Operating Instruction Manual

RIF 1788-DPS

FlexLogix/DriveLogix Daugther Card

Edition: 2
Language: English (EN)

Hilscher Gesellschaft für Systemautomation mbH

Web: www.hilscher.com

Contact

Hilscher Europe

Contact	Germany	France
Address	Hilscher Gesellschaft für Systemautomation mbH Rheinstrasse 15 D-65795 Hattersheim	Hilscher France S.A.R.L. 12 rue de 35ème Régiment d'Aviation Miniparc du Chêne F-69500 Bron
Phone	+49 (6190) 99070	+33 (4) 72379840
Phone Sales	+49 (6190) 990790	+33 (4) 72379840
Phone Support	+49 (6190) 990799	+33 (4) 72379840
Fax	+49 (6190) 990750	+33 (4) 78268327
Email	info@hilscher.com	
Email Sales	sales@hilscher.com	
Email Support	hotline@hilscher.com	
Homepage	www.hilscher.com	www.hilscher.com

Contact	Italy	Switzerland
Address	Hilscher Italia s.r.l. Via Grandi, 25 I-20090 Vimodrone (MI)	Hilscher Swiss GmbH Hubelmattstrasse 29 CH-4500 Solothurn
Phone	+39 (02) 25007068	+41 (32) 6236633
Phone Sales	+39 (02) 25007068	+41 (32) 6236633
Phone Support	+39 (02) 25007068	+49 (6190) 990799
Fax	+39 (02) 25029973	+41 (32) 6236632
Email	info@hilscher.it	info@hilscher.com
Email Sales	sales@hilscher.it	sales@hilscher.com
Email Support	supporto@hilscher.it	hotline@hilscher.com
Homepage	www.hilscher.it	www.hilscher.ch

Hilscher North America

Contact	North America
Address	Hilscher North America, Inc. 4320 Winfield Road, Suite 200 Warrenville, Illinois 60555, USA
Phone	(630) 836 8677
Phone Sales	(630) 836 8677
Phone Support	(630) 836 8677
Fax	(630) 836 8010
Email	info-usa@hilscher.com
Email Sales	info-usa@hilscher.com
Email Support	info-usa@hilscher.com
Homepage	www.hilscher.com

Worldwide: Distributors

Please visit our Homepage at
www.hilscher.com

List of Revisions

Index	Date	Chapter	Revisions
1	21.09.03	all	created
2	07.01.04	all	revised

Although this program has been developed with great care and intensively tested, Hilscher Gesellschaft für Systemautomation mbH cannot guarantee the suitability of this program for any purpose not confirmed by us in writing.

Guarantee claims shall be limited to the right to require rectification. Liability for any damages which may have arisen from the use of this program or its documentation shall be limited to cases of intent.

We reserve the right to modify our products and their specifications at any time in as far as this contribute to technical progress. The version of the manual supplied with the program applies.

Table of Contents

1 Introduction

1.1 Functional Description

The communication module RIF 1788-DPS is a slot extension module for a FlexLogix or a DriveLogix Controller from Rockwell Automation. It allows to connect a FlexLogix or a DriveLogix Controller to a PROFIBUS network. The RIF 1788-DPS is a PROFIBUS-DP Slave device. The data exchange between the FlexLogix/DriveLogix controller and the module is done via an I/O process data image by the FlexLogix backplane technology. The following figure shows the data flow between controller and communication module.

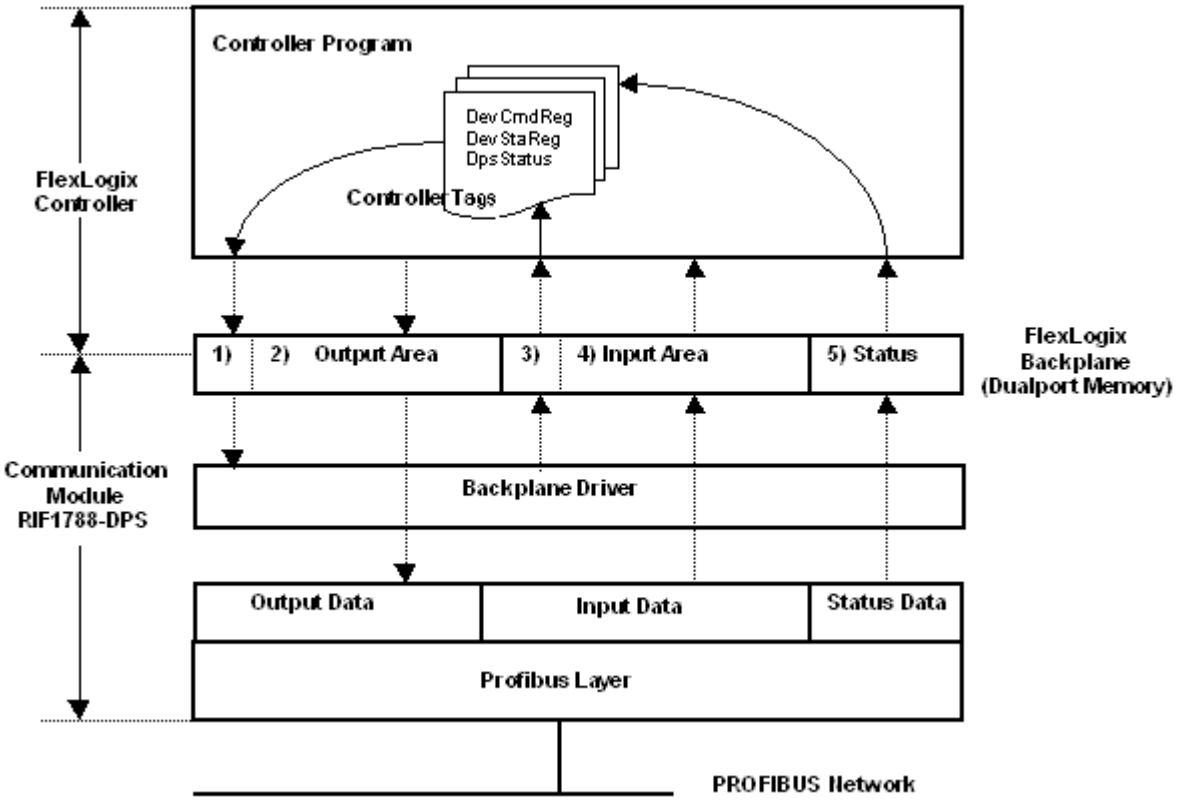

Figure 1: Data flow between controller and communication module

1) Device Command Register 4 Byte

2) Output Data 252 Byte (*)

3) Device Status Register 4 Byte

4) Input Data 252 Byte (*)

5) Status information 128 Byte

(*) The max. packet length is 244 for each In and Out data.

Values with a following ‘h’ are in hexadecimal notation such as 1Eh = 30.
Values without any folloing letter are in decimal notation.

1.2 Product Properties

Characteristic	Description	Note
Description	RIF 1788-DPS	
Type	PROFIBUS-DP Slave	
Characterization	PROFIBUS-DP Slave Communication Module	for Rockwell Controller „FlexLogix“ and „DriveLogix“
Output Data	256 Byte	4 Byte Device Command Register 244 Byte Output Data 8 Byte Reserve
Input Data	256 Byte	4 Byte Device Status Register 244 Byte Input Data 8Byte Reserve
Status Data	128 Byte	32 Byte static status information and 96 Byte variable extended status information
Messaging	PROFIBUS Messaging with CIP Messages	DPV1 Messages (Support with suitable firmware revision)
Configuration and Diagnostic Tool	not required	Configuration is done automatically via PROFIBUS network or via controller Diagnostic via status information in the controller
Programming Tool	RSLogix5000 V12.00	inclusive example projects

Table 1: Product Properties

The FlexLogix/DriveLogix daugther card RIF 1788-DPS was developed by Hilscher GmbH with licensed FlexLogix backplane technology of Rockwell Automation Technologies, Inc.

