Modular_Station
Here, the largest sum of the lengths of the input/output data of a modular station is specified in bytes.
Type: Unsigned16
Example: Max_Data_Len= 200

Unit_Diag_Bit: (O)
To display manufacturer-specific status- and error messages of a DP slave centrally, it is possible to assign a text (Diag_Text) to a bit in the device-related diagnostic field.
Parameters used:
Bit:
Type: Unsigned16
Meaning: Bit position in the device-related diagnostic field
(LSB in the first byte is Bit 0).
Diag_Text:
Type: Visible String (32)
Example: Unit_Diag_Bit(0x12)="Short circuit on Channel 0...7" ; Bit No. 18 decimal means that a
; short circuit is present in the
; area of Channel 0 ... 7

Unit_Diag_Area: (O)
Between the key words Unit_Diag_Area and Unit_Diag_Area_End, the assignment of values in a bit field in the device-related diagnostic field to texts (Diag_Text) is specified.
Parameters used:
First_Bit:
Type: Unsigned16
Meaning: first bit position of the bit field
(LSB in the first byte is Bit 0)

[3] should always be specified

Last_Bit:
Type: Unsigned16
Meaning: Last bit position of the bit field. The bit field may consist of 16 bits maximum.
Value:
Type: Unsigned16
Meaning: Value in the bit field

Diag_Text:
Type: Visible String (32)
Example:
Unit_Diag_Area = 0 to 5 ;
Value(0) = "Faultless"
Value(1) = "Error on Input 0 to 23"
Value(2) = "Error on Output 0 to 15"
Value(3) = "24V failed"
Unit_Diag_Area_End

Module: (M) (refer also to Chapter 3.2)
Between the key words Module and EndModule, the following is provided: IDs of a DP compact device and the IDs of a module of a modular DP slave are specified; manufacturer-specific error types in the channel-related diagnostic field are specified; the Ext_User_Prm_Data is described. If, in the case of modular slaves, empty slots are to be defined as blank module (ID(s) 0x00), the empty module has to be defined. Otherwise, empty slots will not show up in the configuration data.
If the key word Channel_Diag is used outside the key words Module and EndModule, the same manufacturer-specific error type in the channel-related diagnostic field for all other modules.
If the key words Ext_User_Prm_Data_Ref or Ext_User_Prm_Data_Const are used outside the key words Module and EndModule, the associated User_Prm_Data area refers to the entire device , and the data in the parameter Offset to the entire User_Prm_Data. This User_Prm_Data area is placed at the start of the User_Prm_Data.
The module-specific User_Prm_Data is directly appended to the device-specific User_Prm_Data in the sequence in which the associated modules were configured. If the key words Ext_User_Prm_Data_Ref or Ext_User_Prm_Data_Const are used within the key words Module and EndModule, the data in the parameter Offset refers only to the start of the User_Prm_Data area that is assigned to this module.
Parameters used:

Mod_Name:
Type: Visible String (32)
Meaning: Name of a module used in a modular DP station, or device designation of a compact DP slave.

Config:
Type: Octet String (17)
Type: Octet String (244) (O starting with GSD_Revision 1)
Meaning: Here, the ID or IDs of the module of a modular DP slave or of a compact DP device is/are specified.

Module_Reference: (O starting with GSD_Revision 1)
Type: Unsigned16
Meaning: Here, the reference of the module description is specified. This reference has to be unique for a device (same Ident_Number). This type of referencing is useful in order to make language-independent configuring possible in a language-dependent system, or in order to recognize modules.

Examples:
```
Modular_Station=1                ;modular station
Max_Module=4
Module="Leerslot" 0x00           ; 0 is the ID for an empty slot (for example, PS module, etc.)
EndModule
                                 ; The selection possibilities between Module ... EndModule
                                 ; are displayed in the configuring tool
Module="2 Bytes Output" 0x21     ; in plain text
EndModule

Module="2 Bytes Input" 0x11      ;
EndModule
```

Ext_Module_Prm_Data_Len: (O starting with GSD_Revision 1)
Type: Unsigned8
Meaning: Here, the length of the associated User_Prm_Data is defined (the user parameters of a special module)

Channel_Diag: *(O)*
With the key word Channel_Diag, the assignment of manufacturer-specific error types (Error_Type) in the channel-related diagnostic field to the texts (Diag_Text) is specified.

Parameters used:
Error_Type:
Type: Unsigned8 (16 <= Error_Type <= 31)

Diag_Text:
Type: Visible String(32)

Ext_User_Prm_Data_Ref: (O starting with GSD_Revision 1)
Here, the reference to a User_Prm_Data description is specified. The definition of this key word excludes the evaluation of User_Prm_Data. If areas overlap when describing User_Prm_Data, the area defined last in the GSD file has priority.
Parameters used:
Reference_Offset:
Type: Unsigned8
Meaning: Here, the offset is defined within the associated part of the User_Prm_Data.

Reference_Number:
Type: Unsigned8
Meaning: This reference number has to be the same as the reference number that is defined in the User_Prm_Data description.

Ext_User_Prm_Data_Const: (O starting with GSD_Revision 1)
Here, a constant part of the User_Prm_Data is specified. The definition of this key word excludes the evaluation of User_Prm_Data. If the areas overlap when describing the User_Prm_Data , the area defined last in the GSD file has priority.
Parameters used:
Const_Offset:
Type: Unsigned8
Meaning: Here, the offset is defined within the associated part of the User_Prm_Data.

Const_Prm_Data:
Type: Octet String
Meaning: Here, constants or pre-assignments are defined within the

User_Prm_Data.

ExtUserPrmData: (O starting with GSD_Revision 1)
Between the key words ExtUserPrmData and EndExtUserPrmData, a parameter of the User_Prm_Data is described. The definition of this key word excludes the evaluation of User_Prm_Data.
Parameters used:
Reference_Number:
Type: Unsigned8
Meaning: Here, the reference of the User_Prm_Data description is specified. This refeence has to be unique.

Ext_User_Prm_Data_Name:
Type: Visible String (32)
Meaning: Plain text description of the parameter

Data_Type_Name:
Type: Visible String (32)
Meaning: Name of the data type of the parameter described

Default_Value:
Type: DataType (has to correspond to the Data_Type_Name)
Meaning: Default value of the parameter described

Min_Value:
Type: Data_Type (has to correspond to the Data_Type_Name)
Meaning: Minimum value of the parameter described

Max_Value:
Type: Data_Type (has to correspond to the Data_Type_Name)
Meaning: Maximum value of the parameter described

Allowed_Values:
Type: Data_Type_Array (16) (has to correspond to the Data_Type_Name)
Meaning: Permissible values of the parameter described

Prm_Text_Ref:
Type: Unsigned8
Meaning: This reference number has to be the same as the reference number defined in the PrmText description.

PrmText:
Between the key words PrmText and EndPrmText, possible values of a parameter are described. Texts are assigned to these values for symbolic configuring.
Parameters used:
Reference_Number:
Type: Unsigned8
Meaning: Here, the reference of the PrmText description is specified. This reference has to be unique.

Text_Item:
Parameters used:
Prm_Data_Type:
Type: Data_Type (has to correspond to the Data_Type_Name in the parameter description)
Meaning: Here, the value of the parameter is specified that is to be described.
Text:
Type: Visible String (32)
Meaning: Description of the parameter value

Example of Reference Texts:

```
ExtUserPrmData=9 "Threshold reached" ; Text Reference 9
        Bit (4-5)  2  0000-0003          ;  Bits 4 to 5 in the User Octet No. x mean, that a
                                         ;threshold that has been reached is to be displayed.
Prm_Text_Ref=1                  ;        The value ranges from 0..3,  and the default setting  =2
        .                                ;The reference text that is located under PrmText = 1 is
                                         ; displayed in the configuring tool.
        .
PrmText= 1
        Text (0)= "Threshold Limit 100"
        Text (1)= "Threshold Limit 200"
        Text (2)= "Threshold Limit 300"
        Text (3)= "Threshold Limit 400"
EndPrmText
Ext_User_Prm_Data_Ref(x)=9          ;Text Reference 9
                                    ;The (x)th user byte is influenced (is explained in
Chapter 3.1)
```


3 Preface to GSD Revision 3

The development of the PROFIBUS product range also entails enhancements as regards the device properties and features to be described in the GSD files. These developments, in particular the introduction of DP-V1, new physical interfaces and requirements from PROFIBUS PA are the reason for the extension to the GSD file for the present GSD Revision 3.

Examples as templates for own developments are now available at www.profibus.com -> GSD Library.

In general new PROFIBUS devices supporting new features should get a new Ident_Number. But with introducing DP extensions existing devices will be updated. These devices will be compatible regarding the original DP functions. Thats why it is possible, that they keep the same Ident_Number.
In practice this will result into the following scenario:

	Original GSD	New GSD
Original device	OK	2.)
New device	1.)	OK

1.) Case of replacement and maintanance. New devices have to be compatible with original GSD. Otherwise a new Ident_Number has to be assigned to the new device.

2.) With the new GSD new features can be selected for the original device which are not supported. This can cause malfunctions. This is the reason why both GSD can be administrated by the configuration tool. The versions of the GSD must differ in the following items:

- Manufacturer specific characters of the GSD file name
- Keyword Revision
- Keyword Model_Name

Additionally it has to be ensured

- that shipping of old devices will be stopped when new devices are available
- that the assignment from device release to GSD is well described

3.1.1 Who need a GSD Revision 3 ?

With the functionality of DP-V0 only cyclic communication is specified. The acyclic communication between master and slave is only possible when using the DP-V1 extensions.

The new services between master class 1 and slave are the following

Read data set
Write data set
Alarm handling
The new services between master class 2 and slave are the following
Initiate to set up a communication port
Read data block
Write data block
Data Transfer
Abort

Regarding DP-V0 services it was only possible to address the field device. With th eDP-V1 services it is possible to address each module inside of a device.

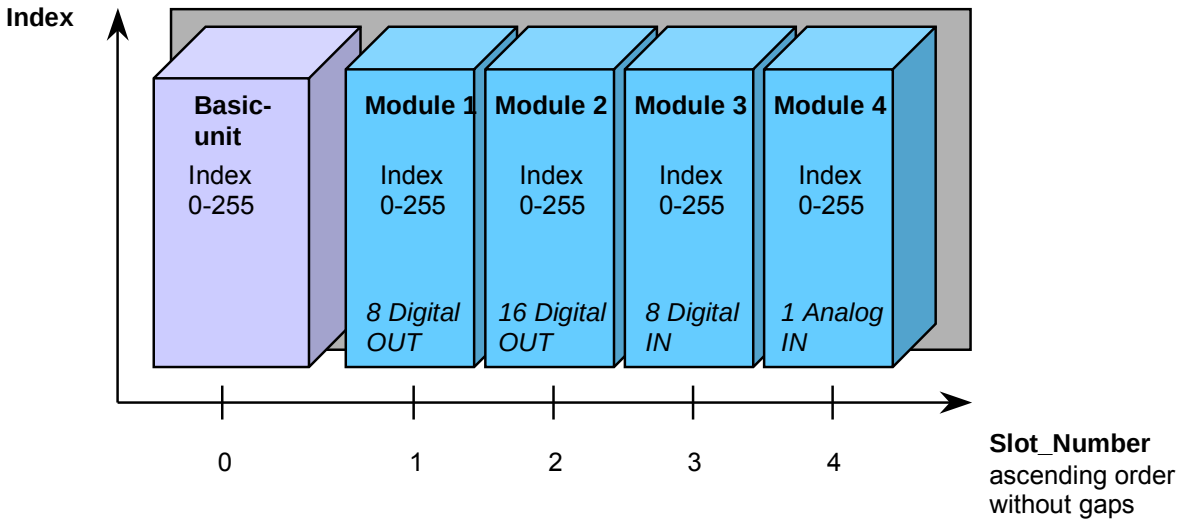
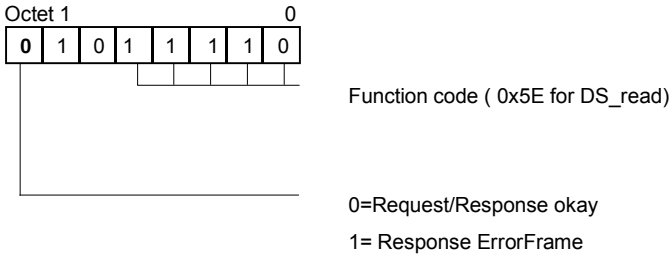


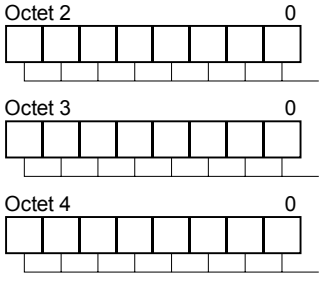
Figure 3.1 Modular slave

The different modules can be directly addressed using slot and index enumeration in the dat unit structure.

Construction of a DS_Read.req telegram for addressing modules via slot and index

SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS	ED
68H	x	x	68H	8x	8x	x	51/33H	51/33H	x ..	x	16H





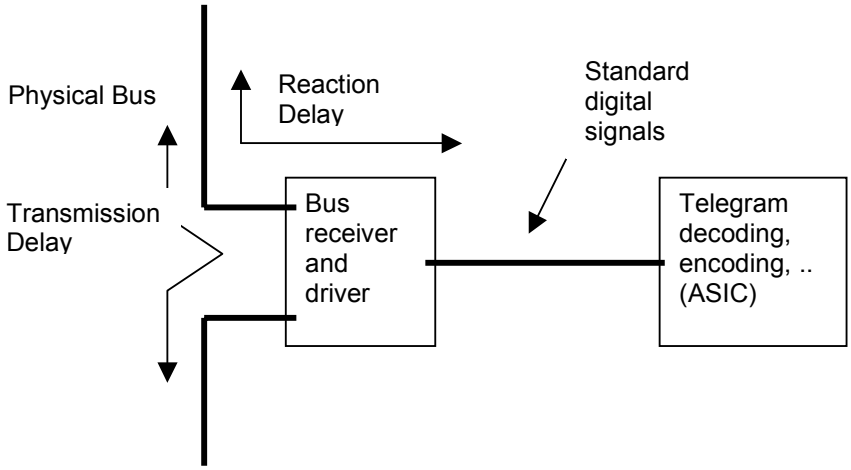
Slot_Number (0...254 for addressing the module)

Index for addressing parts inside the module

Length

3.1.2 Additional keywords for different physical interfaces

Physical_Interface: (O starting with GSD_Revision 3)
This value specifies the execution of the Physical Layers of PROFIBUS. With this parameter it is possible to have devices with more than one physical interface or interfaces different from RS485. If this keyword is not used, then RS485 standard copper is the only supported physical interface. Between the keywords Physical_Interface and End_Physical_Interface, the Transmission_Delays and the Reaction_Delay of a DP slave device are specified, for the physical interface used in the device. The Transmission_Delay defines the delay time for the signal which is to be transmitted through the device. The Reaction_Delay defines the delay of signals processed by the device.



Example: The Transmission_Delay with RS485 is 0, the Reaction_Delay is also 0, because the delay in the driver is lower than 1 bit time.

Especially with optical interfaces these parameters are necessary for the bus timing calculation.

Both the Transmission_Delay and the Reaction_Delay has to be defined for each supported transmission rate. Otherwise the transmission rate is not valid for this physical layer.

