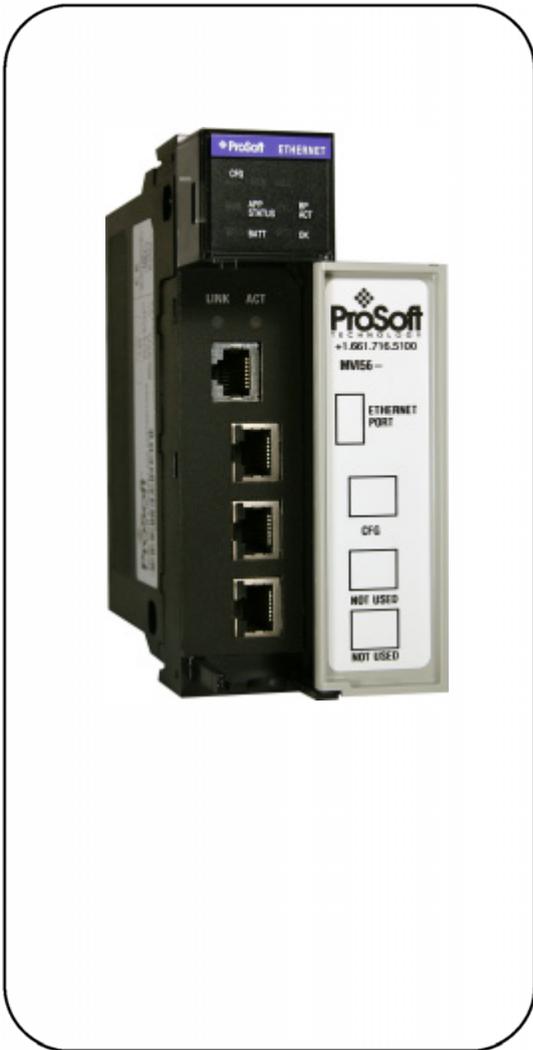


inRAx



MVI56-SMGC

ControlLogix Platform

SM General Communication Protocol
Module

User Manual

August 6, 2021



Please Read This Notice

Successful application of this module requires a reasonable working knowledge of the Rockwell Automation ControlLogix hardware, the MVI56-SMGC Module and the application in which the combination is to be used. For this reason, it is important that those responsible for implementation satisfy themselves that the combination will meet the needs of the application without exposing personnel or equipment to unsafe or inappropriate working conditions.

This manual is provided to assist the user. Every attempt has been made to ensure that the information provided is accurate and a true reflection of the product's installation requirements. In order to ensure a complete understanding of the operation of the product, the user should read all applicable Rockwell Automation documentation on the operation of the Rockwell Automation hardware.

Under no conditions will ProSoft Technology be responsible or liable for indirect or consequential damages resulting from the use or application of the product.

Reproduction of the contents of this manual, in whole or in part, without written permission from ProSoft Technology is prohibited.

Information in this manual is subject to change without notice and does not represent a commitment on the part of ProSoft Technology Improvements and/or changes in this manual or the product may be made at any time. These changes will be made periodically to correct technical inaccuracies or typographical errors.

Warnings

UL Warnings

- 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.
Warning - Explosion Hazard - Do not disconnect equipment unless power has been switched off or the area is known to be nonhazardous.
- C** 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.

Electrical Ratings

- Backplane Current Load: 800 mA @ 5 V DC; 3mA @ 24V DC
- Operating Temperature: 0 to 60°C (32 to 140°F)
- Storage Temperature: -40 to 85°C (-40 to 185°F)
- Shock: 30g Operational; 50g non-operational; Vibration: 5 g from 10 to 150 Hz
- Relative Humidity 5% to 95% (non-condensing)
- All phase conductor sizes must be at least 1.3 mm(squared) and all earth ground conductors must be at least 4mm(squared).

Markings:



II 3 G 0C <=Ta<= 60C EEx nA IIC T4 DEMKO 07ATEX0710717X

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 the product, documentation or support, please write or call us.

ProSoft Technology, Inc.

+1 (661) 716-5100

+1 (661) 716-5101 (Fax)

<http://www.prosoft-technology.com>

Copyright © ProSoft Technology, Inc. 2021. All Rights Reserved.

MVI56-SMGC User Manual

August 6, 2021

ProSoft Technology ®, ProLinx ®, inRAx ®, ProTalk® and RadioLinx ® are Registered Trademarks of ProSoft Technology, Inc.