1.3 Relative Manuals

Manual	Description	Note
1794-IN002E-EN-P	FlexLogix System Installation Instructions	By Rockwell Automation
1794-UM001D-EN-P	FlexLogix System User Manual	By Rockwell Automation
rif1788_cd.pdf	Booklet	By Hilscher GmbH
DPS_PRE.pdf	Protocol Manual PROFIBUS-DP Slave	By Hilscher GmbH **The protocol manual contains partly continued information to some status information and parameter**
PB_OIE.pdf	Operating Instruction Manual SyConPB (System Configurator PROFIBUS)	By Hilscher GmbH

Table 2: Relative Manuals

1.4 Example Projects

Project name	Description	Note
RIF_1788_DPS.ACD	Fundamental example project for the module configuration and PROFIBUS I/O data exchange	Located on the System Software Partner Products CD
RIF_1788_DPS_config. ACD	Example project for configuration of the DP-Slave via controller application	Located on the System Software Partner Products CD

Table 3: Example Projects

1.5 Firmware

If it‘s necessary to update the firmware, this has to be done via the square post connector X5 on the communication module with a special serial cable. Contact your local distributor. The name of the firmware file for this communication board is „DPS.E25“

The firmware of the FlexLogix/DriveLogix daugther card was developed and tested with following controller firmware revisions.

Firmware RIF 1788-DPS	Firmware FlexLogix	Firmware DriveLogix
V1.000	V12.23	V12.09 V12.16

Table 4: Firmware

2 Configuration and Start-Up

2.1 General

In the following sections the individual steps for starting up of the module are described. Install the PROFIBUS Slave module into a empty slot in the FlexLogix/DriveLogix controller. Please follow the installation guidlines for daugther cards in FlexLogix/DriveLogixs systems (see manual: 1794-IN002E-EN-P).

The configuration and parameterization of the module is carried out in two steps

- Configuration of the module in a FlexLogix/DriveLogix project of the RSLogix5000 programming tool
- Creating the data objects and the ladder diagram in RSLogix5000

Note: The easiest way to startup the module in a RSLogix5000 project is to use the “RIF_1788_DPS.ACD” example project. In this example the slot number in the configuration dialog of the module may have to be adjusted.

2.2 Configuration in RSLogix 5000

2.2.1 Module selection in RSLogix 5000

Create a new project in RSLogix5000 with a FlexLogix or DriveLogix controller. First the module must be defined in the project. To insert the RIF 1788-DPS module into the configuration right click on the I/O configuration of the controller project. Select the menu option **New Module ...** in the context menu of the I/O configuration.

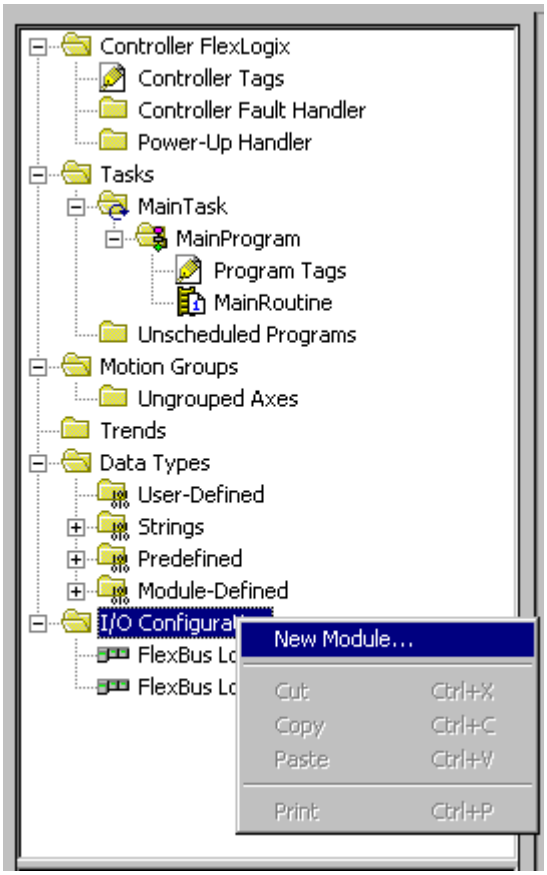

Figure 1: Insert new module

The following dialog is shown for selection of the module.

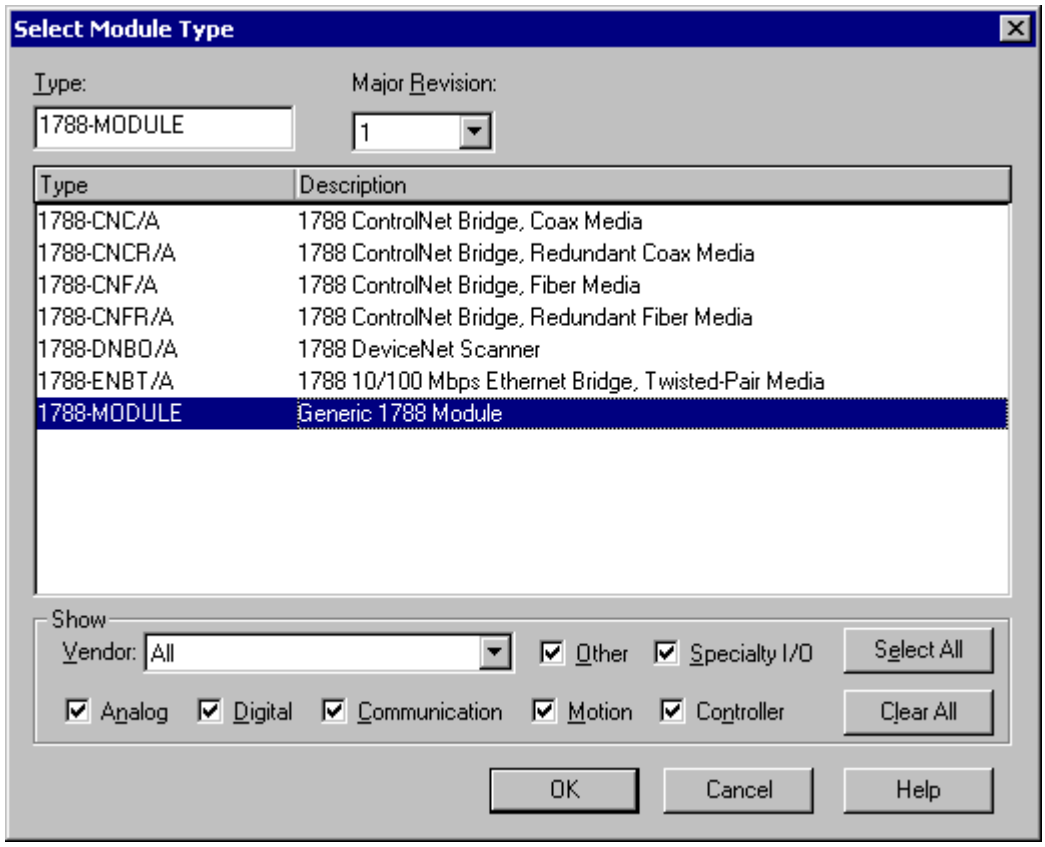

Figure 2: Select Module Type

Select the **"1788-MODULE - Generic 1788 Module"** from the selection list with **OK**.

