ProSoft TECHNOLOGY

Where Automation Connects.

PS69-DPM

CompactLogix or MicroLogix Platform

PROFIBUS DPV1 Master

October 1, 2014

USER MANUAL

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

ProSoft Technology
5201 Truxtun Ave., 3rd Floor
Bakersfield, CA 93309
+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

PS69-DPM User Manual

October 1, 2014

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed DVD and are available at no charge from our web site: http://www.prosoft-technology.com

Throughout this manual, you will see references to other product names such as:

- RIF 1769-DPM
- SYCON.net

These product names (RIF 1769, SYCON.net) are legacy versions, and are mentioned for backward compatibility with existing implementations. These products are now supported and maintained by ProSoft Technology.

The ProSoft and legacy versions of these products may not be interchangeable.

Contents

1 Start Here

In This Chapter

To get the most benefit from this User Manual, you should have the following skills:

- **Rockwell Automation® RSLogix™ software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows®:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect PROFIBUS DPV1 and CompactLogix or MicroLogix devices to a power source and to the PS69-DPM module's application port(s)

1.1 General Information

The communication module PS69-DPM is a slot extension module for a CompactLogix or MicroLogix 1500 Controller which enables controllers to communicate with a PROFIBUS network. The PS69-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 System Configuration tool PROSOFT.fdt (SYCON.net). The data exchange between controller and module is done via the I/O process data image using CompactLogix / MicroLogix backplane technology.

1.2 Software Requirements

Follows are the software requirements for using the PS69-DPM module within a CompactLogix or MicroLogix 1500 system. You must have the following software installed on your computer unless otherwise noted:

CompactLogix System

- RSLogix 5000, V13.00 or higher

MicroLogix 1500 System

- RSLogix 500, V6.30 or higher

Configuration Tool

- PROSOFT.fdt (SYCON.net) for Rockwell Interfaces V1.023 or higher

Requirements for PROSOFT.fdt (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

1.3 Hardware Requirements

The following minimum hardware is required to use the PS69-DPM PROFIBUS module.

CompactLogix System

- 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 PROSOFT.fdt (SYCON.net) to Diagnostic port connection ProSoft part number CAB-SRV-MD8

MicroLogix 1500 System

- Personal Computer
- 1764: MicroLogix 1500 Programmable Controller
- 1769: Right handed Termination End Cap
- Serial Cable for interface to the 1764-Programmable Controller.
- Serial Cable for PROSOFT.fdt (SYCON.net) to Diagnostic port connection ProSoft part number CAB-SRV-MD8

1.4 Reference Systems

The firmware of the communication module PS69-DPM was developed and tested with following CompactLogix Controller types and firmware revisions.

CompactLogix System

PS69-DPM	CompactLogix 1769-L20	CompactLogix 1769-L32E
Firmware V10.2	Firmware V13.18	Firmware V13.28

MicroLogix 1500 System

PS69-DPM	MicroLogix 1500 (Processor 1764-LRP/A Rev2.0)
Firmware V10.2	Firmware: OS 1510; Series C ; Revision 9.0

1.5 Programmable Controller Functionality

PROFIBUS-DP supports acyclic services through messages. These PROFIBUS-DP services are supported by the RSLogix 5000 programming tool using CIP messages. Not all of the 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 Diagnostic request are also executable in addition to the CIP method by using the I/O area. Follows is a matrix of Programmable Controllers and the functionality that they support.

CompactLogix System

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

MicroLogix 1500 System

Processor/ Features	1764-LRP	1764-LSP
I/O	yes	yes
CIP Messaging	no	no

yes = functionality supported
no = functionality not supported

1.6 Package Contents

The following components are included with your PS69-DPM module, and are all required for installation and configuration.

Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	PS69-DPM Module	PS69-DPM	PROFIBUS DPV1 Master
1	Cable	CABLE-SRV-MD8	For Connection to Module's Port
1	ProSoft Solutions DVD	DVD-001	Contains sample programs, utilities and documentation for the PS69-DPM module.

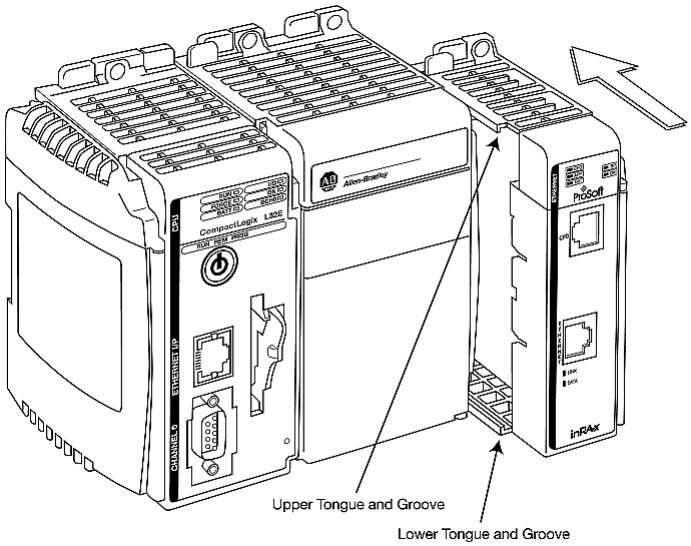
If any of these components are missing, please contact ProSoft Technology Support for replacement parts.

1.7 Installing the Module

Before you attempt to install the module, make sure that the bus lever of the adjacent module is in the unlocked (fully right) position.

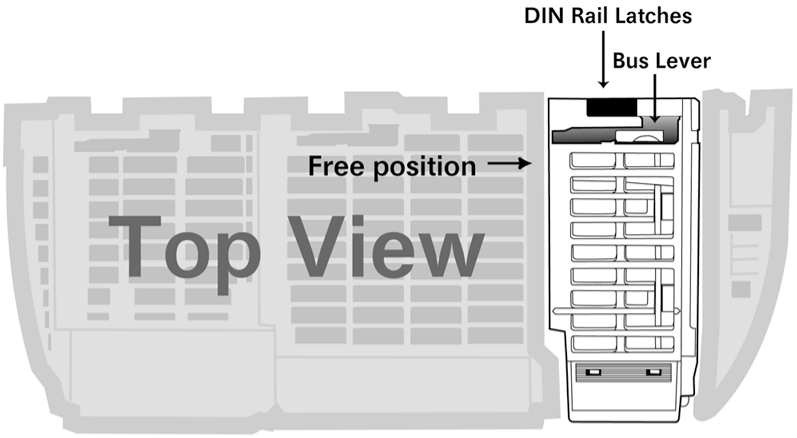
Warning: This module is not hot-swappable! Always remove power from the rack before inserting or removing this module, or damage may result to the module, the processor, or other connected devices.

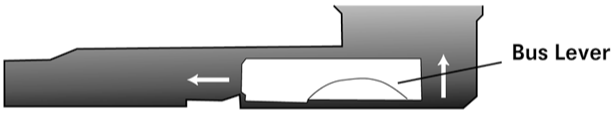
1 Align the module using the upper and lower tongue-and-groove slots with the adjacent module and slide forward in the direction of the arrow.



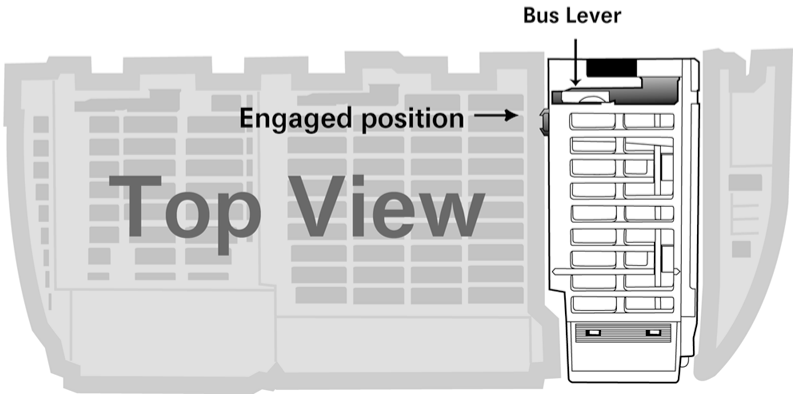
2 Move the module back along the tongue-and-groove slots until the bus connectors on the PS69 module and the adjacent module line up with each other.

3 Push the module's bus lever back slightly to clear the positioning tab and move it firmly to the left until it clicks. Ensure that it is locked firmly in place.



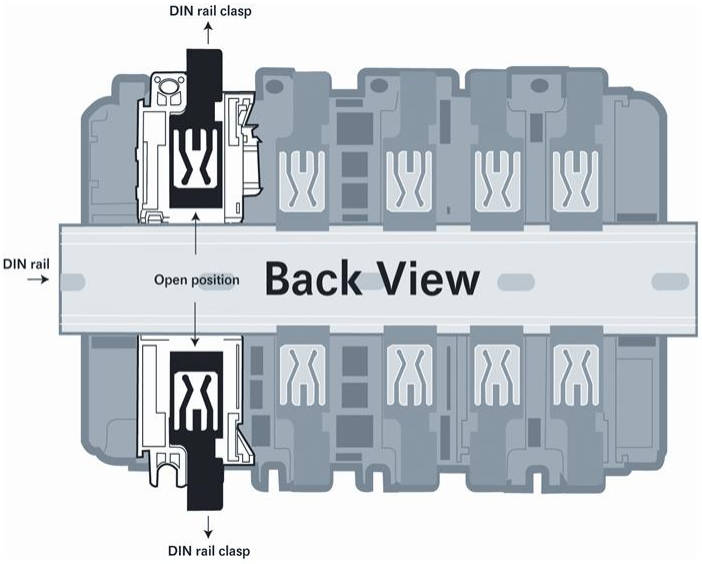


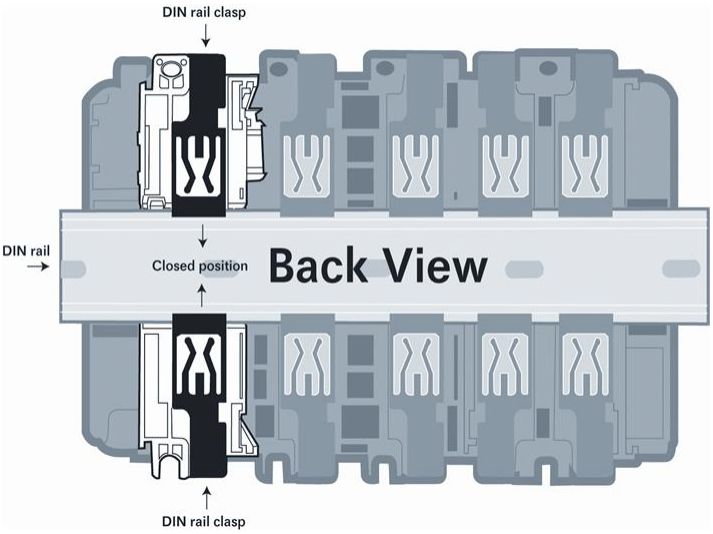
Move the Bus Lever to the left until it clicks



4 Close all DIN-rail latches.

5 Press the DIN-rail mounting area of the controller against the DIN-rail. The latches will momentarily open and lock into place.





1.8 Sample Add-On Instruction Import Procedure

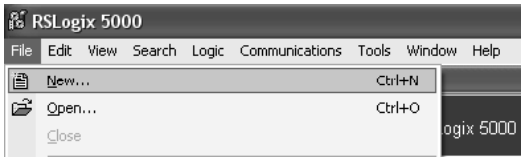
Note: This section only applies if you are using RSLogix 5000 version 16 or higher.

The following file is required before you start this procedure. Copy the file from the ProSoft Solutions DVD, or download it from **Error! Hyperlink reference not valid.**.

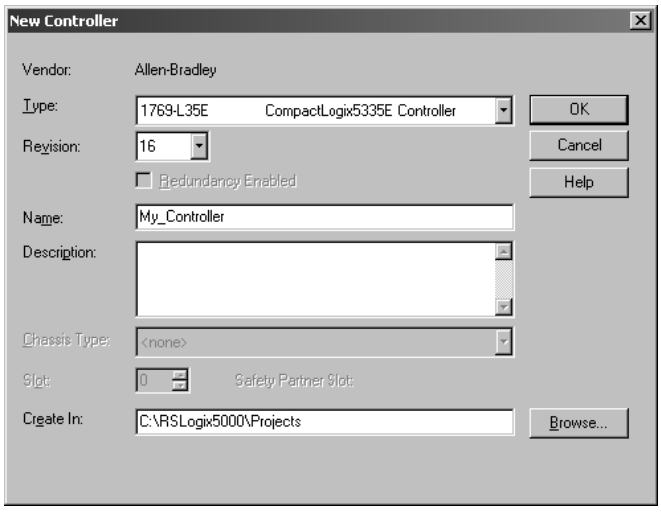
File Name	Description
AOIPS69DPM.L5X	L5X file contains the Add-On Instruction, the user defined data types, data objects and ladder logic required to set up the PS69-DPM module

1.8.1 Create a new RSLogix 5000 project

1 Open the **FILE** menu, and then choose **NEW…**

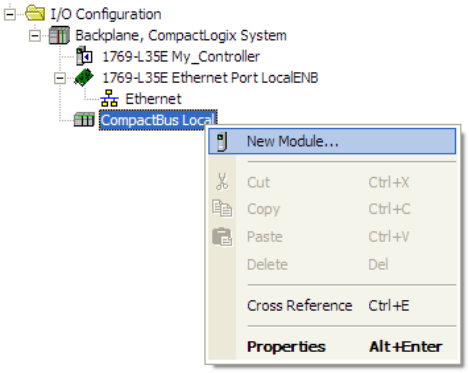


2 Select **REVISION** 16 or newer.

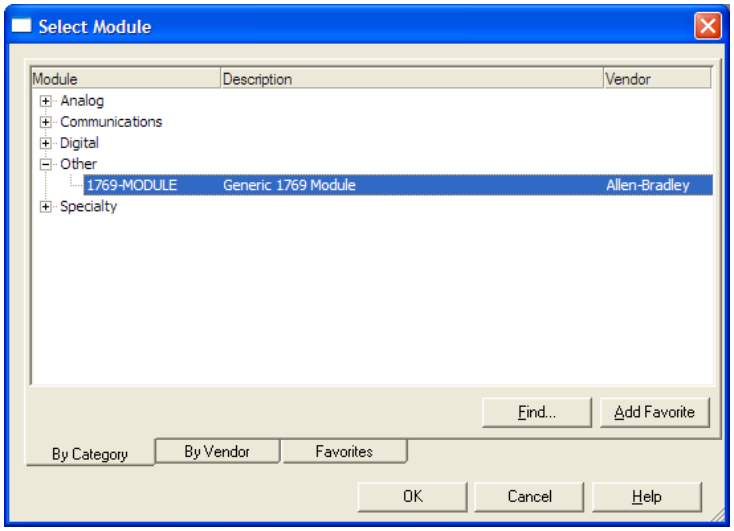


1.8.2 Create the Module

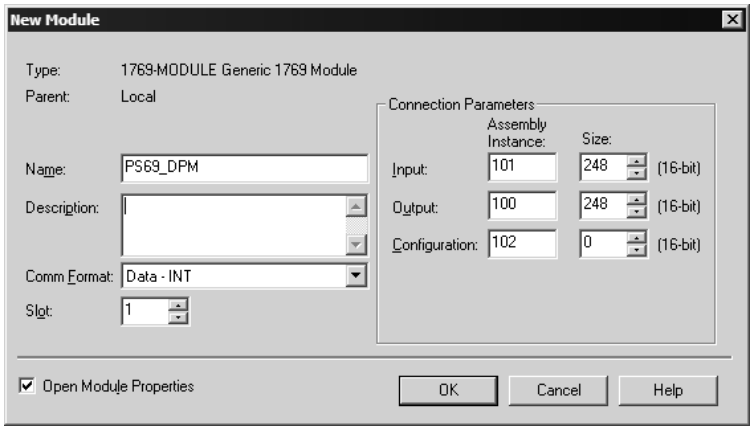
1 Right-click I/O Configuration and choose **NEW MODULE…**



2 Select **1769-MODULE**

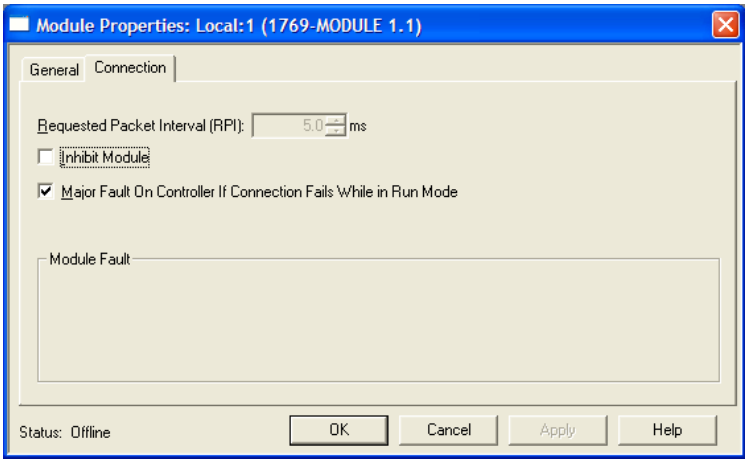


3 Set the Module Properties values as follows:

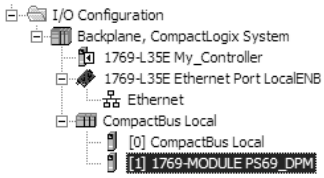


Parameter	Value
Name	Enter a module identification string. Example: PS69PDPMV1
Description	Enter a description for the module. Example: PROFIBUS DPV1 Master.
Comm Format	Select **DATA-INT**
Slot	Enter the slot number in the rack where the PS69-DPM module will be installed.
Input Assembly Instance	101
Input Size	248
Output Assembly Instance	100
Output Size	248
Configuration Assembly Instance	102
Configuration Size	0

4 On the *Connection* tab, check or un-check, as desired the **MAJOR FAULT** option.

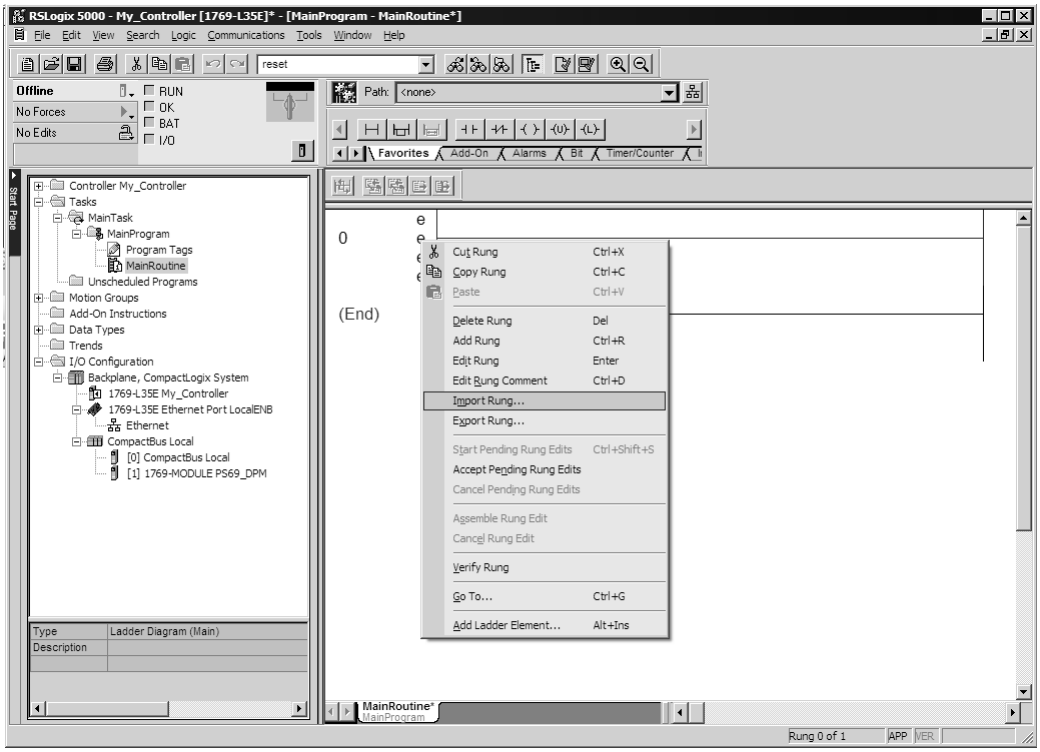


Now the PS69-DPM module will be visible at the *I/O Configuration* section.

