ProSoft TECHNOLOGY

Where Automation Connects.

MVI56-101S

ControlLogix Platform

IEC 60870-5-101 Slave
Communication Module

August 5, 2021

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.

How to Contact Us

ProSoft Technology, Inc.

+1 (661) 716-5100
+1 (661) 716-5101 (Fax)
www.prosoft-technology.com
support@prosoft-technology.com

MVI56-101S User Manual

August 5, 2021

ProSoft Technology® Product Documentation

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 at: www.prosoft-technology.com

For professional users in the European Union

If you wish to discard electrical and electronic equipment (EEE), please contact your dealer or supplier for further information.

Warning – Cancer and Reproductive Harm – www.P65Warnings.ca.gov

Agency Approvals & Certifications

Please visit our website: www.prosoft-technology.com

Important Installation Instructions

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;
WARNING - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES
WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.
THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

MVI (Multi Vendor Interface) Modules

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT - RISQUE D'EXPLOSION - AVANT DE DÉCONNECTER L'ÉQUIPEMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

Warnings

North America Warnings

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

A Warning - Explosion Hazard - Substitution of components may impair suitability for Class I, Division 2.
B Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or rewiring modules.
C Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.

Avertissement - Risque d'explosion - Avant de déconnecter l'équipement, couper le courant ou s'assurer que l'emplacement est désigné non dangereux.

D Suitable for use in Class I, Division 2 Groups A, B, C and D Hazardous Locations or Non-Hazardous Locations.

ATEX Warnings and Conditions of Safe Usage

Power, Input, and Output (I/O) wiring must be in accordance with the authority having jurisdiction.

A Warning - Explosion Hazard - When in hazardous locations, turn off power before replacing or wiring modules.
B Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be non-hazardous.
C These products are intended to be mounted in an IP54 enclosure. The devices shall provide external means to prevent the rated voltage being exceeded by transient disturbances of more than 40%. This device must be used only with ATEX certified backplanes.
D DO NOT OPEN WHEN ENERGIZED.

Battery Life Advisory

The MVI46, MVI56, MVI56E, MVI69, and MVI71 modules use a rechargeable Lithium Vanadium Pentoxide battery to backup the real-time clock and CMOS. The battery should last for the life of the module. The module must be powered for approximately twenty hours before the battery becomes fully charged. After it is fully charged, the battery provides backup power for the CMOS setup and the real-time clock for approximately 21 days. When the battery is fully discharged, the module will revert to the default BIOS and clock settings.

Note: The battery is not user replaceable.

Markings

Electrical Ratings

- Backplane Current Load: 800 mA @ 5.1 Vdc; 3 mA @ 24 Vdc
- Operating Temperature: 0°C to 60°C (32°F to 140°F)
- Storage Temperature: -40°C to 85°C (-40°F to 185°F)
- Shock: 30 g, operational; 50 g, non-operational; Vibration: 5 g from 10 Hz to 150 Hz
- Relative Humidity: 5% to 95% with no condensation
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Label Markings

ATEX

II 3 G

EEx nA IIC T6

0°C <= Ta <= 60°C

cULus

E183151

Class I Div 2 Groups A,B,C,D

T6

-30°C <= Ta <= 60°C

Contents

Guide to the MVI56-101S User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 13)	This section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Diagnostic and Troubleshooting	→	Diagnostics and Troubleshooting (page 77)	This section describes Diagnostic and Troubleshooting procedures.
Reference Product Specifications	→	Reference (page 95) Product Specifications (page 96)	These sections contain general references associated with this product and its Specifications..
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 183) Index	This section contains Support, Service and Warranty information. Index of chapters.

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 101 and ControlLogix processor devices to a power source and to the MVI56-101S module's application port(s)

1.1 What's New

This new and enhanced version 3 of ProSoft Technology's IEC 60870-5-101 Slave implementation includes several new features not found on previous versions. Version 3 now supports 32-bit bitstring data types. Version 3 also supports more Causes of Transmission than previous versions. For additional details, please see the Interoperability Statement (page 166).

1.2 System Requirements

The MVI56-101S module requires the following minimum hardware and software components:

- Rockwell Automation ControlLogix processor™ processor, with compatible power supply and one free slot in the rack, for the MVI56-101S module. The module requires 800 mA of available power.
- Rockwell Automation RSLogix 5000 programming software version 2.51 or higher
- Rockwell Automation RSLinx communication software
- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows XP Professional with Service Pack 1 or 2
  - Microsoft Windows 2000 Professional with Service Pack 1, 2, or 3
  - Microsoft Windows Server 2003
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- ProSoft Configuration Builder, HyperTerminal or other terminal emulator program.

Note: You can install the module in a local or remote rack. For remote rack installation, the module requires EtherNet/IP or ControlNet communication with the processor.

1.3 Package Contents

The following components are included with your MVI56-101S 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	MVI56-101S Module	MVI56-101S	IEC 60870-5-101 Slave Communication Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
3	Cable	Cable #14, RJ45 to DB9 Male Adapter cable	For DB9 Connection to Module's Port
2	Adapter	1454-9F	Two Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to Port 1 and 2 of the Module

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

1.4 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the module. You can always get the newest version of *ProSoft Configuration Builder* from the ProSoft Technology website.

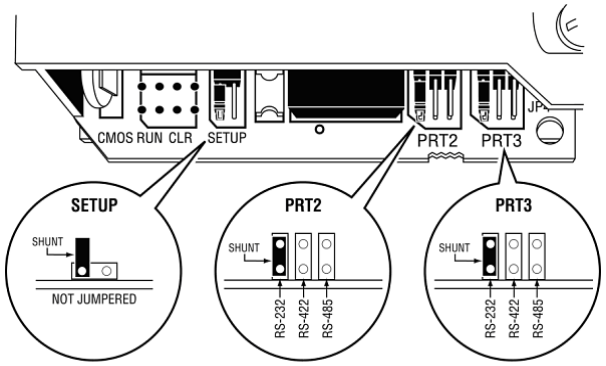
To install ProSoft Configuration Builder from the ProSoft Technology website

1. Open your web browser and navigate to *http://www.prosoft-technology.com/pcb*
2. Click the **DOWNLOAD HERE** link to download the latest version of *ProSoft Configuration Builder*.
3. Choose **SAVE** or **SAVE FILE** when prompted.
4. Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
5. When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

1.5 Setting Jumpers

If you use an interface other than RS-232 (default), you must change the jumper configuration to select the interface you wish to use. There are three jumpers located at the bottom of the module.

The following illustration shows the MVI56-101S jumper configuration:



1. Set the PRT 2 (for application port 1) and PRT 3 (for application port 2) jumpers select RS232, RS422, or RS485 to match the wiring needed for your application. The default jumper setting for both application ports is RS-232.
2. The Setup Jumper acts as "write protection" for the module's flash memory. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support.

1.6 Installing the Module in the Rack

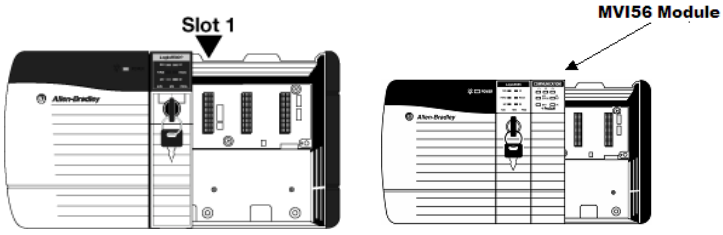
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-101S module. Refer to your Rockwell Automation product documentation for installation instructions.

Warning: You must follow all safety instructions when installing this or any other electronic devices. Failure to follow safety procedures could result in damage to hardware or data, or even serious injury or death to personnel. Refer to the documentation for each device you plan to connect to verify that suitable safety procedures are in place before installing or servicing the device.

After you have checked the placement of the jumpers, insert MVI56-101S into the ControlLogix chassis. Use the same technique recommended by Rockwell Automation to remove and install ControlLogix modules.

Warning: When you insert or remove the module while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations. Verify that power is removed or the area is non-hazardous before proceeding. Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

1. Turn power OFF.
2. Align the module with the top and bottom guides, and slide it into the rack until the module is firmly against the backplane connector.



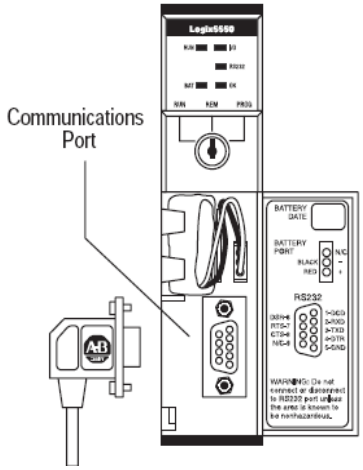
3. With a firm but steady push, snap the module into place.
4. Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
5. Make a note of the slot location. You must identify the slot in which the module is installed in order for the sample program to work correctly. Slot numbers are identified on the green circuit board (backplane) of the ControlLogix rack.
6. Turn power ON.

Note: If you insert the module improperly, the system may stop working, or may behave unpredictably.

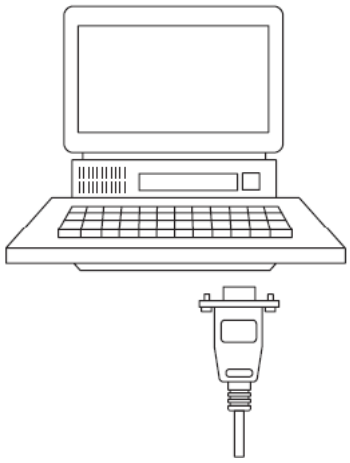
1.7 Connecting Your PC to the ControlLogix Processor

There are several ways to establish communication between your PC and the ControlLogix processor. The following steps show how to establish communication through the serial interface. It is not mandatory that you use the processor's serial interface. You may access the processor through whatever network interface is available on your system. Refer to your Rockwell Automation documentation for information on other connection methods.

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.



1.8 Using the Sample Ladder Logic

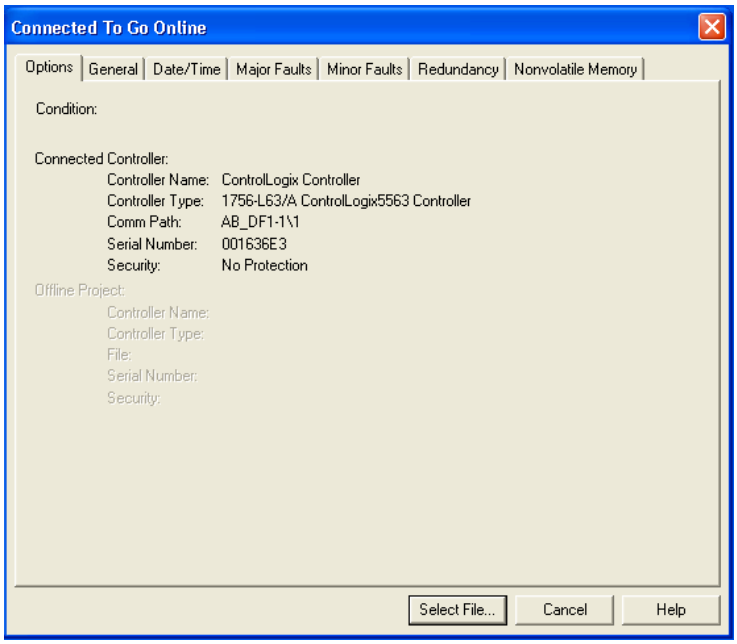
The sample program for your MVI56-101S module includes custom tags, data types, and ladder logic for data I/O and status monitoring. For most applications, you can run the sample ladder program without modification, or, for advanced applications, you can incorporate the sample program into your existing application.

The version number appended to the file name corresponds with the firmware version number of your ControlLogix processor. The firmware version and sample program version must match.

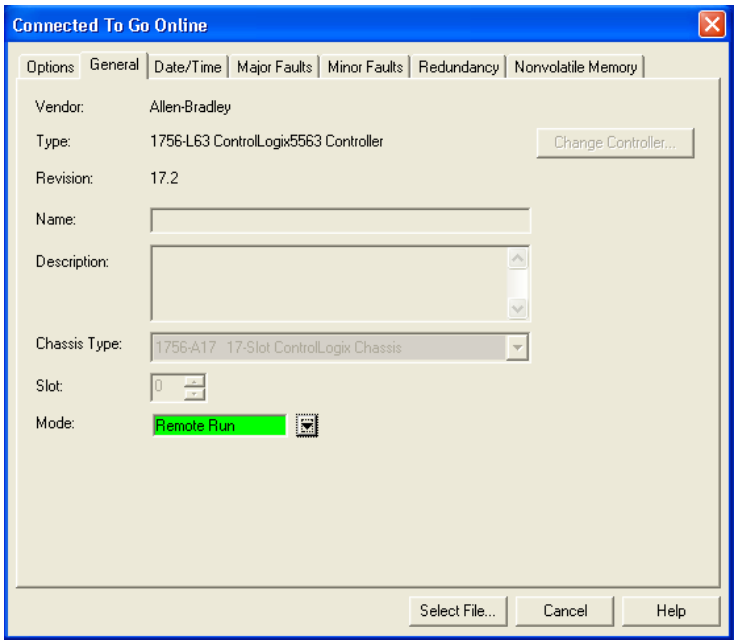
1.8.1 Determining the Firmware Version of Your Processor

Important: The RSLinx service must be installed and running on your computer in order for RSLogix to communicate with the processor. Refer to your RSLinx and RSLogix documentation for help configuring and troubleshooting these applications.

1. Connect an RS-232 serial cable from the COM (serial) port on your PC to the communication port on the front of the processor.
2. Start RSLogix 5000 and close any existing project that may be loaded.
3. Open the **COMMUNICATIONS** menu and choose **GO ONLINE**. RSLogix will establish communication with the processor. This may take a few moments.
4. When RSLogix has established communication with the processor, the *Connected To Go Online* dialog box will open.



5 In the *Connected To Go Online* dialog box, click the **GENERAL** tab. This tab shows information about the processor, including the *Revision* (firmware) version. In the following illustration, the firmware version is 17.2.

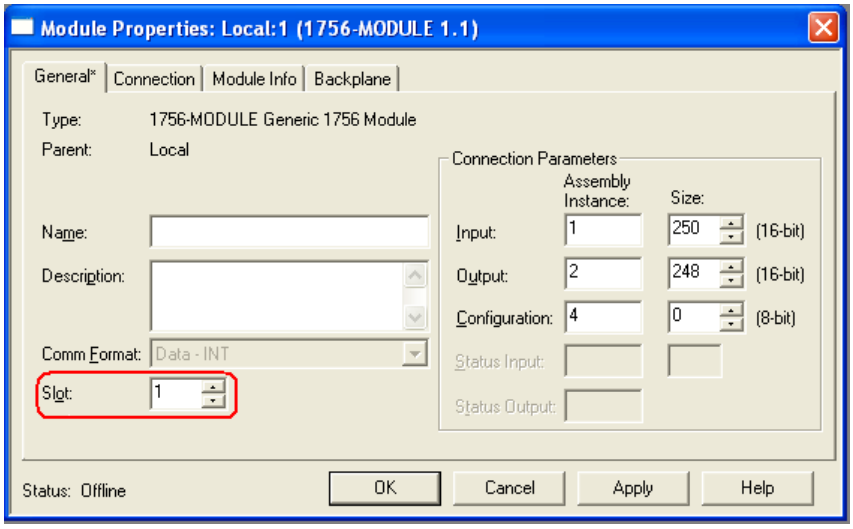


1.8.2 Selecting the Slot Number for the Module

This sample application is for a module installed in Slot 1 in a ControlLogix processor rack. The ladder logic uses the slot number to identify the module. If you are installing the module in a different slot, you must update the ladder logic so that program tags and variables are correct, and do not conflict with other modules in the rack.

To change the slot number

1. In the *Controller Organization* list, select the module and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **PROPERTIES**. This action opens the *Module Properties* dialog box.



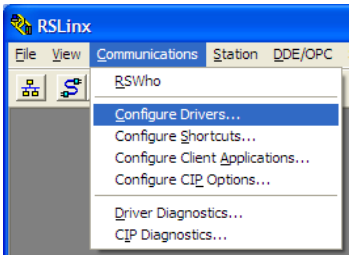
3. In the *Slot* field, use the spinners on the right side of the field to select the slot number where the module will reside in the rack, and then click **OK**.

RSLogix will automatically apply the slot number change to all tags, variables and ladder logic rungs that use the MVI56-101S slot number for computation.

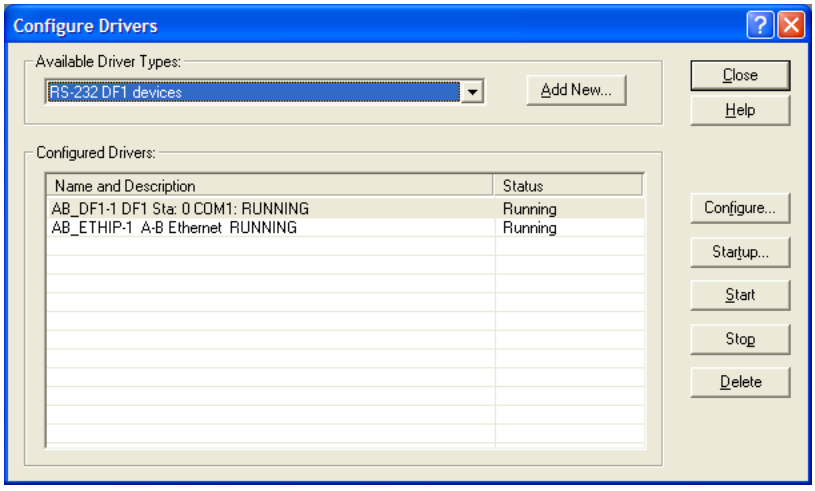
1.8.3 Configuring the RSLinx Driver for the PC COM Port

If RSLogix is unable to establish communication with the processor, follow these steps.

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

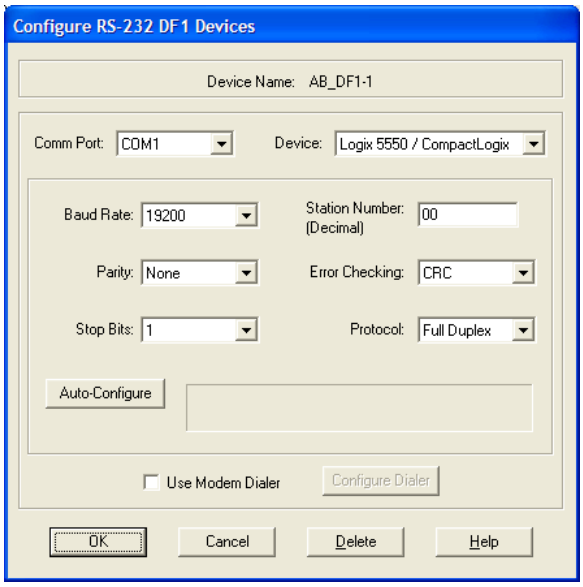


This action opens the *Configure Drivers* dialog box.



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.

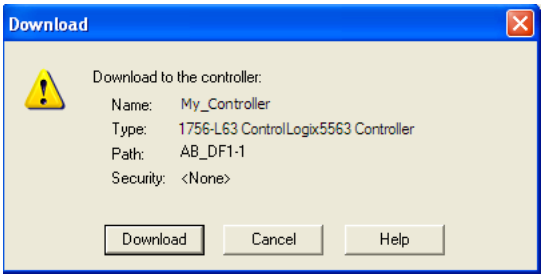
5 When you see the message *Auto Configuration Successful*, click the **OK** button to dismiss the dialog box.

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.

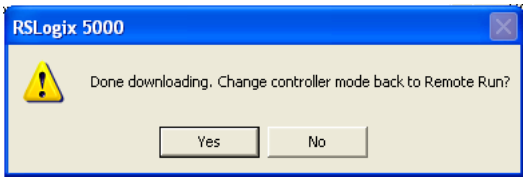
1.9 Downloading the Sample Program to the Processor

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

1. If you are not already online with the processor, open the *Communications* menu, and then choose **DOWNLOAD**. RSLogix 5000 will establish communication with the processor. You do not have to download through the processor's serial port, as shown here. You may download through any available network connection.
2. When communication is established, RSLogix 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



3. RSLogix 5000 will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, RSLogix 5000 will open another confirmation dialog box. If the key switch is in the REM position, click **OK** to switch the processor from PROGRAM mode to RUN mode.

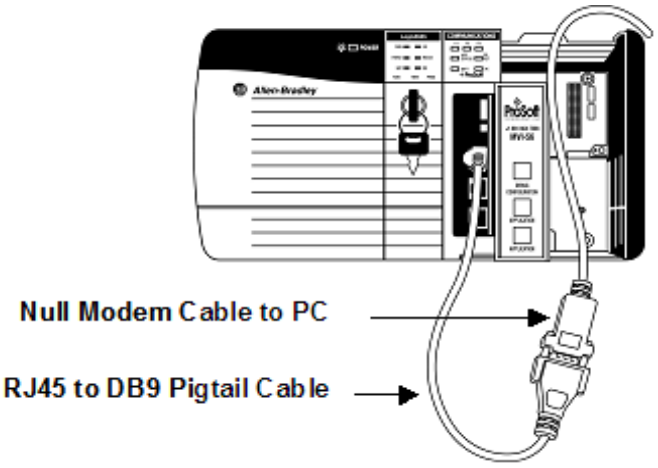


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

1.10 Connecting Your PC to the Module

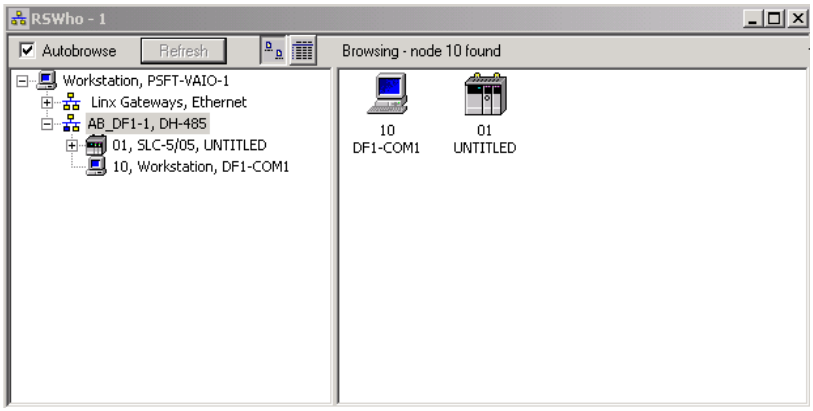
With the module securely mounted, connect your PC to the **Configuration/Debug** port using an RJ45-DB-9 Serial Adapter Cable and a Null Modem Cable.

1. Attach both cables as shown.
2. Insert the RJ45 cable connector into the *Config/Debug* port of the module.
3. Attach the other end to the serial port on your PC.

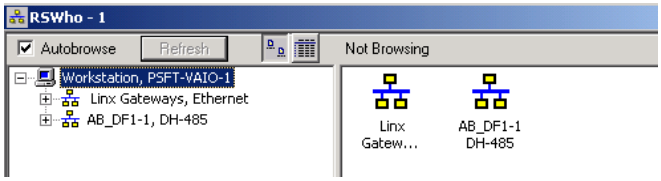


The communication port driver in *RSLinx* can occasionally prevent other applications from using the PC's COM port. If you are not able to connect to the module's configuration/debug port using *ProSoft Configuration Builder (PCB)*, *HyperTerminal* or another terminal emulator, follow these steps to disable the *RSLinx* driver.

1. Open *RSLinx* and go to **COMMUNICATIONS** > **RSWHO**.
2. Make sure that you are not actively browsing using the driver that you wish to stop. The following shows an actively browsed network.



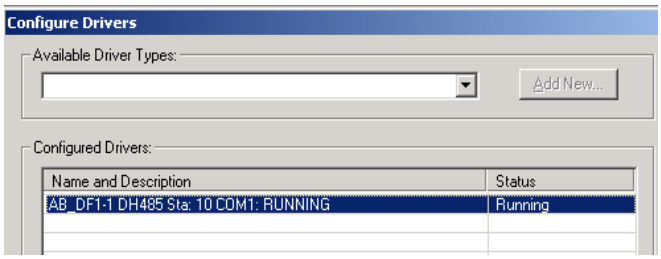
3 Notice how the DF1 driver is opened, and the driver is looking for a processor on Node 1. If the network is being browsed, then you will not be able to stop this driver. To stop the driver your *RSWho* screen should look like this:



Branches are displayed or hidden by clicking on the ⊞ or the ⊟ icons.

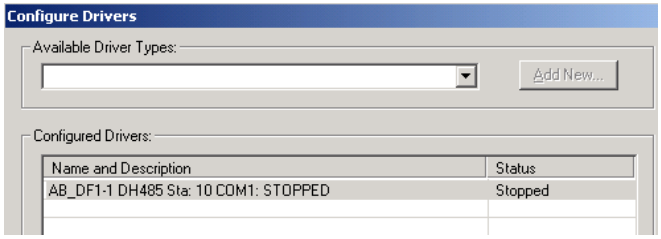
4 When you have verified that the driver is not being browsed, go to **COMMUNICATIONS > CONFIGURE DRIVERS**.

You may see something like this:



5 If you see the status as running, you will not be able to use this COM port for anything other than communication to the processor. To stop the driver press the **STOP** button on the side of the window:

6 After you have stopped the driver you will see the following.



7 You may now use the COM port to connect to the *Config/Debug* port of the module.

Note: You may need to shut down and restart your PC before it will allow you to stop the driver (usually only on *Windows NT* machines). If you have followed all of the above steps, and it will not stop the driver, then make sure you do not have *RSLogix* open. If *RSLogix* is open, you will not be able to stop the DF1 driver. If *RSLogix* is not open, and you still cannot stop the driver, then reboot your PC.

2 Configuring the MVI56-101S Module

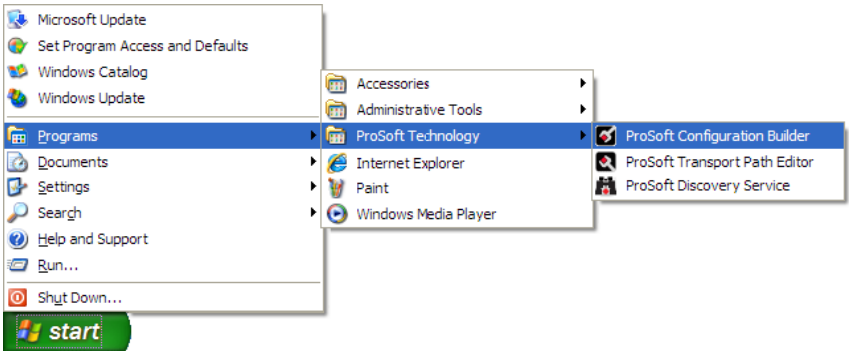
In This Chapter

2.1 Using ProSoft Configuration Builder

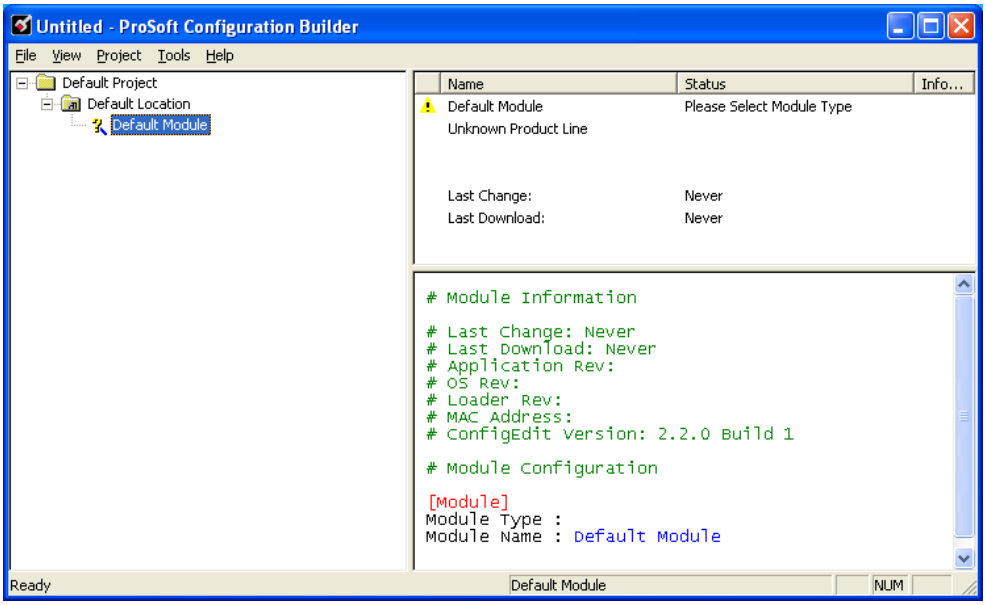
ProSoft Configuration Builder (PCB) provides a convenient way to manage module configuration files customized to meet your application needs. *PCB* is not only a powerful solution for new configuration files, but also allows you to import information from previously installed (known working) configurations to new projects.

2.1.1 Setting Up the Project

To begin, start **PROSOFT CONFIGURATION BUILDER** (PCB).

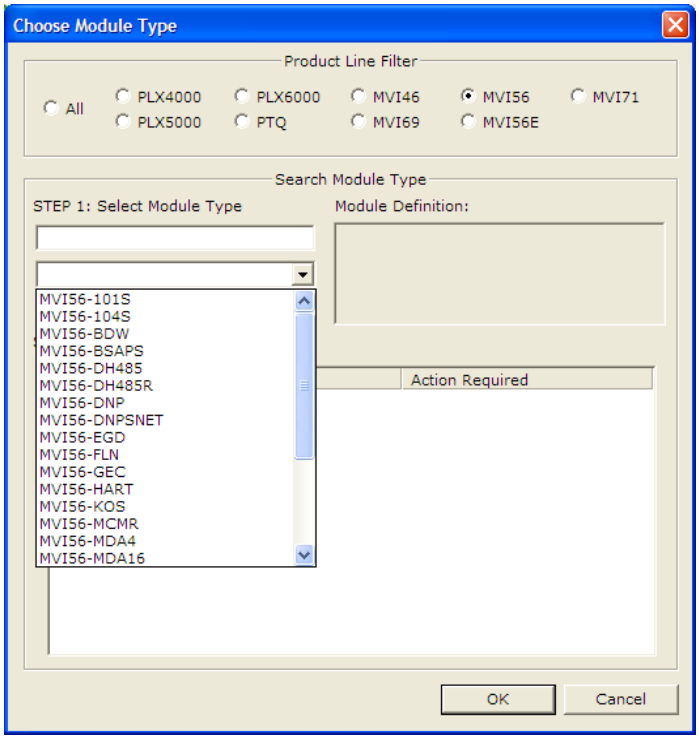


If you have used other Windows configuration tools before, you will find the screen layout familiar. *PCB*'s window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. When you first start *PCB*, the tree view consists of folders for *Default Project* and *Default Location*, with a *Default Module* in the *Default Location* folder. The following illustration shows the *PCB* window with a new project.



Adding the MVI56-101S module to the project

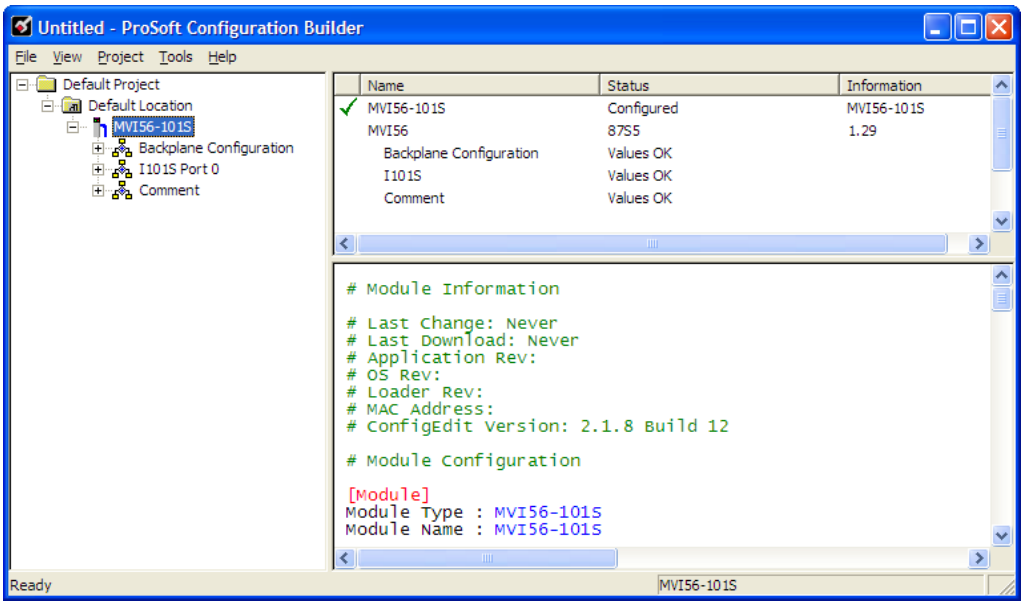
1. Use the mouse to select **DEFAULT MODULE** in the tree view, and then click the right mouse button to open a shortcut menu.
2. On the shortcut menu, choose **CHOOSE MODULE TYPE**. This action opens the *Choose Module Type* dialog box.



3. In the *Product Line Filter* area of the dialog box, select **MVI56**. In the *Select Module Type* dropdown list, select **MVI56-101S**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.

2.1.2 Renaming PCB Objects

Notice that the contents of the information pane and the configuration pane changed when you added the module to the project.



At this time, you may wish to rename the *Default Project* and *Default Location* folders in the tree view.