The following interfaces are specified:

Type: Unsigned8

0: RS485 Standard Copper
1: Synchronous (31,25 kbit/s, voltage mode, 100Ω wire medium)
2: Optical Plastic
3: Optical Glass
4: Optical HCS
5-127: Reserved
128-255: Manufacturer specific

Parameters Used:

Transmission_Delay_9.6: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.
Example: Transmission_Delay_9.6 = 6;

Transmission_Delay_19.2: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_31.25: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_45.45: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_93.75: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_187.5: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_500: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_1.5M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_3M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_6M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Transmission_Delay_12M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the transmission delay of the device attached to the corresponding physical layer.

Reaction_Delay_9.6: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.
Example: Reaction_Delay_9.6 = 10;

Reaction_Delay_19.2: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_31.25: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_45.45: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_93.75: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_187.5: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_500: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_1.5M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_3M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_6M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

Reaction_Delay_12M: (G starting with GSD_Revision 3)
Type: Unsigned16
Time Base: Bit Time
This parameter specifies the reaction delay of the device attached to the corresponding physical layer.

3.1.3 Additional Keywords for Module Assignment

SlotDefinition: (O only if Modular_Station, starting with GSD_Revision 3)
Between the keywords SlotDefinition and EndSlotDefinition, the possibilities of using the modules within the slots is described.
The modules are referenced by the Module_Reference. The names of the slots are mandatory. The default module will be integrated automatically in the configuration (-telegram). This module can be replaced with one of the permitted modules from the list.
The modules can be encountered using permitted values (8,9,13,...) or using a complete range (17-22).

Slot: (O starting with GSD_Revision 3)
This parameter specifies the modules which can be used in the specified slot

Slot_Number:
Type: Unsigned8
Meaning: Here the number of the slot within the device is specified. The number of the slot must be starting with 1 and arise without gaps. If the SlotDefinition is used, then it´s highly recommended, that the Modul_Offset is also equal 1. Not every slot of a device must be described by this slot definition. Additional modules may appear behind the highest defined Slot_Number.

Slot_Name:
Type: Visible-String (32)
Meaning: Text description of the slot (This means the application function name).

Default_Value:
Type: Unsigned16
Meaning: Default value, Module_Reference of the module used in this slot.

Min_Value:
Type: Unsigned16
Meaning: Minimum value, lowest Module_Reference of the modules which can be used in this slot.

Max_Value:
Type: Unsigned16
Meaning: Maximum value, highest Module_Reference of the modules which can be used in this slot.

Allowed_Values:
Type: Data_Type_Array (256) of Unsigned16
Meaning: Permitted values, list of Module_Reference of the modules which can be used in this slot.

3.1.4 DP Extensions (DP-V1)

The following table illustrates the dependence of GSD keywords regarding the PROFIBUS-DP extensions. Some of the keywords become only valid when other keywords (main selectors for DP-V1 protocol functions) are set TRUE. The right column of the table shows the resulting features and behavior of the device described by the GSD definitions of the left two columns.
In this GSD description the acyclic channel between master class1 and slave has the name MS1 and between master class2 and slave has the name MS2
A configuration tool for the DP extensions has to handle the defined first three byte of the user parameter data itself.
These bytes can also be defined by the known mechanism of the GSD (Ext_User_Prm_Dat_Ref,...), but the configuration tool for the DP extensions overwrites than GSD definitions. At last these bytes can be defined by the keywords for DP extensions, the configuration tool for the DP extensions overwrites the definitions from the user parameter and ext user parameter.

Main Condition	Additional Condition	Conclusion
DP-V1_Slave=0		Device is conform to PROFIBUS-DP. Device can not be operated with the following DP extensions (no acyclic services MS1, no data type support, no DP-V1 specific parameterization, no DP-V1 diagnosis model)
DP-V1_Slave=0	C1_Read_Write_supp = 1 or DP-V1_Data_Types = 1 or Check_Cfg_Mode = 1	**invalid combination**
DP-V1_Slave=1		Device is conform to PROFIBUS-DP extensions. Device supports DP-V1 specific parameterization and DP-V1 diagnosis model. This is an assumption for acyclic services MS1, Data_Types and Check_Cfg_Mode which are supported as stated by the corresponding keywords.

Main Condition	Additional Condition	Conclusion
DP-V1_Slave=1 and C1_Read_Write _supp =0	C1_Max_Data_Len > 0 or C1_Response_Time out > 0 or C1_Read_Write _required = 1 or Diagnostic_Alarm_supp = 1 or Process_Alarm_supp= 1 or Pull_Plug _Alarm_supp= 1 or Status_Alarm_supp = 1 or Update_Alarm_supp = 1 or Manufacturer_Specific_Alarm_supp = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp =1		Device is conform to PROFIBUS-DP extensions and supports MS1 connection. This is an assumption for defining features of the MS1 connection and for Alarm support which are stated by the corresponding keywords.
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Diagnostic _Alarm_supp = 0	Diagnostic_Alarm _required = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Process_Alarm_supp = 0	Process_Alarm _required = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Pull_Plug_Alarm_supp = 0	Pull_Plug_Alarm _required = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Status_Alarm_supp = 0	Status_Alarm _required = 1	**Invalid combination**

Main Condition	Additional Condition	Conclusion
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Status_Alarm_supp = 0	Status_Alarm_required = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Update_Alarm_supp = 0	Update_Alarm_required = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp = 1 and Manufacturer_Specific_ Alarm_supp = 0	Manufacturer_Specific_Alarm_r equired = 1	**Invalid combination**
DP-V1_Slave=1 and C1_Read_Write _supp =1 and Diagnostic_Alarm_supp = 1 or Process_Alarm_supp= 1 or Pull_Plug _Alarm_supp= 1 or Status_Alarm_supp = 1 or Update_Alarm_supp = 1 or Manufacturer_Specific_ Alarm_supp = 1		Device is conform to PROFIBUS-DP extensions and supports MSAC_C1 connection and Alarms. This is an assumption for defining features of the Alarms which are stated by the corresponding keywords.
C2_Read_Write _supp =0	C2_Max_Data_Len > 0 or C2_Response_Timeout > 0 or C2_Read_Write _required =1 or C2_Max_Count_Channels > 0 or Max_Initiate_PDU _Length > 0	**Invalid combination**

Main Condition	Additional Condition	Conclusion
C2_Read_Write _supp =1		Device supports MS2 connection. The support of DP-V1 specific parametrization and DP-V1 diagnosis model is strongly recommended for migration of the whole DP extensions. Features of the MS2 connection are stated by the corresponding keywords.
WD_Base_1ms _supp		This works independent from the other DP extensions. The assumption is that User_Prm_Data_Len > 0 are supported.

3.1.5 Minimum Services of PROFIBUS DP-V1

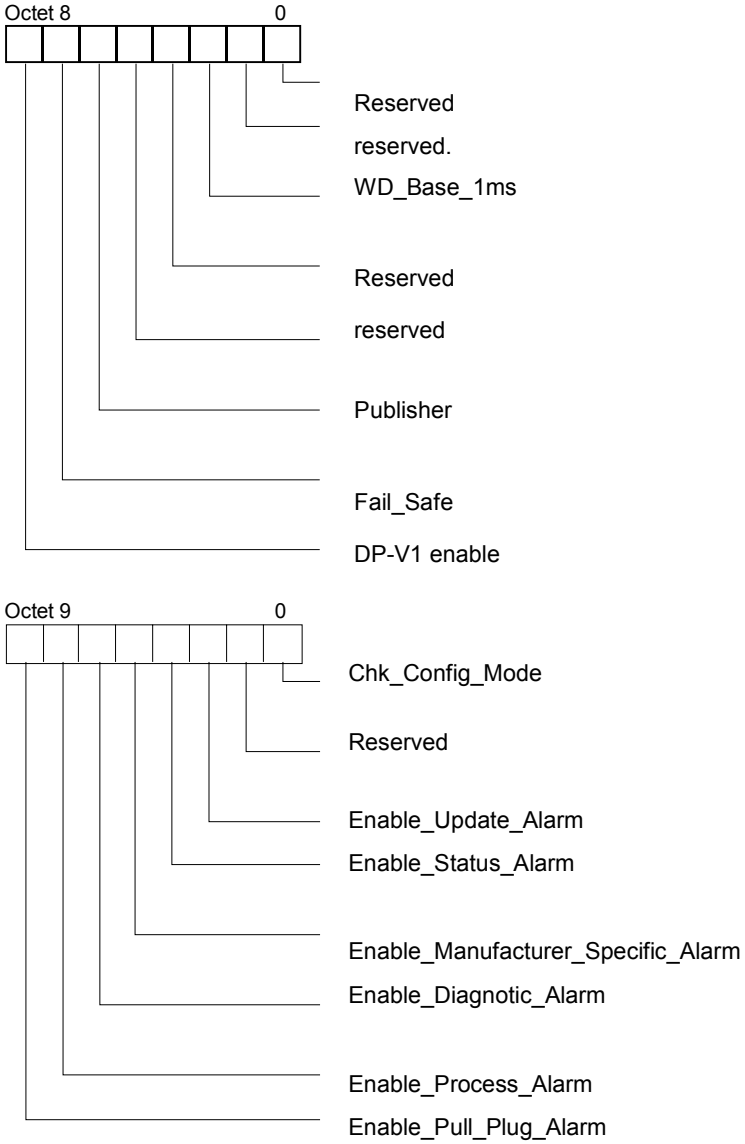
A PROFIBUS device can be named a PROFIBUS DP-V1 device when
at least the **3 user parameter bytes** are supported and the device related diagnosis has changed to alarms or status information (only when supported).

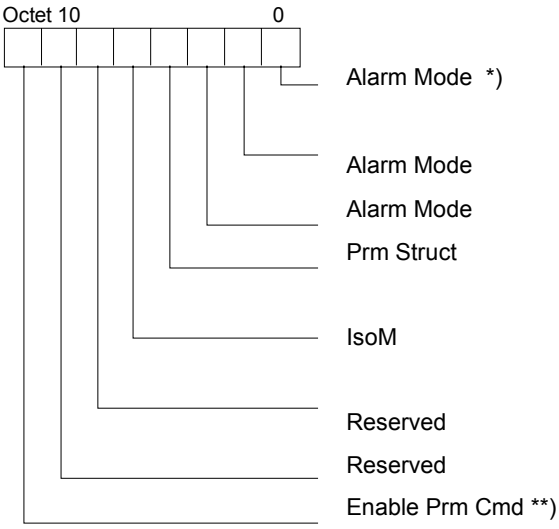
Take care of the entries in the GSD-file (GSD Revision 3 is necessary).
The acyclic communication services have to be defined separately (C1_Read_Write_supp, C2_Read_Write_supp).
It is not absolutely necessary to support the acyclic services (MS1→ acyclic Master Class 1/Slave communication or MS2 → acyclic Master Class 2/Slave communication).

3.1.6 The Meaning of the User Parameterization Bytes (DP-V1)

The following data structure are representing the user parameterization bytes which are transferred from the master to the slave during start up procedure.

The user parameters from 8 .. 10 should only be used for DP-V1 services and should not be used for DP-V0 (exception: byte 8 describes special ASIC behavior of the SPC3)





*) 0 = 1 alarm of each byte,
1 = 2 alarms in total
2 = 4 alarms in total
7 = 32 alarms in total

**) for redundant systems

3.1.7 Slave realated Keywords for DP-V1

DP-V1_Slave (D starting with GSD_Revision 3)
True, if the device uses DP-V1 functionality. This keyword is an extension to "Station_Type" and indicates if the slave operates as an standard DP- or DP-Slave with extended functionality.

The support of the several DP-V1 functionalities is defined in the following function specific keywords.

Type: Boolean (1: True)

C1_Read_Write_supp (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module with extended functionality is supporting the Read and Write services on the MS1-communication relationship.

Type: Boolean (1: True)

C2_Read_Write_supp (D starting with GSD_Revision 3)
The DP-Slave with extended functionality is supporting the Read and Write services on the C2-communication relationship.

Type: Boolean (1: True)

C1_Max_Data_Len: (D starting with GSD_Revision 3)
The parameter specifies the maximum length of user data excluding Function_Num, Slot_number, Index, Length, transferred on the MS1 communication channel.

Type: Unsigned8 (0 .. 240)

C2_Max_Data_Len: (D starting with GSD_Revision 3)
The parameter specifies the maximum length of user data excluding Function_Num, Slot_number, Index, Length, transferred on the MS2 communication channel.
Type: Unsigned8 (0,48 .. 240)

C1_Response_Timeout: (O starting with GSD_Revision 3)
The parameter C1_Response_Timeout represents the efficiency of a DP-Slave with extended functionality. Each DP-Slave with extended functionality has to ensure that the parameter C1_Response_Timeout reaches the smallest value that is possible. By means of this parameter the DP-Slave with extended functionality indicates the maximum time to process an acylic service (read, write, alarm_ack) on the C1-communication relationship.
Type: Unsigned16 (1 .. 65535)
Timebase: 10 ms

C2_Response_Timeout: (O starting with GSD_Revision 3; M if the DP-Slave supports C2_Read_Write_supp, starting with GSD_Revision 4)
The parameter C2_Response_Timeout represents the efficiency of a DP-Slave with extended functionality. Each DP-Slave with extended functionality has to ensure that the parameter C2_Response_Timeout reaches the smallest value that is possible. By means of this parameter the DP-Slave with extended functionality indicates the maximum time to process an acylic service (read, write, Data_Transport) on the C2-communication relationship.
Type: Unsigned16 (1 .. 65535)
Timebase: 10 ms

C1_Read_Write_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires C1_Read_Write services to be accessed.
Type: Boolean (1: True)

C2_Read_Write_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires C2_Read_Write services to be accessed.
Type: Boolean (1: True)

C2_Max_Count_Channels: (D starting with GSD_Revision 3)
The parameter defines the maximal amount of active C2 channels of the DP-V1 Slave.
Type: Unsigned8 (0 .. 49)

Max_Initate_PDU_Length: (D starting with GSD_Revision 3)
The parameter specifies the maximum length of an Initiate Request PDU including the Function_Num to the Resource Manager.
Type: Unsigned8 (0,48.. 244)

Diagnostic_Alarm_supp (D starting with GSD_Revision 3)
The DP device supports Diagnostic_Alarm. A diagnostic alarm signals an event within a slot, for instance overtemperature, short circuit, etc..
Type: Boolean (1: True)

Process_Alarm_supp (D starting with GSD_Revision 3)
The DP device supports Process_Alarm. A process alarm signals the occurrence of an event in the connected process, for instance upper limit value exceeded.
Type: Boolean (1: True)

Pull_Plug_Alarm_supp (D starting with GSD_Revision 3)
The DP device supports Pull_ Plug_Alarm. A pull alarm signals the withdrawal of a module at a slot.
Type: Boolean (1: True)

Status_Alarm_supp (D starting with GSD_Revision 3)
The DP device supports Status_Alarm. A status alarm signals a change in the state of a module, for instance run, stop or ready.
Type: Boolean (1: True)

Update_Alarm_supp: (D starting with GSD_Revision 3)
The DP device supports Update_Alarm. An update alarm signals the change of a parameter in a slot e.g. by a local operation or remote access.
Type: Boolean (1: True)

Manufacturer_Specific_Alarm_supp: (D starting with GSD_Revision 3)
The DP device supports Manufacturer_Specific_Alarm. A manufacturer specific alarm signals an event defined by the manufacturer.
Type: Boolean (1: True)

Extra_Alarm_SAP_supp (D starting with GSD_Revision 3)
Additional to SAP51 it is possible to handle the MSAL_Alarm_Ack via SAP 50 if the Bit Sl_Flag.Extra_Alarm_SAP in the corresponding Slave Parameter Set is set. In this case there may be a higher performance because SAP 50 is used exclusively for the MSAL_Alarm_Ack service and the service can not be delayed by a running MS1_Write or MS1_Read service.
Type: Boolean (1: True)

Alarm_Sequence_Mode_Count: (D starting with GSD_Revision 3)
The DP slave supports the Alarm_Sequence_Mode for alarm handling when this parameter is not 0. If this parameter is set to 0 only the Type Mode is supported by the slave.
The Sequence Mode is an option of the parallel alarm handling.
Several alarms (2 to 32) of the same or different type can be active (unacknowledged) at one time (fixed by the DDLM_Set_Prm service) at the DP-V1 Slave.
Type: Unsigned8 (0, 2 .. 32)

Alarm_Type_Mode_supp: (D starting with GSD_Revision 3;
M if the DP-Slave supports alarms, starting with GSD_Revision 4)
The DP slave supports the Type Mode for alarm handling.
The Type Mode is mandatory if the DP-Slave supports alarms.
Only one alarm of a specific Alarm_Type can be active at one time (fixed by the DDLM_Set_Prm service).
Type: Boolean (shall always be set to 1: True)

Diagnostic_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Process_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Pull_Plug_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Status_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Update_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

Manufacturer_Specific_Alarm_required: (D starting with GSD_Revision 3)
The DP-Slave or a Slave Module requires alarm handling to be accessed.
Type: Boolean (1: True)

DP-V1_Data_Types: (O starting with GSD_Revision 3)
The DP-Slave uses the vendor specific data of the extended identifier format for all modules with extended identifier format for coding of data types.
Type: Boolean (1: True)

WD_Base_1ms_supp: (D starting with GSD_Revision 3)
The DP-Slave supports the time base of 1 millisecond for the watchdog.
Type: Boolean (1: True)

Check_Cfg_Mode: (D starting with GSD_Revision 3)
With this parameter the slave indicates the possibility of a different user specific way to check the Cfg-Data.
This mode is switched on by the "Check_Cfg_Mode" in the DP-V1_Status_2 of the Prm data.
Type: Boolean (1: True)

3.1.8 Slave related Keywords for Data Exchange with Broadcast

Publisher_supp: (D starting with GSD_Revision 3)
The DP-Slave supports the Publisher functionality of Data Exchange with Broadcast.
Type: Boolean (1: True) [1)4]

[1]) since nearly every ASIC can work as a publisher it became necessary to define this keyword already in revision 3. It has to be seen in combination with subscriber_supp defined in revision 4.

