Operating Instruction Manual

RIF 1769-DPM

PROFIBUS-DP Master for CompactLogix 1769 Series

Edition: 1
Language: English (EN)

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 du 35ième Régiment d'Aviation Miniparc du Chêne FR-69500 Bron
Phone	+49 (0) 6190 9907-0	+33 (0) 472379840
Phone Sales	+49 (0) 6190 9907-90	+33 (0) 472379840
Phone Support	+49 (0) 6190 9907-99	+33 (0) 472379840
Fax	+49 (0) 6190 9907-50	+33 (0) 478268327
E-Mail	info@hilscher.com	info@hilscher.fr
E-Mail Sales	sales@hilscher.com	
E-Mail Support	hotline@hilscher.com	
Web	www.hilscher.com	www.hilscher.com

Contact	Italy	Switzerland
Address	Hilscher Italia s.r.l. Via Grandi, 25 IT-20090 Vimodrone (MI)	Hilscher Swiss GmbH Hubelmattstraße 29 CH-4500 Solothurn
Phone	+39 / 0225007068	+41 (0) 32 6236 633
Phone Sales	+39 / 0225007068	+41 (0) 32 6236 633
Phone Support	+39 / 0225007068	+49 (0) 6190 9907-99
Fax	+39 / 0225029973	+41 (0) 32 6236 632
E-Mail	info@hilscher.it	info@hilscher.ch
E-Mail Sales	sales@hilscher.it	sales@hilscher.com
E-Mail Support	supporto@hilscher.it	hotline@hilscher.com
Web	www.hilscher.com	www.hilscher.com

Hilscher North America

Contact	North America
Address	Hilscher North America, Inc. 2443 Warrenville Road, Suite 100 Lisle, Illinois 60532, USA
Phone	(+1) 630 505 5301
Phone Sales	(+1) 630 505 5301
Phone Support	(+1) 630 505 5301
Fax	(+1) 630 505 7532
E-Mail	info@hilscher.us
E-Mail Sales	info@hilscher.us
E-Mail Support	info@hilscher.us
Web	www.hilscher.com

Worldwide: Distributors

Please visit our Homepage at
www.hilscher.com

List of Revisions

Index	Date	Version	Chapter	Revisions
1	17.05.2005		all	created

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.

Microsoft and Windows are registered trademarks of Microsoft Corporation. Pentium is a registered trademark of Intel Corporation. Adobe and Acrobat are trademarks of Adobe Systems Incorporated. RSLogix and CompactLogix are the trademarks of Rockwell Automation. CIF and SYCON.net are trademarks of Hilscher Gesellschaft für Systemautomation mbH.

Values with a following 'hex' are in hexadecimal notation such as 1E hex = 30. Values without any following letter are in decimal notation.

Table of Contents

1 Introduction

1.1 Intended Audience

The intended audiences for this manual are the individuals responsible for designing, installing, programming, or troubleshooting control systems that use Rockwell CompactLogix programmable controllers and the Hilscher RIF 1769-DPM PROFIBUS-DP Master module. You should have a basic understanding of electrical circuitry and familiarity with relay logic. If you do not, obtain the proper training before using this product.

1.2 General Information RIF 1769-DPM

The communication module RIF 1769-DPM is a slot extension module for a CompactLogix Controller which enables controllers to communicate with a PROFIBUS network. The RIF 1769-DPM is a PROFIBUS-DP Master. The configuration and diagnostic of the PROFIBUS system is done via the serial diagnostic interface of the module using the Hilscher System Configuration tool SYCON.net. The data exchange between controller and module is done via the I/O process data image using CompactLogix back plane technology.

1.3 Software Requirements

Follows are the software requirements for using the RIF 1769-DPM module within a CompactLogix system. You must have the following software installed on your computer unless otherwise noted:

Rockwell Software

- RSLogix 5000, V13.00 or higher

Hilscher Software (Configuration Tool)

- SYCON.net for Rockwell Interfaces V1.023 or higher

Requirements for SYCON.net

- PC with 1 GHz processor or higher
- Windows 2000 (Service Pack 2 or higher) or
- Windows XP (either Home or Professional Edition)
- Internet Explorer 5.5 or higher
- Adobe Acrobat Reader 4.0 or higher
- Free disk space: 10–15 MByte
- CD ROM drive
- RAM: min. 256 MByte
- Graphic resolution: min. 1024 x 768 pixel
- Keyboard and Mouse

1.4 Hardware Requirements

The following minimum hardware is required to use the 1769 PROFIBUS module.

- Personal Computer
- 1769 – Programmable Controller
- 1769 – Power Supply
- 1769 – Right or Left handed Termination End Cap
- Serial Cable for interface to the 1769-Programmable Controller.
- Serial Cable for SYCON.net to Diag port connection Hilscher part number CAB-SRV-MD8

1.5 Reference Manuals

Manual	Description	Note
1769-IN047C-EN-P	CompactLogix Controller Installation Instructions	Rockwell Automation
1769-UM007D-EN-P	CompactLogix System User Manual	Rockwell Automation
RIF1769 Booklet.pdf	Booklet (Hardware installation RIF 1769, Wiring, LED displays, and technical data)	Hilscher GmbH
pbm__00e.pdf	Operating Instruction Manual for Hilscher DTM for PROFIBUS-DP Master Devices (Configuration of Hilscher Master Devices)	Hilscher GmbH
pbg__00e.pdf	Operating Instruction Manual for Generic DTM for PROFIBUS-DP Slave Devices (Configuration of PROFIBUS-DP Slave Devices)	Hilscher GmbH
DPM_PRE.pdf	Protocol Manual PROFIBUS-DP Master	Hilscher GmbH

Table 1 : Reference Manuals

1.6 Reference Systems

The firmware of the communication module RIF 1769-DPM was developed and tested with following CompactLogix Controller types and firmware revisions.

RIF 1769-DPM	CompactLogix 1769-L20	CompactLogix 1769-L32E
Firmware V1.000	Firmware V13.18	Firmware V13.28

Table 2 : Reference System

1.7 1769-Programmable Controller Functionality

PROFIBUS-DP supports acyclic services through messages. These PROFIBUS-DP services are supported by the RSLogix5000 programming tool using CIP messages. Not all of the 1769 Programmable Controllers support CIP messaging. If the Controller does not support messaging, the named acyclic PROFIBUS-DP services are not available.

The basic PROFIBUS-DP acyclic services Global Control or Slave Diag request are also executable in addition to the CIP method by using the I/O area. Follows is a matrix of 1769 Programmable Controllers and the functionality that they support.

Processor/ Featuers	**1769-L20**	**1769 -L30**	**1769 -L31**	**1769 -L32E**	**1769- L35E**
I/O	yes	yes	yes	yes	yes
CIP Messaging	no	no	yes	yes	Yes

Table 3 : 1769-Programmable Controller Functionality

yes = functionality supported

no = functionality not supported

2 Installation, Wiring and System Planning

This section describes how to install and wire the RIF 1769-DPM Master module. The following table describes what this chapter contains and where to find specific information. When planning, installing and wiring your system please refer to following manuals:

RIF1769 Booklet.pdf

This manual can be found on the CD delivered with the RIF 1769-DPM module and contains detailed information about:

- How to assemble the RIF 1769-DPM module into a CompactLogix system.
- PROFIBUS wiring
- Modules LED displays
- Modules technical data and specifications
- ...

1769-IN047C-EN-P.PDF / 1769-UM007D-EN-P.PDF

These manuals are available from Rockwell Automation and can be found on every RSLogix CD or on the Homepage of Rockwell Automation.

These manuals containe **important** information about:

- CompactLogix System planning
- CompactLogix Controller Installation Instructions
- CompactLogix System specifications
- ...

Consider the following when planning your system:

- The PROFIBUS Master Module can communicate with up to 125 PROFIBUS Slave devices.
- A 1769-ECR (right end cap) or 1769-ECL (left end cap) is required to terminate the end of the Compact I/O bus.
- Each bank of Compact I/O must have its own power supply.
- A Compact I/O power supply has limits on the amount of +5V dc and +24V dc current it can supply to modules in its I/O bank. These limits depend on the catalog number (e.g. 1769-PA2) of the supply. A bank of modules must not exceed the current limits of the I/O bank power supply. Refer to the Compact 1769 Expansion I/O Power Supplies Installation Instructions.
- The PROFIBUS scanner has a distance rating of 6; therefore, the scanner must be within 6 modules of the I/O bank's power supply.
- Determine the PROFIBUS baud rate based on standard PROFIBUS System considerations.
- Consider the number of words of I/O data the host controller supports.

3 PROFIBUS Functionality

3.1 DPV0 Services

DPV0 services in PROFIBUS refer to the cyclic data exchange mechanism between a class 1 master and a network slave. PROFIBUS-DP defines two types of masters. The class 1 master handles data communication with slaves assigned to it. A class 2 master should only be used for commissioning purposes. In a PROFIBUS telegram, class 1 masters and slaves transmit up to 244 bytes per telegram. Valid station addresses on PROFIBUS range from 0 to 126.

3.1.1 Fail Safe Mode

For safety reasons, the PROFIBUS master informs connected slaves of its current control status at certain intervals using a "Global Control" telegram. If the master goes to Clear Mode, the Fail Safe enabled slaves will switch to a Fail Safe state. Slaves capable of the Fail Safe state can be configured to either to hold the last state of the outputs or set its outputs to "0". Slaves that do not support the Fail Safe state set their outputs to "0".

3.1.2 Global Control

With the Global Control telegram, the master can send unsolicited commands like Sync/Unsync, Freeze/Unfreeze and Clear Data to a slave or a group of slaves for synchronization purposes. Group membership is defined during network start-up and can be set in SYCON.net.

3.1.3 Sync and Freeze

Sync and Freeze are optional commands and slaves do not need to support them. However, they must be able to process the Global Control telegram. With a Freeze command, the master prompts a slave or a group of slaves to "freeze" their inputs to the current state. A Sync telegram causes the current output data to latch at their current state until the next Sync telegram arrives. Unfreeze and Unsync cancel each corresponding state.

3.1.4 Extended Device Diagnostics

Using diagnostic telegrams, the slave informs the network master of its current state in a high-priority telegram. The first 6 bytes of the diagnostic telegram are comprised of information such as its identity code ("Ident Code") or correct/incorrect configuration. The remaining bytes of this telegram are referred to as Extended Device Diagnostics and they contain information that is specific to the particular slave.

3.1.5 Watchdog

Using the Watchdog functionality a network slave is able to monitor bus traffic in order to ensure that the network master is still active and process data sent and received are still being updated. The Watchdog time is configured in SYCON.net and is transmitted to the slave during the network start-up phase. If the Watchdog time out has been reached the slaves go to their Fail Safe state (if supported) or set their outputs to "0".

3.2 DPV1 Services

As an addition to cyclic DPV0 services, non-cyclic services called Read, Write and Alarm were added to PROFIBUS. These services are referred to as DPV1. With DPV1, it is possible to address individual modules within the slave. In addition, DPV1 services allow transferring non-time critical data to slaves who require a large amount of configuration data or slaves that have to change measurement ranges during runtime. DPV1 data exchange takes place after cyclic data exchange in a PROFIBUS network cycle.

3.2.1 Read Request

With a Read Request telegram, the class 1 master can read data addressed by slot and index within the data range of a slave device. This may take several DPV0 cycles. If the master discovers a timeout, it aborts both DPV1 and DPV0 communication with the slave. Then the communication to the slave has to be re-established. The master initiates the Read Request service.

3.2.2 Write Request

With a Write Request telegram, the class 1 master can write data addressed by slot and index into the data range of a slave device. The timeout handling is identical to the Read Request. The master initiates the Write Request service.

3.2.3 Alarm Indication

DPV1 Alarm handling is an addition to the Device Diagnostic function in PROFIBUS. Alarms are reported to the master as device specific diagnostic information. Therefore, the slave initiates an Alarm Indication. Other than Device Diagnostic messages, Alarms have to be acknowledged by the Master.

3.3 Start/Stop Communication

Start/Stop communication with one bit: With the “NRDY” (NotReady) Bit the user program can start or stop communication with the PROFIBUS-DP system. When this Bit is set from the user program, the communication between the module and all Slave devices connected, is stopped. All slaves will clear their outputs and the Master will be in Stop mode. This control bit allows the user program to make a controlled start of the communication with the PROFIBUS system.

4 Configuration and Start-Up

The following sections will describe the individual steps for configuration and start-up of the RIF 1769-DPM module. Install the PROFIBUS Master module into a free slot in the CompactLogix controller. The information for installation of communication modules in CompactLogix systems can be found in the section Installation and Wiring or in the Rockwell installation manual for the 1769 system.

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

- Configuration of the module in a CompactLogix project of the RSLogix5000 programming tool.
- Parameterization and configuration of the PROFIBUS Master with the SYCON.net configuration tool.
- Creating the data objects and the ladder diagram in RSLogix5000.

Note: The simplest way to startup the module in a RSLogix5000 project is to use the “RIF_1769_DPM.ACD” example project. In this example project, the slot number in the configuration dialog of the module may have to be changed to match the users system.

4.1 RSLogix

Contained in the sections below are the instruction for configuring the RIF 1769-DPM module in a CompactLogix system using RSLogix5000.

4.1.1 Module Selection

Create a new project in RSLogix5000 using a CompactLogix controller. Then the first step is to select the module and add it to your project. Right-click the mouse on the I/O configuration CompactBus Local of the controller project. Select **New Module** as shown below.

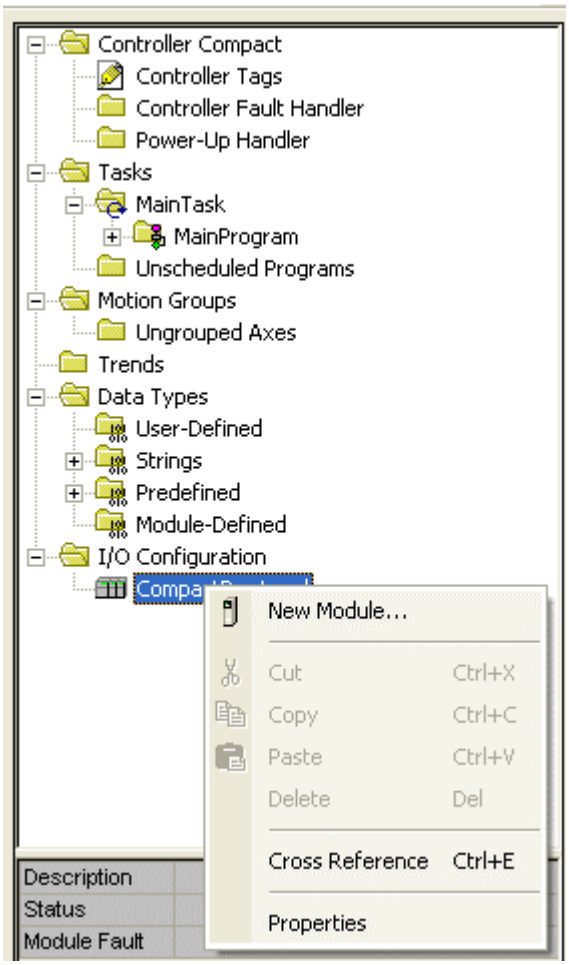

Figure 1 : Insert New Module

The following dialog box appears for the selection of the new Module.

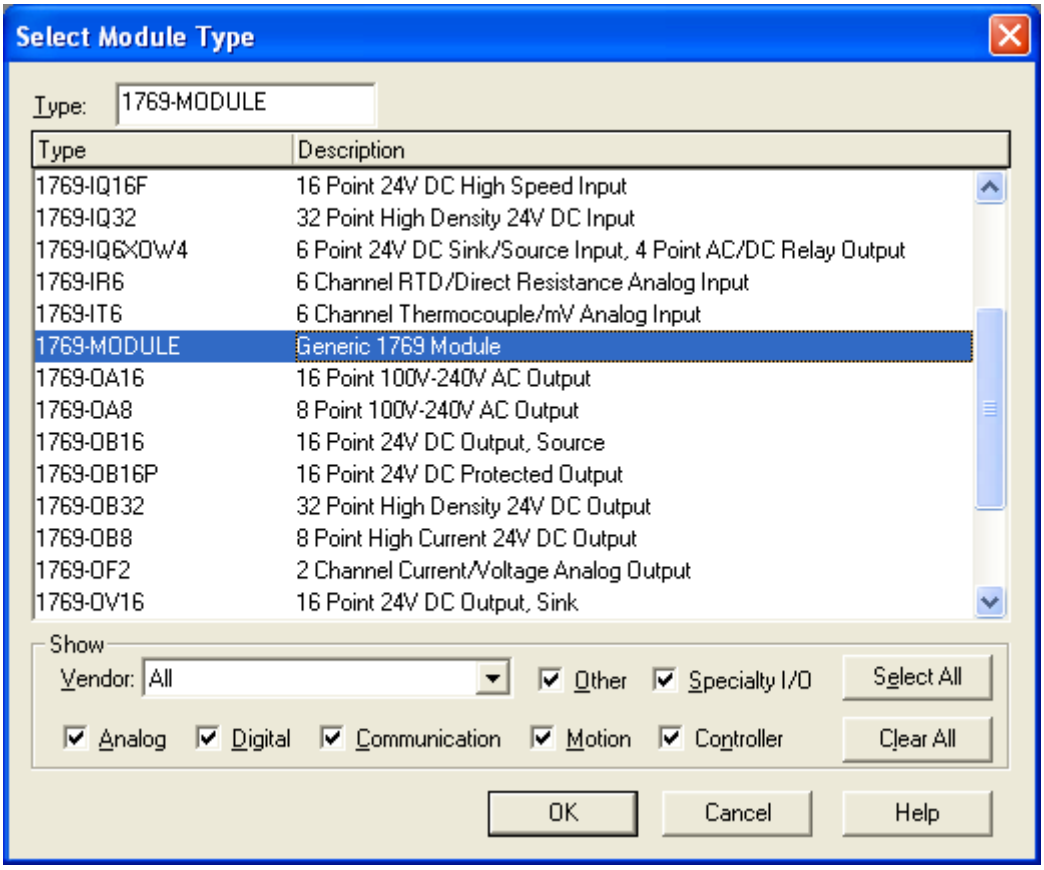

Figure 2 : Select Module Type

Select **"1769-MODULE Generic 1769 module"** from the select module type list and then OK.

4.1.2 Module Properties 1

The communications parameters for the module should be set as shown in the dialog below.

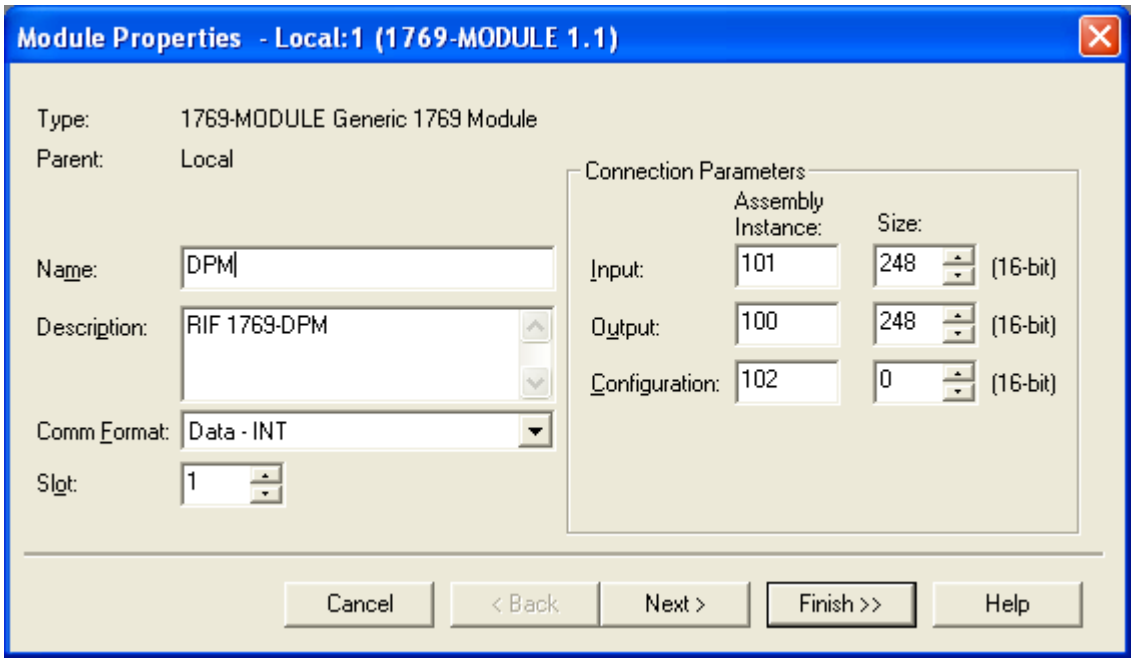

Figure 3 : Module Properties 1

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

Connection Parameter	Assembly Instance	Size (in Words)
Input	101	44 + X ... 248
Output	100	4 + Y... 248
Configuration	102	0

Table 4 : Connection Parameters

X = Number of PROFIBUS input data configured in SYCON.net

Y = Number of PROFIBUS output data configured in SYCON.net

- Input Size – The input size must be at least 88 Bytes (44 Words). It must be large enough to contain the status information required by the module, which is 88 Bytes (44 Words) and the number of PROFIBUS input data. The user can increase the size of this area using the size of each Input module connected. The Input area starts with byte 88.
- Output Size – The output size must be at least 16 Bytes (8 Words). It must be large enough to contain the command information required by the module, which is 16 bytes (8 Words), and the number of PROFIBUS output data. The user can increase the size of this area using the size of each Output module connected. The Output area starts with byte 16.

Note: If the parameters do not correspond to the template values, then the controller cannot build up communication with the module.

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

4.1.3 Module Properties 2

The Requested Packet Interval RPI is shown in the following dialog box. Within this time interval, the I/O data between module and controller are exchanged.

It is not possible to change the RPI in this dialog separately for each module. The RPI can be changed in the properties dialog of the “CompactBus Local” for all I/O modules. Values in 1.0 ms steps are possible. The PROFIBUS module supports all possible RPI values.