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

Contents

PLEASE READ THIS NOTICE	2
Warnings	2
Your Feedback Please	3
GUIDE TO THE MVI56-SMGC USER MANUAL	7
1 START HERE	9
1.1 System Requirements.....	9
1.2 Package Contents	10
1.3 Install the Module in the Rack.....	11
1.4 Connect your PC to the Processor	12
1.5 Download the Sample Program to the Processor	13
1.5.1 Configuring RSLinx	15
1.6 Connect your PC to the Module.....	17
2 MODULE CONFIGURATION	19
2.1 Installing and Configuring the Module.....	19
2.2 Obtain the Sample Configuration Files	24
2.3 Modifying the Configuration File	24
2.3.1 [Module]	26
2.3.2 [SMGC Client 0]	26
2.3.3 [AI Data]	27
2.3.4 [AO Data]	28
2.3.5 [DI Data]	28
2.3.6 [DO Data]	28
2.4 Uploading and Downloading the Configuration File	28
2.4.1 Required Hardware	29
2.4.2 Required Software.....	29
2.4.3 Transferring the Configuration File to Your PC.....	30
2.4.4 Transferring the Configuration File to the Module	32
2.5 IP Address	35
2.6 Module Data	36
2.6.1 Main Data Object (SMGCModule).....	36
2.6.2 Database Values	40
2.6.3 Helper Objects.....	42
3 LADDER LOGIC	43
4 DIAGNOSTICS AND TROUBLESHOOTING - MVI & PTQ	45
4.1 Reading Status Data From the Module.....	45
4.1.1 The Configuration/Debug Menu	45
4.1.2 Required Hardware	46
4.1.3 Required Software.....	47
4.1.4 Using the Configuration/Debug Port.....	47
4.1.5 Main Menu	48
4.1.6 Network Menu	51
4.2 LED Status Indicators	52
4.2.1 Ethernet LED Indicators	52
4.2.2 Clearing a Fault Condition	52
4.2.3 Troubleshooting.....	53
5 REFERENCE	55
5.1 Product Specifications	55
5.1.1 Features and Benefits	55
5.1.2 General Specifications	55
5.1.3 Hardware Specifications.....	56

5.1.4	Functional Specifications	56
5.2	Functional Overview	57
5.2.1	General Concepts	57
5.3	Cable Connections	75
5.3.1	Ethernet Connection	76
5.3.2	RS-232 Configuration/Debug Port.....	76
5.3.3	DB9 to RJ45 Adaptor (Cable 14).....	79
5.4	MVI56-SMGC Status Data Area	79
5.5	MVI56-SMGC Server Communication Module Configuration	81
6	SUPPORT, SERVICE & WARRANTY.....	83

Guide to the MVI56-SMGC User Manual

Function		Section to Read	Details
Introduction (Must Do)	→	Start Here (page 9)	This Section introduces the customer to the module. Included are: package contents, system requirements, hardware installation, and basic configuration.
Verify Communication, Diagnostic and Troubleshooting	→	Verifying Communication (page 45) Diagnostics and Troubleshooting (page 45)	This section describes how to verify communications with the network. Diagnostic and Troubleshooting procedures.
Reference Product Specifications Functional Overview Glossary	→	Reference (page 55) Functional Overview (page 57) Product Specifications (page 55)	These sections contain general references associated with this product, Specifications, and the Functional Overview.
Support, Service, and Warranty Index	→	Support, Service and Warranty (page 83)	This section contains Support, Service and Warranty information. Index of chapters.

1 Start Here

In This Chapter

- System Requirements 9
- Package Contents 10
- Install the Module in the Rack 11
- Connect your PC to the Processor 12
- Download the Sample Program to the Processor 13
- Connect your PC to the Module 17

Installing the MVI56-SMGC module requires a reasonable working knowledge of the Rockwell Automation ControlLogix hardware, the MVI56-SMGC Module and the application in which they will be used.



Caution: It is important that those responsible for implementation can complete the application without exposing personnel, or equipment, to unsafe or inappropriate working conditions. Safety, quality and experience are key factors in a successful installation.

1.1 System Requirements

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

- Rockwell Automation ControlLogix™ processor, with compatible power supply and one free slot in the rack, for the MVI56-SMGC module. The module requires 800mA 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 x 768 recommended)
- 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.

1.2 Package Contents

The following components are included with your MVI56-SMGC 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-SMGC Module	MVI56-SMGC	SM General Communication Protocol Module
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection to the CFG Port
1	Cable	RJ45 to DB9 Male Adapter	For DB9 Connection to Module's Port

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

1.3 Install the Module in the Rack

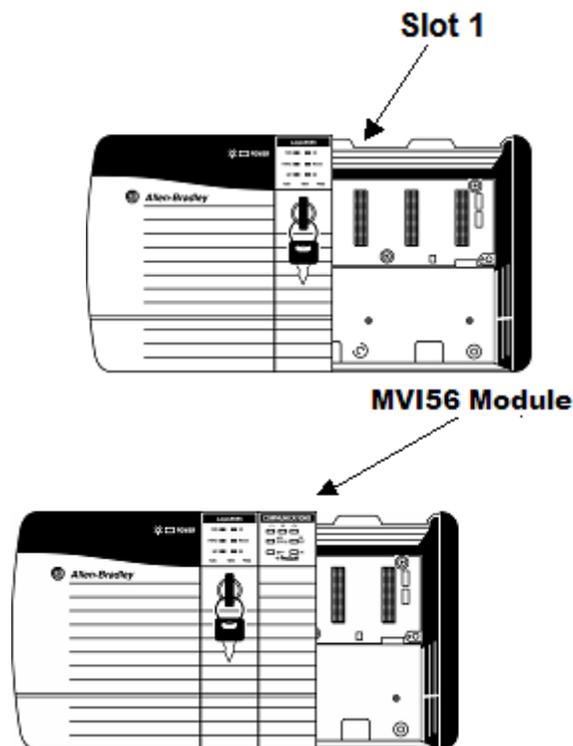
If you have not already installed and configured your ControlLogix processor and power supply, please do so before installing the MVI56-SMGC module. Refer to your Rockwell Automation product documentation for installation instructions.

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-SMGC 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.