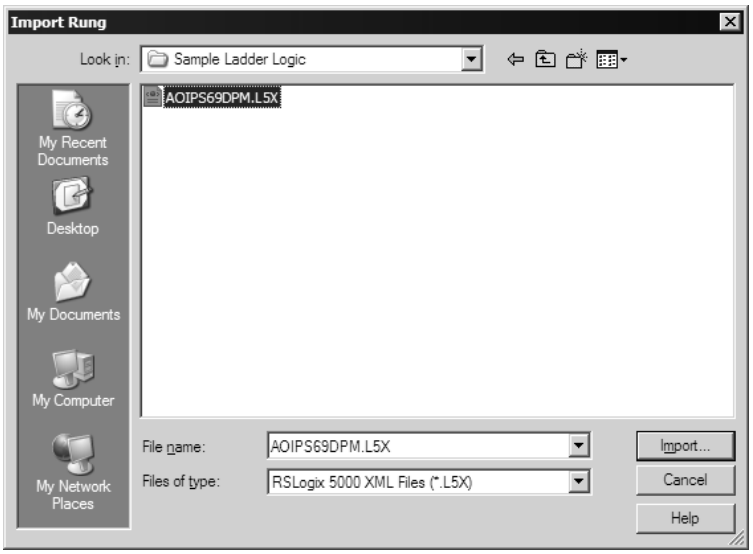


1.8.3 Import the Ladder Rung

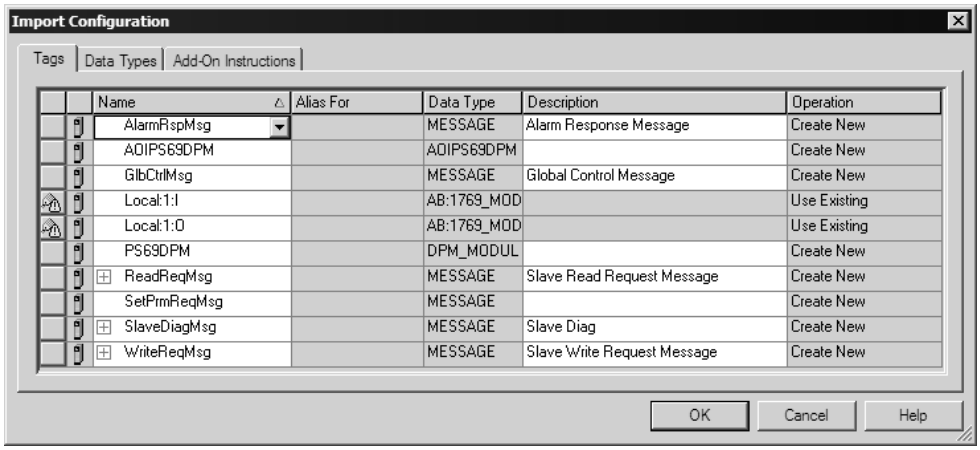
1 Open your application in RSLogix 5000.
2 To create a new routine, expand the **TASKS** folder, and then expand the **MAIN TASK** folder.
3 On the Main Program folder, click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW ROUTINE**.
4 In the *New Routine* dialog box, enter the name and description of your routine, and then click **OK**. In this example we are demonstrating the importing of the ladder rung using the default *MainRoutine*. In the case where you create a routine by an other name for placing the Add-On instruction, then in your original routine where your other ladder logic is located you need to add a rung with a jump instruction to the new routine holding the Add-On instruction.
5 Select an empty rung in the new routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNG…**



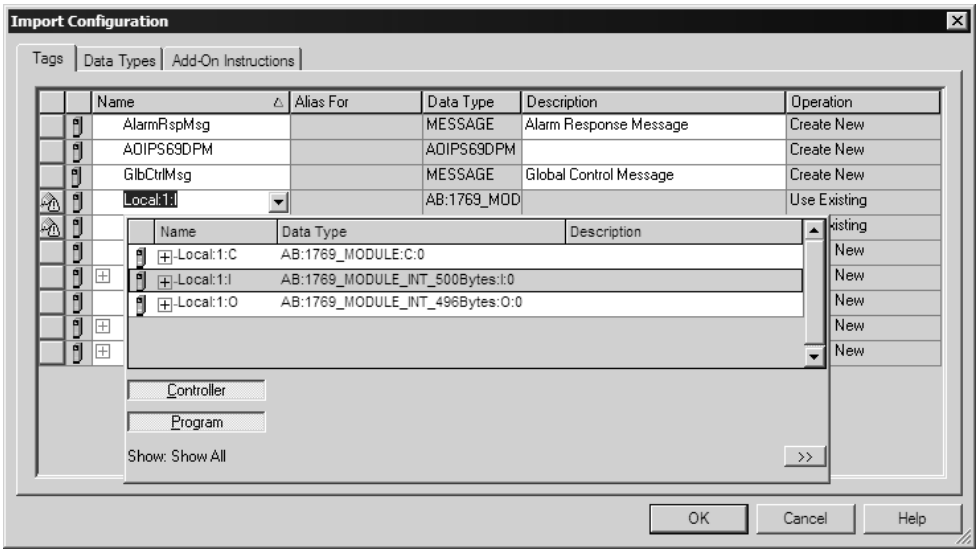
6 Select the *AOIPS69DPM.L5X* file



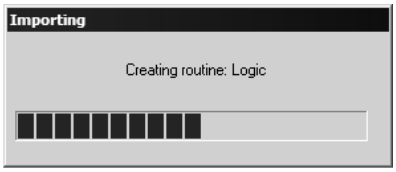
7 The following window will be displayed showing the controller tags to be created during the import procedure: If desired, the description, "PS69-DPM Interface AOI" may be typed into the description field for AOIPS69DPM.L5X file.



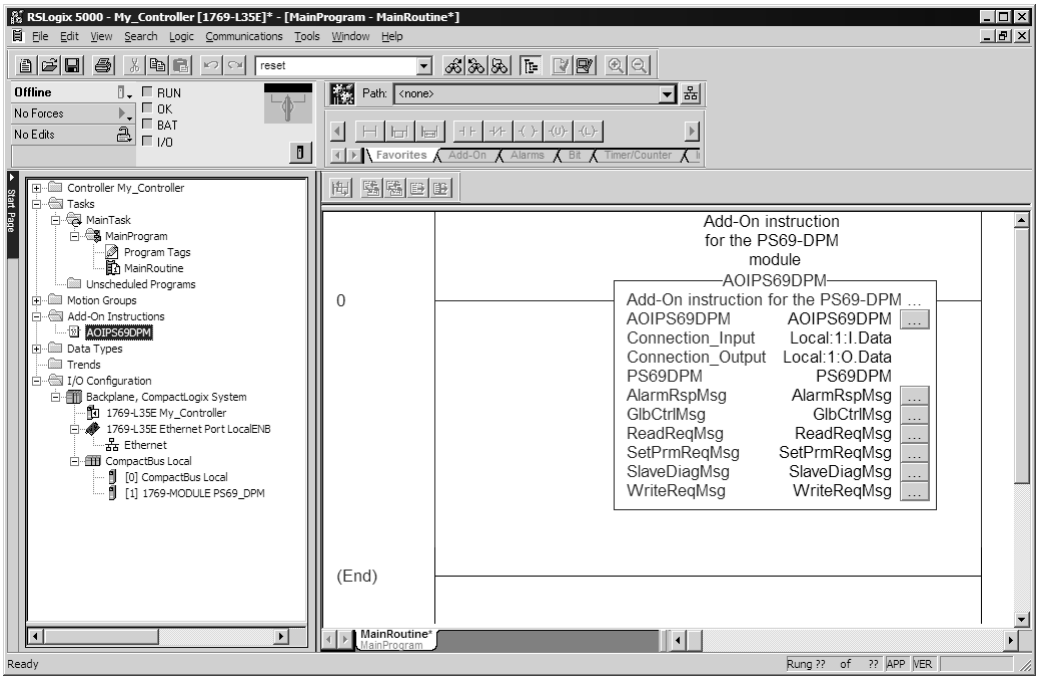
8 If you are using the module in a different slot (or remote rack) select the correct connection input and output variables associated to the module. If your module is located in slot 1 of the local rack this step is not required.



Click **OK** to confirm the import. RSLogix will indicate that the import is under progress:



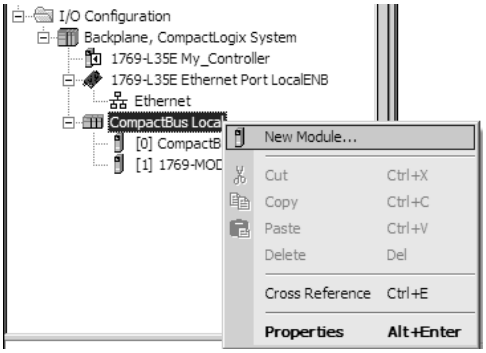
When the import is completed, the new rung with the Add-On instruction will be visible as shown in the following illustration.



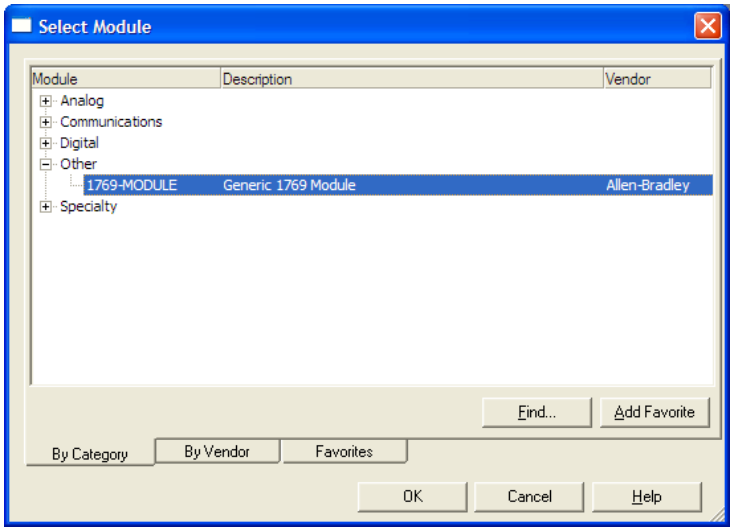
The procedure has also imported new user defined data types, data objects and the Add-On instruction to be used at your project.

1.8.4 Adding Multiple Modules (Optional)

Important: If your application requires more than one PS69-DPM module into the same project, follow the steps below and make certain that both modules are assigned identical Block Transfer Sizes.
In the I/O Configuration folder, click the right mouse button to open a shortcut menu, and then choose New Module.

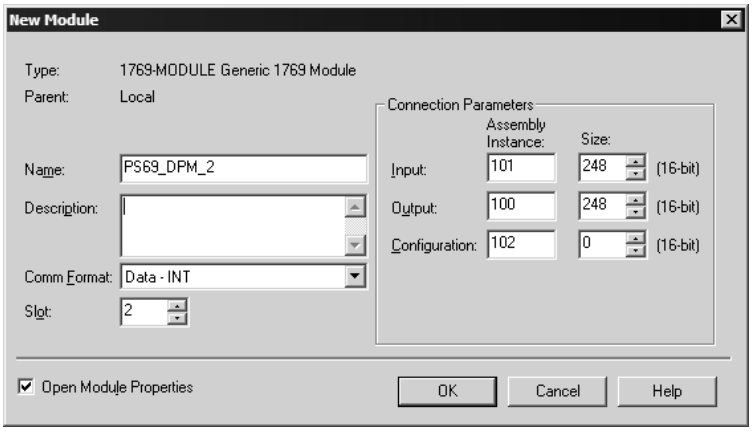


1 Select **1769-MODULE**.

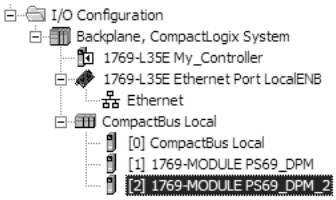


2 Fill the module properties as follows:

Parameter	Value
Name	Enter a module identification string. Example: PS69PDPMV1_2
Description	Enter a description for the module. Example: PROFIBUS DPV1 Master
Comm Format	Select **Data-INT**
Slot	Enter the slot number in the rack where the PS69-DPM module will be installed.
Input Assembly Instance	101
Input Size	248
Output Assembly Instance	100
Output Size	248
Configuration Assembly Instance	102
Configuration Size	0

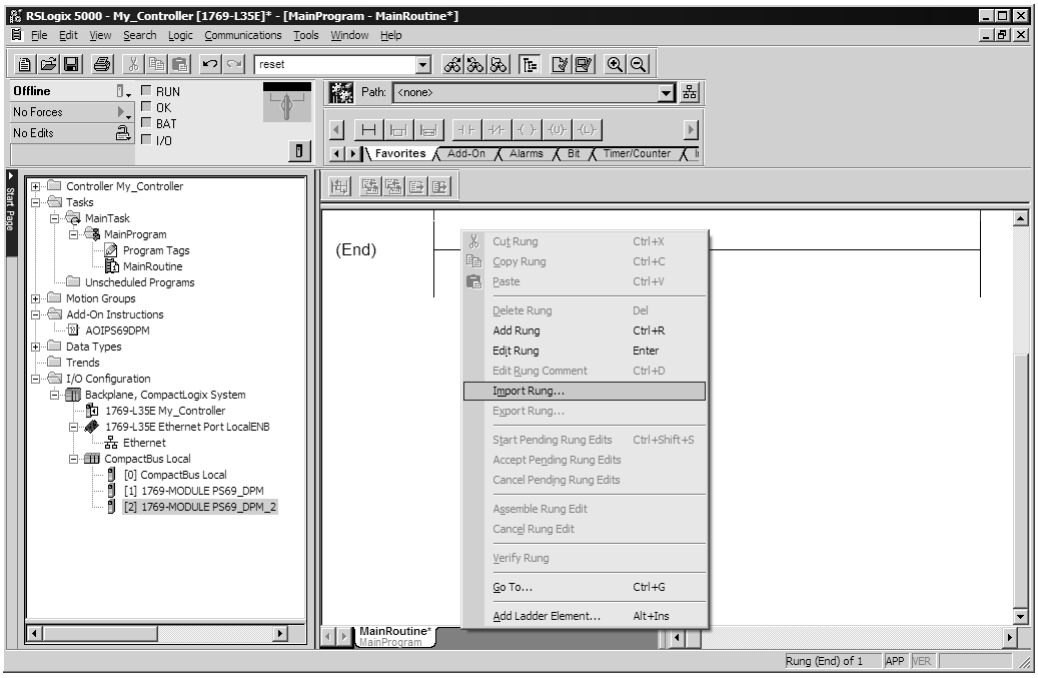


3 Click **OK** to confirm. The new module is now visible:

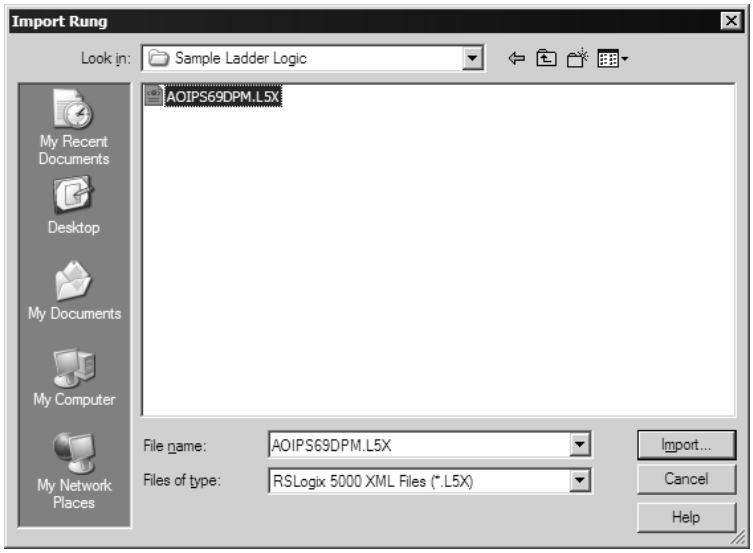


4 Expand the **TASKS** folder, and then expand the **MAINTASK** folder.
5 On the *MainProgram* folder, click the right mouse button to open a shortcut menu. On the shortcut menu, choose New Routine.
6 In the *New Routine* dialog box, enter the name and description of your routine, and then click **OK**.
7 Select an empty rung in the new routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNG…**

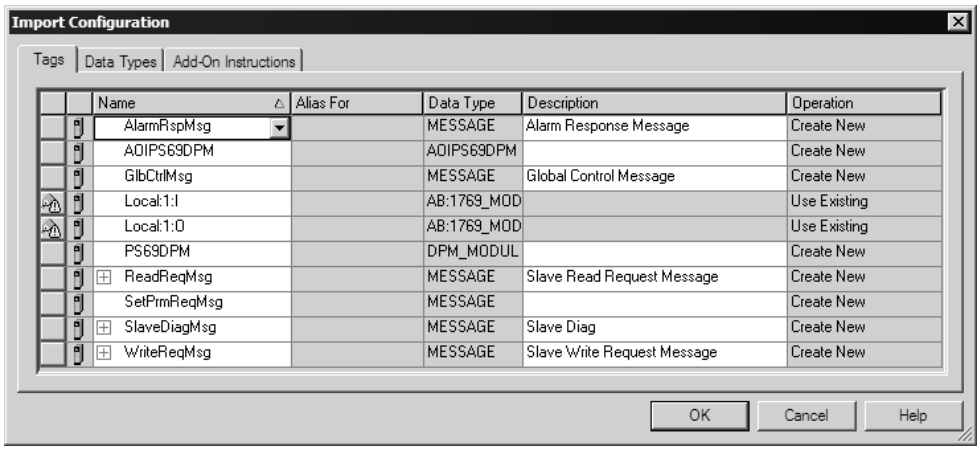
Note: It is not necessary to create a completely new routine. It is possible to add the PS69-DPM_2 module in the previously created routine. If you need to create a new routine, insert a jump instruction in the previous routine to the new routine.



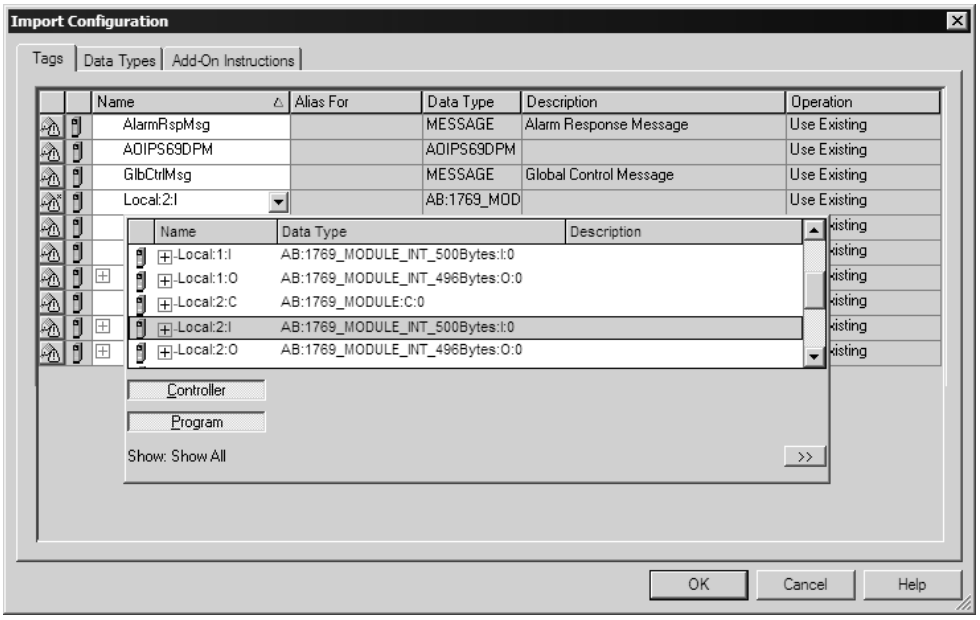
8 Select the *AOIPS69DPM.L5X* file

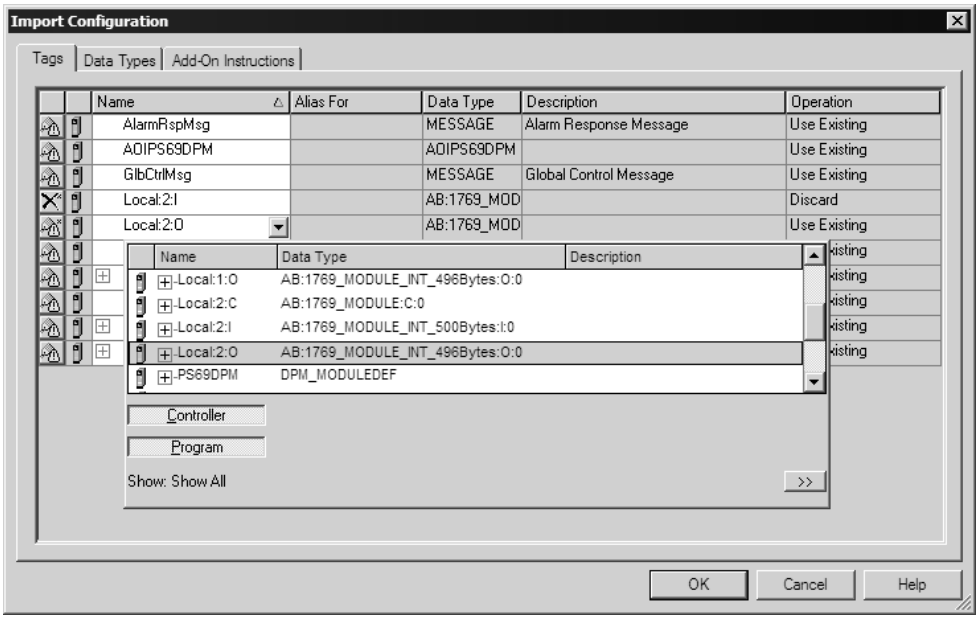


9 The following window will be displayed showing the tags to be imported:

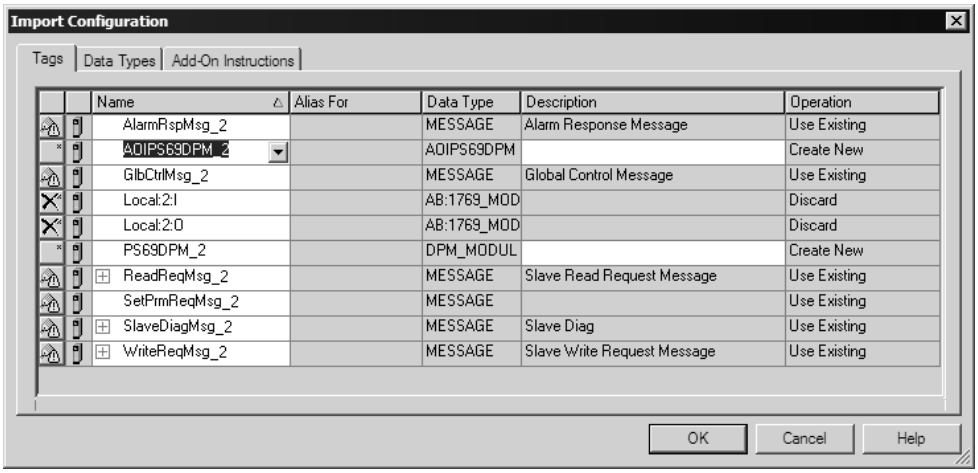


10 Associate the I/O connection variables to the correct module. The default values are Local:1:I and Local:1:O. These require re-assignment to the new module's location.