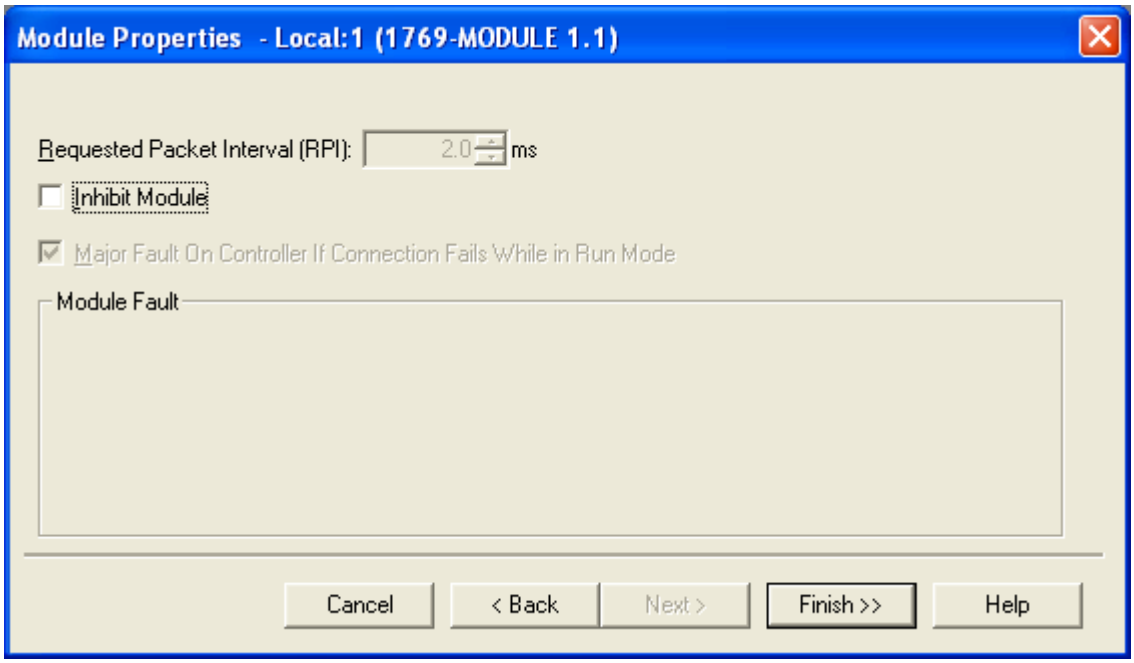

Figure 4 : Module Properties 2

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

4.2 SYCON.net

The following section will detail the basics of using the configuration and diagnostic software SYCON.net to configure the PROFIBUS-DP Master module and Slave I/O system.

4.2.1 General

The PROFIBUS-DP system is configured by using the configuration and diagnostic tool SYCON.net. The configuration can be downloaded to the module and stored into flash memory by using the download function of SYCON.net. Download of the configuration is done via the diagnostic interface. Connect the diagnostic interface to a serial interface of the PC. Start the SYCON.net from the installation folder. Follows are the basic steps for the creation of a PROFIBUS configuration in SYCON.net. A comprehensive explanation for all configuration steps can be found in the Online help in the **Help>Topics...** menu and the reference manuals Operating Instruction Manual for Hilscher DTM for PROFIBUS Master Devices and Generic DTM for PROFIBUS-DP Slave Devices.

4.2.2 Create a New Project

Create a new PROFIBUS project. Select the menu **File >New...** in the SYCON.net. The following appears:

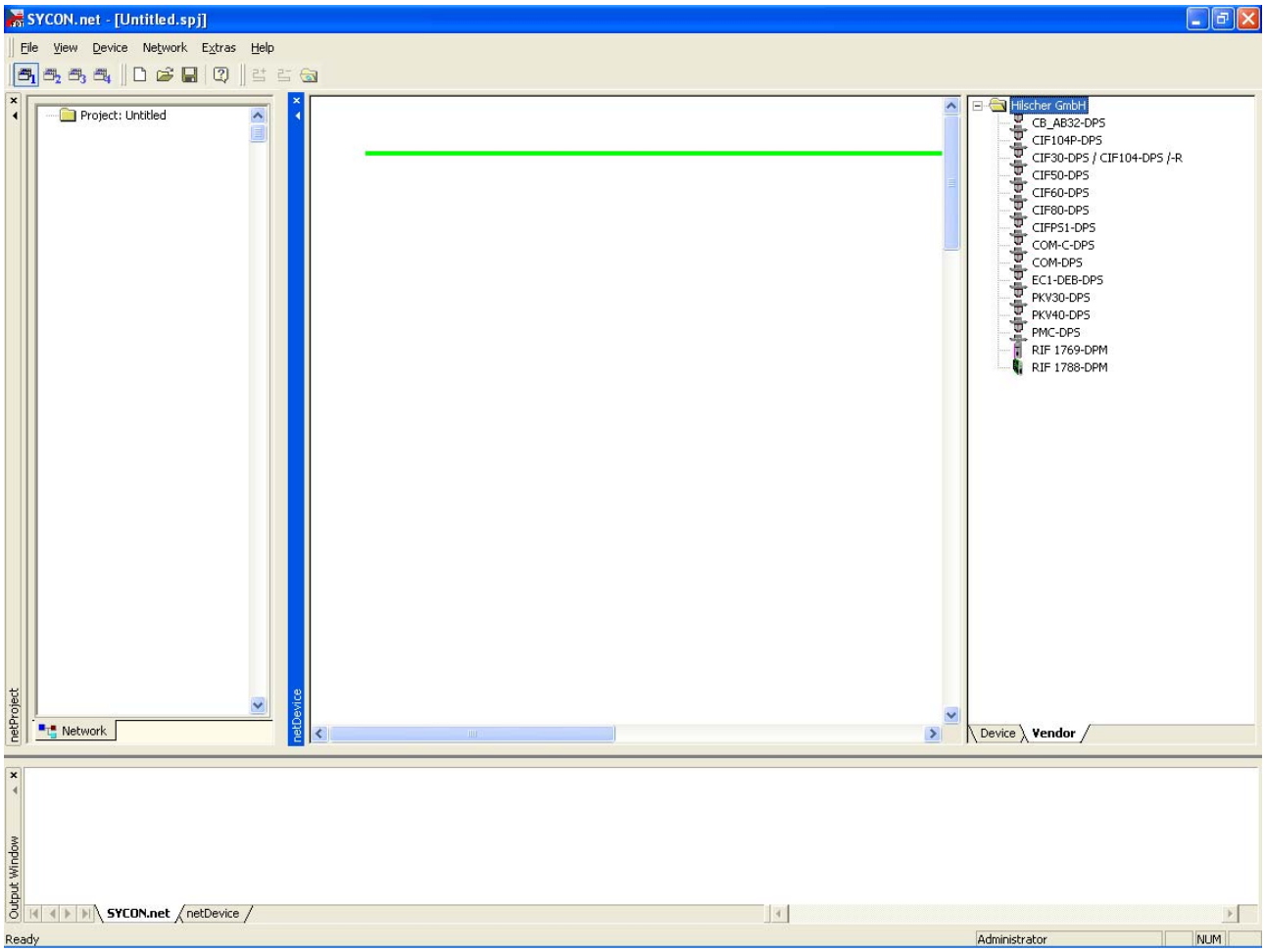

Figure 5 : File New SYCON.net

Then, in SYCON.net, click and hold the left mouse button and drag the RIF 1769-DPM device from the device catalog area to either the netDevice bus or the netProject screen, add the device to the project by releasing the left button when the + sign appears. Your screen should show the following:

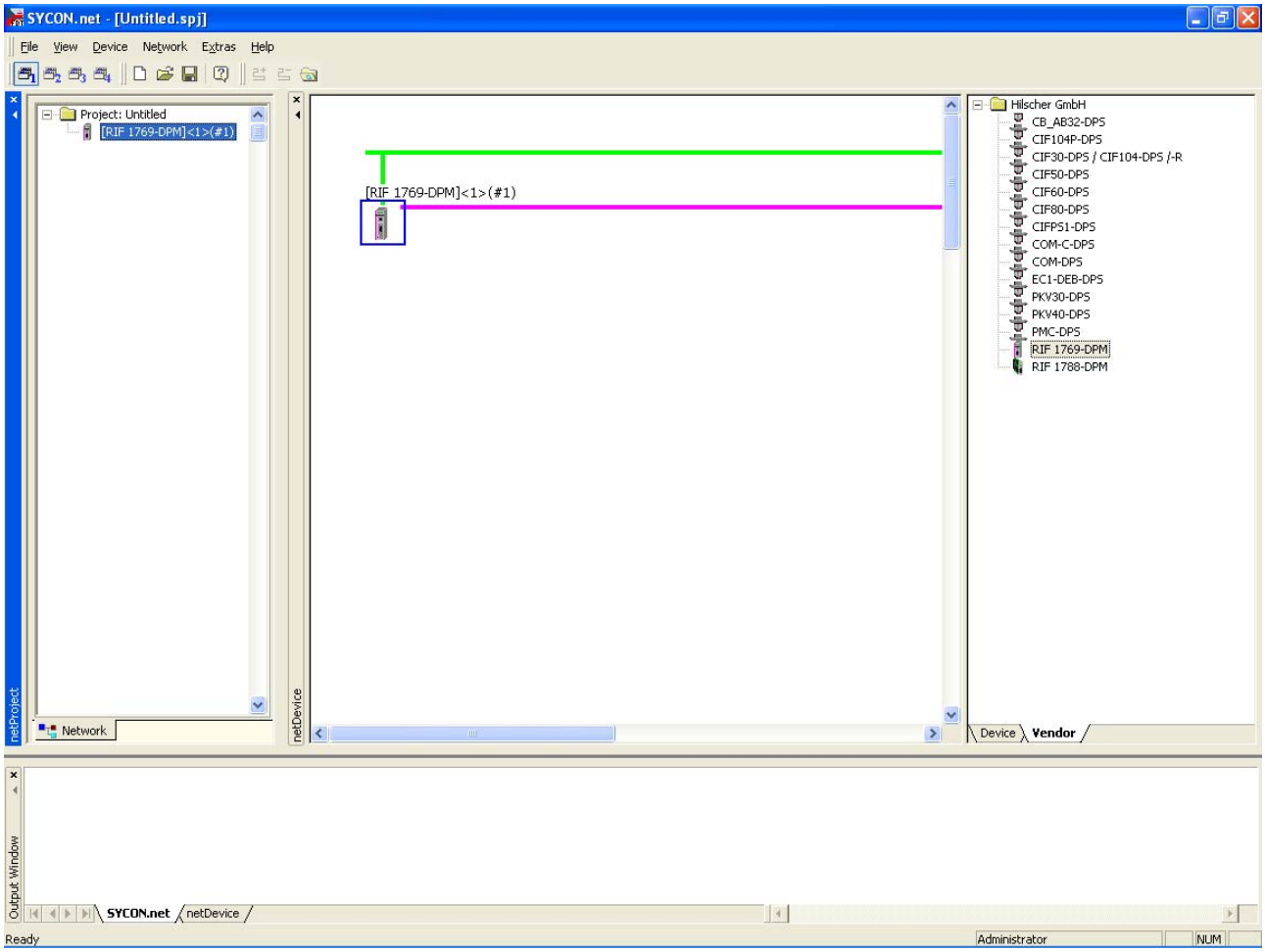

Figure 6 : Add Master SYCON.net

The Master is now ready for configuration.

4.2.3 Configuration of the RIF 1769-DPM Master

Double click on the Master that appears in the netDevice or the netProject window. The following dialog appears.

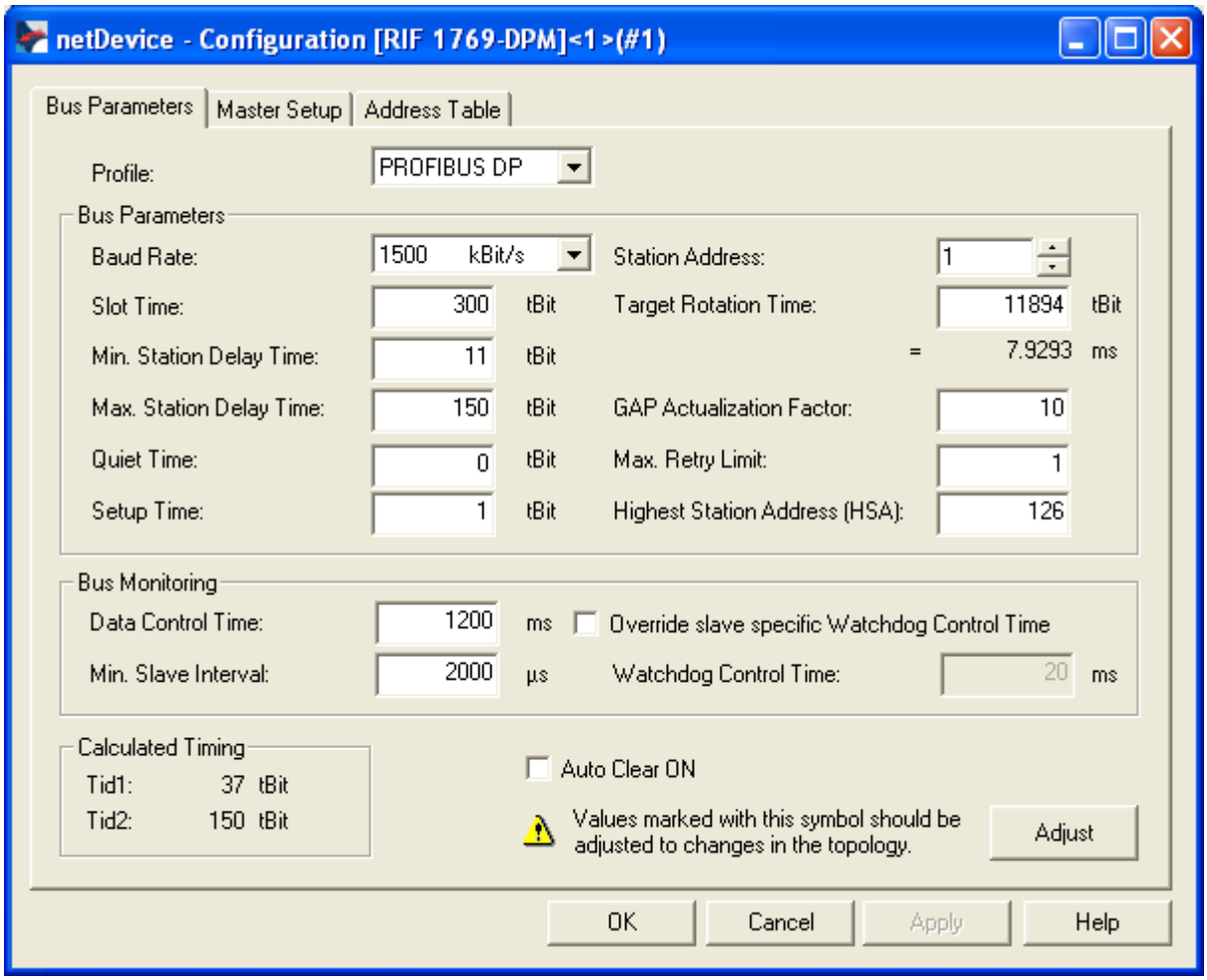

Figure 7 : Master Settings - Bus Parameters Tab

Select the Baud Rate and Station Address for your Master. The rest of the settings do not need adjustment and should be automatically calculate when changing the Baud Rate setting. The default settings cover the most part of applications.

In some cases it is necessary to adjust these values. If these settings need to be changed, please look to the Operating Instruction Manual for Hilscher DTM for PROFIBUS Master Devices and the PROFIBUS specification for the meaning of these values and proper settings.

4.2.3.1 Master Setup Tab

Select the Master Setup tab on this dialog box. The setting will appear as shown in the figure below.

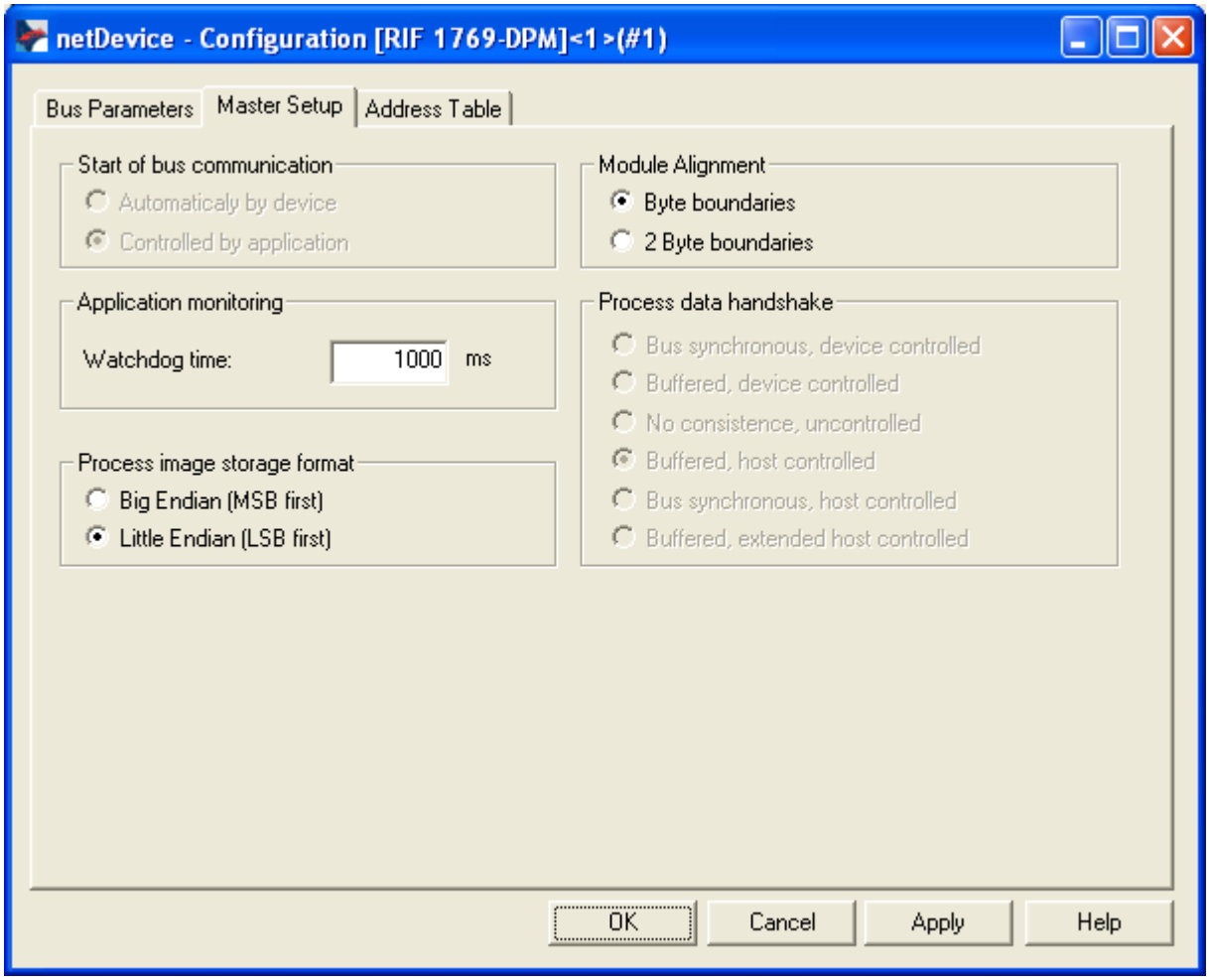

Figure 8 : Master Settings - Master Setup

The only settings available to the user are the following noted in the table below.

Parameter	Selection	Remarks
Module Alignment	Byte Boundaries 2 Byte Boundaries	The **Module Alignment** (addressing mode of the process data image) determines how the addresses (Offsets) of the process data are interpreted. Either of the addressing modes **Byte addresses** or **Word addresses** is possible.
Watchdog	default 1000	The watchdog of the module is triggered by the controller via the RPI. For this reason the RPI must always be smaller than the value of the watchdog. A value of 0 deactivates the watchdog function. The value base is 1ms. The default value of 1000 corresponds to a watchdog time of 1Second.
Storage format	Big Endian Little Endian	For further information see SYCON.net Online help

Table 5 : DP Master Settings

4.2.3.2 Address Table Tab

Select the Address Table tab from the dialog box. The dialog will appear as shown in the following figure.