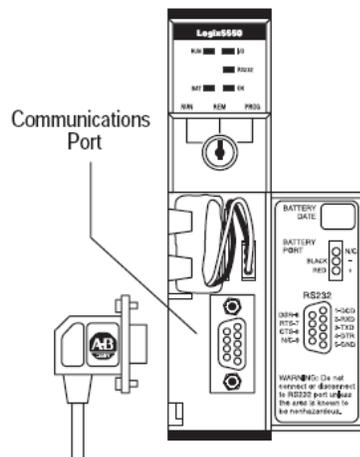
- 3 With a firm but steady push, snap the module into place.
- 4 Check that the holding clips on the top and bottom of the module are securely in the locking holes of the rack.
- 5 Make a note of the slot location. You will need to identify the slot in which the module is installed in order for the sample 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.

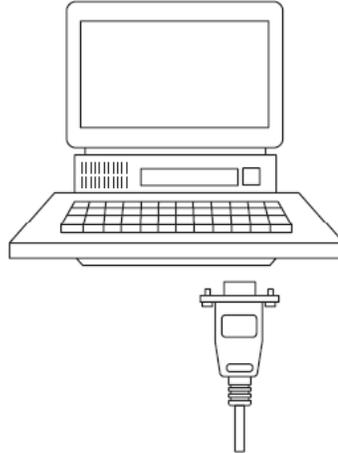
Note: If you are installing MVI56-SMGC with other modules connected to the PCI bus, the peripheral modules will not have holding clips. Make sure all of the modules are aligned with their respective slots before you snap them into place.

1.4 Connect your PC to the Processor

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



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

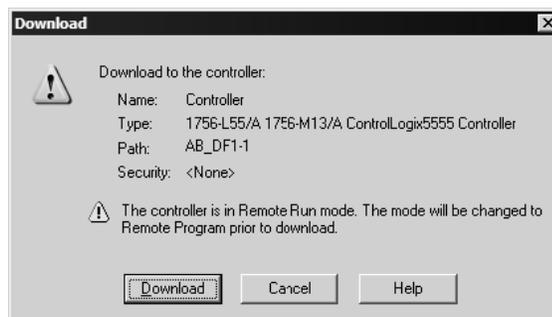


1.5 Download the Sample Program to the Processor

- **To download the sample program from RSLogix 5000 to the ControlLogix processor:**

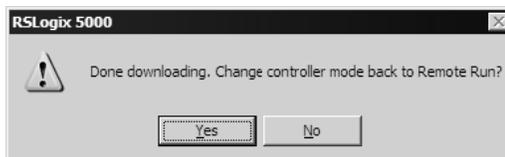
Note: The key switch on the front of the ControlLogix module must be in the REM position.

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



- 3 RSLogix will compile the program and transfer it to the processor. This process may take a few minutes.

- 4 When the download is complete, RSLogix will open another confirmation dialog box. Click OK to switch the processor from Program mode to Run mode.

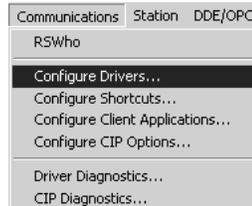


Note: If you receive an error message during these steps, refer to your RSLogix documentation to interpret and correct the error.

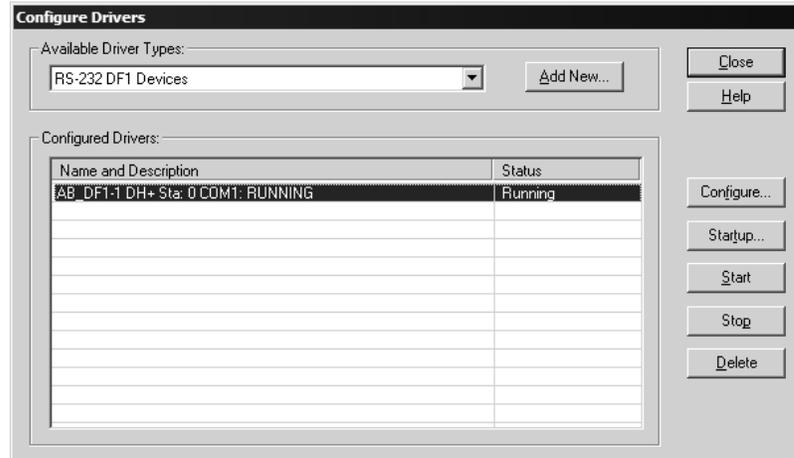
1.5.1 Configuring RSLinx

➤ **If RSLinx 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 Allen-Bradley DF1 Communications Device dialog box.



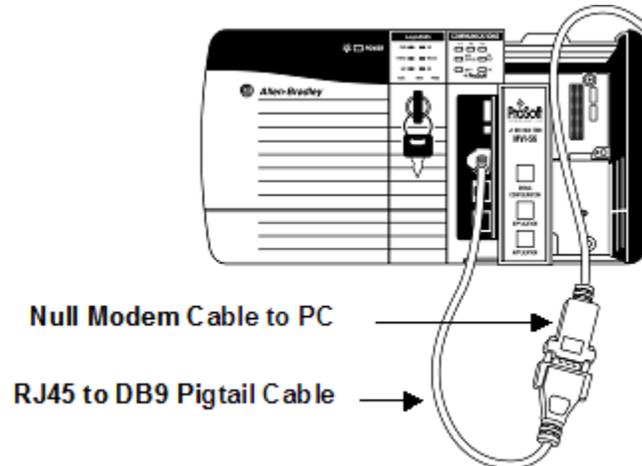
- 4 Click the Auto-Configure button. RSLinx will attempt to configure your serial port to work with the selected driver.
- 5 When you see the message "Auto Configuration Successful", click the OK button to dismiss the dialog box.

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.