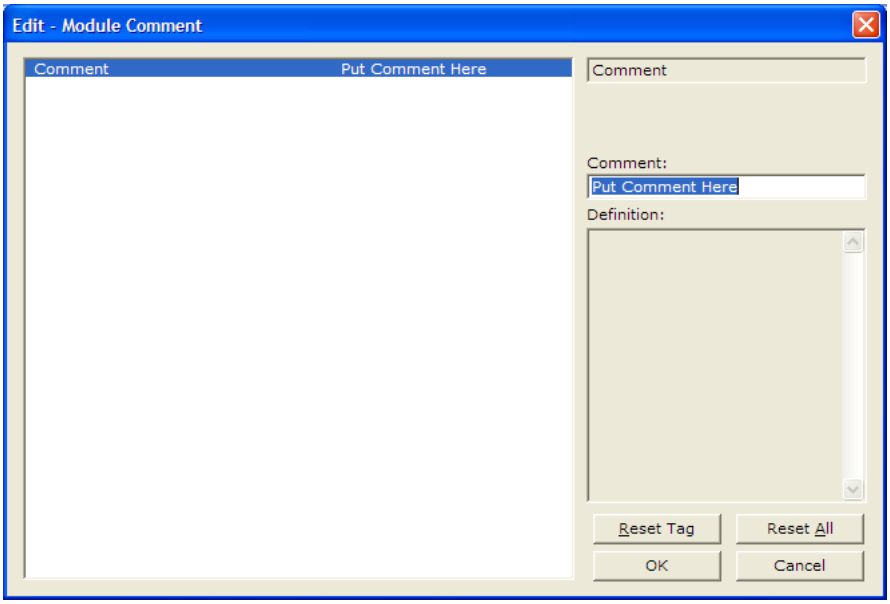
1. Select the object, and then click the right mouse button to open a shortcut menu. From the shortcut menu, choose **RENAME.**
2. Type the name to assign to the object.
3. Click *away* from the object to save the new name.

Configuring Module Parameters

1. Click the **[+]** sign next to the module icon to expand module information.
2. Click the **[+]** sign next to any icon to view module information and configuration options.
3. Double-click any icon to open an *Edit* dialog box.
4. To edit a parameter, select the parameter in the left pane and make your changes in the right pane.
5. Click **OK** to save your changes.

Creating Optional Comment Entries

1 Click the **[+]** to the left of the Comment icon to expand the module comments.
2 Double-click the Module Comment icon. The *Edit - Module Comment* dialog box appears.



3 Enter your comment and click **OK** to save your changes.

Printing a Configuration File

1 Select the module icon, and then click the right mouse button to open a shortcut menu.
2 On the shortcut menu, choose **VIEW CONFIGURATION.** This action opens the *View Configuration* window.
3 In the *View Configuration* window, open the **FILE** menu, and choose **PRINT**. This action opens the *Print* dialog box.
4 In the *Print* dialog box, choose the printer to use from the drop-down list, select printing options, and then click **OK**.

2.2 [Backplane Configuration]

This section provides the module with:

- a unique name
- designates database addresses for input and output on the module and on the processor
- identifies the method of failure for the communications for the module if the PLC is not in run mode
- describes how to initialize the module upon startup

2.2.1 Module Name

0 to **80** characters

This parameter assigns a name to the module that can be viewed using the configuration/debug port. Use this parameter to identify the module and the configuration file.

2.2.2 Read Register Start

0 to **3999**

This parameter specifies the starting register address of a block of data registers to transfer from the module to the processor.

2.2.3 Write Register Count

0 to **4000**

This parameter specifies the number of registers to transfer from the processor to the module.

2.2.4 Write Register Start

0 to **3999**

This parameter specifies the starting register address of a module register block where data transferred from the processor will be stored.

2.2.5 Write Register Count

0 to **4000**

This parameter specifies the number of registers to transfer from the processor to the module.

2.2.6 Failure Flag Count

0 through **65535**

This parameter specifies the number of successive transfer errors that must occur before halting communication on the application port(s). If the parameter is set to **0**, the application port(s) will continue to operate under all conditions. If the value is set larger than **0** (**1** to **65535**), communications will cease if the specified number of failures occur.

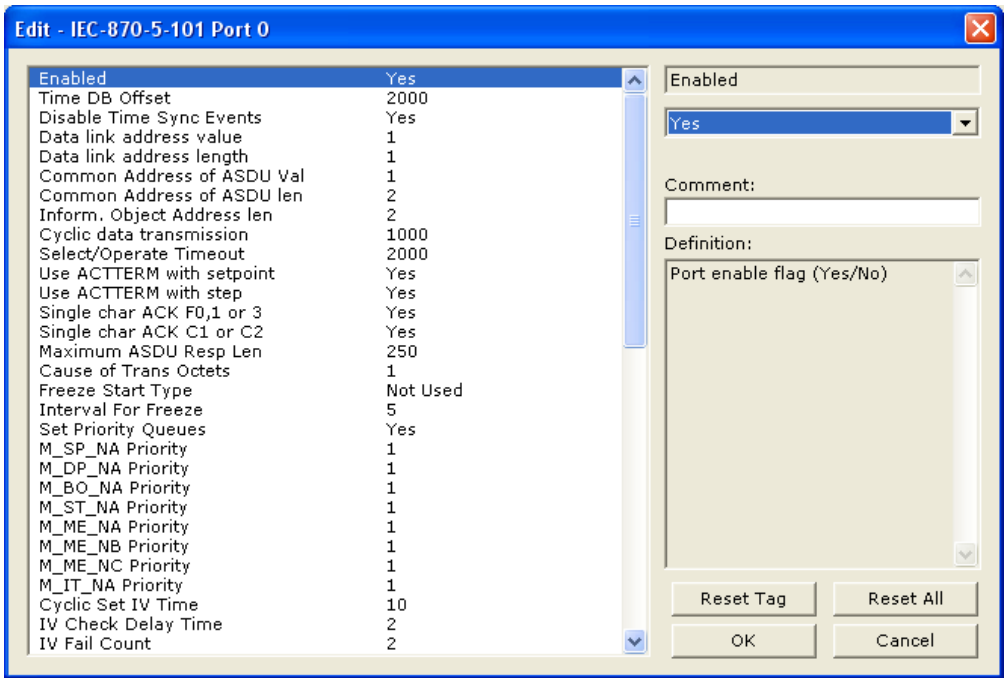
2.2.7 Error Offset

0 to 8899, -1 to disable

This parameter specifies the register location in the module's database where module status data will be stored. If a value less than 0 is entered, the data will not be stored in the database. If the value specified is in the range of 0 to 8966, the data will be placed in the modules database.

2.3 [IEC-870-5-101 Port 0]

This section provides information required to configure a slave application with the module.



2.3.1 Enabled

YES or **NO**

Enables or disables the primary port. If the port is not enabled (No), then the module will not use the port. If the port is enabled (Yes), the module will emulate an IEC 60870-101 slave device on the port.

2.3.2 Time DB Offset

-1, or **0** to **3994**

This parameter defines the location in the database where the time maintained for the IEC protocol is copied. This time is updated whenever a time synchronization command is received from the host and continually as the program runs. If the parameter is set to -1, the time will not be placed in the database.

2.3.3 Disable Time Sync Events

YES or **NO**

This feature can be used when the Master receives the event timestamp with only minutes and milliseconds information for each event message (CP24 time type). If the parameter is set to **N,** the spontaneous time sync events will be generated to indicate the change of hour. If this parameter is set to **Y**, then the spontaneous time sync event messages are not generated to indicate the change of hour.

2.3.4 Data Link Address Value

0 to **65535**

This parameter defines the Data Link Address for the emulated device on the module. This address identifies the module on the network along with the common address of ASDU.

2.3.5 Data link address length

0, **1**, or **2**

This parameter specifies the number of octets used for the data link address. This parameter must be set the same for all devices on the network. A value of 0 is only valid when the balanced mode is used. If unbalanced mode is used, a value of 1 or 2 must be used.

2.3.6 Common Address of ASDU Val

0 to **65535**

This parameter specifies the common address of the ASDU (section address) for access to data in the module. There is only one value entered for access to all data in the module.

Refer to ASDU Configuration for more information on Application Service Data Unit configuration.

2.3.7 Common Address of ASDU Len

1 or **2**

This parameter specifies the number of octets used for the common address of ASDU. This parameter must be set the same for all devices on the network.

2.3.8 Inform. Object Address Len

1, **2**, or **3**

This parameter specifies the number of octets used to define the address of an information object (point address).

2.3.9 Cyclic Data Transmission

0 to **4,294,967,296** (2 raised to the power of 32) milliseconds

This parameter defines the number of milliseconds between cyclic updates. The range of values allow for update times between 1 millisecond and 49.7 days. Set this parameter to 0 to disable cyclic data reporting.

2.3.10 Select/Operate Timeout

0 to **4,294,967,296** (2 raised to the power of 32) milliseconds

This parameter sets the number of milliseconds to wait for a valid Execute command after receiving a Select command. The range of values allows for update times between 1 millisecond and 49.7 days. Set this parameter to **0** to disable this feature.

2.3.11 Use ACTTERM with Set Point

Y - YES or **N - NO**

This parameter determines if an ACTTERM will be sent. If the parameter is set to **YES**, then Set point commands will issue an ACTTERM when the command is complete. If the parameter is set to **NO**, ACTCON is the last response to a Set point command.

2.3.12 Use ACTTERM with Step

Y - YES or **N - NO**

This parameter determines if an ACTTERM will be sent. If the parameter is set to **YES**, then Step commands will issue an ACTTERM when the command is complete. If the parameter is set to **NO**, ACTCON is the last response to a Step command.

2.3.13 Single char ACK F0, 1 or 3

Y - YES or **N - NO**

If set to Yes, a single character ACK (0xE5) will be sent instead of a fixed length ACK (secondary function code 0) in response to a primary link function code 0, 1 or 3 if there is no access demand for class 1 data (ACD=1). If set to No, the fixed length ACK will be sent.

2.3.14 Single char ACK C1 or C2

Y -YES or **N - NO**

If set to Yes, a single character ACK (0xE5) will be sent instead of a fixed length NACK (secondary function code 9) when no response user data is available. If set to No, the fixed length NACK will be sent.

2.3.15 Maximum ASDU Resp Len

25 to **252**

Sets the maximum ASDU response message length (usually 252).

2.3.16 Cause of Trans(mission) Octets

1 or **2**

Specifies the number of COT octets (1 or 2)

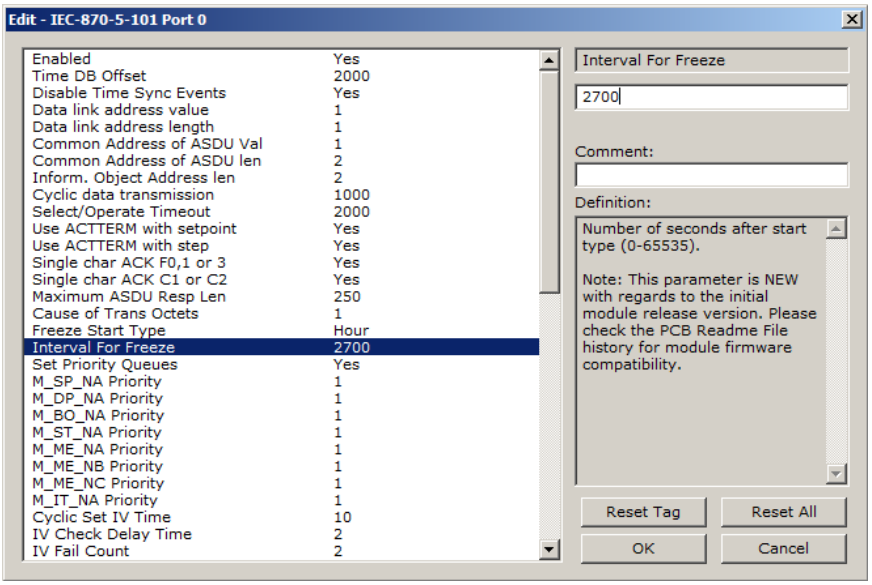
2.3.17 Freeze Start Type

D = DAY, H = HOUR, M = MINUTE, N = NOT USED

The *Freeze Start Type* parameter defines when the module starts sending the M_IT messages.

Example I - Freeze Start Type

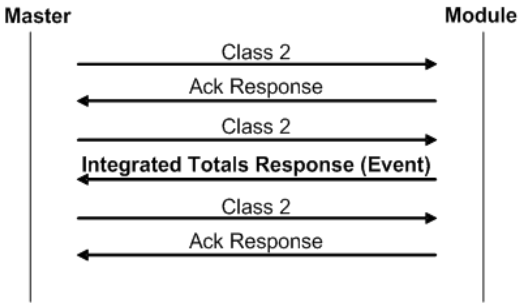
If the module should send the counter points on the hourly turn around time and also 45 minutes later, the Mode A parameters should be configured as follows:



So the module would send events as follows (Hours:Minutes:Seconds):

```
17:00:00
17:45:00
18:00:00
18:45:00
19:00:00
19:45:00
…
```

The following illustration shows a typical communication example when the Mode A is selected during unbalanced mode operation:

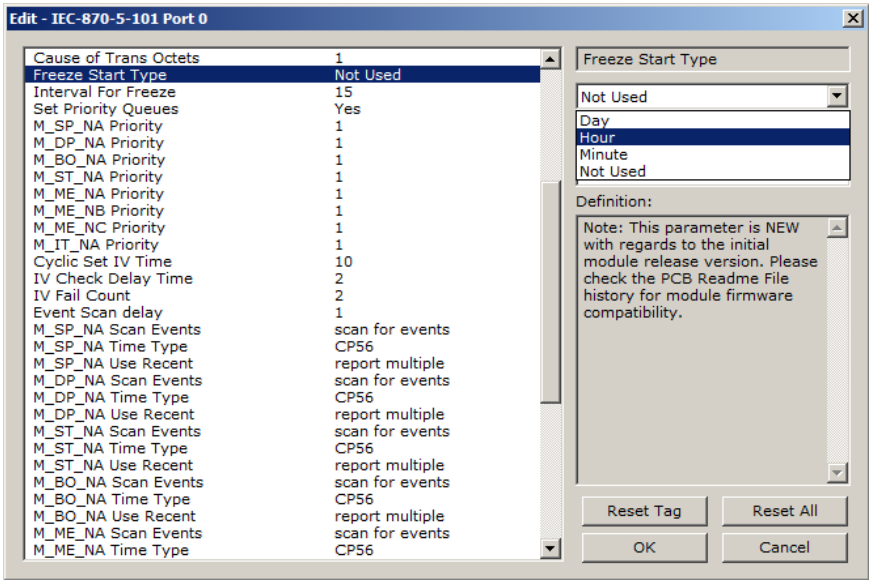


Example II - Freeze Start Type

If the module powers up with the following date and time settings on its internal clock:

03/25/2004 18:07:42

And if you configure the *Interval For Freeze* parameter to 15 seconds, as follows:



The Freeze Start Type parameter determines when the module will begin sending these messages, as follows:

If the Freeze Start Type is set to:	Then Messages will start being sent at:
D	03/26/2004 00:00:00 and every 15 seconds thereafter
H	03/25/2004 19:00:00 and every 15 seconds thereafter
M	03/25/2004 18:08:00 and every 15 seconds thereafter

Once message transmission begins, the module will freeze and transmit counter values at whatever interval has been configured using the *Interval For Freeze* parameter. In this example, that would be every 15-seconds.

2.3.18 Interval for Freeze

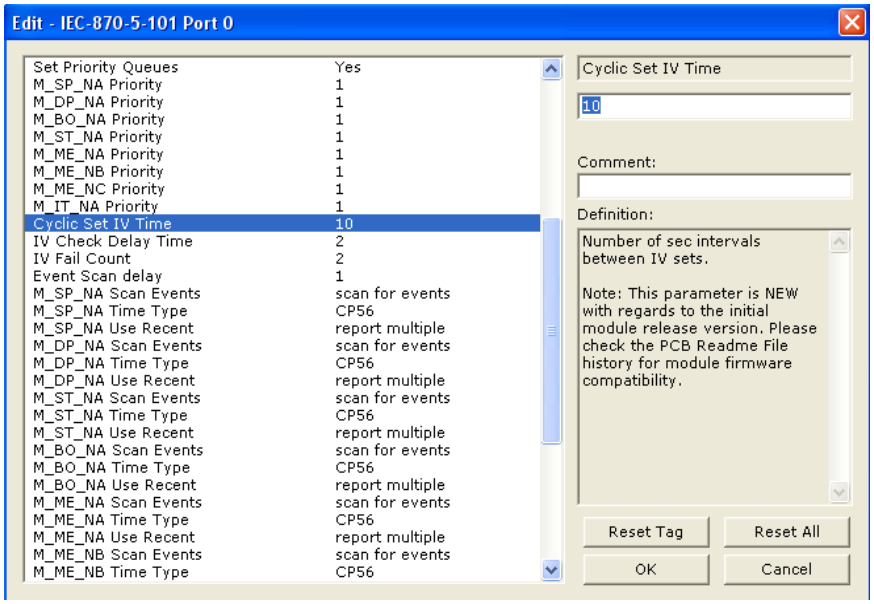
0 to **65535** seconds

Freeze Start Type and *Interval for Freeze* are used if Mode A operation is to be used for the counter freeze operation. If they are not used, the module will operate in Mode D.

2.3.19 Set Priority Queues

YES or **NO**

This section defines priority queues for the module. You can assign priorities to data types that return events so that events of some data types will be returned before events of other data types. If this feature is utilized, each data type must be assigned a unique index from 0 to 7. The lower the index, the higher the priority (0=highest priority, 7=lowest priority).



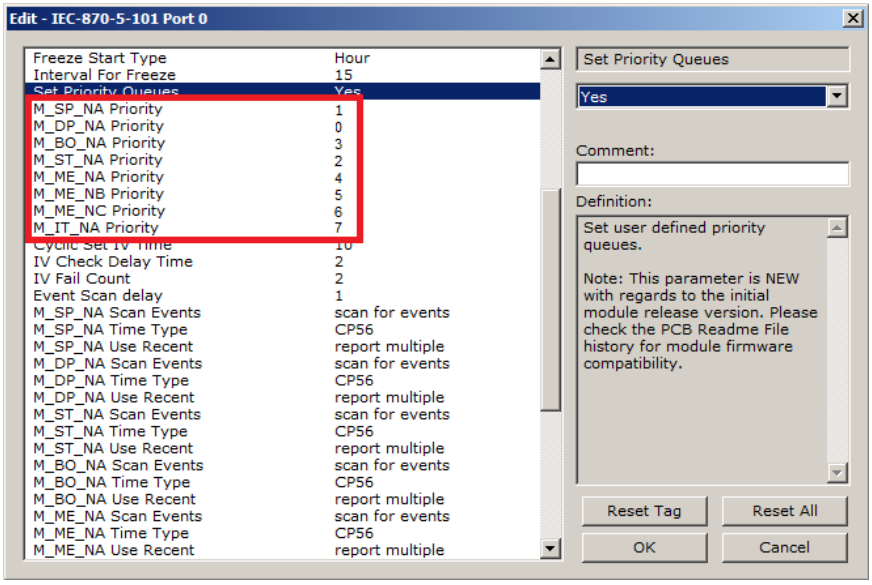
Events of ASDUs with lower numbers will be reported before events from other ASDUs with higher numbers.

Note: In some configurations, depending on the total number of events generated for all data types, setting priorities may cause some events to be lost, as the event buffers for low priority queues may overflow before their events can be reported.

For more information, refer to Event Priority (page 44).

2.3.20 Event Priority

Event Priority permits the user to set reporting priorities for data change events generated for each data type. The configuration file contains the following parameters to support this feature:



The *Set Priority Queues* parameter must be enabled for this feature to be used. Each of the Monitored Point ASDUs must be assigned a unique priority index from 0 to 7. When events of the ASDU with a lower priority number are present, they will always be reported before events from any other ASDUs with higher priority numbers.

2.3.21 Cyclic Set IV Time

1 to **65535** seconds, or **0** to disable Invalid Bit Monitoring

The parameter should be set to a value significantly greater than the value of the *IV Check Delay Time* parameter, multiplied by the value of the *IV Fail Count* parameter. Example:

If, *IV Check Delay* = **3**;
And, *IV Fail Count* = **5**;
Then set *Cyclic Set IV Time* significantly greater than 15; preferably some multiple of this value, like **30**, **45**, **60,** or more.

The *Cyclic Set IV Time* parameter is one of three parameters needed to establish a fail-safe data validation system for this protocol implementation. This parameter is used with the *IV Check Delay Time* and the *IV Fail Count* parameters to create a way to alert the remote IEC-101 Master to situations where data being sent by the module might be invalid.

The module will automatically set ON (set to one) the IV Bits of all IEC database Monitor Points (M_xx_xx point) that have been configured to use invalid bit monitoring. This means that all IEC database Monitor Points configured with IV Bit addresses greater than zero (0) will have their IV Bit addresses in the module database set to a value of one (1) by the module firmware in a timed, periodic cycle, based on the number of seconds entered in *Cyclic Set IV Time*.

The IV Bits are checked at the interval specified by the *IV Check Delay Time* parameter. At the end of each check delay interval, if an IV Bit for any IEC database point is ON, an IV Bit Check Fail accumulator for that point will be incremented. If the value in any IV Bit Check Fail accumulator becomes equal to the value of the *IV Fail Count* parameter, the module will consider the data value of that point invalid and will report the invalid status of that point to the Master along with the data value.

If the IV Bit is cleared (reset, set OFF, set to zero) before the IV Bit Check Fail accumulator value becomes equal to the value of the *IV Fail Count* parameter, the IV Bit Check Fail accumulator will be reset to 0. The module will consider the data value of that point valid and will report the valid status of that point to the Master along with data value.

To properly use data validity checking with a ProLinx Gateway, the application that communicates to the gateway through the other gateway protocol (Modbus, EtherNet/IP, etc.) must be programmed to provide data validity information, as well as the actual data to be transferred. This other application must write bit values of zero (0) or one (1) to the proper module database addresses along with the data for each IEC database point.

This means the other protocol application will be responsible for clearing the IV Bits for data points which contain valid data at a time interval less than the amount of time required to accumulate the required number of IV Bit Check failures. If the other protocol application does not clear the IV Bits, then those IEC database monitor point values will be reported to the Master as invalid.

NOTE: If one or more of the *Cyclic Set IV Time*, the *IV Check Delay Time*, or the *IV Fail Count* parameters are set to zero, the data validity checking feature will be disabled for all points in the IEC database and the data reported to the remote Master will always be reported as valid data. If you wish to have actual data validity information relayed to the Master, be sure the *Cyclic Set IV Time*, the *IV Check Delay Time*, and the *IV Fail Count* parameters are set to non-zero values and be sure the application has been programmed to properly manipulate the IV Bit addresses in the module database. Once data validity checking has been enabled, you can still turn off validity checking on a point-by-point basis by setting the IV Bit Address to zero (0) for any point you wish to exclude from data validity checking and reporting.

For more information on the data validity features of this protocol implementation, refer to Invalid Bit Monitoring.

2.3.22 IV Check Delay Time

1 to **65535** seconds, or **0** to disable Invalid Bit Monitoring

This parameter sets the number of seconds between IV Bit value checks. Every IEC database Monitor information object which has an IV Bit address set greater than 0 will have that bit address checked at the interval specified by this parameter.

Setting this parameter to zero (**0**) will disable data validity checking.

2.3.23 IV Fail Count

1 to **65535** seconds, or **0** to disable Invalid Bit Monitoring

This parameter sets the number of successive IV Bit check failures which must occur before the data from an IEC database Monitor information object will be reported to the remote Client as invalid data.

An IV Bit Check Failure occurs when the IV Bit value in the module database is set ON {when it contains a value of one (1)} at the time an IV Bit Check is performed. IV Bit Check failures are counted and held in separate IV Bit Check Failure accumulators for each IEC Monitor information object configured for validity checking. If the value in any information object's failure accumulator becomes equal to the value set in this *IV Fail Count* parameter, the module will consider data from this information object invalid and report its invalid status to the Client, along with its data value.

Setting this parameter to zero (**0**) will disable data validity checking.

2.3.24 Event Scan Delay

1 to **65535** milliseconds, or **0** to disable

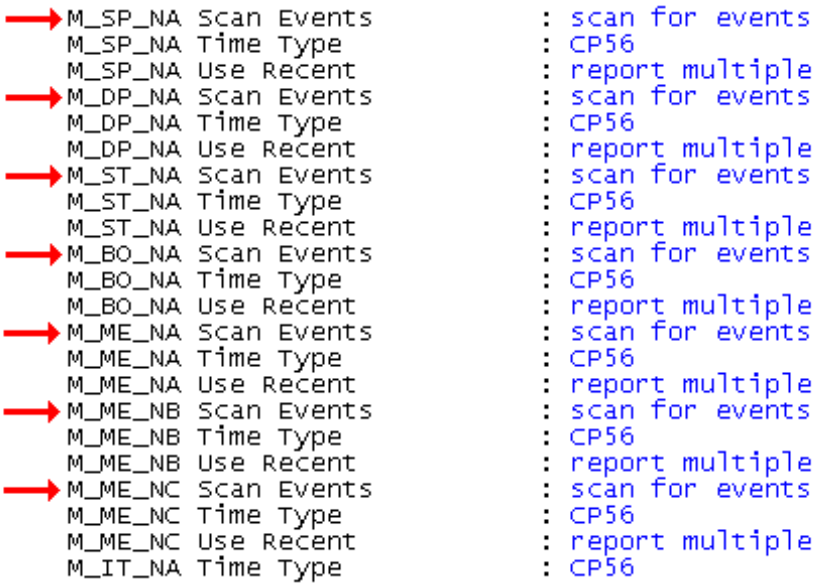
If set to **0**, the feature will be disabled and the module will not generate any events. If set from **1** to **65535**, the parameter represents the number of milliseconds between event scanning. This parameter defines how often the program will scan for new events in the databases.

NOTE: The lower you set this value, the more negative effect it will have on overall module data update and transfer performance. Scanning for new events causes an interrupt of other data manipulation functions. The more often these scan interrupts occur, the less processor time the module will have available for other data operations. When choosing a value for this parameter, you should consider such factors as how often events might be generated by your application and how often you need to recognize and respond to such events. This will allow you to set this value as high as it can be set and still maintain your overall event capturing and reporting strategy. Doing so can reduce the amount of interrupts, prevent excessive event scanning, and maximize data update and transmission rates.

2.3.25 Scan Events

[0] = **NO SCANNING** or **[1]** = **SCAN FOR EVENTS**

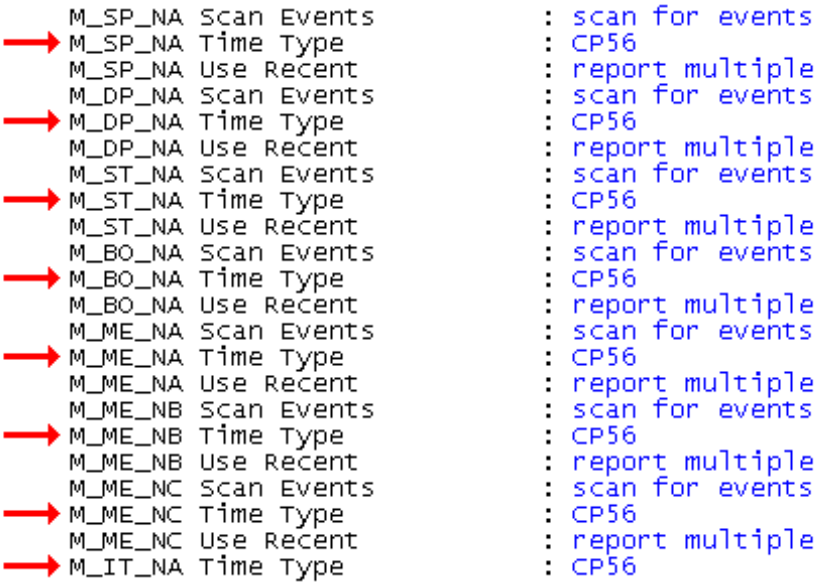
Separate parameters exist for each Monitor ASDU and define whether or not that particular ASDU's events will be generated by the module. If the *Scan Events* parameter is set to zero **[0]** or **NO SCANNING**, then events will not be generated. If this parameter is set to **[1]** or **SCAN FOR EVENTS**, events will be scanned and generated on data change.



2.3.26 Time Type

[0] = *None*, **[1]** = *CP24*, or **[2]** = *CP56*

Separate parameters exist for multiple ASDUs. This parameter defines the time format used with data events.



2.3.27 Use Balanced Mode

Y - YES or **N - NO**

This parameter specifies if the port will use balanced mode. If balanced mode is used, only one controlled station will be permitted on the port. If unbalanced mode is used, multiple controlled stations can be used on a port.

Refer to module Initialization (page **Error! Bookmark not defined.**) for more information on these modes.

2.3.28 Retry Count

0 to 255

In balanced mode, this parameter specifies the number of retries (0 to 255) if a response is not received. In unbalanced mode, this parameter is ignored.

2.3.29 Response Timeout

0 to **65535** milliseconds

This parameter specifies the minimum number of milliseconds to wait for a response to a primary message. Do not set this parameter too small or timeout conditions may prevent successful data transmission. If the timeout is recognized, the message will be retransmitted up to the number of times specified in the Retry Count parameter. This parameter is only used in balance mode.

2.3.30 Baud Rate

300, **600**, **1200**, **2400**, **4800**, **9600**, **19200**, or **38400**

This parameter specifies the baud rate for the primary port on the module. Baud rates from 300 to 38400 are supported on the module.

2.3.31 Parity

None, Odd, Even

Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use.

All devices communicating through this port must use the same parity setting.

2.3.32 RTS On

0 to **65535** milliseconds

This parameter sets the number of milliseconds to delay after *Ready To Send* (RTS) is asserted before data will be transmitted.

2.3.33 RTS Off

0 to **65535** milliseconds

This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.

2.3.34 Minimum Delay

0 to **65535** milliseconds

This parameter defines the minimum number of milliseconds to wait before a response is sent from the unit.

2.3.35 Receive Timeout

0 to **65535** milliseconds

This parameter specifies the minimum number of milliseconds to wait after the first byte of a frame is received before a timeout condition is set. Be careful not to set this parameter too small. If the timeout condition is set, all bytes in the frame received will be discarded.

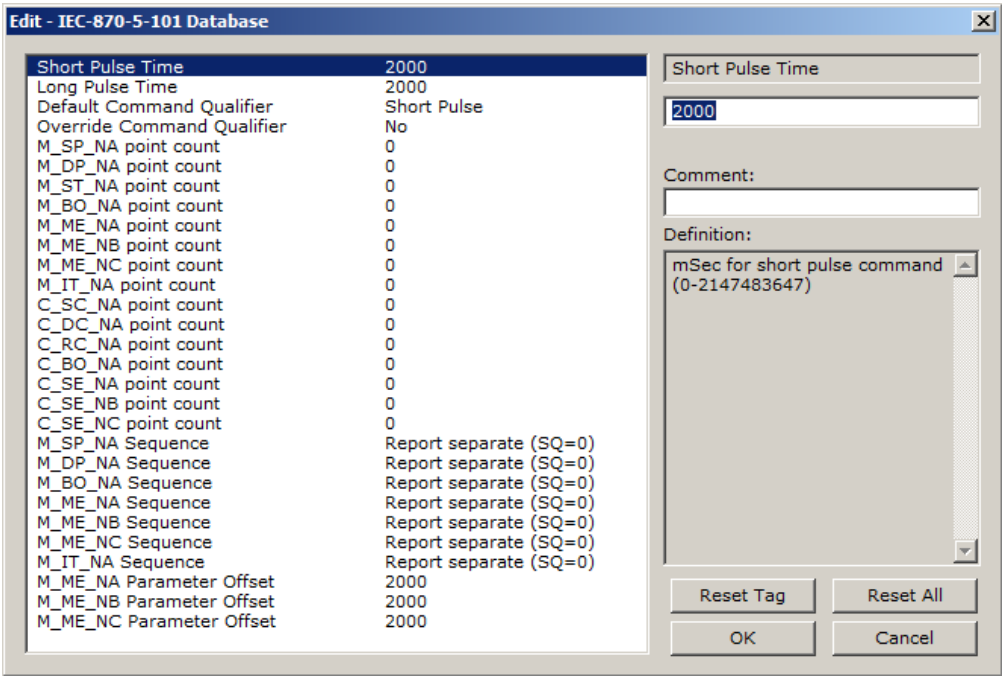
2.3.36 Hardware Handshaking

NONE, **RTS/CTS**, **DTR/DSR**, **MODEM**

Selects the hardware handshaking type. **RTS/CTS**, **DTR/DSR** and **MODEM** options are most commonly used for older telephone line modems. **RTS/CTS** and **DTR/DSR** are also commonly used with leased-line data modems. The **MODEM** option is also commonly used with dial-up modems. When the messages must go out if the modem is not connected, the module will assert the DTR line which will cause the modem to dial the number in the modem's configuration. The module does not have the functionality to dial the number to be called for the modem. The module reconnects after 30 seconds of inactivity.

2.4 [IEC-870-5-101 Database]

This section configures the IEC-870-5-101 Database.



2.4.1 Short Pulse Time

0 to **2,147,483,647** (2 raised to the power of 31, minus 1) milliseconds

This parameter defines the number of milliseconds to be associated with a short pulse command.

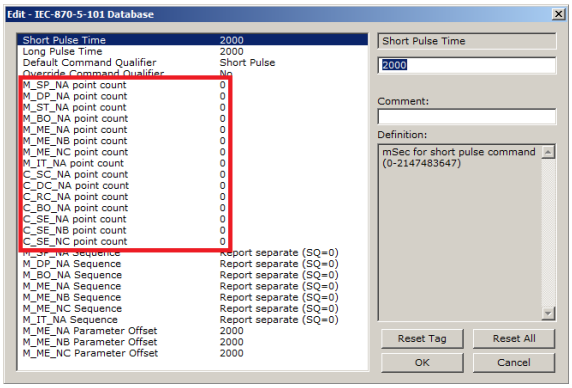
2.4.2 Long Pulse Time

0 to **2,147,483,647** (2 raised to the power of 31, minus 1) milliseconds

This parameter defines the number of milliseconds to be associated with a long pulse command.

2.4.3 Point Count

Point Count configuration ranges in the following configuration items are based on the assumption that you will be using only one of the available data types for your application. The number of point counts you configure will have an effect on module performance, in particular, the accuracy of the module's internal clock.