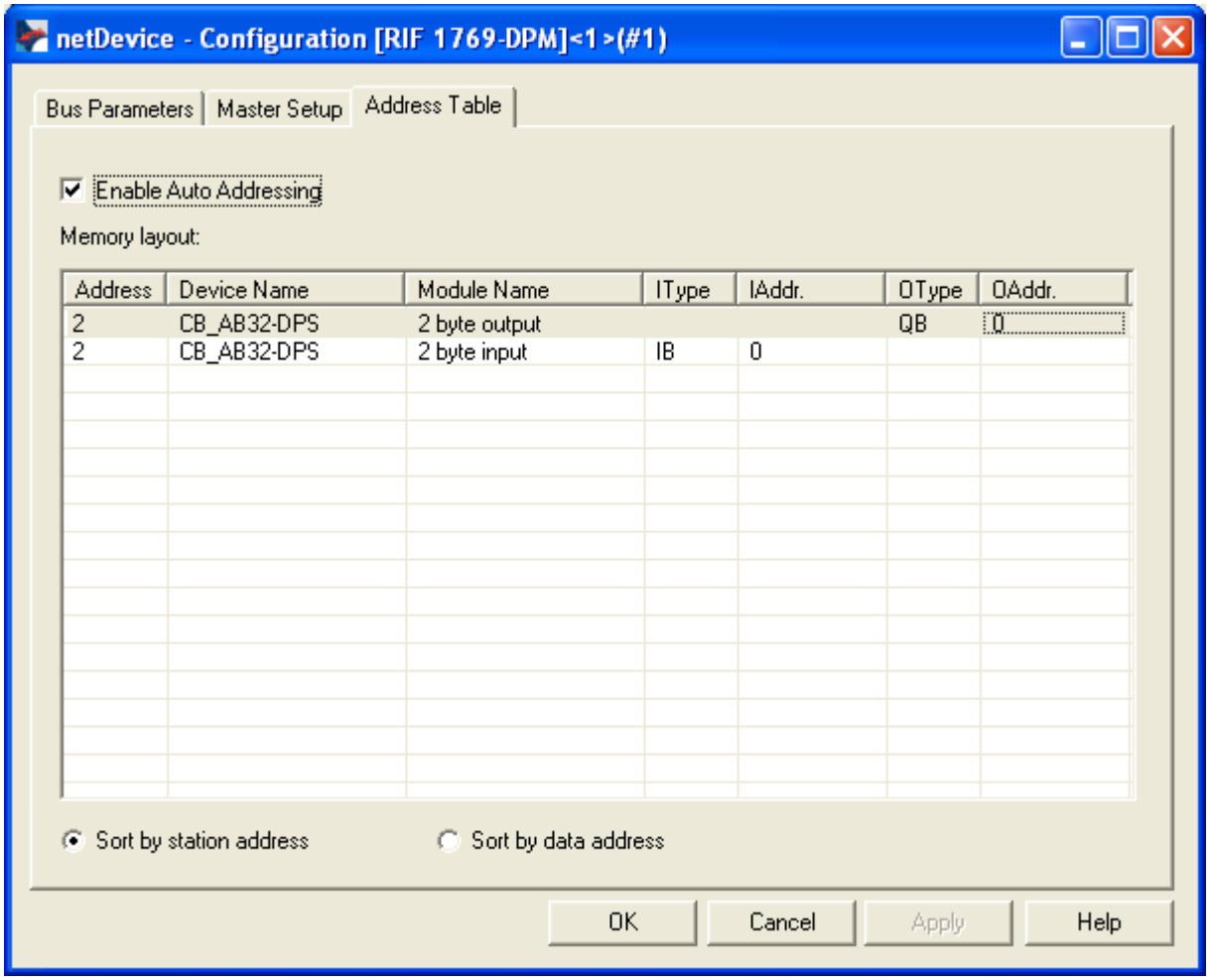

Figure 9 : Master Settings - Address Table Tab

The Address Table tab provides an overview of the Slaves configured and their dual port memory offsets for input and output for output data. The user does not have to change anything in this setup screen as long as the Enable Auto Addressing checkbox is checked. If unchecked, the user can change the offsets manually. Complete the process by clicking the OK or Apply button to close the dialog box. The Master settings are now complete.

4.2.4 Configuration of PROFIBUS Slaves

4.2.4.1 Add a Slave to a project

In the SYCON.net project screen, click and hold the left mouse button and drag a Slave device from the device catalog area to either the netDevice bus or the netProject screen, add the device to the Master by releasing the left button when the + sign appears. Your screen should show the following:

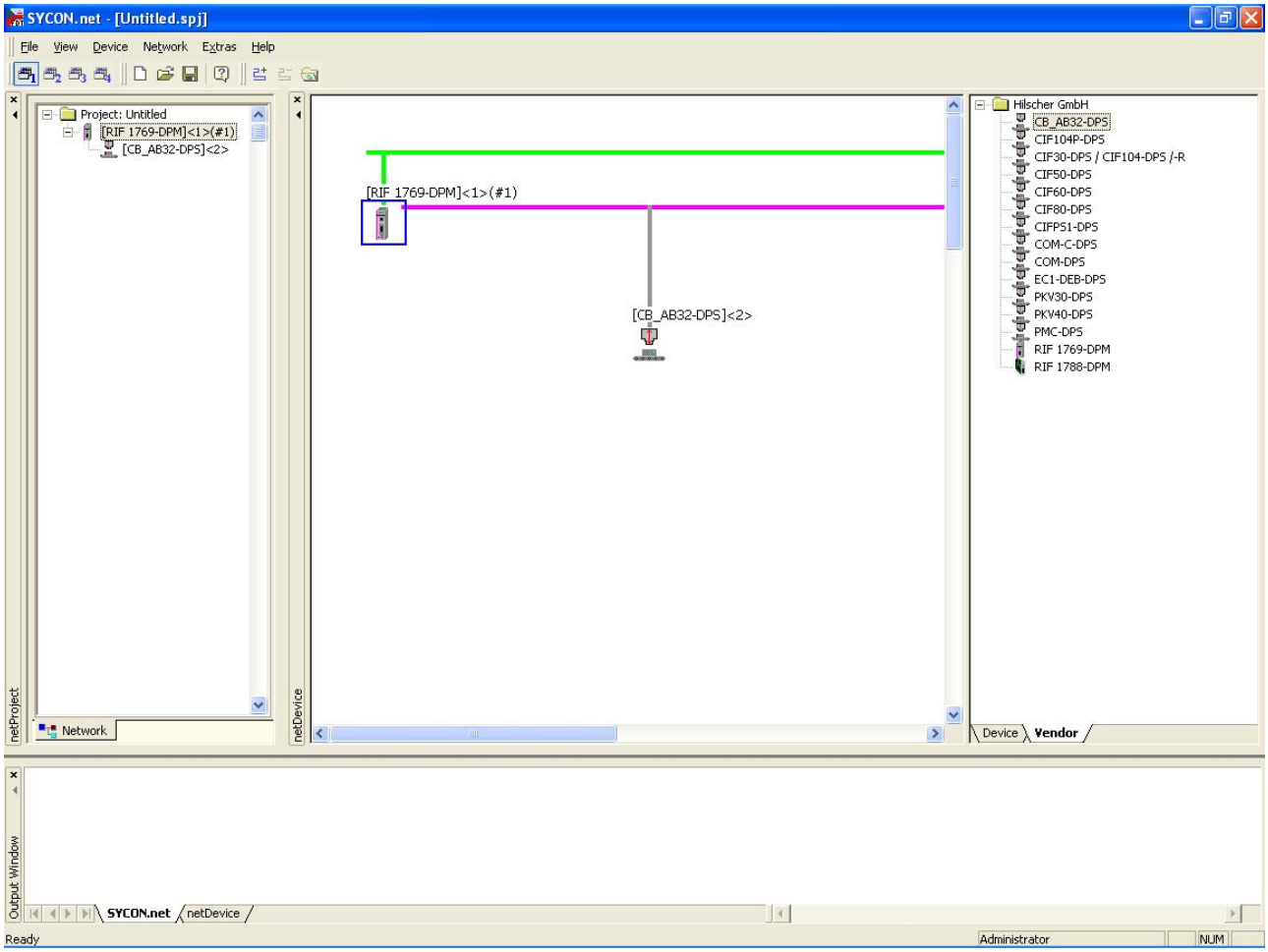

Figure 10 : Add a Slave

4.2.4.2 Add a Slave to SYCON.net Device Catalog

If the PROFIBUS Slave is not listed in the Device Catalog it has to be added to SYCON.net. To add a slave to SYCON.net depends on the configuration method of the slave, which is either the new FDT/DTM technology or typically by the PROFIBUS GSD file. The user will use the GSD file most of the time.

Slave with DTM Technology

If the slave is to be configured by DTM technology then install the DTM software on your PC that was delivered with the slave. Then reload the Device Catalog in SYCON.net

Slave with GSD File (Typical Install)

If you have a GSD file for your slave then perform the following steps:

- Close any open SYCON.net application.
- Copy the GSD file manually into the folder:

  ...\Program Files\Hilscher GmbH\SYCONnet\PBGenericSlaveDTM\GSD

Reload Device Catalog

- Start SYCON.net application.
- Create a new empty project without any device by using the Menu item **File>New**
- Open the SYCON.net Device Catalog with the menu item **Network>Device Catalog...**

Table 6 : Device Catalog

- Click the Reload button.

Now the new slave device is available in the Device Catalog list.

4.2.4.3 Slave Settings Configuration Tab

Double click on the slave that has been added. The following dialog box should appear.

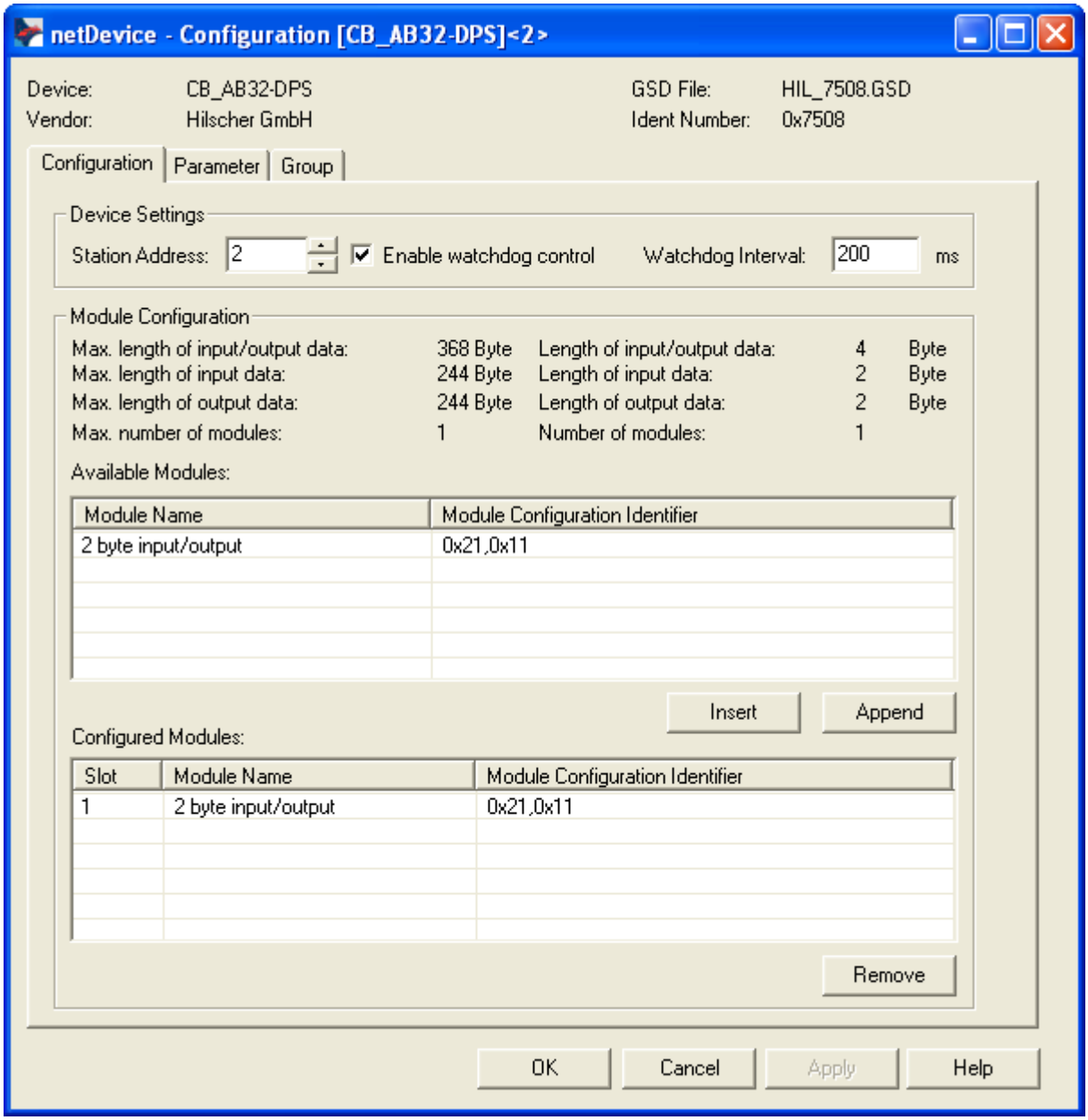

Figure 11 : Slave Settings - Configuration Tab

Insert the appropriate module configuration information and Slave Address.

4.2.4.4 Slave Settings Parameter Tab

Click on the Parameter Tab. The dialog should appear as shown in the figure below.

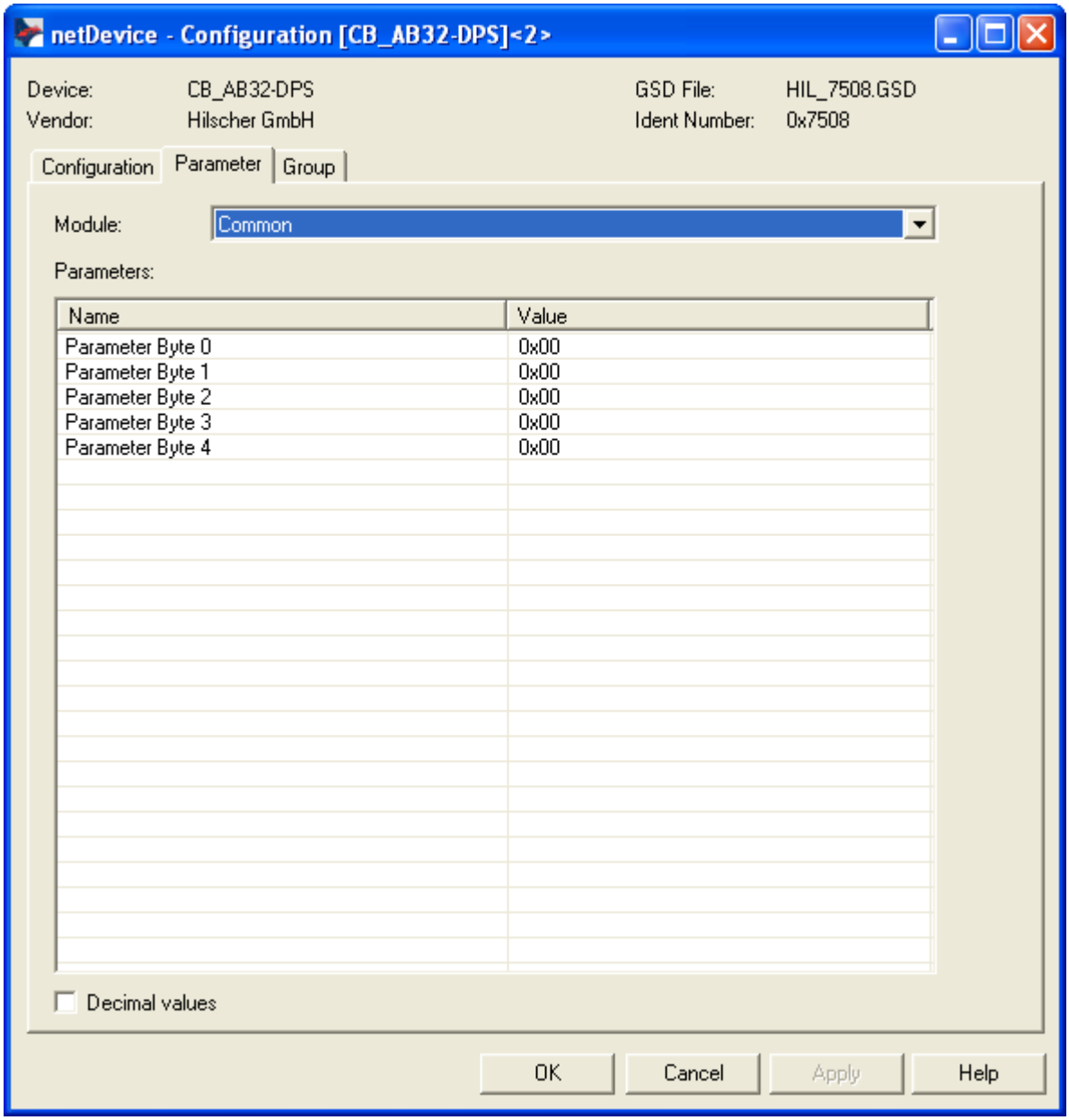

Figure 12 : Slave Settings - Parameter Tab

Verify or change the parameters as required. Normally the user does not have to change anything in this dialog. Parameters are vendor and device specific.

4.2.4.5 Slave Settings Group Tab

Click on the Group Tab. The dialog should appear as shown in the figure below.

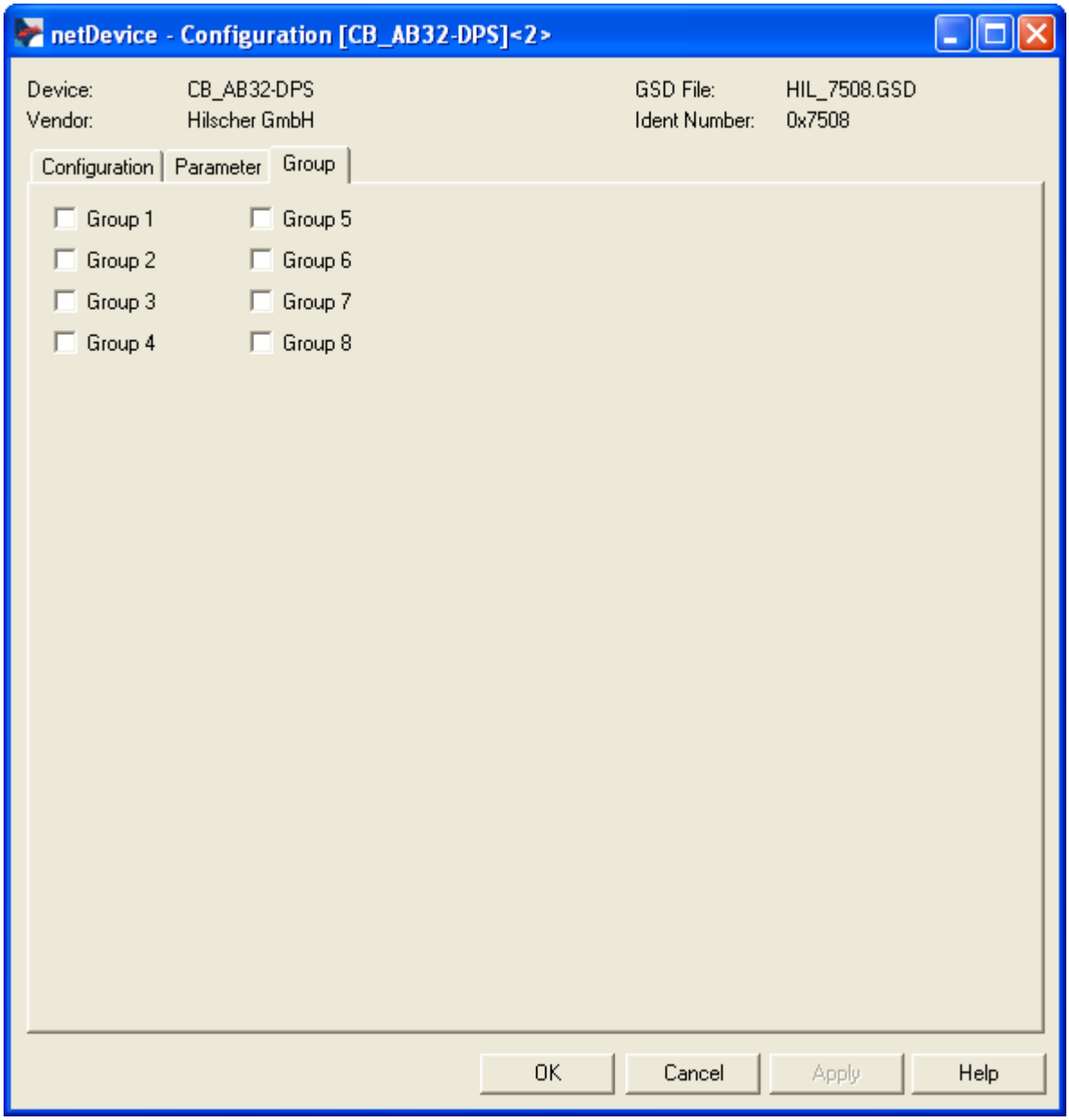

Figure 13 : Slave Settings - Group Tab

A Slave can be assigned as a member of one or more groups. The group membership acts as a filter for the Sync and Freeze global commands. These are output as Broadcast telegrams in order to synchronize the input and output data of several Slaves. Only the Slaves in whose group these commands have been assigned react on it.

Make the appropriate settings for your application. Complete the process by clicking the OK or Apply button to close the dialog box. Repeat the above process for every Slave that needs added to your system. Save the created project with **File > SaveAs..** .

4.2.5 Project Download

Once saved your project is now ready for downloading to the Module. Connect the serial port of your PC to the Diag port on the front of the module. Follow the steps below to download your project.

Step1: Device Assignment

In the SYCON.net project screen, click on the Master you have added in either the netDevice bus or the netProject screen, the Master should be highlighted with a Blue box. Right Click on the Master in either netDevice bus or the netProject screen selecting **Additional Functions>>Device Assignment**. The follow Device Assignment dialog box should appear.

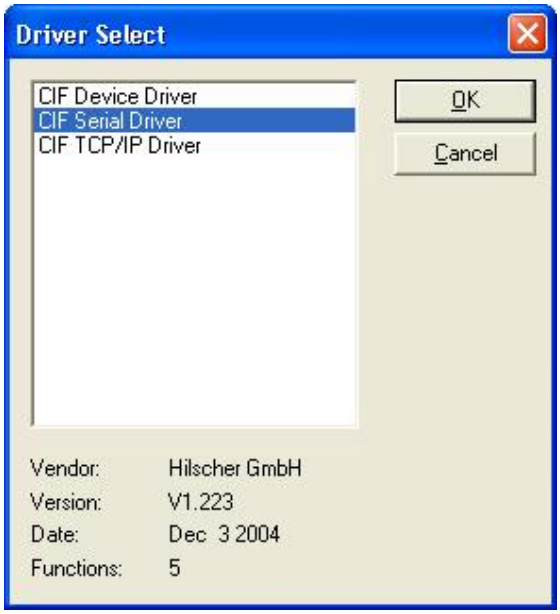

Figure 14 : Project Download - Driver Select

From the Driver Select dialog choose **CIF Serial Driver**. The module will always use the Serial Driver to communicate. Once the OK button is clicked the following dialog will appear.