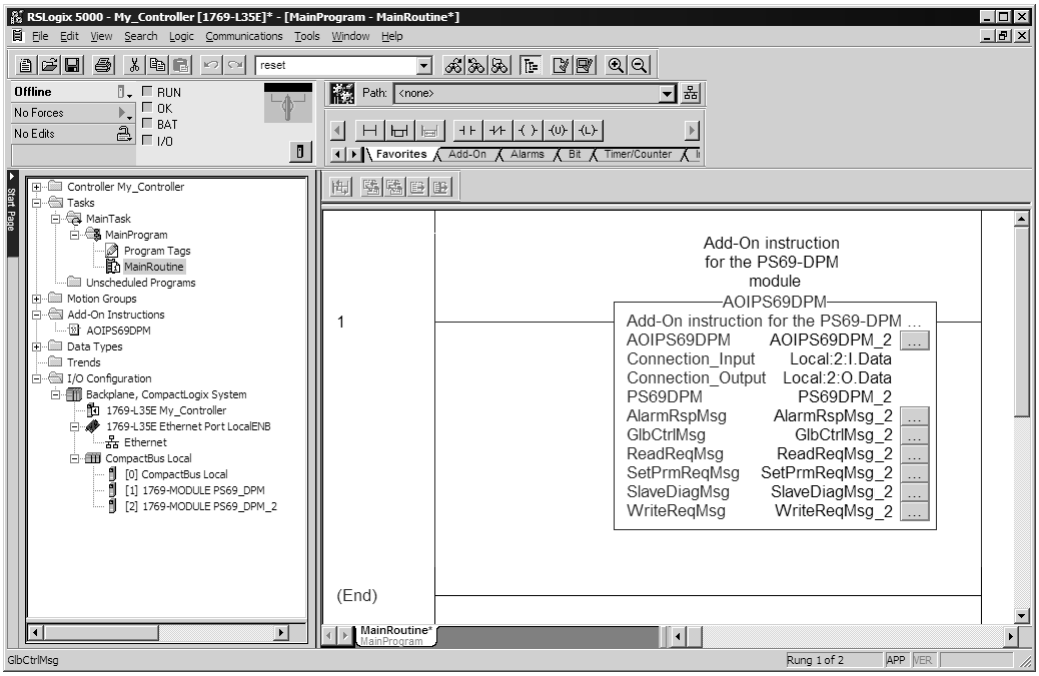




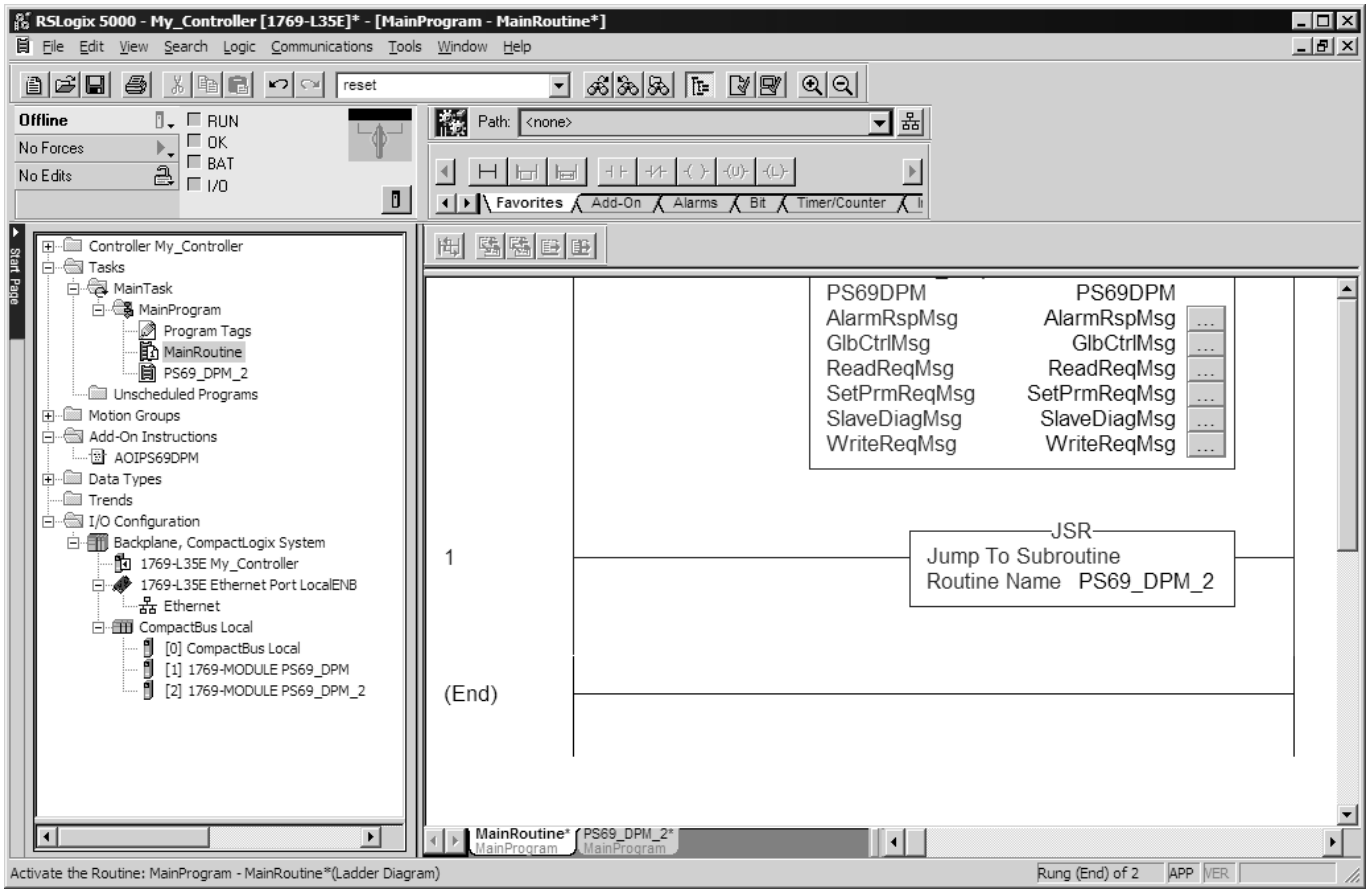
11 Change all the default tag names to avoid conflict with existing tags from previous imports. In this step, you should append a string to the default tag names, such as "_2", as shown in the following illustration.



12 You will be prompted to confirm your change. Click **OK** to continue.



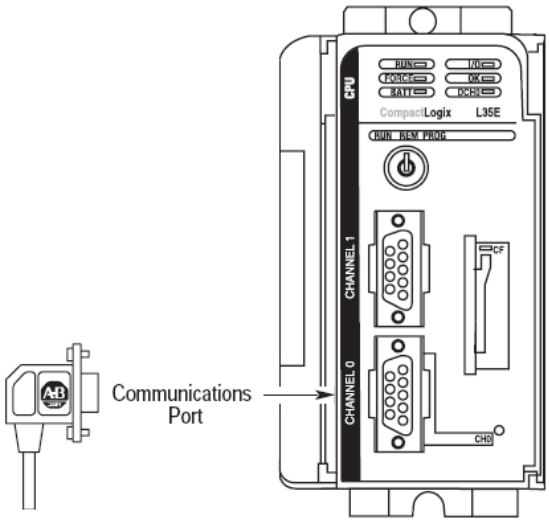
Because the second module's logic was created in a new routine, enter a rung in the Main routine with a JSR instruction to the new routine to enable the PLC logic to communicate with both modules.



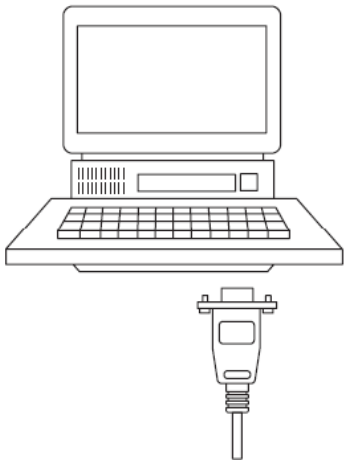
The setup procedure is now complete. Save the project and download the application to your CompactLogix processor.

1.9 Connecting Your PC to the Processor

1 Connect the right-angle connector end of the cable to your controller at the communications port.



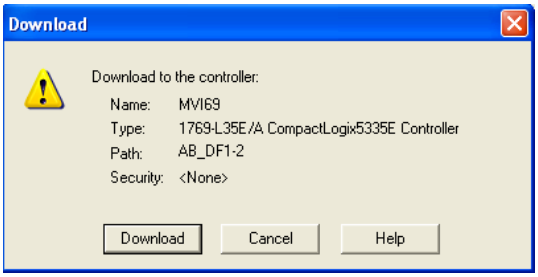
2 Connect the straight connector end of the cable to the serial port on your computer.



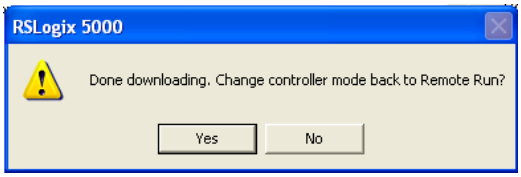
1.10 Downloading the Sample Program to the Processor

Note: The key switch on the front of the CompactLogix processor must be in the REM or PROG position.

1. If you are not already online to the processor, open the **COMMUNICATIONS** menu, and then choose **DOWNLOAD**. RSLogix will establish communication with the processor.
2. When communication is established, RSLogix will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



3. RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, RSLogix will open another confirmation dialog box. Click **OK** to switch the processor from PROGRAM mode to RUN mode.

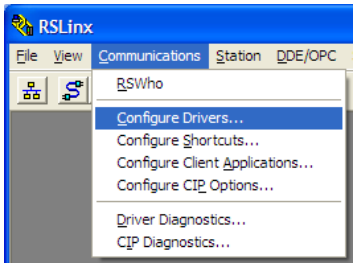


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

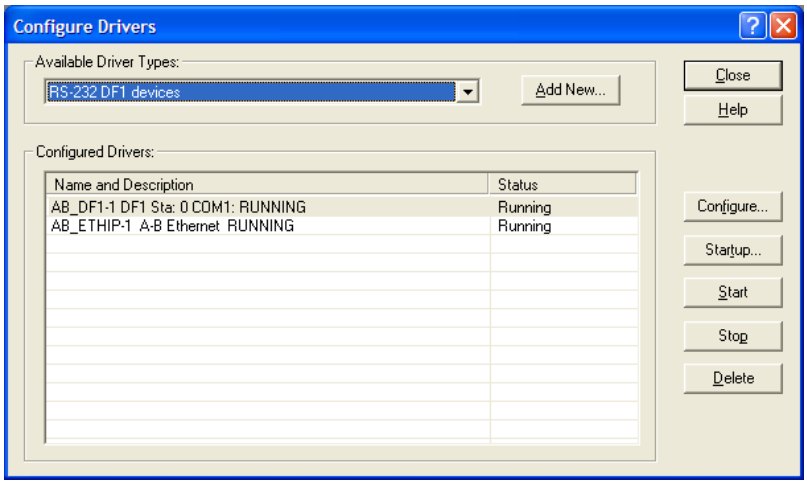
1.10.1 Configuring the RSLinx Driver for the PC COM Port

When trying to connect serially, if RSLogix is unable to establish communication with the processor, follow these steps.

1 Open *RSLinx*.
2 Open the **COMMUNICATIONS** menu, and click **CONFIGURE DRIVERS**.

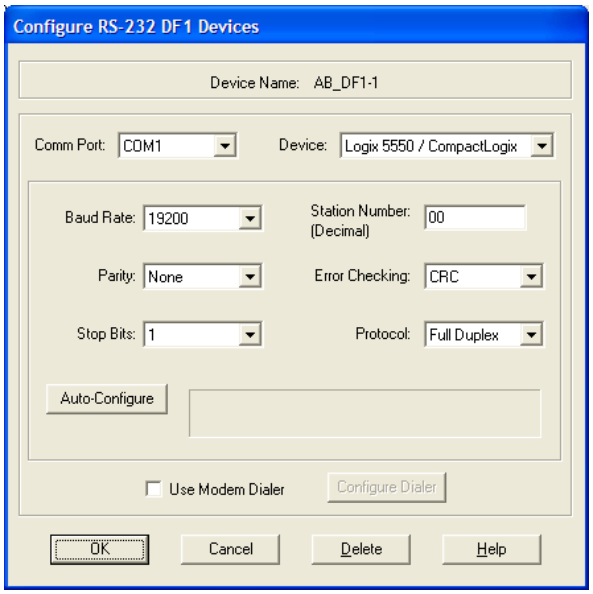


This action opens the *Configure Drivers* dialog box.



Note: If the list of configured drivers is blank, you must first choose and configure a driver from the *Available Driver Types* list. The recommended driver type to choose for serial communication with the processor is *RS-232 DF1 Devices*.

3 Click to select the driver, and then click **CONFIGURE**. This action opens the *Configure RS-232 DF1 Devices* dialog box.



4 Click the **AUTO-CONFIGURE** button. RSLinx will attempt to configure your serial port to work with the selected driver.

5 When you see the message *Auto Configuration Successful*, click the **OK** button to dismiss the dialog box.

Note: If the auto-configuration procedure fails, verify that the cables are connected correctly between the processor and the serial port on your computer, and then try again. If you are still unable to auto-configure the port, refer to your RSLinx documentation for further troubleshooting steps.

2 Configure the PROFIBUS Network

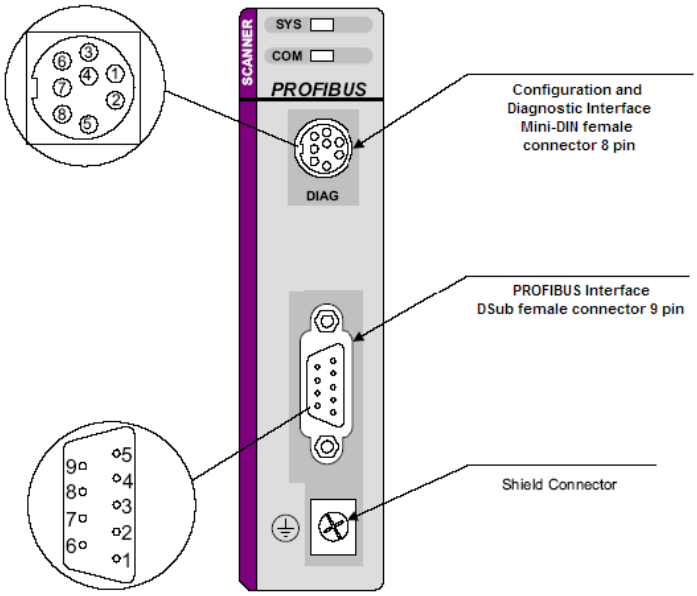
In This Chapter

The following sections describe the individual steps for configuration and start-up of the PS69-DPM module. Install the PROFIBUS Master module into a free slot in the CompactLogix or MicroLogix 1500 controller. The information for installation of communication modules in CompactLogix or MicroLogix 1500 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 / MicroLogix 1500 project of the RSLogix 5000 / RSLogix 500 programming tool.
- Parameterization and configuration of the PROFIBUS Master with the PROSOFT.fdt (SYCON.net) configuration tool.
- Creating the data objects and the ladder diagram in RSLogix 5000 / RSLogix 500.

2.1 Scanner (PROFIBUS-DP-Master)



2.1.1 RIF 1769-DPM Compatibility

You can easily convert an existing RIF 1769-DPM project in PROSOFT.fdt without modifying module configuration or ladder logic.

Use this procedure if you are:

- Replacing an existing RIF1769-DPM module with a new PS69-DPM module

  or
- Adding a PS69-DPM module to an existing RIF 1769-DPM project.

The configuration and ladder logic from your RIF 1769-DPM project will be fully compatible with the new PS69-DPM module.

Important: This procedure converts your Sycon.net SPJ files. SPJ files that are opened and converted in PROSOFT.fdt can no longer be edited in Sycon.net. Take care to save copies of your Sycon.net project files.
Important: SYCON.net and PROSOFT.fdt share components. The two applications cannot coexist on the same PC.

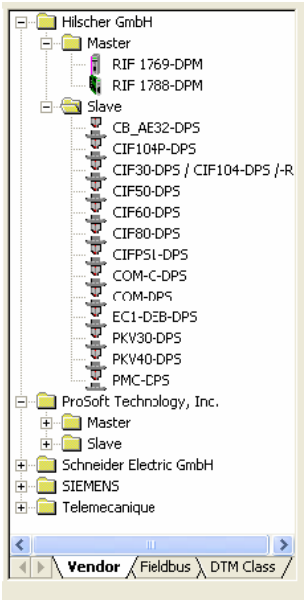
First, back up your SYCON.net files

1 Open your project (*.spj file) in SYCON.net, and then save it to your "My Documents" folder.
2 From SYCON.net use the "Save As" option and enter a meaningful file name. SYCON.net will create a subfolder with the filename you entered. In that subfolder, SYCON.net will also create an XML file containing additional configuration information.
   a In Windows Explorer, open your My Documents folder, and locate the subfolder created by SYCON.net. Make a note of the filename and location.
   b Open that folder and any subfolders, and locate the SYCON_net.XML file. The file will located in a subfolder with a naming pattern similar to "_S129"

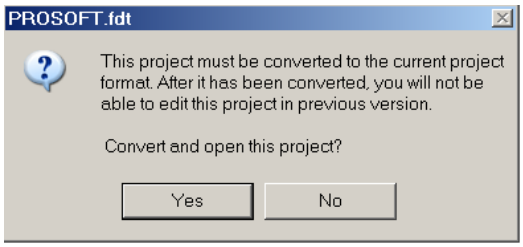
   For example, if you saved your project as "MyProject", you will find a folder named "MyProject" in your My Documents folder. In the "MyProject" folder, you will find another subfolder, named similar to "_S129". Make a note of the filename and location.
3 Close SyCon.net.
4 Create backup copies of your existing SyCon.net project files and related *.GS* files.
5 Uninstall SyCon.net software and make sure all folders are deleted.

Next, open and convert your SYCON.net files in PROSOFT.fdt

1 Install PROSOFT.fdt, if you have not already done so.
2 Start PROSOFT.fdt, and wait while it searches your computer for GSD files. The GSD files will appear in the device catalog, in the right pane of the PROSOFT.fdt window.