2.2.2 Module Properties 1

The communication parameters of the module are to be set in the dialog shown below:

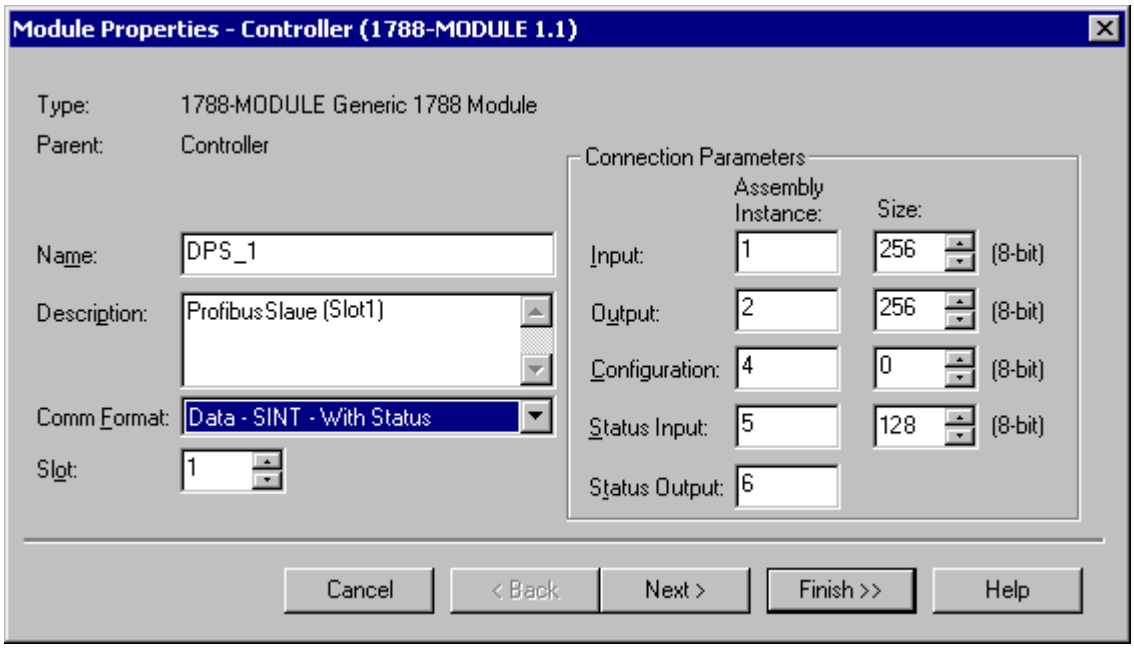

Figure 3: Module Properties 1

Determine a name and a short description for the module. Select the slot number in which the module is installed in the controller. Select **Data - SINT - With Status** as the **Comm_Format**. Set the connection parameters as they are shown in the dialog.

Connection Parameter	Assembly Instance	Size
Input	1	256
Output	2	256
Configuration	4	0 **or** 64 **(*)**
Status Input	5	128
Status Output	6	

Table 5: Connection Parameter

(*)Note: If the module has to be configured automatically via PROFIBUS, enter **0** (byte) for the parameter „Configuration Size“. If the module has to be configured via controller application, enter **64** (byte) for the parameter „Configuration Size“. If these parameter do not correspond to the template values, then the controller cannot build up communication with the module.

Select **Next >>** for the next configuration dialog.

2.2.3 Module Properties 2

The Requested Packet Interval (RPI) is set in the following dialog. In this time interval, the I/O data between module and controller are exchanged. Values in 1.0 ms steps are possible for the PROFIBUS module. The smallest possible value is 2.0 ms. For intermediate values; the module automatically adjusts itself to the nearest multiple of 1.0 ms.

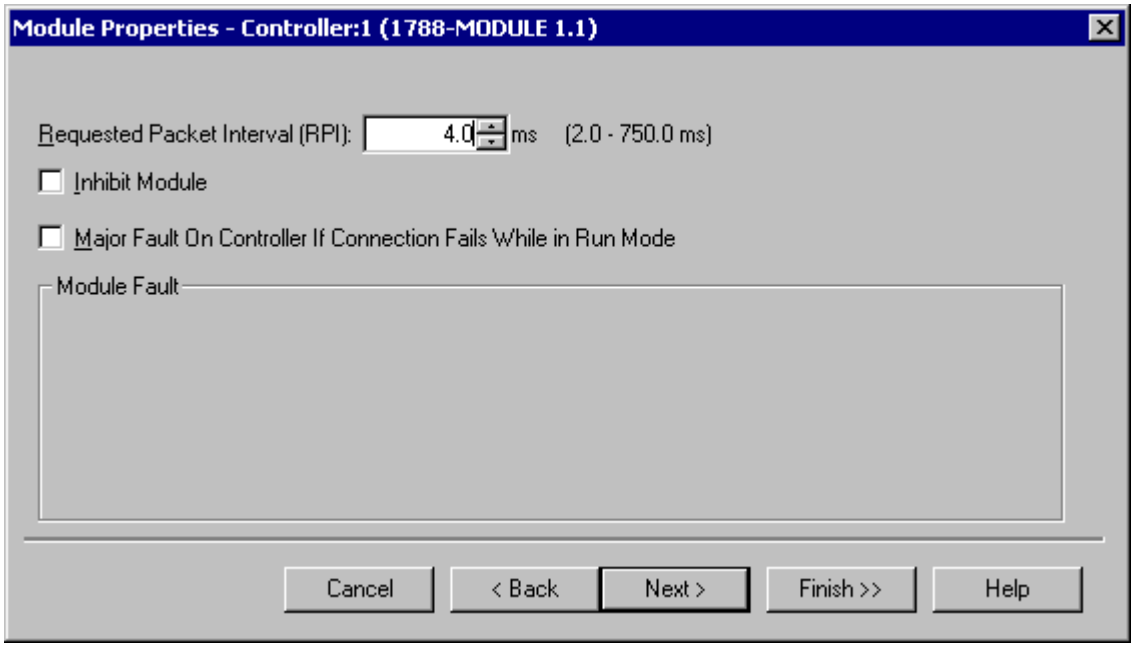

Figure 2: Module Properties 2

End the configuration of the module with **Finish>>**.

2.3 Data Objects

2.3.1 General

Various data objects between the module and the controller are exchanged as Byte array via the input / output and the status region. In order to access the data in a structured manner, they must be laid down with the data type editor in the RSLogix5000 programming tool.

Note: These data types can be taken over into the project from the "RIF_1788_DPS.ACD" example project by means of Copy and Paste

2.3.2 Device Status Register

By means of the 256 Byte input region, the input data and a “Device Status Register” are transferred from the module to the controller. The Device State Register has the size 4 Bytes and it always lies in the first 4 Bytes of the input region. It has the following structure:

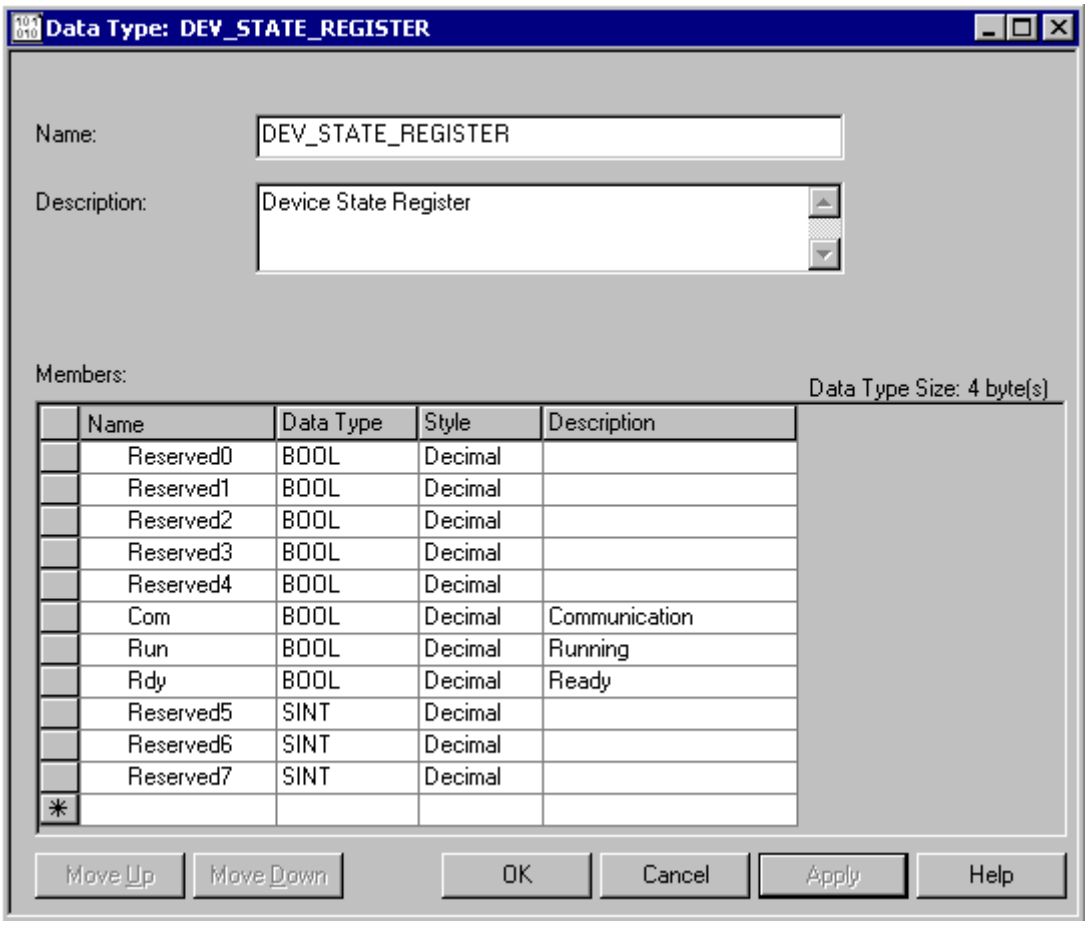

Figure 4: Device Status Register

The Device State Register contains general information of the module and its communication state.

Byte 3	Byte 2	Byte 1	Byte 0
Reserved	Reserved	Reserved	Status bits

Table 6: Device State Register Byte

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Rdy	Run	Com	Reserved	Reserved	Reserved	Reserved	Reserved

Table 7: Status Bits

- **Rdy Bit**

  When this Bit is set, the module is operational. The Rdy Bit is always set in the application program. Otherwise a system error has occured and the communication between controller and module is generally not possible.

- **Run Bit**

  When the Run Bit is set, the module is ready for communication. Otherwise an initialization error or an incorrect parameterization has occured. Further diagnostic is carried out with the LED's of the module or the status information.

- **Com Bit**

  When this Bit is set, the communication is started and the module is in cyclic data exchange with the connected Master.

2.3.3 Input Data

By means of the 256 Byte input region, the input data are transferred from the module to the controller. The input data always starts at the 5th Byte in the input region (Data_I[4]). The actually used size of the input area depends on the configuration of the slave adaptor. The actually used size of the input area is shown in the status area.

2.3.4 Device Command Register

The Device Command Register and the output data are transferred from the controller to the module via 256 Byte output area. The Device Command Register always lies in the first 4 Bytes of the output region. Then the PROFIBUS output data follows. The Device Command Register has the following structure:

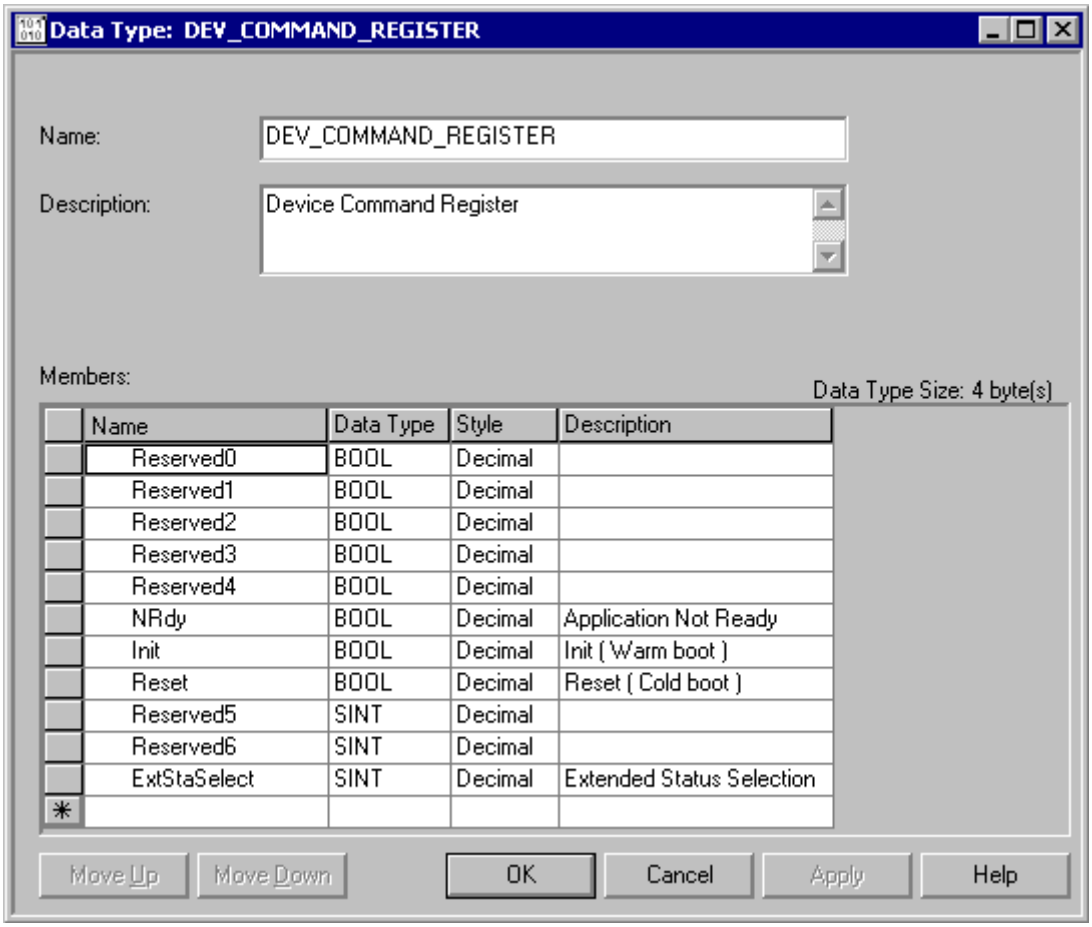

Figure 5: Device Command Register

The Device Command Register controls the module.

Byte 3	Byte 2	Byte 1	Byte 0
ExtStaSelect	Reserved	Reserved	Command bits

Table 8: Device Command Register

Command bits:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reset	Init	NRdy	Reserved	Reserved	Reserved	Reserved	Reserved

Table 9: Command Bits

- **Reset**

  With this Bit, the user program can carry out a Reset (Cold Start) of the module.

Attention: With the Reset the communication to the bus will be interrupted.

- **Init**

  With this Bit, the user program can carry out a Reset (Warm Start) of the module. This function is presently not implemented.

- **NRdy (NotReady)**

  With this Bit, the user program can start or stop the communication. If this Bit is set by the controller application, the cyclic data exchange to the master will be stopped and the slave indicates static diagnosis to the master. In this way the controller program is able to start the communication from the Slave module to the Master.

Note: The module first starts the communication when the controller is in the RUN mode and the NotReady Bit is deleted.

Controller RUN Mode	NotReady Bit (controlled by the user program)	Start of communication with the Master
0	0	No
0	1	No
1	0	Yes
1	1	No

Table 10: Start Communication

ExtStaSelect:

With this byte the controller application is in the position to choose the extended status information, which is transferred. This is displayed in the status area. Valid values are described in section *Status Information* on page *21*.