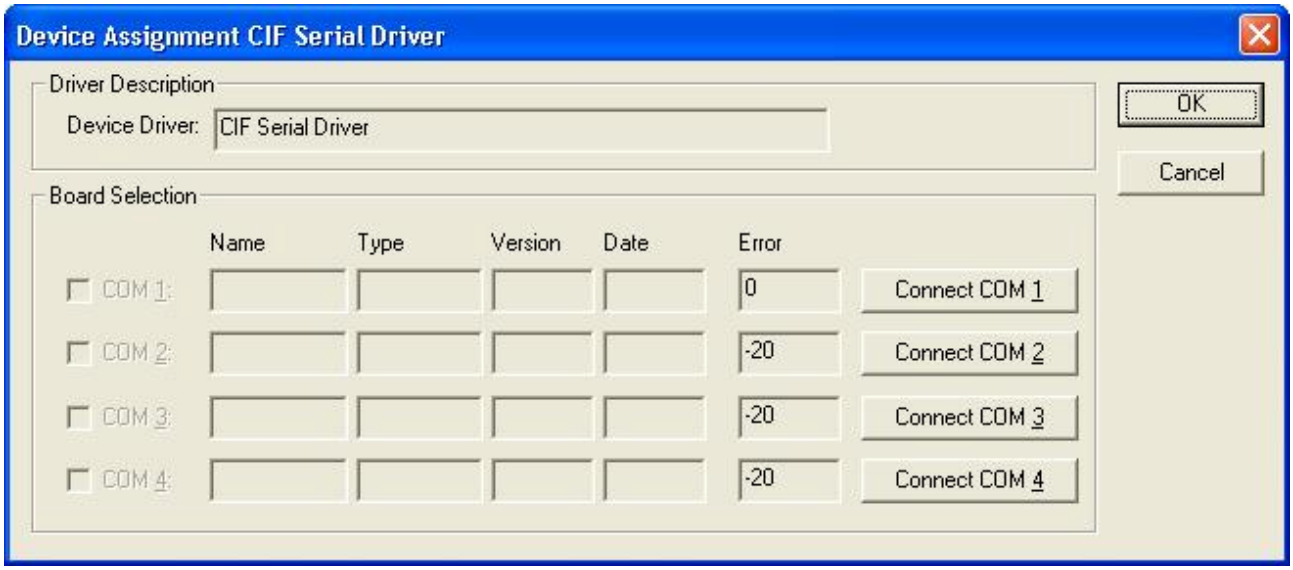

Figure 15 : Project Download - Device Assignment

Using the Device Assignment CIF Serial Driver dialog click on the button Connect COM x which corresponds to the serial port of your PC of which you are using. The following should appear.

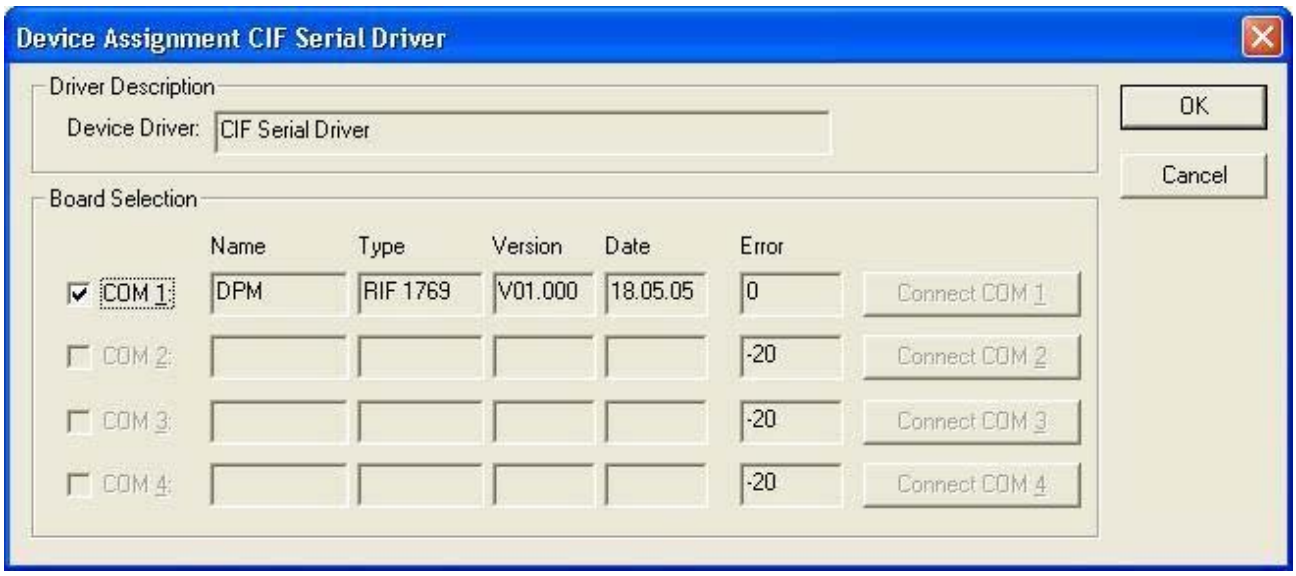

Figure 16 : Project Download - Device Assignment 2

The firmware Name, Type, Version and Date should appear if there is a valid connection to your device. Finally, check the check box next to your device and click the OK button. The Device Assignment process is now complete.

Step2: Connect to Master

In the SYCON.net project screen, click on the Master you have added in either the netDevice bus or the netProject screen, the Master should be highlighted with a Blue box. Using the Menu item **Device>Connect** or Right Click on the Master in either netDevice bus or the netProject screen selecting **Connect** to establish a connection to the Master.

Step3: Download to Device

In the project screen, using the Menu item **Device>Download** or Right Click on the Master in either netDevice bus or the netProject screen selecting **Download** to begin the download process. The following dialog box will appear.

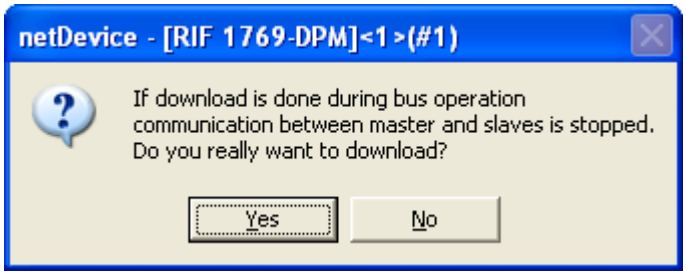

Figure 17 : Project Download - Download Warning

This dialog box is a warning about the bus communications during download. Click Yes to begin the download. The download progress dialog box will appear.

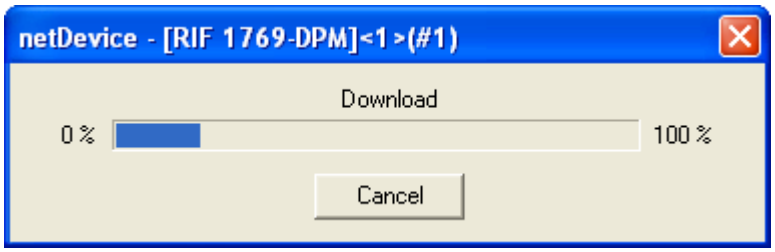

Figure 18 : Project Download - Download Progress

After the Download has been completed, all the required steps for configuration and start up have been done.

5 Communication

5.1 IO Communication and IO Memory Map

Contained in the following sections are the I/O memory mapping for the RIF 1769-DPM interface. The I/O area will be used for communication of status and command information as well as standard I/O.

5.1.1 IO Arrays Overview

5.1.1.1 Input Arrays

Below is a summary of the register layout of the input area of the PROFIBUS Master module. The offset values are defined as byte.

Offset	Register Type	Name
0	Device Status Register	Status Bits
1	Device Status Register	Handshake acknowledge bits
2	Device Status Register	Reserved
3	Device Status Register	Reserved
4	Firmware Revision	Minor Version
5	Firmware Revision	Major Version
6-7	Reserved	Reserved
8	Global State Field	Ctrl
8	Global State Field	Aclr
8	Global State Field	Nexc
8	Global State Field	Fat
8	Global State Field	Eve
8	Global State Field	NRdy
8	Global State Field	Tout
8	Global State Field	Reserved
9	Global State Field	DPM_State
10	Global State Field	Err_rem_adr
11	Global State Field	Err_event
12-13	Global State Field	Bus_err_cnt
14-15	Global State Field	Time_out_cnt
16-23	Global State Field	Reserved[8]
24-39	Global State Field	Sl_cfg[128]
40-55	Global State Field	Sl_state[128]
56-71	Global State Field	Sl_diag[128]
72	Slave Diagnostic Field	Slave Address
73	Slave Diagnostic Field	Slave Diag Failure
74	Slave Diagnostic Field	Station Status_1
75	Slave Diagnostic Field	Station Status_2
76	Slave Diagnostic Field	Station Status_3
77	Slave Diagnostic Field	Master address
78-79	Slave Diagnostic Field	Ident number

Offset	Register Type	Name
80	DPV1 Alarm Indication	Alarm_Status
81	DPV1 Alarm Indication	Rem_Add
82	DPV1 Alarm Indication	Alarm_Cnt
83	DPV1 Alarm Indication	Slot_Number
84	DPV1 Alarm Indication	Seq_Nr
85	DPV1 Alarm Indication	Alarm_Type
86	DPV1 Alarm Indication	Alarm_Spec
87	DPV1 Alarm Indication	Reserved
88-495	PROFIBUS Input Area	Inputs (408 bytes)

Table 7 : Input Register Summary

5.1.1.2 Output Arrays

Below is a summary of the register layout of the output area of the PROFIBUS Master module. The offset values are defined as byte.

Offset	Register Type	Name
0	Device Command Register	Command Bits
1	Device Command Register	Handshake request bits
2	Device Command Register	Reserved
3	Device Command Register	Reserved
4	Slave Diag	Slave Address
5	Slave Diag	Function
6	Slave Diag	Reserved
7	Slave Diag	Reserved
8	Global Control Command	Slave_ Address
9	Global Control Command	Control _Command
10	Global Control Command	Group_Select
11	Global Control Command	Reserved
12	Reserved Register	Reserved
13	Reserved Register	Reserved
14	Reserved Register	Reserved
15	Reserved Register	Reserved
16-495	PROFIBUS Output area	Outputs (480 bytes)

Table 8 : Output Register Summary

5.1.2 Input Arrays

5.1.2.1 Device Status Registers

The RIF 1769-DPM module uses the first 4 bytes of the CPUs input area to transfer Device Status Register information. The Device State Register contains information indicating the modules communication status and command status. The CPUs input area mapping of this information is shown below.

Byte Offset	Structure Member	Data Type	Description
0	MSB	SINT	Module Status Bit
1	HSA	SINT	Handshake Acknowledge Bits
2	Reserved	INT	Reserved
3	Reserved	INT	Reserved

Table 9 : Device State Register

MSB := Module Status Bits

Bit Offset	Structure Member	Data Type	Description
0	Reserved	BOOL	Reserved
1	Reserved	BOOL	Reserved
2	Reserved	BOOL	Reserved
3	Reserved	BOOL	Reserved
4	Reserved	BOOL	Reserved
5	COM	BOOL	Communication
6	RUN	BOOL	Run
7	RDY	BOOL	Ready

Table 10 : Module Status Bits

- **RDY (Ready)**

  When this Bit is set, the module is operational. The RDY-Bit should always be set by the module. If this bit is not set a system error has occurred and the communication between controller and module is not possible.

- **RUN (Run)**

  When the RUN-Bit is set, the module is ready for communication. Otherwise an initialization error or incorrect Parameterization occurs. Further diagnostic is carried out with the SYCON.net configuration tool.

- **COM (Communication)**

  When this Bit is set, the communication is started and the module is engaged in cyclic data exchange with at least one of the connected Slaves.

HSA := Handshake Acknowledge Bits

Bit Offset	Data Type	Structure Member	Description
0	BOOL	HsAck0	SlvDiagCnf ,Slave Diag Confirmation
1	BOOL	HsAck1	GlbCtrCnf , Global Control Confirmation
2	BOOL	HsAck2	Reserved
3	BOOL	HsAck3	Reserved
4	BOOL	HsAck4	Reserved
5	BOOL	HsAck5	Reserved
6	BOOL	HsAck6	Reserved
7	BOOL	HsAck7	Reserved

Table 11 : Handshake Acknowledge Bit

The handshake acknowledge bits provide an indication to the user application if a command has been processed. Every handshake acknowledge bit has a corresponding handshake request bit. A command can be triggered by setting the corresponding handshake request bit in the device command register of the output array. If the handshake acknowledge bit is equal to the corresponding handshake request bit the command has completed and the user program can begin the next command. If unequal, the command is still being processed.

HsAck0 := SlvDiagCnf

This bit indicates the processing of a SlaveDiag request. If this bit is equal to **SlvDiagReq** in the Command register the command has been processed. If unequal, the command is still in progress.

HsAck1 := GlbCtrlCnf

This bit indicates the processing of a Global Control request. If this bit is equal to **GlbCtrlReq** in the Command register the command has been processed. If unequal, the command is still in progress.

HsAck2..7 := Reserved

Reseved for future use.

5.1.2.2 Firmware Revison

This data field, which is part of the input image of the CompactLogix PROFIBUS Master module, will contain the current firmware revision. The Minor revision indication will be in the low byte and the Major revision will be in the high byte. The Firmware Field is placed in the Input area as shown in the table below.

Byte Offset	Structure Member	Data Type	Description
4	FwMajor	SINT	Firmware Major Revision
5	FwMinor	SINT	Firmware Minor Revision
6-7	Reserved	INT	Reserved

Table 12 : Firmware Field

Example:

If FwMajor = 10 and FwMinor = 1 then the firmware revision is 10.1.

Due to a different Hilscher internal firmware numbering scheme than Major/Minor version the following scheme is used to utilize this information to support requirements for a major revision/minor revision. Details are provided in the table below. Because the first release of the modules internal firmware will start with at least V01.000 the first firmware version in Major Minor scheme will be at least 10.00.

Hilscher FW Revision	FW Major	FW Minor
V01.000	10	00
V01.001	10	01

Table 13 : Firmware Major/Minor mapping

5.1.2.3 Global State Field

The 64-byte Global State Field is available to the user program via the input area of the controller. This field contains status information of the PROFIBUS-DP system. It always begins at Byte offset 8 of the input area. The input area mapping of the Global State Field is shown in the table below.

Byte Offset	Structure member	Data Type	Signification	Explanation
8	Global_bits	BOOL	GLOBAL-BITS	Global error bits, for a detailed description of each bit and its meaning see table below
9	DPM_State	SINT	PROFIBUS-DP master state	Main state of the PROFIBUS-DP Master system 00hex: OFFLINE 40hex: STOP 80hex: CLEAR C0hex: OPERATE
10	Err_rem_adr	SINT	Error source	0 ... 125 Error detected with a Slave device 255 Error with Master.
11	Err_event	SINT	Error event	Error number, use the Err_rem_adr value to determine if the error occurred with a connected slave or the Master itself. See error numbers in table below.
12-13	Bus_err_cnt	INT	Bus Error Counter	Number of major bus errors.
14-15	Time_out_cnt	INT	Time Out Counter	Number of bus time outs.
16-23	Reserved[8]	SINT(8)	Reserved	Reserved 8 Bytes
24-39	Sl_cfg[128]	BOOL(128)	slave config bit array	If the Sl_cfg bit of the corresponding slave is logical '1' the slave is configured in the master, and serviced in its states. '0' the slave is not configured in the master
40-55	Sl_state[128]	BOOL(128)	slave state bit array	If the Sl_state bit of the corresponding slave station is logical '1' the slave and the master are exchanging their I/O data. '0' the slave and the master are not exchanging their I/O data. The values in variable Sl_state are only valid, if the master runs the main state OPERATE
56-71	Sl_diag[128]	BOOL(128)	slave diagnostic bit array	If the Sl_diag bit of the corresponding slave station is logical '1' latest received slave diagnostic data are available in the internal diagnostic buffer. This data can be read by the user with a message. '0' since the last diagnostic buffer read access of the HOST, no values were change in this buffer

Table 14 : Global State Field

GLOBAL-BITS

Bit Offset	Member Name	Data Type	Signification	Meaning if Bit is set
0	Ctrl	BOOL	CONTROL-ERROR	Parameterization error
1	Aclr	BOOL	AUTO-CLEAR-ERROR	Module stopped communication with all slaves and reached the auto-clear-end state.
2	Nexc	BOOL	NON-EXCHANGE-ERROR	At least one slave has not reached the data exchange state and no process data will be exchanged.
3	Fat	BOOL	FATAL-ERROR	Because of major bus error, no further bus communication is possible.
4	Eve	BOOL	EVENT-ERROR	The module has detected bus short circuits. The number of detected events is contained in the Bus_error_cnt variable. This bit is set when the first event was detected and will remain set.
5	NRdy	BOOL	HOST-NOT-READY-NOTIFICATION	Indicates if the Application program has set its state to operative or not. If this bit is set the Application program is not ready to communicate
6	Tout	BOOL	TIMEOUT-ERROR	The module has detected an overstepped timeout supervision value because of rejected PROFIBUS telegrams. It is an indication for bus short circuits while the master interrupts the communication. The number of detected timeouts is available in the Time_out_cnt variable. This bit will be set when the first timeout is detected and will remain set.
7	Reserved1	BOOL	Reserved	Reserved

Table 15: Global bits

ERROR-EVENT Codes for an Err_Rem_adr of 255

Code	Indication	Source	Corrective Action
0	No errors are present.	None	None.
50	USR_INTF-Task not found.	Master	Firmware is invalid. Module must be updated.
51	No global data-field.	Master	Firmware is invalid. Module must be updated.
52	FDL-Task not found.	Master	Firmware is invalid. Module must be updated.
53	PLC-Task not found.	Master	Firmware is invalid. Module must be updated.
54	Non existing master parameters.	Master	Execute download of configuration database again.
55	Faulty parameter-value in the master parameters	Configuration	Firmware is invalid. Module must be updated.
56	Non existing slave parameters.	Configuration	Execute download of configuration database again.
57	Faulty parameter-value in a slave parameters data file.	Configuration	Check GSD file for possible incorrect slave parameterization values.
58	Duplicate slave address.	Configuration	Check configured slave addresses in project.

Code	Indication	Source	Corrective Action
59	Projected send process data offset address of a slave is outside the allowable border of 0- 255.	Configuration	Check slave configuration in active project.
60	Projected receive process data offset address of a slave is outside the allowable border of 0- 255.	Configuration	Check slave configuration in active project.
61	Data-areas of slaves are overlapping in the send Process data.	Configuration	Check slave configuration in active project.
62	Data-areas of slaves are overlapping in the receive Process data.	Configuration	Check slave configuration in active project.
63	Unknown process data Handshake.	Master	Problem with master's startup parameters.
64	Free RAM exceeded.	Master	Master has a hardware issue.
65	Faulty slave parameter dataset.	Configuration	Check GSD file for possible incorrect slave parameterization datasets.
202	No memory segment free.	Master	Master has a hardware issue.
212	Faulty reading of a database.	Configuration	Execute download of configuration database again.
213	Structure used by the operating system is faulty.	Master	Master has a hardware issue.
220	Software Watchdog error.	Host	Firmware watchdog has an error.
221	No Data Acknowledge in process data handshake.	Host	Firmware is having trouble with Host acknowledgement.
222	Master in Auto_Clear.	Slave Device	The auto_clear mode was activated, because one slave is missing during runtime.
225	No further Segments.	Master	Contact hotline

Table 16 : ERROR-EVENT Codes for Err_Rem_ Adr equal to 255

ERROR_EVENT Codes for an Err_Rem_Adr of not equal to 255

Code	Indication	Source	Corrective Action
2	Slave station reports data overflow.	Master Telegram	Check length of configured slave parameter or configuration data.
3	Request function of master is not supported in the slave.	Master Telegram	Check if slave is PROFIBUS-DP norm compatible.
9	No answer-data, although the slave must respond with data.	Slave	Check configuration data of the slave and compare it with the physical I/O data length.
17	No response from the slave.	Slave	Check bus cable, and bus address of slave.
18	Master not in the logical token ring.	Master	Check FDL-Address of master or highest-station-address of other master systems. Examine bus cabling for bus short circuits.
21	Faulty parameter in request.	Master Telegram	Master has a firmware issue.

Table 17 : ERROR-EVENT Codes for an Err_Rem_Adr not equal to 255

5.1.2.4 Slave Diagnostics Field

The Slave Diagnostics array is an array of 8 bytes which will include slave diagnostic information based on the settings used to execute this command. The definition of this array and its indications are shown in the Table below.

Note: The same Slave Diagnostic function can also be executed by CIP message functionality described later in this manual. Some types of the CompactLogix controller family do not support messaging. This method can be used for controllers that only support I/O or I/O and CIP messaging. The slave diagnostic via I/O has the limitation that it cannot show extended diagnostic information if a slave supports this. It can only give the mandatory diagnostic information of a slave. Extended diagnostic information can be received with the CIP message functionality.

Byte Offset	Structure member	Data Type	Description
72	Slave Address	SINT	Address of Slave with the Diagnostic request.
73	Slave Diag Failure	SINT	See definition below.
74	Station Status_1	SINT	See definition below.
75	Station Status_2	SINT	See definition below.
76	Station Status_3	SINT	See definition below.
77	Master Address	SINT	This byte contains the master address of the PROFIBUS-DP master which has done the parameterization of the slave. If a slave is not parameterized the value is 255.
78-79	Ident Number	INT	PROFIBUS Ident number from Slave in which the diagnostic request was made.

Table 18 : Slave Diagnostics Field

Slave Diag Failure

This byte reflects the status of the DDLM_DIAG request. See table below for possible error codes.

Error Code	Significance	Error source	Help
0	Service could be executed without an error	No error	-
17	No response from the station	SLAVE	Check network wiring, check bus address of slave or baud rate support
18	Master not into the logical token ring	Network in general	Check master DP-Address or highest-station-address of the Master. Examine bus wiring for bus short circuits
161	Remote Address in request service out of range	APPLICA-TION	Check address parameter in diag request

Table 19 : Slave Diag Failure

Station Status_1

This status byte will be zero indicating that the slave device has no errors. The non-zero values which are errors are defined in the table below.

Bit Offset	Member Name	Data Type	Meaning if Bit is set
0	Sta_Non_Exist	BOOL	No response from slave device. The station is non existent.
1	Sta_Not_Ready	BOOL	Slave not ready.
2	Cfg_Fault	BOOL	Slave has incorrect parameterization.
3	Ext_Diag	BOOL	The extended diagnostics area is used.
4	Not_Supp	BOOL	Unknown command is detected by the slave.
5	Inv_Slv_Res	BOOL	Invalid slave response.
6	Prm_Fault	BOOL	Last parameterization telegram was faulty.
7	Master_Lock	BOOL	Slave is controlled by another master.