3 Open each project file (SPJ file) that you saved from Sycon.net. PROSOFT.fdt will prompt you to convert the project.



4 Save and download the project to the module.

2.2 PROSOFT.fdt (SYCON.net)

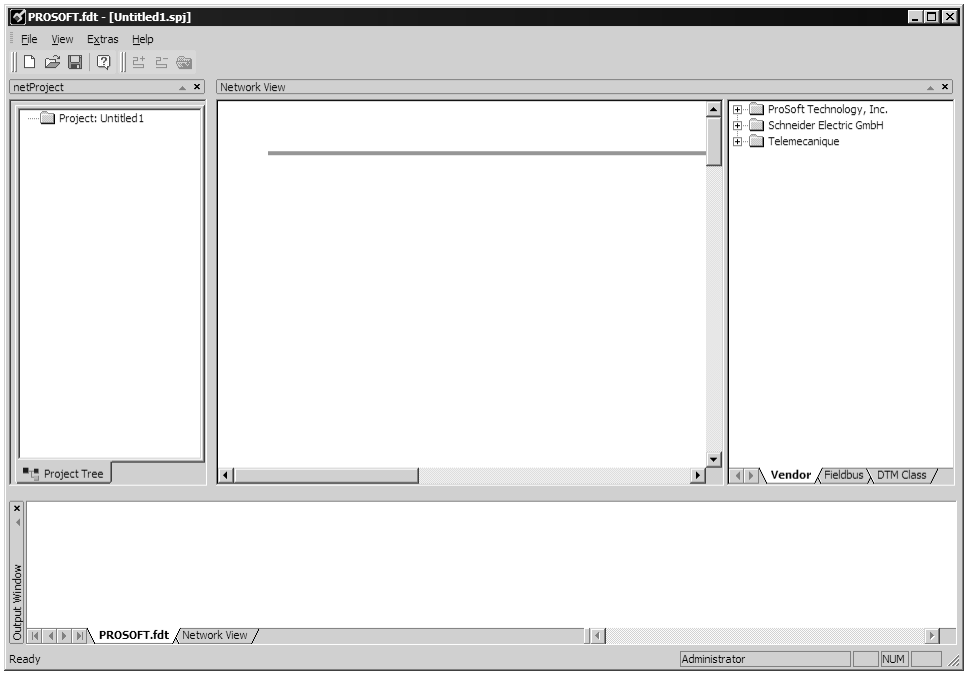
The following section will detail the basics of using the configuration and diagnostic software PROSOFT.fdt (SYCON.net) to configure the PROFIBUS-DP Master module and Slave I/O system.

2.2.1 General

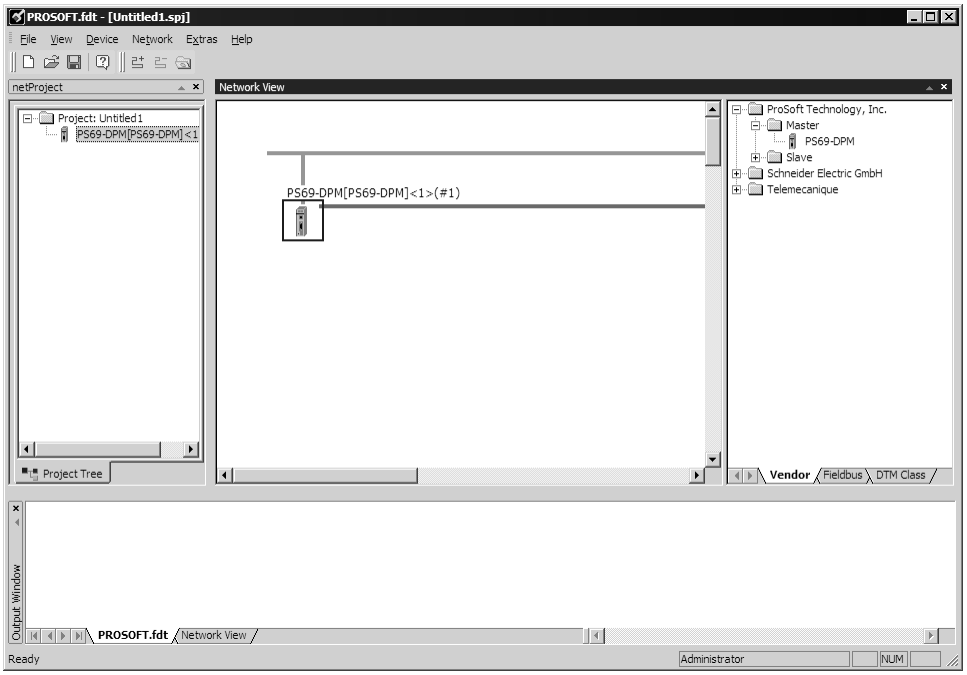
The PROFIBUS-DP system is configured by using the configuration and diagnostic tool PROSOFT.fdt (SYCON.net). The configuration is downloaded to the module and stored into the Flash memory of the Master module by using the download function of PROSOFT.fdt (SYCON.net). Downloading of the configuration is done via the diagnostic interface. Connect the diagnostic interface to a serial interface of the PC. Start PROSOFT.fdt (SYCON.net) from the installation folder. Follow the basic steps to create a PROFIBUS configuration. A comprehensive explanation for all configuration steps can be found in the Online help in the **Help > Topics...** menu.

2.2.2 Create a New Project

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



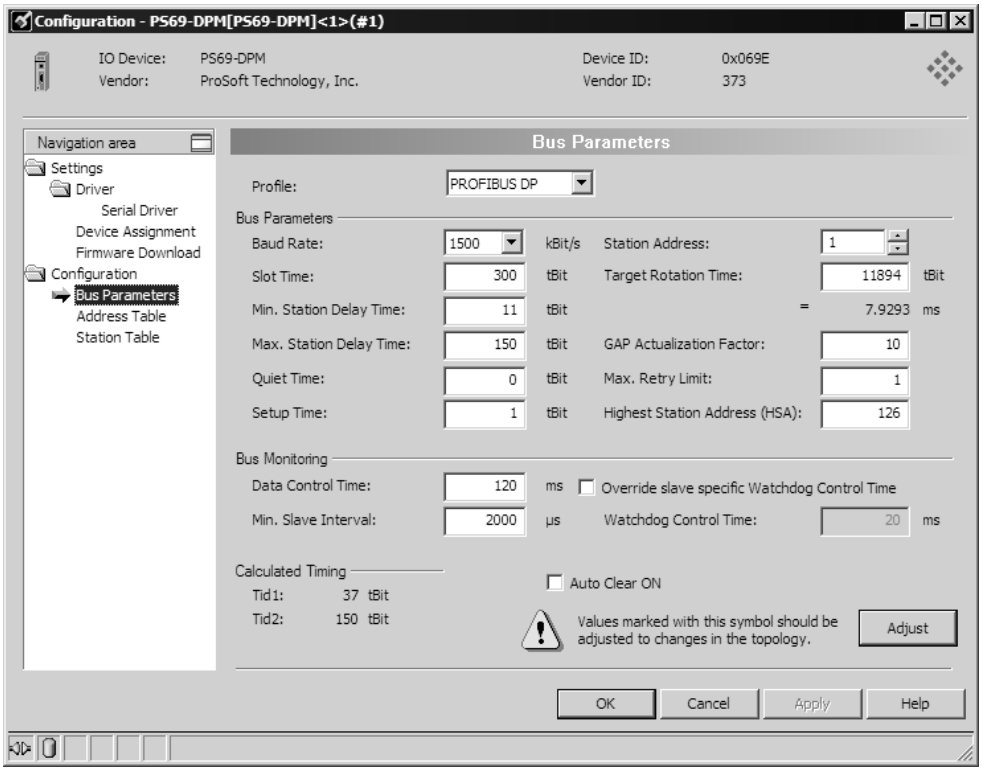
Then, in PROSOFT.fdt (SYCON.net), click and hold the left mouse button and drag the PS69-DPM device from the device catalog area to either the Network View bus or the netProject screen, add the device to the project by releasing the left button when the **"+**" sign appears. Your screen should show the following:



The Master is now ready for configuration.

2.2.3 Configuration of the PS69-DPM Master

Double click on the Master that appears in the Network View or the netProject window. In the navigation area on the left side of the Configuration dialog box, select Bus Parameters.

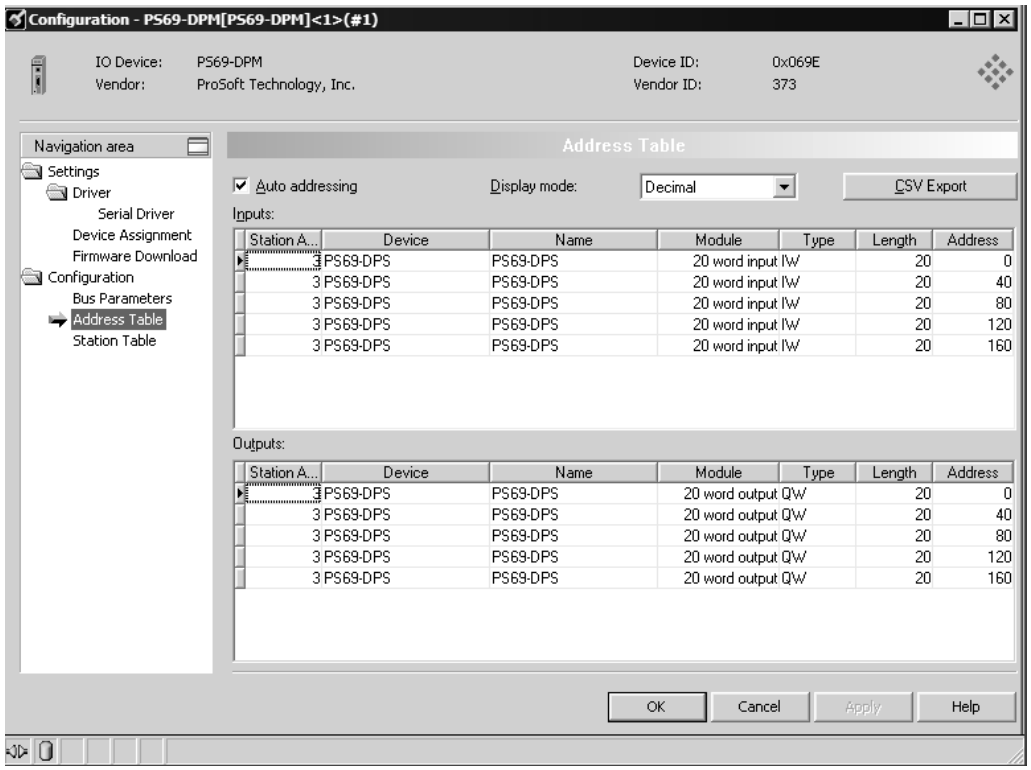


Select the Baud Rate and Station Address for your Master. The rest of the settings do not need adjustment and should be automatically calculate when changing the Baud Rate setting. The default settings cover the most of cases.

Under some circumstances it might be necessary to adjust these values. If these settings need to be changed, please refer to the Operating Instruction Manual for "DTM for PROFIBUS Master Devices" for the meaning of these values and proper settings.

Address Table

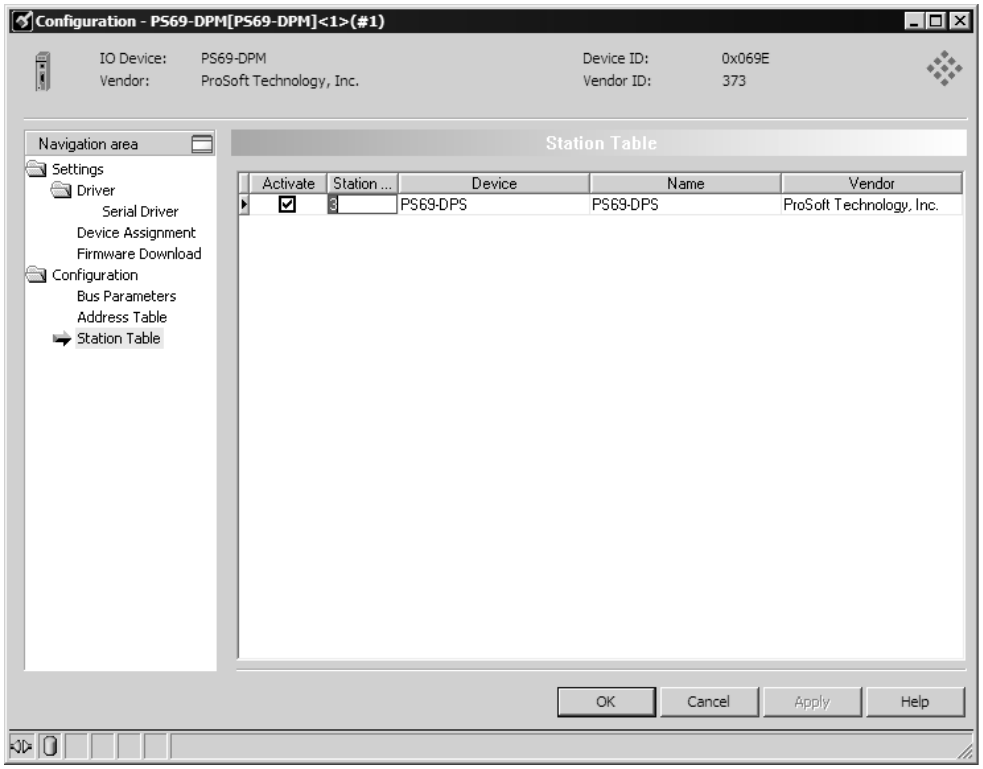
Select Address Table in the navigation area. The dialog will appear as shown in the following illustration.



The Address Table describes the Slaves configured and their dual port memory offsets for input and output data. You do not have to change anything in this setup screen as long as the Enable Auto Addressing checkbox is checked. If unchecked, you can change the offsets manually.

Station Table

Note: The following illustration shows how to change the bus address for a Slave. This dialog box is the only place that allows you to change the slave address.



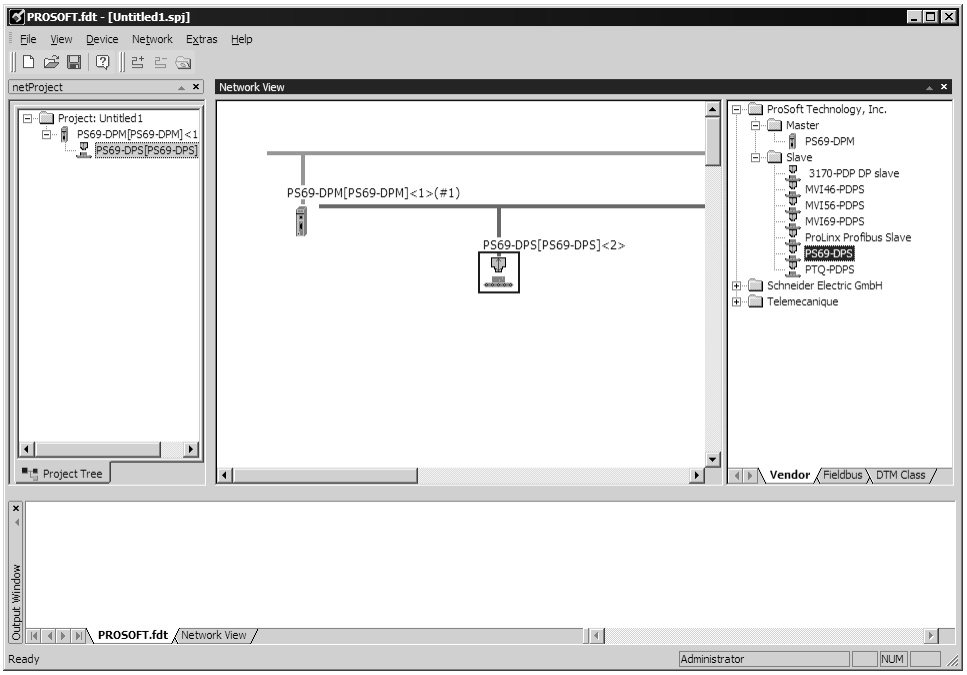
Complete the process click the **OK** or **Apply** button and close the dialog box. The Master settings are now complete.

Note: The slave offset addresses shown here are not the same you will find in the PLC memory. You have to add 44 words the Input address (to allow space for the Status Information area) and 8 words to the Output address (to allow space for the Command Information area). See IO Arrays Overview for details.

2.2.4 Configuration of PROFIBUS Slaves

Add a Slave to a project

In the PROSOFT.fdt (SYCON.net) project screen, click and hold the left mouse button and drag a Slave device from the device catalog area to either the Network View 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:



Add a Slave to PROSOFT.fdt (SYCON.net) Device Catalog

If the PROFIBUS Slave is not listed in the Device Catalog it has to be added to PROSOFT.fdt (SYCON.net). To add a slave to PROSOFT.fdt (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 configured by DTM technology, install the DTM software on your PC that was delivered with the slave. Then reload the Device Catalog in PROSOFT.fdt (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 PROSOFT.fdt (SYCON.net) application.
- Copy the GSD file manually into the folder:
  - for PROSOFT.fdt

```
\Program Files\ProSoft Technology\PROSOFTnet\PBGenericSlaveDTM\GSD
```

  - for SYCON.net

```
\Program Files\Hilscher GmbH\SYCONnet\PBGenericSlaveDTM\GSD for SYCON.net
```

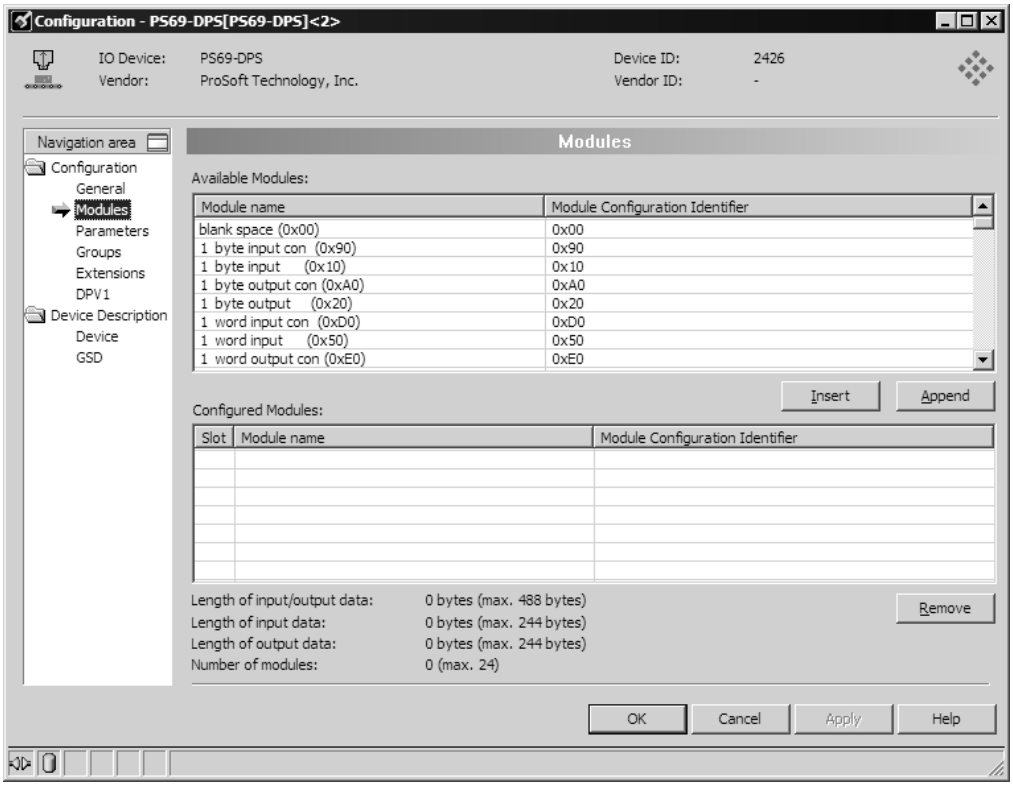
Reload Device Catalog

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

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

Slave Settings

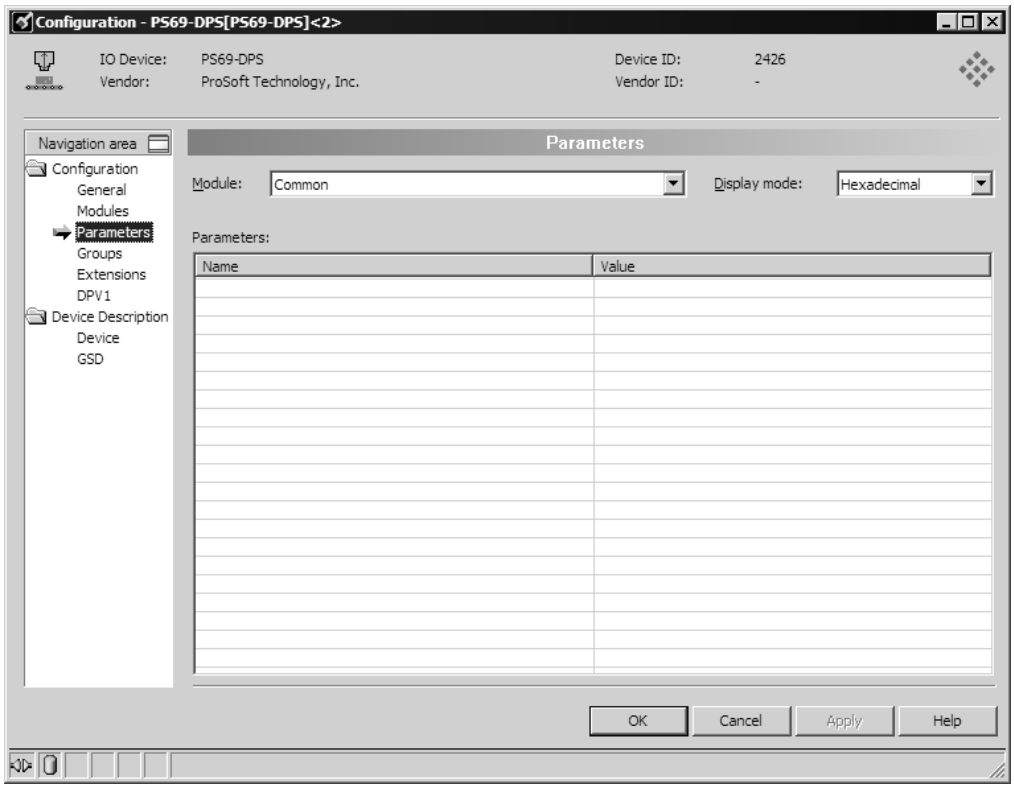
Double click on the slave that has been added. A dialog box similar to the one show below appears.