2.3.5 Output Data

By means of the 256 Byte output region, the output data are transferred from the controller to the module. The output data always start at the 5th Byte in the output region (Data_O[4]). The actually used size of the output area is depending on the configuration of the slave module. The actually used size of the output region is shown in the status region of the Slave.

2.3.6 Status Information

A 128 Byte state field is transferred to the user program via the state region. This field contains information on the status of the slave module. It always begins in Byte 0 of the state region. The status information is assembled by 32 Byte static information and 96 Byte variable extended status information. The selection of the extended status is controlled by the application program. The displayed status is controlled by the command "ExtStaSelect“ byte in the Device Command Register.

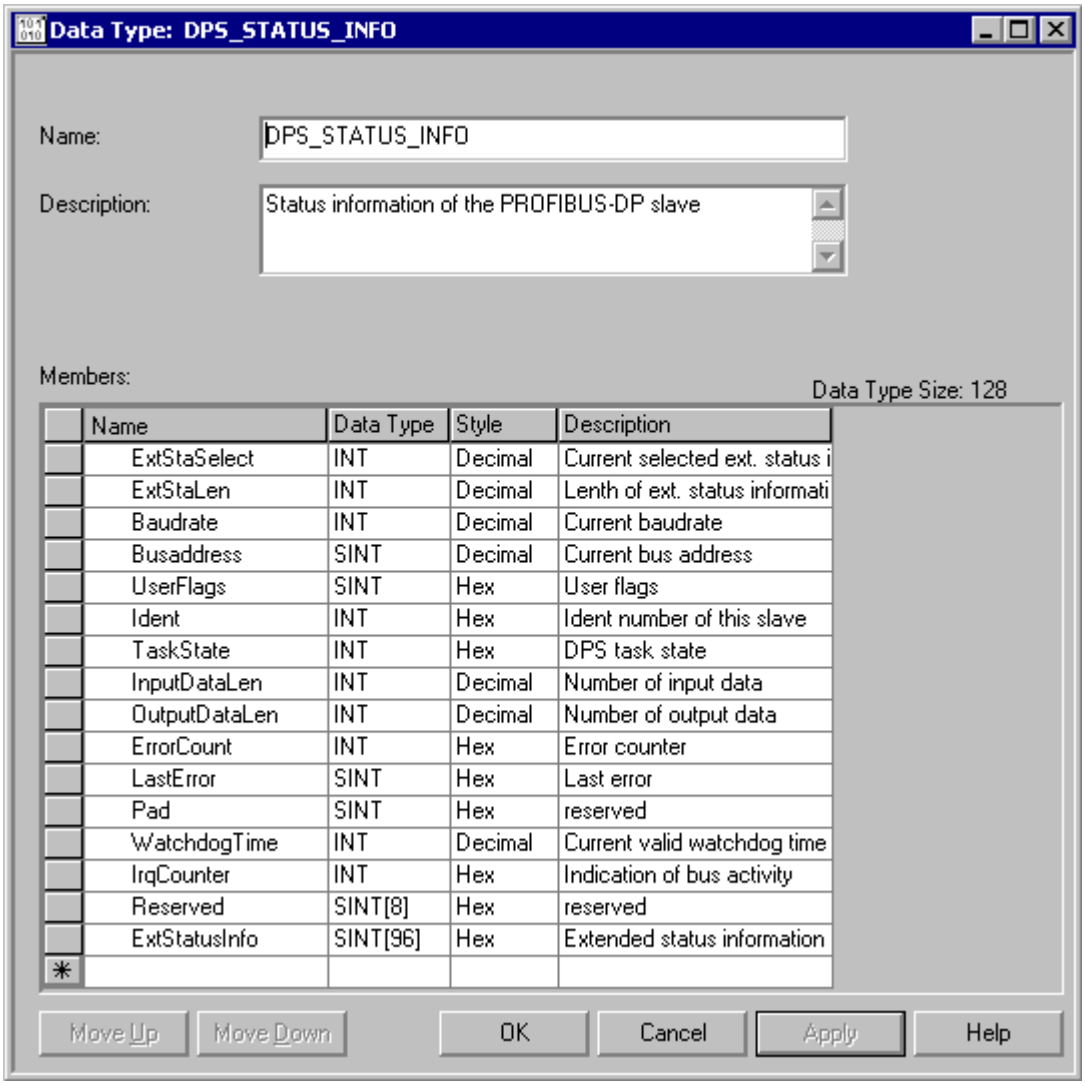

Figure 6: Status Information

Structure member	Data type	Description	Valid Values
ExtStaSelect	INT	Shows which extended status information are currently transmitted in the field "Extended Status Information"	0 = no extended status information 1 = Firmware version 2 = Slave configuration 3 = Master configuration 4 = Parameter data 5 = SPC3-DPM 6 = DPV1 C1 Diag 7 = DPV1 C2 Diag 8 = Code Diag
ExtStaLen	INT	Number of valid bytes in the region "Extended Status Information"	Depends on the selected extended status 0 = 0 Byte 1 = 16 Byte 2 = 49 Byte 3 = 49 Byte 4 = 33 Byte 5 = 40 Byte 6 = 80 Byte 7 = 72 Byte 8 = 30 Byte
Baudrate	INT	Baudrate on PROFIBUS	12000 = 12 MBaud 6000 = 6 MBaud 3000 = 3 MBaud 1500 = 1,5 MBaud 500 = 500 kBaud 187 = 187,5 kBaud 93 = 93,75 kBaud 9 = 9,6 kBaud 0 = not detected
Busaddress	SINT	Bus address of the Slave	1 … 126
UserFlags	SINT	User Fags	D0 = Parameter data changed D1 = Configuration data changed D2 ... D7 Don’t care
Ident	INT	Slave ident number	07C6h
TaskState	INT	Slave status	See following table
InputDataLen	INT	Length of input data(*)	0 ... 244
OutputDataLen	INT	Length of output data(*)	0 ... 244
ErrorCount	INT	Error counter	0 ... FFFFh
LastError	SINT	Last error	See following table
Pad	SINT	Reserved	
WatchdogTime	INT	Current watchdog time	5 ... 65535 ms
IrqCounter	INT	Indication of bus activity	0 ... 0xFFFF
Reserved	SINT[8]	Reserved	Reserved
ExtStatusInfo	SINT[96]	Extended status information	See following section

Table 11: Status Information of the Slave

(*)Note: The status informations ‘InputDataLen’ and ‘OutputDataLen’ are related to the definition of inputs and outputs from point of view of PROFIBUS. There is a clear definition of inputs and outputs by PROFIBUS. They are always defined from point of view of a PROFIBUS-Master. Do not mix it up with the input and output area of the communication module. Example: If in status ‘OutputDataLen’ is indicated a value of 4 Bytes, then it is related to the input area of the communication module, because the input area of the communication module are outputs from point of view of a PROFIBUS-Master. The same relation applies to the status ‘InputDataLen’ and the output area of the communication module.

TaskState:

Value (hex) (x =don’t care)	Meaning	Description
xxx1	Task is during initialization	If this state stays for some seconds, the configuration parameters may be invalid.
xx1x	Task running	The initialization happened without error, generally the task is able to run communication on the bus.
x1xx	Diagnostic	Slave diagnostic telegrams will be sent at the moment on the bus. Reasons could be, if the user program or the DP master orders this.
1xxx	Data exchange	The data exchange mode is active. The user-data will be transferred on the bus between the master and the slave actually.