Table 20 : Station Status_1

Station Status_2

Bit Offset	Member Name	Data Type	Meaning if Bit is set
0	Prm_Req	BOOL	Slave must be parameterized.
1	Stat_Diag	BOOL	This bit remains active until all diagnostic data has been retrieved from the slave.
2	Slave_Device	BOOL	This bit is always set by the Slave.
3	WD_On	BOOL	Slave watchdog is activated.
4	Freeze_Mode	BOOL	Freeze command active.
5	Sync_Mode	BOOL	Sync command active
6	Reserved	BOOL	Reserved.
7	Deactivated	BOOL	Slave not active.

Table 21 : Station Status_2

Station Status_3

Bit Offset	Member Name	Data Type	Meaning if Bit is set
0 .. 6	Reserved0..6	BOOL	Reserved
7	ExtDiagOverflow	BOOL	Slave has a large amount of diagnostic data and cannot send it all

Table 22 : Station Status_3

5.1.2.5 DPV1 Alarm Indication

The DPV1 alarm indication register is mapped to 8 bytes of the input area. These registers provide incoming alarm indication data required to respond to a DPV1 alarm created and sent by a Slave device. The definition of registers contained within the Alarm Indication is detailed in the Table below.

Byte Offset	Member Name	Data Type	Description
80	Alarm_Status	SINT	Status of Alarm pending
81	AlarmCnt	SINT	AlarmCounter
82	Rem_Add	SINT	Address of Slave with Alarm (0-126)
83	Slot_Number	SINT	Slot Number (0-254)
84	Seq_Nr	SINT	Sequence Number (0-31)
85	Alarm_Type	SINT	Alarm Type (1-6, 32-126)
86	Alarm_Spec	SINT	Alarm Specification (0-7)
87	Reserved	SINT	Reserved

Table 23 : DPV1 Alarm Indication Registers

Alarm_Status

Bit Offset	Member Name	Data Type	Description
0	AlarmInd	BOOL	Alarm Indication
1	Reserved	BOOL	Reserved
2	Reserved	BOOL	Reserved
3	Reserved	BOOL	Reserved
4	Reserved	BOOL	Reserved
5	Reserved	BOOL	Reserved
6	Reserved	BOOL	Reserved
7	AlarmOverrun	BOOL	Alarm Overrun

Table 24 : Alarm Status Bits

The alarm indication registers start with the byte “Alarm_Status”. This byte is a collection of bits to indicate alarm status. Bit0 (AlarmInd) is set to “1” to indicate to the user application that an alarm request has been received by the module. The specifics of the alarm request will be present in the remaining alarm indication fields, which contain the slave station address “Rem_Add”, the slot number “Slot_Number”, etc. according to the PROFIBUS specification. If an alarm is indicated, the user application has to decide what to do with its application specific reaction. The application has to respond to the alarm with a CIP message, which is described later in this manual. With its response, the application confirms to the slave that the alarm was received. When the user application responds to the alarm using a CIP message Bit0 will be set to “0” indicating that the alarm has been acknowledged and is no longer pending. The alarm information Rem_Adr, Slot_Number, etc. is not cleared. It can happen that a second or more alarms are pending. In this case Bit0 will not be reset to “0” when the application has responded to one alarm. Only the alarm information Rem_Add, Slot_Number etc. will be update in the case of multiple alarms pending.

To handle this situation the application has to look also to the variable “AlarmCnt”. This counter will be incremented every time a new alarm is pending.

NOTE: It is possible to receive several alarms from one or multiple slaves. The module has a buffer for only 32 alarms. The application has to respond as fast as possible to the alarms. If not and the internal alarm buffer runs over it comes to a lost of alarms. This is indicated “Alarm_Status” in bit D7 “AlarmOverrun”. If an alarm was loss, this bit goes to static “1”. It will be cleared only if the bus communication is stopped for example if the controller goes to stop or the application stops the bus communication by the NRDY bit in the command register.

5.1.2.6 PROFIBUS Input Data

The remainder of the input area is used for the PROFIBUS input data from the connected Slaves. The input information is transferred from the module to the controller. Input data from the PROFIBUS system always starts at the 88^{Th} Byte (based on Start Index 0) in the input region. Thus, the module has a maximum of 408 Byte input data (496Byte input region – 88 Byte status). The input data of the Slaves are linear corresponding to the I/O Mapping assigned by the SYCON.net configuration tool. SYCON.net is capable of configuring more than 408 Bytes of input data. Should the input data of the system be greater, only the first 408 Bytes of the input data will be transferred to the controller.

5.1.3 Output Arrays

5.1.3.1 Device Command Register

The Device Command Register is transferred from the controller to the module via the output region. The Command register always lies in the first 4 Bytes of the output region. Follows is the mapping for the Device Command Register.

Byte Offset	Structure Member	Data Type	Description
0	MCB	SINT	Module Command Bits
1	HSR	SINT	Handshake Request Bits
2	Reserved	INT	Reserved
3	Reserved	INT	Reserved

Table 25 : Device Command Register

MCB := Module Command Bits

Bit Offset	Structure Member	Data Type	Description
0	Reserved	BOOL	Reserved
1	Reserved	BOOL	Reserved
2	Reserved	BOOL	Reserved
3	Reserved	BOOL	Reserved
4	Reserved	BOOL	Reserved
5	NRDY	BOOL	Application not ready
6	INIT	BOOL	Init
7	RST	BOOL	Reset

Table 26 : Module Command Bits

NRDY := Not Ready

With this Bit, the user program can start or stop communication with the PROFIBUS system. When this Bit is set from the user program, the communication between the module and all Slave devices connected, is stopped. All slaves will clear their outputs and the Master will be in Stop mode. This control bit allows the user program to make a controlled start of the communication with the PROFIBUS system.

INIT := Init

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

RST := Reset

The user program can use this bit to execute a Reset (Cold Start) of the module.

Attention: Using the Reset command will cause an interruption in bus communication. All connected slaves will clear their outputs.

HSR := Handshake Request Bits

Bit Offset	Data Type	Structure Member	Description
0	BOOL	HsReq0	SlvDiagReq, Slave Diag Request
1	BOOL	HsReq1	GlbCtrReq, Global Control Request
2	BOOL	HsReq2	Reserved
3	BOOL	HsReq3	Reserved
4	BOOL	HsReq4	Reserved
5	BOOL	HsReq5	Reserved
6	BOOL	HsReq6	Reserved
7	BOOL	HsReq7	Reserved

Table 27 : Device Command Register "Handshake Request Bits"

With the handshake request bits the user application can trigger different functions supported by the module. Every handshake request bit HsReq has a corresponding handshake acknowledge bit HsAck in the DeviceStatusRegister in the input array. If the module set an HsReq bit unequal to the corresponding HsAck bit, the module will execute the command. If the module sets the corresponding HsAck bit equal to the HsReq bit then the module has executed the command and the application can execute another command.

HsReq0 := Slave Diag Request

The user program can use this bit to execute a Slave Diagnostic request. This bit is to be used with the SlvDiagCnf bit in the Handshake Acknowledge Bits to determine if the command has been processed. See section on Device Status Registers for more details.

HsReq1 := Global Control Request

The user program can use this bit to execute a Global Control command. This bit is to be used with the GlbCtrCnf bit to determine if the command has been processed. See section on Device Status Registers for more details.

HsReq0..7 := Reserved

Reserved for future use.

5.1.3.2 Slave Diagnostic Request Register

The Slave Diagnostic Request register in combination with the handshake request bits is used to provide the user program with an easy method to execute a diagnostic request to a specified slave in the system.

The slave diag command can be executed by using the corresponding handshake request bit HsReq0 in the “Device Command Register”. The result of the slave diag command can be read in the Slave Diagnostic field of the input array.

Byte Offset	Data Type	Structure Member	Description
4	SINT	SLA	Slave Address
5	SINT	FNC	Function
6	SINT	Reserved	Reserved
7	SINT	Reserved	Reserved

Table 28 : Slave Diagnostic Request Register

SLA := Slave Address

Address of the slave of which the diagnostic data are requested.

FNC := Function

If FNC is 0 the diagnostic data are requested from internal buffer of the Master. This is the recommended method, because the master has always the most recent diagnostic data in its internal buffer from the Slave.

If FNC is 1 the diagnostic data are requested directly from the slave itself. This is not the preferred method because, this method causes additional bus loading and will influence the bus cycle time.

Note: The same Slave Diagnostic function can also be executed by CIP message functionality described later in this manual. Some types of the CompactLogix controller family do not support messaging. This method can be used for controllers that only support I/O or I/O and CIP messaging. The slave diagnostic via I/O has the limitation that it cannot show extended diagnostic information if a slave supports this. It can only give the mandatory diagnostic information of a slave. Extended diagnostic information can be received with the CIP message functionality.

5.1.3.3 Global Control Array

The Global Control Array is an array of 4 bytes following the Device Command Register. The Global Control request makes it possible, to send commands to one or several DP slaves. A DP slave accepts a control command only from the DP master that has parameterized it. This request makes it possible to do Sync and Freeze functions.

Note: The same Global Control function can be also executed by CIP message functionality described later in this manual. Some types of the CompactLogix controller family do not support messaging. This method is used only for controllers that support I/O not messaging.

Byte Offset	Data Type	Structure Member	Description
8	SINT	SLA	Slave Address
9	SINT	CC	Function
10	SINT	GS	Reserved
11	SINT	Reserved	Reserved

Table 29 : Global Control Array

SLA := Slave Address

The parameter SLA allows the user to set the address of the Slave in which the Global Control command is to be sent. The value of 127 is a special global broadcast address. When this address is selected, all Slaves are effected by this command at the same time.

CC := Control Command

The parameter CC determines the function that is to be executed when using the Global Control Command.

Bit Offset	Data Type	Structure Member	Description
0	BOOL	Reserved	Reserved
1	BOOL	Clear_data	Clear output data
2	BOOL	UnFreeze	Unfreeze input data
3	BOOL	Freeze	Freeze input data
4	BOOL	Unsync	Neutralize the sync command or unsync
5	BOOL	Sync	Freeze output data, until sync command is neutralized
6	BOOL	Reserved	Reserved
7	BOOL	Reserved	Reserved

Table 30 : Control Command

Combination of the unsync/sync and unfreeze/freeze bits

Bit 2 or 4	Bit 3 or 5	Meaning
0	0	No function
0	1	Function (sync or freeze) is active
1	0	Function (unsync or unfreeze) is active
1	1	Function (unsync or unfreeze) is active

Table 31 : Bit Combination

GS := Group_Select

The parameter GS allows the user program to select which of the 8 possible slave groups is addressed by this service. This command is activated in the Slave when the AND linkage between its internal Group_Ident and the desired Group_Select logic result is a '1'. The Group_Ident parameter is configured by the Master during the Startup phase. If the Group_Ident parameter is set to a value of ‘0’, the Slave does not carry out a group evaluation (AND linkage) with the received command.

The global control command is processed by using the two handshake bits GlbCtrlReq in the device command register and GlbCtrCnf in the device status register. The command is sent on every High to Low **and** Low to High transition of the GlbCtrlReq bit. If both bits are equal a command can be sent. To send a command setup the Global_Control_array with the desired command. Set the bits GlbCtrlReq and GlbCtrlCnf to unequal by transition of the GlbCtrlReq bit. If the GlbCtrlCnf was set equal to the GlbCtrlReq bit the command was sent. The truth table below provides an explaination of this process.

GlbCtrlReq	GlbCtrlCnf	Meaning
0	0	No Control_Command is active. Next command can be send.
1	0	Control command in progress.
1	1	No Control_Command is active. Next command can be send
0	1	Control command in progress.
0	0	Process repeats

Table 32 : Global_Control_Command Truth Table

5.1.3.4 PROFIBUS Output Data

The remainder of the output area is used for the PROFIBUS output data for the connected Slaves. The output information is transferred from the controller to the module. Output data from the PROFIBUS system always starts at the 16Th Byte (based on Start Index 0) in the output region.

Thus, the module has a maximum of 480 Byte output data that it can use for Slave devices. The output data of the Slaves are arranged in this area according to the I/O Mapping table assigned by the SYCON.net configuration. SYCON.net is able to configure more than 480 Bytes of output data. Should the output data be greater, the controller will only deal with the first 480 Bytes.

5.2 CIP Messaging

PROFIBUS-DP supports acyclic services through messages. These PROFIBUS-DP services are supported by the RSLogix5000 programming tool by means of CIP messages using the “MSG” instruction. The outline and usage of these commands for the PROFIBUS-DP Master are contained with in this section.

5.2.1 Using the MSG Instruction in RSLogix

CIP messages are possible by the use of the “MSG” function block in RSLogix5000. The “MSG” function block can be found under Input/Output Instructions within the RSLogix Instruction Set. The MSG instruction asynchronously reads or writes a block of data to another module on a network. The following is an example of how this instruction is assembled using the acyclic PROFIBUS-DP service DDLM_Slave_Diag command.

Step1: Create New Controller Tag

Double click on the Controller Tags tree selection under Controller CompactLogix. The Controller Tags dialog box will appear. Select the Edit Tags tab. Add a new tag called SlaveDiagMsg and make its Type equal to MESSAGE.

Step2: Insert the “MSG” instruction

From the language element tool bar in RSLogix select the Input/Output tab and click on the “MSG” button. The instruction will be inserted into your ladder logic as shown in the figure below.

Figure 19 : "MSG" Instruction

Select the ? And enter the MESSAGE type created SlaveDiagMsg as shown below.

Figure 20 : "MSG" Instruction with SlaveDiagMsg

Step3: Message Configuration

Select the button ..., which will open the Message Configuration Dialog. The configuration dialog will allow the user to input the appropriate information needed to execute the SlaveDiagMsg. The entries should be as follows.

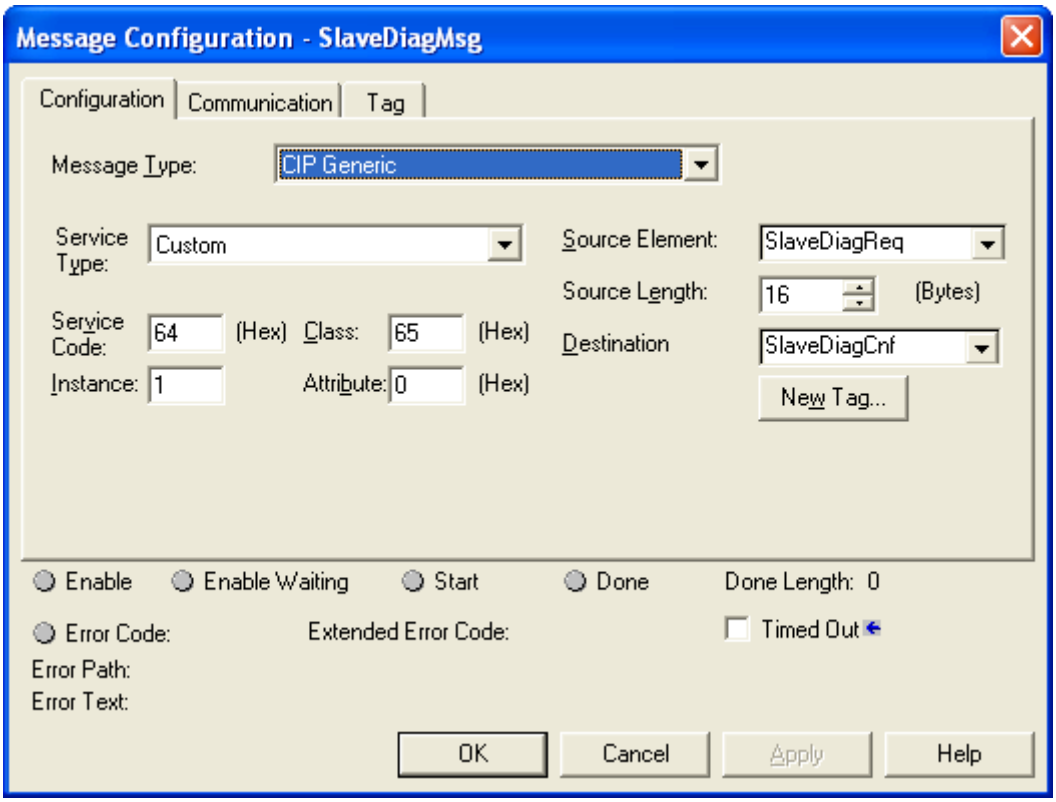

Figure 21 : Message Configuration - Configuration Tab

Note: The user must create two user defined data types to send and receive the information for this command message. In this example SlaveDiagReq and SlaveDiagCnf were created to hold the command specific information.

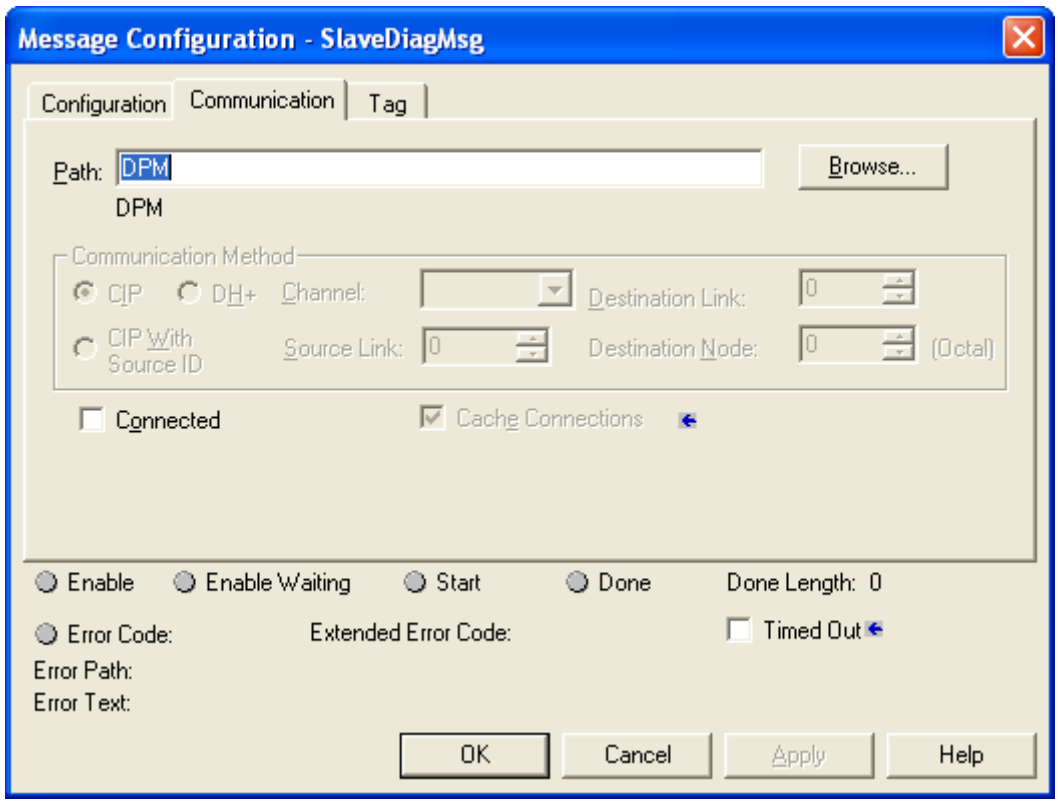

Figure 22 : Message Configuration - Communication Tab

The Path in the dialog above must point to the 1769-Module. Use the Browse button to select the path.

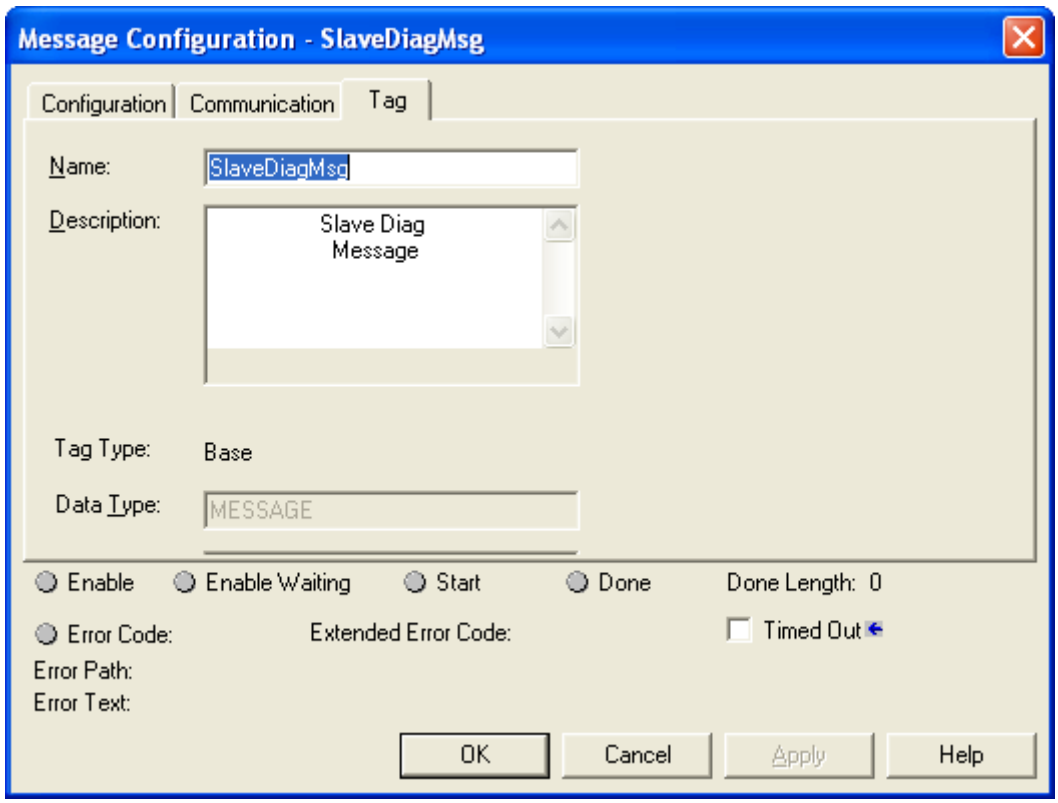

Figure 23 : Message Configuration - Tag Tab

Step4: Add Logic to Execute MSG Instruction

With the “MSG” instruction now configured the user can add the required logic needed to execute the instruction. The example below shows the “MSG” instruction used in the example logic in RIF_1769_DPM_Messaging_L32E.ACD.