4 Preface to GSD Revision 4

In process and production technology, the trend towards more “intelligent” – i.e. more powerful – sensors and actuators is unstoppable. More and more powerful and increasingly fast microprocessors take over tasks from central controllers or permit the utilization of physical effects that were unexploited up to now. Where drives are concerned, for example, the palette of different capacities ranges from speed-controlled units via position-controlled units up to devices that are designed for special technological tasks. Since usually several drives (axes) must run strictly synchronously, there are different requirements on the structure of a closed loop. With “simple” speed-controlled drives, this still means a high control effort in the controller and a high clock synchronization between controller, bus and drive. With increasing performance, more and more information is relocated “downwards” and the drives need more direct data exchange among each other.
Due to the required high performance values, PROFIBUS has now implemented the requirements based upon the base communication, and specified in the new version DP-V2.

Extension

- Isochronous Mode
- Data Exchange with Broadcast
  - Subscriber_Supp, ...
- F-Parameter
- Extended Parameterization
- Extended Diagnostic Description
- Automatically SlotNumber mapping
- Subsystems
  - HMD ...(HART Master Devices)
- Extended Description of MaxTsdr for Optimizing

4.1.1 Who needs GSD Revision 4 ?

To ensure a deterministic behavior on PROFIBUS the Isochron_Mode has been specified. In some applications it is also necessary to have a slave to slave communication without using the bus master for direct data exchange. That has been specified by the Data Exchange Broadcast (DxB) functionality. For safety data transmission (i.e. using PROFIsafe profile) the new keywords are describing the so called F-Parameters.
In the following picture the master sends a data exchange request to the publisher and the publisher sends the response as a broadcast telegram on the bus. In the parameterization telegram the master has to switch on a device as a publisher. Then afterwards this device can operate as a publisher. A subscriber must be loaded with a so called subscriber table where the

complete information is stored from which publisher the information is allowed to be received. Make sure that the selected ASIC will support the functionality.

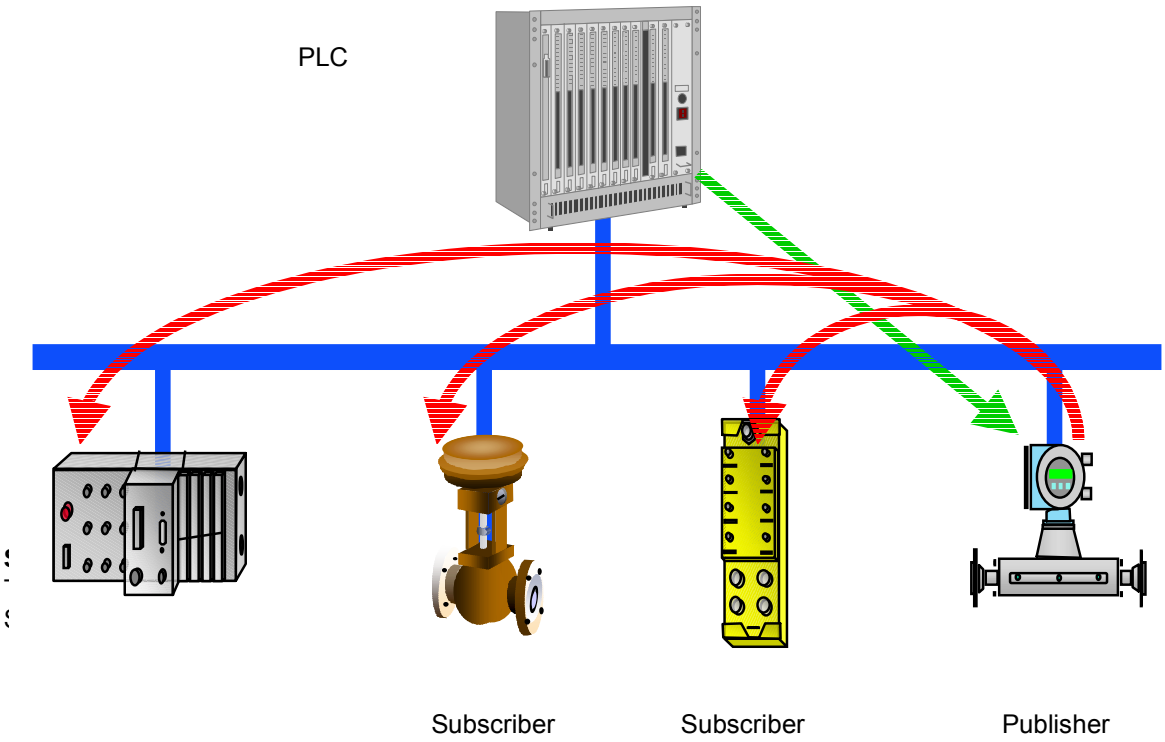


Figure 4.1: Principle of Data Exchange Broadcast

Note: In order to secure the optimized performance of the publisher / subscriber functionality it is necessary to set the MaxTsdr_xx values (§ 3.3.2) according to the actual values of the device.

DXB_Max_Link_Count: (O starting with GSD_Revision 4)
The maximum number of supported links to different Publishers. Has to be unequal 0, if Subscriber_supp = 1.
Type: Unsigned8 (0 - 125)

DXB_Max_Data_Length: (O starting with GSD_Revision 4)
The maximum data length (in one piece) for a supported link to one publisher. Has to be unequal 0, if Subscriber_supp = 1.
Type: Unsigned8 (1 - 244)

Example:

```
; Slave related keywords for DXB - Start
Publisher_supp      = 1;
Subscriber_supp     = 1;
```


DXB_Max_Link_Count = 10;
DXB_Max_Data_Length = 32;
; Slave related keywords for DXB - End

4.1.2 Slave related keywords for Isochronous Mode

Very important note: If a field device is supporting Isochronous Mode it is absolutely necessary also to support DP-V1 Mode. It is **not permitted** to support only the combination DP-V0 (cyclic data exchange and DPV2- Isochronous mode) !!!!!! (DP-V1_Slave = 1 **and** Isochron_Mode_supp= 1);
Also take into account that a field device has to support the structured parameterization when Isochronous mode or Subscriber functionalityis supported.

4.1.3 Who needs Isochronous Mode ?

If a plant needs to fulfill a deterministic behavior (for high speed communication the bus cycle normally is 1 or 2 ms) then all the field devices which are assigned to the part of the plant for deterministic data exchange have to support Isochronous Mode. Other devices which do not support Isochronous mode can also be connected to the same bus.

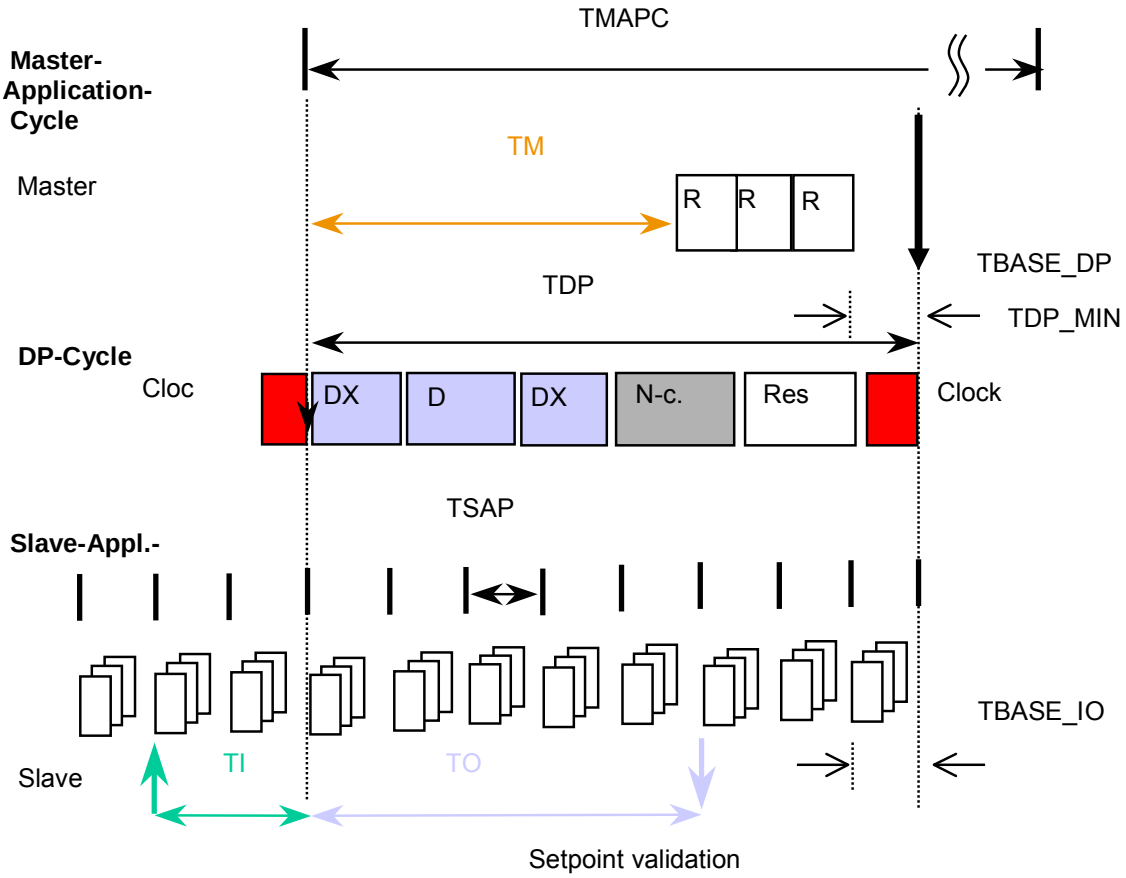


Figure 4.2 Principle of Isochronous Mode

The previous picture shows a PROFIBUS DP bus cycle and the used time bases.

Isochron_Mode_supp: (D starting with Revision 4)
This parameter indicates if the slave supports the Isochron_Mode. If the parameter is set to FALSE, all other Isochron parameters are not significant.
Type: Boolean (1: True)

Isochron_Mode_required: (D starting with Revision 4)
This parameter indicates whether the slave does require the master to support Isochron_Mode. If the parameter is set to TRUE, the slave cannot be operated by a master that does not support Isochron_Mode
Type: Boolean (1: True)

TBASE_DP: (O starting with Revision 4)
Time base of T_{DP}, the DP cycle time, TDP_MIN and TDP_MAX, in units of $^{1}/_{12}$ µs.
Type: Unsigned32, allowed values are 375,750,1500,3000,6000,12000 which correspond to 31.25,62.5,125,250,500,1000us respectively.

TDP_MIN: (G starting with Revision 4)
Minimum of T_{DP}, the DP cycle time, based on TBASE_DP.
Type: Unsigned16, with range from 1 to 2^{16}-1

TDP_MAX: (O starting with Revision 4)
The maximum DP cycle time supported by the DP device in Isochron mode, based on TBASE_DP.
Type: Unsigned16, with range from 1 to 2^{16}-1

T_PLL_W_MAX: (O starting with Revision 4)
The maximum value of the jitter which is acceptable at the device input (RS485 receiver) based on $^1/_{12}$ µs. Mandatory, when Isochron_Mode_supp is enabled.
Type: Unsigned16

TBASE_IO: (O starting with Revision 4)
Time base of T_I and T_O, where T_I is the point in time when the input values are collected and T_O is the point in time when the output values are taken over. The allowed values for the time base are equal to the definition for TBASE_DP (see above).
Type: Unsigned32

TI_MIN: (O starting with Revision 4)
The minimum time based on TBASE_IO that is necessary to get and update the input values of an individual DP Slave.
Type: Unsigned16

TO_MIN: (O starting with Revision 4)
The minimum time based on TBASE_IO that is necessary at the end of the cyclic part of the Isochron DP cycle (T_{DX}) to get and output the output values given in units of TBASE_IO of an individual DP Slave.
Type: Unsigned16

Example:

```
; Slave related keywords for Isochron Mode –
Start
Isochron_Mode_supp     = 1
Isochron_Mode_required = 0
TBASE_DP     = 1500  ; means125µs
TDP_MAX      =  256  ; means  32ms
TDP_MIN      =   16  ; means   2ms
TBASE_IO     = 1500  ; means 125µs
TI_MIN       =    1
TO_MIN       =    1
T_PLL_W_MAX  =   12  ; means 12*1/12 µs = 1µs
; Slave related keywords for Isochron Mode –
End
```

This example means, the device supports Isochron_Mode and can be run by either master whether it supports Isochron_Mode or not. Further, the time base for both, the DP cycle time and the T_I/T_O values is 1500 which corresponds to 125µs. Therefore the minimal DP cycle time necessary for 3 Mbit/s is 16*125µs which equals 2ms, for 6 Mbit/s is 8*125µs which equals 1ms, the maximum cycle time supported by the device is 256*125ms which equals 32ms, the T_I and T_O

can be calculated with 125µs each (T_O 125ms greater than T_{DX}), the maximum value of the jitter is $12*^1/_{12}$ µs which equals 1 µs.