Table 12: Values for TaskState

LastError:

Value	Meaning	Description
52	Invalid bus address	Valid addresses are between 0 and 125
54	Invalid ‚Module Type‘	The configured code of the ‚Module Type‘ parameter is invalid. If this error happens after a configuration by the controller application check the configured ‚ModuleTypes’ also the value ‚Number of valid config bytes‘.
55	Invalid ‚ModuleLength‘	The configured code for a parameter 'ModuleLength' is not defined.
61	No address-switches available on the hardware	Please contact your distributor
70	I/O-data too long	The maximum size of I/O-data has been exceeded. Please check the length of all modules.
71	SPC3/ASPC2 initialization error	The SPC3 returns an error during initialization. Please contact our hotline.

Table 13: Values for Last Error

2.3.7 Extended Status Information

Via the extended status area the Slave module is in the position to transfer 96 Byte extended status information to the controller application. Which information is transferred depends on the parameter „ExtStaInfo“ in the „Device Command Register“. This can be controlled by the application program. If the controller application selects a specific extended status, it will be acknowledged by the Slave module in the status region in „ExtStaSelect“. As long as the slave adapter has not acknowledged this selection, the extended information is invalid. How many bytes in the extended status area are valid, depends on the selected status. The number of valid bytes will be shown in the status area in „ExtStaLen“.

Ext. Status 0: (Length 0 Byte)**:**

No extended status information transferred.

Ext Status1: Firmware (Length 16 Byte)

Structure member	Data type	Description	Example
FwName	SINT[8]	Firmware Name	“DPS “
FwType	SINT[8]	Firmware Type	“RIF 1788“
FwVersion	SINT[8]	Firmware Version	“V01.000 “
FwDate	SINT[8]	Firmware Date	“15.10.03 “

Table 14: Extended Status Information of the Firmware

Ext. Status2: Slave Configuration (Length 49 Byte)

Structure member	Data type	Description
CfgLength	SINT	Number of valid configuration bytes
CfgByte1	SINT	Configuration byte 1
CfgByte2	SINT	Configuration byte 2
CfgByte3	SINT	Configuration byte 3
CfgByte4	SINT	Configuration byte 4
....	....	....
CfgByte48	SINT	Configuration byte 48

Table 15: Extended Status Information of the Slave Configuration

Ext. Status 3: Master Configuration (Length 49 Byte)

Structure member	Data type	Description
CfgLength	SINT	Number of valid configuration bytes
CfgByte1	SINT	Configuration byte 1
CfgByte2	SINT	Configuration byte 2
CfgByte3	SINT	Configuration byte 3
CfgByte4	SINT	Configuration byte 4
....	....	....
CfgByte48	SINT	Configuration byte 48

Table 16: Extended Status Information of the Master Configuration

Ext. Status 4: Parameter Data (Length 33 Byte)

Structure member	Data type	Description
PrmLength	SINT	Number of valid parameter bytes
PrmByte1	SINT	Parameter byte 1
PrmByte2	SINT	Parameter byte 2
PrmByte3	SINT	Parameter byte 3
PrmByte4	SINT	Parameter byte 4
....	....	....
PrmByte32	SINT	Parameter byte 32

Table 17: Extended Status Information of the Parameter Data

Ext. Status 5: SPC3-DPM (Length 40 Byte)

Structure member	Data type	Description
----	---	For internal use only

Table 18: Extended Status Information of the SPC3-DPM

Ext. Status 6: DPV1-C1-Diag (Length 80 Byte)

Structure member	Data type	Description
StaReqUsr	DINT	Status Request from User
StaMsgSen	DINT	Status Messages Sent
NegStaCnf	DINT	Negative Status Confirmations to User
DiagReqUsr	DINT	Diagnosis Request from User
DiagMsgSen	DINT	Diagnosis Messages Sent
NegDiagCnf	DINT	Negative Diag Confirmations to User
AlaReqUsr	DINT	Alarm Request from User
AlaMsgSen	DINT	Alarm Messages Sent
PosAlaCnf	DINT	Positive Alarm Confirmations to User
NegAlaCnf	DINT	Negative Alarm Confirmations to User
Requests	DINT	Requests
ImmNegCnf	DINT	Immediate Negative Confirmations
RW_Ind	DINT	R/W Indications to User
PosRWResp	DINT	Positive R/W Responses from User
NegRWResp	DINT	Negative R/W Responses from User
AlaAckInd	DINT	Alarm Ack Indications
AlaAckResp	DINT	Alarm Ack Responses
AlaAckErr	DINT	Alarm Ack Errors
ErrRespUsr	DINT	Erroneous Responses from User
UnxRespUsr	DINT	Unexpected Responses from User

Table 19: Extended Status Information of the DPV1-C1-Diag

Ext. Status 7: DPV1-C2-Diag (Length 72 Byte)

Structure member	Data typ	Description
PduRecv	DINT	PDUs Received
PduProv	DINT	PDUs Provided
PduFetched	DINT	PDUs Fetched by Master
InitIndUsr	DINT	Initiate Indications to User
PosInitRespUsr	DINT	Positive Initiate Responses from User
PosInitRespUsr	DINT	Negative Initiate Responses from User
RWTIndUsr	DINT	R/W/T Indications to User
PosRWTIndUsr	DINT	Positive R/W/T Responses from User
NegRWTIndUsr	DINT	Negative R/W/T Responses from User
IdleReqRecv	DINT	Idle Requests Received
IdleReqSend	DINT	Idle Requests Sent
AbtReqRecv	DINT	Abort Requests Received
AbtReqSend	DINT	Abort Requests Sent
AbtReqUsr	DINT	Abort Requests from User
AbtIndUsr	DINT	Abort Indications to User
ErrRespUsr	DINT	Erroneous Responses from User
UnxRespUsr	DINT	Unexpected Responses from User
SrcStopp	DINT	Services Stopped

Table 20: Extended Status Information of the DPV1-C2-Diag

Ext. Status 8: Code-Diag (Length 30 Byte)

Structure member	Data type	Description
---	----	For internal use only

Table 21: Extended Status Information of the Code Diag

2.4 Configuration of the Slave

2.4.1 General

For the slave itself no configuration tool is necessary. The GSD-File is named „HIL_07C6.GSD“ and it is located on the CD supplied with the module. There are two ways to configure the module, which are described in the following.

2.4.2 Configuration by the Master

The "Configuration by Master“ is the easiest way to configure the Slave. The project „RIF_1788_DPS.ACD“ is an example for this method. Before a PROFIBUS Master starts its communication, it carried out a comparison between its own configuration and the current configuration of the Slaves connected to the bus in fact. The slave RIF 1788-DPS automatically takes over the configuration which is send by the master during its comparison of the configuration. This method is activated by default, since the parameter „Configuration Size“ is initialized with 0 (Byte) in the setup dialog module.

Note: This is the easiest way to configure the Slave. But be aware that the master can send a new configuration to the slave at any time. This can cause inconsitency, if the new configuration does not match to the controller application. For more safty use the method “Configuration by Controller Application”. With this method the slave module does not start any communication as long as the slave and master configuration do not match to each other.

2.4.3 Configuration by Controller Application

The second option to configure the Slave module is to configure it by the controller application. For this the parameter „Configuration Size“ has to be set to 64 (Byte) in the setup dialog of the Slave module. Because of this parameter the controller transfers the first 64 bytes of the configuration area during its connection establishing phase to the Slave module. By initialization of these values the controller program can configure the Slave.