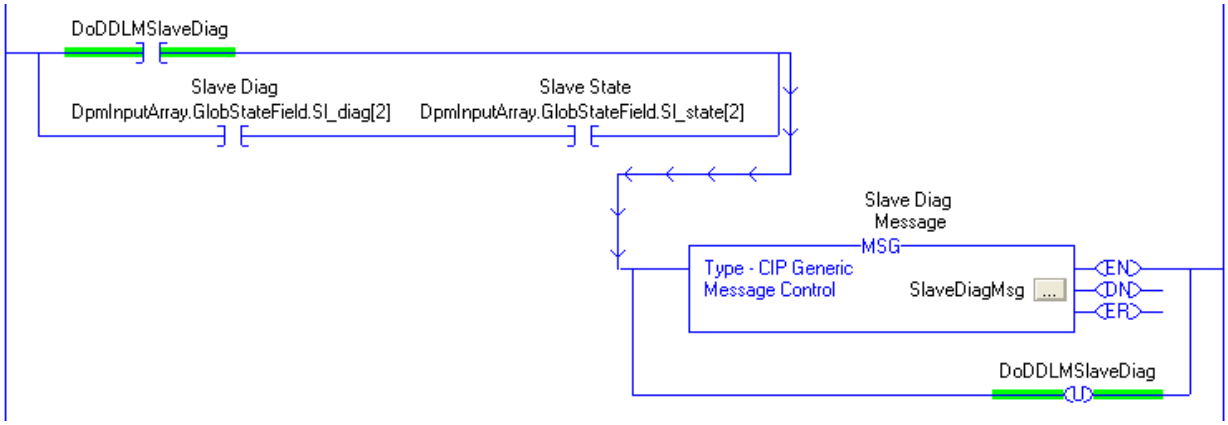

Figure 24 : Example MSG Logic

5.2.2 Supported PROFIBUS-DP Messages

The section shall define the message functions supported by the CompactLogix Master module. Below is a summary of the functions that are supported.

Service	Cmd Code	Group	Description
DDLM_Slave_Diag	66	DDLM	Reading out the diagnostic information from a DP Slave
DDLM_Global_Control	70	DDLM	Sending a command to one or several DP Slaves
DDLM_Set_Parameter	74	DDLM	Sending parameter data to a specific DP Slave during its run time
MSAC1_Read	17	DPV1	With this service, a read request for a particular data block is sent to a DPV1 Slave. This service works Slot- and Index-referenced.
MSAC1_Write	17	DPV1	With this service, a write request is transferred to a DPV1 Slave, to write a particular data block in the DPV1 Slave. This service works Slot- and Index-referenced.
MSAL1M_Alarm_Res	18	DPV1	This service provides the means to acknowledge a DPV1 Alarm indication sent to the Master by a Slave.

Table 33 : Supported PROFIBUS Messages

Note: Contained with in the “RIF_1796_DPM_messaging_L32E.ACD” project is an example for each of these services.

5.2.3 Standard Messaging

Provided in the sections below are the descriptions of each Standard Message “DDLM” supported by the PROFIBUS Master module.

5.2.3.1 DDLM_Slave_Diag

The DDLM_SLAVE_Diag request is used to query the status of a PROFIBUS Slave by using its address on the bus. This request can be used to determine the general health of the slave device. The MSG instruction Request /Confirmation format is as follows.

DDLM_SLAVE_DIAG_REQUEST

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Reserved3	INT	0	Reserved
Command	SINT	66	Command for the DDLM_Slave_Diag service.
Reserved4	SINT	0	Reserved
DeviceAdr	SINT	0..125	Address of the PROFIBUS Slave.
DataArea	SINT		Reserved
DataAdr	INT		Reserved
DataIdx	SINT		Reserved
DataCnt	SINT		Reserved
DataType	SINT		Reserved
Function	SINT	1,3	1 Read the diagnostic information from the internal buffer of the Master. 3 Read the diagnostic information directly from the Slave.

Table 34 : DDLM_Slave_Diag_Request

DDLM_SLAVE_DIAG_CONFIRM

Name	Data Type	VALUE	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Answer	SINT	66	Answer DDLM_Slave_Diag
Failure	SINT	e	Error, status (see following section)
Reserved3	INT	0	Reserved
DeviceAdr	SINT	0 .. 125	Slave address
DataArea	SINT	0	Reserved
DataAdr	INT	0	Reserved
DataIdx	SINT	0	Reserved
DataCnt	SINT	6 + x	Length of the diagnostic structure (starting with StationState_1)
DataType	SINT	0	Reserved
Function	SINT	1,3	Read function
StationState_1	SINT	S1	Station status_1
StationState_2	SINT	S2	Station status_2
StationState_3	SINT	S3	Station status_3
MasterAddress	SINT	MA	Master address
IdentNumber	INT	ID	Ident number
Reserved4	INT	0	Reserved
ExtDiag[0..99]	SINT	EX	Extended diagnostic

Table 35 : DDLM_Slave_Diag_Confirmation

MA := Master Address

This Byte contains the address of the Master that has parameterized the Slave.

ID := Ident Number

In this word the Slave answers with its Ident Number.

EX:= Extended Diagnostic

EX is an extended diagnostic buffer. Valid values can be found in the manual of the corresponding Slave or can be found in the PROFIBUS specification.

S1 := Station Status_1

This status byte will be zero indicating that the slave device has no errors. The non-zero values which are errors are defined in the table below.

Bit Offset	Member Name	Data Type	Meaning if Bit is set
0	Sta_Non_Exist	BOOL	No response from slave device. The station is non existent.
1	Sta_Not_Ready	BOOL	Slave not ready.
2	Cfg_Fault	BOOL	Slave has incorrect parameterization.
3	Ext_Diag	BOOL	The extended diagnostics area is used.
4	Not_Supp	BOOL	Unknown command is detected by the slave.
5	Inv_Slv_Res	BOOL	Invalid slave response.
6	Prm_Fault	BOOL	Last parameterization telegram was faulty.
7	Master_Lock	BOOL	Slave is controlled by another master.

Table 36 : Station_status_1

S2 := Station Status_2

Bit Offset	Member Name	Data Type	Meaning if Bit is set
0	Prm_Req	BOOL	Slave must be parameterized.
1	Stat_Diag	BOOL	This bit remains active until all diagnostic data has been retrieved from the slave.
2	Slave_Device	BOOL	This bit is always set by the Slave.
3	WD_On	BOOL	Slave watchdog is activated.
4	Freeze_Mode	BOOL	Freeze command active.
5	Sync_Mode	BOOL	Sync command active
6	Reserved	BOOL	Reserved.
7	Deactivated	BOOL	Slave not active.

Table 37 : Station_status_2

S3 := Station_status_3

Bit Offset	Member Name	Data Type	Meaning if Bit is set
0 .. 6	Reserved0..6	BOOL	Reserved
7	ExtDiagOverflow	BOOL	Slave has a large amount of diagnostic data and cannot send it all

Table 38 : Staion_status_3

The CIP MSG Parameterization of this request is as follows.

Parameter	Value	Remarks
Message Type	CIP Generic	
Service Type	Custom	
Service Code	64 hex	Service Code “Bridge Message”
Class	65 hex	CIP Object “CIP_MSG_BRIDGE”
Instance	1	
Attribute	0	
Source Element	SlaveDiagReq	Reference to a Tag of type DDLM_SLAVE_DIAGNOSTIC_REQUEST
Destination	SlaveDiagCnf	Reference to a Tag of type DDLM_SLAVE_DIAG_CONFIRM
Source Length	16	Corresponds to the size of the DDLM_SLAVE_DIAGNOSTIC_REQUEST structure

Table 39 : CIP Message Parameters for DDLM_Slave_Diag

5.2.3.2 DDLM_Global_Control

The DDLM_Global_Control request makes it possible, to send commands to one or several DP slaves. A PROFIBUS-DP slave accepts a control command only from the DP master which has parameterized it. This request makes it possible to do Sync and Freeze functions. The MSG instruction Request / Confirmation format is as follows.

DDLM_GLOBAL_CONTROL_REQUEST

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Reserved3	INT	0	Reserved
Command	SINT	70	Command for the DDLM_Global_Control service
Reserved4	SINT	0	Reserved
DeviceAdr	SINT	SLA	Address of the PROFIBUS Slave
Conrol-Command	SINT	CC	Control_Command
GroupSelect	SINT	GS	Group Select

Table 40 : Parameter DDLM_Global_Control Request

SLA := Slave Address

The parameter SLA allows the user to set the address of the Slave in which the Global Control command is to be sent. The value of 127 is a special global broadcast address. When this address is selected, all Slaves are effected by this command at the same time.

CC := Control Command

The parameter CC determines the function that is to be executed when using the Global Control Command.

Bit Offset	Data Type	Structure Member	Description
0	BOOL	Reserved	Reserved
1	BOOL	Clear_data	Clear output data
2	BOOL	UnFreeze	Unfreeze input data
3	BOOL	Freeze	Freeze input data
4	BOOL	Unsync	Neutralize the sync command or unsync
5	BOOL	Sync	Freeze output data, until sync command is neutralized
6	BOOL	Reserved	Reserved
7	BOOL	Reserved	Reserved

Table 41 : Control Command

Combination of the unsync/sync and unfreeze/freeze bits

Bit 2 or 4	Bit 3 or 5	Meaning
0	0	No function
0	1	Function (sync or freeze) is active
1	0	Function (unsync or unfreeze) is active
1	1	Function (unsync or unfreeze) is active

Table 42 : Bit Combination

GS := Group_Select

The parameter GS allows the user program to select which of the 8 possible slave groups is addressed by this service. This command is activated in the Slave when the AND linkage between its internal Group_Ident and the desired Group_Select logic result in a '1'. The Group_Ident parameter is configured by the Master during the Startup phase. If the Group_Ident parameter is set to a value of ‘0’, the Slave does not carry out a group evaluation (AND linkage) with the received command.

DDLM_GLOBAL_CONTROL_CONFIRM

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Answer	SINT	70	Answer DDLM_Global_Control
Failure	SINT	0	Error, status
Reserved3	INT	0	Reserved
DeviceAdr	SINT	0..127	Slave Address

Table 43 : DDLM_Global_Control Confirmation

The CIP parameterization of this MSG request is as follows.

Parameter	Value	Remarks
Message Type	CIP Generic	
Service Type	Custom	
Service Code	64 hex	Service Code “Bridge Message”
Class	65 hex	CIP Object “CIP_MSG_BRIDGE”
Instance	1	
Attribute	0	
Source Element	GlbCtrlReq	Reference to a Tag of type DDLM_GLOBAL_CONTROL_REQUEST
Destination	GlbCtrlCnf	Reference to a Tag of type DDLM_GLOBAL_CONTROL_CONFIRM
Source Length	12	Corresponds to the size of the DDLM_GLOBAL_CONTROL_REQUEST structure

Table 44 : CIP Message Parameters for DDLM_Global_Control

5.2.3.3 DDLM_Set _Parameter

The DDLM_Set_Parameter request is used to manually send new Slave parameters. This service is only applicable for Slave devices which are configured within the SYCON.net project. This service activates the DP-Norm Primitive DDLM_Set_Parameter in order to send parameters to a specific Slave during the run time. The Master builds up the parameters that are to be sent to the Slave in such a way that it adds 7 Bytes to the USR_PRM_DATA of the DDLM_Set_Parameter service. These 7 Bytes contain standard parameters of a Slave (Ident_Number, Watchdog_Factor, Group_Ident, etc.). The parameters come from the internal configuration of the Master through the SYCON.net configuration. The user program with this service has no influence on these 7 Bytes. The MSG instruction Request /Confirmation format is as follows.

DDLM_SET_PARAMETER_REQUEST

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Reserved3	INT	0	Reserved
Command	SINT	74	Command for the DDLM_Set_Parameter service
Reserved4	SINT	0	Reserved
UsrPrm[0]	SINT	0..125	Address of the PROFIBUS Slave
UsrPrm[1..233]	SINT	n	Slave parameter data, Slave specific

Table 45 : Parameter DDLM_Set_Parameter Request

Note: The Bytearray UsrPrm[234] can be made smaller if not too many parameters have to be transferred by the service. This saves memory. The value 234 is only the maximum number of parameters in Bytes that can be transferred with the service.

DDLM_SET_PARAMETER_CONFIRM

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Answer	SINT	74	Answer DDLM_Set_Parameter
Failure	SINT	e	Error, Status (see following section)
Reserved3	INT	0	Reserved
DeviceAdr	SINT	0..125	Slave Address

Table 46 : DDLM_Set_Paramter Confirmation

Parameter	Value	Remarks
Message Type	CIP Generic	
Service Type	Custom	
Service Code	0x64	Service Code “Bridge Message”
Class	0x65	CIP Object “CIP_MSG_BRIDGE”
Instance	1	
Attribute	0	
Source Element	SetPrmReq	Reference to a Tag of type DDLM_SET_PARAMETER_REQUEST
Destination	SetPrmCnf	Reference to a Tag of type DDLM_SET_PARAMETER_CONFIRM
Source Length	9 + x (x = 0 .. 233)	9 = Constant part of the DDLM_SET_PARAMETER_REQUEST service x = No. of parameters to be written

Table 47 : CIP Messaging Parameters for DDLM_Set_Parameter

5.2.4 DPV1 Messaging

Provided in the sections below are the descriptions of the DPV1 messaging functions supported by the PROFIBUS Master module.

5.2.4.1 MSAC1_Read

The MSAC1_Read request is used by the master to perform a DPV1 read request to a slave device. The MSG instruction Request/Confirmation format is as follows.

MSAC1_READ_REQUEST

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Reserved3	INT	0	Reserved
Command	SINT	17	Command for the MSAC1_Read and MSAC1_Write service
Reserved4	SINT	0	Reserved
DeviceAdr	SINT	0.. 125	Address of the PROFIBUS Slave
DataArea	SINT	0	Reserved
DataAdr	INT	0.. 254	Slot Number
DataIdx	SINT	0.. 254	Index
DataCnt	SINT	1.. 240	Length of the data block to be read
DataType	SINT	0	Reserved
Function	SINT	1	MSAC1_Read

Table 48 : Parameter MSAC1_Read Request

MSAC1_READ_CONFIRM

Name	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Answer	SINT	17	Answer MSAC1_Read
Failure	SINT	E	Error, status (see following Section)
Reserved3	INT	0	Reserved
DeciceAdr	SINT	0.. 125	Address of the Slave
DataArea	SINT	0	Reserved
DataAdr	INT	0.. 254	Slot Number
DataIdx	SINT	0.. 254	Index
DataCnt	SINT	X = 1.. 240	Length of the received data block
DataType	SINT	0	Reserved
Function	SINT	1	MSAC1_Read
if ‚Failure‘ = CON_AD			
Data[0]	SINT		Error_Code_1
Data[1]	SINT		Error_Code_2
if ‚Failure‘ = 0			
Data[0..x-1]	SINT		Data to be received from the Slave

Table 49 : MSAC1_Read Confirmation

Parameter	Value	Remarks
Message Type	CIP Generic	
Service Type	Custom	
Service Code	64 hex	Service Code “Bridge Message”
Class	65 hex	CIP Object ”CIP_MSG_BRIDGE”
Instance	1	
Attribute	0	
Source Element	ReadReq	Reference to a Tag of type MSAC1_READ_REQUEST
Destination	ReadCnf	Reference to a Tag of type MSAC1_READ_CONFIRM
Source Length	16	Corresponds to the size of the MSAC1_READ_REQUEST structure

Table 50 : CIP Message Parameters for MSAC1_Read

5.2.4.2 MSAC1_Write

The MSAC1_Write request is used by the master to perform a DPV1 write to a slave device. The MSG instruction Request /Confirmation format is as follows.

MSAC1_WRITE_REQUEST

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Reserved3	SINT	0	Reserved
Command	SINT	17	Command for Service MSAC1_Write
Reserved4	INT	0	Reserved
DeviceAdr	SINT	0.. 125	Address of the PROFIBUS Slave
DataArea	SINT	0	Reserved
DataAdr	INT	0.. 254	Slot Number
DataIdx	SINT	0.. 254	Index
Cnt	SINT	x =1.. 240	Length of the Data block to be written
DataType	SINT	0	Reserved
Function	SINT	2	MSAC1_Write
Data[0 .. x-1]	SINT		Data to be written

Table 51 : Parameters MSAC1_Write Request

MSAC1_WRITE_CONFIRM

Parameter	Data Type	Value	Meaning
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Answer	SINT	17	Answer MSAC1_Write
Failure	SINT	E	Error, status (see following section)
Reserved3	INT	0	Reserved
DeviceAdr	SINT	0.. 125	Slave address
DataArea	SINT	0	Reserved
DataAdr	INT	0.. 254	Slot Number
DataIdx	SINT	0.. 254	Index
DataCnt	SINT	1 .. 240	Length of the data block that was written
Function	SINT	2	MSAC1_Write
If ‚Failure‘ == CON_AD			
ErrorCode1	SINT		Error_Code_1
ErrorCode1	SINT		Error_Code_2

Table 52 : MSAC1_Write Confirmation

Parameter	Value	Remarks
Message Type	CIP Generic	
Service Type	Custom	
Service Code	64 hex	Service Code “Bridge Message“
Class	65 hex	CIP Object “CIP_MSG_BRIDGE““
Instance	1	
Attribute	0	
Source Element	WriteReq	Reference to a Tag of type MSAC1_WRITE_REQUEST
Destination	WriteCnf	Reference to a Tag of type MSAC1_WRITE_CONFIRM
Source Length	16 + x (x = 1 .. 240)	16 = Constant part of the service MSAC1_WRITE_REQUEST x = Number of data to be transferred

Table 53 : CIP Message Parameters for MSAC1_Write

5.2.4.3 MSAL1M_Alarm_Res

The MSAL1M_Alarm_Res request is used by the master to perform a DPV1 Alarm acknowledgement to a slave device. This message must be sent to acknowledge this alarm when the Alarm information appears in the DPV1 Alarm Indication area. The information mapped to these area must be used in the DPV1 Alarm Response message in order to process the alarm properly. The mapping of these information shall be as follows.

MSAL1M_ALARM_RES_REQUEST

Parameter	Data Type	Value	Description
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Reserved3	INT	0	Reserved
Command	SINT	18	Command for Service MSAL1M_Alarm_Res
Reserved4	SINT	0	Reserved
DeviceAdr	SINT	0.. 125	Address of the PROFIBUS Slave from DPV1 Alarm Indication Register.
SlotNum	SINT	0.. 254	Slot Number from DPV1 Alarm Indication Register.
SeqNum	SINT	0.. 31	Sequence Number from DPV1 Alarm Indication Register.
AlarmType	SINT	1-6,32-126	Alarm Type from DPV1 Alarm Indication Register.
AlarmSpec	SINT	0..7	Alarm Spec from DPV1 Alarm Indication Register.
Reserved5	SINT	0	Reserved

Table 54 : Parameter MSAL1M_Alarm_Res Request

MSAL1M_ALARM_RES_CONFIRM

Parameter	Data Type	Value	Meaning
Reserved1	INT	0	Reserved
Reserved2	INT	0	Reserved
Answer	SINT	18	Answer MSAL1M_Alarm_Res
Failure	SINT	E	Error, status (see following section)
Reserved3	INT	0	Reserved
DeviceAdr	SINT	0.. 125	Address of the PROFIBUS Slave from DPV1 Alarm Indication Register.
SlotNum	SINT	0.. 254	Slot Number from DPV1 Alarm Indication Register.
SeqNum	SINT	0.. 31	Sequence Number from DPV1 Alarm Indication Register.
AlarmType	SINT	1-6,32-126	Alarm Type from DPV1 Alarm Indication Register.
AlarmSpec	SINT	0..7	Alarm Spec from DPV1 Alarm Indication Register.
Reserved5	SINT	0	Reserved

Table 55 : MSAL1M_Alarm_Res Confirmation

Parameter	Value	Remarks
Message Type	CIP Generic	
Service Type	Custom	
Service Code	64 hex	Service Code “Bridge Message“
Class	65 hex	CIP Object “CIP_MSG_BRIDGE““
Instance	1	
Attribute	0	
Source Element	AlarmReq	Reference to a Tag of type MSAL1M_ALARM_RES_REQUEST
Destination	AlarmCnf	Reference to a Tag of type MSAL1M_ALARM _RES_CONFIRM
Source Length	14	14 is constant for the Source Length of the MSAL1M_Alarm_Res

Table 56 : CIP Message for MSAL1M_Alarm_Res

5.2.5 Messaging Error Codes

The section includes all errors codes and conditions that can occur when using the CIP messaging commands outlined in the previous sections.

Your application should be constructed in a manner in which it catches the two possible error cases listed below:

- CIP Message instruction failed itself
- The requested command returns an error in its request confirmation

Only if both possibilities are without any error has the requested command been successful.

5.2.5.1 CIP Messaging General

Applicable are the generally known error codes for CIP Messages such as “Service Not Supported”. In this case, the parameters of the CIP Message must be checked (Service Code, Class, Instance ...). All CIP error codes that are returned by the module and their cause are described in the following table.

Note: Some CIP error codes are public and can be generated also by the Controller. Make sure the error was not generated by the controller.

CIP Status	Extended Status	Meaning	Cause	Help
02 hex	00CA hex	Resources unavailable Out of segments	System has no segments left to execute the command	
02 hex	03E8 hex	Resources unavailable Out of CIP com buffer	System has no CIP communication buffer left to execute the command	Check the number of parallel CIP messages send to the module. The module can process 5 CIP messages in parallel. Note that RSLinx can already consume 2 of this CIP com buffers if the online browser is active.
02 hex	0519 hex	Resources unavailable Out of command buffer	System has no command buffer left to execute the command	Call support
08 hex	0000 hex	Service not supported	The service code of the requested object is not supported	Check parameter of the CIP Message
14 hex	0000 hex	Attribute not supported	The attribute of the requested object is not supported	Check parameter of the CIP Message
13 hex	0000 hex	Insufficient data	Too little data was transferred with the CIP Message	Check the “Source Length” parameter in the parameter dialog of the CIP Message and check the consistency of all length parameter within the requested command.
15 hex	0000 hex	Configuration data size too large	Too much data transferred with the CIP Message	Check if the overall length of the requested command send with the CIP message and the consistency of all length parameter within the requested command is correct.
16 hex	0000 hex	Object not supported	The requested object doesn’t exist within the module.	