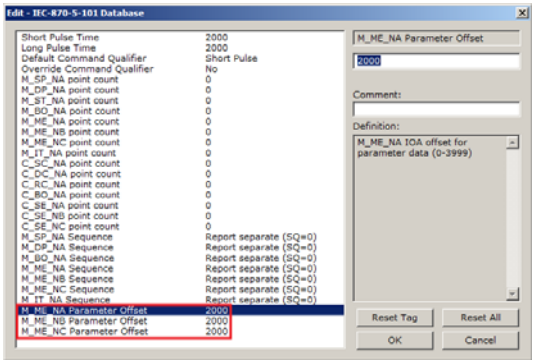


Parameter	Description	Range
M_SP_NA point count	The number of point values assigned in monitored single-point database.	0 to 1000
M_DP_NA point count	The number of point values assigned in monitored double-point database.	0 to 1000
M_ST_NA point count	The number of point values assigned in monitored step-point database.	0 to 1000
M_BO_NA point count	The number of 32-bit values assigned in monitored bitstring database	0 to 1000
M_ME_NA point count	The number of point values assigned in monitored normalized-point database.	0 to 1000
M_ME_NB point count	The number of point values assigned in monitored scaled-point database.	0 to 1000
M_ME_NC point count	The number of point values assigned in monitored scaled short-float point database.	0 to 50
M_IT_NA point count	The number of point values assigned in monitored counter-point database.	0 to 99
C_SC_NA point count	The number of point values assigned in command single-point database.	0 to 1000
C_DC_NA point count	The number of point values assigned in command dual-point database.	0 to 1000
C_RC_NA point count	The number of point values assigned in command step-point database.	0 to 1000
C_BO_NA point count	The number of 32-bit values assigned in command bitstring database	0 to 1000
C_SE_NA point count	The number of point values assigned in command normalized-point database.	0 to 1000
C_SE_NB point count	The number of point values assigned in command scaled-point database.	0 to 1000
C_SE_NC point count	The number of point values assigned in command short-float point database.	0 to 50

2.4.4 Sequence Flag

In order to save bandwidth, you can configure the module to use the Sequence Flag feature. If this feature is not selected, the module will send the Information Object Address (IOA) and the data value for every Monitor information object in a poll response to the Master. If this parameter is enabled, the module will turn on the Sequence Flag bit in a Monitor response; send the Informaiton Object Address of only the first object in the response, along with its data value; and send only the data values for all the other information objects in the response. The Master assumes that all other points use IOAs in a contiguous order (using the first point as the starting reference address). Since the IOAs are omitted for all but the first point, this feature allows each message packet to contain more process data, which improves network throughput and optimizes bandwidth utilization.

2.4.5 Parameter Offset



This parameter specifies the IOA (Information Object Address) offset to the parameter data for the normalized parameter data. The value entered is added to the Information Object address for the associated point to compute the parameter IOA address.

The Master may send a "Parameter of Measured Normalized" or "Parameter of Measured Scaled" command using the parameter IOA in order to change the deadband values for specific points.

Note: The Low Limit and High Limit values are always calculated based on the deadband value as described in the following table.

Point	Value
Threshold	Determined by the deadband set in the configuration file or altered by the write command.
Low	Last reported event value - threshold.
High	Last reported event value + threshold.

For example, if the configuration sets two M_ME_NA points with IOA (Point #) of 600 and 601 and an M_ME_NA Parameter Offset value of 3000, it would result in the parameter points shown in the table.

2.5 [M_SP_NA_1]

This section defines the monitored single-point database. This information is sourced from the database and is transferred to the remote Master unit.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **IV DB BIT**

Each point consumes one bit (1 = On, 0 = Off state). The **DB ADDRESS VALUE** corresponds to the bit offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.6 [M_DP_NA_1]

This section defines the monitored dual-point database. This information is sourced from the database and is transferred to the remote Master unit.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **IV DB BIT**

Each point consumes two bits (00 = intermediate, 01 = off, 10 = on and 11 = intermediate). The **DB ADDRESS** value corresponds to the bit offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.7 [M_ST_NA_1]

This section defines the monitored step database. This information is sourced from the database and is transferred to the remote Master unit.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **IV DB BIT**

Each point consumes one, 8-bit byte. The **DB ADDRESS** value corresponds to the byte offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.8 [M_BO_NA_1]

This section defines the *Monitored 32-bit Bitstring Data Point* database. This information is sourced from the database and is transferred to the remote Master unit.

This section takes the following parameters:

- Point #
- DB Address
- Groups
- IV DB Bit

Each point is four 8-bit bytes (two 16-bit words) and the *DB Address* value corresponds to the double word offset in the module memory database.

The *DB address* should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.9 [M_ME_NA_1]

This section defines the monitored measured value, normalized database. This information is sourced from the database and is transferred to the remote Master unit.

To determine the IOA (Information Object Address) for each object, add the **POINT #** in the following section to the value of the M_ME_NA parameter offset parameter set in the previous section.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **DEFAULT DEADBAND**
- **IV DB BIT**

Each point consumes one, 16-bit word. The **DB ADDRESS** value corresponds to the word offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.10 [M_ME_NB_1]

This section defines the monitored measured value, scaled database. This information is sourced from the database and is transferred to the remote Master unit.

To determine the IOA (Information Object Address) for each object, add the **POINT #** in the following section to the value of the M_ME_NB parameter offset parameter set in the previous section.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **DEFAULT DEADBAND**
- **IV DB BIT**

Each point consumes one, 16-bit word. The **DB ADDRESS** value corresponds to the word offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.11 [M_ME_NC_1]

This section defines the monitored short-float point database. This information is sourced from the database and is transferred to the remote Master unit.

To determine the IOA (Information Object Address) for each object, add the **POINT #** in the following section to the value of the M_ME_NC Parameter Offset parameter set in the previous section.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **DEFAULT DEADBAND**
- **IV DB BIT**

Each point consumes four 8-bit bytes (two 16-bit words). The **DB ADDRESS** value corresponds to the double-word offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.12 [M_IT_NA_1]

This section defines the monitored integrated totals (counter) database. This information is sourced from the database and is transferred to the remote Master unit. This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **GROUPS**
- **IV DB BIT**

Each point consumes four 8-bit bytes (two 16-bit words). The **DB ADDRESS** value corresponds to the double-word offset address in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the module to the Master.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.13 [C_SC_NA_1]

This section defines the single point command database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct **MONITOR POINT #** and **MONITOR DB ADDRESS** values in the table. If the **REQUIRE SELECT** parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **MONITOR POINT #**
- **MONITOR DB ADDRESS**
- **REQUIRE SELECT**

Each point consumes one bit. The **DB ADDRESS** value corresponds to the bit offset in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.14 [C_DC_NA_1]

This section defines the double point command database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored double point database value to coordinate the command/monitor operation. You must enter the correct **MONITOR POINT #** and **MONITOR DB ADDRESS** values in the table. If the **REQUIRE SELECT** parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **MONITOR POINT #**
- **MONITOR DB ADDRESS**
- **REQUIRE SELECT**

Each point consumes two bits. The **DB ADDRESS** value corresponds to the bit offset in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.15 [C_RC_NA_1]

This section defines the step command database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct **MONITOR POINT #** and **MONITOR DB ADDRESS** values in the table. If the **REQUIRE SELECT** parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **MONITOR POINT #**
- **MONITOR DB ADDRESS**

Each point consumes one byte. **THE DB ADDRESS** value corresponds to the byte offset in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.16 [C_BO_NA_1]

This section defines the *Command 32-bit Bitstring Point* database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a *Monitored Double-Point* database value to coordinate the command/monitor operation. You must enter the correct *Monitor Point #* and *Monitor DB Address* values in the table. If the *Require Select* parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- Point #
- DB Address
- Monitor Point #
- Monitor DB Address

Each point is four 8-bit bytes and the *DB Address* value corresponds to the double word offset in the module memory database.

The *DB address* should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.17 [C_SE_NA_1]

This section defines the normalized setpoint database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct **MONITOR POINT #** and **MONITOR DB ADDRESS** values in the table. If the **REQUIRE SELECT** parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **MONITOR POINT #**
- **MONITOR DB ADDRESS**
- **REQUIRE SELECT**

Each point consumes one word. The **DB ADDRESS** value corresponds to the word offset in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.18 [C_SE_NB_1]

This section defines the scaled setpoint database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct **MONITOR POINT #** and **MONITOR DB ADDRESS** values in the table. If the **REQUIRE SELECT** parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **MONITOR POINT #**
- **MONITOR DB ADDRESS**
- **REQUIRE SELECT**

Each point consumes one word. The **DB ADDRESS** value corresponds to the word offset in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.19 [C_SE_NC_1]

This section defines the short-float setpoint database. This information is sourced from the remote Master and is transferred to the database.

You can associate a command with a monitored single-point database value to coordinate the command/monitor operation. You must enter the correct **MONITOR POINT #** and **MONITOR DB ADDRESS** values in the table. If the **REQUIRE SELECT** parameter is not set to zero, a select command must be received before an execute command will be processed.

This section takes the following parameters:

- **POINT #**
- **DB ADDRESS**
- **MONITOR POINT #**
- **MONITOR DB ADDRESS**
- **REQUIRE SELECT**

Each point consumes two words. The **DB ADDRESS** value corresponds to the double-word offset in the database.

The **DB ADDRESS** should be located in a database area that is being constantly moved from the Master to the module.

For additional information on how to set these parameters, see the *Reference* chapter of this manual.

2.20 Cyclic Data Transmission

0 to **4,294,967,296** (2 raised to the power of 32) milliseconds

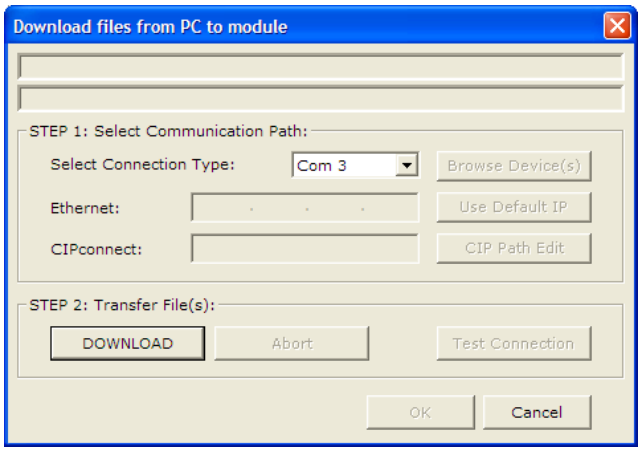
This parameter defines the number of milliseconds between cyclic updates. The range of values allow for update times between 1 millisecond and 49.7 days. Set this parameter to 0 to disable cyclic data reporting.

2.21 Downloading the Project to the Module Using a Serial COM Port

For the module to use the settings you configured, you must download (copy) the updated *Project* file from your PC to the module.

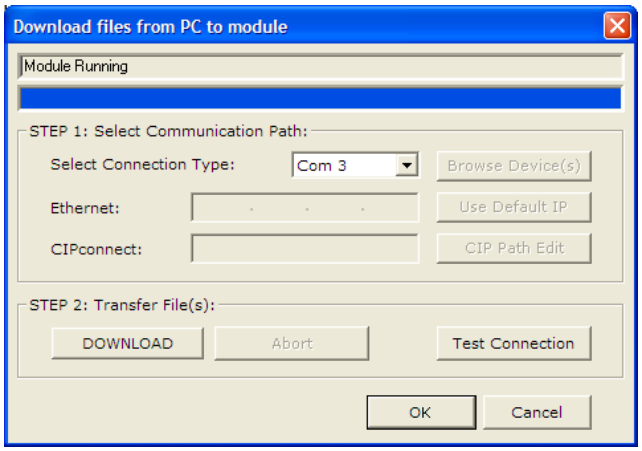
To download the project file

1. In the tree view in *ProSoft Configuration Builder*, click once to select the module.
2. Right-click the module icon to open a shortcut menu. From the shortcut menu, choose **DOWNLOAD FROM PC TO DEVICE**. The program will scan your PC for a valid com port (this may take a few seconds). When *PCB* has found a valid COM port, the *Download* dialog box will open.



3. Choose the COM port to use from the dropdown list, and then click the **DOWNLOAD** button.

   The module will perform a platform check to read and load its new settings. When the platform check is complete, the status bar in the *Download* dialog box will display the message *Module Running*.



3 Ladder Logic

In This Chapter

Ladder logic is required for the MVI56-101S module to work. Tasks that must be handled by the ladder logic are module data transfer, special block handling, and status data receipt. Additionally, a power-up handler may be needed to handle the initialization of the module's data and to clear any processor fault conditions.

The sample ladder logic is extensively commented, to provide information on the purpose and function of each rung. For most applications, the sample ladder will work without modification.

3.1 Module Data Object (ModuleDef870)

All data related to the MVI56-101S module is stored in a user defined data type. An instance of the data type is required before the module can be used. This is done by declaring a variable of the data type in the Controller Tags Edit Tags dialog box.

The following table describes the structure of this object.

Name	Data Type	Description
InStat	InStat870	Status information in each read block
ReadData	INT[600]	Data read from module
WriteData	INT[600]	Data to write to module
BP	Backplane870S	Data to handle backplane logic
ReadClock	Clock870	
WriteClock	Clock870	
CommandBits	CommandsBits870S	
PassEvents	Event870[10]	
EventCount	INT	

This object contains objects that define user and status data related to the module. Each of these object types is discussed in the following topics of the document.

3.1.1 Status Object (InStat870)

This object views the status of the module. The **InStat870** object shown below is updated each time a read block is received by the processor. Use this data to monitor the state of the module at a "real-time rate".

Name	Data Type	Description
PassCnt	INT	Program cycle counter for module
Product	INT[2]	Product code for module
Rev	INT[2]	Revision level of module's code
OP	INT[2]	Operating system version for module
Run	INT[2]	Run number for module
BlkErrs	BlkStat870	Data block transfer statistics
Read	INT	Total number of blocks read by the module
Write	INT	Total number of blocks written by the module
Parse	INT	Total number of blocks parsed by the module
Err	INT	Total number of block transfer errors
Port_Select	INT	Current port on module being utilized
Bad_CRC	INT	Total number of frames with bad CRC errors
Sync_Err	INT	Total number of sync errors
Len_Err	INT	Total number length errors
Timeouts	INT	Total number timeout errors
RX_Frames	INT	Total number RX frames
TX_Frames	INT	Total number TX frames

Refer to the Reference chapter for a complete listing of the data stored in this object.

3.1.2 User Data Objects

These objects hold data to be transferred between the processor and the MVI56-101S module. The user data is the read and write data transferred between the processor and the module as "pages" of data up to 200 words long.

Name	Data Type	Description
ReadData	INT[600]	
WriteData	INT[600]	

The read data (**ReadData**) is an array set to match the value entered in the **Read Register Count** parameter of the 101S.CFG file. For ease of use, this array should be dimensioned as an even increment of 200 words. This data is paged up to 200 words at a time from the module to the processor. The ReadData task places the data received into the proper position in the read data array. Use this data for status and control in the ladder logic of the processor.

The write data (**WriteData**) is an array set to match the value entered in the **Write Register Count** parameter of the 101S.CFG file. For ease of use, this array should be dimensioned as even increments of 200 words. This data is paged up to 200 words at a time from the processor to the module. The WriteData task places the write data into the output image for transfer to the module. This data is passed from the processor to the module for status and control information for use in other nodes on the network.

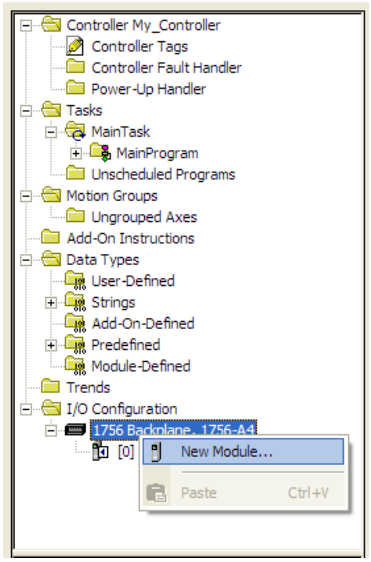
3.1.3 Block Control Data (Backplane870S)

This data object stores the variables required for the data transfer between the processor and the MVI56-101S module. The following table describes the structure of this object.

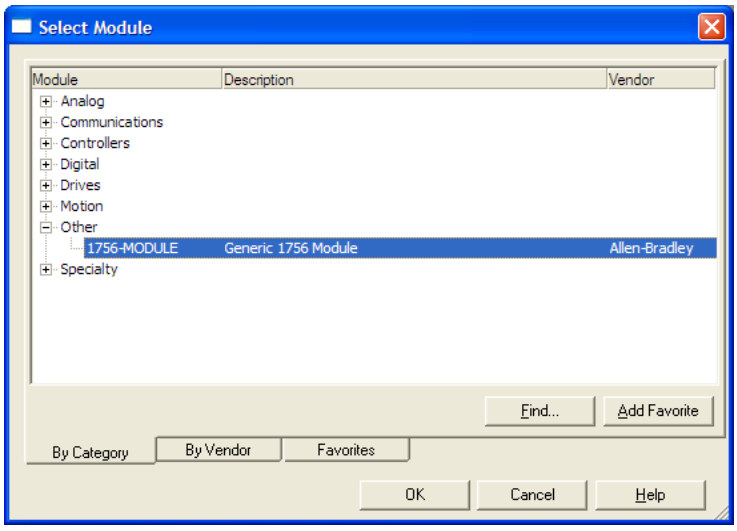
Name	Data Type	Description
LastRead	INT	Index of last read block
LastWrite	INT	Index of last write block
BlockIndex	INT	Computed block offset for data table

3.2 Adding the Module to an Existing Project

1 Select the *I/O Configuration* folder in the *Controller Organization* window of RSLogix 5000, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE**.



This action opens the *Select Module* dialog box:

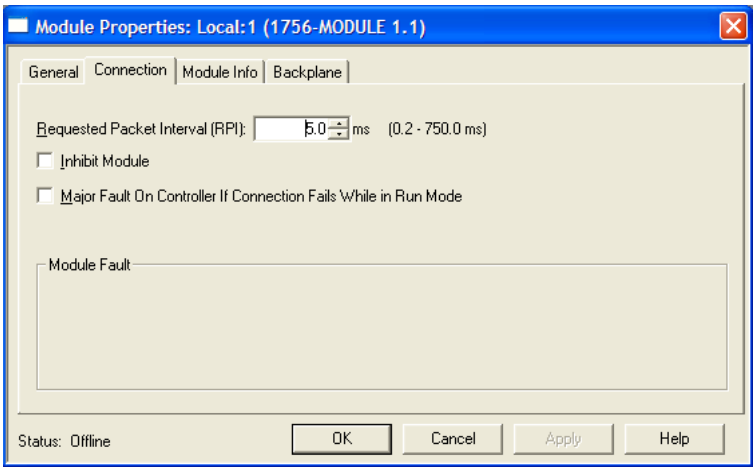


2 Select the **1756-MODULE (GENERIC 1756 MODULE)** from the list and click **OK**. This action opens the *New Module* dialog box.

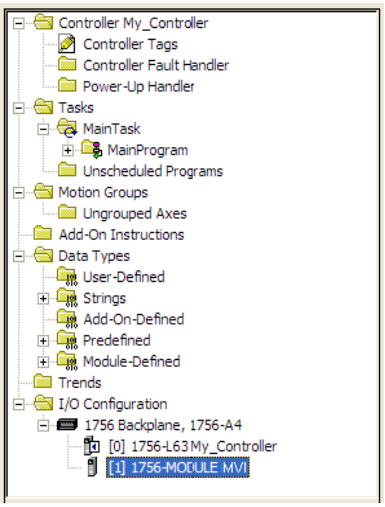
3 Enter the *Name, Description* and *Slot* options for your application. You must select the *Comm Format* as **DATA - INT** in the dialog box, otherwise the module will not communicate. Click **OK** to continue.

Parameter	Value
Name	Enter a module identification string. Example: **101S_2**
Description	Enter a description for the module. Example: **IEC 60870-5-101 SLAVE COMMUNICATION MODULE**
Comm Format	Select **DATA-INT**.
Slot	Enter the slot number in the rack where the MVI56-101S module is located.
Input Assembly Instance	**1**
Input Size	**250**
Output Assembly Instance	**2**
Output Size	**248**
Configuration Assembly Instance	**4**
Configuration Size	**0**

4 Select the *Requested Packet Interval* value for scanning the I/O on the module. This value represents the minimum frequency that the module will handle scheduled events. This value should not be set to less than **1** millisecond. The default value is **5** milliseconds. Values between **1** and **10** milliseconds should work with most applications.



5 Save the module. Click **OK** to dismiss the dialog box. The *Controller Organization* window now displays the module's presence.



6 Copy the *User-Defined Data Types* from the sample program into your existing RSLogix 5000 project.
7 Copy the *Controller Tags* from the sample program into your project.
8 Copy the *Ladder Rungs* from the sample program into your project.

4 Diagnostics and Troubleshooting

In This Chapter

The module provides information on diagnostics and troubleshooting in the following forms:

- Status data values can be transferred from the module to processor memory and can be monitored there manually or by customer-created logic.
- Status data contained in the module can be viewed through the Configuration/Debug port, using the troubleshooting and diagnostic capabilities of *ProSoft Configuration Builder (PCB)*.
- LED status indicators on the front of the module provide general information on the module's status.

4.1 Reading Status Data from the Module

The MVI56-101S module returns a 20-word Status Data block that can be used to determine the module's operating status. This data can be located in the module's database at registers at the location specified in the configuration. This data is transferred to the ControlLogix processor continuously with each read block.

4.1.1 Using ProSoft Configuration Builder (PCB) for Diagnostics

The *Configuration and Debug* menu for this module is arranged as a tree structure, with the *Main* menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the *Main* menu.

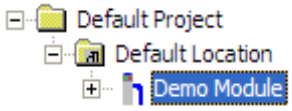
Because this is a text-based menu system, you enter commands by typing the [command letter] from your computer keyboard in the *Diagnostic* window in *ProSoft Configuration Builder (PCB)*. The module does not respond to mouse movements or clicks. The command executes as soon as you press the **[COMMAND LETTER]** — you do not need to press **[ENTER].** When you type a **[COMMAND LETTER]**, a new screen will be displayed in your terminal application.

Using the Diagnostic Window in ProSoft Configuration Builder

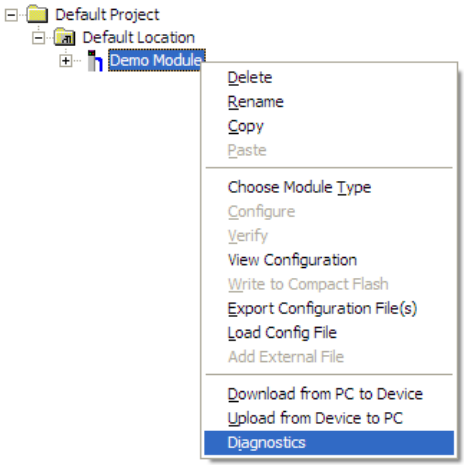
Tip: You can have a ProSoft Configuration Builder Diagnostics window open for more than one module at a time.

To connect to the module's Configuration/Debug serial port

1 Start *PCB*, and then select the module to test. Click the right mouse button to open a shortcut menu.

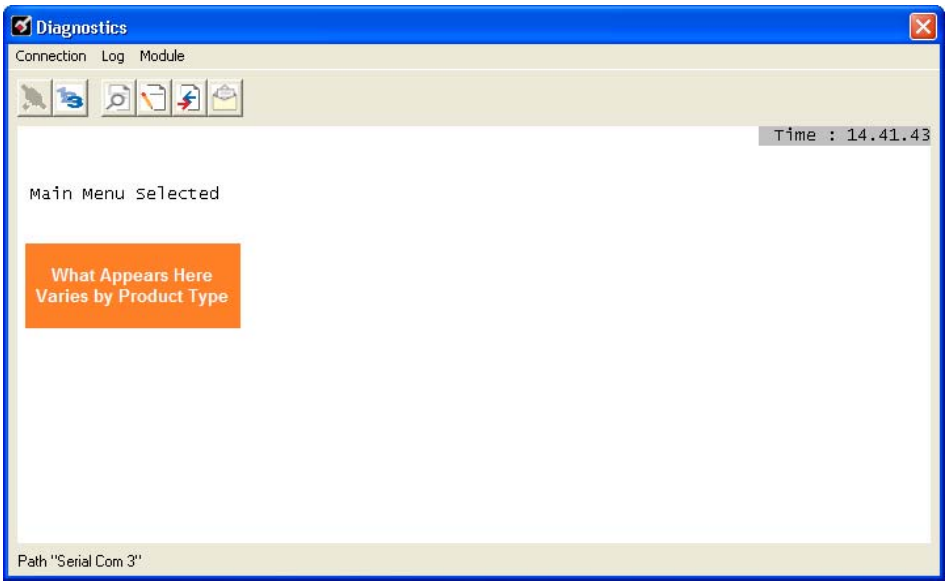


2 On the shortcut menu, choose **DIAGNOSTICS**.



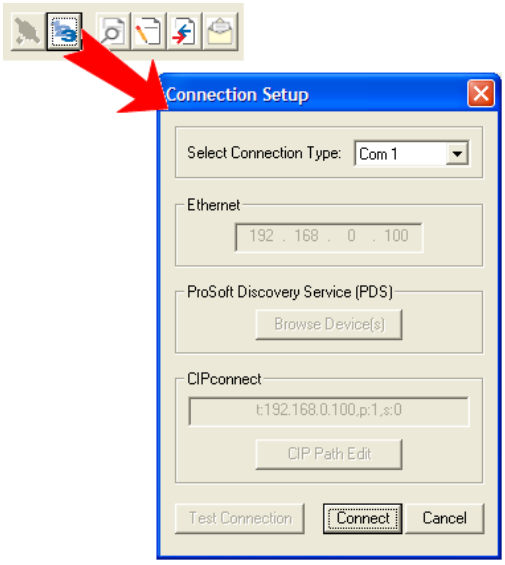
This action opens the *Diagnostics* dialog box.

3 Press **[?]** to open the *Main* menu.



If there is no response from the module, follow these steps:

1 Click to configure the connection. On the *Connection Setup* dialog box, select a valid com port or other connection type supported by the module.



2 Verify that the null modem cable is connected properly between your computer's serial port and the module. A regular serial cable will not work.

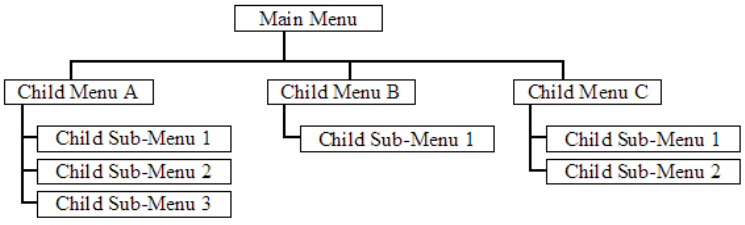
3 On computers with more than one serial port, verify that your communication program is connected to the same port that is connected to the module.

If you are still not able to establish a connection, contact ProSoft Technology for assistance.

Navigation

All of the submenus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a submenu to the next higher menu by pressing **[M]** on your keyboard.

The organization of the menu structure is represented in simplified form in the following illustration:



The remainder of this section shows the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are usually not case sensitive. You can enter most commands in lowercase or uppercase letters.

The menus use a few special characters (**?**, **-**, **+**, @) that must be entered exactly as shown. Some of these characters will require you to use the **SHIFT**, **CTRL**, or **ALT** keys to enter them correctly. For example, on US English keyboards, enter the **?** command as **SHIFT** and */*.

Also, take care to distinguish the different uses for uppercase letter "eye" (**I**), lowercase letter "el" (**L**), and the number one (**1**). Likewise, uppercase letter "oh" (**O**) and the number zero (**0**) are not interchangeable. Although these characters look alike on the screen, they perform different actions on the module and may not be used interchangeably.

4.1.2 Main Menu

When you first connect to the module from your computer, your terminal screen will be blank. To activate the main menu, press the **[?]** key on your computer's keyboard. If the module is connected properly, the following menu will appear.

```
IEC-870-5-101 SLAVE COMMUNICATION MODULE MENU
  ?=Display Menu
  A=Data Analyzer
  B=Block Transfer Statistics
  C=Module Configuration
  D=Database View
  E=Display Program Status
  F=Priority Queues
  G=Invalid State Monitoring
  P=Port Cfg
  R=Receive Configuration File
  S=Send Configuration File
  V=Version Information
  W=Warm Boot Module
  1 = M_SP_NA Setup          2 = M_DP_NA Setup
  3 = M_ST_NA Setup          4 = M_ME_NA Setup
  5 = M_ME_NB Setup          6 = M_IT_NA Setup
  ^ = M_ME_NC Setup
  7 = C_SC_NA Setup          8 = C_DC_NA Setup
  9 = C_RC_NA Setup          0 = C_SE_NA Setup
  ! = C_SE_NB Setup          $ = C_SE_NC Setup
  @ = IEC-870 Database Cfg   Esc=Exit Program
```

Caution: Some of the commands available to you from this menu are designed for advanced debugging and system testing only, and can cause the module to stop communicating with the processor or with other devices, resulting in potential data loss or other communication failures. Use these commands only if you fully understand their potential effects, or if you are specifically directed to do so by ProSoft Technology Technical Support Engineers.
There may be some special command keys that are not listed on the menu but that may activate additional diagnostic or debugging features. If you need these functions, you will be advised how to use them by Technical Support. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

Viewing Block Transfer Statistics

Press **[B]** from the *Main* menu to view the *Block Transfer Statistics* screen.

Use this command to display the configuration and statistics of the backplane data transfer operations between the module and the processor. The information on this screen can help determine if there are communication problems between the processor and the module.

Tip: To determine the number of blocks transferred each second, mark the numbers displayed at a specific time. Then some seconds later activate the command again. Subtract the previous numbers from the current numbers and divide by the quantity of seconds passed between the two readings.

Opening the Data Analyzer Menu

Press **[A]** to open the Data Analyzer Menu. Use this command to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Refer to Data Analyzer (page 88) for more information about this menu.

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Viewing Module Configuration

Press **[C]** to view the *Module Configuration* screen.

Use this command to display the current configuration and statistics for the module.

Opening the Database Menu

Press **[D]** to open the Database View menu. Use this menu command to view the current contents of the module's database.

Viewing Error and Status Data

Press **[E]** to display the error/status data for the module.

Viewing Port Configuration

Press **[6]** or **[7]** from the Main Menu to view configuration information for ports 1 and 2.

Use this command to display detailed configuration information for the selected port.

Receiving the Configuration File

Press **[R]** to download (receive) the current configuration file from the module.

Sending the Configuration File

Press **[S]** to upload (send) a configuration file from the module to your PC.

Viewing Version Information

Press **[V]** to view version information for the module.

Use this command to view the current version of the software for the module, as well as other important values. You may be asked to provide this information when calling for technical support on the product.

Values at the bottom of the display are important in determining module operation. The *Program Scan Counter* value is incremented each time a module's program cycle is complete.

Tip: Repeat this command at one-second intervals to determine the frequency of program execution.

Warm Booting the Module

Press **[W]** from the *Main* menu to warm boot (restart) the module.

This command will cause the program to exit and reload, refreshing configuration parameters that must be set on program initialization. Only use this command if you must force the module to reboot.

Viewing Data Type Setup

To view setup information for each data type, press the matching key from the Menu.

Key	Data Type	Screen Example
1	M_SP_NA	M_SP_NA Setup Menu Selected M_SP_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value 0 11 0 80000001 0 1 12 1 80000001 0
2	M_DP_NA	M_DP_NA Setup Menu Selected M_DP_NA Setup (0 to 2) Index Point# DB Addr Group(s) Bits 0 21 16 00000002 0 0 1 22 17 00000002 0 0
3	M_ST_NA	M_ST_NA Setup Menu Selected M_ST_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value 0 31 6 00000004 0 1 32 7 00000004 0
4	M_ME_NA	M_ME_NA Setup Menu Selected M_ME_NA Setup (0 to 2) Index Point# DB Addr Group(s) Deadband Value Norm. Val 0 41 4 00000008 0 0 0.000000 1 42 5 00000008 0 0 0.000000
5	M_ME_NB	M_ME_NB Setup Menu Selected M_ME_NB Setup (0 to 2) Index Point# DB Addr Group(s) Deadband Value 0 51 6 00000010 1 0 1 52 7 00000010 1 0
6	M_IT_NA	M_IT_NA Setup Menu Selected M_IT_NA Setup (0 to 2) Index Point# DB Addr Group(s) Value FROZEN 0 61 4 00020000 0 0 1 62 5 00040000 0 0
Shift 6	M_ME_NC	M_ME_NC Setup (0 to 10) Index Point# DB Addr Group(s) Deadband Value 0 600 25 00000002 1.0000E+02 0.0000E+00 1 601 26 00000002 1.0000E+02 0.0000E+00
7	C_SC_NA	C_SC_NA Setup Menu Selected C_SC_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 700 1600 2000 3200 0 0 1 701 1601 2010 3201 0 0
8	C_DC_NA	C_DC_NA Setup Menu Selected C_DC_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Bits 0 800 1616 3000 3216 0 0 0 1 801 1617 3090 3217 0 0 0

Key	Data Type	Screen Example
9	C_RC_NA	C_RC_NA Setup Menu Selected C_RC_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 900 51 8000 30 0 0 1 901 52 8888 31 0 0
0	C_SE_NA	C_SE_NA Setup Menu Selected C_SE_NA Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value Norm 0 1000 105 0 0 0 0 0.0 1 1001 106 0 0 0 0 0.0
Shift 1	C_SE_NB	C_SE_NB Setup Menu Selected C_SE_NB Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 1100 107 0 0 0 0 1 1101 108 0 0 0 0
Shift 4	C_SE_NC	C_SE_NC Setup Menu Selected C_SE_NC Setup (0 to 2) Index Point# DB Addr MPnt# MPntDb ReqSel Value 0 1200 55 0 0 0 0 1 1201 56 0 0 0 0

Viewing IEC 60870 Database Configuration

Press **[@]** to view the database size configuration information.

```
IEC-870-5-101 DATABASE CONFIGURATION:
 PMENA Offs: 2000    PMENB Off: 2000
 M_SP_NA point count = 10       Event Scanning Enabled : Yes
 M_DP_NA point count = 10       Event Scanning Enabled : Yes
 M_ST_NA point count = 10       Event Scanning Enabled : Yes
 M_ME_NA point count = 10       Event Scanning Enabled : Yes
 M_ME_NB point count = 10       Event Scanning Enabled : Yes
 M_IT_NA point count = 4
 C_SC_NA point count = 10
 C_DC_NA point count = 10
 C_RC_NA point count = 10
 C_SE_NA point count = 10
 C_SE_NB point count = 10
```

Exiting the Program

Press **[ESC]** to restart the module and force all drivers to be loaded. The module will use the configuration stored in the module's flash memory to configure the module.