4.1.4 Combinations and Dependencies for Publisher, Subscriber and Isochronous Mode

supported service	***functionality that has to be supported***	***remarks***
DxB Publisher	**standard DP** with additional DP-V1_status_1 parameterization octet	
DxB Subscriber	**DP-V1** with additional diagnosis DXB-Link Status. The DXB Linktable / DXB Subscribertable is loaded. with Check User Prm service or Check Ext UserPrm service into the slave. These additional parameters are described as Structured_Prm_Data in the actual valid PNO specifications	**See note**
Isochronous Mode	**DP-V1 with Fail_Safe***) supported. The IsoMparameters are loaded with Check User Prmservice or Check Ext User Prm service into theslave. These IsoM parameters are described asStructured_Prm_Data in the actual valid PNO specifications	**See note**
Note: DP-V1 supports standard DP with additional 3 parameterization octet (DP-V1_status_1 to DP-V1_status_3) **and** the device specific diagnosis is according Alarm / Status description. of the actual valid PNO specifications, if supported		

*) to recognize when master left DXCHG

4.1.5 The Structured Parameter Data for DP-V2

Substitution Name	**Structure**
Structured parameter data →	Structure_Length, Structure_Type, Slot_Number, reserved (1 octet), User_Prm_Data*
Subscriber table --> Isochron Mode -->	Structured Length, Structure_Type(=7), Slot_Number(=0), reserved (1 octet),Version(=1), (Publisher_Addr, Publisher_Length, Sample_Offset, Dest_Slot_Number,Offset_Data_Area, Sample_Length)*Special Case IsoM Parameter Structured Length, Structure_Type(=4), Slot_Number(=0), reserved (1 octet),Version(=1), T_{BaseDP}, T_{DP}, T_{MAPC}, T_{Base_IO}, T_i, T_o, T_{DX} T_{PLL_W}, T_{PLL_D}.Special Case PrmCmd

Example with Isochronous parameter

Structured length: = 28 / 0x1C
Structure_type: = 4
Slot = 0
reserved = 0
Version = 1
T_{BASE_DP} = Unsigned 32 (375, 750, 1500, 3000, 6000, with default 1500) [$^1/_{12}$µs]
T_{DP} = Unsigned 16 (1 bis 2^{16}-1) [T_{BASE_DP}]
T_{MAPC} = Unsigned 8 (1 bis 14) [T_{DP}]

T_{BASE_IO} = Unsigned 32 (375, 750, 1500, 3000, 6000, with default 1500) [$^1/_{12}\mu s$]
T_I = Unsigned 16 (0;1 bis 2^{16}-1) [T_{BASE_IO}]
T_O = Unsigned 16 (0;1 bis 2^{16}-1) [T_{BASE_IO}]
T_{DX} = Unsigned 32 (0;1 bis 2^{32}-1) [$^1/_{12}\mu s$]
T_{PLL_W} = Unsigned 16 (1 bis 2^{16}-1, mit default 12) [$^1/_{12}\mu s$]
T_{PLL_D} = Unsigned 16 (0 bis 2^{16}-1) [$^1/_{12}\mu s$]

4.2 Slave related Keywords for PROFIsafe Profile

A DP-Slave device that implements a behaviour according to the PROFIsafe profile shall specify its capabilities and the user parameters with the following set of keywords.

F_ParamDescCRC (O starting with GSD_Revision 4)
In order to read the PROFIsafe parameter description safely from the GSD file, 2 byte of CRC code are necessary. The CRC code has to be calculated according to the PROFIsafe guidelines and certified by a registered authority (e.g. TUEV). The value of this parameter will not be transferred to the slave device but is needed to avoid errors during the parametrization with the configuration tool.
Type: Unsigned16

F_Ext_User_Prm_Data_Ref: (O starting with GSD_Revision 4)
Here, a reference to a User_Prm_Data description is specified. The definition of this key word excludes the evaluation of User_Prm_Data. If areas overlap when describing the ExtUserPrmData, the area defined last in the device description block has priority.
Parameters used:
Reference_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of the ExtUserPrmData is defined.

Reference_Number:
Type: Unsigned16
Meaning: This reference number has to be the same as the reference number that is defined in the ExtUserPrmData description.

F_Ext_User_Prm_Data_Const: (O starting with GSD_Revision 4)
Here, a constant part of the ExtUserPrmData is specified. The definition of this key word excludes the evaluation of User_Prm_Data. If areas overlap when describing the ExtUserPrmData, the area defined last in the Generic Station Description (GSD) file has priority.
Parameters used:
Const_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of User_Prm_Data is defined.

Const_Prm_Data:
Type: Octet-String
Meaning: Here, the constants or default selections within the ExtUserPrmData are defined.

4.3 Sample GSD File for a modular F Slave

```
;
;=============================================================================
==
; Sample GSD file for a slave with F-module parameterization based on GSD specification
V4.11.
; This is a demonstration only, no real product.
; The sample slave "F-Device" comprises 3 modules: one with pure F data, one with F and
additional
; standard data and one with standard data only.
; Yellow fields mark areas of CRC secured data: F_ParamDescCRC
; For more and actual GSD file samples see: www.profibus.com
; File name : SAFE0000.gsd
; Revision : 1.0
; Last changes : 16.07.01 12:39
;
;=============================================================================
==
#Profibus_DP
;
; ********************************************
; * Definition of  F parameter value choices is following   *
; ********************************************
;
;Text definition for SILs
;
PrmText                 = 1                     ; Reference number 1,  SIL choices
Text(0) = "SIL 1"
Text(1) = "SIL 2"
Text(2) = "SIL 3"
Text(3) = "SIL 4"
EndPrmText
;
;Text definition for Check/no check
;
PrmText                 = 2                     ; Reference number 2, Check choices
Text(0) = "No Check"
Text(1) = "Check"
EndPrmText
;
;Text definition for CRC-Length
;
PrmText                 = 3                     ; Reference number 3, CRC choices
Text(1) = "2 Byte CRC"
Text(2) = "4 Byte CRC"
EndPrmText
;
;Text definition for Block-ID
;
PrmText                 = 4                     ; Reference number 4,  Mode choices
Text(0) = "F-Host/F-Slave"
EndPrmText
;
;Text definition for Versions
;
```



```
PrmText                 = 5                     ; Reference number 5, Version choices
Text(0) = "PROFIsafe V1"
EndPrmText
;
; ***************************************************
; * Definition of  standard parameter value choices is following   *
; ***************************************************
;
;Text definition for Bit cells
;
PrmText                 = 6                     ; Reference number 6, Bit choices
Text(0) = "Bit = 0"
Text(1) = "Bit = 1"
EndPrmText
;
;Text definition for BitArea
;
PrmText                 = 7                     ; Reference number 7, Bit area choices
Text(0) = "BitArea = 00"
Text(1) = "BitArea = 01"
Text(2) = "BitArea = 10"
Text(3) = "BitArea = 11"
EndPrmText
;
;Text definition for no check
;
PrmText                 = 8                     ; Reference number 8, Check choices
Text(0) = "No Check"
EndPrmText
;
; ***********************************
; * Definition of  F-parameters is following  *
; ***********************************
;
; First part of Ext-User-Prm-Data-Def-List:
;
;User_Prm_Data definition 1
ExtUserPrmData          = 1 "[SlotNumber]"              ; Reference number 1
Unsigned8 1 1-254                                       ; Default = 1,  Max = 254
EndExtUserPrmData
;
;User_Prm_ Data definition 2
ExtUserPrmData          = 2 "F_Dest_Add"                ; Reference number 2
Unsigned16 1 1-65534                                    ; Default = 1, Max = 65534
EndExtUserPrmData
;
;User_Prm_ Data definition 3
ExtUserPrmData          = 3 "F_Source_Add"              ; Reference number 3
Unsigned16 1 1-65534                                    ; Default = 1, Max = 65534
EndExtUserPrmData
;
;User_Prm_ Data definition 4
ExtUserPrmData          = 4 "F_WD_Time (ms)"                    ; Reference number 4
Unsigned16 3 1-65535                                    ; Default = 3, Max = 65535, see chap. 4.5.2
EndExtUserPrmData
;
```



```
;User_Prm_ Data definition 5
ExtUserPrmData      = 5 "F_SIL"                 ; Reference number 5
BitArea(2-3) 1 0-3                              ; Default = 1, Min = 0, Max = 3
Prm_Text_Ref        = 1                         ; Pointer to text definition 1, SIL choices
EndExtUserPrmData
;
;User_Prm_ Data definition 6
ExtUserPrmData      = 6 "F_Check_SeqNr"                 ; Reference number 6
Bit(0) 0 0-1                                    ; Default = 0, Max = 1 within PROFIsafe
release 1
Prm_Text_Ref        = 2                         ; Pointer to text definition 2, Check choices
EndExtUserPrmData
;
;User_Prm_ Data definition 7
ExtUserPrmData      = 7 "F_Check_iPar"          ; Reference number 7
Bit(1) 0 0-0                                    ; Default = 0, Max = 1 within
PROFIsafe release 1
Prm_Text_Ref        = 8                         ; Pointer to text definition 8, Check choices
EndExtUserPrmData
;
;User_Prm_ Data definition 8
ExtUserPrmData      = 8 "F_CRC_Length"          ; Reference number 8
BitArea(4-5) 2 0-3                              ; Default = 2, Min = 0, Max = 3
Prm_Text_Ref        = 3                         ; Pointer to text definition 3, CRC choices
EndExtUserPrmData
;
;User_Prm_ Data definition 9
ExtUserPrmData      = 9 "F_Block_ID"            ; Reference number 9
BitArea(3-5) 0 0-0                              ; Default = 0, Min = 0, Max = 0
Prm_Text_Ref        = 4                         ; Pointer to text definition 4, Mode choices
EndExtUserPrmData
;
;User_Prm_ Data definition 10
ExtUserPrmData      = 10 "F_Par_Version"        ; Reference number 10
BitArea(6-7) 0 0-0                              ; Default = 0, Min = 0, Max = 0
Prm_Text_Ref        = 5                         ; Pointer to text definition 5, Version choices
EndExtUserPrmData
;
;User_Prm_ Data definition 11
ExtUserPrmData      = 11 "F_Par_CRC"            ; Reference number 11
Unsigned16 24269 0-65535                        ; Default = 24265=0x5EC9 Min = 0,
Max = 65535
EndExtUserPrmData

; ***************************************
; * Definition of  standard parameters is following  *
; ***************************************
;
; Second part of Ext-User-Prm-Data-Def-List:
;
;User_Prm_Data definition 11
ExtUserPrmData      = 11 "Header Prm Bit"               ; Reference number 11: see
GSD Spec.
Bit(0) 0 0-1                                    ; Default = 1,  Max = 1
Prm_Text_Ref        = 6                         ; Pointer to text definition 6, Bit value
choices
```



```
EndExtUserPrmData
;
;User_Prm_ Data definition 12
ExtUserPrmData      = 12 "Header Prm BitArea"     ; Reference number 12: see GSD
Spec.
BitArea(1-2) 0 0-3                                ; Default = 0, Min = 0, Max = 3
Prm_Text_Ref        = 7                     ; Pointer to text definition 7, Version choices
EndExtUserPrmData
;
;User_Prm_ Data definition 13
ExtUserPrmData      = 13 "Header Prm Unsigned 16"       ; Reference number 13: see
GSD Spec.
Unsigned16 2000 0-10000                           ; Default = 2000, Min = 0, Max = 10000
EndExtUserPrmData
;
;User_Prm_Data definition 14
ExtUserPrmData      = 14 "Module rel Prm Bit"     ; Reference number 14: see GSD
Spec.
Bit(0) 0 0-1                            ; Default = 1,  Max = 1
Prm_Text_Ref        = 6                     ; Pointer to text definition 6, Bit value
choices
EndExtUserPrmData
;
;User_Prm_ Data definition 15
ExtUserPrmData      = 15 "Module rel Prm BitArea 1"       ; Reference number 15: see
GSD Spec.
BitArea(1-2) 0 0-3                                ; Default = 0, Min = 0, Max = 3
Prm_Text_Ref        = 7                     ; Pointer to text definition 7, Version choices
EndExtUserPrmData
;
;User_Prm_ Data definition 16
ExtUserPrmData      = 16 "Module rel Prm Unsigned 8"  ; Reference number 16: see GSD
Spec.
Unsigned8 6 0-100                                 ; Default = 6, Min = 0, Max = 100
EndExtUserPrmData
;
; ****************************
; * Slave specific data is following  *
; ****************************
;
Vendor_Name         = "PNO WG PROFIsafe"
GSD_Revision        = 4
Model_Name                  = "F-Device"
Revision                    = "1.0"
Ident_Number        = 0x0000                ; ID available from PNO office:
www.profibus.com
Protocol_Ident      = 0                     ; 0 = PROFIBUS-DP
Slave_Family        = 9                     ; = Others
Prm_Struct_supp             = 1                     ; 1 = block structure supported
Station_Type        = 0                     ; 0 = DP-Slave
FMS_supp            = 0                     ; no FMS/DP mixed device
Hardware_Release    = "A1"
Software_Release    = "V1.0"
9.6_supp            = 1
19.2_supp           = 1
93.75_supp          = 1
```



```
187.5_supp              = 1
500_supp                = 1
1.5M_supp               = 1                     ; 9.6 up to 12,000 Kbaud supported
3M_supp                         = 1
6M_supp                         = 1
12M_supp                = 1
MaxTsdr_9.6             = 60
MaxTsdr_19.2            = 60
MaxTsdr_93.75           = 60
MaxTsdr_187.5           = 60
MaxTsdr_500             = 100
MaxTsdr_1.5M            = 150
MaxTsdr_3M              = 250
MaxTsdr_6M              = 450
MaxTsdr_12M             = 800
Redundancy              = 0                     ; redundancy not supported
Repeater_Ctrl_Sig       = 2                     ; TTL
24V_Pins                = 0                     ; no 24 Volt
Bitmap_Device           = "FGENSLVE"            ; icon of  the specific F device: 70 x 40 Pixel
(width x heigth), 16 colors
;
; **********************
; * more slave specific data *
; **********************
;
Freeze_Mode_supp                = 0             ;Freeze-Mode not supported
Sync_Mode_supp                  = 0             ;Sync.-Mode not supported
Auto_Baud_supp                          = 1             ;automatic Transmission rate check
Max_Diag_Data_Len               = 6
Set_Slave_Add_supp              = 1
Min_Slave_Intervall             = 6             ;0.6ms
Modular_Station                         = 1
Max_Module                      = 5             ;max. Nr. of modules to choose from
Max_Input_Len                           = 100
Max_Output_Len                          = 100
Max_Data_Len                    = 200
User_Prm_Data_Len               = 100           ;Length of the total User-Prm-Data
Max_User_Prm_Data_Len           = 100           ;max. length of User-Prm-Data
;
; ****************************************************
; * Module 1: 1 Float Input and 8 Bit "Qualifier",        Safety only   *
; ****************************************************
;
Module = "F_IN_OUT_5 1float 8Bit Quali  F only" 0xC3, 0x83, 0x88, 0x0A, 0x31,0x0A
1
; Start CRC calculation across F parameters (exclude comments)
F_Ext_Module_Prm_Data_Len       = 14            ; see Fig. 4-23
; default values for all parameters
F_Ext_User_Prm_Data_Const(0)    =
0x0E,0x05,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, 0x00,0x00
; keep default F_Prm-Block length
; keep default F_Prm-Block identifier
; keep default Slot number
; keep default Specifier
; keep default F_Prm-Flag1
; keep default F_Prm-Flag2
```



```
F_Ext_User_Prm_Data_Ref(2) = 1          ; see definition 1, Slot, see Figure 4-23 for
offset (2)
F_Ext_User_Prm_Data_Ref(4) = 6          ; see definition 6, F_Prm_Flag1,
F_Check_SeqNr
F_Ext_User_Prm_Data_Ref(4) = 7          ; see definition 7, F_Prm_Flag1,
F_Check_iPar
F_Ext_User_Prm_Data_Ref(4) = 5          ; see definition 5, F_Prm_Flag1, F_SIL
F_Ext_User_Prm_Data_Ref(4) = 8          ; see definition 8, F_Prm_Flag1,
F_CRC_Length
F_Ext_User_Prm_Data_Ref(5) = 9          ; see definition 9, F_Prm_Flag2, F_Block_ID
F_Ext_User_Prm_Data_Ref(5) = 10         ; see definition 10, F_Prm_Flag2,
F_Par_Version
F_Ext_User_Prm_Data_Ref(6) = 3          ; see definition 3, F_Source_Add
F_Ext_User_Prm_Data_Ref(8) = 2          ; see definition 2, F_Dest_Add
F_Ext_User_Prm_Data_Ref(10)     = 4          ; see definition 4, F_WD_Time
F_Ext_User_Prm_Data_Ref(12)     = 11         ; see definition 11, F_Par_CRC
; End CRC calculation across F parameters
F_ParamDescCRC          = 0xABCD    ; "0xABCD" = result of CRC calculation
EndModule
;
; ***************************************************************
; * Module 2: 32 Bit Input and 32 Bit Output,  standard and safety mixed  *
; ***************************************************************
;
Module = "F_IN_OUT_1 32 Bit I/O  mixed F and Std" 0xCA, 0x87, 0x87, 0x05, 0x05,0x05,
0x05, 0x0A, 0x05, 0x05, 0x05, 0x05, 0x0A
2
; Standard Parameters
Ext_Module_Prm_Data_Len    = 7          ; length of standard Prm block of the module
Ext_User_Prm_Data_Const(0)      = 0x07,0x04,0x00,0x00,0x00,0x00,0x00
Ext_User_Prm_Data_Ref(2)   = 14         ; see definition 14, Module rel Prm Bit
Ext_User_Prm_Data_Ref(4)   = 15         ; see definition 15, Module rel Prm BitArea 1
Ext_User_Prm_Data_Ref(4)   = 16         ; see definition 16, Module rel Prm Unsigned
8
; End Standard Parameters
;
; F Parameters
; Start CRC calculation across F parameters
F_Ext_Module_Prm_Data_Len       = 14
; precaution measure is following: default values for all the parameter values
F_Ext_User_Prm_Data_Const(0)    =
0x0E,0x05,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00, 0x00,0x00, 0x00,0x00
; keep predefined parameter values like in module 1
F_Ext_User_Prm_Data_Ref(2) = 1          ; see definition 1, Slot choice 1...254
F_Ext_User_Prm_Data_Ref(4) = 6          ; see definition 6, F_Check_SeqNr
F_Ext_User_Prm_Data_Ref(4) = 7          ; see definition 7, F_Check_iPar
F_Ext_User_Prm_Data_Ref(4) = 5          ; see definition 5, F_SIL
F_Ext_User_Prm_Data_Ref(4) = 8          ; see definition 8, F_CRC_Length
F_Ext_User_Prm_Data_Ref(5) = 9          ; see definition 9, F_Block_ID
F_Ext_User_Prm_Data_Ref(5) = 10         ; see definition 10, F_Par_Version
F_Ext_User_Prm_Data_Ref(6) = 3          ; see definition 3, F_Source_Add
F_Ext_User_Prm_Data_Ref(8) = 2          ; see definition 2, F_Dest_Add
F_Ext_User_Prm_Data_Ref(10)     = 4          ; see definition 4, F_WD_Time
F_Ext_User_Prm_Data_Ref(12)     = 11         ; see definition 11, F_Par_CRC
; End CRC calculation across F parameters
F_ParamDescCRC          = 0xEFGH    ; "0xEFGH" = result of CRC calculation
```



```
EndModule
;
; ****************************************
; * Module 3: 3 Byte Input,      Standard  only  *
; ****************************************
;
Module = "Standard Module  3 Byte Input   SKF" 0x45,0x02,0x05,0x04,0x03,0x02,0x01
3
; Standard Parameters
Ext_Module_Prm_Data_Len    = 7              ; length of standard Prm block of the module
Ext_User_Prm_Data_Const(0)          = 0x07,0x04,0x00,0x00,0x00,0x00,0x00
Ext_User_Prm_Data_Ref(2)    = 14            ; see definition 14, Module rel Prm Bit
Ext_User_Prm_Data_Ref(4)    = 15            ; see definition 15, Module rel Prm BitArea 1
Ext_User_Prm_Data_Ref(4)    = 16            ; see definition 16, Module rel Prm Unsigned
8
; End Standard Parameters
EndModule
```