Insert or append appropriate data modules from the list of available modules into the list of configured modules.

Slave Settings Parameter Tab

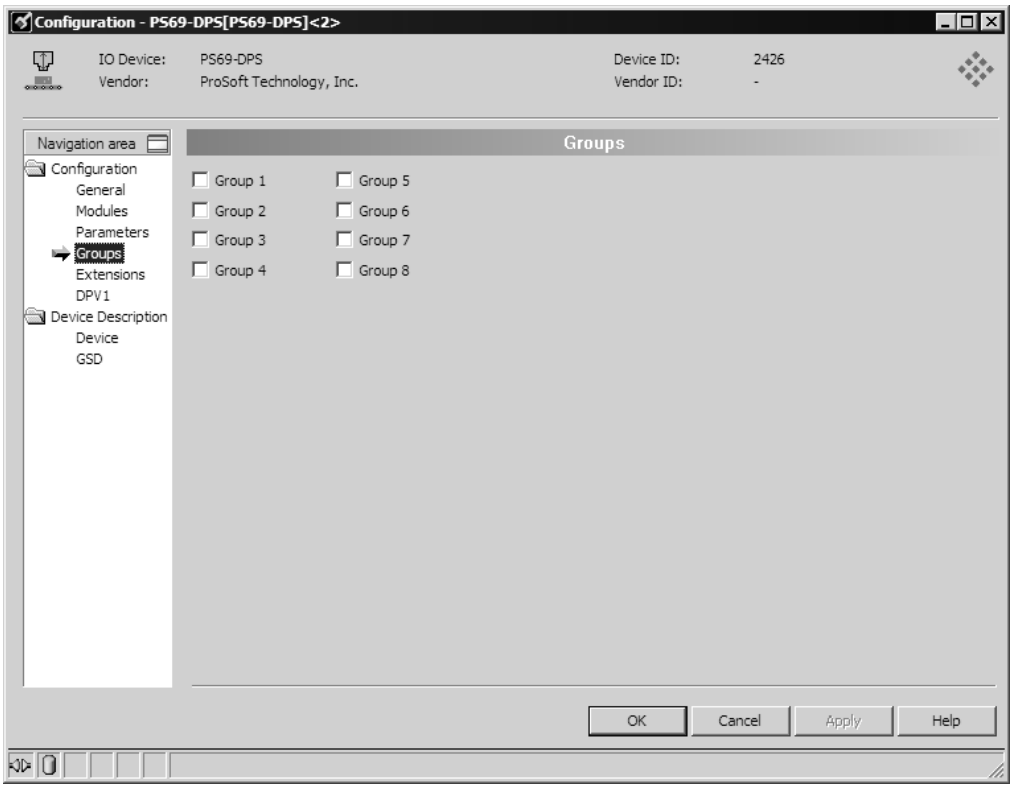
Select **Parameters** in the navigation area. The dialog should appear as shown in the figure below.



Verify or change the parameters as required. These parameters are vendor and device specific.

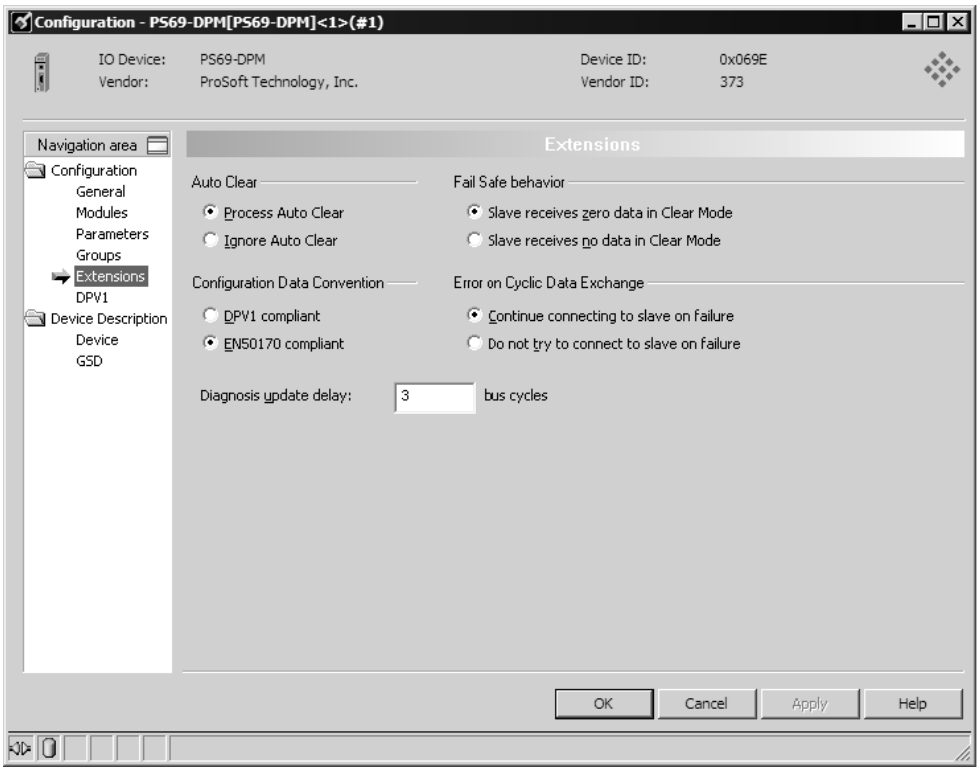
Slave Groups

Select **Groups** in the navigation area. The dialog should appear as shown in the figure below.



A Slave can be assigned to be member of one or more groups. The group membership functions as a filter for the Sync and Freeze commands. The Global Control telegram (containing Sync and Freeze commands) is sent as a broadcast telegram that allow slaves synchronizing their input and output data. Only the Slaves assigned to groups react on Sync and Freeze commands.

Select **Extensions** in the navigation area to view and change additional slave settings.



Make the appropriate settings for your application. Complete the process and click the **OK** or **Apply** button to close the dialog box. Repeat the above process for every Slave in your system. Save the project with **File > Save As...**

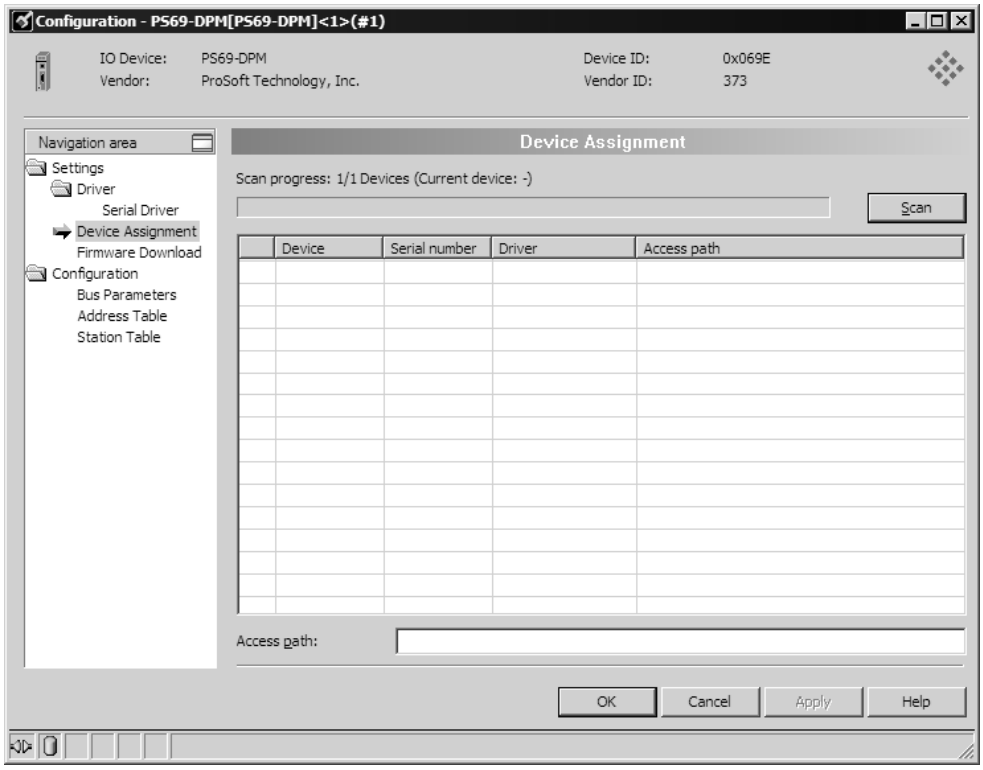
2.2.5 Project Download

Important: Before you download the project, you must change the processor to PROG mode. You cannot download the project to the PS69-DPM module while the processor is in RUN mode".

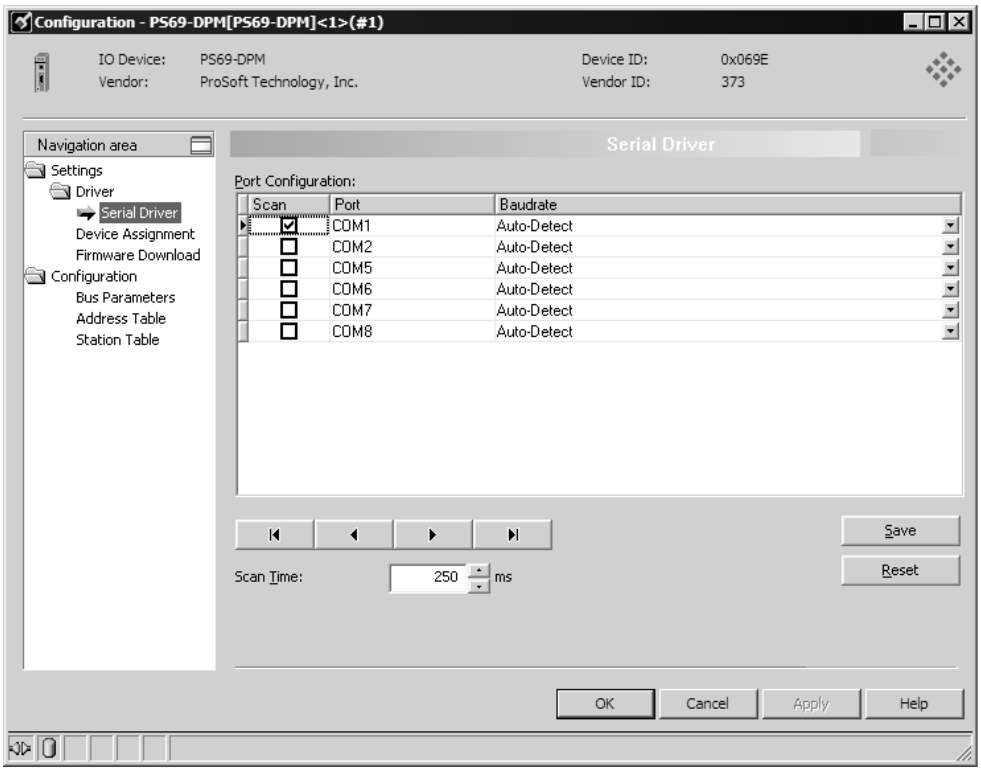
Once saved your project is now ready to be downloaded to the Module. Connect the serial port of your PC to the Diagnostic port on the front of the Master module using supplied interface cable (CABLE-SRV-MD8). Follow the steps below to download your project.

Step1: Device Assignment

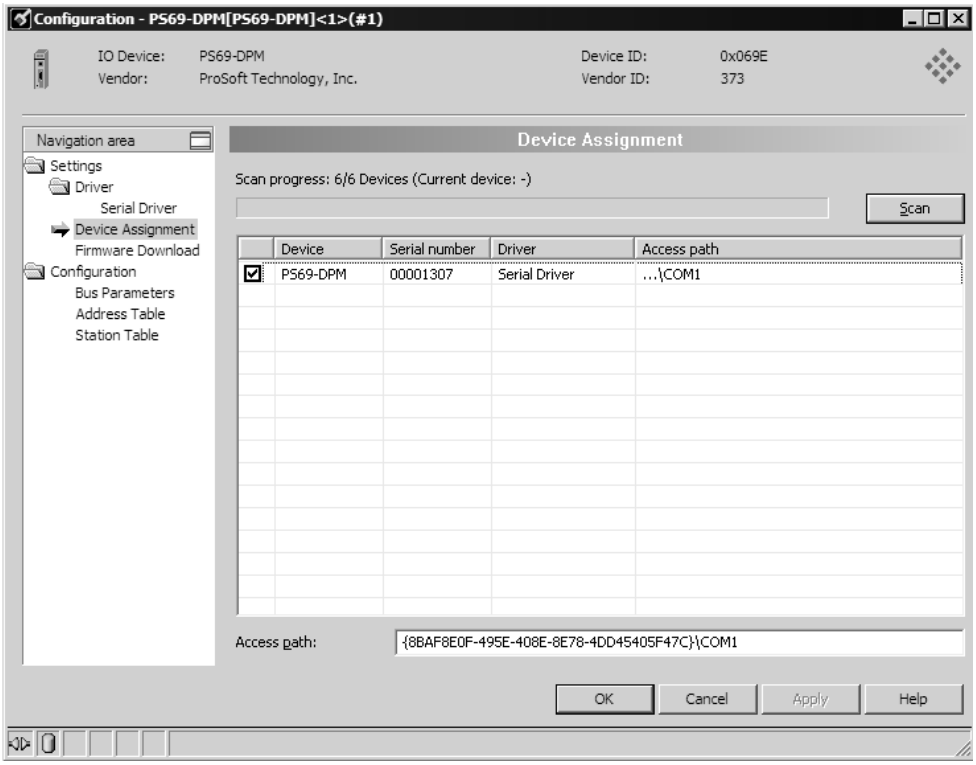
In the PROSOFT.fdt (SYCON.net) project screen, double-click on the Master you have added in either the Network View bus or the netProject screen, the Master should be highlighted with a Blue box. In the navigation area on the left of the Configuration dialog box, select **Device Assignment**.



If the master is connected to a COM port on the PC, it will scan automatically for available COM ports. Select the COM port to associate with the PS69-DPM.



Click the **Scan** button to retrieve a list of available devices and drivers, and then select the PS69-DPM from the list.



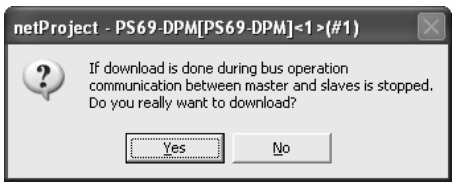
Step2: Connect to Master

In the PROSOFT.fdt (SYCON.net) project screen, click on the Master you have added in either the Network View 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 Network View bus or the netProject screen selecting **Connect** to establish a connection to the Master.

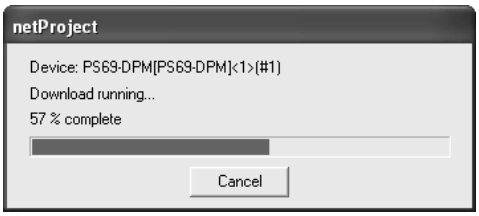
Important: It is not possible at this time to upload the project from the PS69-DPM. You must create the project as described in Create a New Project (page 37), and then download it to the module.

Step3: Download to Device

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



This dialog box is a warning regarding a possible interruption of the bus communication during the download. Click **Yes** to begin the download. The download progress dialog box will appear.



Step 4 - Set Processor to Run Mode

After the download has been completed, all required steps configuring the Master module have been done.

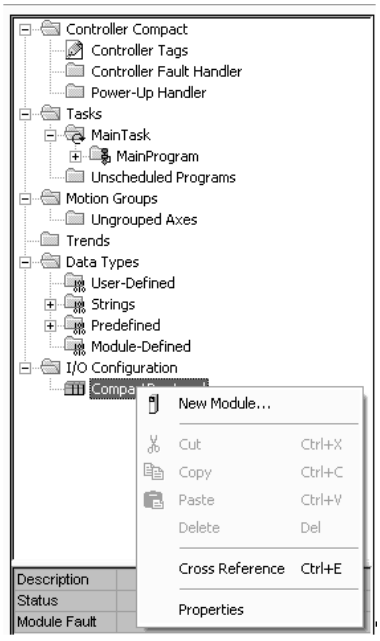
2.3 RSLogix 5000 (version 15 or lower)

The section below contains instructions for configuring the PS69-DPM module in a CompactLogix system using RSLogix 5000 version 15 or lower.

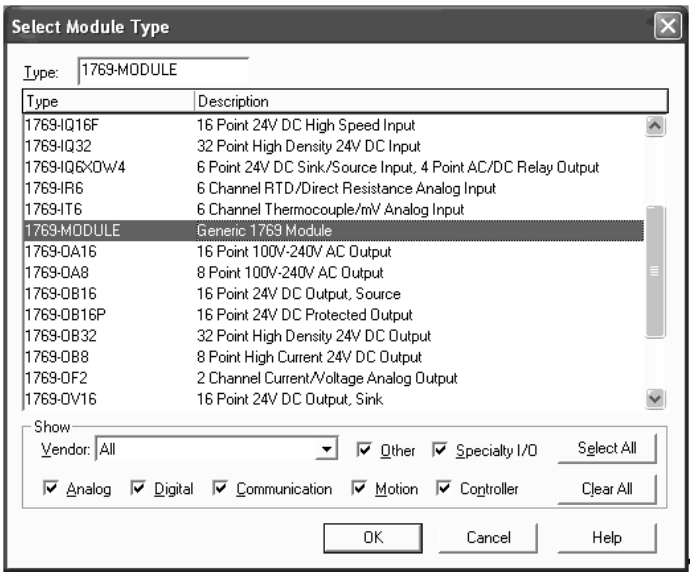
Note: The simplest way to startup the module in an RSLogix 5000 project is to use the example project provided on the ProSoft Solutions DVD. In this example project, the slot number in the configuration dialog of the module may have to be changed to match the users system.

2.3.1 Module Selection

Create a new project in RSLogix 5000 using a CompactLogix controller. Then the first step is to select the module and add it to your project. Right click on the I/O Configuration CompactBus Local of the controller project. Select **New Module** as shown below.



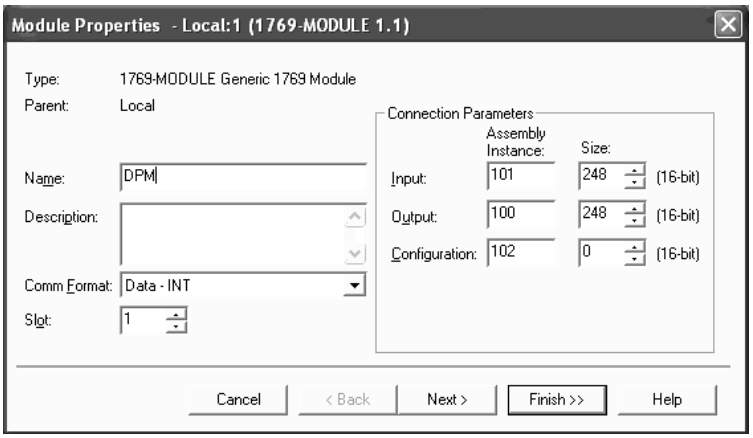
The following dialog box appears for the selection of the new module.



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

2.3.2 Module Properties 1

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



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 (X = 0 ... 204)
Output	100	8 + Y (Y = 0 ... 240)
Configuration	102	0

X = Number of Words configured for the Master module (PROFIBUS input data); input size can be in the range between 44 and 248 words
Y = Number of Words configured for the Master module (PROFIBUS output data); output size can be in the range between 8 and 248 words

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

Note: If the parameters do not correspond to the template values, then the controller cannot establish a communication relationship with the module.