4.1.3 Database View Menu

Press **[D]** from the *Main* menu to open the *Database View* menu. Use this menu command to view the current contents of the module database. Press **[?]** to view a list of commands available on this menu.

```
DB Menu Selected

DATABASE VIEW MENU
 ?=Display Menu
 0-9=Display 0-9000
 S=Show Again
 -=Back 5 Pages
 P=Previous Page
 +=Skip 5 Pages
 N=Next Page
 D=Decimal Display
 H=Hexadecimal Display
 F=Float Display
 A=ASCII Display
 M=Main Menu
```

Viewing Register Pages

To view sets of register pages, use the keys described below:

Command	Description
[0]	Display registers 0 to 99
[1]	Display registers 1000 to 1099
[2]	Display registers 2000 to 2099

And so on. The total number of register pages available to view depends on your module's configuration.

Displaying the Current Page of Registers Again

Press **[S]** from the *Database View* menu to show the current page of registers again.

```
DATABASE DISPLAY 0 TO 99 (DECIMAL)
   100   101   102     4     5     6     7     8     9    10
    11    12    13    14    15    16     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0     0     0
```

This screen displays the current page of 100 registers in the database.

Moving Back Through 5 Pages of Registers

Press **[-]** from the *Database View* menu to skip five pages back in the database to see the 100 registers of data starting 500 registers before the currently displayed page.

Moving Forward (Skipping) Through 5 Pages of Registers

Press **[+]** from the *Database View* menu to skip five pages ahead in the database to see the 100 registers of data starting 500 registers after the currently displayed page.

Viewing the Previous Page of Registers

Press **[P]** from the *Database View* menu to display the previous page of data.

Viewing the Next Page of Registers

Press **[N]** from the *Database View* menu to display the next page of data.

Viewing Data in Decimal Format

Press **[D]** from the *Database View* menu to display the data on the current page in decimal format.

Viewing Data in Hexadecimal Format

Press **[H]** from the *Database View* menu to display the data on the current page in hexadecimal format.

Viewing Data in Floating-Point Format

Press **[F]** from the *Database View* menu to display the data on the current page in floating-point format. The program assumes that the values are aligned on even register boundaries. If floating-point values are not aligned as such, they are not displayed properly.

Viewing Data in ASCII (Text) Format

Press **[A]** from the *Database View* menu to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Returning to the Main Menu

Press **[M]** to return to the *Main* menu.

4.1.4 Data Analyzer

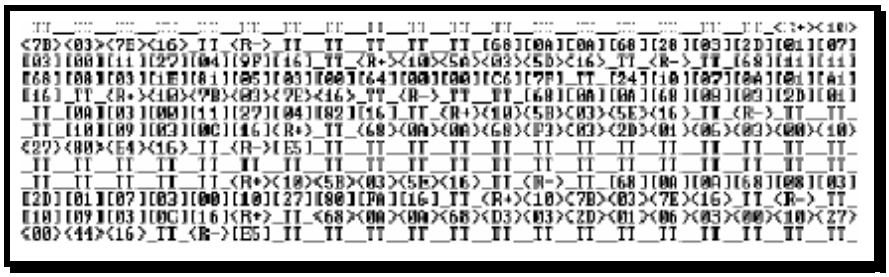
The data analyzer mode allows you to view all bytes of data transferred on each port. Both the transmitted and received data bytes are displayed. Use of this feature is limited without a thorough understanding of the protocol.

Note: The Port selection commands on the Data Analyzer menu differs very slightly in different modules, but the functionality is basically the same. Use the illustration above as a general guide only. Refer to the actual data analyzer menu on your module for the specific port commands to use.

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Analyzing Data for the first application port

Press **[1]** to display I/O data for the first application port in the Data Analyzer. The following illustration shows an example of the Data Analyzer output.



Analyzing Data for the second application port

Press **[2]** to display I/O data for the second application port in the Data Analyzer.

Displaying Timing Marks in the Data Analyzer

You can display timing marks for a variety of intervals in the data analyzer screen. These timing marks can help you determine communication-timing characteristics.

Key	Interval
[5]	1 milliseconds ticks
[6]	5 milliseconds ticks
[7]	10 milliseconds ticks
[8]	50 milliseconds ticks
[9]	100 milliseconds ticks
[0]	Turn off timing marks

Removing Timing Marks in the Data Analyzer

Press **[0]** to turn off timing marks in the Data Analyzer screen.

Viewing Data in Hexadecimal Format

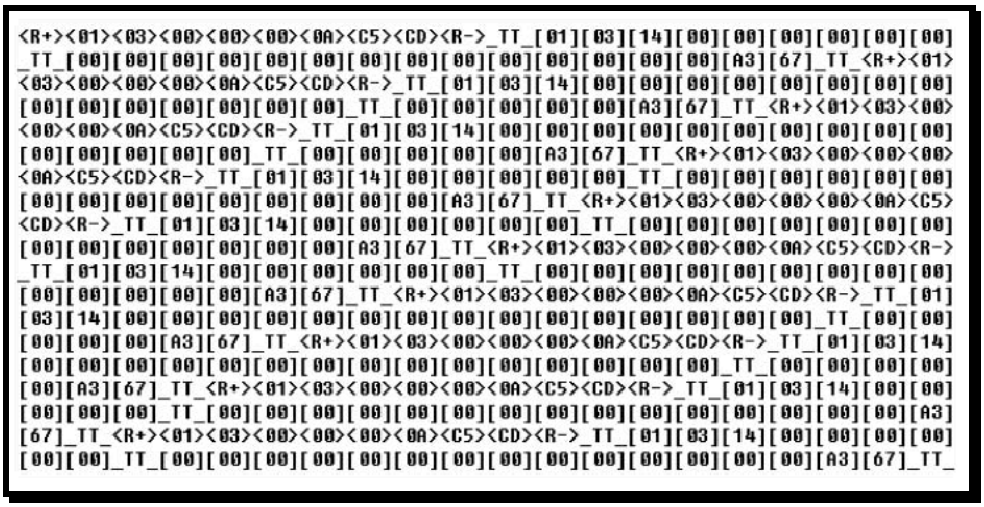
Press **[H]** from the *Database View* menu to display the data on the current page in hexadecimal format.

Viewing Data in ASCII (Text) Format

Press **[A]** from the *Database View* menu to display the data on the current page in ASCII format. This is useful for regions of the database that contain ASCII data.

Starting the Data Analyzer

Press **[B]** to start the data analyzer. After the key is pressed, all data transmitted and received on the currently selected port will be displayed. The following illustration shows an example.



The Data Analyzer displays the following special characters:

Character	Definition
[ ]	Data enclosed in these characters represent data received on the port.
< >	Data enclosed in these characters represent data transmitted on the port.
<R+>	These characters are inserted when the RTS line is driven high on the port.
<R->	These characters are inserted when the RTS line is dropped low on the port.
<CS>	These characters are displayed when the CTS line is recognized high.
_TT_	These characters are displayed when the timing mark interval has been reached. This parameter is user defined.

Stopping the Data Analyzer

Press **[S]** to stop the data analyzer. Use this option to freeze the display so the data can be analyzed. To restart the analyzer, press **[B].**

Important: When in analyzer mode, program execution will slow down. Only use this tool during a troubleshooting session. Before disconnecting from the Config/Debug port, please press [S] to stop the data analyzer, and then press [M] to return to the main menu. This action will allow the module to resume its normal high speed operating mode.

Data Analyzer Tips

From the main menu, press **[A]** for the "Data Analyzer". You should see the following text appear on the screen:

```
Data Analyzer Mode Selected
```

After the "Data Analyzer" mode has been selected, press **[?]** to view the Data Analyzer menu. You will see the following menu:

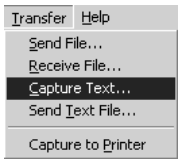
```
DATA ANALYZER VIEW MENU
 ?=Display Menu
 1=Select Port 1
 2=Select Port 2
 5=1 mSec Ticks
 6=5 mSec Ticks
 7=10 mSec Ticks
 8=50 mSec Ticks
 9=100 mSec Ticks
 0=No mSec Ticks
 H=Hex Format
 A=ASCII Format
 B=Start
 S=Stop
 M=Main Menu

 Port = 1, Format=HEX, Tick=10
```

From this menu, you can select the "Port", the "format", and the "ticks" that you can display the data in.

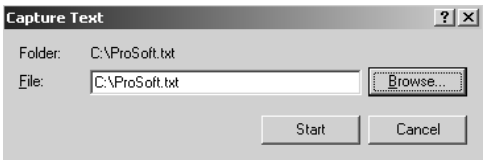
For most applications, HEX is the best format to view the data, and this does include ASCII based messages (because some characters will not display on HyperTerminal and by capturing the data in HEX, we can figure out what the corresponding ASCII characters are supposed to be).

The Tick value is a timing mark. The module will print a _TT for every xx milliseconds of no data on the line. Usually 10milliseconds is the best value to start with.

After you have selected the Port, Format, and Tick, we are now ready to start a capture of this data. The easiest way to do so is to go up to the top of you HyperTerminal window, and do a **TRANSFER / CAPTURE TEXT** as shown below:



After selecting the above option, the following window will appear:



Next name the file, and select a directory to store the file in. In this example, we are creating a file ProSoft.txt and storing this file on our root C: drive. After you have done this, press the Start button.

Now you have everything that shows up on the HyperTerminal screen being logged to a file called ProSoft.txt. This is the file that you will then be able to email to ProSoft Technical Support to assist with issues on the communications network.

To begin the display of the communications data, you will then want to press **[B]** to tell the module to start printing the communications traffic out on the debug port of the module. After you have pressed **[B],** you should see something like the following:

```
[03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+><01><02>
<00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT__TT_<R+>
<01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT__TT_<R+>
<01><01><00><00><00><A0><3C><72><R->_TT__TT_[01][01][14][00][00][01][00][02]_TT_
[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]_TT__TT_<R+>
<01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00][01][00]_TT_
[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]_TT__TT_<R+>
<01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02][00][00][B9][B8]_TT_
_TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00][00][01]
[00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD][51]_TT_
_TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00][00][01]
[00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00][B7][52]
_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00][00][00]
[01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][FB][B7]
_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT_[01][02][02][00][00][B9]
[B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_[01][03][14][00][00]
[00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09][CD]
[51]_TT__TT_<R+><01><01><00><00><00><A0><3C><72><R->_TT__TT__TT_[01][01][14][00]
[00][01][00][02]_TT_[00][03][00][04][00][05][00][06][00][07][00][08][00][09][00]
[B7][52]_TT__TT_<R+><01><04><00><00><00><0A><70><0D><R->_TT__TT_[01][04][14][00]
[00][00][01][00]_TT_[02][00][03][00][04][00][05][00][06][00][07][00][08][00][09]
[FB][B7]_TT__TT_<R+><01><02><00><00><00><0A><F8><0D><R->_TT__TT__TT_[01][02][02]
[00][00][B9][B8]_TT__TT_<R+><01><03><00><00><00><0A><C5><CD><R->_TT__TT_
```

The <R+> means that the module is transitioning the communications line to a transmit state.

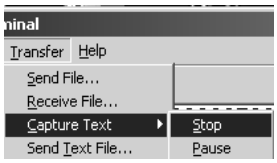
All characters shown in <> brackets are characters being sent out by the module.

The <R-> shows when the module is done transmitting data, and is now ready to receive information back.

And finally, all characters shown in the [ ] brackets is information being received from another device by the module.

After taking a minute or two of traffic capture, you will now want to stop the "Data Analyzer". To do so, press the [S] key, and you will then see the scrolling of the data stop.

When you have captured the data you want to save, open the Transfer menu and choose Capture Text. On the secondary menu, choose Stop.



You have now captured, and saved the file to your PC. This file can now be used in analyzing the communications traffic on the line, and assist in determining communication errors.

Returning to the Main Menu

Press **[M]** to return to the *Main* menu.

4.2 LED Indicators

The LEDs indicate the module's operating status as follows:

LED	Color	Status	Indication
CFG	Green	On	Data is being transferred between the module and a remote terminal using the Configuration/Debug port.
		Off	No data is being transferred on the Configuration/Debug port.
P1	Green	On	Data is being transferred between the module and the network on Port 1.
		Off	No data is being transferred on the port.
P2	Green	On	Data is being transferred between the module and the network on Port 2.
		Off	No data is being transferred on the port.
APP	Amber	Off	The MVI56-101S is working normally.
		On	The MVI56-101S module program has recognized a communication error on one of its ports.
BP ACT	Amber	On	The LED is on when the module is performing a write operation on the backplane.
		Off	The LED is off when the module is performing a read operation on the backplane. Under normal operation, the LED should blink rapidly on and off.
OK	Red/ Green	Off	The card is not receiving any power and is not securely plugged into the rack.
		Green	The module is operating normally.
		Red	The program has detected an error or is being configured. If the LED remains red for over 10 seconds, the program has probably halted. Remove the card from the rack and re-insert the card to restart the module's program.
BAT	Red	Off	The battery voltage is OK and functioning.
		On	The battery voltage is low or battery is not present. Allow battery to charge by keeping module plugged into rack for 24 hours. If BAT LED still does not go off, contact ProSoft Technology, as this is not a user serviceable item.

If the APP, BP ACT and OK LEDs blink at a rate of every one-second, this indicates a serious problem with the module. Call ProSoft Technology support to arrange for repairs.

4.2.1 Clearing a Fault Condition

Typically, if the OK LED on the front of the module turns RED for more than ten seconds, a hardware problem has been detected in the module or the program has exited.

To clear the condition, follow these steps:

1. Turn off power to the rack.
2. Remove the card from the rack.
3. Verify that all jumpers are set correctly.
4. If the module requires a Compact Flash card, verify that the card is installed correctly.
5. Re-insert the card in the rack and turn the power back on.
6. Verify correct configuration data is being transferred to the module from the ControlLogix processor controller.

If the module's OK LED does not turn GREEN, verify that the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology Technical Support.

4.2.2 Troubleshooting

Use the following troubleshooting steps if you encounter problems when the module is powered up. If these steps do not resolve your problem, please contact ProSoft Technology Technical Support.

Processor Errors

Problem description	Steps to take
Processor fault	Verify that the module is plugged into the slot that has been configured for the module in the I/O Configuration of RSLogix. Verify that the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI56-101S. Verify that all modules in the rack are correctly configured in the ladder logic.

Module Errors

Problem description	Steps to take
BP ACT LED (not present on MVI56E modules) remains OFF or blinks slowly MVI56E modules with scrolling LED display: <*Backplane Status*> condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the Configuration/Debug port to see if the module is running. If the program has halted, turn off power to the rack, remove the card from the rack and re-insert it, and then restore power to the rack.

5 Reference

In This Chapter

5.1 Product Specifications

The MVI56 IEC 60870-5-101 Slave Communication Module allows Rockwell Automation ControlLogix processors to interface easily with IEC 60870-5-101 protocol compatible hosts.

The MVI56-101S module is the fastest and easiest way to add IEC 60870-5-101 protocol interface support to the ControlLogix processor platform. It is a single-slot, backplane compatible solution. It also has two powerful and highly configurable redundant IEC 60870-5-101 slave ports, allowing the many SCADA and field devices supporting the IEC protocol to be integrated into the ControlLogix processor platform.

The standards used in developing the product are listed in the following table.

Publication	Title
IEC 60870-5-101	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-101 Amendment 1	Companion Standard for Basic Telecontrol Tasks
IEC 60870-5-1	Transmission Frame Formats
IEC 60870-5-2	Link Transmission Procedures
IEC 60870-5-3	General Structure of Application Data
IEC 60870-5-4	Definition and Coding of Application Information Elements
IEC 60870-5-5	Basic Application Functions

Refer to these standards for any questions on the protocol supported.

5.1.1 General Specifications

- Single Slot - 1756 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.
- Configuration data obtained from configuration text file downloaded to module. Sample configuration file included
- Local or remote rack

5.1.2 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 Vdc 3 mA @ 24 Vdc
Operating Temperature	0°C to 60°C (32°F to 140°F)
Storage Temperature	-40°C to 85°C (-40°F to 185°F)
Shock	30 g operational 50 g non-operational Vibration: 5 g from 10 Hz to 150 Hz
Relative Humidity	5% to 95% (with no condensation)
LED Indicators	Module Status Backplane Transfer Status Application Status Serial Activity
Debug/Configuration port (CFG)	
CFG Port (CFG)	RJ45 (DB-9M with supplied cable) RS-232 only
Application ports (PRT1 & PRT2) **Full hardware handshaking control, providing radio, modem and multi-drop support**	
Software configurable communication parameters	Baud rate: 110 to 38,400 baud RS-232 and 422 Parity: none, odd or even Data bits: 5, 6, 7, or 8 Stop bits: 1 or 2 RTS on/off delay: 0 to 65535 milliseconds
App Ports (P1,P2) (Serial modules)	RJ45 (DB-9M with supplied cable) RS-232 handshaking configurable 500 V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port 6-foot RS-232 configuration cable

5.1.3 Functional Specifications

The MVI56-101S module accepts commands from an attached master unit. A port configured as a virtual slave permits a remote master to interact with all data contained in the module. This data can be derived from the ControlLogix processor. The remote master device uses the fully-configured databases in the module to control outputs and monitor inputs. The module can operate in balanced or unbalanced mode.

- Supports time stamp events
- Supports time and data synchronization from a master or the processor
- Supports monitored data
- Event queue supports 99 points for each data type
- Reports events by configurable priority order
- Order monitored points by interrogation groups
- Configurable deadband for monitored measured points
- Supports Master Class 1 and Class 2 polls with configurable parameters
- Acknowledgement transmission is handled internally by the module
- Configurable data link address, Common ASDU address and Information Object Address
- Configurable pulse duration

5.2 Functional Overview

Important: The MVI56-101S module complies with section 6.2.2.5 of the Norwegian User Convention. Users should not assume that the entire Norwegian User Convention specifications are supported.

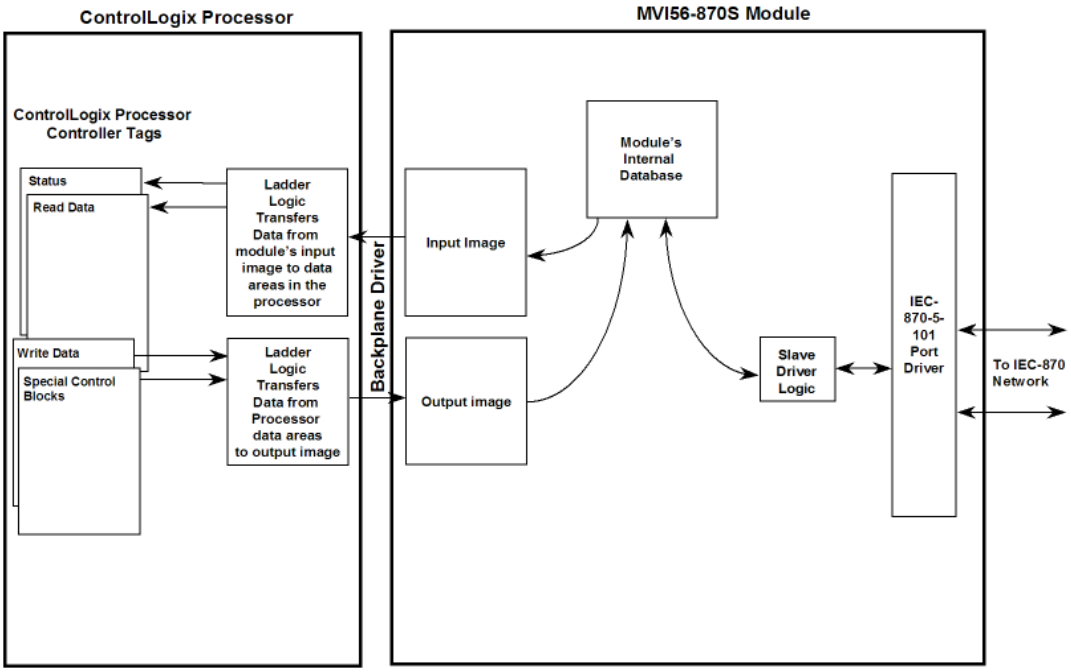
5.2.1 Backplane Data Transfer

The MVI56-101S module communicates directly over the ControlLogix processor backplane. Data is paged between the module and the ControlLogix processor across the backplane using the module's input and output images. The update frequency of the images is determined by the scheduled scan rate defined by the user for the module and the communication load on the module. Typical updates are in the range of 1 to 10 milliseconds.

This bi-directional transference of data is accomplished by the module filling in data in the module's input image to send to the processor. Data in the input image is placed in the Controller Tags in the processor by the ladder logic. The input image for the module is set to 250 words. This large data area permits fast throughput of data between the module and the processor.

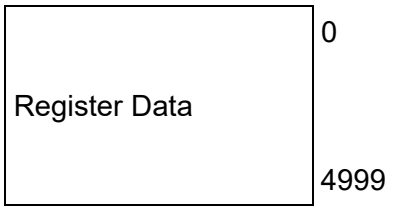
The processor inserts data into the module's output image to transfer to the module. The module's program extracts the data and places it in the module's internal database. The output image for the module is set to 248 words. This large data area permits fast throughput of data from the processor to the module

The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56-101S module and the IEC 60870-5-101 network.



All data transferred between the module and the processor over the backplane is through the input and output images. Ladder logic must be written in the ControlLogix processor to interface the input and output image data with data defined in the Controller Tags. All data used by the module is stored in its internal database. The following illustration shows the layout of the database:

Module's Internal Database Structure



Data contained in this database is paged through the input and output images by coordination of the ControlLogix processor ladder logic and the MVI56-101S module's program. Up to 248 words of data can be transferred from the module to the processor at a time. Up to 247 words of data can be transferred from the processor to the module. The read and write block identification codes in each data block determine the function to be performed or the content of the data block. The module uses the following block numbers:

Block Range	Descriptions
-1	Null Block
0	Null Block
1 to 25	Read or Write Data
1000 to 1024	Request Output Data from Processor
9958	Event Messages from CLX
9970	Set PLC time using module's time
9971	Set module's time using CLX time
9998	Warm-boot control block
9999	Cold-boot control block

Each image has a defined structure depending on the data content and the function of the data transfer.

5.2.2 Normal Data Transfer

Normal data transfer includes the paging of the user data found in the module's internal database in registers 0 to 4999 and the status data. These data are transferred through read (input image) and write (output image) blocks. Refer to the **Module Configuration** and **Ladder Logic** sections for a description of the data objects used with the blocks and the ladder logic required. The structure and function of each block is discussed in the following topics.

Read Block

This block transfers IEC output (control) data sent from the remote IEC 104 Client through the module to the ControlLogix processor. The following table outlines the structure of data contained in the input image of a normal *Read Block*.

Offset	Description	Length
0	Reserved	1
1	Write Block ID	1
2 to 201	Read Data	200
202	Program Scan Counter	1
203 to 204	Product Code	2
205 to 206	Product Version	2
207 to 208	Operating System	2
209 to 210	Run Number	2
211 to 214	Data Transfer Status	4
215	Port Select	1
216	Bad Checksum Count	1
217	Sync Error Count	1
218	Length Error Count	1
219	Timeout Error Count	1
220	RX Frame Count	1
221	TX Frame Count	1
222 to 248	Spare	27
249	Read Block ID	1

The *Read Block ID* is an index value sent by the module and used by the ladder logic to determine where the current 200-word block of input data should be placed in the ControlLogix processor *IEC870S.ReadData* controller tag array. Each input data block transfer can move up to 200 words of user application data (*Read Block* offsets 2 to 201) from the module to the processor. In addition to moving user data, the block also contains module status data. This status data is transferred with each new input image block and can be used for module diagnostics.

Block Request from Processor to Module

This block transfers IEC input (monitor) data to be sent to the remote IEC 104 Client through the module from the ControlLogix processor. The following table outlines the structure of data contained in the output image of a normal *Write Block*.

Offset	Description	Length
0	Write Block ID	1
1 to 200	Write Data	200
201 to 247	Spare	47

The *Write Block ID* is sent in the *Read Block* as an index value used by the ladder logic to determine where the current 200-word block of output data should be taken from the ControlLogix processor *IEC870S.WriteData* controller tag array. Each output data block transfer can move up to 200 words of user application data (*Write Block* offsets 1-200) from the processor to the module.

During normal operation, the module sequentially sends *Read Blocks* and requests *Write Blocks*. For example, if the application uses three read and two write blocks, the sequence will be as follows:

R1W1→R2W2→R3W1→R1W2→R2W1→R3W2→R1W1→

This sequence will continue until interrupted by special function write block numbers sent by the controller or operator control through the module's Configuration/Debug port.

5.2.3 Output Data Initialization

The module can be configured to have the output data in the module set to that stored in the ControlLogix processor. This feature requires ladder logic support. When the module performs a restart operation, it will send a set of one or more special block requests to the processor. These blocks must be handled by the ladder logic and return the requested output data. The format of the blocks used by the module to request this data has the following format.

Offset	Description	Length
0	Reserved	1
1	1000 to 1024	1
2 to 248	Spare	247
249	1000 to 1024	1

The value in the block at offset 249 represents the output data set required. The module will only request the data set in the user configuration. Each block requested must receive a response block from the processor. The format of the response block is as follows:

Offset	Description	Length
0	1000 to 1024	1
1 to 200	Output Data	200
201 to 247	Spare	47

The module places the received output data in the block into the module's database.

5.2.4 Sending Events

The module operates by sending data when the master sends Class 2 polls. When the slave has an event, it sends the event to the module queue. If the communication mode is unbalanced, the slave notifies the master that it has an event to be sent and the master's next poll should be a Class 1 poll. The slave then replies to this poll sending the first event to the queue.

There are two ways of sending timestamp events to the module queue. The first is when a configured point changes its value in the module's database. This is actually the way the module would normally send events.

The second method is by using block 9958 (refer to the next section), in which case ladder logic should be used to send events in this manner. This method is limited due to the number of events that can be sent to the master. The important issue about events is that the slave should have its clock synchronized with the master. The master should send a Sync. Command to the MVI56-101S module in order to synchronize both clocks. You can also synchronize the module and the processor clocks using blocks 9970 and 9971 (Refer to the next section).

5.2.5 Command Control Blocks

Command control blocks are special blocks used to control the module. The current version of the software supports five command control blocks.

Event Messages

Block 9958 identification code sends event messages from the ControlLogix processor

Block Format for Write

Word Offset in Block	Data Fields	Description
0	Block ID	This field contains the value of 9958 identifying the block type to the module.
1	Event Count	Number of events present in the block. This field can have a value from 1 to 10.
2 to 16	Event #1	Event data to add to event message queue.
17 to 31	Event #2	Event data to add to event message queue.
32 to 46	Event #3	Event data to add to event message queue.
47 to 61	Event #4	Event data to add to event message queue.
62 to 76	Event #5	Event data to add to event message queue.
77 to 91	Event #6	Event data to add to event message queue.
92 to 106	Event #7	Event data to add to event message queue.
107 to 121	Event #8	Event data to add to event message queue.
122 to 136	Event #9	Event data to add to event message queue.
137 to 151	Event #10	Event data to add to event message queue.
152 to 247	Not Used	Not Used

The structure of each event record in the block is shown in the following table.

Word Offset in Event Record	Data Field(s)	Description
0	DB Index	This is the index for the point in the module's database. This corresponds to the order of point definition for the module data types. This is not the point address for the event.
1	ASDU	This is the ASDU data type for the event message. Valid entries for this field are as follows: 1=single point 3=double-point 5=step 9-normalized 11=scaled 15= integrated total
2	Qualifier	This is the qualifier code for the event message. Refer to the IEC protocol specification for a full listing of qualifier codes for each ASDU type.
3	Year	This field contains the four-digit year for the event.
4	Month	This field contains the month value for the event. Valid entry for this field is in the range of 1 to 12.
5	Day	This field contains the day value for the event. Valid entries for this field are in the range of 1 to 31.
6	Hour	This field contains the hour value for the event. Valid entries for this field are in the range of 0 to 23.
7	Minute	This field contains the minute value for the event. Valid entries for this field are in the range of 0 to 59.
8	Seconds and Milliseconds	This field contains the seconds and milliseconds value for the event. Valid entries for this field are in the range of 0 to 59999.
9 to 14	Data	These words contain the data for the event. For single- and double-point, step and measured value events, the first word is used. For integrated total events, the first two words are used.

Get Module Time Block (9970)

Block 9970 identification code requests the module's date and time. Use this data to set the ControlLogix processor clock.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the value of 9970 identifying the block type to the module.
1 to 247	Not Used	Not Used

The module responds to a valid block 9970 request with a block containing the requested date and time. The format of this block is shown in the following table.

Block Format for Read

Word Offset in Block	Data Field(s)	Description
0	Reserved	Reserved (0)
1	Block Write ID	This is the next block requested by the module.
2	Year	This field contains the four-digit year for the new time value.
3	Month	This field contains the month value for the new time. Valid entries for this field are in the range of 1 to 12.
4	Day	This field contains the day value for the new time. Valid entries for this field are in the range of 1 to 31.
5	Hour	This field contains the hour value for the new time. Valid entries for this field are in the range of 0 to 23.
6	Minute	This field contains the minute value for the new time. Valid entries for this field are in the range of 0 to 59.
7	Seconds	This field contains the second value for the new time. Valid entries for this field are in the range of 0 to 59.
8	Milliseconds	This field contains the millisecond value for the new time. Valid entries for this field are in the range of 0 to 999.
9 to 248	Not Used	Not Used
249	Block Read ID	This field contains the block identification code of 9970 for the block.

Set Module Time Block (9971)

Block identification code 9971 passes clock time from the ControlLogix processor to the module. The date and time provided sets the module's clock.

Block Format for Write

Word Offset in Block	Data Field(s)	Description
0	Block ID	This field contains the block identification code of 9971 for the block.
1	Year	This field contains the four-digit year for the new time value.
2	Month	This field contains the month value for this new time. Valid values are in the range of 1 to 12.
3	Day	This field contains the day value for the new time. Valid values are in the range of 1 to 31.
4	Hour	This field contains the hour value for the new time. Valid values are in the range of 0 to 23.
5	Minute	This field contains the minute value for the new time. Valid values are in the range of 0 to 59.
6	Seconds	This field contains the second value for the new time. Valid entries for this field are in the range of 0 to 59.
7	Milliseconds	This field contains the millisecond value for the new time. Valid values for this field are in the range of 0 to 999.
8 to 247	Not Used	Not Used

Warm Boot Block (9998)

This block is sent from the ControlLogix processor to the module (output image) when the module is required to perform a warm-boot (software reset) operation. This block is commonly sent to the module any time configuration data modifications are made. This will cause the module to read the new configuration information and to restart. The following table describes the format of the control block.

Offset	Description	Length
0	9998	1
1 to 247	Spare	247

Because many of the parameters set in the configuration file require the program to restart, this function works the same as the cold-boot function. It will cause the module to exit and restart the program.

Cold Boot Block (9999)

This block is sent from the ControlLogix processor to the module (output image) when the module is required to perform the cold boot (hardware reset) operation. This block is sent to the module when a hardware problem is detected by the ladder logic that requires a hardware reset. The following table describes the format of the control block.

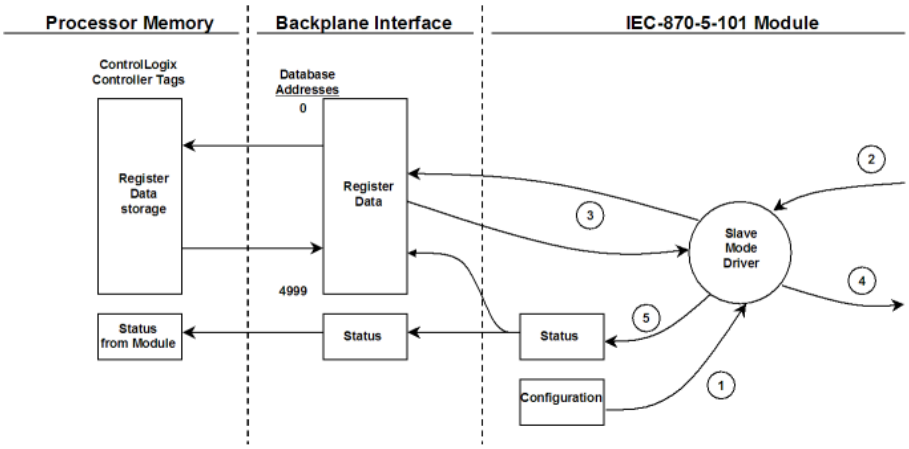
Offset	Description	Length
0	9999	1
1 to 247	Spare	247

5.2.6 Data Flow Between MVI56-101S Module and ControlLogix processor

The following topics describe the flow of data between the three pieces of hardware, the ControlLogix processor, the MVI56-101S module, and the IEC 60870-5-101 Master unit. Each port on the module is configured to emulate a common slave device. The database used in the module is used for both ports and only one port can be utilized at one time.

Slave Driver

The Slave Driver allows the MVI56-101S module to respond to data read and write commands issued by a master unit on the telecontrol network. The following flow chart and associated table describe the flow of data into and out of the module.



Step	Description
1	The slave port driver receives the configuration information from the Compact Flash Disk in the module. This information configures the serial port and define the slave node characteristics.
2	A Host device issues a read or write command to the module's node address. The port driver qualifies the message before accepting it into the module.
3	After the module accepts the message, the data is immediately transferred to or from the internal database in the module. If the command is a read command, the data is read out of the database and a response message is built. If the command is a write command, the data is written directly into the database and a response message is built.
4	After the data processing has been completed in Step 3, the response is issued to the originating master node.
5	Counters are available in the Status Block that permit the ladder logic program to determine the level of activity of the Slave Driver.

Review the **Installing and Configuring the Module** section for a complete list of the parameters that must be defined for a slave port. The IEC 60870-5-101 Interoperability Document for the MVI56-101S Slave Module contains a listing of the protocol support supplied in the module.