CIP Status	Extended Status	Meaning	Cause	Help
FE hex	0000 hex	Message Timeout	No answer message was received.	
FF hex	0514 hex	General Error Non specified error occured		Call support
FF hex	0517 hex	General Error Unknown command	The value in Req.Command is unknown	The value Req.Command must be initialized

Table 57 : CIP Message Error Codes

5.2.5.2 DDLM_Slave_Diag

Failure	Significance	Error source	Help
0	Service could be executed without an error		
17	No response from the station	DEVICE	Check network wiring, check bus address of slave or baud rate support
18	Master not into the logical token ring	Network in general	Check master DP-Address or highest-station-address of the Master. Examine bus wiring for bus short circuits
161	Remote Address in request service out of range	HOST	Check parameter in request message

Table 58 : Error Codes DDLM_Slave_Diag

5.2.5.3 DDLM_Global_Control

The DDLM_Global_Control command is sent using a multicast command. Therefore, this command is always successfully executed and no error will be placed in Cnf.Failure of the answer message.

Failure	Significance	Error source	Help
0	Service was executed without an error		

Table 59 : Error Codes DDLM_Global_Control

5.2.5.4 DDLM_Set_Parameter

Failure	Significance	Error source	Help
0 = CON_OK	Service was executed without an error		
2 = CON_RR	Resource unavailable	Slave	Slave has no buffer space left for the requested service
3 = CON_RS	Requested function of Master is not activated within the Slave	Slave	Remote SAP is not activated
17 = CON_NA	No response of the station	Slave	Check network wiring, check bus address of Slave or baud rate support
18 = CON_DS	Master not into the logical token ring	Network in general	Check master DP address or highest-station-address of other Masters. Examine bus wiring for bus short circuits.
54 = CON_AD	Negative response received, access denied	Slave	access denied

Table 60 : Error Codes DDLM_Set_Parameter

5.2.5.5 MSAC1_Read and MSAC1_Write

Failure	Significance	Error source	Help
0 = CON_OK	Service was executed without an error		
2 = CON_RR	Resource unavailable	Slave	Slave has no buffer space left for the requested service
3 = CON_RS	Requested function of master is not activated within the slave	Slave	Slave is not activated in its DPV1support
9 = CON_NR	No answer-data, although the slave has to response with data	Slave	Slave has not sent any data back
17 = CON_NA	No response of the station	Slave	Check network wiring, check bus address of Slave or baud rate support
18 = CON_DS	Master not into the logical token ring	Network in general	Check Master DP address or highest-station-address of other Masters. Examine bus wiring to bus short circuits
25 = CON_NP	No plausible reaction of remote partner	Slave	Slave does not conform to DPV1 norm
54 = CON_AD	Negative response received, access denied	Slave	Access denied to requested data. Check Error_Code_1 and Error_Code_2 in response message to get further error information
81 hex = REJ_SE	DEVICE is about to stop the DPV1-communication or the DPV1 is not in OPEN state	HOST, configuration	DPV1 communications must be configured to be activated by the DEVICE
82 hex = REJ_ABORT	DEVICE has stopped the DPV1 communication automatically	Slave	A previously addressed Slave has responded with non conform parameters
83 hex = REJ_PS	A previous service is still in process	HOST	Wait for the outstanding answer first. Parallel services are not allowed
84 hex = REJ_LE	The length indicator msg.data_cnt exceeds maximum configured size	HOST	Reduce length of message or enlarge maximum buffer size in SYCON.net or in SLAVE data set
85 hex = REJ_IV	Wrong parameter in request	HOST	Check msg.function or msg.device_adr parameter of requested message
9a hex = REJ_COM	Unknown msg.b command	HOST	Correct the requested msg.b parameter of message

Table 61 : Error Codes MSAC1_Read and MSAC1_Write

5.2.5.6 MSAL1M_Alarm_Res

Failure	Significance	Error source	Help
86 hex = REJ_INT	the alarm handler is not initialized	DEVICE	no DPV1 capable device configured within the card
87 hex = REJ_SRT	the alarm handler is currently stopped	DEVICE	no DPV1 capable slave device is in process data exchange with the DEVICE. Check if network is running
88 hex = REJ_ENA	the alarm that shall be acknowledged is not enabled in slave parameter data	HOST	enable the corresponding alarm in slave configurtion data set
89 hex = REJ_NPD	the alarm that shall be acknowledge is not pending on a MSAL1_Alarm_Res	HOST	check the parameter Alarm_Type and Seq_Nr. Both must be equal to the MSAL1_Alarm_ind parameter
9a hex = REJ_COM	unknown msg.b command	HOST	correct the requested msg.b parameter of message

Table 62 : Error Codes MSAL1M_Alarm_Res

6 Diagnostics and Troubleshooting

This section details the possible diagnostics and troubleshooting procedures for the RIF 1769-DPM Master module.

6.1 Hardware Diagnostics (LED)

The following section contains the LED diagnostic indications and there meaning for both the CPU in use and the RIF 1769-DPM module. The table below shows the possible LED indications of the CompactLogix CPU.

Indicator	Color/Status	Description
RUN	Off	no task(s) running; controller in Program mode
	Green	one or more tasks are running; controller is in the Run mode
FORCE	Off	no forces enabled
	Amber	forces enabled
	Amber Flashing	one or more input or output addresses have been forced to an On or Off state, but the forces have not been enabled.
OK	Off	no power applied
	Green	controller OK
	Red flashing	recoverable controller fault
	Red	Non-recoverable controller fault: Cycle power. The OK LED should change to flashing red. If LED remains solid red, replace the controller.
I/O	Off	no activity; no I/O or communications configured
	Green	communicating to all devices
	Green flashing	one or more devices not responding
	Red flashing	not communicating to any devices controller faulted

Table 63 : CompactLogix CPU LEDs

6.1.1 RIF 1769 LEDs

The LEDs as shown on the front panel will be used to indicate status information of the RIF 1769-DPM Master module. Each LED has a specific function during Run, configuration download, and error indications. The table below shows the reaction of each during these states for Master and Slave.

LED	Color	State	Description
SYS			
	Yellow	Flashing cyclic at 1Hz	Device is in boot loader mode and is waiting for firmware download.
	Yellow	Flashing cyclic at 5Hz	Firmware download is in progress.
	Yellow	Flashing irregular (*)	Hardware or runtime error detected.
	Green	Static On	Communication is running. The device has established at least one configured fieldbus connection.
	Green	Flashing cyclic at 5Hz	No error in configuration found, communication is stopped.
	Green	Flashing irregular (*)	Power Up: Configuration missing or faulty, device needs commissioning. Runtime: Host Watchdog timeout
	Off	Off	Device has no power supply or hardware defect.
COM			
	Green	On	Device is holding the PROFIBUS token and is able to transmit telegrams.
	Green	Flashing acyclic (**)	Device is sharing the PROFIBUS token with other master devices in the PROFIBUS network.
	Red	On	Device has found a communication problem to at least one PROFIBUS-DP slave device or has detected a short circuit.
	Off	Off	Device is not configured or has not received the Token permission on the PROFIBUS network.
(*) 3 times fast at 5 Hz, 8 times between 0,5Hz and 1Hz (**) between 0,5Hz and 100Hz			

Table 64 : LED Diagnostic Indications

6.2 SYCON.net Diagnostics

SYCON.net provides Master or individual Slave diagnostics. The sections below will describe each.

6.2.1 Master Diagnostics

SYCON.net allows the user to monitor the operating condition of the PROFIBUS Master via the serial port connection. To use this diagnostic functionality open your project in SYCON.net. In the SYCON.net project screen, right click on the Master you have added in either the netDevice bus or the netProject screen. From the pop up menu, select **Connect**. The text above the Master should be highlighted in Green. Double click on the Master in either the netDevice bus or the netProject screen. The following diagnostic dialog should appear.

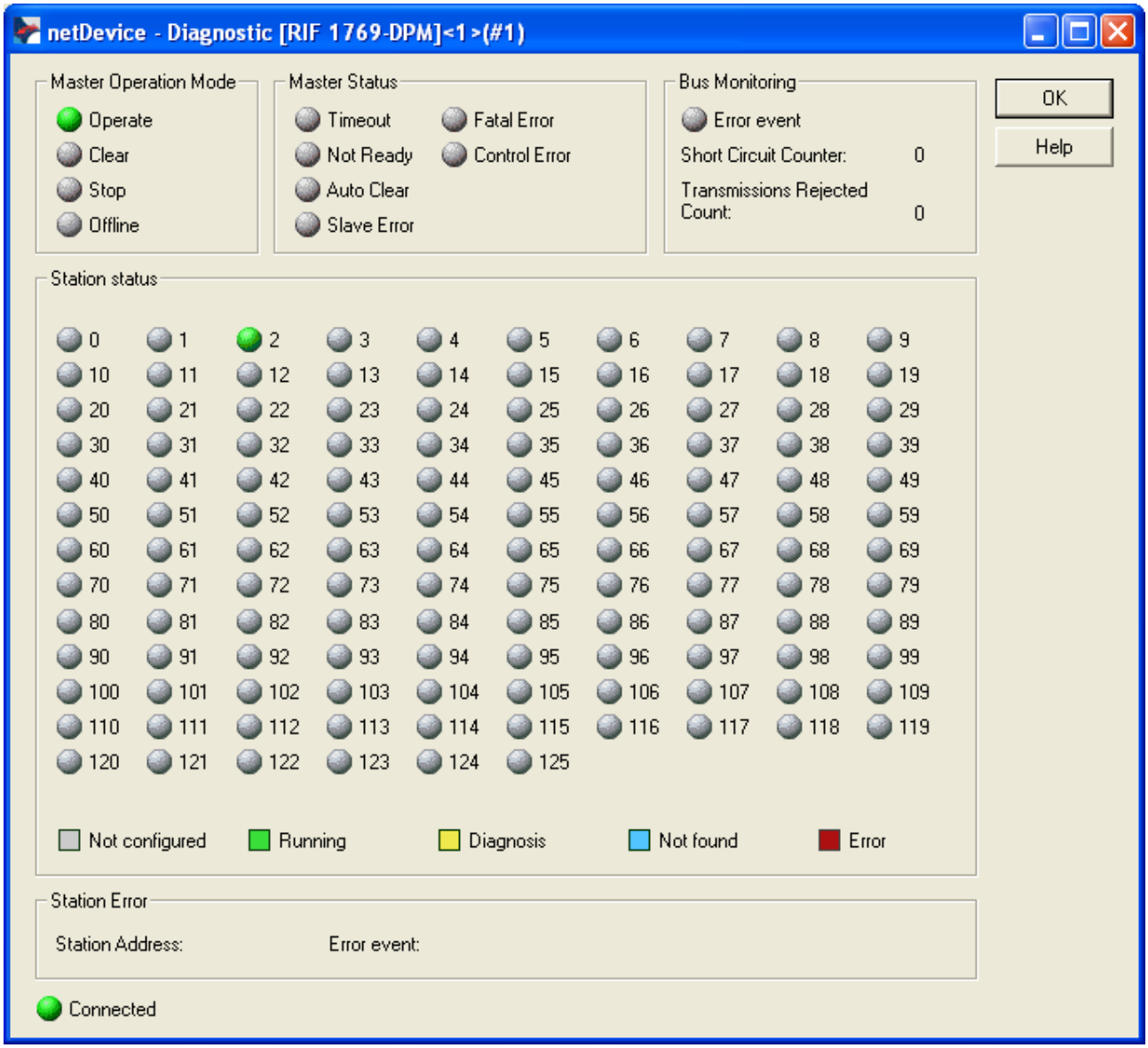

Figure 25 : Master Diagnostic Dialog

Using the Master Diagnostic Dialog the user can determine the current running state of the Master. For further definition of each item in this Diagnostic Dialog, refer to the Help within SYCON.net and the Operating Instruction Manual for Hilscher DTM for PROFIBUS Master Devices.

6.2.2 Slave Diagnostics

SYCON.net provides the user a way to monitor the operating condition of each individual PROFIBUS Slave via the serial port connection. To use this diagnostic functionality open your project in SYCON.net. In the SYCON.net project screen, right click on the Slave you have added in either the netDevice bus or the netProject screen. From the pop up menu, select **Connect**. The text above the Slave should be highlighted in Green. Double click on the Slave in either the netDevice bus or the netProject screen. The following diagnostic dialog should appear.

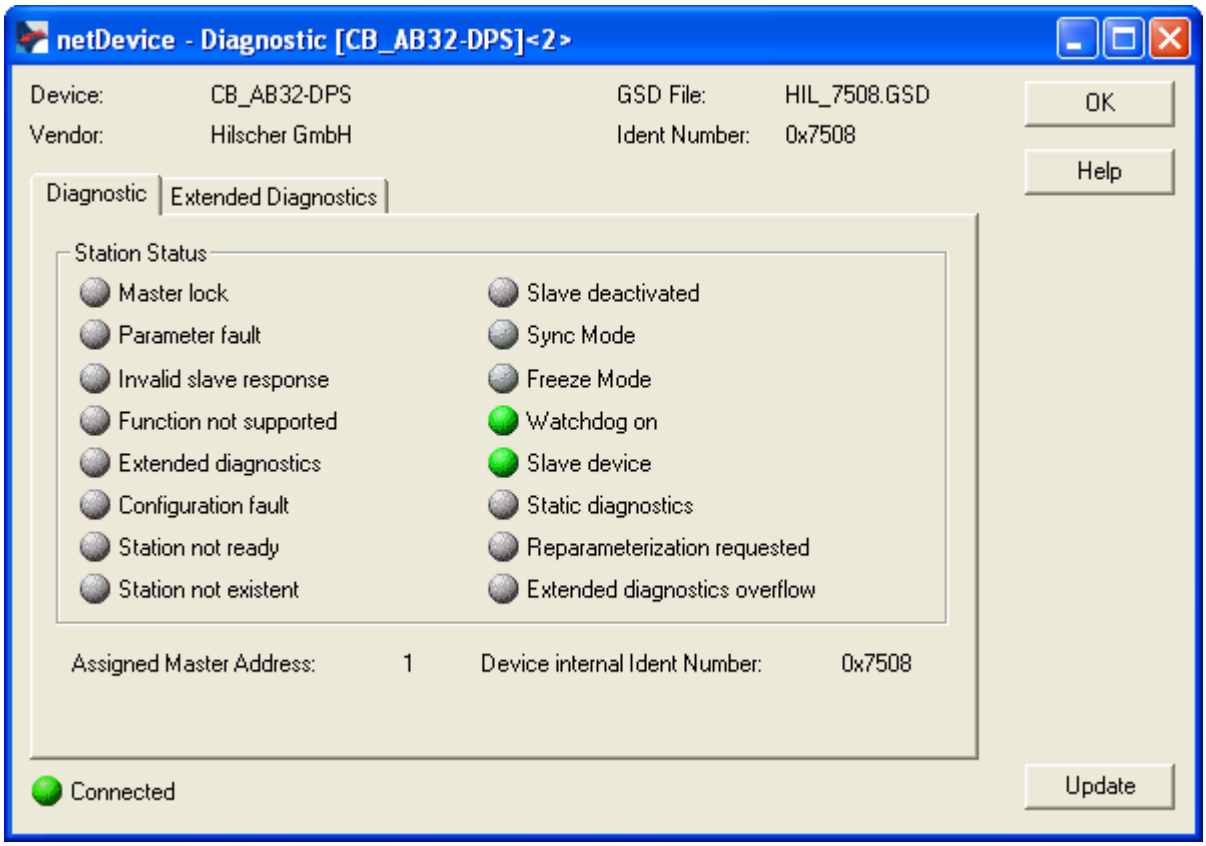

Figure 26 : Slave Diagnostic Dialog

Using the Slave Diagnostic Dialog the user can determine the current running state of the Slave as well as extended diagnostics data. For further definition of each item in this Diagnostic Dialog, refer to the Help within SYCON.net and the Operating Instruction Manual for Generic DTM for PROFIBUS-DP Slave Devices

6.3 Troubleshooting

Troubleshooting of the system is done by examining the LED's on the front panel of the CPU and the LED's on the front of the module. The following sections contain some troubleshooting ideas.

6.3.1 I/O LED Controller

Communication between the module and controller is displayed via the I/O LED of the Controller. The faultless communication state is reached, if the I/O LED of the CompactLogix Controller is static Green. If this LED is flashing or off, no communication between controller and card takes place.

6.3.2 SYS and COM Status LEDs

This RIF1769-DPM module has two bicolor status LEDs. They inform the user about the communication state of the module. 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 or red. The meaning of the LEDs is described in the booklet of the System Software CD. If the SYS-LED is solid green and the COM-LED static yellow, the card is in cyclic data exchange with the connected Slaves and the communication is running with out fault.

6.3.3 Error Sources and Reasons

This chapter describes typical problems, error sources and questions that come up while commissioning the PROFIBUS-DP master module RIF 1769-DPM. The following table summarizes the typical error sources and gives a hint of possible reasons for the problem.

Behaviour	Significance	Typical Reason	Help
PLC's I/O LED is Green flashing	No communication with the RIF module (or other modules)	- Modules slot number in RSLogix program does not match with the physical slot of the module - Configured Input / Output size is wrong	- Check modules slot number in RSLogix project - Compare configured Input / Output size with required values
RIF 1769-DPM COM LED is off SYS LED Flashing irregular green	Configuration missing or faulty	No configuration stored	Download a Configuration to the card with SYCON.net
RIF 1769-DPM COM LED is static green and SYS LED flashing cyclic fast green	Application is not ready	- PLC is not in RUN Mode. - PLC application has set the NRDY bit. - PLC has no I/O communication with the module	- Bring PLC into RUN Mode. - Check that the PLC application has deleted the NRDY bit. - Check PLC's I/O LED
RIF 1769-DPM COM LED is static red and SYS LED static green	At least one slave is not in data exchange	- Master configuration does not match with physical bus configuration - Configured slave is not connected with PROFIBUS or has a problem	- Check PROFIBUS configuration, slave addresses etc. - Use SYCON.net diagnostic to find the wrong slave
RIF 1769-DPM COM LED is static red and	No communication to any slave	- PROFIBUS cable not connected	- PROFIBUS wiring

Behaviour	Significance	Typical Reason	Help
SYS LED flashing cyclic fast green		- No slaves connected - PLC is not in RUN mode	- Check is slaves are connected - Check is PROFIBUS configuration matches with physical configuration - Bring PLC is into in RUN mode
Slave input data can not be found in RSLogix program	Input array mismatch	Configured input size in RSLogix to small Configured input address table in SYCON.net does not match with PLC program	Check if the configured input size in RSLogix covers the mandatory size of 88 byte status data plus the in SYCON.net configured PROFIBUS input array Check if configured Input address table in SYCON.net matches with PLC program
Outputs are not transferred to slave although PROFIBUS is running	Output array mismatch	Configured output size in RSLogix to small Configured output address table in SYCON.net does not match with PLC program	Check if the configured output size in RSLogix covers the mandatory size of 16 byte status data plus the in SYCON.net configured PROFIBUS output array Check if configured Output address table in SYCON.net matches with PLC program
The serial device assignment dialog in SYCON.net shows error – 20 to the wished serial COM port	COM port not available	COM port is physically not available or used by another application	Check if the wished COM port is available and not used by another application
The serial device assignment dialog in SYCON.net shows error – 51 to the wished serial COM port	Module does not answer to SYCON.net request	Diagnostic cable not connected	Check if the mode if the mode and the PC are wired correctly with the diagnostic cable
SYCON.net configuration download results in error 100	Download not allowed	Configuration download is not allowed while PLC is in RUN mode	Bring PLC into STOP mode

Table 65 : Troubleshooting

7 RSLogix Example Program

Provided on the installation CD are two example Ladder Logic programs RIF_1769_DPM_L32E.acd and RIF_1769_DPM_Messaging_L32E.acd. These two examples should be used as templates for starting your project. An explanation of each project is in the following sections. If you are using another type of CompactLogix Controller, change the ControllerType in RSLogix and then store it to your individual project. If you setup up a new controller project you can use the Copy and Paste functionality of RSLogix to transfer the user defined data types or ladder logic needed with the module RIF 1769-DPM from the template projects to your own application

Sample Project	Controller Type	RSL5K Version	Description
RIF_1769_DPM_L32E.acd	1769-L32E	V13	Basic I/O example
RIF_1769_DPM_Messaging_L32E.acd	1769-L32E	V13	Basic messaging example

Table 66 : Sample Projects

7.1 I/O Example

This ladder logic program is a basic example for the setup of the PROFIBUS-DP master communications module "RIF 1769-DPM" in RSLogix5000. This example can be used to start a project when using a CPU 1769-L32E. Basic PROFIBUS I/O data exchange, Diagnostic requests, and Global Control are shown. Details on the Subroutines created and the User Defined Data Types are as follows.

- **MainRoutine** – The MainRoutine calls all of the following routines based on conditions like doing a diagnostic request or a Global Control command. This routine also contains a simple I/O transfer function block.
- **IO_Global_Control** – The IO_Global_Control routine serves as an example of how the user can execute a Global Control to issue a Sync or Freeze to a slave module group. This routine is executed based on the state of the DoIOGlobCtrl tag. The command, Group, and Slave address data must be filled in before using the command.
- **IO_Slave_Diag** – The IO_Slave_Diag routine shows an example of send a Slave diagnostics request. The routine is call automatically by the logic found in the MainRoutine or can be executed manually by toggling the DoIOSlaveDiag bit.
- **SR_Copy_Input** – The SR_Copy_Input routine on every scan updates the DpmInputArray structure with the Input Data of the module.
- **SR_Copy_Output** – The SR_Copy_Output routine on every scan updates the DpmOutputArray structure with the Output Data of the module.

Numerous user defined data types have been created to make it easier to address different elements of the Input and Output array of the module. The two main structures are DpmInputArray and DpmOutputArray there definitions and the structures included in each are shown in the following tables.