Configuration byte	Data type	Description	Valid values
DPS_1:C.Data[0]	SINT	Busaddress	0 ... 125
DPS_1:C.Data[1]	SINT	Force Master Config	0 ; 1
DPS_1:C.Data[2]	SINT	Length DPV1 C1 Buffer	0 ... 244
DPS_1:C.Data[3]	SINT	Length DPV1 C2 Buffer	0 ... 244
DPS_1:C.Data[4]	SINT	Watchdog Timeout Low Byte	Watchdog
DPS_1:C.Data[5]	SINT	Watchdog Timeout High Byte	Watchdog
DPS_1:C.Data[6]	SINT	Number of valid config bytes	0 ... 48
DPS_1:C.Data[7]	SINT	Reserved	
...	SINT	Reserved	
DPS_1:C.Data[15]	SINT	Reserved	
DPS_1:C.Data[16]	SINT	Module 1 Type	see table 'Module Types'
DPS_1:C.Data[17]	SINT	Module 1 Length	see table 'Module Types'
...	SINT	...	
...	SINT	...	
DPS_1:C.Data[62]	SINT	Module24 Type	see table 'Module Types'
DPS_1:C.Data[63]	SINT	Module24 Length	see table 'Module Types'

Table 22: DP Slave Configuration data

Explanation of settable configuration values:

Busaddress

Slave addresses are valid from 0 to 125. The module has two rotating address switches from 0 to 99. If you choose 0 as software parameter (what is recommended), then the module will read the selected address from the rotating switches. Otherwise the software is used. With this you are able to setup bus addresses above 99.

Address switches	Software address parameter	Active bus address	Description
0 .. 99	0	0 .. 99	Address switches are valid
XX	1 ... 125	1 .. 125	Software parameter is valid (xx don't care)
XX	> 125	XX	Invalid (will cause an initialization error)

Table 23: HW and SW address combinations

Force Master Config

If this value is set to 0, the slave will not start it's communication while master and slave configuration do not match with each other. If this value is set to 1 the slave takes over the configuration from the master.

Length DPV1 C1 Buffer

Buffer reserved for DPV1 C1 services.

Length DPV1 C2 Buffer

Buffer reserved for DPV1 C2 services.

Watchdog timeout

The Slave module supervises its I/O exchange with the controller with a timeout. If the controller does not update the output data within this time, the Slave stops the cyclic data exchange to the master and goes into a safe state.

If this parameter is initialized with **0**, the module calculates automatically a timeout value by the connection parameter between controller modules.

WATCHDOG_TIME = RPI * CNX_TIO_MULTIPLYER

The RPI (Requested Packet Interval) is adjusted in the setup dialog of the communication module. The value CNX_TIO_MULTIPLYER (Connection Timeout Multiplyer) is laid down by the controller. The current value for the watchdog timeout can be checked in the status area in "WatchdogTime".

If you initialize the parameter „Watchdog timeout" **unequal to 0**, then this value is used as timeout parameter.

Number of valid config byte

Number of valid configuration bytes of the defined modules.

Module n Type / **Module n Length**

The PROFIBUS-DP Slave offers a flexible, modular composition. That means parts of the input and output areas can be seen as single modules. The master can put the different modules from the PROFIBUS-DP Slave to different locations in its I/O area.

The individual configured modules are laid down linearly in the I/O area of the Slave module. It is possible to configure up to 24 I/O modules.

A module is defined by a module type and its module length:

Parameter	Data type	Valid values	Description
Module Type	SINT	0 = IN Byte 1 = IN Word 2 = OUT Byte 3 = OUT Word 4 = IN Byte con 5 = IN Word con 6 = OUT Byte con 7 = OUT Word con 8 = Blank space	Input Byte without consistence Input Word without consistence Output Byte without consistence Output Word without consistence Input Byte with consistence Input Word with consistence Output Byte with consistence Output Word with consistence Blank space
ModuleLength	SINT	0 1 2 3 4 5 6 7 8 9	1 Byte/Word 2 Byte/Word 3 Byte/Word 4 Byte/Word 8 Byte/Word 12 Byte/Word 16 Byte/Word 20 Byte/Word 32 Byte/Word 64 Byte/Word

Table 24: Coding of the Module Types

Note: Please notice the definition of Input/Output modules and do not mix it up with the input and output area of the communication module. PROFIBUS gives a clear definition of Inputs/Outputs. Inputs/Outputs modules are always defined from point of view of the PROFIBUS master. If you configure a Output module you will see this in the input area of the communcation module, because the input area of the communication module are output datas from point of view of a PROFIBUS master. The same applays to a Input module. If you define a Input module it is layd down in the output area of the communiction module, because the output area are input datas from point of view of a PROFIBUS master.

The project „RIF_1788_DPS_config.ACD“ is an example for a module configuration by the controller application. The following figure shows the configured parameter in this sample project.

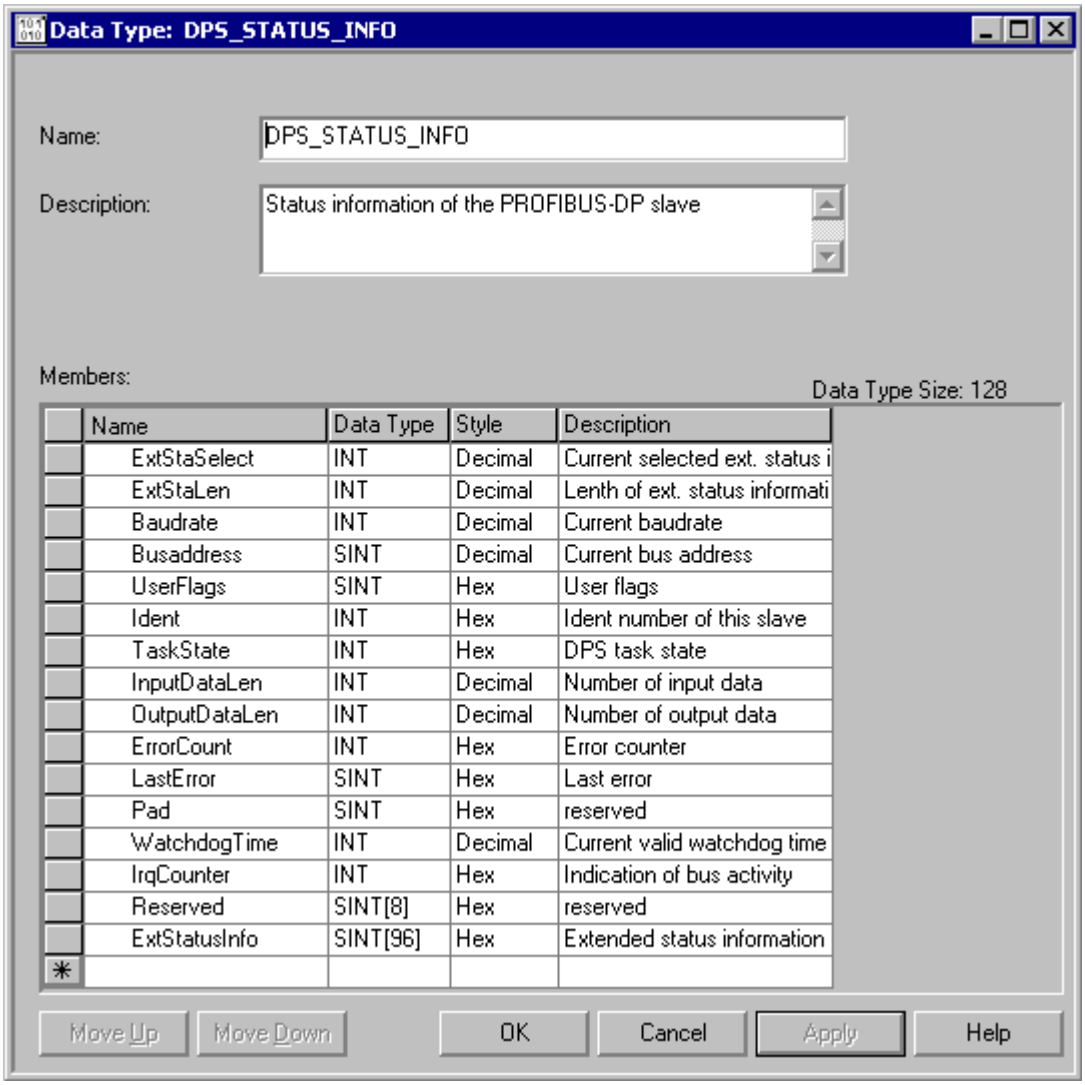

Figure 7: Configured Values of the Sample Project

Configuration Byte	Value	Description
Busaddress	0	Address switches are used
Force Master Config	0	Master and slave configuration have to match with each other
Number of valid config bytes	4	4 Bytes of module configuration are valid
Watchdog timeout Low Byte	100 (64h)	User watchdog of 100 ms
Watchdog timeout Low Byte	0	
Module 1 Type	4	4 Byte Input with consistence
Module 1 Length	3	
Module 2 Type	6	4 Byte Output with consistence
Module 2 Length	3	

Table 25: Configured Values ot the Sample Project

2.5 Ladder Logic

Enter each one example of the DEV_STATE_REGISTER, DEV_COMMAND_REGISTER and DPS_STATUS_INFO in the **TagEditor** of the data modules in order to obtain structured access to the data module from a RSLogix5000 project.