5.2.7 Databases

The read and write areas can be placed anywhere in the module's database area. Because each point is defined individually to the module, the data for a specific type need not be contiguous in the module's database. This means that the module error/status area can be passed to the controlling station using the M_ME_NB_1 database. In the database definition for this type, establish a M_ME_NB_1 information object for each status value to be monitored by the controlling station and use the module's status database as the DB Address.

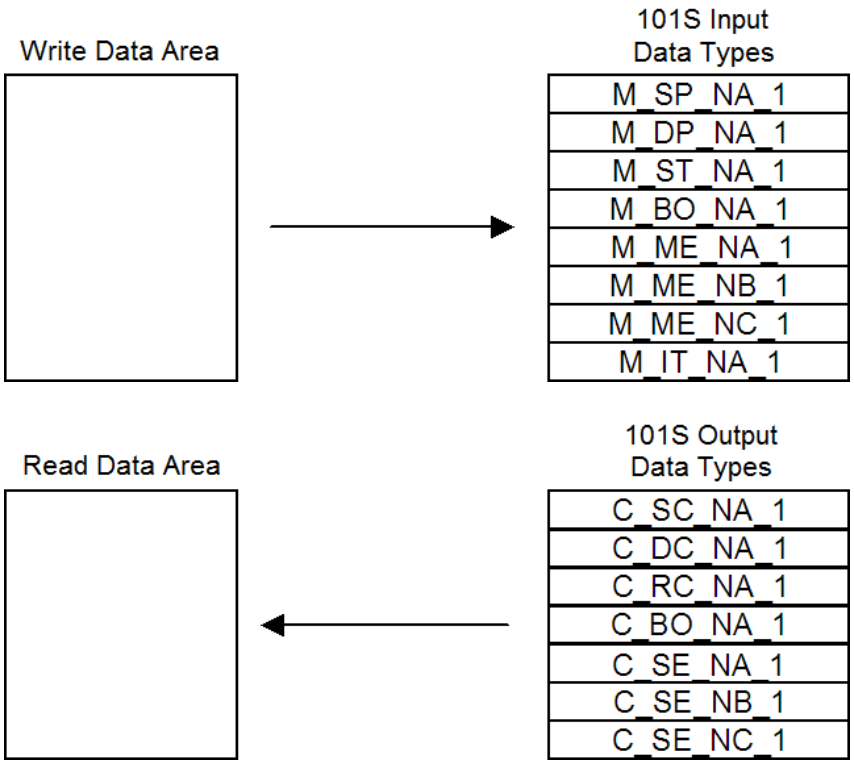
This section discusses the databases utilized by the module to support the IEC 60870-5-101 protocol. The user is responsible for defining the databases for their specific application. The size of each ASDU database and definition of each point within each data type is established in the module's configuration file or by using ProSoft Configuration Builder. Refer to the *Module Configuration* section for a complete discussion of configuration options. The following table summarizes the ASDU data types used in each of the15 individual databases.

Type ID	Type	Description	Data Representation
1	M_SP_NA_1	Monitored Single-point Information: This data type stores a single binary input point. Associated time-tagged event information for this type are M_SP_TA_1 (2) and M_SP_TB_1 (30).	Single bit value with 0=Off and 1=On.
3	M_DP_NA_1	Monitored Double-point Information: This data type stores a dual-point binary input value (that is, valve status). Associated time-tagged event information for this type are M_DP_TA_1 (4) and M_DP_TB_1 (31).	Double-bit status with 00b (0 decimal) = indeterminate or intermediate, 01b (1 decimal) = Off, 10b (2 decimal) = On and 11b (3 decimal) = indeterminate.
5	M_ST_NA_1	Monitored Step-point Information: This data type is used for step position of transformers or other step position information. The value for the position ranges from -64 to 63. Associated time-tagged event information for this type are M_ST_TA_1 (6) and M_ST_TB_1 (32).	Step data is stored in a single character value with bits 0 to 6 (-64 to +63) representing the step position and bit 7 representing the following states: 0 = Equipment is not in transient state 1 = Equipment in transient state
7	M_BO_NA_1	Monitored 32-Bit Bitstring Binary State Information: This data type is used to hold and transmit binary state data as bitstrings of 32 bits.	Bitstring data is stored as groups of 32 bits, with each bit representing the state of an object: 0=Off and 1=On
9	M_ME_TA_1	Monitored Normalized Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TA_1 (10) and M_ME_TD_1 (34).	Normalized values are stored in a (16-bit) word data area with a range of $-1..+1-2^{-15}$
11	M_ME_NB_1	Monitored Scaled Measured Value: This data type is used for analog input data. Associated time-tagged event information for this type are M_ME_TB_1 (12) and M_ME_TE_1 (35).	Scaled values are stored in a (16-bit) word data area with a range of $-2^{15}..{}^{+215}-1$

Type ID	Type	Description	Data Representation
13	M_ME_NC_1	Monitored Measured Value, Short Floating-Point Number: This data type is used for analog input data stored in floating point format according to the IEEE STD 754, QDS format. Associated time-tagged event information for this type are M_ME_TC_1 (14) and M_ME_TE_1 (36).	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign)
15	M_IT_NA_1	Monitored Integrated Total-point Information: This data type stores meter or other count data. Associated time-tagged event information for this type are M_IT_TA_1 (15) and M_IT_TB_1 (37).	Binary counter data is stored in a double-word (32-bit) value with a range of -2^{31}..+2^{31}-1.
45	C_SC_NA_1	Single-point Command: This command controls a single binary point such as a relay.	Single bit value with 0 = Off and 1 = On
46	C_DC_NA_1	Double-point Command: This command controls a dual-point binary control device such as a trip/close relay.	Double Command with 0 = Not permitted 1 = Off 2 = On 3 = Not permitted
47	C_RC_NA_1	Regulating Step Command: This command controls a stepping device such as a transformer.	Regulating Step Command with 0 = Not permitted 1 = Next step lower 2 = Next step higher 3 = Not permitted
48	C_SE_NA_1	Setpoint Command, Normalized Value: This command controls an analog device.	Normalized values are stored in a (16-bit) word data area with a range of -1..+1-2^{-15}
49	C_SE_NB_1	Setpoint Command, Scaled Value: This command controls an analog device.	Scaled values are stored in a (16-bit) word data area with a range of -2^{15}.. $^{+215}$-1
50	C_SE_NC_1	Setpoint Command, Short Floating-Point Value: This data type is used for analog input data stored in floating point format according to the IEEE STD 754, QDS format.	Short floating-point number stored in IEEE STD 754 format (Fraction, Exponent, Sign)
51	C_BO_NA_1	32-Bit Bitstring Binary State Command: This data type is used to hold and transmit binary state data as bitstrings of 32 bits.	Bitstring data is stored as groups of 32 bits, with each bit representing the state of an object: 0=Off and 1=On

A key concept in interfacing the protocol with the internal database is the relationship between the IEC-60870-5-101 databases and the data transfer operation between the driver and module's internal database. The control data types, C_XX_XX_1, are transferred from the driver to the module's internal database. This data can then be used by the other gateway protocol or processor in which the module is installed to control other devices. The monitor data types (M_XX_XX_1) are sourced from the module's internal database and is transferred to the remote Master on the IEC-60870-5-101 network.

The relationships between the data types and the read and write data areas of themodule are displayed in the following diagram:



The Write Data Area is used to hold data for the 101S Monitored Point databases, which is data that will be sent by the 101S Slave driver from the module to the remote Master. The Read Data Area is used to hold data for the 101S Control Point databases, which is data that has been sent from the remote Master to the 101S driver to be stored in the module. The read and write areas can be placed anywhere in the module's database. Because the module address of each point is defined individually, the data area for a specific data type need not be contiguous in the module's memory. However, most users find it convenient to keep data of each type in contiguous address blocks.

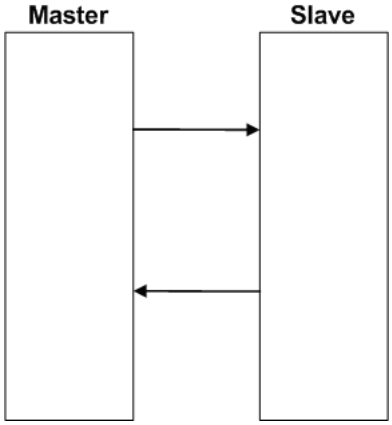
Module error/status data can be passed to the Master using the M_ME_NB_1 database. In the database definition for the type, establish a point for each status value to be monitored by the Master and set the module's database address for the point in the definition.

5.3 IEC-60870-5-101 Protocol Implementation

This section presents an overview of how the MVI56-101S works, while skipping the complex details of the protocol specification. If you require more information about the implementation of the protocol, refer to the protocol specification (IEC 60870-5-101 2003). For more information on the configuring and modifying the backplane implementation of the protocol with the MVI56-101S, refer to Configuring the Module (page 31).

The IEC-60870-5-101 protocol applies to telecontrol equipment and systems with coded bit serial data transmission for monitoring and controlling geographically widespread processes.

Any application using the IEC-60870-5-101 protocol will have a master (controlling station) and one or more slaves (controlled stations). The master will constantly monitor and control the data from each slave in the network.

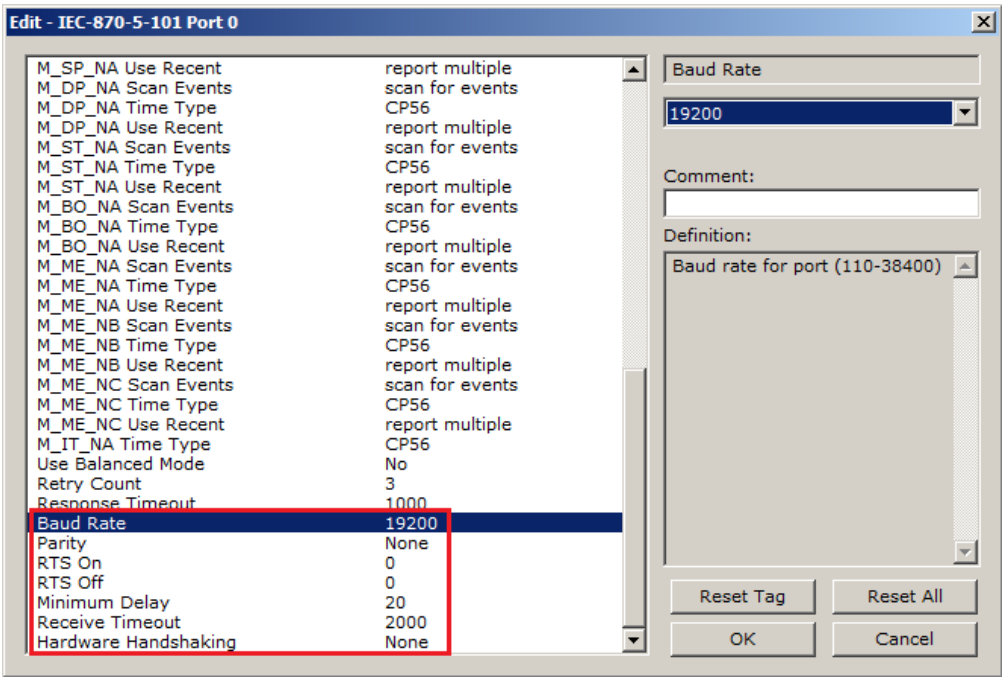


The MVI56-101S module works as a IEC-60870-5-101 slave. It can send monitor data, receive commands or generate events to the master unit, as explained in the following topics.

5.3.1 General Parameter Configuration

Communication Parameters

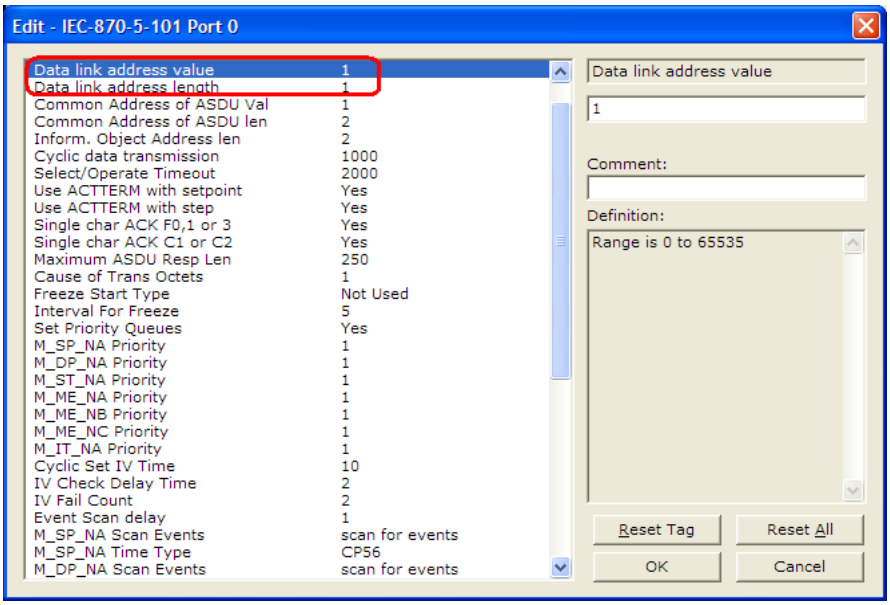
The following parameters should be configured for serial communication:



Adjust these parameters for your application.

Data Link Configuration

The protocol specification document IEC 60870-5-2 specifies an unambiguous address (number) must be used for each link. Each address may be unique within a specific system, or may be unique within a group of links sharing a common channel. The protocol specification defines that the Data Link Address may have 0, 1 or 2 octets. The first option should be used only during Balanced Mode.



Configure the Data Link Address Length parameter to set the number of octets to be used for the Data Link Address value. It is essential that the Master unit also uses the same number of octets configured in the 101 Slave. Values shown here are for example only. Your application may require different settings.

- Data Link Address Length: 1

You must also configure the actual Data Link Address value using the following parameter:

- Data link address: 1

This value identifies the module's address in the network.

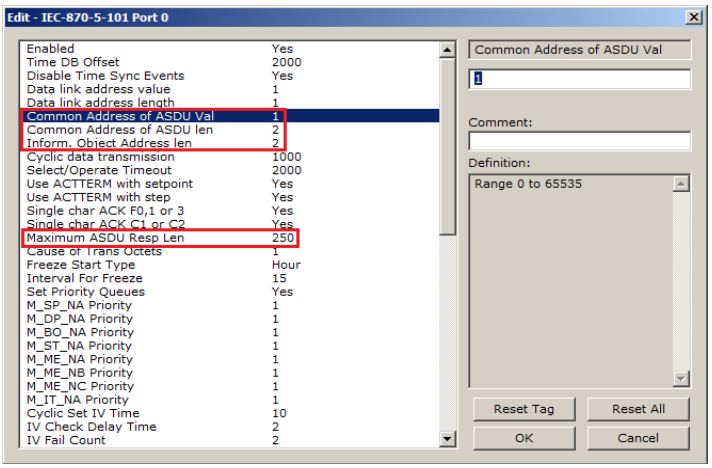
ASDU Configuration

The protocol specification document IEC 60870-5-3 describes the Basic Application Data Units that are used in the protocol. It also defines the Application Service Data Unit (ASDU) used by the protocol for data communication.

You can configure the number of bytes to be used for the following ASDU components:

Component	Abbreviation	Possible Number of Octets
Cause of Transmission	COT	1 or 2
Common Address of ASDU	CASDU	1 or 2
Information Object Address	IOA	1, 2 or 3

These parameters must be configured to match the Master's configuration (values shown are for example only; your application may require different values):



The Common Address of ASDU is the station address. The module allows the addressing of only the whole station (some devices allows different Common Addresses to identify particular station sectors). You should configure the Common ASDU Address with the following parameter:

- Common Address of ASDU: 1

You should also configure the maximum number of bytes that the module will support for each ASDU response to the Master unit. The range is from 25 to 252 bytes. You should verify the maximum number of bytes supported by the Master because some IEC 60870-5-101 Master devices support messages with less than 252 bytes.

Configure the following parameter to set the maximum number of bytes to be transferred at every ASDU response.

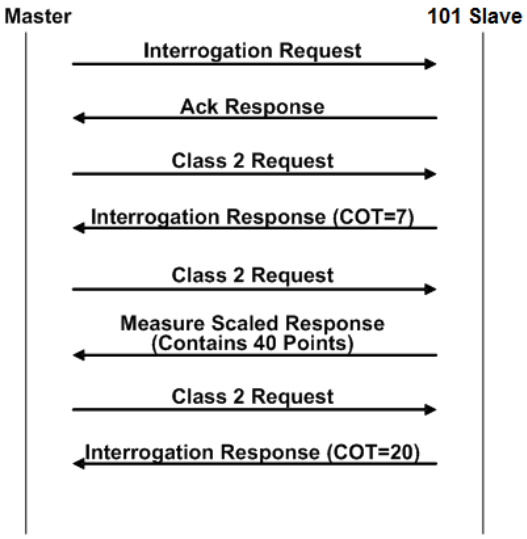
- Maximum ASDU Resp Len: 252

A value less than 252 can cause the module to break down the response to send all points using more response messages, with each message containing fewer bytes.

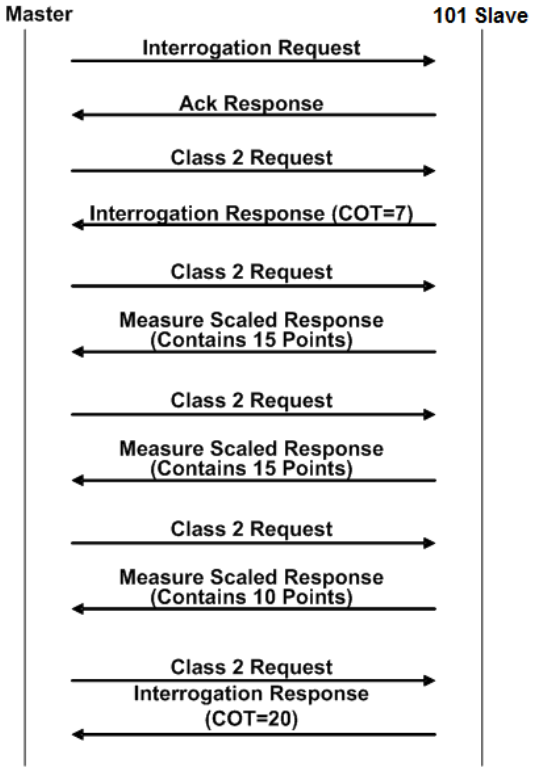
Example - Changing the ASDU Length:

Considering that the Master sends a General Interrogation request to poll forty M_ME_NB points (measured scaled points in Unbalanced Mode):

If Maximum ASDU Resp Len = 252:



This example shows that the module sends all 40 measured scaled points in one single message, if the message is not greater than 252 bytes. However, some Master devices may not support messages containing this number of bytes. If the Master for a given application supports only 100 bytes, the following communication procedure would occur:



As shown in the previous diagram, the module sends out three (3) messages (15 points, 15 points, and 10 points) instead of sending the whole 40 points in one single message.

Note: This example shows the case where IOA Length = 3 bytes, COT Length = 2 bytes and CASDU Length = 2 bytes.
Important: If the database parser gets a point index that is not valid, the whole database is set as invalid and no points are reported. Because the index 0 is not valid (the protocol does not support this index value), the driver considers it invalid. For example, if you set the size of the ASDU to 1 and you set a value of 1000 for a point index, this is also invalid as the indexes can only go from 1 to 255.

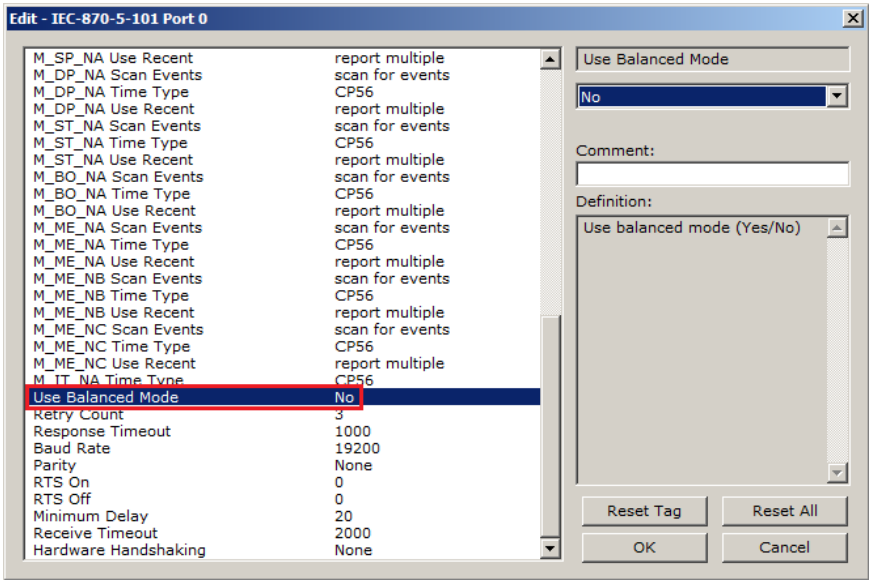
5.3.2 Module Initialization

After the module powers up, a specific initialization procedure occurs, depending on the communication mode you selected (Balanced or Unbalanced).

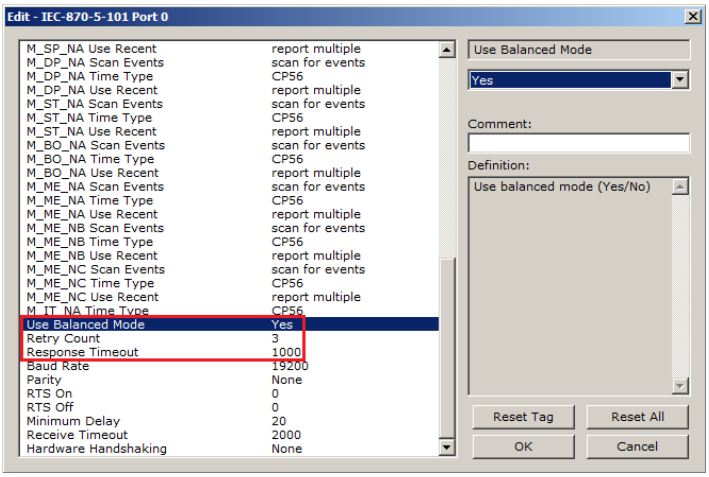
Balanced and Unbalanced Modes

The module supports balanced and unbalanced modes. In balanced mode, each station may initiate message transfers. If the links from the Master unit to several slaves share a common physical channel, then these links must be operated in an unbalanced mode to avoid the possibility of more than one device attempting to transmit on the channel at the same time.

Select the communication mode with the following parameter:



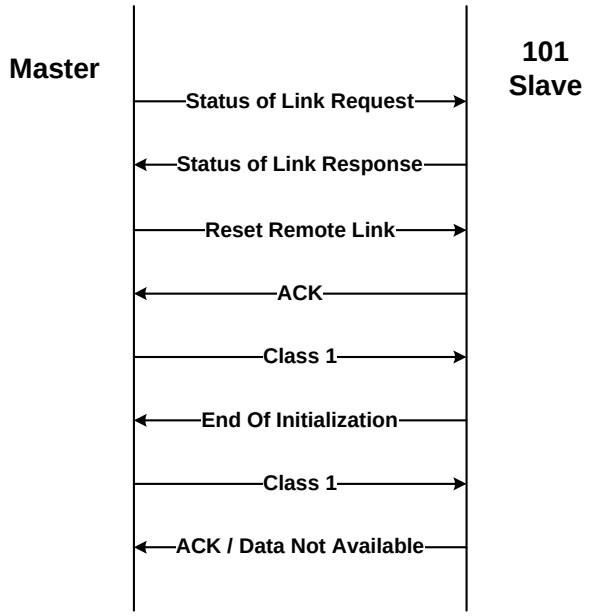
To use balanced mode, configure the following parameters:



Unbalanced mode

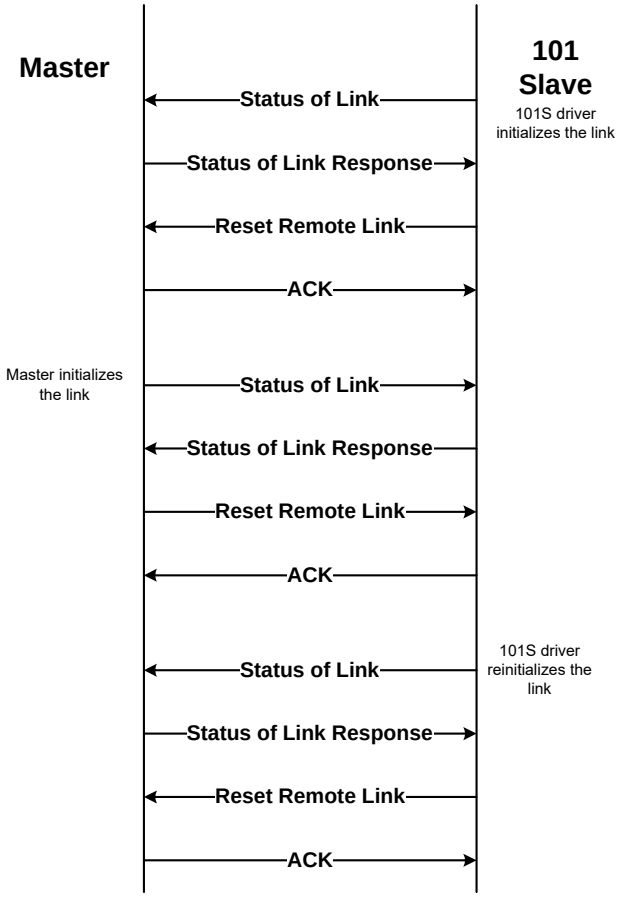
In order to start communications between the Master and the slave units, the Master tries to establish the link connection by transmitting repeated "Request Status of Link" at specific time out intervals. When the module's link is available, it will respond with a "Status of Link" response. Then, the Master transmits a "Reset of Remote Link" message and the slave responds with an Acknowledge (ACK) response. Then the Master sends two consecutive Class 1 requests. The slave responds the first Class 1 request with an "End of Initialization" response and the second Class 1 request with an ACK message.

The following illustration shows a typical initialization procedure for the unbalanced mode:



Balanced Mode

During balanced mode, the link must be initialized in both directions. The MVI56-101S module also always reinitializes the link after it receives an initialization request from the Master. Therefore, the following initialization occurs during balanced mode, after MVI56-101S boots.



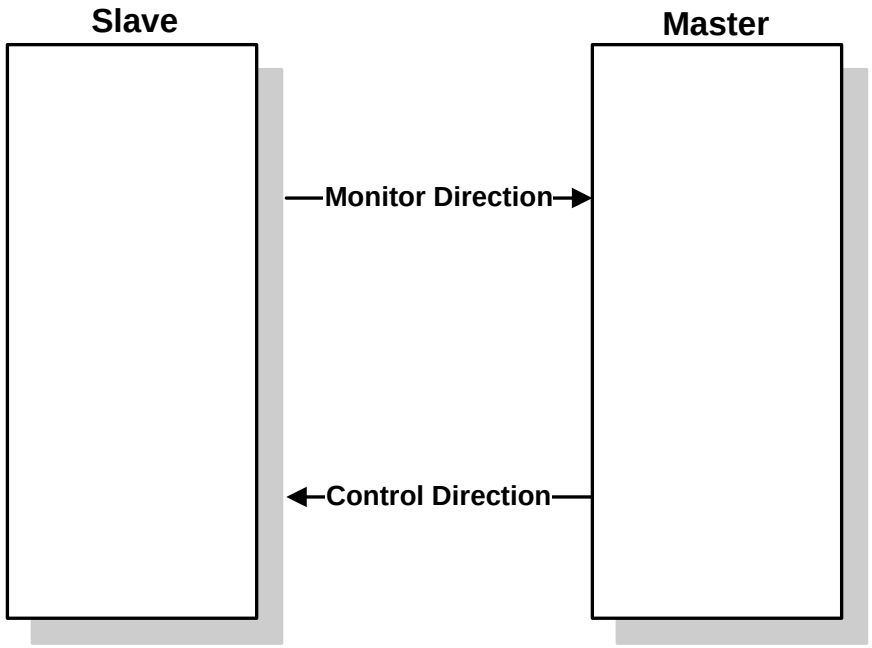
After the initialization procedure is completed, the Master and the MVI56-101S start communicating. During communication, if the Master fails to respond to a message from the module, the module will retry for a configured number of times. If the Master still fails to respond, the module will attempt to initialize the line again.

5.3.3 Monitor Direction and Control Direction

The protocol specification defines two directions of data: monitor direction and control direction. These directions are defined by the protocol specification as follows:

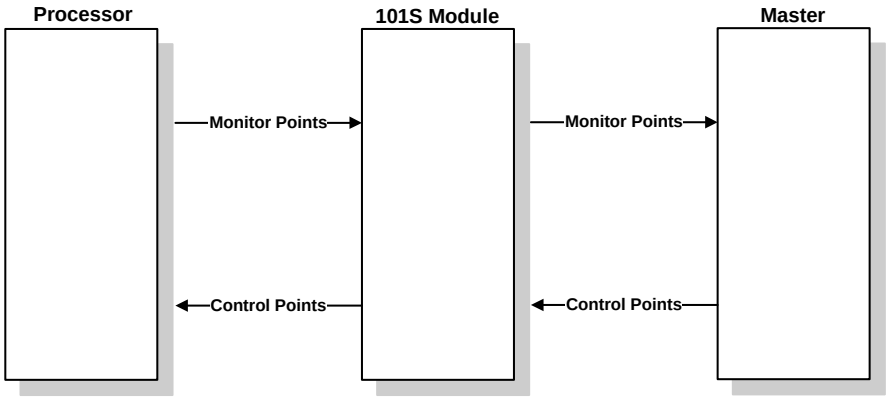
Monitor Direction: The direction of transmission from a slave to the master

Control Direction: The direction of transmission from the master to a slave



The points that are typically transferred from the slave to the master are also known as Monitor Points (or Monitor Information Objects). The points that are typically transferred from the master to the slave are also known as Control Points (or Command Information Objects).

For the MVI56-101S, the control and monitor points would be transferred as follows:



You must make sure that all points are configured in the correct location in the MVI56-101S module database in order to be properly updated from/to the processor.

5.3.4 Using Monitor Points

The following monitor points are supported by the MVI56-101S module:

Symbol	Description	Data Size in Database	Addressing Type
M_SP_NA	Monitored Single-Points	1 bit	Bit
M_DP_NA	Monitored Dual-Points	2 bits	Bit
M_ST_NA	Monitored Step-Points	1 byte	Byte
M_ME_NA	Monitored Measured Normalized-Points	1 word	Word
M_BO_NA	Monitored 32-bit Bitstring Points	2 words	Word
M_ME_NB	Monitored Measured Scaled-Points	1 word	Word
M_ME_NC	Monitored Measured Short Floating Points	2 words	Double-Word
M_IT_NA	Monitored Counter-Points	2 words	Double-Word

Each monitor point is identified by its Information Object Address (it should be unique for each Common ASDU Address in the network). For each monitor point, configure the following parameters:

Point # - The information object address of the point. It identifies the point in the network.

DB Address - The database location in the MVI56-101S module associated with the point. You must associate each point to a database address in the MVI56-101S module. The interpretation of this parameter depends on the point type configured. For example, for an M_SP_NA point, this value represents the bit address. For a M_ME_NA point, this value represents the Word address.

Group(s) - This is the group definition for the point. It sets how the point will be polled by the Master (cyclic or group interrogation). It can also be used to enable or disable the event generation for one specific point. The group parameter is discussed in the Data Communication section.

Deadband - Sets the deadband for each Measured point. If the value changes from more than the configured deadband , the module will generate an event for this point.

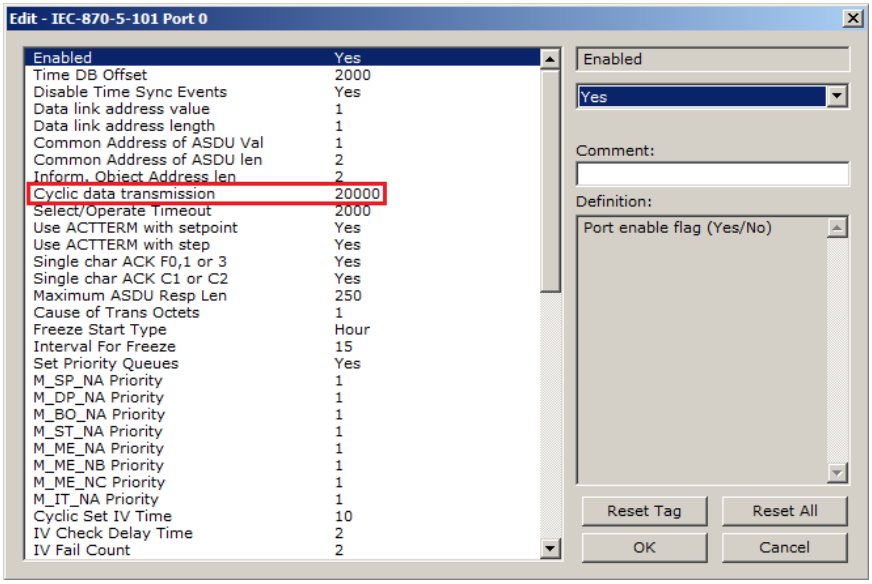
IV DB Bit - This feature allows the application to set the invalid (IV) quality bit of the protocol for all the monitored ASDU types supported. If you enable this feature, the Master can determine the individual IV quality bit status of each point you configured.

Monitor Data Transfer

In unbalanced mode, the polling procedure is initiated from the Master unit through Class 1 and Class 2 requests. In general, ASDUs containing periodic(cyclic) causes of transmission are typically assigned to be transmitted in response to Class 2 requests from the Master. All ASDUs containing time-tagged or spontaneous causes of transmission are typically assigned to be transmitted in response to Class 1 requests. Other ASDUs with other causes of transmission of low priority, such as background scans, may also be assigned to be transmitted in response to Class 2 requests.