4.4 Slave related keywords for extended parameterization

Note: Due to a big amount of parameters that can be transferred to a slave (parameters for isochron mode, subscriber table, profile parameters,…) it may happen that not all parameters fit into the parameterization telegram. Therefore an extra parameter telegram (DSAP= 35H) is defined to solve that problem. The support of that extra DSAP is described with the following keywords.

X_Prm_SAP_supp: (D starting with GSD_Revision 4)
Indicates, if the X_Prm_SAP is supported by the slave. Can only be True, if DP-V1_Slave = 1.
Type: Boolean (1: True)

X_Max_User_Prm_Data_Len: (O starting with GSD_Revision 4)
Here, the maximum length of the ExtUserPrmData is specified. The use of this keyword is only allowed if DP-V1_Slave = 1.
Type: Unsigned8 (1 - 244)

X_Ext_Module_Prm_Data_Len: (O starting with GSD_Revision 4)
Here, the length of the associated ExtUserPrmData is defined.
The use of this keyword is only allowed if DP-V1_Slave = 1 and if X_Prm_SAP_supp = 1.
Type: Unsigned8 (1 - 244)

X_Ext_User_Prm_Data_Ref: (O starting with GSD_Revision 4)
Here, a reference to ExtUserPrmData description is specified. If areas overlap when describing the ExtUserPrmData, the area defined last in the Device Description Block has priority.
Parameters used:
Reference_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of the ExtUserPrmData is defined.

Reference_Number:
Type: Unsigned16
Meaning: This reference number has to be the same as the reference number that is defined in the ExtUserPrmData description.

X_Ext_User_Prm_Data_Const: (O starting with GSD_Revision 4)
Here, a constant part of the ExtUserPrmData is specified. If areas overlap when describing the ExtUserPrmData, the area defined last in the Generic Station Description (GSD) file has priority.
Parameters used:
Const_Offset:
Type: Unsigned8
Meaning: Here, the offset within the associated part of ExtUserPrmData is defined.

Const_Prm_Data:
Type: Octet-String
Meaning: Here, the constants or default selections within the ExtUserPrmData are defined.

X_Prm_Block_Structure_supp: (O starting with GSD_Revision 4)
Here, the slave indicates that the block structure of the extended parameterization is supported when using the X_Prm_Service.
Can only be True, if DP-V1_Slave = 1.
Type: Boolean (1: True)

4.5 Slave related Keywords for Subsystems

A Profibus–DP slave device which has gateway capability towards an underlying communication system, also called *subsystem*, can provide a directory which holds DP indexes of the internal buffers representing the addressable Process Data objects. The user needs the information where to find this directory in order to get access to the data buffers representing the underlying communication system. The device manufacturer may provide one directory in slot 0 (this makes sense for a compact slave) or one directory in each slot for a modular slave.
Both keywords are optional, but only one keyword shall be used at the same time. This is because a modular slave could also use slot 0 for this directory, which is then valid for all type of modules. In that case no module specific definition is required.

Subsys_Dir_Index: (O starting with GSD_Revision 4)
The device has capabilities of a gateway towards a subsystem. The index of the subsystem object directory is given by this value. This definition has to appear within the unit definition. In order to decode the directory, the kind of the subsystem shall be specified in brackets.
Type of Index: unsigned 8
Type of Subsystem: unsigned 8, the values standing for:

HART Master Device	*1*
Reserved	***0, 2 .. 127***
User specific	***128 .. 255***

Example:
Subsys_Dir_Index (1) = 15
means, the device is a HART Master Device (HMD) where the HART object directory can be found in slot 0 at index 15.

Subsys_Module_Dir_Index: (O starting with GSD_Revision 4)
The device has capabilities of a gateway towards a subsystem. The index of the subsystem object directory is module specific and is given by this value. The slot corresponds to the module. This definition has to appear within the module definition In order to decode the directory, the kind of the subsystem shall be specified.

Type of Index: unsigned 8

Type of Sybsystem: unsigned 8, the values standing for:

HART Master Device	1
Reserved	0, 2 .. 127
User specific	128 .. 255

Example:

Subsys_Module_Dir_Index (1) = 42

means, the device is a HART Master Device (HMD). The HART object directory of the module where this definition appears can be found in the corresponding slot at index 42.

4.5.1 Formal Description of the Generic Station Description (GSD) File Format

The Formal Description is missing here because it doesn´t improve your understanding. It is described in detail in the GSD Specification. This manual can be downloaded from http://www.profibus.com

5 Relationship of GSD File, Configuring Tool, and DPS2/DPSE or V1SL Software

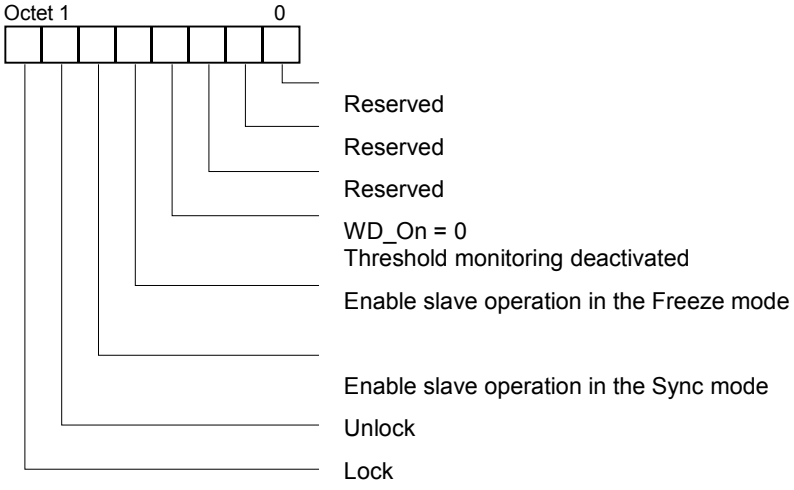
5.1 Parameter Assignment

The performance of a field device can be determined through settings with a DIP switch. Assigning parameters with a handheld provides a more convenient solution. However, with PROFIBUS DP, the device attribute and the performance of the modules defined within a device can also be described once by using the entries in a GSD file. During configuring, the definitive selection of the defined parameters is made, and thus the definitive performance of the field device. When the system is powered up, the bus master sends a parameter assignment message to the configured slaves. The first 7 octets in the parameter assignment message (from the master to the slave) are defined by the system. Starting with Octet 8 up to Octet 244, user-specific information can be defined which is also to be evaluated user-specific. In the user parameters, for example, setting parameters and value ranges can be defined. The slave's response to a parameter assignment message is always "E5H" (positive acknowledgement). Before a field device branches into data exchange, a check is made with a diagnostic query whether the parameters were assigned successfully.

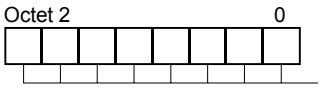
The Parameterization Telegram acc to IEC 61158

SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	**DU..**	FCS	ED
0x68	x	x	x	8x	8x	X	0x3D	0x3E	**x ..**	x	0x16

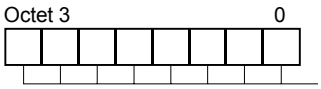
Structure of the DU



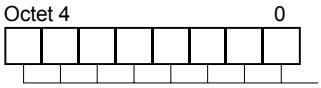
Lock	Unlock	Meaning
0	0	min TSDR and slave-specific parameters may be overwritten
0	1	DP slave is enabled for other masters
1	0	DP slave is disabled for other masters, all parameters are accepted
1	1	DP slave is enabled for other masters



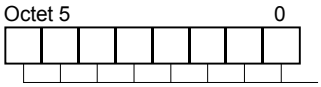
WD_Fact_1 *)



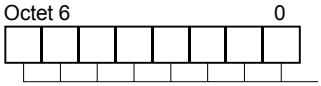
WD_Fact_2: TWD (s) =
10ms * WD_Fact_1 * WD_Fact_2 *)



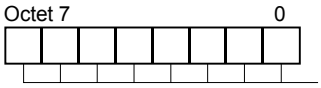
TSDR (time in Tbit indicating when the slave may respond at the at the earliest. 11 Tbit are defined as minimum in the standard. This value must be less than max TSDR.)



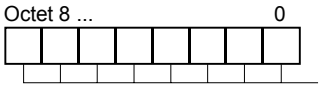
ID high



ID low



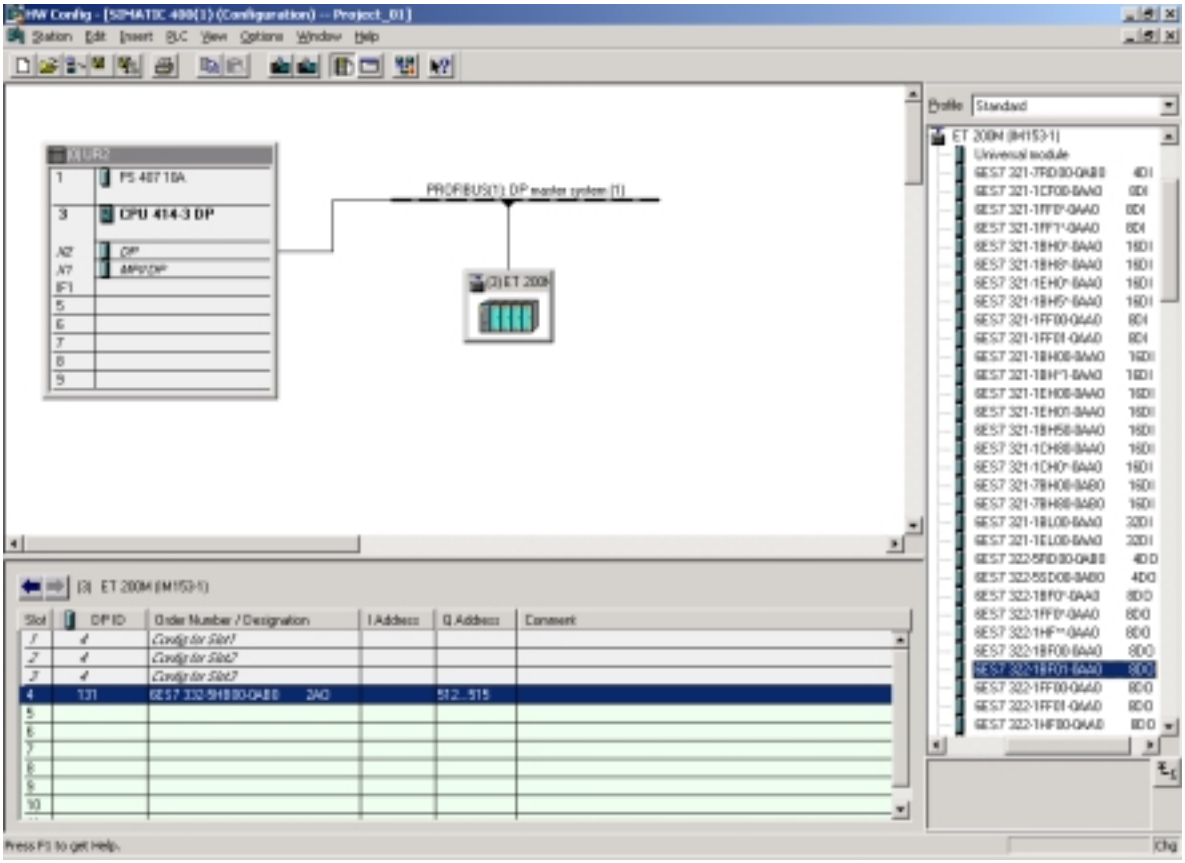
Group_ID



User_Prm_data

To make configuring easier for the configuring engineer -that is, the configuring engineer doesn't have to know the meaning of the bits and bytes for the field device- plain texts can be assigned to the defined bit combinations. Below, a few examples are provided that describe the relationship of GSD file, configuring tool, and DPS2/DPSE software or V1SL software (the DPS2/DPSE/V1SL)is a software available from Siemens that appreciably simplifies controlling the PROFIBUS ASIC SPC3).