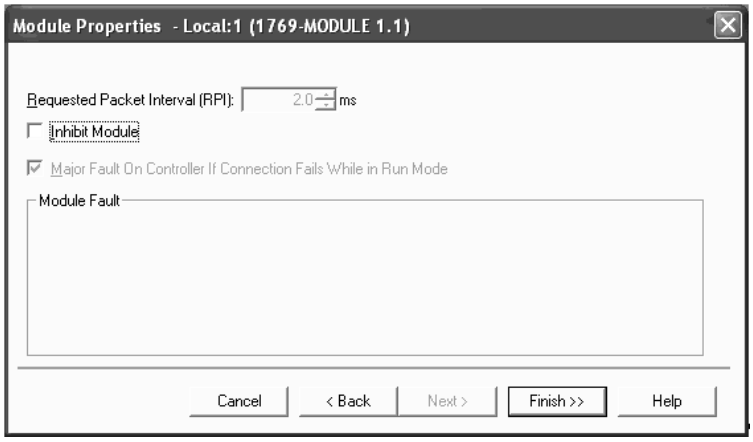
Select **Next >** for the next configuration dialog.

2.3.3 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.



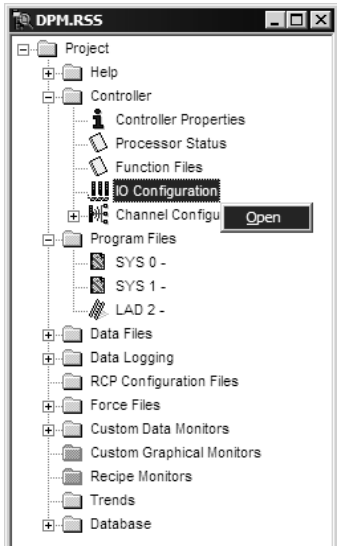
End the configuration of the module with **Finish>>**.

2.4 RSLogix 500

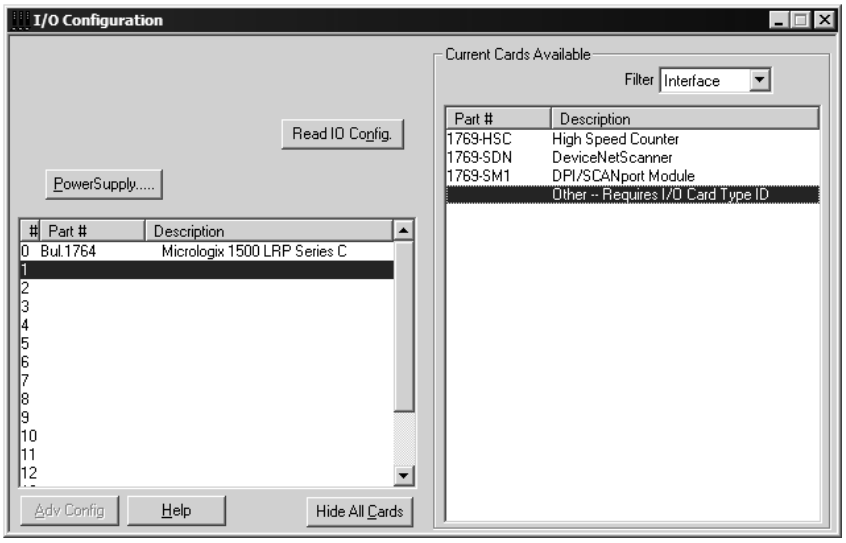
The sections below contain instructions for configuring the PS69-DPM module in a MicroLogix 1500 system using RSLogix 500.

2.4.1 Module Selection

Create a new project in RSLogix 500 using a MicroLogix 1500 controller. Then the first step is to select the module and add it to your project. Right click on the I/O configuration of the controller project. Select **Open** as shown below.



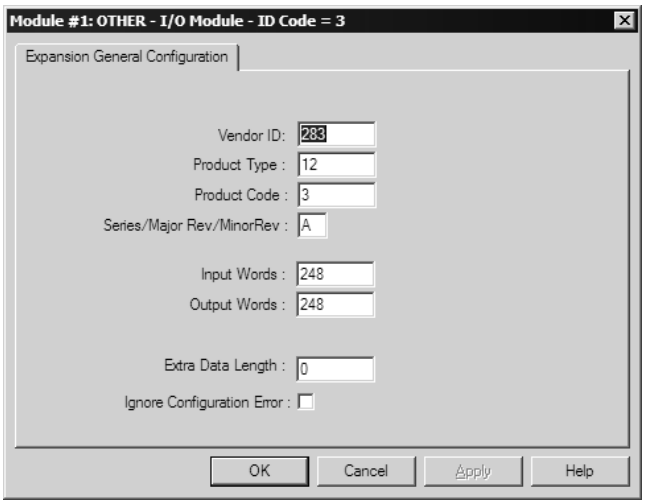
The following dialog box appears for the selection of the new Module.



On the left side of the dialog set the focus to the slot number where the module is installed. Set the focus to **"Other - Requires I/O Card Type ID"** from the available module types and then double click on it.

2.4.2 Expansion General Configuration

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



Expansion General Configuration	Value
Vendor	283
Product Type	12
Product Code	3
Series/Major Rev/Minor Rev	A
Input Words	44 + X (X = 0 ... 204)
Output Words	8 + Y (Y = 0... 240)
Extra Data Length	0

X = Number of Words configured for the Master module (PROFIBUS input data); input size can be in the range between 44 and 248 words
Y = Number of Words configured for the Master module (PROFIBUS output data); output size can be in the range between 8 and 248 words

- Vendor ID / Product Type / Product Code for the PS69-DPM module are as shown below:
  - Vendor ID: 283
  - Product Code: 12
  - Product Code: 3
- Input Words: The input size must be at least 88 Bytes (44 Words). It must be large enough to accommodate the status information required by the module, which is 88 Bytes (44 Words) plus the number of PROFIBUS input data. You can increase the size of this area using the size of each Input module connected. The Input image starts with byte 88.

- Output Word: The output size must be at least 16 Bytes (8 Words). It must be large enough to accommodate the command information required by the module, which is 16 bytes (8 Words), and the number of PROFIBUS output data. You can increase the size of this area using the size of each Output module connected. The Output area image with byte 16.
- Configuration Size: The size for the configuration array must be always 0.

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

Click **OK** to end the I/O configuration of the module.

2.4.3 Generic Extra Data Config

No generic extra data is necessary for the PS69-DPM module. The configuration is carried out with the configuration tool PROSOFT.fdt (SYCON.net) which is described in the section below.

2.5 Back Up the Project

In this step, you will create a backup copy of your project and configuration files. The backup procedure saves your data for reuse on another machine, or allows you to restore your data in the event of a system failure. The configuration data will be saved in an .XML file called *PROSOFT.fdt.xml* file. This file will reside in a sub-folder in the same directory where the *PROSOFT.spj* file is stored. The sub-folder name will be the same as the .spj file name.

To save your configuration files

1. Save your *PROSOFT.fdt* configuration, showing your Master and all slaves using the **SAVE AS** option and call the file *PS69_DPM_My_Backup*.
2. Go to the folder where the main file, *PS69_DPM_My_Backup.spj*, is located and look for a folder with the name *PS69_DPM_My_Backup*.
3. Look in any sub-folder of that folder for a file named *PROSOFT.fdt.xml*. Make a note of the sub-folder name. It will be *_S129,* or another similar name.

If you have followed the previous steps in order, your PS69 module is now configured with the settings for your PROFIBUS Master and Slaves.

3 Communication

In This Chapter

3.1 I/O Communication and I/O Memory Map

This section contains the I/O memory mapping for the PS69-DPM module. The I/O area will be used for communicating status and command information as well as cyclic I/O.

3.1.1 I/O Arrays Overview

Input Array

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 to 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 to 13	Global State Field	Bus_err_cnt
14 to 15	Global State Field	Time_out_cnt
16 to 23	Global State Field	Reserved[8]
24 to 39	Global State Field	Sl_cfg[128]

Offset	Register Type	Name
40 to 55	Global State Field	SI_state[128]
56 to 71	Global State Field	SI_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 to 79	Slave Diagnostic Field	Ident Number
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 to 495	PROFIBUS Input Area	Inputs (408 Bytes)

Output Array

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 to 495	PROFIBUS Output area	Outputs (480 Bytes)

3.1.2 Input Array

Device Status Registers

The PS69-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

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

- 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 has occurred. Further diagnostic is carried out with the PROSOFT.fdt (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	Structure Member	Data Type	Description
0	HsAck0	BOOL	SlvDiagCnf ,Slave Diag Confirmation
1	HsAck1	BOOL	GlbCtrCnf , Global Control Confirmation
2	HsAck2	BOOL	Reserved
3	HsAck3	BOOL	Reserved
4	HsAck4	BOOL	Reserved
5	HsAck5	BOOL	Reserved

Bit Offset	Structure Member	Data Type	Description
6	HsAck6	BOOL	Reserved
7	HsAck7	BOOL	Reserved

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.

Firmware Revision

This data field, which is part of the input image of the CompactLogix PROFIBUS Master module, contains 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 following table.

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

Example:

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

Due to a different 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 following table. 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.

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

Global State Field

The 64 byte Global State Field is available to the user program via the input area of the master module. 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 following table.

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 to 13	Bus_err_cnt	INT	Bus Error Counter	Number of major bus errors.
14 to 15	Time_out_cnt	INT	Time Out Counter	Number of bus time outs.
16 to 23	Reserved[8]	SINT(8)	Reserved	Reserved 8 Bytes
24 to 39	Sl_cfg[128]	BOOL(128)	Slave Configuration 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 to 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 to 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

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

ERROR-EVENT Codes for an Err_Rem_adr of 255

Code	Indication	Source	Corrective Action
0	No errors	None	None.
50	USR_INTF-Task Not Found	Master	Firmware is invalid. Module must be updated.
51	No Global Data	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.

Code	Indication	Source	Corrective Action
56	Non Existing Slave Parameters	Configuration	Execute download of configuration database again.
57	Faulty Parameter Value in a Slave Parameters	Configuration	Check GSD file for possible incorrect slave parameterization values.
58	Duplicate Slave Address	Configuration	Check configured slave addresses in project.
59	Projected Send Process Data Offset Address of a Slave is Outside the Limits of 0 to 255	Configuration	Check slave configuration in active project.
60	Projected Receive Process Data Offset Address of a Slave is Outside the Limits of 0 to 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 Available	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

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, 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.

Code	Indication	Source	Corrective Action
18	Master not in 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.

Slave Diagnostics Field

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

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 below 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 only.

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 to 79	Ident Number	INT	PROFIBUS Ident number from Slave in which the diagnostic request was made

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

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 following table.

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

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

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

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 following table.

Byte Offset	Member Name	Data Type	Description
80	Alarm_Status	SINT	Status of Alarm Pending
81	AlarmCnt	SINT	Alarm Counter
82	Rem_Add	SINT	Address of Slave with Alarm (0 to 126)
83	Slot_Number	SINT	Slot Number (0 to 254)

Byte Offset	Member Name	Data Type	Description
84	Seq_Nr	SINT	Sequence Number (0 to 31)
85	Alarm_Type	SINT	Alarm Type (1 to 6, 32 to 126)
86	Alarm_Spec	SINT	Alarm Specification (0 to 7)
87	Reserved	SINT	Reserved

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

The alarm indication registers start with the byte "Alarm_Status". This byte is a collection of bits to indicate alarm status. Bit D0 (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", and so on. 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 Bit 0 will be set to "0" indicating that the alarm has been acknowledged and is no longer pending. The alarm information Rem_Adr, Slot_Number, and so on. is not cleared. It can happen that a second or more alarms are pending. In this case bit D0 will not be reset to "0" when the application has responded to one alarm. Only the alarm information Rem_Add, Slot_Number and so on. 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 more slaves. The master module has a buffer for only 32 alarms. The application has to respond as fast as possible to alarms. If the internal alarm buffer runs over, further alarms are lost. This situation is indicated by the "Alarm_Status" bit D7 "AlarmOverrun". If an alarm was loss, this bit is set to "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.

PROFIBUS Input Data

The remainder of the input area is used for PROFIBUS input data from connected Slaves. The input information is transferred from the module to the controller. Input data from the PROFIBUS system always starts at the 88th Byte (based on start index 0) in the input region. Thus, the module has a maximum of 408 bytes input data (496 byte input region – 88 byte status). The input data of the Slaves are linear corresponding to the I/O Mapping assigned by PROSOFT.fdt (SYCON.net). PROSOFT.fdt (SYCON.net) is capable of configuring more than 408 Bytes of input data. Should the input data of the system exceed 408 bytes, only the first 408 bytes of the input data will be transferred to the controller.

3.1.3 Output Array

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. The following tables describe 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

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

NRDY := Not Ready

With this bit, the user program can start or stop communication with the PROFIBUS system. When this bit is set, the communication between the master module and all Slave devices is stopped. All slaves go to fail safe mode and the Master goes to Stop mode. This control bit allows the user program to make a controlled start of the communication with the PROFIBUS system.

INIT := Init

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 go to fail safe mode.

HSR := Handshake Request Bits

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

With the handshake request bits the user application can trigger different functions supported by the master module. Every handshake request bit (HsReq) has a corresponding handshake acknowledge bit (HsAck) in the Device Status Register in theDevice Status Registers (page 61). 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.

Note: The user application can only initiate a new command as long as the *HsReq* bit and the *HsAck* bit are equal.

HsReq0 := Slave Diag Request

The user program can use this bit to execute a Slave Diagnostic request. This bit is used with the SlvDiagCnf bit in the Handshake Acknowledge bits to determine if the command has been processed. Refer to Device Status Registers for more Input: DPM_DEV_STATUS_REGISTER (page 110).

HsReq1 := Global Control Request

The user program can use this bit to execute a Global Control command. This bit is used with the GlbCtrCnf bit to determine if the command has been processed. See section on Device Status Registers for more Input: DPM_DEV_STATUS_REGISTER (page 110).

HsReq0..7 := Reserved

Reserved for future use.

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 diagnostic command can be executed by using the corresponding handshake request bit HsReq0 in the Device Command Register (page 69). The result of the slave diagnostic command can be read in the Slave Diagnostic field of the input array.

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

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.

Global Control Array

The Global Control array is a field 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	Structure Member	Data Type	Description
8	SLA	SINT	Slave Address
9	CC	SINT	Function
10	GS	SINT	Reserved
11	Reserved	SINT	Reserved

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 affected by this command at the same time.

CC := Control Command

The parameter CC determines the function when using the Global Control Command.

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

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

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 explanation 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

PROFIBUS Output Data

The remainder of the output area is used for PROFIBUS output data for 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 bytes 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 PROSOFT.fdt (SYCON.net). PROSOFT.fdt (SYCON.net) is able to configure more than 480 bytes of output data. Should the output data be greater, the controller will only use the first 480 bytes.

3.2 CIP Messaging

PROFIBUS-DP supports acyclic services through messages. These PROFIBUS-DP services are supported by the RSLogix 5000 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.

Note: Up to time of release of the PS69-DPM module not all of the MicroLogix 1500 controller support CIP messaging or CIP messaging for generic Compact I/O modules. That's why CIP messaging, which means PROFIBUS DPV1 services, are not yet supported with a MicroLogix 1500 System and a PS69-DPM module.

3.2.1 Using the MSG Instruction in RSLogix 5000

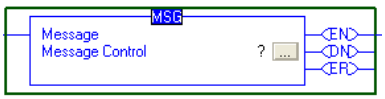
CIP messages are possible by the use of the "MSG" function block in RSLogix 5000. The MSG function block can be found under Input/Output Instructions within the RSLogix Instruction Set. The MSG instruction asynchronously reads or writes a block of data to another module on a network. The following is an example of how this instruction is assembled using the acyclic PROFIBUS-DP service DDLM_Slave_Diag command.

Step 1: 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.

Step 2: Insert the "MSG" instruction

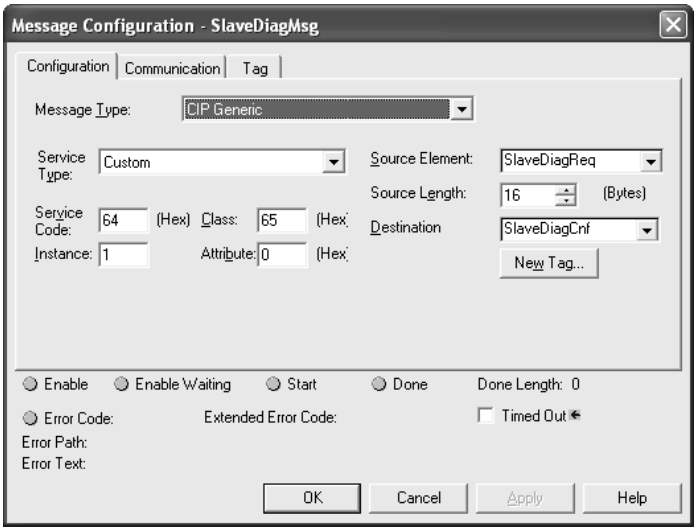
From the language element tool bar in RSLogix select the Input/Output tab and click on the "MSG" button. The instruction will be inserted into your ladder logic as shown in the figure below.



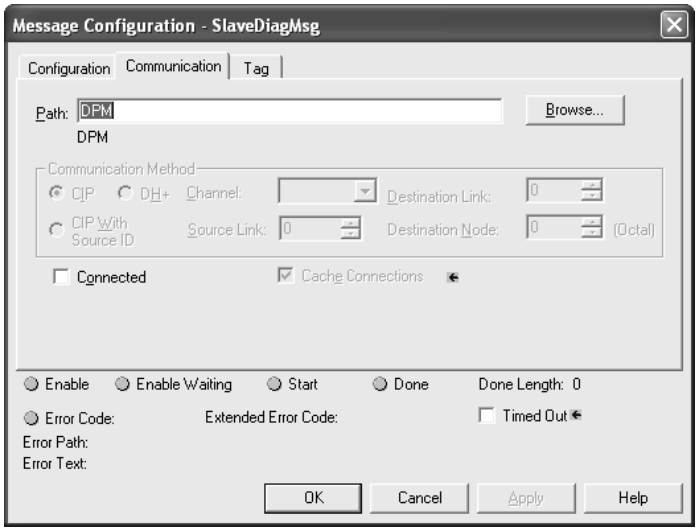
Select the "?" and enter the MESSAGE type created SlaveDiagMsg as shown below.

Step 3: Message Configuration

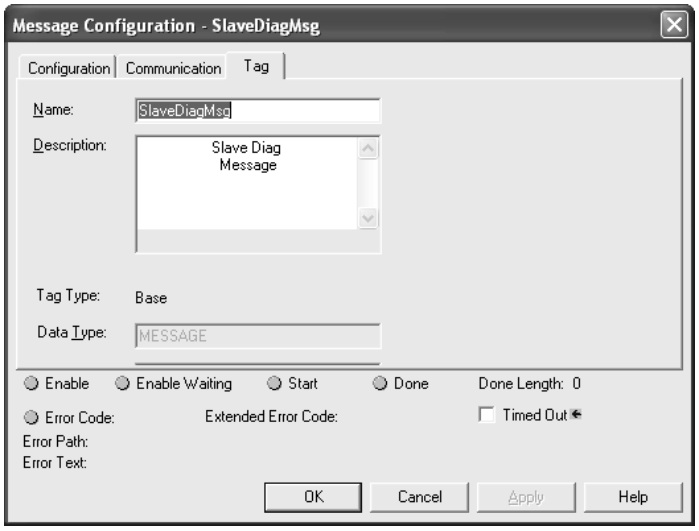
Select the button ... and open the Message Configuration dialog box. The configuration dialog will allow the user to input the appropriate information needed to execute the SlaveDiagMsg. The entries should be as follows.



Note: You 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.

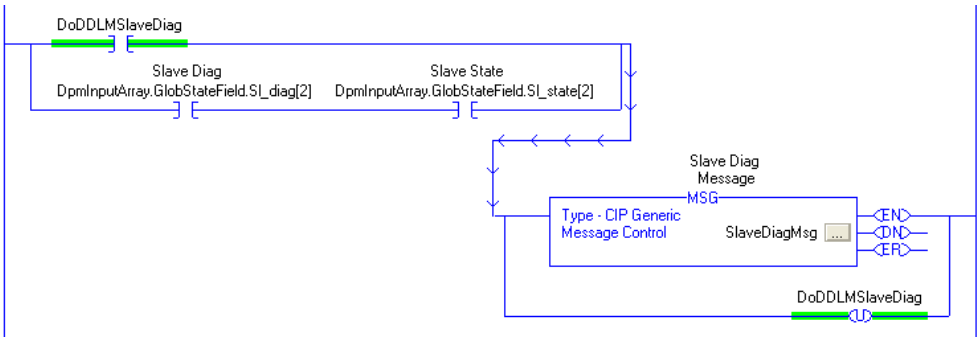


The path in the dialog above must point to the 1769-Module. Use the **Browse** button to select the path.



Step 4: Add Logic to Execute MSG Instruction

With the MSG instruction now configured you 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.