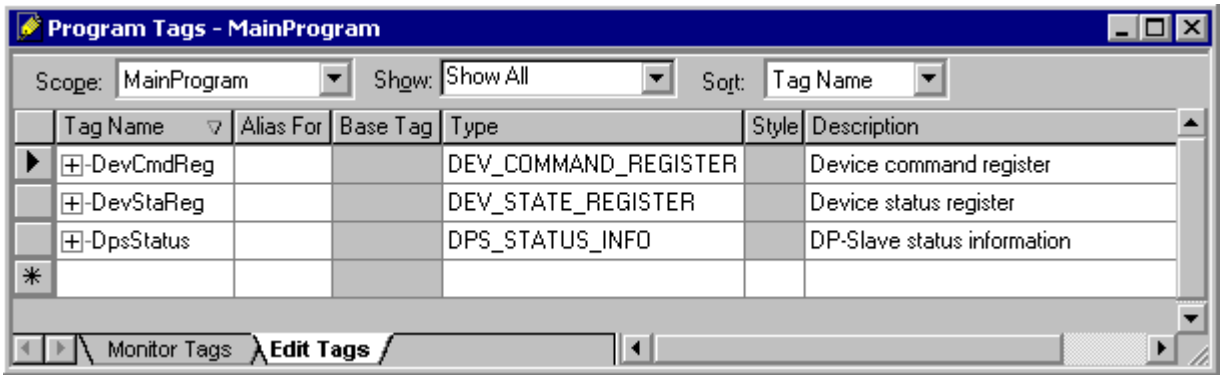

Figure 8: Tag Edit

Enter a copy module which copies the data between the input and output and state region of the module and the laid down data modules into the **MainRoutine** of the project:

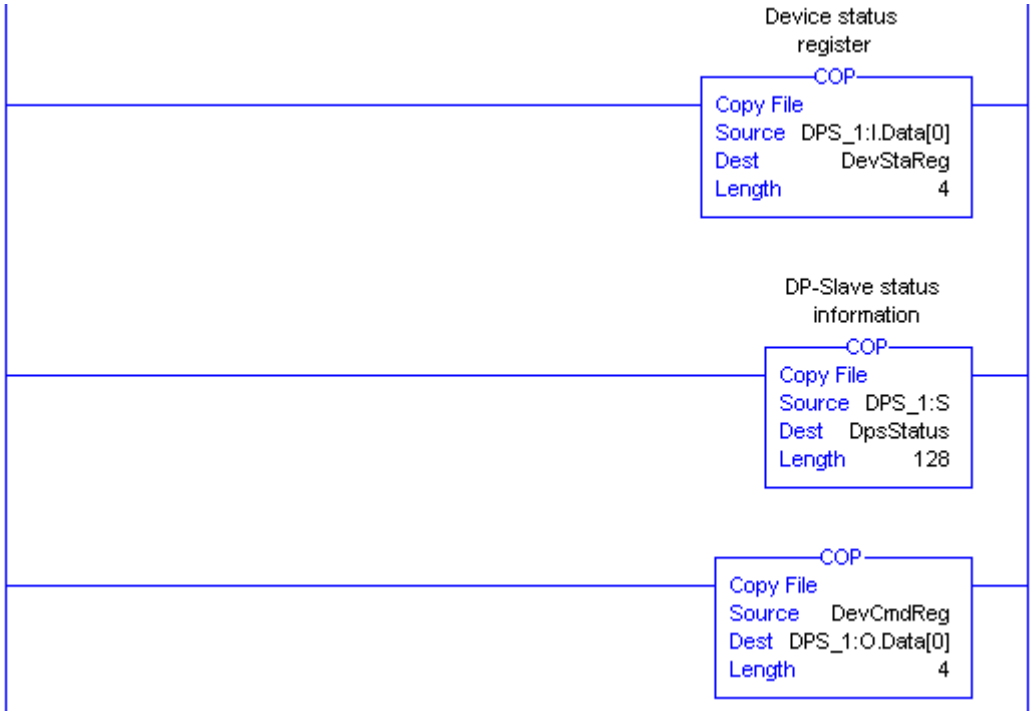

Figure 9: Module Copy

Rung No.	Data Module	from	to	Length
1	Device Status Register	DPS_I.[0]	**Tag** DevStaReg	4 Byte
2	PROFIBUS Status	DPS_S.[0]	**Tag** DpsStatus	128 Byte
3	Device Command Register	**Tag** DevCmdReg	DPS_O.[0]	4 Byte

Table 26: Module Copy

Note: The copy function can also be carried out from the example project "RIF_1788_DPS.ACD". In the example project the copy functions are collated in the **SR_Copy** subroutine. This is called up cyclically from the MainRoutine.

This completes all the required steps for the configuration of the module in RSLogix 5000. Save the project and load it on to the controller.

3 Diagnostic

3.1 I/O LED of the Controller

The general communication between card and controller is displayed via the I/O LED of the Controller. The faultless communication state is reached, when the I/O LED of the FlexLogix Controller is flashing green durable. As long this LED is flashing or off, no communication between controller and card takes place.

3.2 Status LEDs

This card has two bicolor status LEDs. They inform about the communication state of the card. The **SYS**-LED shows the common system status of the card. It can flash yellow or green. The **COM**-LED displays the status of the PROFIBUS communication. It can flash yellow. The meaning of the LEDs is described in the booklet of the Partner Products System Software CD.

If the SYS-LED is green durable and the COM-LED is yellow durable, the card is in cyclic data exchange with the PROFIBUS Master and the communication is running faultless.

3.3 Trouble Shooting

Some error sources and their reasons are listed in the following table:

Error	Reason	Remedy
I/O LED of the FlexLogix Controller is off	No PROJECT with a configured RIF1788-DPS is loaded on the controller	Load controller project with RIF1788-DPS card
I/O LED of the FlexLogix Controller is flashing	Wrong communication parameter set for the card during setup in RSLogix	Check communication parameter of the card in the controller project
SYS LED is flashing green irregularly (COM LED is off)	Various reasons	Startup: initialization error, Check error which is displayed in the status area of the module in „LastError“ Run time: Watchdog expired, card has to be reset
SYS LED is flashing cyclic green (COM LED is off)	Various reasons	Check that the PROFIBUS cable is connected correctly Check for bus activity in the status area of the adapter in „IrqCounter“ Check that the PROFIBUS Master works correct Use PROFIBUS Master diagnostic capabilities Check that no error number is displayed in the status area in „LastError“ Check that the controller is in RUN mode and that application program has cleared the NotReady Bit

Table 27: Trouble Shooting

4 Lists

4.1 List of Figures

4.2 List of Tables