1.6 Connect 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 Configuration/Debug port of the module.
- 3 Attach the other end to the serial port on your PC or laptop.



2 Module Configuration

In This Chapter

- Installing and Configuring the Module 19
- Obtain the Sample Configuration Files..... 24
- Modifying the Configuration File..... 24
- Uploading and Downloading the Configuration File 28
- IP Address 35
- Module Data..... 36

This section contains the setup procedure, data, and ladder logic for successful application of the MVI56-SMGC module. Each step in the setup procedure is defined in order to simplify the use of the module.

2.1 Installing and Configuring the Module

This chapter describes how to install and configure the module to work with your application. The configuration process consists of the following steps.

- 1 Use RSLogix to identify the module to the processor and add the module to a project.

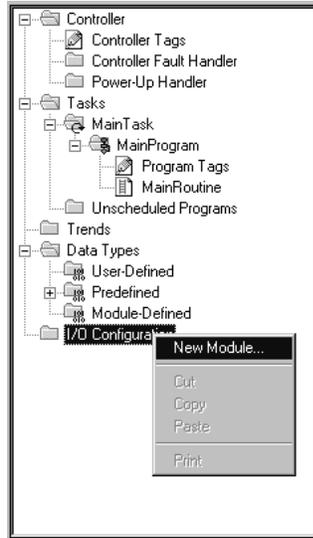
Note: The RSLogix software must be in "offline" mode to add the module to a project.

- 2 Modify the module's configuration files to meet the needs of your application, and copy the updated configuration to the module. Refer to the Modifying the Example Configuration File section, later in this chapter, for more information on the configuration files.
- 3 Modify the example ladder logic to meet the needs of your application, and copy the ladder logic to the processor.

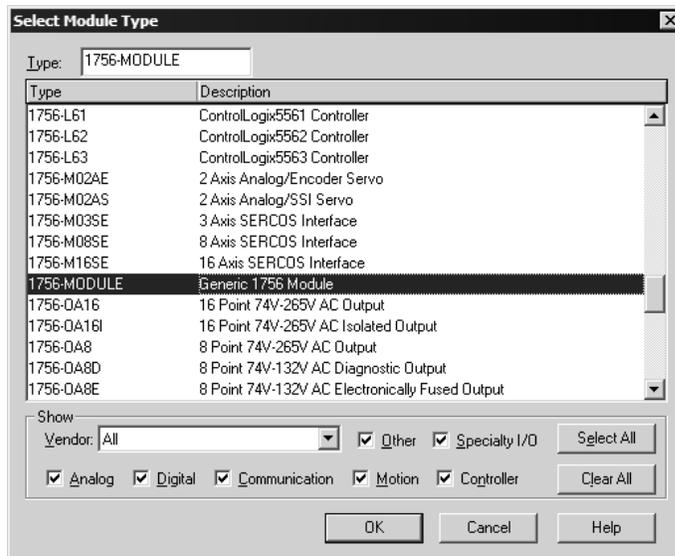
Note: If you are installing this module in an existing application, you can copy the necessary elements from the example ladder logic into your application.

The rest of this chapter describes these steps in more detail.

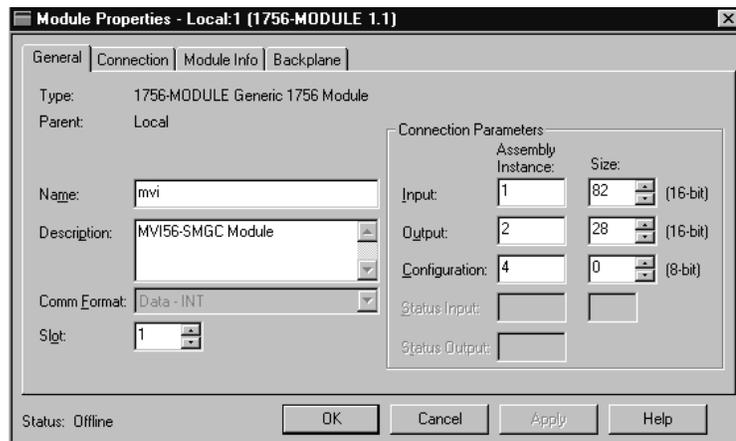
The first step in installing and configuring the module is to define the module to the system. Right-click the mouse button on the I/O Configuration option in the Controller Organization window to display a pop-up menu. Select the New Module option from the I/O Configuration menu.



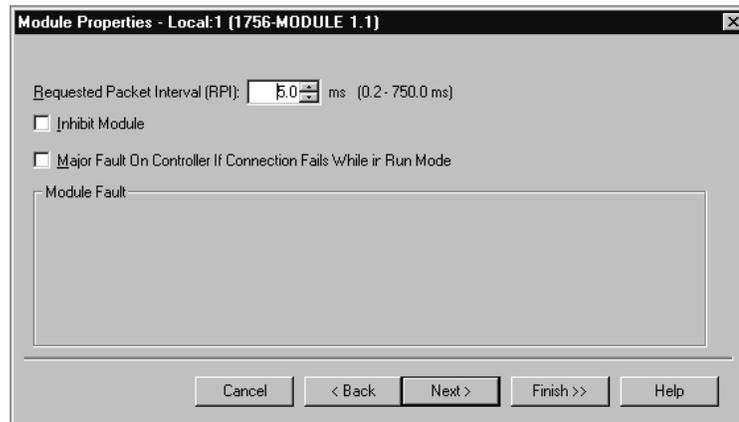
This action opens the following dialog box.