7.2 Messaging Example

This ladder logic program is a CIP messaging example for the setup of the PROFIBUS-DP master communications module "RIF 1769-DPM" in RSLogix5000. This example can be used to start a project when using a CPU 1769-L32, which supports CIP messaging. Basic PROFIBUS I/O data exchange and all messaging function examples are shown. Details on the Subroutines created and the User Defined Data Types are as follows.

- **MainRoutine** – The MainRoutine calls all of the following routines based on conditions like doing a diagnostic request or a Global Control command. This routine also contains a simple I/O transfer function block.
- **AlarmHandler** – This routine shows an example on how to handle unsolicited DPV1 alarms from a Slave. The user must modify this routine to the DPV1 slave or slaves used in the application.
- **Init_AlarmResMsg** – The AlarmHandler routine is used to trigger this routine. When the alarm event has occurred, this routine will format and send the response to a DPV1 Alarm from a Slave.
- **Init_GlobalControlMsg** – This routine serves as an example of how the user can execute a Global Control command used to issue a Sync or Freeze to a slave module group. This routine will execute based on the state of the DoDDLMGlbCtrl tag. The command, Group, and Slave address data must be filled in before using the command.
- **Init_GlobalVariables** – Initializes the Slave address used by several other routines. Make changes as need to support the Slave address for your application.
- **Init_ReadReqMsg** – This routine is used to form the DPV1 read request CIP message. The routine is triggered by using the DoDPV1ReadReq tag. Once triggered a MSAC1_Read_req command is sent to the Slave requesting a block of data. The return data from this command shall appear MSAC1_READ_CONFIRM user defined tag. The user should make changes to this routine as need for their application.
- **Init_SetParameterMsg** – The Init_SetParameterMsg routine shows a simple example of using CIP messaging to send user parameter data to a Slave. To trigger this routine the DoDDLMSetPrm tag is used. Changes is this routine should be made to fit the users particular Slave. See the Slaves user manual for the user settable values.
- **Init_SlaveDiagMsg** – This routine shows an example of sending a Slave diagnostics request using CIP messaging. The routine is call automatically by the logic found in the MainRoutine or can be executed manually by toggling the DoDDLMSlaveDiag bit. The response data from this message is contained in DDLM_SLAVE_DIAGNOSTIC_CONFRIM user defined data type.

- **Init_WriteReqMsg** – This routine is used to form the DPV1 write request CIP message. The routine is triggered by using the DoDPV1WriteReq tag. Once triggered a MSAC1_Write_req command is sent to the Slave containing a block of data. The return data from this command shall appear MSAC1_WRITE_CONFIRM user defined tag. The user should make changes to this routine as need for their application.
- **SR_Copy_Input** – The SR_Copy_Input routine on every scan updates the DpmInputArray structure with the Input Data of the module.
- **SR_Copy_Output** – The SR_Copy_Output routine on every scan updates the DpmOutputArray structure with the Output Data of the module.
- **SR_Main_Init** – Initializes several variables used by different routines.

Numerous user defined data types have been created to make it easier to address different elements of the Input and Output array of the module. The two main structures are DpmInputArray and DpmOutputArray there definitions and the structures included in each are shown in the following tables.

8 A-Specifications

8.1 User Defined Data Types

Contained in this appendix are all the user defined data types created and used in the example programs.

Name	Data Type	Description
DevStaReg	DPM_DEV_STATUS_REGISTER	
FwRev	DPM_FW_REVISION	
GlobStateField	DPM_GLOBAL_STATE_FIELD	
SlaveDiag	DPM_SLAVE_DIAG_DATA	
AlarmInd	DPM_DPV1_ALARM_INDICATION	
InputData	INT[20]	

Table 67 : Input - DPM_INPUT_ARRAY

Name	Data Type	Description
Reserved0	BOOL	
Reserved1	BOOL	
Reserved2	BOOL	
Reserved3	BOOL	
Reserved4	BOOL	
Com	BOOL	Communication
Run	BOOL	Running
Rdy	BOOL	Ready
HsAck0	BOOL	Slave Diag Acknowledge
HsAck1	BOOL	Global Control Acknowledge
HsAck2	BOOL	
HsAck3	BOOL	
HsAck4	BOOL	
HsAck5	BOOL	
HsAck6	BOOL	
HsAck7	BOOL	
Reserved5	SINT	
Reserved6	SINT	

Table 68 : Input - DPM_DEV_STATUS_REGISTER

Name	Data Type	Description
FwMajor	SINT	Firmware Major Revision
FwMinor	SINT	Firmware Minor Revision
Reserved	INT	Reserved

Table 69 : Input - DPM_FW_REVISION

Name	Data Type	Description
Ctrl	BOOL	Control error
Aclr	BOOL	Auto clear error
Nexc	BOOL	Non exchange error
Fat	BOOL	Fatal error
Eve	BOOL	Event error
NRdy	BOOL	Host not ready notification
Tout	BOOL	Timeout
Reserved1	BOOL	Reserved
DPM_State	SINT	Master main state
Err_rem_adr	SINT	Faulty remote address
Err_event	SINT	Error Number
Bus_err_cnt	INT	Heavy bus error counter
Time_out_cnt	INT	Number of rejected PROFIBUS Telegr.
Reserved	SINT[8]	Reserved
Sl_cfg	BOOL[128]	Slave Config
Sl_state	BOOL[128]	Slave State
Sl_diag	BOOL[128]	Slave Diag

Table 70 : Input - DPM_GLOBAL_STATE_FIELD

Name	Data Type	Description
SlaveAddress	SINT	Slave address
Sta1_StationNotExist	BOOL	No response
Sta1_StationNotReady	BOOL	Station not ready
Sta1_CfgFault	BOOL	Configuration faulty
Sta1_ExtDiag	BOOL	Extended diagnostic
Sta1_NotSupp	BOOL	Sync,Freeze not supported
Sta1_InvalidResponse	BOOL	Response faulty
Sta1_PrmFault	BOOL	Parameters faulty
Sta1_MasterLock	BOOL	Locked by a master
Sta2_PrmReq	BOOL	Request new parameter
Sta2_StatDiag	BOOL	Static diagnostic
Sta2_Slave	BOOL	Set to 1 by a slave
Sta2_Watchdog	BOOL	Watchdog ON/OFF
Sta2_FreezeMode	BOOL	Freeze mode active
Sta2_SyncMode	BOOL	Sync mode active
Sta2_Reserved	BOOL	Reserved
Sta2_Deactivated	BOOL	Slave deactivated
Sta3_Reserved0	BOOL	Reserved
Sta3_Reserved1	BOOL	Reserved
Sta3_Reserved2	BOOL	Reserved
Sta3_Reserved3	BOOL	Reserved
Sta3_Reserved4	BOOL	Reserved
Sta3_Reserved5	BOOL	Reserved
Sta3_Reserved6	BOOL	Reserved
Sta3_ExtDiagOverflow	BOOL	Extended diagnostic overflow
MasterAddress	SINT	Corresponding master address

Name	Data Type	Description
IdentNumber	INT	PROFIBUS Ident number

Table 71 : Input - DPM_SLAVE_DIAG_DATA

Name	Data Type	Description
AlarmIndication	BOOL	Indicates of an alarm
Reserved1	BOOL	
Reserved2	BOOL	
Reserved3	BOOL	
Reserved4	BOOL	
Reserved5	BOOL	
Reserved6	BOOL	
AlarmOverrun	BOOL	Overflow of the modules internal alarm buffer
RemoteAddress	SINT	Address of Slave with Alarm
Slot	SINT	Slot Number
Sequence	SINT	Sequence Number
AlarmType	SINT	Alarm Type
AlarmSpec	SINT	Alarm Specification
Reserved7	SINT	
Reserved8	SINT	

Table 72 : Input - DPM_DPV1_ALARM_INDICATION

Name	Data Type	Description
DevCmdReg	DPM_DEV_COMMAND_REGISTER	
DiagReqAdr	DPM_SLAVE_DIAG_COMMAND	
GlobCtrl	DPM_GLOBAL_CONTROL_COMMAND	
Reserved	SINT[6]	
OutputData	INT[56]	

Table 73 : Output - DPM_OUTPUT_ARRAY

Name	Data Type	Description
Reserved0	BOOL	Reserved
Reserved1	BOOL	Reserved
Reserved2	BOOL	Reserved
Reserved3	BOOL	Reserved
Reserved4	BOOL	Reserved
NRdy	BOOL	Application Not Ready
Init	BOOL	Init (Warm boot)
Reset	BOOL	Reset (Cold boot)
HsReq0	BOOL	Slave Diag Request
HsReq1	BOOL	Global Control Request
HsReq2	BOOL	Reserved
HsReq3	BOOL	Reserved
HsReq4	BOOL	Reserved
HsReq5	BOOL	Reserved
HsReq6	BOOL	Reserved
HsReq7	BOOL	Reserved
Reserved5	SINT	Reserved
Reserved6	SINT	Reserved

Table 74 : Output - DPM_DEV_COMMAND_REGISTER

Name	Data Type	Description
SlaveAddress	SINT	Address of Slave
Reserved1	SINT	

Table 75 : Output - DPM_SLAVE_DIAG_COMMAND

Name	Data Type	Description
SlaveAddress	SINT	Slave Address
ControlCommand	SINT	Control Command to be send
GroupSelect	SINT	Group Select
Reserved3	SINT	

Table 76 : Output-DPM_GLOBAL_CONTROL_COMMAND

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
DeviceAdr	SINT	Device Address
ConrolCommand	SINT	Control Command
GroupSelect	SINT	Group Select

Table 77 : DDLM_GLOBAL_CONTROL_REQUEST

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Answer	SINT	
Failure	SINT	
Reserved3	INT	
DeviceAdr	SINT	

Table 78 : DDLM_GLOBAL_CONTROL_CONFIRM

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
UsrPrm	SINT[234]	

Table 79 : DDLM_SET_PARAMETER_REQUEST

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Answer	SINT	
Failure	SINT	
Reserved3	INT	
DeviceAdr	SINT	

Table 80 : DDLM_SET_PARAMETER_CONFIRM

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
DeviceAdr	SINT	
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	

Table 81 : DDLM_SLAVE_DIAGNOSTIC_REQUEST

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Answer	SINT	
Failure	SINT	
Reserved3	INT	
DeviceAdr	SINT	
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	
StationState_1	SINT	
StationState_2	SINT	
StationState_3	SINT	
MasterAddress	SINT	
IdentNumber	INT	
Reserved4	INT	
ExtDiag	SINT[100]	

Table 82 : DDLM_SLAVE_DIAGNOSTIC_CONFIRM

Name	Data Type	Description
AlarmIndication	BOOL	Indicates an alarm
Reserved1	BOOL	
Reserved2	BOOL	
Reserved3	BOOL	
Reserved4	BOOL	
Reserved5	BOOL	
Reserved6	BOOL	
AlarmOverrun	BOOL	Overflow of the modules internal alarm buffer
AlarmCnt	SINT	Alarm indication counter
RemoteAddress	SINT	Address of Slave with Alarm
Slot	SINT	Slot Number
Sequence	SINT	Sequence Number
AlarmType	SINT	Alarm Type
AlarmSpec	SINT	Alarm Specification
Reserved7	SINT	

Table 83 : DPM_DPV1_ALARM_INDICATION

Name	Data Type	Description
StaAlarmInd	BOOL	Indicated alarm is pending
StaAlarmReserved1	BOOL	
StaAlarmReserved2	BOOL	
StaAlarmReserved3	BOOL	
StaAlarmReserved4	BOOL	
StaAlarmReserved5	BOOL	
StaAlarmReserved6	BOOL	
StaAlarmOvrRun	BOOL	Alarm overrun
AlarmCnt	SINT	AlarmCounter
SlaveAdr	SINT	Slave address
SlotNum	SINT	Slot number
SeqNum	SINT	Sequence number
AlarmType	SINT	Alarm type
AlarmSpec	SINT	Alarm specifier
Reserved	SINT	

Table 84 : DPV1_ALARM_INDICATION

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
DeviceAdr	SINT	
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	

Table 85 : MSAC1_READ_REQUEST

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Answer	SINT	
Failure	SINT	
Reserved3	INT	
DeciceAdr	SINT	
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	
Data	SINT[240]	

Table 86 : MSAC1_READ_CONFIRM

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
DeviceAdr	SINT	
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	
data	SINT[240]	

Table 87 : MSAC1_WRITE_REQUEST

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Answer	SINT	
Failure	SINT	
Reserved3	INT	
DeciceAdr	SINT	
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	
ErrorCode1	SINT	
ErrorCode2	SINT	

Table 88 : MSAC1_WRITE_CONFIRM

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
SlaveAdr	SINT	
SlotNum	SINT	
SeqNum	SINT	
AlarmType	SINT	
AlarmSpec	SINT	
Reserved5	SINT	

Table 89 : MSAL1M_ALARM_RESPONSE

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Answer	SINT	
Failure	SINT	
Reserved3	INT	
SlaveAdr	SINT	
SlotNum	SINT	
SeqNum	SINT	
AlarmType	SINT	
AlarmSpec	SINT	
Reserved4	SINT	

Table 90 : MSAL1M_ALARM_CONFIRM

8.2 Firmware Upgrade

If it's nesessary to download a new firmware e.g. because of upgrading the modules functionality then do following steps:

The modules firmware is upgraded using SYCON.net. This section contains the steps required to upgrade the modules firmware.

- Make sure that SYON.net and the module are connected via the serial diagnostic cable.
- Create a project in SYCON.net with a RIF 1769-DPM module.
- Establish the serial connection between SYCON.net and the module by the device assignment procedure like it is described in step 1 of the configuration download process in chapter 4.2.5 Project Download.
- In the SYCON.net project screen, click on the Master you have added in either the netDevice bus or the netProject screen, the Master should be highlighted with a Blue box. If the Master is highlighted in Green use the Menu item **Device>Disconnect** or Right Click on the Master in either netDevice bus or the netProject screen selecting **Disconnect** to switch off the diagnostic mode.
- Double click on the RIF 1769-DPM Master in the netDevice or the netProject window of your SYCON.net project. The following dialog appears. Select the **Firmware Download Tab.**

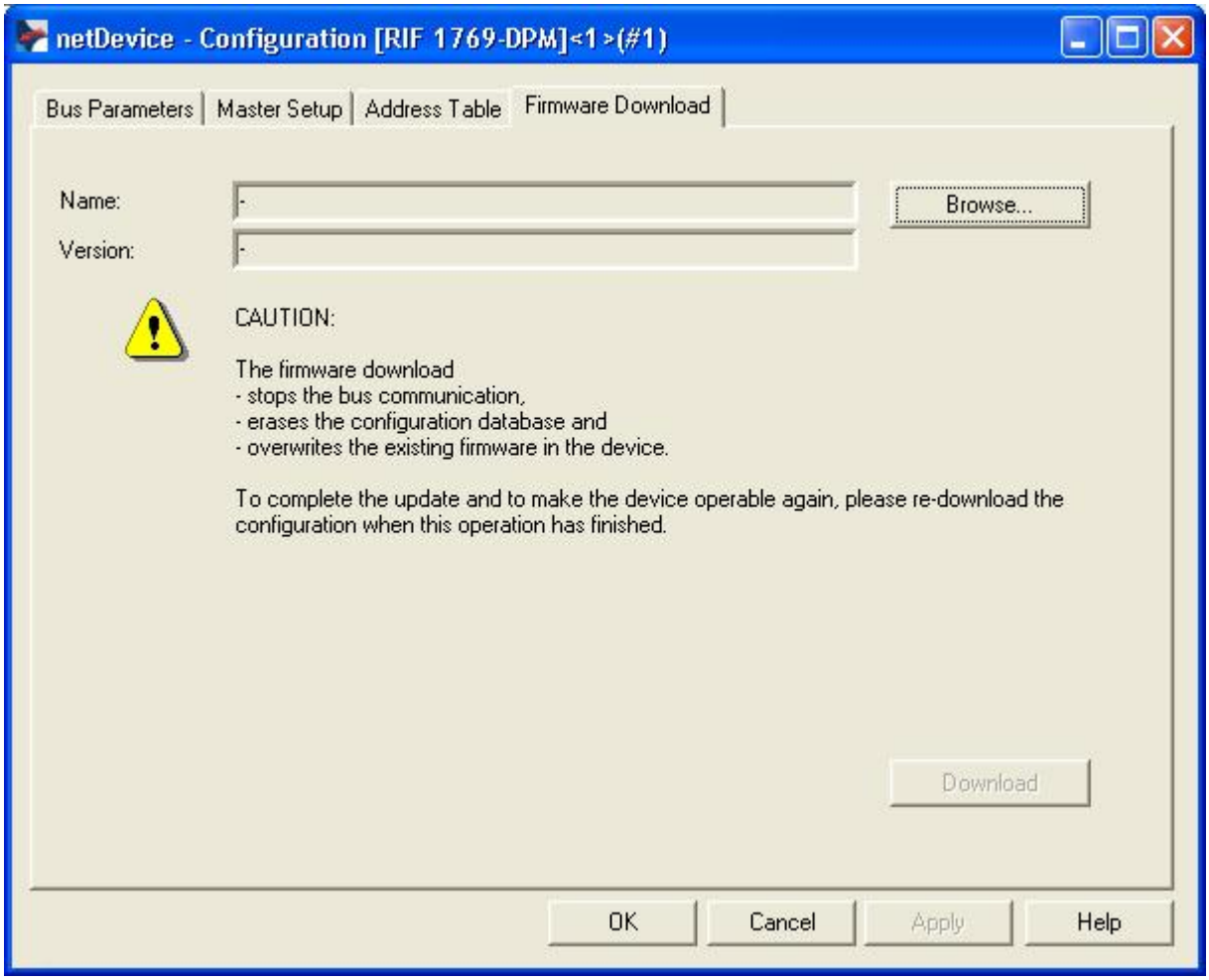

Figure 27 : Firmware Dowload Dialog

Use the **Browse...** button to select the firmware file 1769DPM.E35 which will be the latest firmware you wish to upgrade your module. If you have selected the firmware file, the dialog shows following information depending on the firmware version you wish to download.

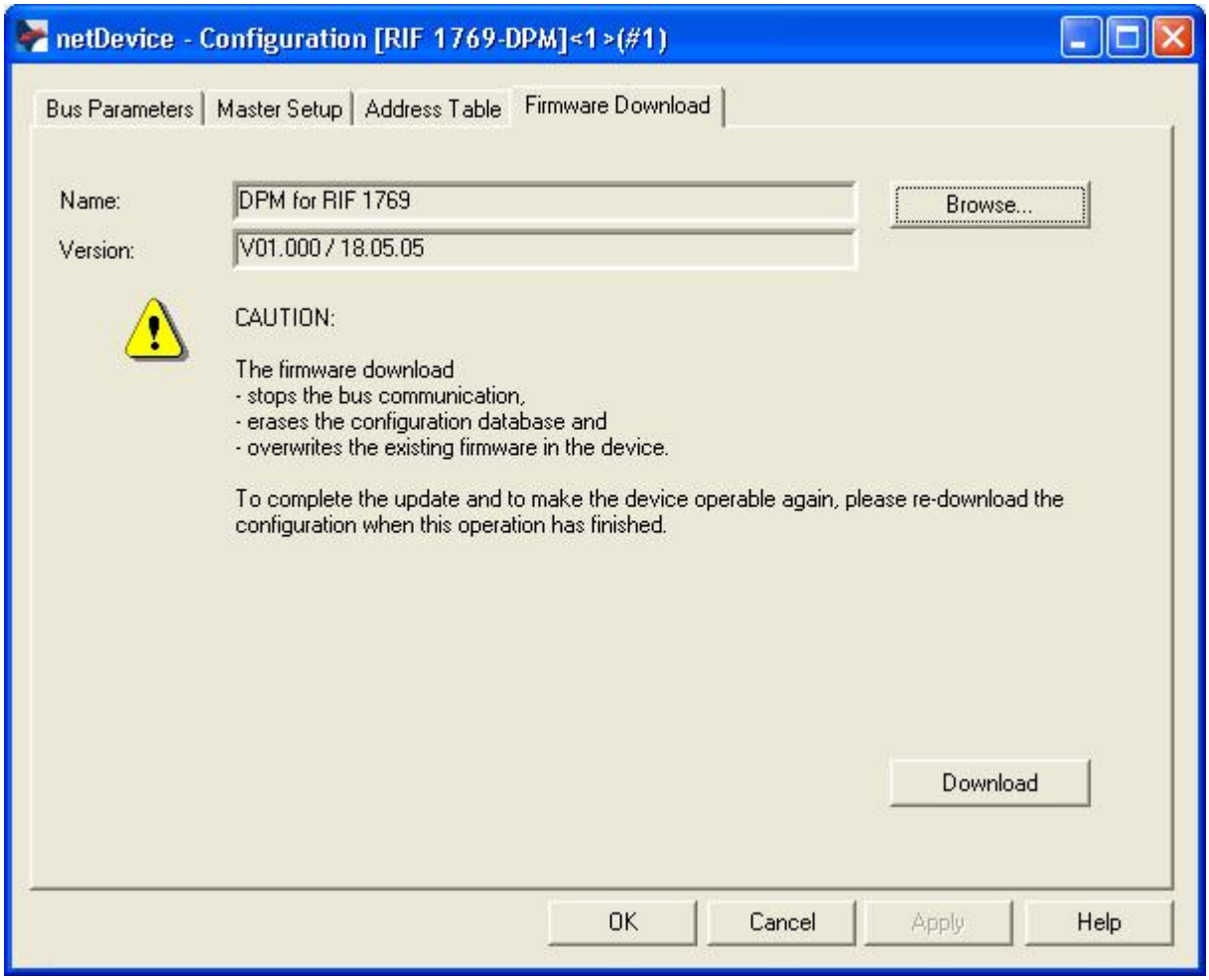

Figure 28 : Firmware Download Selected Firmware

Start the download process by pressing the **Download** button Please wait until the process is complete. Exit the program and reboot the module to ensure proper operation. After the new firmware is downloaded, you must send the Master configuration again.

8.3 Product Specifications

For all technical data and electrical/environmental specifications of the module RIF 1769-DPM refer to the manual **RIF1769 Booklet.pdf** which can be found also on the CD delivered with the RIF 1769-DPM module.

9 Lists

9.1 List of Figures

9.2 List of Tables