The first step in the configuration tool is to select a master. Then the slaves can be graphically connected to the bus. The next figure shows the beginning of a configuration and the list of the defined slaves (HW-Konfig from Siemens) that can be connected to the bus. Those devices that are missing in the current version can be downloaded from www.profibus.com → Libraries → GSD-files.



The next example shows how to do a symbolic parameterization which makes it much easier to understand the meaning of the parameter assignment. You do not have to deal with bits and bytes.

Example 1 General Parameter Assignment (not module-related))

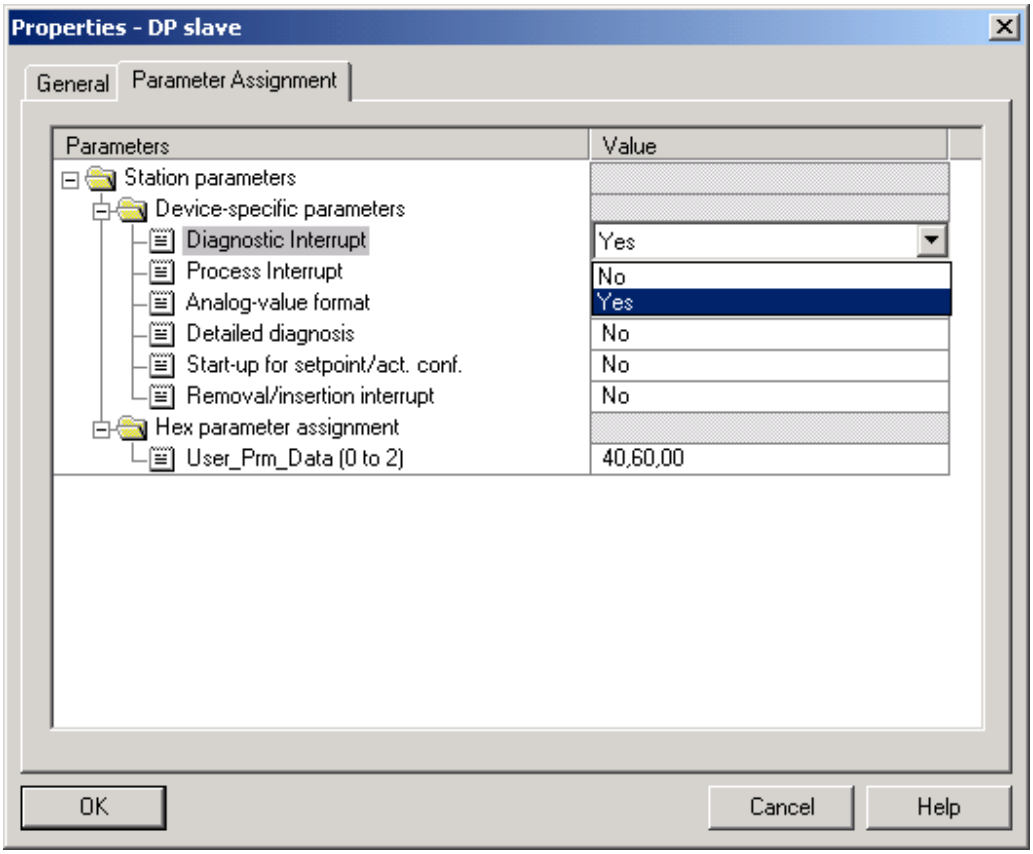
Entry in the GSD file.

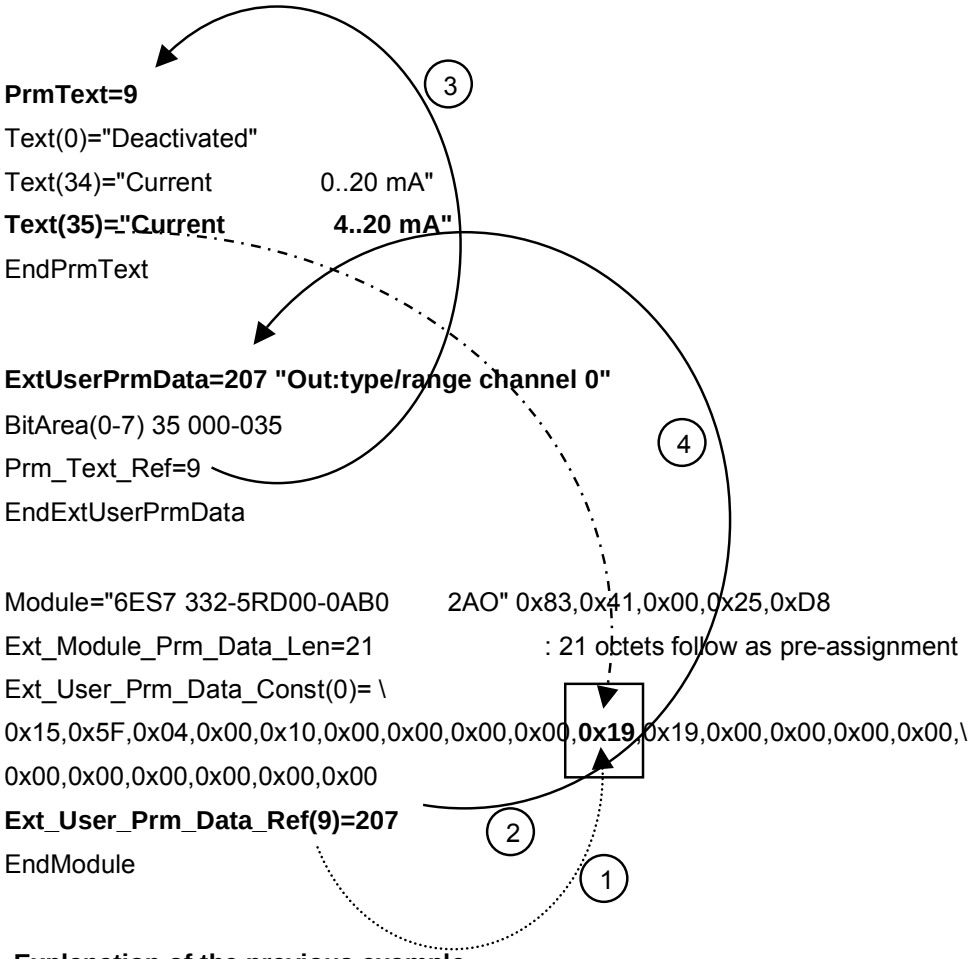
```
PrmText= 1                      ; In the configuring tool, the reference text is displayed,
                                ; under PrmText = 1
        Text (0)= “disable“     ; Value 0 means disable
        Text (1)= “enable“      ; Value 1 means enable
EndPrmText

ExtUserPrmData=1 “Diagnostic Alarm“
        Bit (5)  1  0-1         ; Bit 5 in this reference refers to PrmText=1

                                ; The bit has the value range  0..1
                                ; Bit 5 is preassigned 1
Prm  Text  Ref=1
EndExtUserprmdata               .

Max_User_Prm_Data_Len= 50       ; 50 user-specific parameters maximum can be
                                ; transmitted in the parameter assignment message
Ext_User_Prm_Data_Const (0)= \
0x00,0x01,0x02,0x03             ; The first 4 octets are assigned the values 0, 1, 2, 3
                                ; constantly. The value 3 is to be replaced according to
                                ; the text reference. Bit 5 in Byte 2 = 0
Ext_User_Prm_Text_Ref(2)=1      : Text Reference 1 belongs to Byte 2
```


Explanation of the example above:

In general: The texts that are referenced have to be located before the reference.
Through the instruction Ext_User_Prm_Data_Const(0), the user parameters are preassigned with a constant number sequence.
During configuring, the user wants to specify whether a diagnostic alarm (depending on Octet 2 of the user parameters (Ext_User_Prm_Const(0)) is to be generated. The default selection is that no diagnostic alarm is to be generated. If the user wants to change this, he can symbolically select the response according to previous referencing (**Ext_User_Prm_Text_Ref(2)=1 on ExtUserPrmData=1 "Diagnostic Alarm" on PrmText= 1**).

In the configuring tool (here: HW-Konfig from Siemens), the following is displayed when configuring the user parameters:

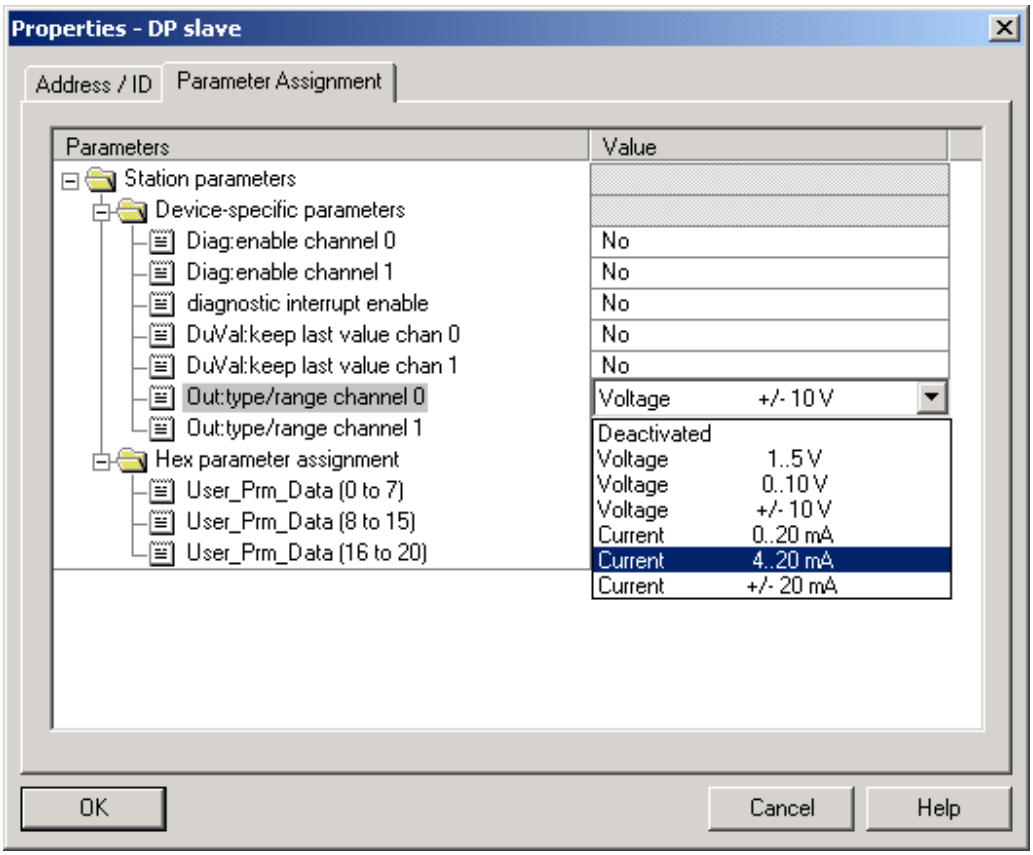




Explanation of the previous example:

In general: The texts that are referenced have to be in front of the reference.
Through the instruction Ext_User_Prm_Data_Const(0), the user parameters are preassigned a constant number sequence.
To the pre-assignment in user parameter Octet 9 according to referencing (**Ext_User_Prm_Data_Ref(9)=207**, count-wise starting with 0 to **ExtUserPrmData=207 "Out:type/range channel 0"** - all values between 0 .. 35 refer to **PrmText=9-**), the final value is to be assigned. Texts are stored for the values 0,34,35. After this step is completed, the configuring tool enters the hex value for **Current 4…20mA** in the user parameters.

The associated part in the configuring tool (here HW-Konfig from Siemens)for example looks like this:



For the PROFIBUS ASIC SPC3, Siemens offers a software that provides a simple interface to the user, and relieves him of the register descriptions of the ASIC. The SPC3 evaluates the standard parameters autonomously. Only if user parameters are defined do they have to be evaluated by the user, and the ASIC has to be informed of the result of the check (.._OK or ..._NOK). The designations in capitals are predefined macros.

When using the **DPS2/DPSE** software for the **PROFIBUS ASIC SPC3**, the relevant code location looks like this, for example:

```
if(DPS2_GET_IND_NEW_PRM_DATA())
{   /*=== New parameter  data ===*/
        UBYTE   SPC3_PTR_ATTR * prm_ptr;
        UBYTE   param_data_len, prm_result;
        UBYTE   ii;

        prm_result = DPS2_PRM_FINISHED;
        do
        { /* Check parameter until no conflict behavior */
                prm_ptr = DPS2_GET_PRM_BUF_PTR();
                param_data_len = DPS2_GET_PRM_LEN();

                /* data_length_netto must be 28 (7 bytes norm + 21 byte user-part) */
                if (param_data_len == 28)
                {
                        if(!user_set_user_prm_values(prm_ptr)) /* call user-specific function */
                        {
                                /* an error was detected in the user-prm-data */
                                prm_result = DPS2_SET_PRM_DATA_NOT_OK();
                        }
                        else
                        {
                                /* user_prm_data is correct, look for range channel 0 */
                                switch(prm_ptr[17])
                                {
                                case 0:
                                        /* deactivated */
                                        break;
                                case 34:
                                        /* current 0..20 mA */
                                        break;
                                case 35:
                                        /* current 4..20 mA */
                                        break;
                                }

                                prm_result = DPS2_SET_PRM_DATA_OK();
                        }
                }
                else
                {
                        prm_result = DPS2_SET_PRM_DATA_NOT_OK();
                }
        } while(prm_result == DPS2_PRM_CONFLICT);
```

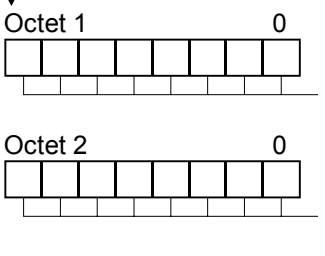

When using the **V1SL Firmware for the PROFIBUS ASIC DPC31**, the relevant code location looks like this, for example:
Checking user parameter data (the first 10 data bytes of the parameter assignment message frame are checked by the DPC31 and V1SL. For that reason, this example only checks for correct length.)

```
/*-------------------------------------------------------------------------*/
/* chk-prm telegram received                                               */
/*------------------------------------------------------------- */

void USR_INT_CODE_ATTR usr_c0_new_prm( V1SL_LL_PRM_PTR prm_ptr, Unsigned8
data_len)
{
   Unsigned8 tmp_prm_result;

   prm_ptr = prm_ptr;

/* todo:    check your user-prm data here */
   /* example begin */
   if( 10 == data_len )
   {
      tmp_prm_result = V1SL_CONTROL_PRM_OK;
   }
   else
   {
      tmp_prm_result = V1SL_CONTROL_PRM_ERROR;
   }
   /* example end */

   if ( V1SL_OK != v1sl_c0_control( tmp_prm_result) )
   {
      sys_fatal_error();
   }
}
```


5.2 Configuring

After the parameter assignment, the field device expects a configuring message. With the configuring data during system power-up, the slave is informed of the number of the input/output data and/or any device-specific configuration. If the transmitted configuration is OK, the slave responds with “E5H”. In the GSD file, a station is described either as a compact station (fixed I/O length can’t be changed), or as a modular station (one or several modules are combined into a station). The data length in both directions, specified during configuring, is monitored by the master as well as the slave at every data exchange. If there is a deviation, the data exchange is cancelled and a diagnostic message is issued.
The configuration of a field device can be described with the general and the special ID format. Below, only an example for the general ID format is provided.