Select the 1756-Module (Generic 1756 Module) from the list and click OK. The following dialog box will be displayed.

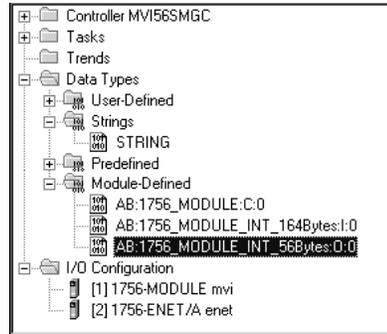


Fill in the dialog box as shown adjusting the Name, Description and Slot options for your application. You must select the **Comm Format** as **Data - INT** in the dialog box. Failure to set the **Assembly Instance** and **Size** values correctly will result in a module that will not communicate over the backplane of the ControlLogix rack. Click Next to display the next dialog box.

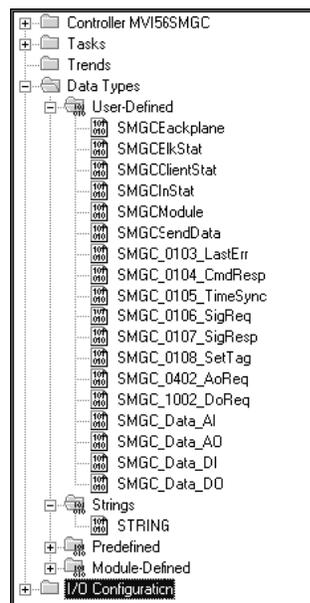


Select the Request Packet Interval value for scanning the I/O on the module. This value represents the minimum frequency that the module will handle scheduled events. This value should not be set to less than 1 millisecond. Values between 1 and 10 milliseconds should work with most applications.

After completing the module setup, the Controller Organization window will display the module's presence. The data required for the module will be defined to the application, and objects will be allocated in the Controller Tags data area. An example of the Controller Organization window is shown below.



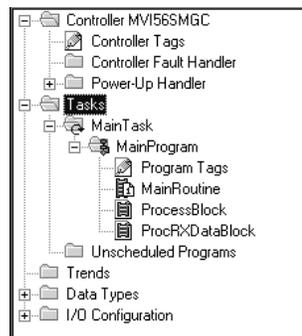
The next step in the module's setup is to define the User Defined Data Types to be used with the module. Copy these data types from the example ladder logic if you are not using the example. They will be defined if you are starting from the example ladder logic. The Controller Organization window should display the User Defined Data Types shown below:



The next step in module setup is to define the data tags to be used to interface with the module and the ladder logic. Open the Controller Tags Edit Tags dialog box and enter the values shown in the following example. The MVI56-SMGC module is defined in the example as SMGC and has a type of SMGCModule. The tag name can be set to any valid tag name desired. When using the example ladder logic, this step has already been performed.

Tag Name	Value	Style	Type	Description
ColdBoot	2#0000_0000	Binary	BOOL	Cold boot MVI56-SMGC module
Local1:C	{...}		AB:1756_MODULE:C:0	
Local1:I	{...}		AB:1756_MODULE:I...	
Local1:O	{...}		AB:1756_MODULE:I...	
MJFAULTS	{...}	Decimal	DINT[12]	
PingRequest	16#0000_0102	Hex	DINT	Ping data block ID
SendClient	{...}	Binary	BOOL[32]	Bits to send different data blocks
Setpoint	0.0	Float	REAL	Setpoint for AO
SMGC	{...}		SMGCModule	MVI56-SMGC main data object
SMGC_Data_AI	{...}		SMGC_Data_AI[253]	Analog input data
SMGC_Data_AO	{...}		SMGC_Data_AO[329]	Analog output data
SMGC_Data_DI	{...}		SMGC_Data_DI[131]	Digital input data
SMGC_Data_DO	{...}		SMGC_Data_DO[148]	Digital output data
WallClockTime	{...}	Decimal	DINT[7]	

The last step in the module setup is to add the ladder logic. If the example ladder logic is used, adjust the ladder to fit the application. When the ladder example is not used, copy the ladder logic shown in the Controller Organization window below to the application.



It is now time to edit the SMGC.CFG file to set up the module for the specific application. Refer to the Configuration File section of this document. Additionally, you must edit the WATTCP.CFG file to match the network settings required for your application. This file contains the IP address to be used by the module on the network. It also contains the IP address of the gateway to utilize on the network. The module is now set up and ready to be used with your application. Insert the module in the rack and attach the serial communication cable to the debug/configuration port and the Ethernet port to a hub.

Download the new application to the controller and place the processor in run mode. Download the new SMGC.CFG and WATTCP.CFG files to the processor using the debug/configuration port. If all the configuration parameters are set correctly and the module is attached to an Ethernet network, the module's Application LED (APP LED) should remain off and the backplane activity LED (BP ACT) should blink rapidly. Refer to the Trouble Shooting section if you encounter errors. Attach a computer or terminal to Port 1 on the module and look at the status of the module using the Configuration/Debug Menu in the module.

2.3 Modifying the Configuration File

In order for the module to operate, a configuration file (SMGC.CFG) is required. This configuration file contains information to set the data transfer characteristics between the module and the processor, to configure the communication ports, to establish the SMGC protocol parameters and to define the databases required to hold the protocol data sets. Each parameter in the file must be set carefully in order for the application to be implemented successfully.