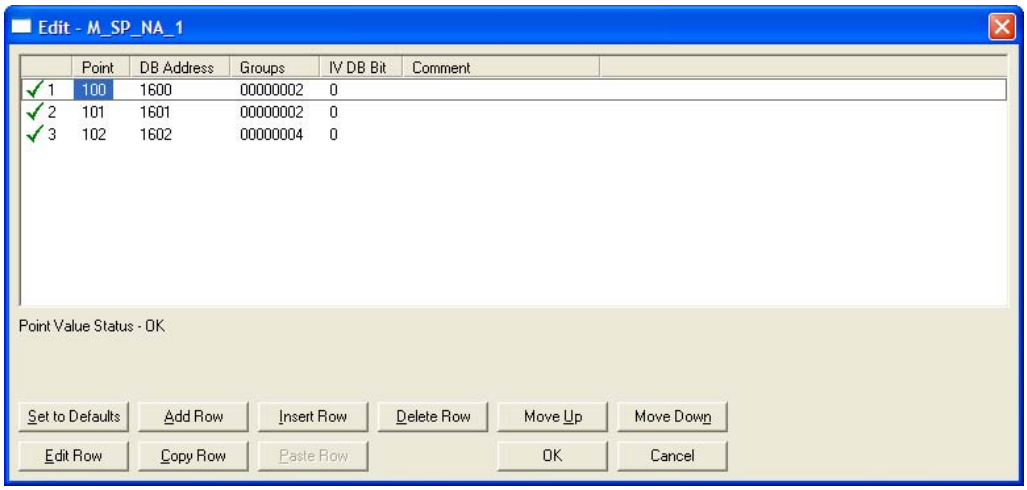
Typically, you should properly configure the group code for each monitor point to define how the Master will poll for the point.

The module will periodically send all points configured for periodic/cyclic poll (0x80000000) at every x milliseconds, where x is configured with the *Cyclic Data Transmission* parameter:



M_SP_NA

A *Monitor Single-Point* information object occupies one binary bit and uses bit addressing. For example, if you configured the following information objects as shown:



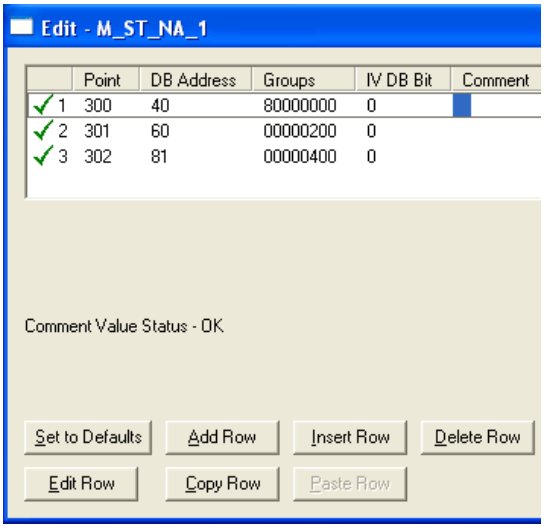
The following table describes how these information objects would be stored in the module:

Inf. Object Address	Module Database Address
100	Bit 0 of word 100 (Bit address 1600)
101	Bit 1 of word 100 (Bit address 1601)
102	Bit 2 of word 100 (Bit address 1602)

M_ST_NA

A *Monitor Step Position* information object occupies one byte and uses byte addressing.

For example, if you configured the following information objects:



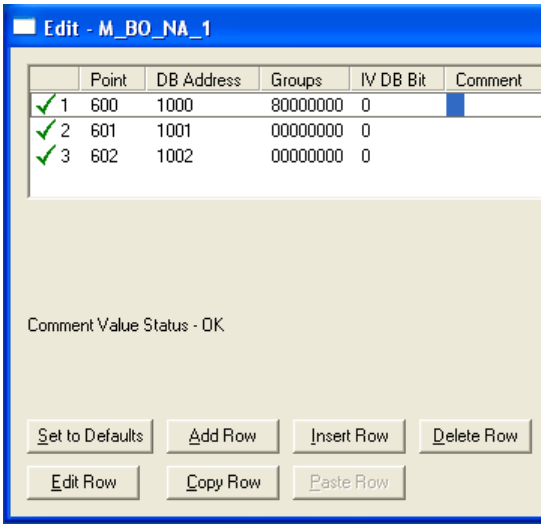
The following table describes how these information objects would be stored in the module.

Inf. Object Address	Module Database Address
300	Low byte of word 20 (Byte address 40)
301	Low byte of word 30 (Byte address 60)
302	High byte of word 40 (Byte address 81)

M_BO_NA

A *Monitored 32-Bit Bitstring Point* occupies two words and uses double-word addressing.

For example, if you configured the following points:



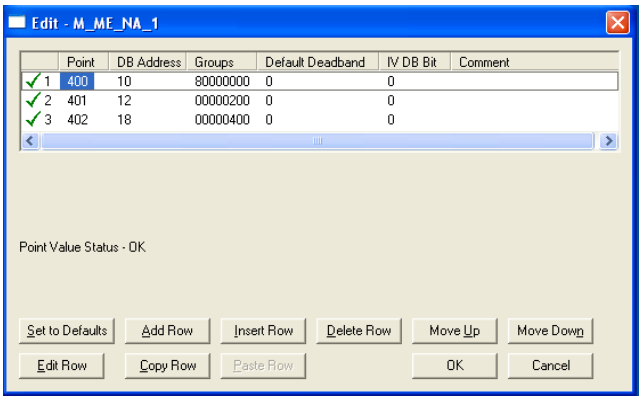
The following table describes how these points would be stored in the module memory database.

Inf. Object Address	Module Database Address
600	Words 2000 and 2001 (Double-word address 1000)
601	Words 2002 and 2003 (Double-word address 1001)
602	Words 2004 and 2005 (Double-word address 1002)

M_ME_NA and M_ME_NB

A *Monitor Normalized Measured* information object or *Monitor Scaled Measured* information object occupies one word and uses word addressing.

For example, if you configured the following information objects:



The following table describes how these information objects would be stored in the module.

Inf. Object Address	Module Database Address
400	Word 10 (Word address 10)
401	Word 12 (Word address 12)
402	Word 18 (Word address 18)

Monitor Normalized Measured information objects use a data representation defined by the protocol specification to represent fractional decimal values. The following table describes the value for each bit as a reciprocal power of two (2), that is two (2) raised to the power of a negative exponent (-1 through -15). Bit 15 is the Sign Bit.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value Hex(h) Decimal	Sign	4000h 2^{-1}	2000h 2^{-2}	1000h 2^{-3}	800h 2^{-4}	400h 2^{-5}	200h 2^{-6}	100h 2^{-7}	80h 2^{-8}	40h 2^{-9}	20h 2^{-10}	10h 2^{-11}	8h 2^{-12}	4h 2^{-13}	2h 2^{-14}	1h 2^{-15}
4000h 0.5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2000h 0.25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1000h 0.125	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6000h 0.75	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3210h 0.395751953125	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Examples:

A value of 4000hex (only Bit 14 set, all others clear) is interpreted as 0.5 decimal

A value of 2000hex (only Bit 13 set, all others clear) is interpreted as 0.25 decimal

A value of 1000hex (only Bit 12 set, all others clear) is interpreted as 0.125 decimal

... and so on until...

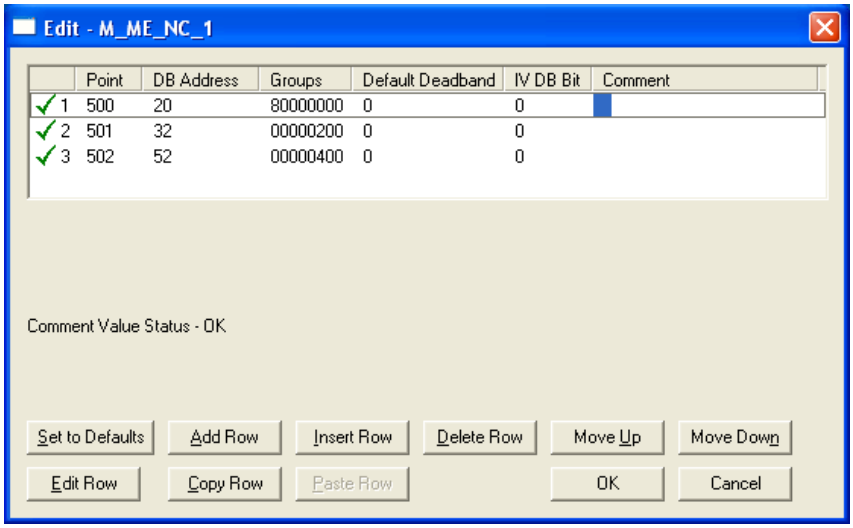
A value of 0001hex (Only Bit 0 set, all others clear) is interpreted as 0.000030517578125

Therefore, the actual data values transmitted may be any combination of the decimal values for any given bit pattern.

M_ME_NC and M_IT_NA

The *Monitor Short Floating-Point Measured Value* and *Monitor Integrated Totals* information objects occupy two words with double-word addressing.

For example, if you configured the following information objects:



The following table describes how these information objects would be stored in the module.

Inf. Object Address	Module Database Address
500	Words 40 and 41 (Double-word address 20)
501	Words 64 and 65 (Double-word address 32)
502	Word 104 and 105 (Double-word address 52)

5.3.5 Using Control (Command) Information Objects

The MVI56-101S module supports the following Control information objects:

ASDU Type	Information Object Description
C_SC_NA	Control Single Command
C_DC_NA	Control Double Command
C_RC_NA	Control Regulating Step Command
C_BO_NA	Control 32-Bit Bitstring Command
C_SE_NA	Control Normalized Value Set Point Command
C_SE_NB	Control Scaled Value Set Point Command
C_SE_NC	Control Short Floating-Point Value Set Point Command

Each Control information object is identified by its Information Object Address. For each Control information object, configure the following parameters:

Point # - This is the Information Object Address of the information object. It identifies the information object in the network. This address must be unique for each Common ASDU Address in the network.

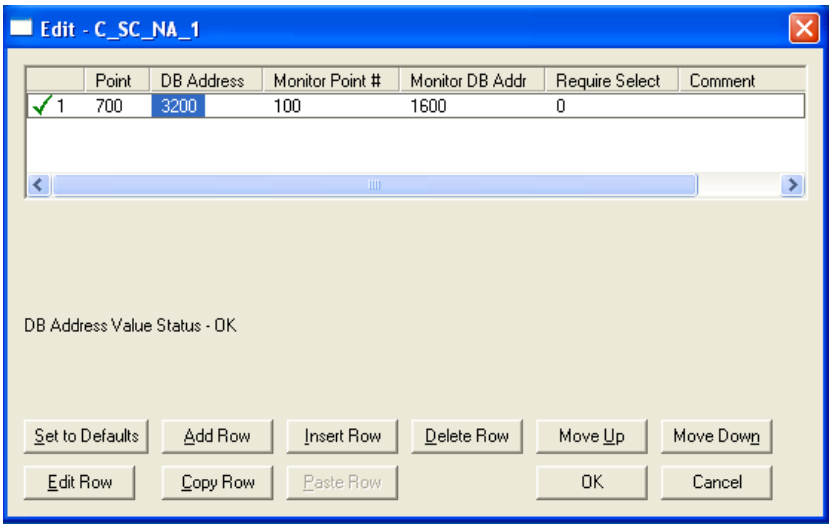
DB Address - This is the database location in the MVI56-101S module associated with the information object.

Monitor Point # - This is the Information Object Address of the associated Monitor information object to be sent to the Client once the Control request is received.

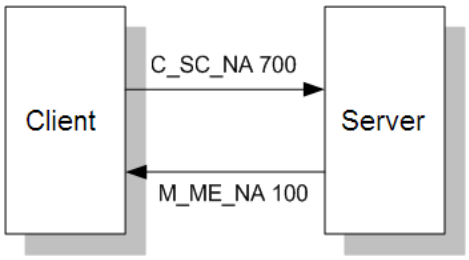
Monitor DB Address - The user might (optionally) configure a Monitor information object to be sent by the MVI56-101S module when it receives the command for that specific information object.

Require Select - This parameter configures the information object to require a Select request before the Operate command.

Example (C_SC_NA)



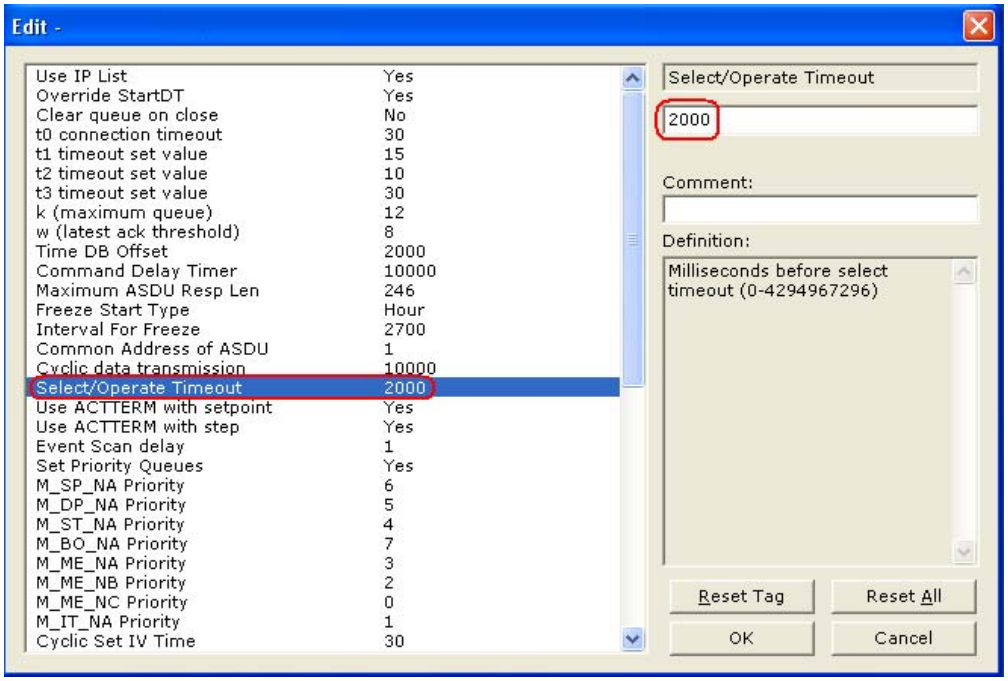
In the example above, each time the module receives a value of one (1) in *Control Single Command* 700, it will send a response by placing a value of one (1) in *Monitor Single-Point* # 100 at database bit address 1600.



Some of the Command information objects may be configured to require a Select command before they will act on an Execute command.

Select/Operate Timeout Parameter

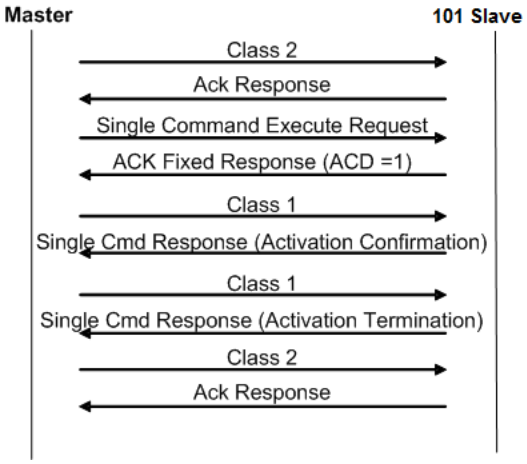
Use the *Select/Operate Timeout* parameter to configure the select/operate timeout period. After the module receives the Select command, it will wait for this period of time to receive the Execute command for the same Information Object Address. If the module does not receive an Execute command within this period of time, it will require another Select operation before the Execute operation will be accepted and acted upon.



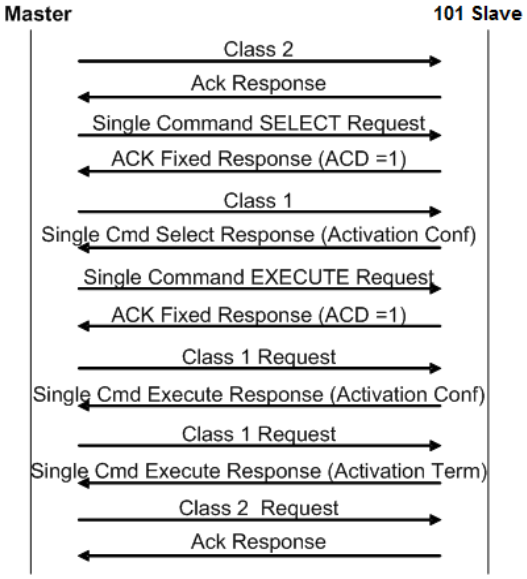
Control Data Transfer

The control communication typically occurs when the Master sends a command request to update the module's command points.

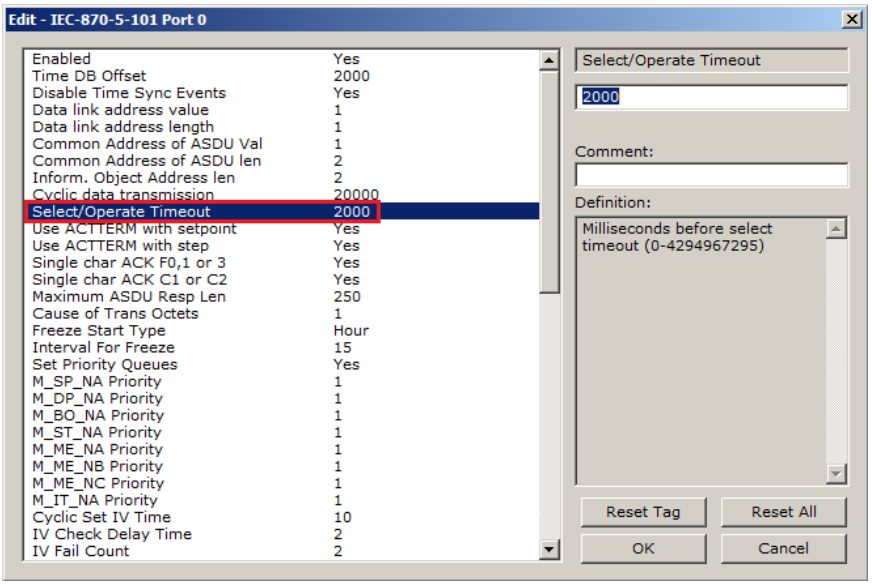
The following illustration shows a typical command communication between the Master and module during unbalanced operation.



Some of the command points may be configured to be selected before executed. The following illustration shows a typical command communication between the Master and module operation with a SELECT operation.



Refer to the following parameter to configure the select/operate timeout period. After the module receives the SELECT operation it will wait for this period of time for the EXECUTE operation. If the module does not receive an EXECUTE operation within this period of time it will require another SELECT operation before the EXECUTE operation.



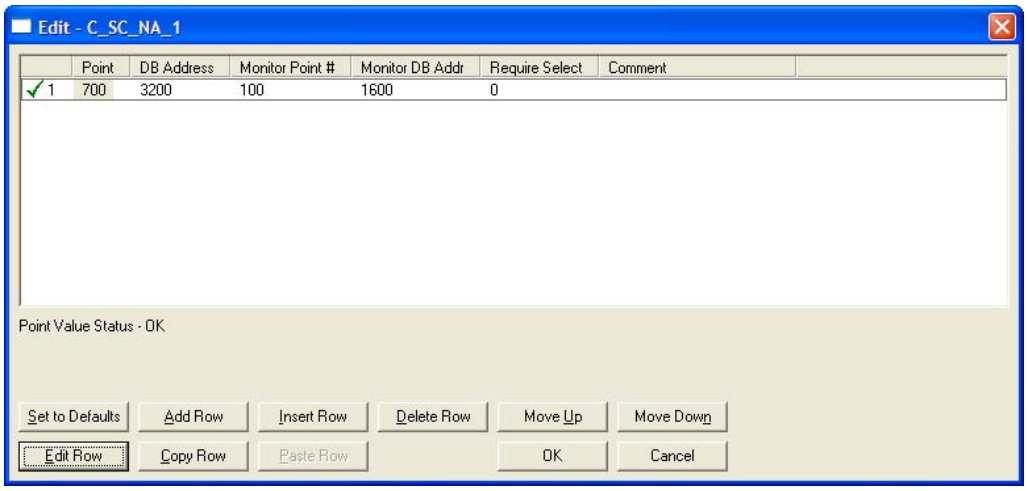
Control Information Objects Addressing

Control information objects must be configured in a database area that is updated in the module. You must associate each information object to a database address in the MVI56-101S module. The interpretation of the *DB Address* parameter in the configuration tables depends on the ASDU configured and the type of addressing associated with that ASDU.

ASDU Type	Data Size	Addressing Type
C_SC_NA	1 bit	Bit
C_DC_NA	2 bits	Bit
C_RC_NA	1 byte	Byte
C_BO_NA	2 words	Double word
C_SE_NA	1 word	Word
C_SE_NB	1 word	Word
C_SE_NC	2 words	Double word

C_SC_NA and C_DC_NA

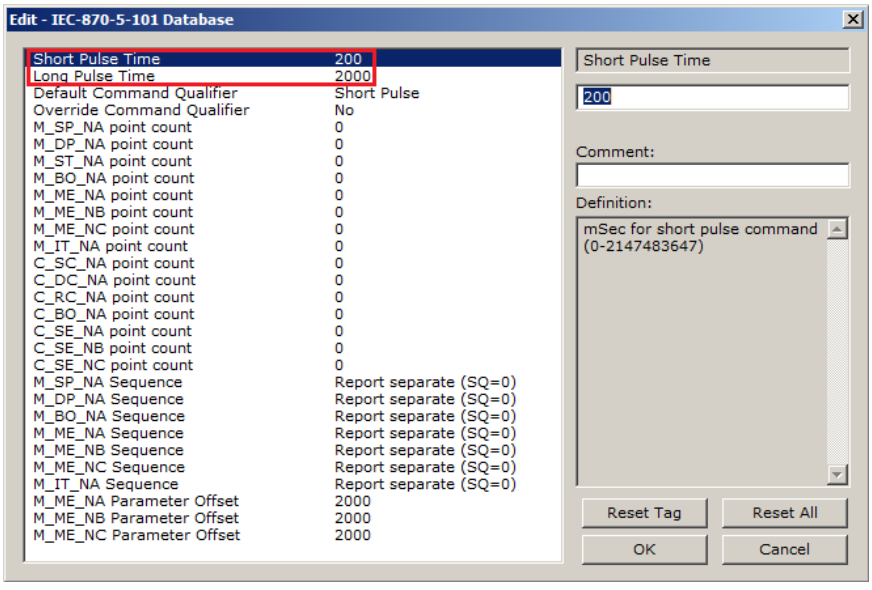
The single-point command and dual-point command points use one bit with bit-addressing. For example, if you configure the following points:



The following table describes how these points would be used.

Inf. Object Address	Module Database Address
100	Bit 0 of word 100
101	Bit 1 of word 100
102	Bit 2 of word 100

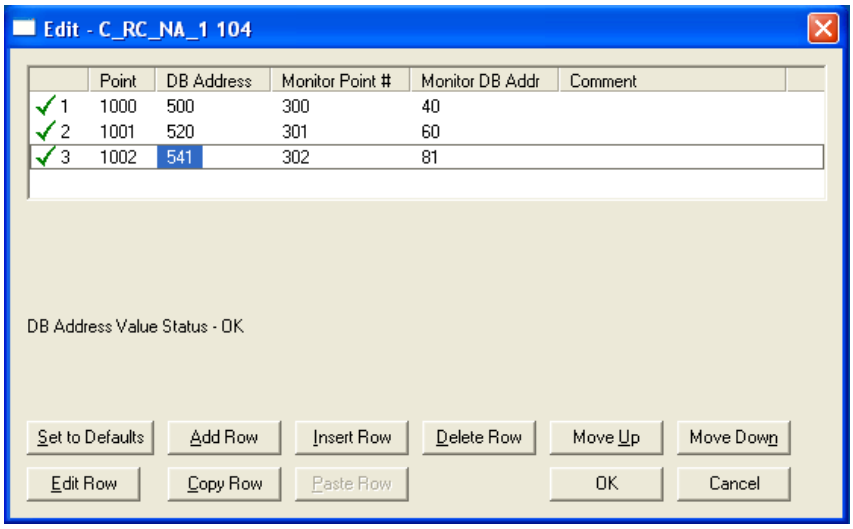
The protocol specification defines a qualifier value that is set by the Master to determine the duration of the pulse (short, long or persistent). Configure the *Short Pulse Time* and *Long Pulse Time* parameters, seen below, to set the duration of the short and long pulses:



C_RC_NA

A *Control Regulating Step Command* information object occupies one byte and uses byte addressing.

For example, if you configured the following information objects:



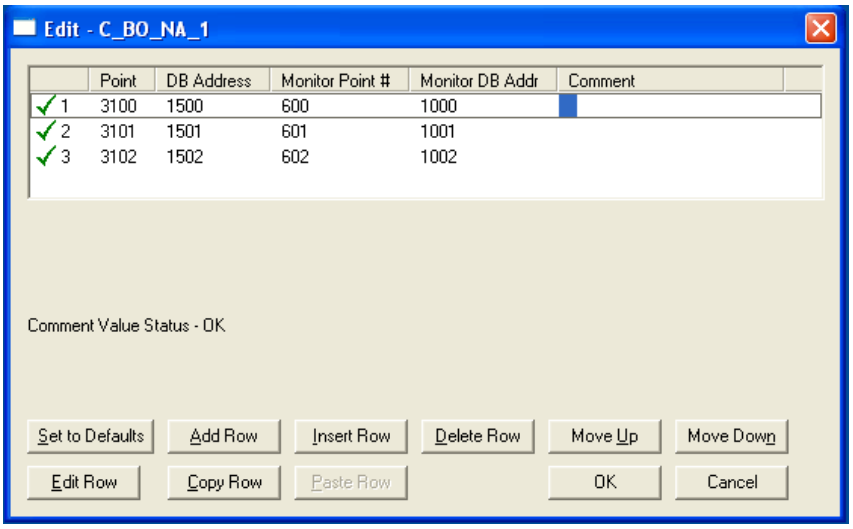
The following table describes how these information objects would be used.

Inf. Object Address	Module Database Address
1000	Low Byte of word 250 (Byte address 500)
300	Low Byte of word 20 (Byte address 40. Must match configuration of point 300 in *M_ST_NA*.)
1001	Low Byte of word 260 (Byte address 520)
301	Low Byte of word 30 (Byte address 60. Must match configuration of point 301 in *M_ST_NA*.)
1002	High Byte of word 270 (Byte address 541)
302	High Byte of word 40 (Byte address 81. Must match configuration of point 302 in *M_ST_NA*.)

C_BO_NA

A *Control 32-Bit Bitstring Command* information object occupies two words and uses double-word addressing.

For example, if you configured the following information objects:

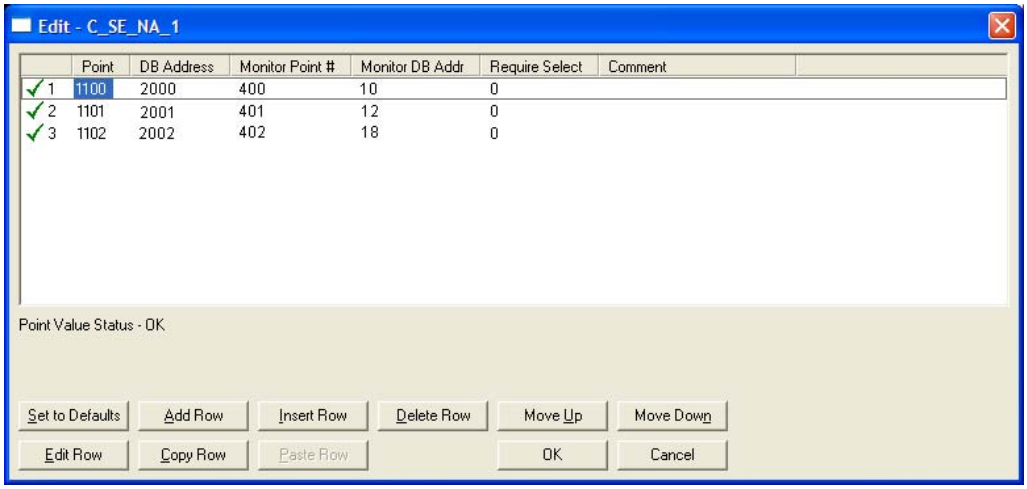


These information objects would be used as follows:

Inf. Object Address	Module Database Address
3100	Words 3000 and 3001 (Double-word address 1500)
600	Words 2000 and 2001(Double-word address 1000. Must match the configuration of point 600 in *M_BO_NA*.)
3101	Words 3002 and 3003 (Double-word address 1501)
601	Words 2002 and 2001 (Double-word address 1001. Must match the configuration of point 601 in *M_BO_NA*.)
3102	Words 3003 and 3004 (Double-word address 1502)
602	Words 2004 and 2005 (Double-word address 1002. Must match the configuration of point 602 in *M_BO_NA*.)

C_SE_NA and C_SE_NB

The *Control Normalized Value Set Point Command* information object and the *Control Scaled Value Set Point Command* information object use one word with word addressing. For example, if you configured the following information objects:



The following table describes how these information objects would be used.

Inf. Object Address	Module Database Address
1100	Word 2000 (Word address 2000)
400	Word 10 (Word address 10. Must match the configuration of point 400 in *M_ME_NA*.)
1101	Word 2001 (Word address 2001)
401	Word 12 (Word address 12. Must match the configuration of point 401 in *M_ME_NA*.)
1102	Word 2002 (Word address 2002)
402	Word 18 (Word address 18. Must match the configuration of point 402 in *M_ME_NA*.)

The *Control Normalized Measured* information objects use a data representation defined by the protocol specification to represent fractional decimal values. The following table describes the value for each bit as a reciprocal power of two (2), that is two (2) raised to the power of a negative exponent (-1 through -15). Bit 15 is the Sign Bit.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value Hex(h) Decimal	Sign	4000h 2^{-1}	2000h 2^{-2}	1000h 2^{-3}	800h 2^{-4}	400h 2^{-5}	200h 2^{-6}	100h 2^{-7}	80h 2^{-8}	40h 2^{-9}	20h 2^{-10}	10h 2^{-11}	8h 2^{-12}	4h 2^{-13}	2h 2^{-14}	1h 2^{-15}
4000h 0.5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2000h 0.25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1000h 0.125	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
6000h 0.75	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3210h 0.395751953125	0	0	1	1	0	0	1	0	0	0	0	1	0	0	0	0

Examples:

A value of 4000hex (only Bit 14 set, all others clear) is interpreted as 0.5 decimal

A value of 2000hex (only Bit 13 set, all others clear) is interpreted as 0.25 decimal

A value of 1000hex (only Bit 12 set, all others clear) is interpreted as 0.125 decimal

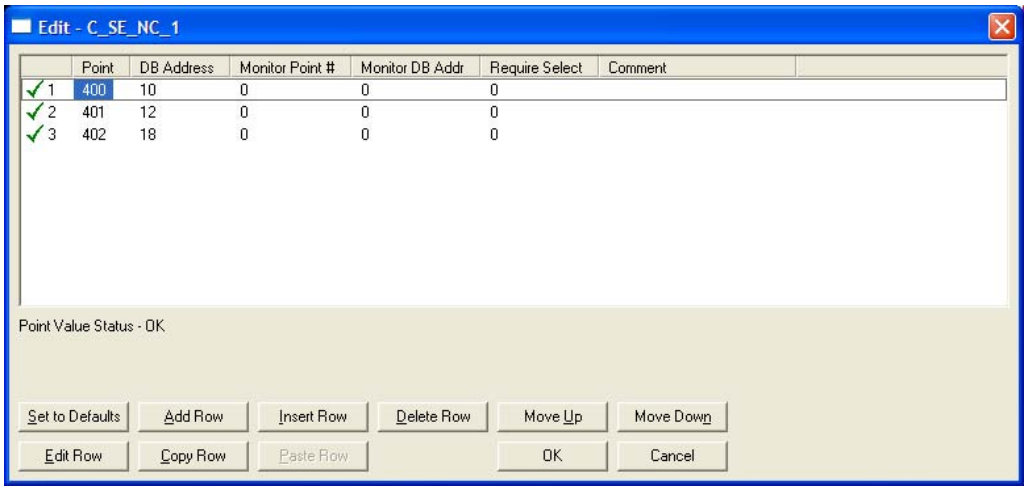
... and so on until...

A value of 0001hex (Only Bit 0 set, all others clear) is interpreted as 0.000030517578125

Therefore, the actual data values transmitted may be any combination of the decimal values for any given bit pattern.

C_SE_NC

The measured short floating point command uses two words with double-word addressing. For example, if you configured the following points:

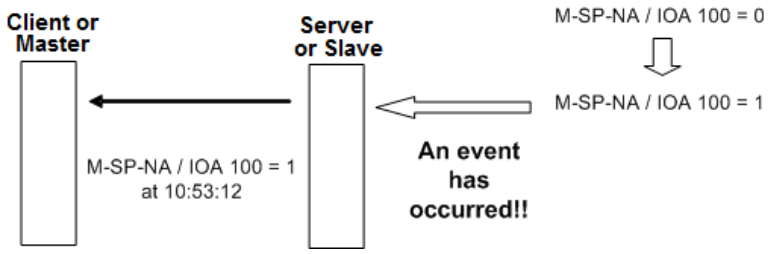


The following table describes how these points would be used.

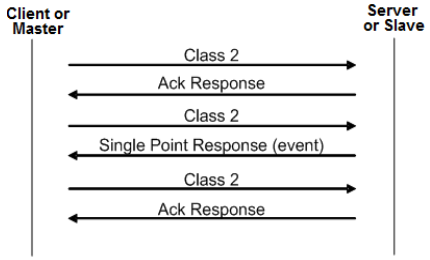
Inf. Object Address	MVI56-101S module Database Address
400	Words 5 and 6
401	Words 6 and 7
402	Words 9 and 10

5.3.6 Events

In order to improve the communication efficiency, most applications will require the Master to periodically poll for data changes with a higher priority than polling for monitor data. Every time data change occurs, the slave should send this information, typically with the date and time information on when the data change occurred.