3.2.2 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.

Note: The sample program "RIF_1796_DPM_messaging_L32E.ACD" has an example for each of these services.

3.2.3 Standard Messaging

This section contains the description of each Standard Message "DDLM" supported by the PROFIBUS Master module.

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

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

MA := Master Address

This byte contains the address of the Master that has configured 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 following table.

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

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

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

The CIP MSG setup 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

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. 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

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 affected 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	Structure Member	Data Type	Description
0	Reserved	BOOL	Reserved
1	Clear_data	BOOL	Clear Output Data
2	UnFreeze	BOOL	Unfreeze Input Data
3	Freeze	BOOL	Freeze Input Data
4	Unsync	BOOL	Neutralize Sync Command or Unsync

Bit Offset	Structure Member	Data Type	Description
5	Sync	BOOL	Freeze Output Data, Until Sync Command is Neutralized
6	Reserved	BOOL	Reserved
7	Reserved	BOOL	Reserved

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

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

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

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 PROSOFT.fdt (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 creates 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, and so on). The parameters come from the internal configuration of the Master through the PROSOFT.fdt (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

Note: The byte array UsrPrm[234] can be made smaller if needed. This saves memory. The value 234 is 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

CIP Messaging Parameters for DDLM_SET_Parameter

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

3.2.4 DPV1 Messaging

This sections describes DPV1 messaging functions supported by the PROFIBUS Master module.

ATTENTION: Do not configure DPV1 services, if your controller does not allow CIP messaging.

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

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

CIP Message Parameters for MSAC1_Read

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

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

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

CIP Message Parameters for MSAC1_Write

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

MSAL1M_Alarm_Res

The MSAL1M_Alarm_Res request is used by the master to send a DPV1 Alarm response to a slave device. The message acknowledges the alarm when the appropriate indication appears in theDPV1 Alarm Indication (page 67) area. The information mapped to this area must be used in the DPV1 Alarm Response message in order to process the alarm properly. The mapping of this 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 to 6,32 to 126	Alarm Type from DPV1 Alarm Indication Register.
AlarmSpec	SINT	0..7	Alarm Spec from DPV1 Alarm Indication Register.
Reserved5	SINT	0	Reserved

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 to 6,32 to 126	Alarm Type from DPV1 Alarm Indication Register.
AlarmSpec	SINT	0..7	Alarm Spec from DPV1 Alarm Indication Register.
Reserved5	SINT	0	Reserved

CIP Message for MSAL1M_Alarm_Res

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

3.2.5 Messaging Error Codes

This 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.

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.

CIP Status	Extended Status	Meaning	Cause	Help
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 does not exist within the module.	
FE hex	0000 hex	Message Timeout	No answer message was received.	
FF hex	0514 hex	General Error Non specified error occurred		Call support
FF hex	0517 hex	General Error Unknown command	The value in Req.Command is unknown	The value Req.Command must be initialized

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

DDLM_Global_Control

The DDLM_Global_Control command initiates a multicast command on the PROFIBUS network to all configured slaves. Therefore, this command is always assumed to be 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		

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

Failure	Significance	Error source	Help
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

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 PROSOFT.fdt (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

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 configuration 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

4 RSLogix Example Program

In This Chapter

There are three example ladder logic programs on the ProSoft Solutions DVD. Two projects are for a CompactLogix system and one for a MicroLogix 1500 system. These examples can 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 Controller, change the Controller Type 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 PS69-DPM.

Sample Project	Controller Type	RSL5K Version	Description
PS69_DPM_L32E.acd	1769-L32E	V13	Basic I/O example
PS69_DPM_Messaging_L32E.acd	1769-L32E	V13	Basic messaging example

4.1 CompactLogix I/O Example

This ladder logic program is a basic example for the setup of the PROFIBUS-DP Master communications module "PS69-DPM" in RSLogix 5000. 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 called 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. Their definitions and the structures included in each are shown in RSLogix 5000 User Defined Data Types (page 110).

4.2 CompactLogix Messaging Example

This ladder logic program is a CIP messaging example for the setup of the PROFIBUS-DP master communications module "PS69-DPM" in RSLogix 5000. 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 needed 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 as MSAC1_READ_CONFIRM user defined tag. The user should make changes to this routine as needed for his 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 in 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 called 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 as MSAC1_WRITE_CONFIRM user defined tag. The user should make changes to this routine as needed for his 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. Their definitions and the structures included in each are shown in RSLogix 5000 User Defined Data Types (page 110).

5 Diagnostics and Troubleshooting

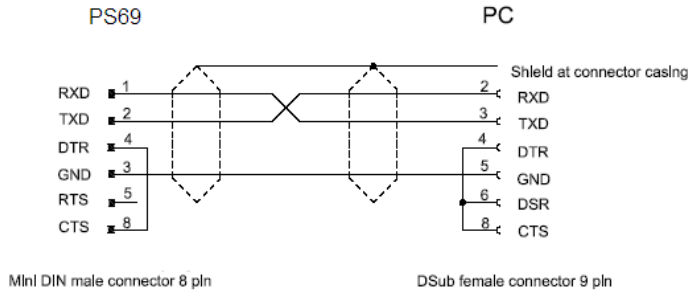
In This Chapter

This section describes the possible diagnostics and troubleshooting procedures for the PS69-DPM Master module.

5.1 Diagnostic Interface

Non isolated RS-232C interface to connect with the COM port at the PC. The ProSoft part number for the diagnostic cable is CABLE-SRV-MD8.

Connection with Mini DIN 8 pin female connector	Signal	Meaning
1	RXD	Receive Data
2	TXD	Send Data
3	GND	Signal Ground
4	DTR	Data Terminal Ready
5	RTS	Ready to Send
8	CTS	Clear to Send



5.2 Hardware Diagnostics (LED)

The following section contains a description of the LED and their meaning for both the CPU module and the PS69-DPM module.

5.2.1 CompactLogix

The following table 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

5.2.2 MicroLogix 1500

To identify problems via possible LED indications of a MicroLogix 1500 controller refer to the MicroLogix 1500 User Manuals section "Troubleshooting Your System". Here you will find a detailed description of fault indications and possible reasons.

5.2.3 PS69 LEDs

The LEDs as shown on the front panel will be used to indicate status information of the PS69-DPM Master module. Each LED has a specific function during Run, configuration download, and error indications. The following table 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.

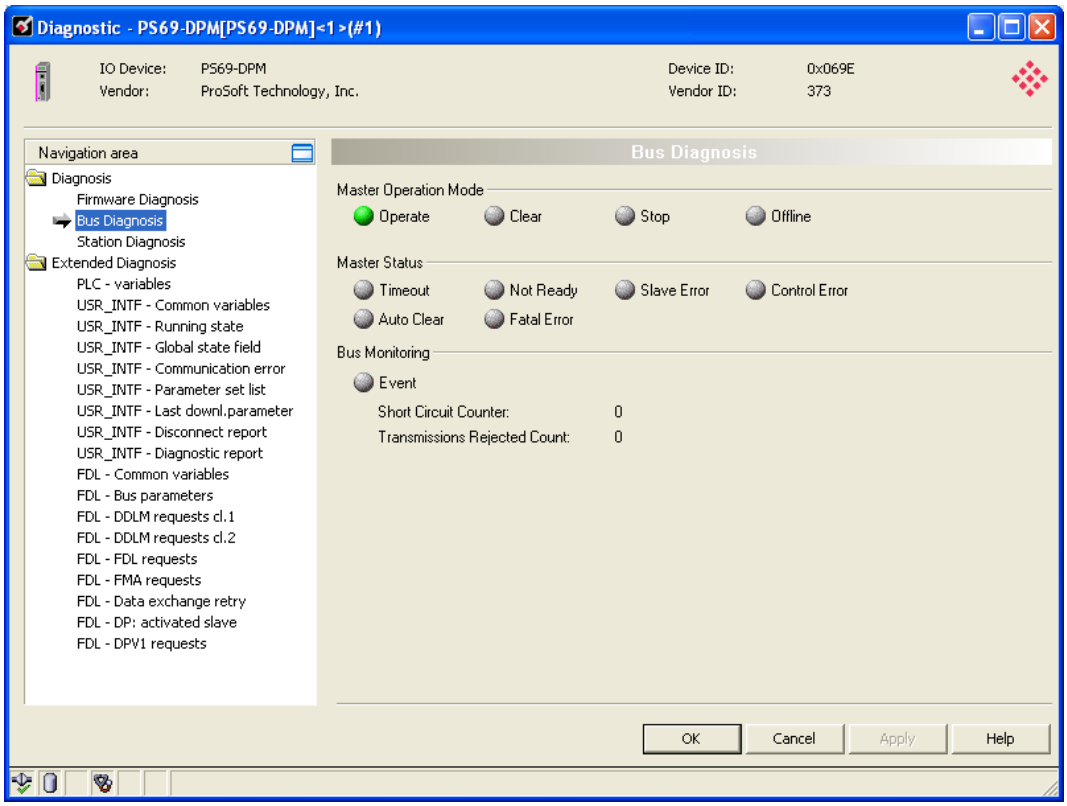
(*) 3 times fast at 5 Hz, 8 times between 0,5Hz and 1Hz
(**) between 0,5Hz and 100Hz

5.3 PROSOFT.fdt (SYCON.net) Diagnostics

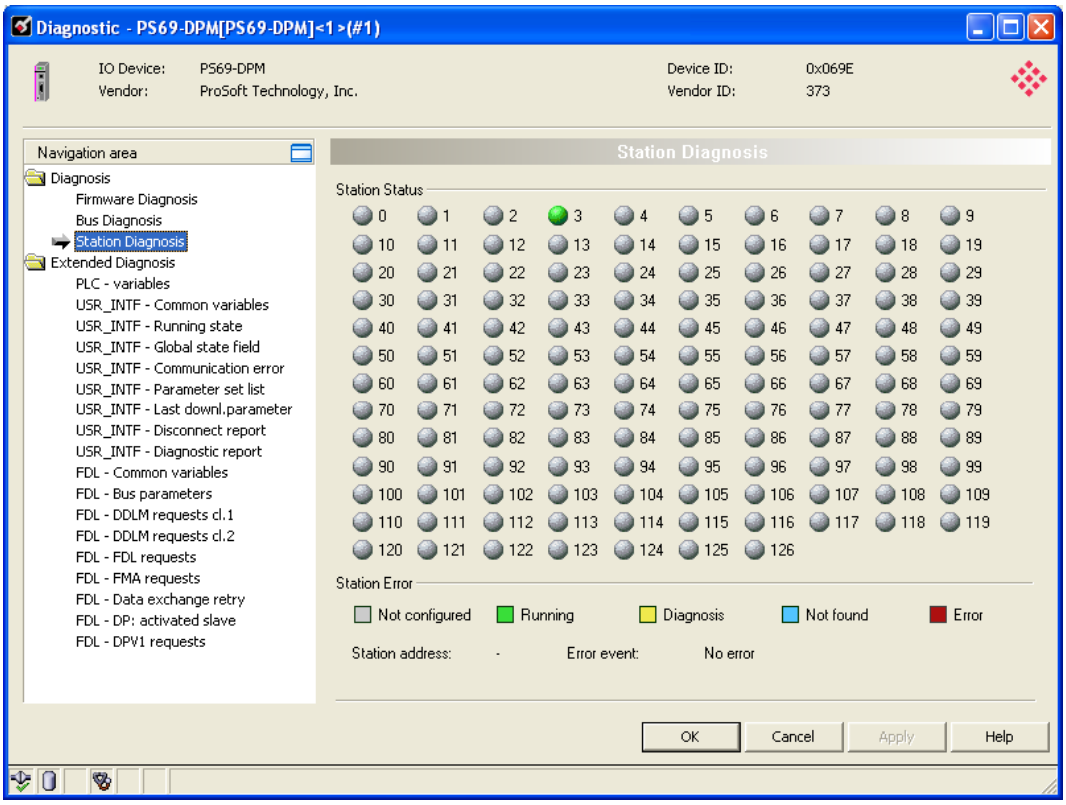
PROSOFT.fdt (SYCON.net) provides Master or individual Slave diagnostics, as described in the following sections.

5.3.1 Master Diagnostics

PROSOFT.fdt (SYCON.net) allows you to monitor the operating conditions of the PROFIBUS Master via the serial port. To use this diagnostic functionality open your project in PROSOFT.fdt (SYCON.net). In the PROSOFT.fdt (SYCON.net) project screen, right click on the Master in either the Network View bus or the netProject screen. From the context menu, select **Connect**. The text above the Master should be highlighted in Green. Double click on the Master in either the Network View bus or the netProject screen. The following diagnostic dialogs should appear.



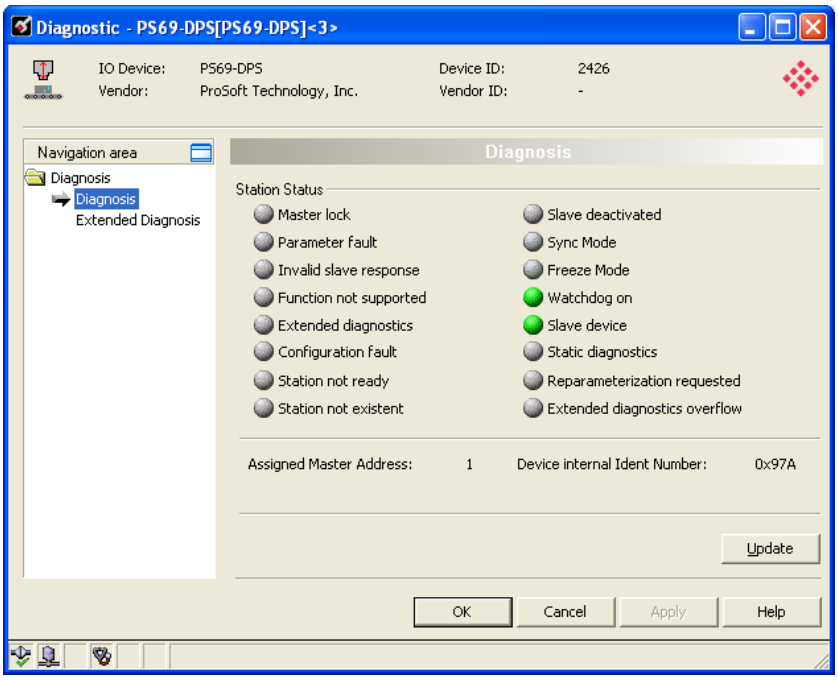
Select Station Diagnosis in the Navigation area to view status for each station connected to the PS69-DPM Master.



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 PROSOFT.fdt (SYCON.net) and the Operating Instruction Manual for **ProSoft DTM for PROFIBUS Master Devices.**

5.3.2 Slave Diagnostics

PROSOFT.fdt (SYCON.net) provides a way to monitor the operating conditions of each individual PROFIBUS Slave via the serial port. To use this diagnostic functionality open your project in PROSOFT.fdt (SYCON.net). In the PROSOFT.fdt (SYCON.net) project screen, right click on the Slave you have added in either the Network View 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 Network View bus or the netProject screen. The following diagnostic dialog should appear.



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 PROSOFT.fdt (SYCON.net) and the Operating Instruction Manual for Generic DTM for PROFIBUS-DP Slave Devices.

5.4 Troubleshooting

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

5.4.1 CompactLogix I/O LED

Communication between the module and controller is displayed via the I/O LED of the Controller. The proper 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 has been established.

5.4.2 MicroLogix 1500 Fault LED

The faultless communication state is reached, if the Fault LED of the MicroLogix Controller is in off state.

If there is a problem with the expansion module the Fault LED is flashing red. Then go online with your RSLogix 500 project and open up the processor status dialog and check the error tab for the fault reason.

5.4.3 SYS and COM Status LEDs

This PS69-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 be yellow or green. The **COM** LED displays the status of the PROFIBUS communication. It can flash green or red. The meaning of the LEDs is described in the booklet that came with the System Software CD and in CompactLogix I/O LED (page 100).

If the SYS LED is solid green and the COM-LED static green, the card is in cyclic data exchange with the connected Slaves and the communication is running with out fault.

5.4.4 Error Sources and Reasons

This chapter describes typical problems, error sources and questions that come up while commissioning the PROFIBUS-DP master module PS69-DPM. The following table summarizes the typical error sources and gives a hint of possible reasons for the problem.

Behavior	Significance	Typical Reason	Help
CompactLogix I/O LED is Green flashing	No communication with the PS69 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
MicroLogix Fault LED is flashing Red	No communication with the PS69 module (or other modules)	Modules slot number in RSLogix program does not match with the physical slot of the module Configured Vendor ID / Module ID / Input / Output / Configuration array size is wrong	Check modules slot number in RSLogix project Compare configured Input / Output size / Vendor ID /Module ID with required values
PS69-DPM COM LED is off SYS LED Flashing irregular green	Configuration missing or faulty	No configuration stored	Download a Configuration to the card with PROSOFT.fdt (SYCON.net)
PS69-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

Behavior	Significance	Typical Reason	Help
PS69-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 and so on. Use PROSOFT.fdt (SYCON.net) diagnostic to find the wrong slave
PS69-DPM COM LED is static red and SYS LED flashing cyclic fast green	No communication to any slave	PROFIBUS cable not connected No slaves connected PLC is not in RUN mode	PROFIBUS wiring Check if slaves are connected Check if PROFIBUS configuration matches with physical configuration Bring PLC into in RUN mode
Slave input data can not be found in RSLogix program	Input array mismatch	Configured input size in RSLogix too small Configured input address table in PROSOFT.fdt (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 PROSOFT.fdt (SYCON.net) configured PROFIBUS input array Check if configured Input address table in PROSOFT.fdt (SYCON.net) matches with PLC program
Outputs are not transferred to slave although PROFIBUS is running	Output array mismatch	Configured output size in RSLogix too small Configured output address table in PROSOFT.fdt (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 PROSOFT.fdt (SYCON.net) configured PROFIBUS output array Check if configured Output address table in PROSOFT.fdt (SYCON.net) matches with PLC program
The serial device assignment dialog in PROSOFT.fdt (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 PROSOFT.fdt (SYCON.net) shows error - 51 to the wished serial COM port	Module does not answer to PROSOFT.fdt (SYCON.net) request	Diagnostic cable not connected	Check if the module and the PC are wired correctly with the diagnostic cable
PROSOFT.fdt (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

5.4.5 Cable

- Check that the cable is Constructing a Bus Cable for PROFIBUS DP (page 119).
- Check to confirm that the bus termination resistors are switched on at the beginning and the end of the cable and switched off at all other connectors in between.

6 Reference

In This Chapter

6.1 Specifications

The PS69 PROFIBUS DP Master module expands the functionality of Rockwell Automation's CompactLogix or MicroLogix family of processors to include PROFIBUS. This module supports both I/O control and messaging, thus integrating DPV0 and DPV1 functionality into CompactLogix or MicroLogix.

The PS69-DPM is a CompactLogix or MicroLogix compatible, PROFIBUS certified module, which enables controllers to communicate with a PROFIBUS network. The configuration and diagnostic of the PROFIBUS system is done via the serial diagnostic interface of the module using the System Configuration tool PROSOFT.fdt (SYCON.net). The data exchange between controller and module is done via the I/O process data image using CompactLogix or MicroLogix backplane technology.

DPV0 Services Supported

- Fail Safe Mode
- Global Control
- Sync and Freeze
- Watchdog

DPV1 Services Supported

- Read Request
- Write Request
- Alarm Indication
- Extended Device Diagnostics

6.1.1 General Specifications

- Single Slot – 1769 backplane compatible
- The module is recognized as an Input/Output module and has access to processor memory for data transfer between processor and module
- Ladder Logic is used for data transfer between module and processor. Sample ladder file included.
- Supports CompactLogix processors with 1769 I/O bus capability.
- Also supports MicroLogix 1500 LRP

6.1.2 PROFIBUS Interface

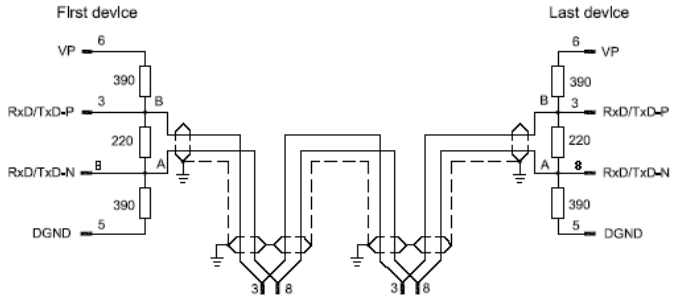
Isolated RS-485 interface per EN 50170.

3	RxD/TxD-P	Receive/Send Data-P respectively connection B plug
5	DGND	Reference potential
6	VP	Positive power supply
8	RxD/TxD-N	Receive/Send Data-N respectively connection A plug

Please ensure that termination resistors are available at both ends of the cable. If special PROFIBUS connectors are being used, these resistors are often found inside the connector and must be switched on. For baud rates above 1.5 MBaud use only PROFIBUS connectors, which also include additional inductance. It is not permitted to have T stubs at high baud rates.