The configuration file is separated into sections with topic header names enclosed in the [] characters. The configuration file consists of the following topics:

[Section]	Description
[Module]	General module configuration information
[SMGC Client 0]	Configuration information from the specific client
[AI Data]	Section containing list of AI tag names
[AO Data]	Section containing list of AO tag names
[DI Data]	Section containing list of DI tag names
[DO Data]	Section containing list of DO tag names

After each section header, the file contains a set of parameters. Unique labels are used under each section to specify a parameter. Each label in the file must be entered exactly as shown in the file for the parameter to be identified by the program. If the module is not considering a parameter, look at the label for the data item. Each parameter's value is separated from the label with the ':' character. This character is used by the program to delimit the position in the data record where to start reading data. All data for a parameter must be placed after the ':' character. For numeric parameter values any text located after the value will not be used. There must be at least one space character between the end of the parameter value and the following text. The following is an example of a parameter entry:

Module Name : MVI56-SMGC Communication Module

The parameter label is "Module Name" and the parameter value is "MVI56-SMGC Communication Module". The characters after the parameter value are ignored and are used for internal documentation of the configuration file.

Any record that begins with the '#' character is considered to be a comment record. These records can be placed anywhere in the file as long as the '#' character is found in the first column of the line. These lines are ignored in the file and can be used to provide documentation within the configuration file. Liberal use of comments within the file can ease the use and interpretation of the data in the file.

The database definition sections in the database are formatted differently than the other sections. These sections contain lists of parameters to be used with each point in the database. Each list of points in begins with the label **START** and ends when the number of points specified for the data type is reached or when the **END** label is reached. When entering the point list data, make certain that the first character in each line is left blank. The point number should never be present as the first character in the line it should be a space or tab character.

Important notes to consider when editing the sample configuration file:

- Comments within the file are preceded by the pound (#) sign. Any text on a line that occurs after the # character will be ignored.
- Do not use tabs or other non-printing characters instead of spaces to separate parameters (spacebar).
- Parameter names must begin in the first column of a line, and may not be preceded with a space (spacebar) or other non-printing character.

2.3.1 [Module]

This section of the file describes the database setup and module level parameters.

```
#  
[Module]  
Module Name : MVI56-SMGC Communication Module Test  
Password :
```

Module Name

0 to 80 characters

This parameter is used to assign a name to the module that can be viewed using the configuration/debug port. It can be used to identify the module and the configuration file.

Password

Up to 20 chars

This parameter is used to set the password for the module. If the parameter is not set, the module will not be password protected. If a password is present, it will be required for FTP access. The module interprets the password as a case-sensitive string. Do not include spaces or special characters in the password. Only alpha and numeric characters should be used.

2.3.2 [SMGC Client 0]

This section is used to define the single client present in the module. The Server IP Address must be set to match the unit the client will connect.

```
#  
[SMGC Client 0]  
Server IP Address : 140.140.0.2  
Service Port Number : 1201 #TCP port for this service on SM-TEM module  
Connection Timeout : 30000 #0=no timeout, else # of mSec before disconnect
```

Server IP Address

Up to 20 Chars.

This parameter is used to specify the IP address in dotted notation (for example, 192.168.0.100) of the server to interface with. If this parameter is set incorrectly, the client will never connect to the server and no communications will be possible.

Service Port Number

1 to 65535

This parameter sets the TCP/IP service port on the server to connect to. If this parameter is set incorrectly, the client will never connect to the server and no communications will be possible.

Connection Timeout

0 or 5000 to 65535

This parameter specifies the number of milliseconds the client will permit the socket to be inactive after a connection is made before closing the socket. This timeout period is reset on each read or write packet. If the parameter is set to 0, the connection will not timeout.

2.3.3 [AI Data]

These indexes are fixed in the module and cannot be changed by the user. They represent the starting tag index value for each data type to be used with the tags provided in the configuration

```
#
# AI First Index : 5000
# DI First Index : 10000
# AO First Index : 15000
# DO First Index : 20000
#
```

The following sections contain the tag descriptors for each of the four data types used by the protocol. The number of tags is determined by the number of entries under each section. Each list is read until the #, [or space character is present in the first column of each list. The tag index used by the protocol is determined by the first index for the data type and the position of the tag in the list. For example, the first AI tag will be assigned the tag index 5000. The second AI tag will be assigned the tag index of 5001.

The following is a list of all the analog input tags.

[AI DATA]	PT-07533A	PT-063161A
ET-06003A	TT-07533A	PT-063133A
ET-06004A	ET-07565A	PT-063134A
ET-06005A	PT-062964A	PT-063137A
ET-06006A	PT-062963A	TT-063137A
ET-06007A	PT-062962A	ZT-063138A
ET-06008A	PT-062961A	PT-063135A
ET-06002A	PT-062933A	PT-063136A
ET-06001A	PT-062934A	TT-063135A
ET-06009A	PT-062935A	TT-063136A
ET-06010A	PT-062936A	PT-063131A
ET-06011A	TT-062935A	TT-063131A
ET-06012A	TT-062936A	PT-063139A
ET-06013A	ZT-062938A	TT-063139A
ET-06014A	PT-062937A	CT-063140A
PT-07564A	TT-062937A	CT-063141A
PT-07563A	FT-062931A	CT-063142A
PT-07562A	ET-062965A	CT-063143A
PT-07561A	PT-063164A	CT-063144A
PT-07531A	PT-063163A	ET-063165A
TT-07531A	PT-063162A	

2.3.4 [AO Data]

The following is a list of all the analog output tags.

[AO DATA]
 HCV-062917A
 HCV-063117A

2.3.5 [DI Data]

The following is a list of all the digital input tags.