The following illustration shows the event communication between the Master and the module during unbalanced mode:



The module can queue up to 99 events per data type. When the queue is full, the oldest event will be deleted each time a new event is added to the queue. You must configure the Master to poll the event queue frequently enough to avoid losing events.

The events can be returned for Class 1 or Class 2 requests according to the data type as follows:

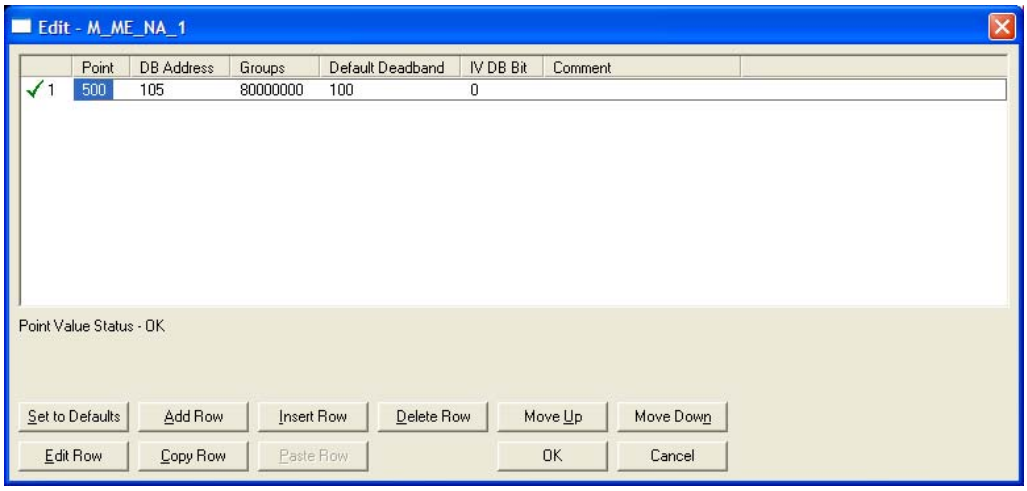
Data Type	Class Request
M_SP_NA	Class 1
M_DP_NA	Class 1
M_ST_NA	Class 1
M_BO_NA	Class 1
M_ME_NA	Class 2
M_ME_NB	Class 2
M_ME_NC	Class 2
M_IT_NA	Class 1

Note: In response to a Class 2 poll the module may respond with Class 1 data when there is no Class 2 data available. So eventually the events for M_SP_NA, M_DP_NA and M_ST_NA points may also be sent during a Class 2 response.

Deadbands

The Monitored Measured points (M_ME_NA and M_ME_NB) will generate events only if the data value changes an amount greater than or equal to the configured deadband value.

For example, if the following point is configured:



Then, if the current value for this point is 130, it would generate events only if:

- NEW VALUE is less than or equal to 30

  OR

- NEW VALUE is greater than or equal to 230.

Each monitored measured point can have it's own deadband value assigned to it in the configuration file or by using the PCB configuration options.

A network Master may also dynamically change the deadband for each monitored point. The Master may send one of the following commands:

Type	Command
110	Parameter of Measured Normalized Data (M_ME_NA)
111	Parameter of Measured Scaled Data (M_ME_NB)
112	Parameter of Measured Short Floating Point (M_ME_NC)

The protocol specification explains that the qualifier value for these commands should be configured as:

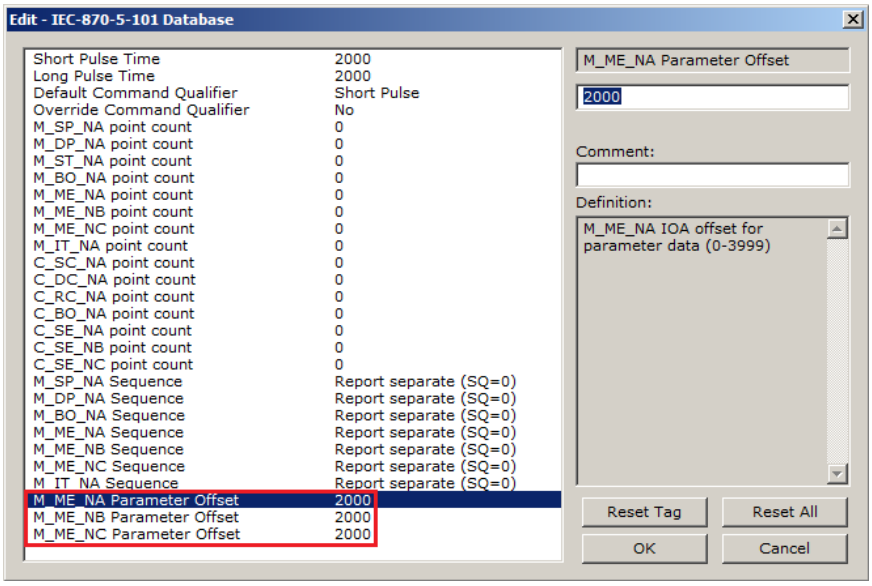
Bits	Value	Description
1 to 6	0	Not Used
	1	Threshold Value (Deadband)
	2	Smoothing Factor (filter time constant) - Not Supported
	3	Low Limit Transmission of Measured Value
	4	High Limit Transmission of Measured Value
	5..31	Reserved
7	0	No Change
	1	Change
8	0	Operation
	1	Not in Operation

The module calculates the Low Limit and High Limit values using the formula below (so these values cannot be changed dynamically)

Low Limit = (LAST REPORTED VALUE) - Deadband

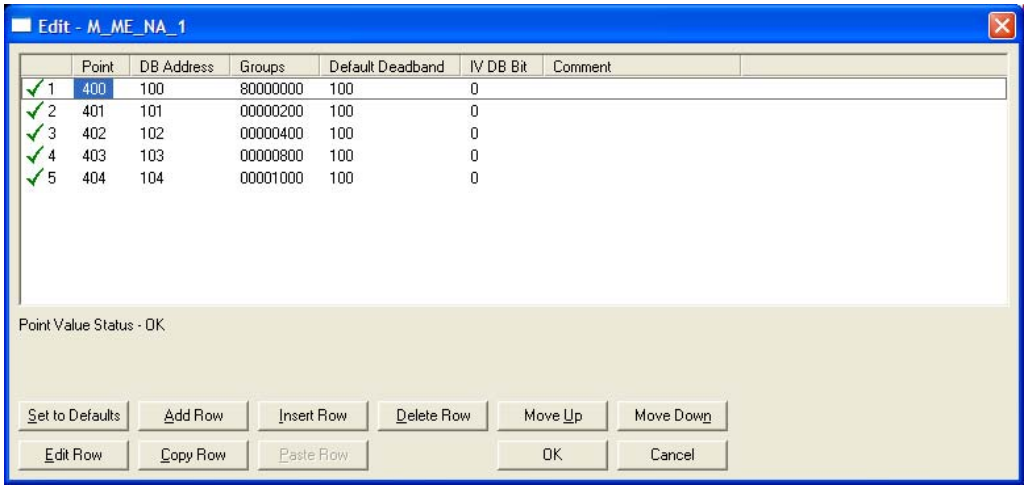
High Limit = (LAST REPORTED VALUE) + Deadband

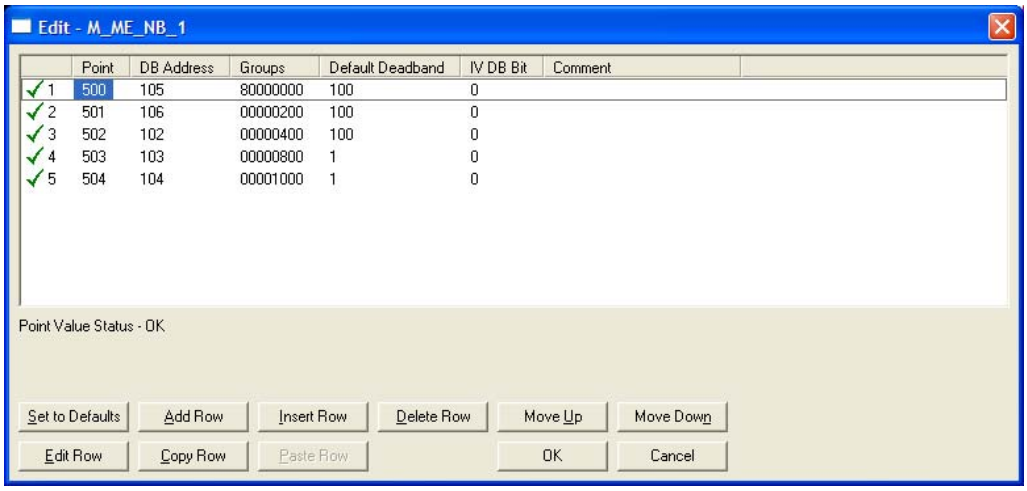
These commands must be sent to a specific Information Object Address. The module associates each monitor measured point with a parameter point through the following configuration parameters:



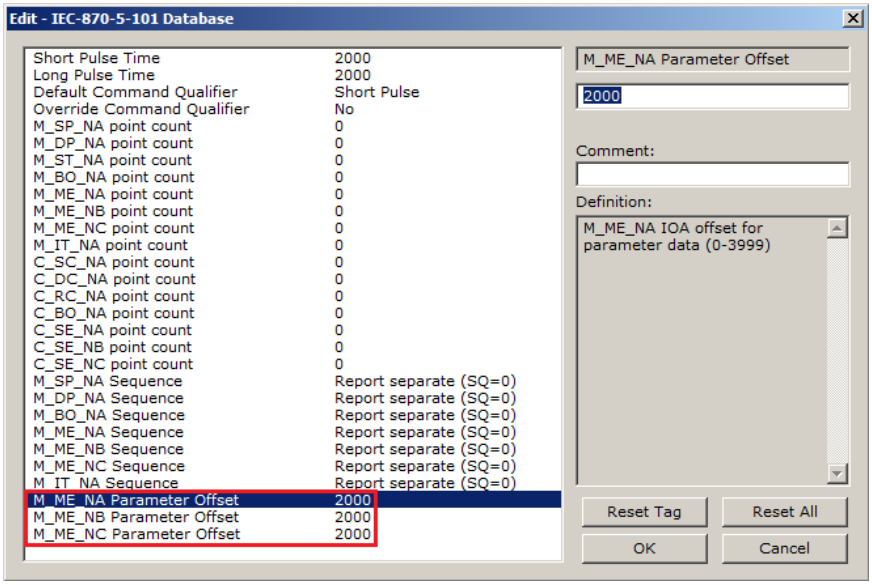
Example - Parameter Data for Monitored Points

If the following monitored measured points are configured:





And if the parameter offsets are configured as follows:

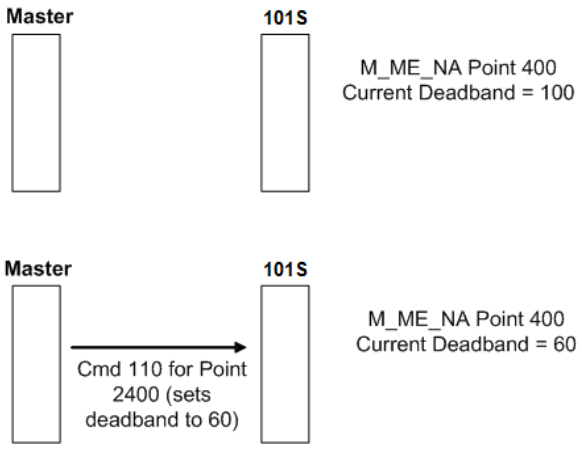


It would imply that the parameter points would be configured as follows:

M_ME_NA

Monitored Measured Normalized Point	Associated Parameter Point
400	2400
401	2401
402	2402
403	2403
404	2404

So, in order to send change the deadband for the M_ME_NA point 400, the Master would send a command type 110 to point 2400:



M_ME_NB

Monitored Measured Normalized Point	Associated Parameter Point
500	2500
501	2501
502	2502
503	2503
504	2504

Controlling the Generation of Events

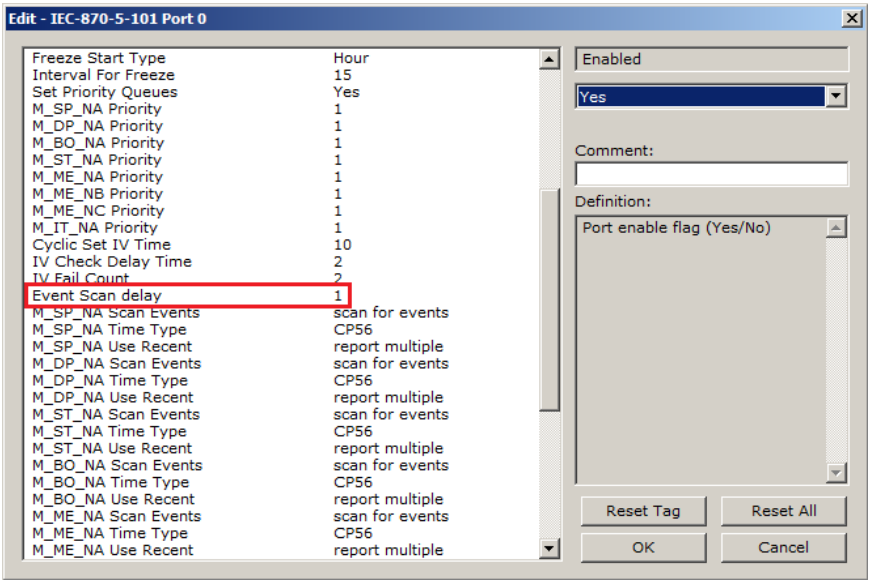
Some applications may require that only some points should generate events. Other applications may require that all configured points should generate events.

The module offers considerable flexibility for event control. You can control how events will be generated at 3 different levels:

1. Application Level (all configured points)
2. Data Type Level (all configured points of specific ASDU type)
3. Point Level (each individual point)

General Events (All Points)

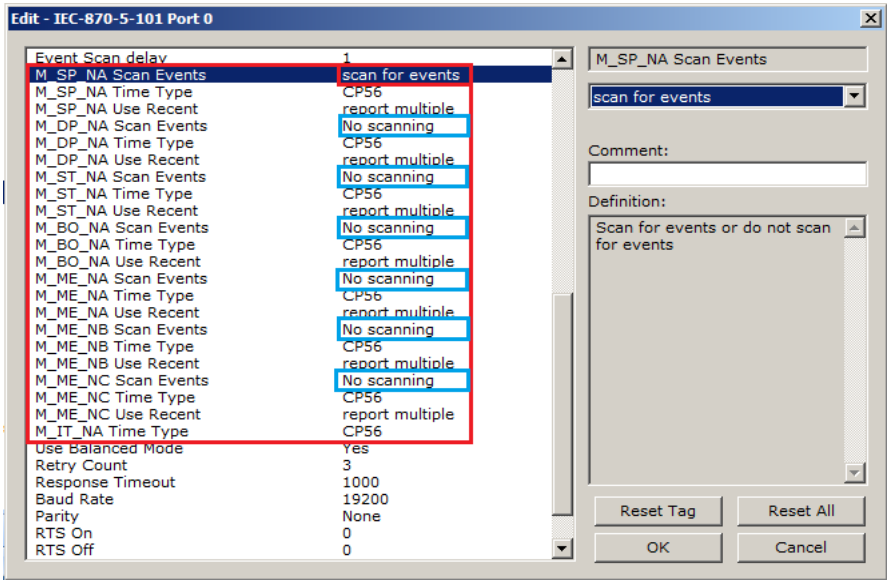
You can control how frequently the module will scan the database for events using the following configuration parameter:



If this parameter is set to 0, the module will not generate events for any points. A non-zero value will configure how frequently the module will scan for events in the database.

Data Type Level

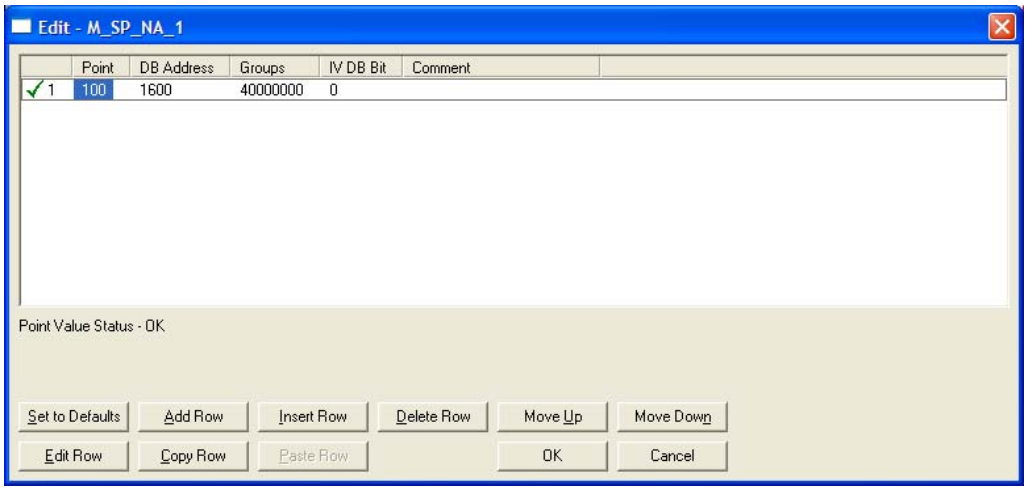
You can configure whether a data type should generate events. Each data type has a configuration parameter to control the generation of events:



In the example above, only the M_SP_NA points would generate events.

Point Level

You can configure whether each point should generate events using the Group field for each point configuration. To disable event generation for a specific port, set the value to 40000000.



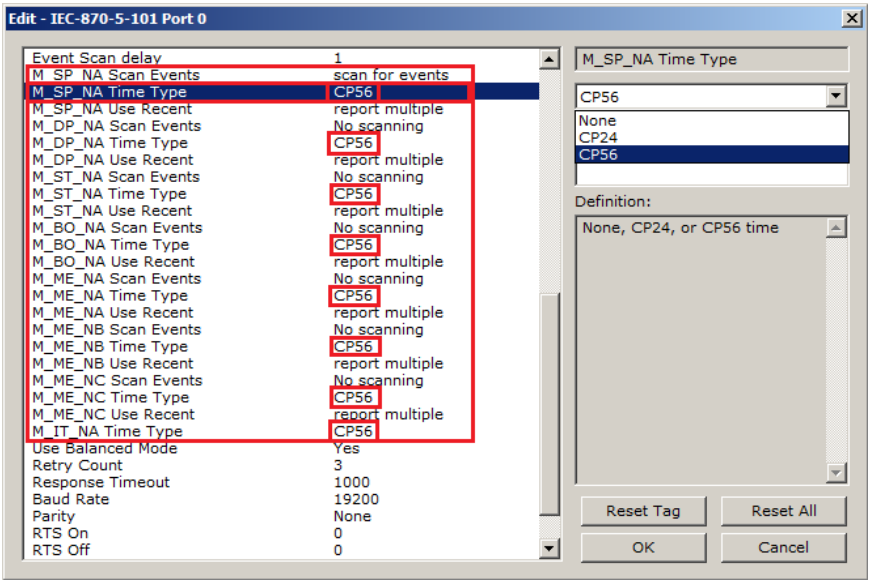
Time Information

Each event may also carry the date and time information when it occurred. The module supports the CP56 and CP24 time formats (as defined in the protocol specification).

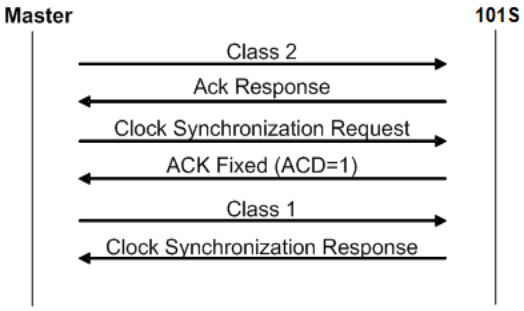
The CP56 time format contains the milliseconds, seconds, minute, hour, day, month and year when the event has occurred.

The CP24 time format contains the milliseconds and minutes when the event has occurred.

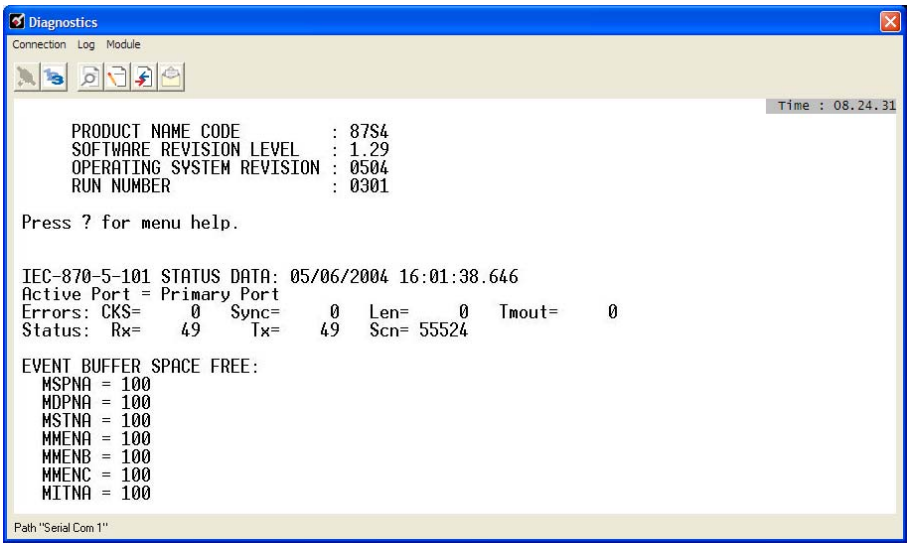
The module may also be configured to not send any time information with each event for certain data types. The following *Time Type* parameters may be used to control the time information for each data type:



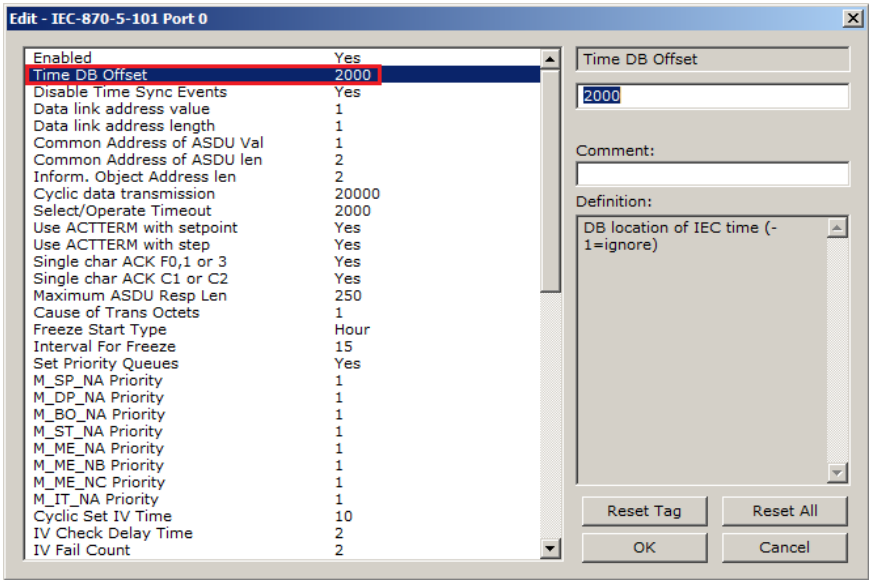
The Master should periodically send a Time Synchronization command to the module driver to synchronize its date and time information. This is a very important step in order to make sure that the Master and the slave are both using the same time information.



You can check the current date and time using the debug menu. Press **[E]** at the main menu to view the module Status Data screen:



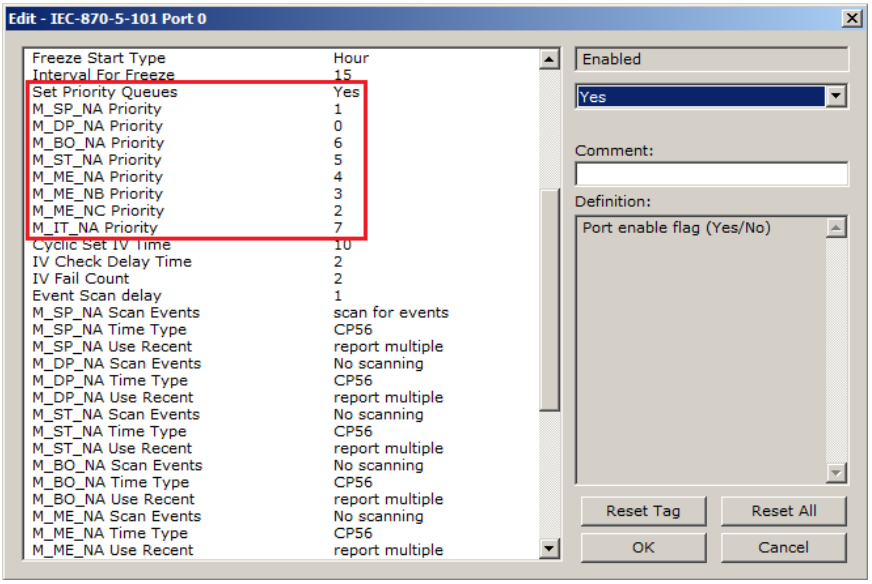
You can also configure the module to copy the date and time information to the module database with the *Time DB Offset* parameter:



As the module sends events to the Master (serial modules) or client (ethernet modules), it also sends a Clock Synchronization response (spontaneous COT) every hour for the transmission of the clock time to the Master (serial modules) or client (ethernet modules).

Event Priority

Event Priority permits the user to set reporting priorities for data change events generated for each ASDU data type. The configuration file contains the following parameters to support this feature:



The *Set Priority Queues* parameter must be set to **YES** for this feature to be used. Each of the Monitored Point ASDUs must be assigned a unique priority index from 0 to 7. If Event Priority is enabled, whenever events of an ASDU with a lower priority number are present, they will always be reported before events from any other ASDUs with higher priority numbers. Multiple events will be assembled into each message packet for transmission. This methodology limits the likelihood of a buffer overflow and still maximizes the bandwidth usage on the communication channel.

Example - Event Priority

If the module is configured with the example values above, and the event queue contains the events generated in the following order:

Event Order	ASDU
1	M_SP_NA
2	M_SP_NA
3	M_DP_NA
4	M_ST_NA
5	M_DP_NA
6	M_SP_NA

Then, the module will respond to a *Class One* data request from the controlling station by returning the data in the event queues in the order shown in the following table.

Packet Order	Content
1	M_DP_NA events 3 and 5
2	M_SP_NA events 1, 2 and 6
3	M_ST_NA event 4

WARNING: Events from the higher priority (lower-numbered) queues are always reported first, when present, before lower priority (higher-numbered) queues. If careful planning is not exercised when deciding how to configure Event Priority queues, or if large volumes of events occur events in the lower queues may be lost due to buffer overflow.

When using Event Priority, events from the highest priority queues, when present, will always be reported before events from lower priority queue. Therefore, if large volumes of events are generated, events in the lower priority queues may be lost due to event buffer overflows.

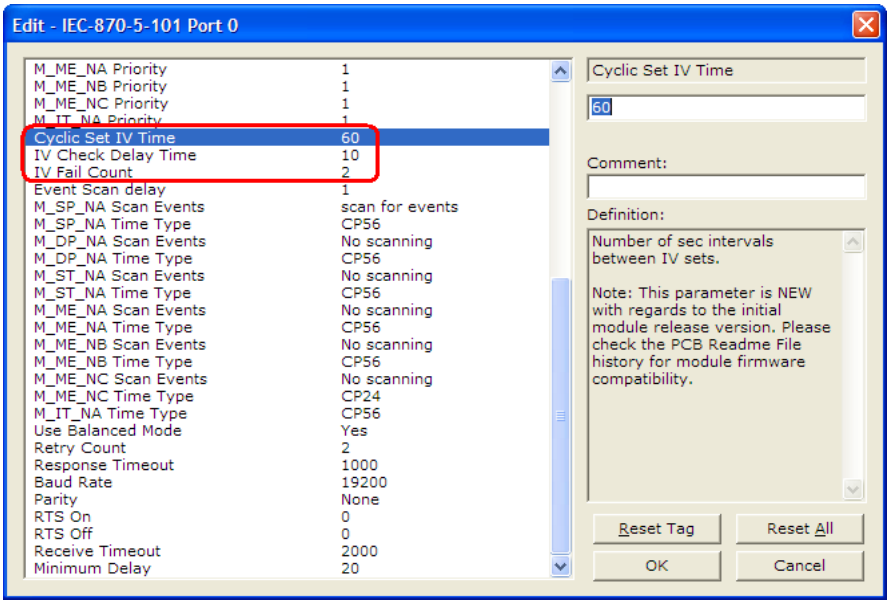
If the *Set Priority Queue* parameter is set to **NO**, this feature is not utilized. Events will be reported based on their order of occurrence, as determined by the event's timestamp. Multiple events will be assembled into message packets for transmission. This methodology limits the likelihood of a buffer overflow and still maximizes the bandwidth usage on the communication channel.

5.3.7 Invalid Bit Monitoring

This feature allows the application to set the invalid (IV) quality bit of data points for all supported Monitored ASDU types. If the feature is enabled, the status of each point configured by the user can have the individual IV quality bit determined by the processor.

The parameters required to support this feature are:

- Cyclic Set IV Time
- IV Check Delay Time
- IV Fail Count

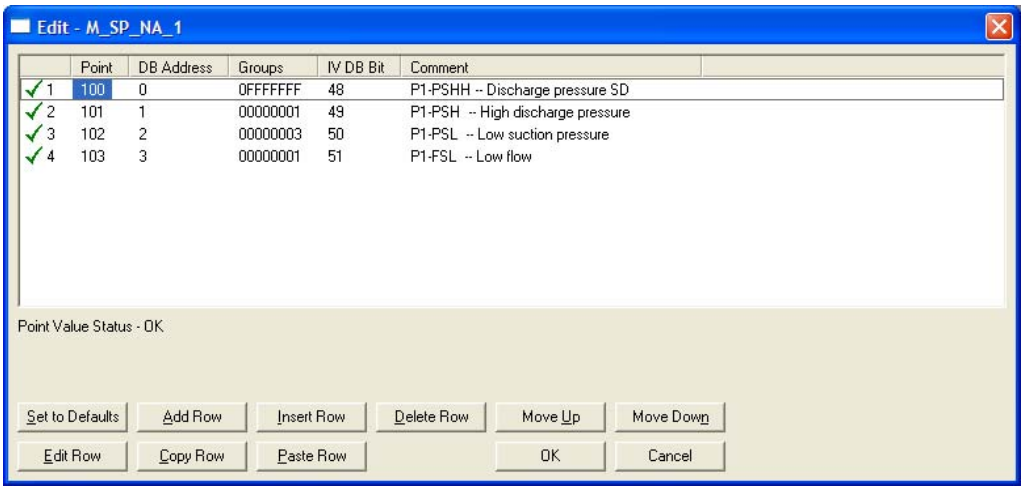


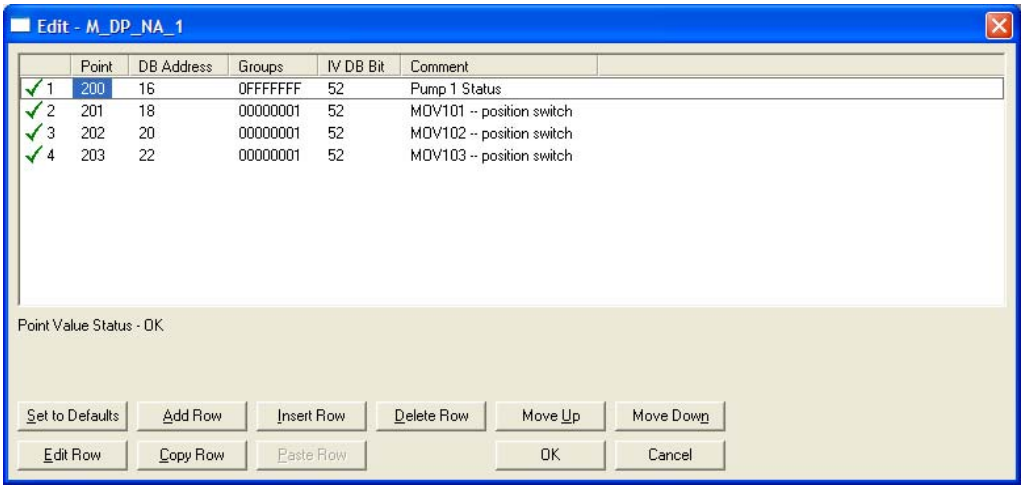
To disable this feature, set the IV Fail Count parameter to 0. If the IV bit field is absent or set to 0, the invalid quality state for the point will always be reported as valid. If used, the Cyclic Set IV Time parameter must be at least 3 times larger than the IV Check Delay Time.

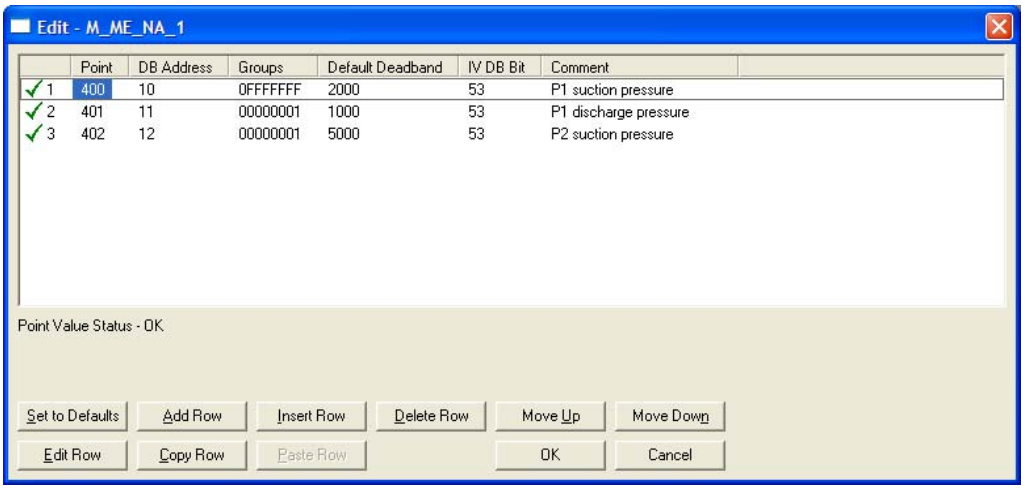
The **CYCLIC SET IV TIME** parameter determines the interval at which the driver will set all the IV bits for the points being monitored in the module. If the IV bit is ON for the number of times specified by the **IV FAIL COUNT** parameter , the point is in an invalid state. The driver will check the state of each bit at the frequency determined by the **IV CHECK DELAY TIME**.