Use only a special cable which is approved for PROFIBUS-DP. Make a solid connection from the cable shield to ground at every device and make sure that there is no potential difference between the grounds at the devices.

If the PS69 is linked with only one other device on the bus, both devices must be at the ends of the bus line. The reason is that these devices must deliver the power supply for the termination resistors. Otherwise the Master can be connected at any desired position.



6.1.3 Functional Specifications

Specification	Description
Slaves	max. 125
Input/Output	max. 244 Bytes per Slave
Input Data	max. 408 Bytes (*)
Acyclic Data	DPV1
Status data	88 Bytes
Output data	max. 480 Bytes (*)
Command data	16 Bytes
DPV1 services	Read / Write class 1, Alarm
Services	Global-Control, Slave-Diag, Set-Parameter
Sync, Freeze Command	supported

Up to 32 PROFIBUS devices can be connected to one bus segment. If several bus segments are linked to each other with repeaters, there can be up to 127 devices on the network.

The maximum length of a bus segment depends on the baud rate used. Only PROFIBUS certified cable, preferably the cable type A, should be used.

Baud rate in kBit/s	Max. Distance
9.6	1,200 m / 4000 ft
19.2	1,200 m / 4000 ft
93.75	1,200 m / 4000 ft
187.5	1,000 m / 3280 ft
500	400 m / 1300 ft
1,500	200 m / 650 ft
3,000	100 m / 325 ft
6,000	100 m / 325 ft
12,000	100 m / 325 ft

Parameter	Value
Impedance	135 to 65 Ohm
Capacity	< 30 pF/m
Loop resistance	110 Ohm/km
Wire gauge	0.64 mm

6.1.4 Hardware Specifications

Specification	Description
Processor	EC1-160P with integrated ASPC2
PROFIBUS Interface	RS-485, max. 12 MBaud, potential free, according EN 50170
Diagnostic Interface	RS232, PS/2 Mini DIN female connector, 9600 Baud, non isolated
Power Supply	+5 V ±5 % / 260 mA
Max. Distance Rating	max. 6 modules to the power supply module
Dimensions	Standard 1769 Single-slot module
Burst	EN 61000-4-4, 2 kV, 5 kHz
Surge	EN 61000-4-5, 2 kV common mode, 1 kV differential mode
ESD	EN 61000-4-2, 4 kV contact, 8 kV air, 4 kV indirect
Radiated/ Conducted Immunity	EN 61000-4-3, 10 V/m, 30...1000 MHz, 80% AM, 1 kHz sinewave EN 61000-4-6, 10 V, 0,15...30 MHz
Radiated/ Conducted Emission	EN 55011 Class A
Vibration/Shock	IEC 600068-2-6, 10-150 Hz, ± 0,75 mm, ± 1 g, 1 Octave/min IEC 600068-2-27, 15 g, 11 ms
Operating Temp.	0 to 60°C (32 to 140°F)
Relative Humidity	5 to 95% (non-condensing)

Specification	Description
Agency Certification: UL/CE	C-UL certified, UL 508 listed, CE
PROFIBUS conformance	certified

6.2 PROFIBUS Functionality

6.2.1 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.

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".

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 PROSOFT.fdt (SYCON.net).

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.

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.

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 PROSOFT.fdt (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".

6.2.2 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.

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.

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.

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.

6.2.3 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.

6.3 RSLogix 5000 User Defined Data Types

The section contains the user defined data types created and used in the example programs.

6.3.1 Input: DPM_INPUT_ARRAY

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]	

6.3.2 Input: DPM_DEV_STATUS_REGISTER

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	

6.3.3 Input: DPM_FW_REVISION

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

6.3.4 Input: DPM_GLOBAL_STATE_FIELD

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

6.3.5 Input: DPM_SLAVE_DIAG_DATA

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

Name	Data Type	Description
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
IdentNumber	INT	PROFIBUS Ident number

6.3.6 Input: DPM_DPV1_ALARM_INDICATION

Name	Data Type	Description
AlarmIndication	BOOL	Indication 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	

6.3.7 Output: DPM_OUTPUT_ARRAY

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]	

6.3.8 Output: DPM_DEV_COMMAND_REGISTER

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 Start, not supported)
Reset	BOOL	Reset (Cold Start)
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

6.3.9 Output: DPM_SLAVE_DIAG_COMMAND

Name	Data Type	Description
SlaveAddress	SINT	Address of Slave
Reserved1	SINT	

6.3.10 Output-DPM_GLOBAL_CONTROL_COMMAND

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

6.3.11 DDLM_GLOBAL_CONTROL_REQUEST

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

6.3.12 DDLM_GLOBAL_CONTROL_CONFIRM

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

6.3.13 DDLM_SET_PARAMETER_REQUEST

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

6.3.14 DDLM_SET_PARAMETER_CONFIRM

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

6.3.15 DDLM_SLAVE_DIAGNOSTIC_REQUEST

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	

6.3.16 DDLM_SLAVE_DIAGNOSTIC_CONFIRM

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]	

6.3.17 DPM_DPV1_ALARM_INDICATION

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	

6.3.18 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	

6.3.19 MSAC1_READ_REQUEST

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	

6.3.20 MSAC1_READ_CONFIRM

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]	

6.3.21 MSAC1_WRITE_REQUEST

Name	Data Type	Description
Reserved1	INT	
Reserved2	INT	
Reserved3	INT	
Command	SINT	
Reserved4	SINT	
DeviceAdr	SINT	

Name	Data Type	Description
DataArea	SINT	
DataAdr	INT	
DataIdx	SINT	
DataCnt	SINT	
DataType	SINT	
Function	SINT	
data	SINT[240]	

6.3.22 MSAC1_WRITE_CONFIRM

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	

6.3.23 MSAL1M_ALARM_RESPONSE

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	

6.3.24 MSAL1M_ALARM_CONFIRM

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	

6.4 Constructing a Bus Cable for PROFIBUS DP

The bus cable for connecting PROFIBUS DP devices must be constructed by the user. A special PROFIBUS cable (twisted pair) is required here. This standard cable is available from various manufacturers and is a Belden part number 3079A.

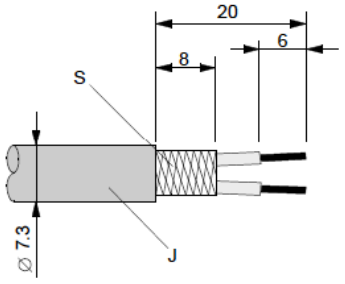
If you plan to construct your own bus cable, the following part numbers are provided for your convenience.

- PROFIBUS connector: Siemens part number 6ES7972-0BA40-0XA0
- PROFIBUS cable: Belden part number 3079A.

To construct the cable, proceed as follows:

1 Cut the cable to the required length.

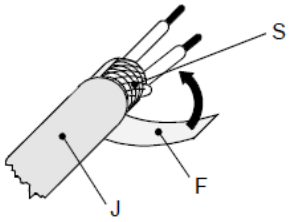
2 Prepare the cable ends as shown in the illustration (dimensions in mm):



J PVC Jacket
S Braided shielding

3 Remove the PVC jacket J to the indicated length.

4 Wrap the provided copper shielding F around the shield braiding S:



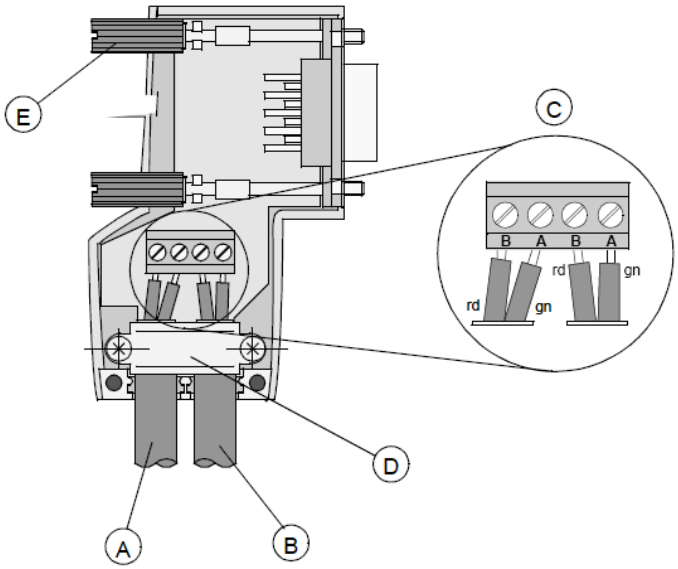
J PVC jacket
S Braided shielding
F Copper foil shielding
Additional foil can be obtained from 3M.

5 Plug the leads of the corresponding cable(s) into the terminals as shown:

- Green leads in terminal A
- Red lead in terminal B

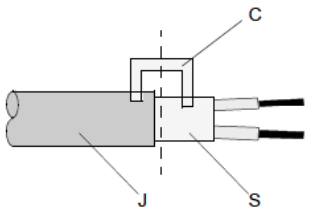
▪ **Note:** Do **not** tighten the corresponding screws yet.

Connection terminal assignment on the PROFIBUS DP:



A Incoming cable
B Outgoing cable
C Connection terminals (only once (B,A))
D Cable cleat for reliving tension
E Bus connector screws

6 Attach the cables with the provided cable cleat to create a robust shielded connection and to relieve any tension as shown:



J PVC Jacket
S Braided shielding with foil shielding
C Cable cleat

- **Note**: Half of the cable jacket must lie under the cable cleat!

Pay attention to the cable cleat installation instructions.

7 Fasten the individual wires of the PROFIBUS cable to the terminals
8 Close the connector housing.

- **Note:** The shielding of both cables is connected internally with the metal housing of the connector.

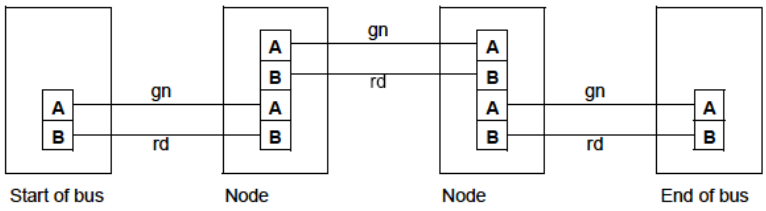
9 Complete the Central Shielding Measures (below) and grounding operations for the shielding before you connect the cable connector to the module.
10 Plug the PROFIBUS DP connector into the module and secure it with the screws.

Bus Begin and Bus End

The PROFIBUS connector with termination is required at the beginning and the end of the bus. These connectors emulate the line impedance.

It is recommended that at least one connector with diagnostics interface is used.

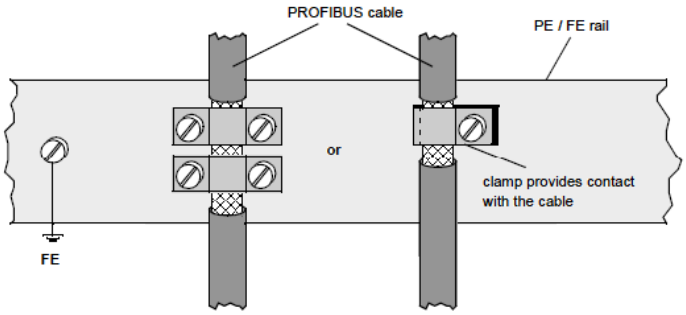
Wiring diagram for a PROFIBUS DP cable



Grounding and Shielding for Systems with Equipotential Bonding

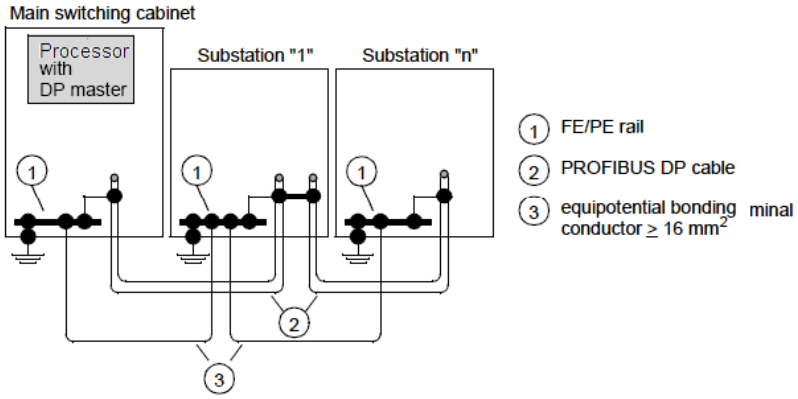
Each cable shield should be galvanically grounded with the earth using FE/PE grounding clamps immediately after the cable has been connected to the cabinet.

This example indicates the shielding connection from the PROFIBUS cable to the FE/PE rail.



Note: An equalization current can flow across a shield connected at both ends because of fluctuations in ground potential. To prevent this, it is imperative that there is potential equalization between all the attached installation components and devices.

This example indicates the system components and devices in a system with equipotential bonding.

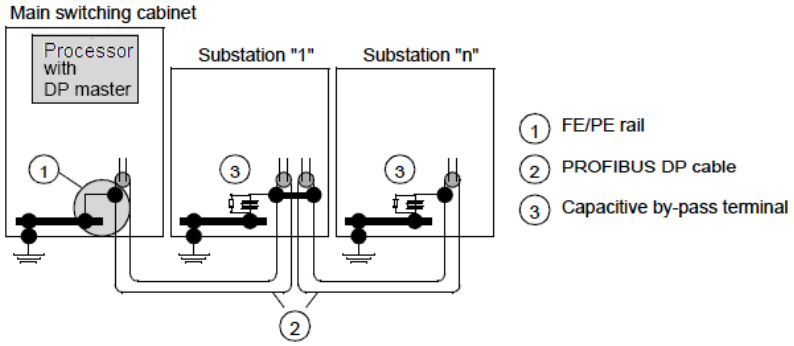


Grounding and Shielding for Systems without Equipotential Bonding

Note: Grounding and shielding is to be carried out the same as for systems **with** equipotential bonding.

If this is not possible because of system or construction specific reasons however, use distributed ground with a capacitive coupling of high frequency interference signals.

This representation shows distributed grounding with capacitive coupling.



7 End-User License Agreement - PROSOFT.fdt (SYCON.net) Software

In This Chapter

7.1 PROSOFT TECHNOLOGY END-USER LICENSE AGREEMENT

This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and ProSoft Technology. Please read this Agreement carefully before you install the software. By installing or otherwise using the software, you accept the terms of this Agreement. If you do not agree to the terms of this EULA, do not install, copy, or use the SOFTWARE PRODUCT.

7.2 SOFTWARE COPYRIGHT PRODUCT LICENSE

This software program (the "Software") is owned (or governed) by ProSoft Technology (and is under distribution agreement with partner software development company). Copyright laws and international copyright treaties, as well as other intellectual property laws and treaties protect the SOFTWARE PRODUCT. The SOFTWARE PRODUCT is licensed, not sold.

7.3 GRANT OF LICENSE

This EULA grants you the following non-exclusive rights:

1 You may install and use the enclosed SOFTWARE PRODUCT on your computer to design, develop, and test configuration files for use with the ProSoft Technology products.

2 You may not reverse-engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that applicable law notwithstanding this limitation expressly permits such activity.

3 You may not sell, rent or lease the SOFTWARE PRODUCT.

4 You may freely distribute the SOFTWARE PRODUCT under this EULA, provided you transfer all of the SOFTWARE PRODUCT (including all component parts, the media and printed materials, any upgrades and this EULA), and the recipient agrees to the terms of this EULA. If the SOFTWARE PRODUCT is an upgrade, any transfer must include all prior versions of the SOFTWARE PRODUCT.

5 Customer agrees that it shall not authorize others to, copy the SOFTWARE PRODUCT provided by ProSoft Technology (except as expressly permitted in any license agreement accompanying such software); transfer software to a third party separately from the SOFTWARE PRODUCT; modify, alter, translate, decode, decompile, disassemble, reverse-engineer or otherwise attempt to derive the source code of the software or create derivative works based on the software; export the software or underlying technology in contravention of applicable US and international export laws and regulations; or use the software other than as authorized in connection with use of the SOFTWARE PRODUCT.

7.4 LIMITATION OF LIABILITY

TO THE FULLEST EXTENT PERMITTED BY APPLICABLE LAW, PROSOFT AND BUYER WILL NOT BE LIABLE FOR ANY BUSINESS INTERRUPTION OR LOSS OF PROFIT, REVENUE, MATERIALS, ANTICIPATED SAVINGS, DATA, AGREEMENT, GOODWILL OR THE LIKE (WHETHER DIRECT OR INDIRECT IN NATURE) OR FOR ANY OTHER FORM OF INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND. PROSOFT'S AND BUYER'S MAXIMUM CUMULATIVE LIABILITY RELATIVE TO ALL OTHER CLAIMS AND LIABILITIES, INCLUDING OBLIGATIONS UNDER ANY INDEMNITY, WHETHER OR NOT INSURED, WILL NOT EXCEED TWO MILLION DOLLARS ($2,000,000). PROSOFT DISCLAIMS ALL LIABILITY RELATIVE TO GRATUITOUS INFORMATION OR ASSISTANCE PROVIDED BY, BUT NOT REQUIRED OF PROSOFT HEREUNDER. ANY ACTION AGAINST PROSOFT MUST BE BROUGHT WITHIN EIGHTEEN (18) MONTHS AFTER THE CAUSE OF ACTION ACCRUES. THESE DISCLAIMERS AND LIMITATIONS OF LIABILITY WILL APPLY REGARDLESS OF ANY OTHER CONTRARY PROVISION HEREOF AND REGARDLESS OF THE FORM OF ACTION, WHETHER IN AGREEMENT, TORT (INCLUDING NEGLIGENCE AND STRICT LIABILITY) OR OTHERWISE, EACH PROVISION HEREOF WHICH PROVIDES FOR A LIMITATION OF LIABILITY, DISCLAIMER OF WARRANTY OR CONDITION OR EXCLUSION OF DAMAGES IS SEVERABLE AND INDEPENDENT OF ANY OTHER PROVISION AND IS TO BE ENFORCED AS SUCH.

7.4.1 No liability for consequential damages

IN NO EVENT WILL PROSOFT TECHNOLOGY (OR ITS DEALER) BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES BASED ON BREACH OF WARRANTY, BREACH OF CONTRACT, NEGLIGENCE, STRICT TORT, OR ANY OTHER LEGAL THEORY. DAMAGES THAT PROSOFT AND ITS DEALER WILL NOT BE RESPONSIBLE FOR INCLUDE, BUT ARE NOT LIMITED TO: LOSS OF PROFITS; LOSS OF SAVINGS OR REVENUE; LOSS OF USE OF THE PRODUCT OR ANY ASSOCIATED EQUIPMENT; LOSS OF DATA; COST OF CAPITAL; COST OF ANY SUBSTITUTE EQUIPMENT, FACILITIES, OR SERVICES; DOWNTIME; THE CLAIMS OF THIRD PARTIES, INCLUDING CUSTOMERS OF THE PURCHASES; AND INJURY TO PROPERTY.

8 Support, Service & Warranty

In This Chapter

8.1 Contacting Technical Support

ProSoft Technology, Inc. is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

1 Product Version Number
2 System architecture
3 Network details

If the issue is hardware related, we will also need information regarding:

1 Module configuration and associated ladder files, if any
2 Module operation and any unusual behavior
3 Configuration/Debug status information
4 LED patterns
5 Details about the serial, Ethernet or Fieldbus devices interfaced to the module, if any.

Note: *For technical support calls within the United States, an emergency after-hours answering system allows 24-hour/7-days-a-week pager access to one of our qualified Technical and/or Application Support Engineers. Detailed contact information for all our worldwide locations is available on the following page.*

Internet	Web Site: www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Asia Pacific (location in Malaysia)	Tel: +603.7724.2080 E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Asia Pacific (location in China)	Tel: +86.21.5187.7337 x888 E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Europe (location in Toulouse, France)	Tel: +33 (0) 5.34.36.87.20 E-mail: support.EMEA@prosoft-technology.com Languages spoken include: French, English
Europe (location in Dubai, UAE)	Tel: +971-4-214-6911 E-mail: mea@prosoft-technology.com Languages spoken include: English, Hindi
North America (location in California)	Tel: +1.661.716.5100 E-mail: support@prosoft-technology.com Languages spoken include: English, Spanish
Latin America (Oficina Regional)	Tel: +1-281-2989109 E-Mail: latinam@prosoft-technology.com Languages spoken include: Spanish, English
Latin America (location in Puebla, Mexico)	Tel: +52-222-3-99-6565 E-mail: soporte@prosoft-technology.com Languages spoken include: Spanish
Brasil (location in Sao Paulo)	Tel: +55-11-5083-3776 E-mail: brasil@prosoft-technology.com Languages spoken include: Portuguese, English

8.2 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS please see the documents on the ProSoft Solutions DVD or go to www.prosoft-technology/warranty.

Documentation is subject to change without notice.

Index

N

O

P

R

S

T

U

W

Y