[DI DATA]	US-06007A	US-06001A
US-06006A	US-06004A	US-07568A
US-07566A	US-06003A	US-062968A
US-062966A	US-06005A	US-063168A
US-063166A	US-06002A	

2.3.6 [DO Data]

The following is a list of all the digital output tags.

[DO DATA]	XS-062953A	XV-063114A
XS-07557A	XS-062956A	XV-063111A
XS-062957A	XV-062908A	XV-063113A
XS-063157A	XV-062914A	XV-063106A
XS-07553A	XV-062911A	XV-063107A
XV-07520A	XV-062913A	XV-063116A
XV-07523A	XV-062912A	XV-063112A
XV-07517A	XV-062903A	XV-063109A
XV-07509A	XV-062904A	XV-063110A
XV-07510A	XV-062910A	XV-063118A
XV-07511A	XV-062909A	XV-063105A
XV-07512A	XV-062905A	XV-063103A
XV-07501A	SCSSV-062901A	XV-063104A
XV-07502A	SCSSV-062902A	XV-063115A
XV-07503A	XS-063153A	SCSSV-063101A
XV-07504A	XS-063156A	SCSSV-063102A
XS-07556A	XV-063108A	

2.4 Uploading and Downloading the Configuration File

ProSoft modules are shipped with a pre-loaded configuration file. In order to edit this file, you must transfer the file from the module to your PC. After editing, you must transfer the file back to the module.

This section describes these procedures.

Important: The illustrations of configuration/debug menus in this section are intended as a general guide, and may not exactly match the configuration/debug menus in your own module. For specific information about the configuration/debug menus in your module, refer to [The Configuration/Debug Menu](#) (page 45).

2.4.1 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

2.4.2 Required Software

In order to send and receive data over the serial port (COM port) on your computer to the module, you must use a communication program (terminal emulator).

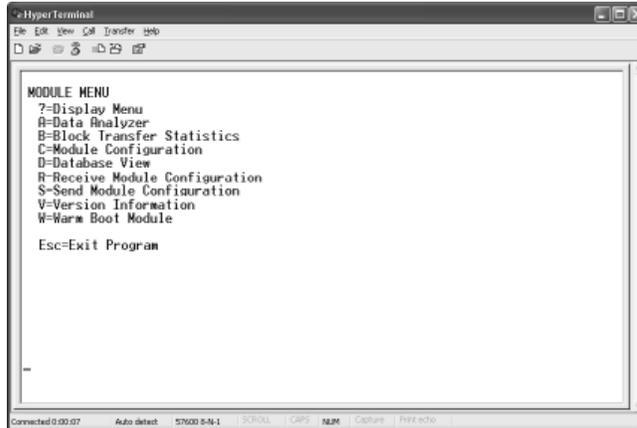
A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have been tested by ProSoft Technology.

DOS	ProComm, as well as several other terminal emulation programs
Windows 3.1	Terminal
Windows 95/98	HyperTerminal
Windows NT/2000/XP	HyperTerminal

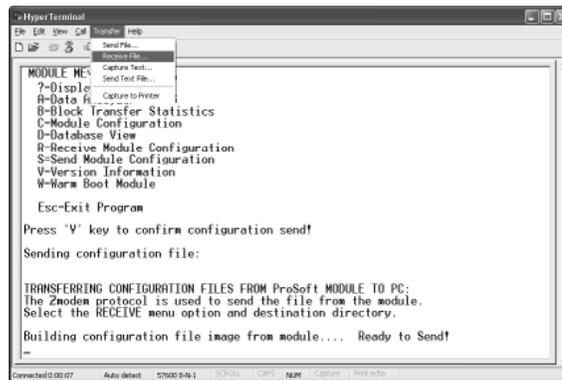
The module uses the Ymodem file transfer protocol to send (upload) and receive (download) configuration files from your module. If you use a communication program that is not on the list above, please be sure that it supports Ymodem file transfers.

2.4.3 Transferring the Configuration File to Your PC

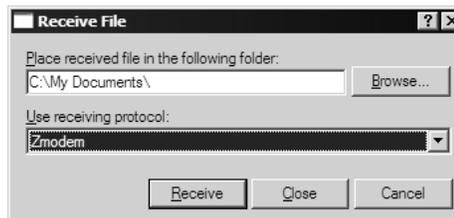
- 1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press [?] to display the main menu.



- 2 From the **Transfer** menu in HyperTerminal, select **Receive File**.

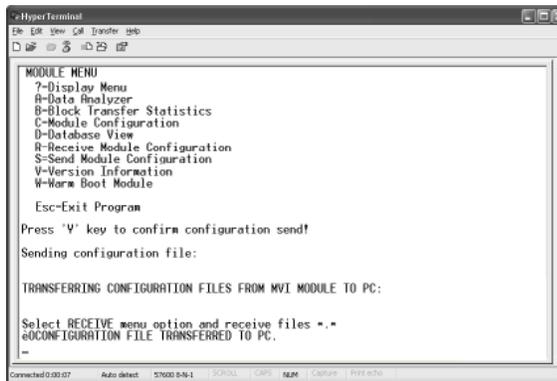


- 3 In the Receive File dialog box, browse to the location on your PC where the configuration file should be stored, and select Zmodem (or Zmodem with Crash Recovery) as the receiving protocol.



When you have completed your selections, click Close.

- 4 Press **[S]** (Send Module Configuration), and then press **[Y]** to confirm the transfer.