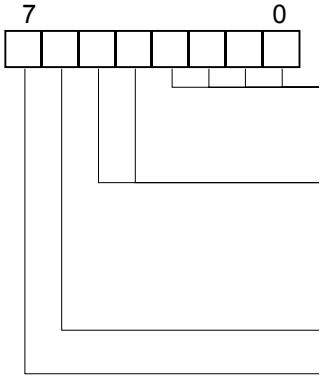
SD	LE	LEr	SD	DA	SA	FC	DSAP	SSAP	DU	FCS	ED
68H	x	x	x	8x	8x	x	62/3EH	62/3EH	x ..	x	16H



Configuring Byte 1

Configuring Byte x

Structure of an Octet in the Configuring Message:



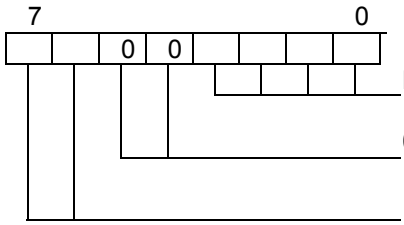
Length of data
00 = 1 Byte / Word
15 = 16 Bytes / Words

Input/Output
00 = spec. ID format
01 = Input
10 = Output
11 = Input/Output

0 = Byte, 1 = Word

0 = Consistency over Byte/Word
1 = Consistency entire length

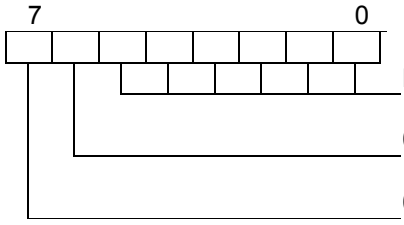
Structure of the special ID format



Length of user specific data

00 – specific ID format

Input/Output – 00 empty space
01 the following byte describes Input data
10 the following byte describes output data
11 the first following byte describes output data, the second one



Length of Input/ output data 00- 1 byte/word, 63- 64 byte/word

0- byte, 1- word

0- no consistent data, 1- consistent over the whole defined

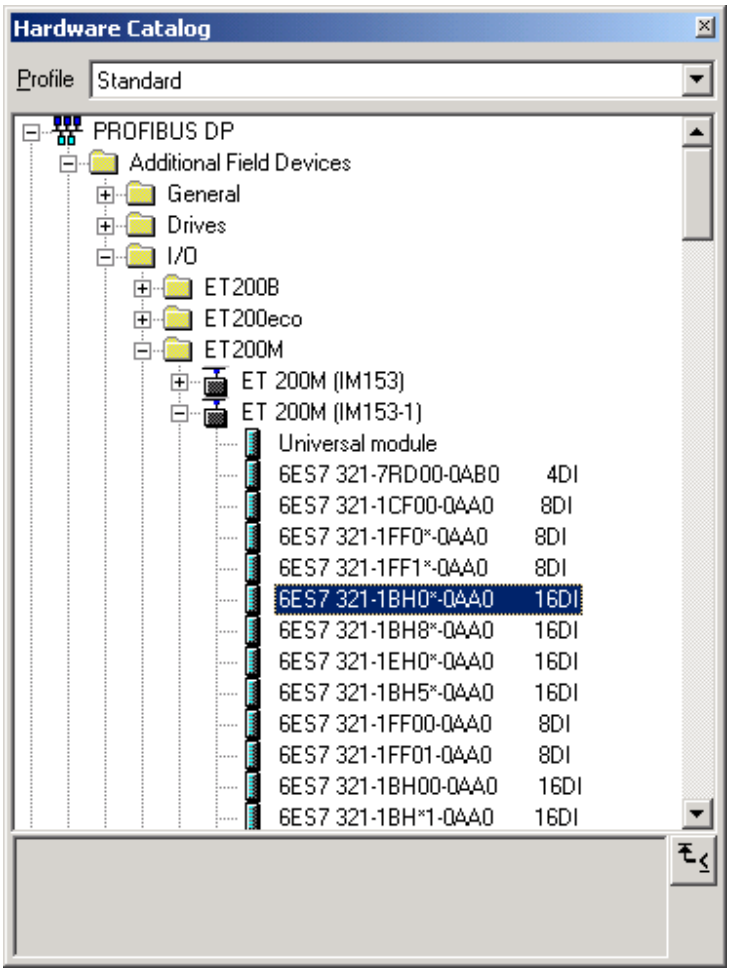
The special configuration format exists always out of at least 2 bytes.

In the GSD file, the corresponding definitions look like this:

Example:

```
Module="6ES7 322-1BF01-0AA0       8DO" 0x83,0x00,0x00,0x2F,0xC8
Ext_Module_Prm_Data_Len=21        ; the module needs 21 data
Ext_User_Prm_Data_Const(0)= \     ; the 21 data is specified as constant values
0x15,0x5F,0x04,0x00,0x10,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,\
0x00,0x00,0x00,0x00,0x00,0x00,0x00
Ext_User_Prm_Data_Ref(2)=28          ; here, the text references for the individual
Ext_User_Prm_Data_Ref(6)=29          ; user parameters are specified
Ext_User_Prm_Data_Ref(7)=30
Ext_User_Prm_Data_Ref(8)=31
Ext_User_Prm_Data_Ref(9)=32
Ext_User_Prm_Data_Ref(10)=33
Ext_User_Prm_Data_Ref(11)=34
```


In the configuring tool (e.g. HW-Konfig from Siemens) , the associated configuration looks like this:



When using the DPS2/DPSE software for the PROFIBUS ASIC SPC3, the relevant code location looks like this, for example:

```
if(DPS2_GET_IND_NEW_CFG_DATA())
   {                                                    /*=== received new configuration ===*/
   UBYTE DPS2_PTR_ATTR * cfg_ptr;
   UBYTE i, config_data_len, cfg_result, result;

   cfg_result = DPS2_CFG_FINISHED;
   result = DPS_CFG_OK;
   do
      {  /* check configuration data until no conflict behavior m*/
      cfg_ptr = DPS2_GET_CFG_BUF_PTR();   /* set pointer to config_data_block  */
config_data_len = DPS2_GET_CFG_LEN();

/* User evaluation*/
/* Checking the received configuration */
```


/* Possibilities of the result of the check */

```
user_io_data_len_ptr = dps2_calculate_inp_outp_len      /* enter buffer organization with */
(cfg_ptr,(UWORD)config_data_len);                       /* the current lengths in SPC3  */


if (( user_io_data_len_ptr -> inp_data_len <= MAX_INP_DATA_LEN ) && (user_io_data_len_ptr -
        > outp_data_len <= MAX_OUTP_DATA_LEN ))
result = DPS_CFG_UPDATE;
          result = DPS_CFG_FAULT ;    */
{
        result = DPS.CFG_UPDATE;
        } else
{
result = DPS_CFG_FAULT;
}
if (result == DPS_CFG_UPDATE)
                {
if (user_io_data_len_ptr != (DPS2_IO_DATA_LEN *)0)
                  {
DPS2_SET_IO_DATA_LEN(user_io_data_len_ptr);
                   }
else
result = DPS_CFG_FAULT;
                  }
              }
switch (result)
              {
case DPS_CFG_OK: cfg_result = DPS2_SET_CFG_DATA_OK();
break;
case DPS_CFG_FAULT: cfg_result = DPS2_SET_CFG_DATA_NOT_OK();
break;
case DPS_CFG_UPDATE: cfg_result = DPS2_SET_CFG_DATA_UPDATE();
break;
              }
           }

} while(cfg_result == DPS2_CFG_CONFLICT);
   }
```

When using the V1SL firmware for the PROFIBUS ASIC DPC31 the relevant code location looks like this, for example:

Checking the configuration data

```
    /*-----------------------------------------------------------------------*/
    /* chk-cfg telegram received                                             */
    /*-----------------------------------------------------------------------*/

    void USR_INT_CODE_ATTR usr_c0_new_cfg( V1SL_LL_UNSIGNED8_PTR cfg_ptr,
    Unsigned8 cfg_len, Unsigned8 app_ready_req, Unsigned8 cfg_mode)
    {
       Unsigned8 tmp_cfg_result;
       cfg_mode = cfg_mode;
       cfg_len  = cfg_len;
```



```
    /* todo: check configuration data here */
    /* example begin */
    {
       if( (cfg_len == 1) &&
          (cfg_ptr[0] == 0x13) )   /* checking for 4 Bytes Input */
       {
          tmp_cfg_result = V1SL_CONTROL_CFG_OK;
       }
       else
       {
          tmp_cfg_result = V1SL_CONTROL_CFG_ERROR;
       }
    }
    /* example end */

    if ( V1SL_OK != v1sl_c0_control( tmp_cfg_result) )
    {
       sys_fatal_error();
    }

    /* todo: set app_ready if your application can provide useful data now;
          do this later, if you need more initialization data first */
    /* example begin */
    if(app_ready_req)
    {
       v1sl_c0_control( V1SL_CONTROL_APP_READY);
    }
    /* example end */
}
#endif
```

After the configuration message, the master once more polls the diagnosis in the slave. If no errors were detected during configuring and parameter assignment, the field device is in data exchange.

5.3 ASIC functionalities

Note: Please take into account that only the slave ASIC DPC31 (not the SPC3) is supporting the complete range of the functionality that is described in the GSD file for slaves.

ASIC Type	DP-V1	Publisher	Subscriber	Master	Slave	IsoM	PLL can be connected
LSPM2	no	Yes	No	no	yes	no	no
SPM2	no	Yes	No	No	Yes	no	no
SPC3	Yes *)	Yes	No	No	Yes	no	no
SPC4-2	yes	Yes	No	No	Yes	no	no
DPC31	yes	Yes	yes	no	Yes	yes	yes
ASPC2	yes	yes	yes	Yes	Yes	yes	Yes

6 Sample Files for GSD File Entries

General Example

```
;=============================================================
; GSD file for product (device name), company (manufacturer)
; Version : (version of the GSD file) - (contact person, phone)
; (General product information; for example,  Sync_mode_supp )
;=============================================================
;General Parameters
;1st line has to start with #Profibus_DP if it is          (M)
;a DP device

#Profibus_DP
;Manufacturer's name, 32 characters max.                   (M)
Vendor_Name = "Manufacturer"
;Product name; 32 characters max.                          (M)
Model_Name = "Product name"
;Version 32 characters max.                                (M)
Revision = "Version 1"
;Ident number of product unsigned 16                       (M)
Ident_Number = 0x8023
;Protocol ID   0=DP device                                 (M)
Protocol_Ident = 0
;Device type   0=Slave, 1=Master(Class1)                   (M)
Station_Type = 0
;DP device type 0=only DP, 1=DP and FMS                    (D)
FMS_supp = 0
;Hardware release  32 characters max.                      (M)
Hardware_Release = "A01"
;Software release  32 characters max.                      (M)
Software_Release = "Z01"
;Here, all supported transmission rates of a
;DP device have to be listed
;Product supports 9.6kBaud                                 (G)
9.6_supp = 1
19.2_supp = 1
93.75_supp = 1
187.5_supp = 1
500_supp = 1
1.5M_supp = 1
3M_supp = 1
6M_supp = 1
12M_supp = 1
MaxTsdr_9.6 = 60
MaxTsdr_19.2 = 60
MaxTsdr_93.75 = 60
MaxTsdr_187.5 = 60
MaxTsdr_500 = 100
MaxTsdr_1.5M = 150
MaxTsdr_3M = 250
MaxTsdr_6M = 450
MaxTsdr_12M = 800
;Redundant transmission engineering  0=No, 1=yes          (D)
Redundancy = 0
```



```
;Signal level (CNTR-P) Pin 4 of the 9-pole SUB-D          (D)
;0-not available, 1-RS485, 2-TTL
Repeater_Ctrl_Sig = 2
;Meaning of the 24V pins of the 9-pole SUB-D              (D)
;0-not available, 1-Input, 2-Output
24V_Pins = 0
;
;--Slave-specific values-----
;
;Freeze Mode is supported   0=No, 1=Yes                  (D)
Freeze_Mode_supp = 0
;Sync Mode is supported   0=No, 1=Yes                    (D)
Sync_Mode_supp = 1
;Autom. transmission rate search is supported  0=No, 1=Yes      (D)
Auto_Baud_supp = 1
;The product can be addressed via the bus
;0=No, 1=Yes                                             (D)
Set_Slave_Add_supp = 0
;Expanded parameterization values (user data length)     (D)
;unsigned 8
User_Prm_Data_Len = 0x05
;Values to be preassigned                                (O)
User_Prm_Data = 0x01,0x02,0x03,0x04,0x05
;Minimum refresh time of a call message                  (M)
;to the slave  unsigned 16  (Basis 100us)
Min_Slave_Intervall = 0x0016
;alternatively, the value can be written in decimals
;Min_Slave_Intervall = 22
```

Example 1: Modular Station

```
;Product description  0=compact device, 1=modular        (D)
Modular_Station = 1
;Max. number of modules that are sent to the slave       (M)
;as configuration   unsigned 8;  in Example 1,
;12 modules maximum can be selected from the available modules
Max_Module = 0x0C
;alternatively, the value can be written in decimals = 12
;Max. number of inputs in bytes  unsigned 8              (M)
Max_Input_Len = 0x10
;alternatively, the value can be written in decimals = 16
;Max. number of outputs in bytes   unsigned 8            (M)
Max_Output_Len = 0x08
;alternatively, the value can be written in decimals = 08
;max. sum of input and output bytes   unsigned 16        (M)
Max_Data_Len = 0x0018
;alternatively, the value can be written in decimals = 24
;Device-related diagnosis in plain text                  (O)
;Bit location in the device-related diagnosis  unsigned 16
;Plain text display 32 characters maximum
Unit_Diag_Bit(0000) = "Slow_Mode active"
Unit_Diag_Bit(0001) = "Wrong_Config_Length"
Unit_Diag_Bit(0002) = "Modul_fault"
Unit_Diag_Bit(0006) = "Power failure"
Unit_Diag_Bit(0009) = "Short circuit to Plus"
```



```
;Module description; each module is inserted between Module - EndModule
;32 characters are available for plain text representation
;The ID is an  octet string
;Module for empty slot
Module= "Leerplatz " 0x00                    <<empty slot>>
EndModule
;Input modules byte-organized
Module = "1 Byte DE " 0x10                   <<DI>>
EndModule
Module = "2 Byte DE " 0x11
Channel_Diag(16) = "Uebertemperatur oder Ueberlast" <<overtemp. or overload>>
Channel_Diag(17) = "Kabelbruch oder Kurzschluss"     <<broken cable or short circuit>>
EndModule
;Output modules byte-organized
Module = "1 Byte DA " 0x20
EndModule
;Input/output modules byte-organized
Module = "1 Byte DE/DA " 0x30                <<DI/DO>>
EndModule
Module = "2 Byte DE/DA " 0x31
EndModule
Module = "2 Byte DE/DA " 0x11,0x21
EndModule
;End GSD file Example 1
```