The values for the parameters must permit the driver to properly execute the logic. For example, the value for **CYCLIC SET IV TIME** should be twice the **IV CHECK DELAY TIME,** multiplied by the **IV FAIL COUNT**. If the cyclic timer is set to a smaller value, the logic may not execute correctly.

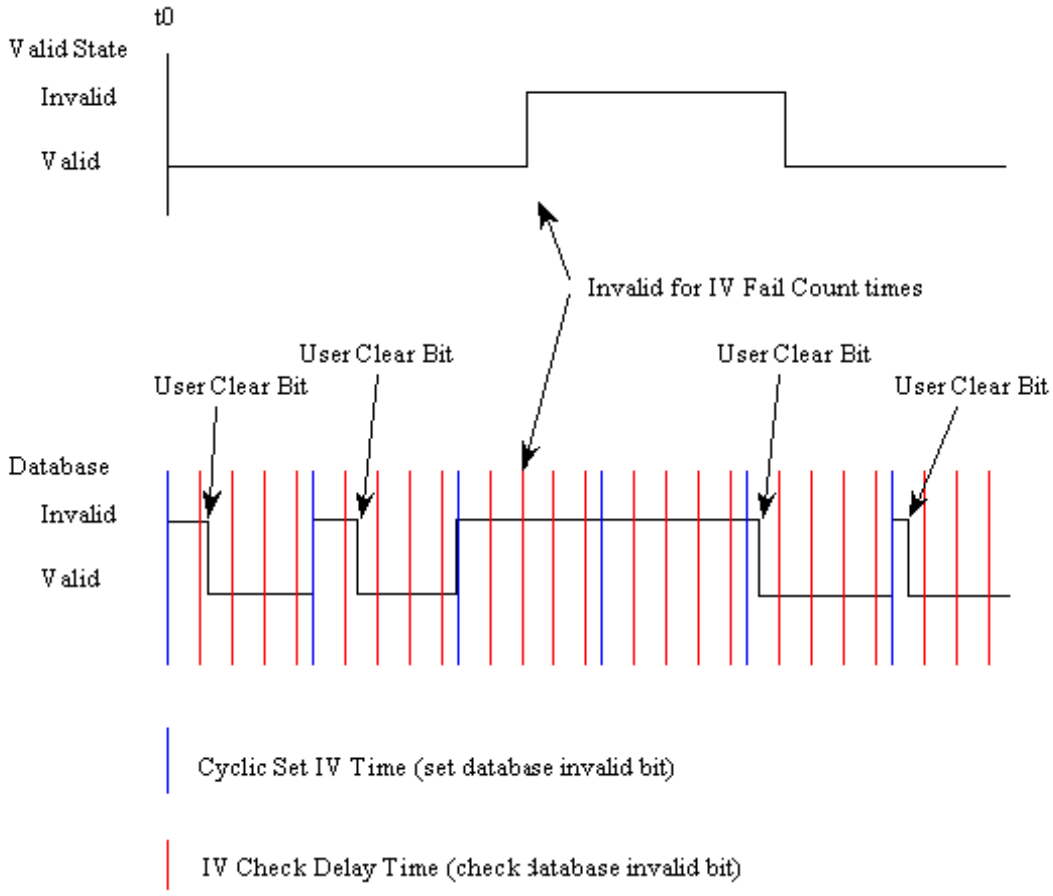
In order to configure points for this feature, the **IV DB BIT** field for each data type must be set to the database configuration records in the configuration file. For example;







The following illustration shows how these parameters are implemented:



If a database bit address (1 to 64000) is present, the application may consider the point with an invalid flag if the previous logic checks the IV bit as 1 during consecutive IV Check Delay scans. The IV bits would have to be reset to 0 to set the point to valid state.

The IV DB bit defined for each point can be unique, or many points may share the same bit. The last case could be used when the points on an I/O module are to be considered as one set. In this case only a single bit is required. For a point that is the result of a computation, the valid quality state could be set for each point individually.

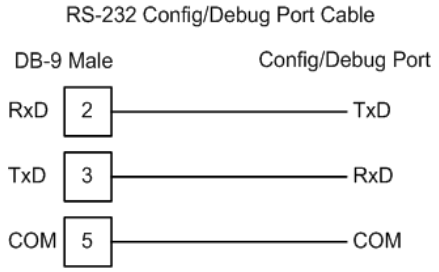
5.4 Cable Connections

The application ports on the MVI56-101S module support RS-232, RS-422, and RS-485 interfaces. Please ensure that the jumpers are set correctly for the type of interface you are using.

Note for modules with RS-232 connection to a radio or modem: Some radios or modems require hardware handshaking (control and monitoring of modem signal lines) on the RTS and CTS lines of an RS-232 connection. Enable this by setting the *UseCTS* parameter in the module configuration to 1.

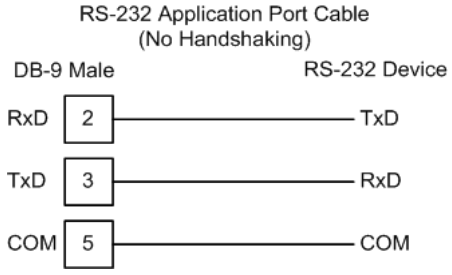
5.4.1 RS-232 Configuration/Debug Port

This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC-based terminal emulation program to view configuration and status data in the module and to control the module. The cable pinout for communications on this port is shown in the following diagram.



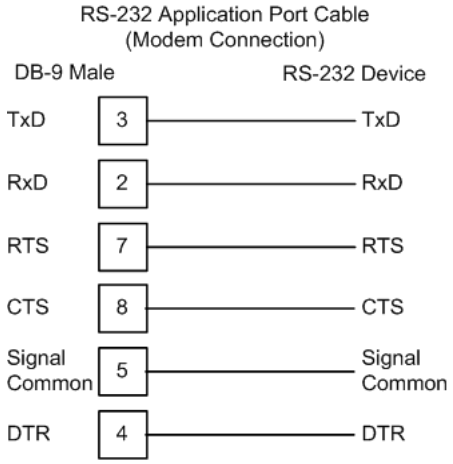
5.4.2 RS-232 Application Port(s)

When the RS-232 interface is selected, the use of hardware handshaking (control and monitoring of modem signal lines) is user definable. If no hardware handshaking will be used, here are the cable pinouts to connect to the port.



RS-232: Modem Connection (Hardware Handshaking Required)

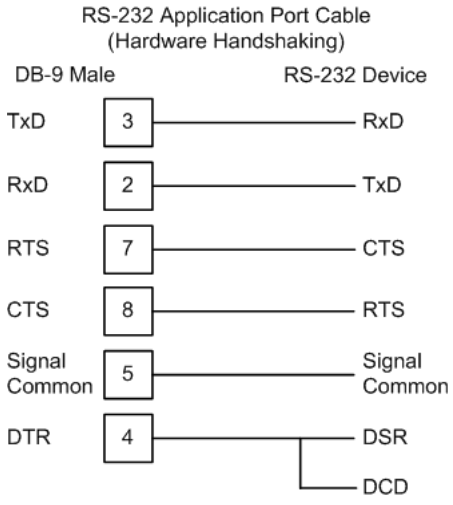
This type of connection is required between the module and a modem or other communication device.



The "Use CTS Line" parameter for the port configuration should be set to 'Y' for most modem applications.

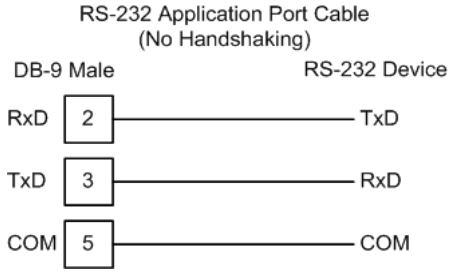
RS-232: Null Modem Connection (Hardware Handshaking)

This type of connection is used when the device connected to the module requires hardware handshaking (control and monitoring of modem signal lines).

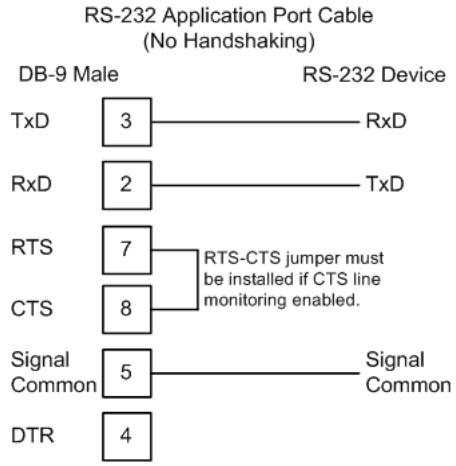


RS-232: Null Modem Connection (No Hardware Handshaking)

This type of connection can be used to connect the module to a computer or field device communication port.

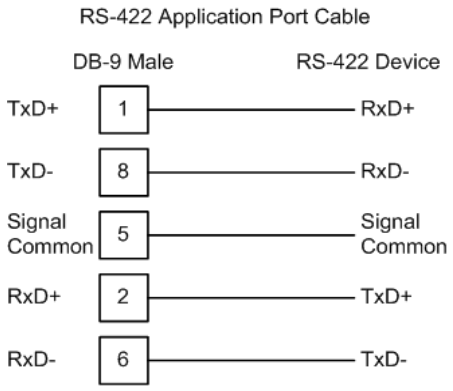


Note: For most null modem connections where hardware handshaking is not required, the *Use CTS Line* parameter should be set to N and no jumper will be required between Pins 7 (RTS) and 8 (CTS) on the connector. If the port is configured with the *Use CTS Line* set to Y, then a jumper is required between the RTS and the CTS lines on the port connection.



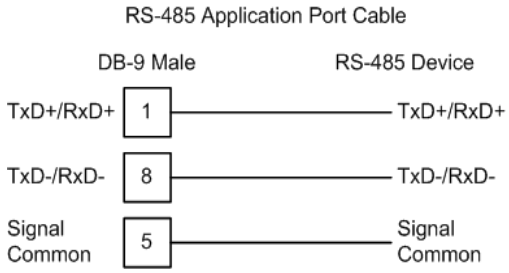
5.4.3 RS-422

The RS-422 interface requires a single four or five wire cable. The Common connection is optional, depending on the RS-422 network devices used. The cable required for this interface is shown below:



5.4.4 RS-485 Application Port(s)

The RS-485 interface requires a single two or three wire cable. The Common connection is optional, depending on the RS-485 network devices used. The cable required for this interface is shown below:

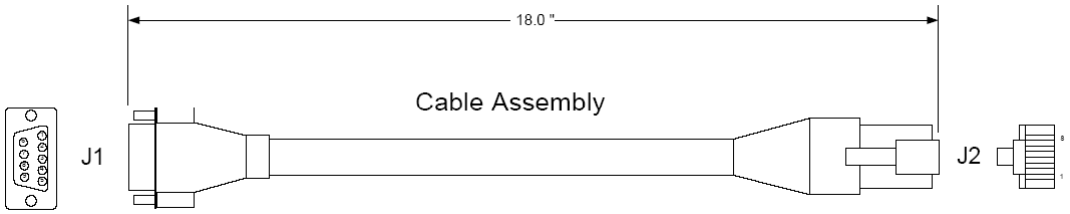


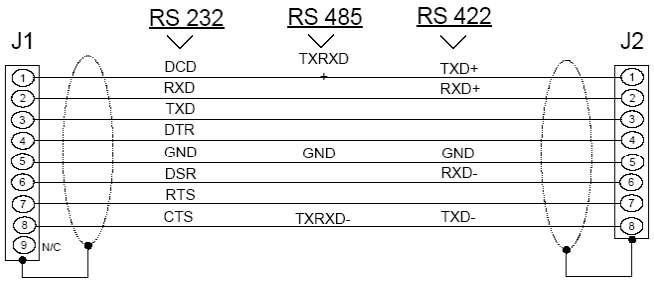
Note: Terminating resistors are generally not required on the RS-485 network, unless you are experiencing communication problems that can be attributed to signal echoes or reflections. In these cases, installing a 120-ohm terminating resistor between pins 1 and 8 on the module connector end of the RS-485 line may improve communication quality.

RS-485 and RS-422 Tip

If communication in the RS-422 or RS-485 mode does not work at first, despite all attempts, try switching termination polarities. Some manufacturers interpret + and -, or A and B, polarities differently.

5.4.5 DB9 to RJ45 Adaptor (Cable 14)





Wiring Diagram

5.5 MVI56-101S Status Data Definition

This section contains a description of the members present in the **InStat870** object. This data is transferred from the module to the processor as part of each read block.

Offset	Parameter	Description
0	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
1 to 2	Product Name	This two-word data area contains the text values representing the product name. These words contain the text "87S5" for the MVI56 platform.
3 to 4	Revision	This two-word data area contains the text values for the revision number.
5 to 6	Op Sys #	This two-word data area contains the text values for the operating system number.
7 to 8	Run Number	This two-word data area contains the text values for the run number.
9	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
10	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
11	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
12	Error Blk Cnt	This word contains the total number of block transfer errors.
13	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
14	Bad CKS	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
15	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
16	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
17	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
18	RX Frames	This word contains the total number of frames received by the module.
19	TX Frames	This word contains the total number of frames transmitted by the module.

5.6 MVI56-101S Error Status Table

This section contains a listing of the MVI56-101S module's status data area. This file is located at the MVI56-101S database starting at address 4000. You may also configure an additional area using the "Error Offset" parameter.

Offset	Parameter	Description
4000	Scan Count	This status value contains a counter incremented on each scan of the module's main loop.
4001 to 4002	Product Name	This two-word data area contains the text values representing the product name.
4003 to 4004	Revision	This two-word data area contains the text values for the revision number.
4005 to 4006	Op Sys #	This two-word data area contains the text values for the operating system number.
4007 to 4008	Run Number	This two-word data area contains the text values for the run number.
4009	Read Blk Cnt	This word contains the total number of block read operations successfully executed.
4010	Write Blk Cnt	This word contains the total number of block write operations successfully executed.
4011	Parse Blk Cnt	This word contains the total number of write blocks successfully parsed.
4012	Error Blk Cnt	This word contains the total number of block transfer errors.
4013	Port Selected	This parameter determines which port on the module is being utilized. If the value is set to 0, the primary port is being used. If the value is set to 1, the backup port is being utilized.
4014	Bad CKS	This word contains the total number of frames received by the module that contain a bad check-sum values in the message.
4015	Sync Errors	This word contains the total number of frames received by the module that have synchronization errors. Each frame in the protocol has a specific header that must be received in a fixed sequence. If this header is not received correctly, this word will be incremented, and the frame will be discarded.
4016	Length Errors	This word contains the total number of frames received by the module that do not have the correct length.
4017	Timeout	This word contains the total number of frames received by the module that were not received within the specified receive timeout parameter.
4018	RX Frames	This word contains the total number of frames received by the module.
4019	TX Frames	This word contains the total number of frames transmitted by the module.
4020	MSP Event Buffer	This parameter shows the number of events available in the event buffer for M_SP_NA points.
4021	MDP Event Buffer	This parameter shows the number of events available in the event buffer for M_DP_NA points
4022	MST Event Buffer	This parameter shows the number of events available in the event buffer for M_ST_NA points
4023	MMENA Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NA points
4024	MMENB Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NB points

Offset	Parameter	Description
4025	MMENC Event Buffer	This parameter shows the number of events available in the event buffer for M_ME_NC points.
4026	MIT Event Buffer	This parameter shows the number of events available in the event buffer for M_IT_NA points

5.7 MVI56-101S Database Design Forms

5.7.1 M_SP_NA, M_DP_NA, M_ST_NA, M_BO_NA, and M_IT_NA Form

Point Number	Database Address	Group Assignment	IV DB Bit Address

5.7.2 M_ME_NA and M_ME_NB Form

Point Number	Database Address	Group Assignment	Default Deadband	IV DB Bit Address

5.7.3 Form for All C_ (Command) Data Types, Except C_RC_NA

Point Number	Database Address	Monitor Point #	Monitor Database Address	Require Select

5.7.4 C_RC_NA Form

Point Number	Database Address	Monitor Point #	Monitor Database Address

6 IEC 60870-5-101 Slave Interoperability Statement

In This Chapter

This companion standard presents sets of parameters and alternatives from which subsets have to be selected to implement particular telecontrol systems. Certain parameter values, such as the number of octets in the COMMON ADDRESS of ASDUs represent mutually exclusive alternatives. This means that only one value of the defined parameters is admitted per system. Other parameters, such as the listed set of different process information in command and in monitor direction allow the specification of the complete set or subsets, as appropriate for given applications. This clause summarizes the parameters of the previous clauses to facilitate a suitable selection for a specific application. If a system is composed of equipment originating from different manufacturers it is necessary that all partners agree on the selected parameters.

Note: In addition, the full specification of a system may require individual selection of certain parameters for certain parts of the system, such as the individual selection of scaling factors for individually addressable measured values.

The selected parameters should be marked in the white boxes as follows:

☐ Function or ASDU is not used

☒ Function or ASDU is used as standardized (default)

A black check box indicates that the option cannot be selected in this companion standard.

6.1 System or Device

☐ System definition

☐ Controlling station definition (Master)

☒ Controlled station definition (slave)

6.2 Network Configuration

(Network-specific parameter)

☒	Point-to-point	☒	Multipoint-party line
☒	Multiple point-to-point	☒	Multipoint-star

6.3 Physical Layer

(Network-specific parameter)

6.3.1 Transmission Speed (Control Direction)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if > 1200 bit/s	Balanced interchange Circuit X.24/X.27	
☐ 100 bit/s	☒ 2400 bit/s	☒ 2400 bit/s	☐ 56000 bit/s
☐ 200 bit/s	☒ 4800 bit/s	☒ 4800 bit/s	☐ 64000 bit/s
☒ 300 bit/s	☒ 9600 bit/s	☒ 9600 bit/s	
☒ 600 bit/s		☒ 19200 bit/s	
☒ 1200 bit/s		☒ 38400 bit/s	

6.3.2 Transmission Speed (Monitor Direction)

Unbalanced interchange Circuit V.24/V.28 Standard	Unbalanced interchange Circuit V.24/V.28 Recommended if > 1200 bit/s	Balanced interchange Circuit X.24/X.27	
☐ 100 bit/s	☒ 2400 bit/s	☒ 2400 bit/s	☐ 56000 bit/s
☐ 200 bit/s	☒ 4800 bit/s	☒ 4800 bit/s	☐ 64000 bit/s
☒ 300 bit/s	☒ 9600 bit/s	☒ 9600 bit/s	
☒ 600 bit/s		☒ 19200 bit/s	
☒ 1200 bit/s		☒ 38400 bit/s	

6.4 Link Layer

(Network-specific parameter)

Frame format FT 1.2, single character 1 and the fixed time out interval are used exclusively in this companion standard.

Link transmission procedure	Address field of link
☒ Balanced transmission	☒ Not present (balanced transmission only)
☒ Unbalanced transmission	☒ One octet
	☒ Two octets
	☒ Structured

Frame length

☒ Unstructured

25 to 252 Maximum length L (number of octets)

6.5 Application Layer

6.5.1 Transmission Mode for Application Data

Mode 1 (Least significant octet first), as defined in clause 4.10 of IEC 60870-5-4, is used exclusively in this companion standard.

6.5.2 Common Address of ASDU - 101S

(System-specific parameter)

☒ One octet ☒ Two octets

6.5.3 Information Object Address

(System-specific parameter)

☒ One octet ☒ Structured

☒ Two octets ☒ Unstructured

☒ Three octets

6.5.4 Cause of Transmission

(System-specific parameter)

☒ One octet ☒ Two octets (with originator address)

6.5.5 Length of APDU - 101S

(System-specific parameter, specify the maximum length of the APDU per system)

The maximum length of the APDU in this protocol implementation is 252 (default). The maximum length may be reduced by the system.

The maximum length of APDU per system is 255.

6.6 Selection of Standard ASDUs

6.6.1 Process Information in Monitor Direction

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒	<1>	:= Single-point information	M_SP_NA_1
☒	<2>	:= Single-point information with time tag	M_SP_TA_1
☒	<3>	:= Double-point information	M_DP_NA_1
☒	<4>	:= Double-point information with time tag	M_DP_TA_1
☒	<5>	:= Step position information	M_ST_NA_1
☒	<6>	:= Step position information with time tag	M_ST_TA_1
☒	<7>	:= Bitstring of 32 bit	M_BO_NA_1
☒	<8>	:= Bitstring of 32 bit with time tag	M_BO_TA_1
☒	<9>	:= Measured value, normalized value	M_ME_NA_1
☒	<10>	:= Measured value, normalized value with time tag	M_ME_TA_1
☒	<11>	:= Measured value, scaled value	M_ME_NB_1
☒	<12>	:= Measured value, scaled value with time tag	M_ME_TB_1
☒	<13>	:= Measured value, short floating-point value	M_ME_NC_I
☒	<14>	:= Measured value, short floating-point value with time tag	M_ME_TC_1
☒	<15>	:= Integrated totals	M_IT_NA_1
☒	<16>	:= Integrated totals with time tag	M_IT_TA_1
☐	<17>	:= Event of protection equipment with time tag	M_EP_TA_1
☐	<18>	:= Packed start events of protection equipment with time tag	M_EP_TB_1
☐	<19>	:= Packed output circuit information of protection equipment with time tag	M_EP_TC_1
☐	<20>	:= Packed single-point information with status change detection	M_PS_NA_1
☐	<21>	:= Measured value, normalized value without quality description	M_ME_ND_1
☒	<30>	:= Single-point information with time tag CP56Time2a	M_SP_TB_1
☒	<31>	:= Double-point information with time tag CP56Time2a	M_DP_TB_1
☒	<32>	:= Step position information with time tag CP56Time2a	M_ST_TB_1
☒	<33>	:= Bitstring of 32 bit with time tag CP56Time2a	M_BO_TB_1
☒	<34>	:= Measured value, normalized value with time tag CP56Time2a	M_ME_TD_1

☒	<35>	:= Measured value, scaled value with time tag CP56Time2a	M_ME_TE_1
☒	<36>	:= Measured value, short floating-point value with time tag CP56Time2a	M_ME_TF_1
☒	<37>	:= Integrated totals with time tag CP56Time2a	M_IT_TB_1
☐	<38>	:= Event of protection equipment with time tag CP56Time2a	M_EP_TD_1
☐	<39>	:= Packed start events of protection equipment with time tag CP56time2a	M_EP_TE_1
☐	<40>	:= Packed output circuit information of protection equipment with time tag CP56Time2a	M_EP_TF_1

6.6.2 Process Information in Control Direction

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒	<45>	:= Single command	C_SC_NA_1
☒	<46>	:= Double command	C_DC_NA_1
☒	<47>	:= Regulating step command	C_RC_NA_1
☒	<48>	:= Set point command, normalized value	C_SE_NA_1
☒	<49>	:= Set point command, scaled value	C_SE_NB_1
☒	<50>	:= Set point command, short floating-point value	C_SE_NC_1
☒	<51>	:= Bitstring of 32 bit	C_BO_NA_1
☐	<58>	:= Single command with time tag CP56Time2a	C_SC_TA_1
☐	<59>	:= Double command with time tag CP56Time2a	C_DC_TA_1
☐	<60>	:= Regulating step command with time tag CP56Time2a	C_RC_TA_1
☐	<61>	:= Set point command, normalized value with time tag CP56Time2a	C_SE_TA_1
☐	<62>	:= Set point command, scaled value with time tag CP56Time2a	C_SE_TB_1
☐	<63>	:= Set point command, short floating-point value with time tag CP56Time2a	C_SE_TC_1
☐	<64>	:= Bitstring of 32 bit with time tag CP56Time2a	C_BO_TA_1

Either the ASDUs of the set <45>-<51> or of the set <58>-<64> are used.

6.6.3 System Information in Monitor Direction

(Station-specific parameter; mark "**X**" if used)

☒	<70>	:= End of initialization	M_EI_NA_1

6.6.4 System Information in Control Direction

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒	<100>	:= Interrogation command	C_IC_NA_1
☒	<101>	:= Counter interrogation command	C_CI_NA_1
☒	<102>	:= Read command	C_RD_NA_1
☒	<103>	:= Clock synchronization command	C_CS_NA_1
☒	<104>	:= Test command	C_TS_NB_1
☒	<105>	:= Reset process command	C_RP_NC_1
☒	<106>	:= Delay acquisition command [Note 1]	C_CD_NA_1

6.6.5 Parameter in Control Direction

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒	<110>	:= Parameter of measured value, normalized value	P_ME_NA_1
☒	<111>	:= Parameter of measured value, scaled value	P_ME_NB_1
☒	<112>	:= Parameter of measured value, short floating-point value	P_ME_NC_1
☐	<113>	:= Parameter activation	P_AC_NA_1

6.6.6 File Transfer

(Station-specific parameter; mark each Type ID "**X**" if it is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐	<120>	:= File ready	F_FR_NA_1
☐	<121>	:= Section ready	F_SR_NA_1
☐	<122>	:= Call directory, select file, call file, call section	F_SC_NA_1
☐	<123>	:= Last section, last segment	F_LS_NA_1
☐	<124>	:= Ack file, ack section	F_AF_NA_1
☐	<125>	:= Segment	F_SG_NA_1
☐	<126>	:= Directory	F_DR_TA_1

6.7 Type Identifier and Cause of Transmission Assignments

(Station-specific parameters)

Shaded boxes: option not required

Black boxes: option not permitted in this companion standard

Blank boxes: functions or ASDU not used

Mark Type Identification/Cause of Transmission combinations: "**X**" if only used in standard direction, "**R**" if only used in reverse direction, and "**B**" if used in both directions.

Type Identification		Cause of transmission																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<1>	M_SP_NA_1		X	X		X						X			X					
<3>	M_DP_NA_1		X	X		X						X			X					
<5>	M_ST_NA_1		X	X		X						X			X					
<7>	M_BO_NA_1		X	X		X						X			X					
<9>	M_ME_NA_1	X		X		X						X			X					
<11>	M_ME_NB_1	X		X		X						X			X					
<13>	M_ME_NC_1	X		X		X						X			X					
<15>	M_IT_NA_1		X	X		X										X				
<20>	M_PS_NA_1																			
<21>	M_ME_ND_1																			
<30>	M_SP_TB_1			X																
<31>	M_DP_TB_1			X																
<32>	M_ST_TB_1			X																
<33>	M_BO_TB_1			X																
<34>	M_ME_TD_1			X																
<35>	M_ME_TE_1			X																
<36>	M_ME_TF_1			X																
<37>	M_IT_TB_1			X												X				
<38>	M_EP_TD_1																			
<39>	M_EP_TE_1																			
<40>	M_EP_TF_1																			
<45>	C_SC_NA_1						X	X	X	X	X						X	X	X	X
<46>	C_DC_NA_1						X	X	X	X	X						X	X	X	X
<47>	C_RC_NA_1						X	X	X	X	X						X	X	X	X
<48>	C_SE_NA_1						X	X	X	X	X						X	X	X	X
<49>	C_SE_NB_1						X	X	X	X	X						X	X	X	X
<50>	C_SE_NC_1						X	X	X	X	X						X	X	X	X
<51>	C_BO_NA_1						X	X			X						X	X	X	X
<58>	C_SC_TA_1						X	X	X	X	X						X	X	X	X
<59>	C_DC_TA_1						X	X	X	X	X						X	X	X	X
<60>	C_RC_TA_1						X	X	X	X	X						X	X	X	X
<61>	C_SE_TA_1						X	X	X	X	X						X	X	X	X
<62>	C_SE_TB_1						X	X	X	X	X						X	X	X	X
<63>	C_SE_TC_1						X	X	X	X	X						X	X	X	X
<64>	C_BO_TA_1						X	X			X						X	X	X	X
<70>	M_EI_NA_1				X															
<100>	C_IC_NA_1						X	X	X	X	X						X	X	X	X
<101>	C_CI_NA_1						X	X			X						X	X	X	X
<102>	C_RD_NA_1					X											X	X	X	X
<103>	C_CS_NA_1						X	X									X	X	X	X
<105>	C_RP_NA_1						X	X									X	X	X	X
<107>	C_TS_TA_1						X	X									X	X	X	X
<110>	P_ME_NA_1						X	X									X	X	X	X
<111>	P_ME_NB_1						X	X									X	X	X	X

Type Identification		Cause of transmission																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	20 to 36	37 to 41	44	45	46	47
<112>	P_ME_NC_1						X	X									X	X	X	X
<113>	P_AC_NA_1																			
<120>	F_FR_NA_1																			
<121>	F_SR_NA_1																			
<122>	F_SC_NA_1																			
<123>	F_LS_NA_1																			
<124>	F_AF_NA_1																			
<125>	F_SG_NA_1																			
<126>	F_DR_TA_1																			
<127>	F_SC_NB_1																			

6.8 Basic Application Functions

6.8.1 Station Initialization

(Station-specific parameter; mark "**X**" if function is used)

☒ Remote initialization

6.8.2 Cyclic Data Transmission

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Cyclic data transmission

6.8.3 Read Procedure

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Read procedure

6.8.4 Spontaneous Transmission

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Spontaneous transmission

6.8.5 Double Transmission of Information Objects

(Station-specific parameter; mark each information type "**X**" where both a Type ID without time and corresponding Type ID with time are issued in response to a single spontaneous change of a monitored object)

The following type identifications may be transmitted in succession caused by a single status change of an information object. The particular Information Object Addresses for which double transmission is enabled are defined in a project-specific list.

☐ Single-point information M_SP_NA_1, M_SP_TA_1, M_SP_TB_1 and M_PS_NA_1

☐ Double-point information M_DP_NA_1, MDP_TA_1 and M_DP_TB_1

☐ Step position information M_ST_NA_1, M_ST_TA_1 and M_ST_TB_1

☐ Bitstring of 32 bit M_BO_NA_1, M_BO_TA_1 and M_BO_TB_1

☐ Measured value, normalized value M_ME_NA_1, M_ME_TA_1, M_ME_ND_1 and M_ME_TD_1

☐ Measured value, scaled value M_ME_NB_1, M_ME_TB_1 and M_ME_TE_1

☐ Measured value, short floating-point number M_ME_NC_1, M_ME_TC_1 and M_ME_TF_1

6.8.6 Station Interrogation

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒	global				
☒	group 1	☒	group 7	☒	group 13
☒	group 2	☒	group 8	☒	group l4
☒	group 3	☒	group 9	☒	group 15
☒	group 4	☒	group 10	☒	group 16
☒	group 5	☒	group 11		
☒	group 6	☒	group 12		

6.8.7 Clock Synchronization

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Clock synchronization

☐ Day of week used

☐ RES1, GEN (time tag substituted/not substituted) used

☐ SU-bit (summertime) used (Optional)

6.8.8 Command Transmission

(Object-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Direct command transmission

☒ Direct set point command transmission

☒ Select and execute command

☒ Select and execute set point command

☒ C_SE_ACTTERM used [note 2]

☒ No additional definition

☒ Short pulse duration (duration determined by a system parameter in the outstation)

☒ Long pulse duration (duration determined by a system parameter in the outstation)

☒ Persistent output

☐ Supervision of maximum delay in command direction of commands and set point commands

60 seconds Maximum allowable delay of commands and set point commands

6.8.9 Transmission of Integrated Totals

(Station- or object-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Mode A: Local freeze with spontaneous transmission

☐ Mode B: Local freeze with counter interrogation

☐ Mode C: Freeze and transmit by counter-interrogation commands

☒ Mode D: Freeze by counter-interrogation command, frozen values reported spontaneously

☒ Counter read

☒ Counter freeze without reset

☐ Counter freeze with reset

☐ Counter reset

☒ General request counter

☒ Request counter group 1

☒ Request counter group 2

☒ Request counter group 3

☒ Request counter group 4

6.8.10 Parameter Loading

(Object-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☒ Threshold value

☐ Smoothing factor

☒ Low limit for transmission of measured value

☒ High limit for transmission of measured value

6.8.11 Parameter Activation

(Object-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Act/deact of persistent cyclic or periodic transmission of the addressed object

6.8.12 Test Procedure

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Test procedure

6.8.13 File Transfer

(Station-specific parameter; mark "**X**" if function is used)

File transfer in monitor direction

☐ Transparent file

☐ Transmission of disturbance data of protection equipment

☐ Transmission of sequence of events

☐ Transmission of sequence of recorded analogue values

File transfer in control direction

☐ Transparent file

6.8.14 Background Scan

(Station-specific parameter; mark "**X**" if function is only used in the standard direction, "**R**" if only used in the reverse direction, and "**B**" if used in both directions)

☐ Background scan

6.8.15 Aquisition of transmission delay

(station-specific parameter, mark with an "**X**" if function is used only in the standard direction, "**R**" if used only in the reverse direction, and "**B**" if used in both directions)

☐ Acquisition of transmission delay

7 Support, Service & Warranty

7.1 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, ProSoft's 24/7 after-hours phone support is available for urgent plant-down issues.*

North America (Corporate Location)	Europe / Middle East / Africa Regional Office
Phone: +1.661.716.5100 info@prosoft-technology.com Languages spoken: English, Spanish REGIONAL TECH SUPPORT support@prosoft-technology.com	Phone: +33.(0)5.34.36.87.20 france@prosoft-technology.com Languages spoken: French, English REGIONAL TECH SUPPORT support.emea@prosoft-technology.com
Latin America Regional Office	**Asia Pacific Regional Office**
Phone: +52.222.264.1814 latinam@prosoft-technology.com Languages spoken: Spanish, English REGIONAL TECH SUPPORT support.la@prosoft-technology.com	Phone: +60.3.2247.1898 asiapc@prosoft-technology.com Languages spoken: Bahasa, Chinese, English, Japanese, Korean REGIONAL TECH SUPPORT support.ap@prosoft-technology.com

For additional ProSoft Technology contacts in your area, please visit:
https://www.prosoft-technology.com/About-Us/Contact-Us.

7.2 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 at:
www.prosoft-technology/legal