Example 2: Compact station described in the modular mode (3 possible configurations)

```
.
.
;Product description  0=compact device, 1=modular        (D)
Modular_Station = 1
;Max. number of modules that are sent to the slave        (M)
;as configuration  unsigned 8; in Example 2,
;1 module maximum can be selected
;from the modules that are available
Max_Module = 01
;Maximum number of inputs  unsigned 8                     (M)
Max_Input_Len = 20
;Max. number of outputs  unsigned 8                       (M)
Max_Output_Len = 20
;Max.  sum of the input and output data  unsigned 16      (M)
Max_Data_Len = 40
;Module description; each module is inserted between Module - EndModule
;32 characters are available for plain text display
;The ID is an octet string
;Module Selection 1
Module= "Auswahl 1 20Byte E/A PPO Typ1" 0xF3,0xF3,0xF1  <<selection;  I/O>>
EndModule
;Module Selection 2
Module= "Auswahl 2 16Byte E/A PPO Typ2" 0xF3,0xF3
EndModule
;Module Selection 3
Module= "Auswahl 3 2Byte E, 7Byte A" 0x11,0x26                  <<selection;  I, O>>
EndModule
```


Example 3: Compact Station

```
;Product Description              (D)
Modular_Station = 0               ;0=compact device
Unit_Diag_Area = 0-5
Value(0) = "Fehlerfrei"                          <<faultless>>
Value(1) = "Fehler auf Eingang 0 - 23"           <<error on input …>>
Value(2) = "Fehler auf Ausgang 0 - 15"           <<error on output …>>
Value(3) = "24V ausgefallen"                     <<24V failed>>
Unit_Diag_Area_End
;Module description; each module is inserted between Module - EndModule ;
;32 characters are available for plain text display
;The ID is an octet string
;Modules for compact station
Module= "Kompaktgeraet 16E/16A " 0x11,0x21  <<compact device  16I/16O>>
EndModule
```

Example 4: Compact station with several modules, to be able to assign a text to each module

```
;Product description  0=compact device, 1=modular       (D)
Modular_Station = 0
;Module description; each module is inserted between Module - EndModule
;32 characters are available for clear text display
;The ID is an octet string
;Modules for compact station
;Output module byte-organized
Module = "1 Byte DA " 0x20
EndModule
;Input/output module byte-organized
Module = "1 Byte DE/DA " 0x30                    <<DI/DO>>
EndModule
Module = "2 Byte DE/DA " 0x31
EndModule
```

Example 5: GSD file of the modular station ET 200 X by Siemens

Based on the GSDfile below, an explanation is provided as to how the parameter assignment message can be structured symbolically. The relevant text passages are in bold print, indented, and marked with a reference consisting of a number of a letter. The references are used only as an explanation in this example, and are not included in the original GSD.
A modular module system is to be configured (refer to Figure 2-2). First, the first 3 user parameter bytes are set (refer to Reference A, with the pre-assignment 0x40,0x20,0x00).

The message structure of the user parameters is as follows:

Octet 1 ... Octet 7	**Octet 8**	**Octet 9**	**Octet 10**
x....x	**0x40**	**0x20**	**0x00**

The 2nd byte for setting the diagnosis can be changed. For this, please refer to Reference B (GSD file following). Reference B permits setting the diagnostic alarm. For this, Bit 5 is evaluated with the Default Value 1 and the Value Range 0..1 (Bit(5) 1 0-1), and reference is made to the text according to Reference C. Here, the user can select whether it wants to lock or unlock the diagnostic texts. If diagnostic processing is locked, the message structure looks like this:

Octet 1 ... Octet 7	**Octet 8**	**Octet 9**	**Octet 10**
x....x	**0x40**	**0x00**	**0x00**

As the next step, parameters are assigned to the first module. The analog input module with the designation "6ES7 144-1FB30-0XB0 2AE 10V" is selected.

Starting with Text Reference 1, the user part of the parameter assignment message now looks like this:

Standard	Presetting			Module 6ES7 144-1FB30-0XB0 2AE 10 V								
Octet 1..7	8	9	10	11	12	13	14	15	16	**17**	18	19
x...x	0x40	0x00	0x00	0x09	0x5F	0x05	0x01	0x00	0x00	**0x0A**	0x19	0x19

The presetting is to be modified (Octet 6 = 0x0A). For that reason, refer to Reference 2). In Reference 2), Bits 0..3 are relevant. The default value is 10, and the value ranges from 0..10. The associated symbolic text reference is 3). In Text Reference 3), the user can select the symbol 50 Hz or 60 Hz. Here, the value 60 Hz is selected.

The user part of the parameter assignment message now looks like this:

Standard	Presetting			Module 6ES7 144-1FB30-0XB0 2AE 10 V								
Octet 1..7	8	9	10	11	12	13	14	15	16	**17**	18	19
x...x	0x40	0x00	0x00	0x09	0x5F	0x05	0x01	0x00	0x00	**0x05**	0x19	0x19

Additional modules can be configured in in the same manner.

GSD file for the above example.

```
;=========================================================
; GSD-File  for ET 200X 8DI-2  DP            SIEMENS AG
; MLFB    : 6ES7 141-1BF01-0XB0              <<order number>>
;
; Version : 18.05.98 SX
; File    : SI__803D.GSG
;=========================================================
#Profibus_DP
; <Prm-Text-Def-List>
Reference C)   PrmText=1              ; here, the user selects lock or unlock
               Text(0)="sperren"      <<lock>>
               Text(1)="freigeben"    <<unlock>>
               EndPrmText
PrmText=2
Text(0)="SIMATIC S7"
Text(1)="SIMATIC S5"
EndPrmText
```



```
PrmText=3
Text(0)="sperren"                    <<lock>>
Text(1)="freigeben"                  <<unlock>>
EndPrmText
Reference 3)   PrmText=4
               Text(5)="60 Hz"                    ;here the user selects 60 Hz
               Text(10)="50 Hz"
               EndPrmText
PrmText=5
Text(0)="deaktiviert"
Text(6425)="Spannung +/- 10 V"
EndPrmText
PrmText=6
Text(8995)="Strom (4-DMU) 4 .. 20 mA"        <<current (4wire transducer)>>
Text(9252)="Strom (4-DMU) +/- 20 mA"
EndPrmText
PrmText=7
Text(0)="deaktiviert"                        <<deactivated>>
Text(13107)="Strom (2-DMU) 4 .. 20 mA"
EndPrmText
PrmText=8
Text(0)="deaktiviert"
Text(33410)="RTD-4L Pt 100 Standard"
EndPrmText
PrmText=9
Text(0)="deaktiviert"
Text(6425)="Spannung +/- 10 V"               <<voltage>>
EndPrmText
PrmText=10
Text(8995)="current 4 .. 20 mA"
Text(9252)="current +/- 20 mA"
EndPrmText
; <Ext-User-Prm-Data-Def-List>
Reference B)   ExtUserPrmData=1 " diagnostic alarm "
               Bit(5) 1 0-1
               Prm_Text_Ref=1          → additional reference according to C)
               EndExtUserPrmData
ExtUserPrmData=2 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData
ExtUserPrmData=3 "format representation"
Bit(0) 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData
Reference 2)   ExtUserPrmData=4 "Stoerfrequenzunterdrueckung E0/1"
               BitArea(0-3) 10 005-010
               Prm_Text_Ref=4          → additional reference according to 3)
               EndExtUserPrmData
ExtUserPrmData=5 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData
ExtUserPrmData=6 "Formatdarstellung"
Bit(0) 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData
ExtUserPrmData=7 " interference frequency suppr E0/1"
```



```
BitArea(0-3) 10 005-010
Prm_Text_Ref=4
EndExtUserPrmData
ExtUserPrmData=8 " type of measurement/meas. range E 0/1"
Unsigned16 9252 8995-9252
Prm_Text_Ref=6
EndExtUserPrmData
ExtUserPrmData=9 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData
ExtUserPrmData=10 "format representation"
Bit(0) 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData
ExtUserPrmData=11 "interference freq. suppressed E0/1"
BitArea(0-3) 10 005-010
Prm_Text_Ref=4
EndExtUserPrmData
ExtUserPrmData=12 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData
ExtUserPrmData=13 "format representation"
Bit(0) 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData
ExtUserPrmData=14 "interference freq. suppressed E0/1"
BitArea(0-3) 10 005-010
Prm_Text_Ref=4
EndExtUserPrmData
ExtUserPrmData=15 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData
ExtUserPrmData=16 "format representation"
Bit(0) 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData
ExtUserPrmData=17 "[SlotNumber]"
Unsigned8 1 1-11
EndExtUserPrmData
ExtUserPrmData=18 "Formatdarstellung"
Bit(0) 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData
ExtUserPrmData=19 "output type /output range A 0/1"
Unsigned16 9252 8995-9252
Prm_Text_Ref=10
EndExtUserPrmData
; <Unit Definition List>
GSD_Revision=1
Vendor_Name="SIEMENS"
Model_Name="ET 200X 8DI-2  DP"
Revision="V2.0a"
Ident_Number=0x803D
Protocol_Ident=0
Station_Type=0
Hardware_Release="A1.0"
```


Software_Release="Z1.0"
9.6_supp=1
19.2_supp=1
93.75_supp=1
187.5_supp=1
500_supp=1
1.5M_supp=1
3M_supp=1
6M_supp=1
12M_supp=1
MaxTsdr_9.6=60
MaxTsdr_19.2=60
MaxTsdr_93.75=60
MaxTsdr_187.5=60
MaxTsdr_500=100
MaxTsdr_1.5M=150
MaxTsdr_3M=250
MaxTsdr_6M=450
MaxTsdr_12M=800
Implementation_Type="SPC3"
Bitmap_Device="ET200X1"
; Slave Specification:
OrderNumber="6ES7 141-1BF01-0XB0"
Periphery="ET 200"
MaxResponseDelay=0
Freeze_Mode_supp=1
Sync_Mode_supp=1
Auto_Baud_supp=1
Fail_Safe=1
Min_Slave_Intervall=3
Max_Diag_Data_Len=32
Modul_Offset=1
Slave_Family=3@TdF@ET200X
Modular_Station=1
Max_Module=11
Max_Input_Len=104
Max_Output_Len=104
Max_Data_Len=208
; UserPrmData: Length and Preset:
User_Prm_Data_Len=3
User_Prm_Data=0x40,0x20,0x00
Max_User_Prm_Data_Len=121
Reference A) Ext_User_Prm_Data_Const(0)=0x40,0x20,0x00
Ext_User_Prm_Data_Ref(1)=1 → additional reference according to B)

; Unit Diagnostics:
Unit_Diag_Bit(0024)=”module fault”
Unit_Diag_Bit(0026=<<external error (wire break)”
Unit_Diag_Bit(0028)="no external auxiliary voltage“
Unit_Diag_Bit(0031)="parameterization error module”
; <Module Definition List>
FixPresetModules=1
Module="Config for Slot1" 0x04,0x00,0x00,0xAD,0xC4
Preset=1
EndModule
Module="Config for Slot2" 0x04,0x00,0x00,0x8B,0x40

Preset=1
EndModule
Module="Config for Slot3" 0x04,0x00,0x00,0x8F,0xC0
Preset=1
EndModule
Module="Config for Slot4" 0x43,0x00,0x00,0x9F,0xC9
Preset=1
EndModule
Module="6ES7 141-1BD30-0XA0 4DE" 0x43,0x00,0x00,0x8F,0xC9
EndModule
Module="6ES7 141-1BF30-0XA0 8DE" 0x43,0x00,0x00,0x9F,0xC9
EndModule
Module="6ES7 142-1BD30-0XA0 4DA 0,5A" 0x83,0x00,0x00,0x8F,0xC8
EndModule
Module="6ES7 142-1BD40-0XA0 4DA 2A" 0x83,0x00,0x00,0x8F,0xC8
EndModule
Reference 1) Module="6ES7 144-1FB30-0XB0 2AE 10V" 0x43,0x41,0x00,0x15,0xC3
Ext_Module_Prm_Data_Len=9
Ext_User_Prm_Data_Const(0)=0x09,0x5F,0x05,0x01,0x00,0x00,0x0A,0x19,0x19
Ext_User_Prm_Data_Ref(2)=2
Ext_User_Prm_Data_Ref(5)=3
Ext_User_Prm_Data_Ref(6)=4
EndModule → additional reference according to 2)
Module="6ES7 144-1GB30-0XB0 2AE 20mA" 0x43,0x41,0x00,0x15,0xC3
Ext_Module_Prm_Data_Len=9
Ext_User_Prm_Data_Const(0)=0x09,0x5F,0x05,0x01,0x00,0x00,0x0A,0x24,0x24
Ext_User_Prm_Data_Ref(2)=5
Ext_User_Prm_Data_Ref(5)=6
Ext_User_Prm_Data_Ref(6)=7
Ext_User_Prm_Data_Ref(7)=8
EndModule
Module="6ES7 144-1GB40-0XB0 2AE 4-20mA" 0x43,0x41,0x00,0x15,0xC3
Ext_Module_Prm_Data_Len=9
Ext_User_Prm_Data_Const(0)=0x09,0x5F,0x05,0x01,0x00,0x00,0x0A,0x33,0x33
Ext_User_Prm_Data_Ref(2)=9
Ext_User_Prm_Data_Ref(5)=10
Ext_User_Prm_Data_Ref(6)=11
EndModule
Module="6ES7 144-1JB30-0XB0 2AE Pt100" 0x43,0x41,0x00,0x15,0xC3
Ext_Module_Prm_Data_Len=9
Ext_User_Prm_Data_Const(0)=0x09,0x5F,0x05,0x01,0x00,0x00,0x0A,0x82,0x82
Ext_User_Prm_Data_Ref(2)=12
Ext_User_Prm_Data_Ref(5)=13
Ext_User_Prm_Data_Ref(6)=14
EndModule
Module="6ES7 145-1FB30-0XB0 2AA 10V" 0x83,0x41,0x00,0x25,0xD8
Ext_Module_Prm_Data_Len=9
Ext_User_Prm_Data_Const(0)=0x09,0x5F,0x05,0x01,0x00,0x00,0x00,0x19,0x19
Ext_User_Prm_Data_Ref(2)=15
Ext_User_Prm_Data_Ref(5)=16
EndModule
Module="6ES7 145-1GB30-0XB0 2AA 20mA" 0x83,0x41,0x00,0x25,0xD8
Ext_Module_Prm_Data_Len=9
Ext_User_Prm_Data_Const(0)=0x09,0x5F,0x05,0x01,0x00,0x00,0x00,0x24,0x24
Ext_User_Prm_Data_Ref(2)=17


```
Ext_User_Prm_Data_Ref(5)=18
Ext_User_Prm_Data_Ref(7)=19
EndModule
Module="6GK7 142-2AH00-0XA0 CP 142-2" 0xC2,0x0F,0x0F,0xBC,0xC3
EndModule
Module="3RK1 300-**S00-0AA* 4DX" 0xC2,0x00,0x00,0xCF,0xC9
EndModule
Module="3RK1 300-**S00-1AA* 4DX" 0xC2,0x00,0x00,0xDF,0xC9
EndModule
Module="3RK1 300-0*S10-0AA* 4DX" 0xC2,0x00,0x00,0xEF,0xC9
EndModule
Module="3RK1 300-0*S10-1AA* 4DX" 0xC2,0x00,0x00,0xFF,0xC9
EndModule
```

7 Contact and addresses

Recommendations for improvements
Manfred Popp
ComDeC
Würzburger Strasse 121
90 766 Fürth
Germany
Phone: ++49 911 750 2080
Email Manfredpopp@siemens.com (**attention**: **No dot** between Manfred and Popp)

Certification Center
Werner Götz
ComDeC
Würzburger Strasse 121
90 766 Fürth
Germany
Phone: ++49 911 750 2074
Email Werner.Goetz@siemens.com

Chairman of the GSD working group
Werner Götz
ComDeC
Würzburger Strasse 121
90 766 Fürth
Germany
Phone: ++49 911 750 2074
Email Werner.Goetz@siemens.com