The file transfer will then begin automatically, using the protocol and location you specified in Step 3.

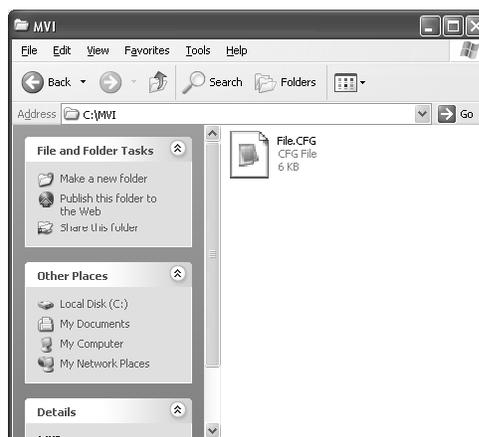
When the configuration file has been transferred to your PC, the dialog box will indicate that the transfer is complete.

```
Confirm Send Configuration File to Remote PC by pressing 'Y' key....
Sending configuration file:

TRANSFERRING CONFIGURATION FILES FROM MVI MODULE TO PC:

Select RECEIVE menu option and receive files *.*
eOCONFIGURATION FILE TRANSFERRED TO PC.
```

The configuration file is now on your PC at the location you specified.

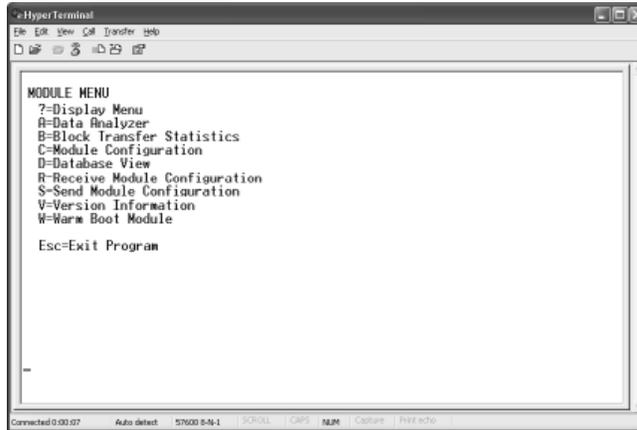


- 5 You can now open and edit the file in a text editor such as Notepad. When you have finished editing the file, save it and close Notepad.

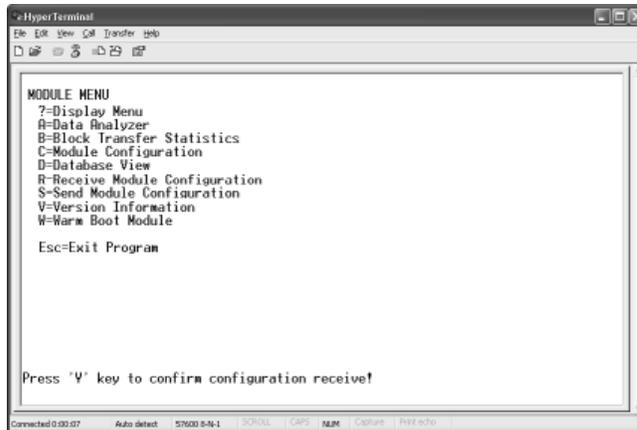
2.4.4 Transferring the Configuration File to the Module

Perform the following steps to transfer a configuration file from your PC to the module.

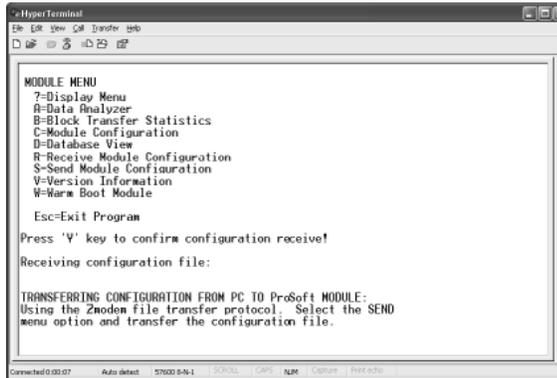
- 1 Connect your PC to the Configuration/Debug port of the module using a terminal program such as HyperTerminal. Press **[?]** to display the main menu.



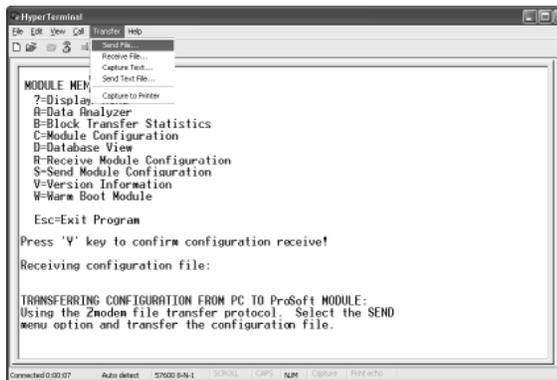
- 2 Press **[R]** (Receive Module Configuration). The message "Press Y key to confirm configuration receive!" is displayed at the bottom of the screen.



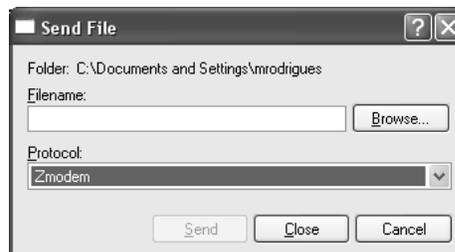
- Press [Y]. The screen now indicates that the PC is ready to send.



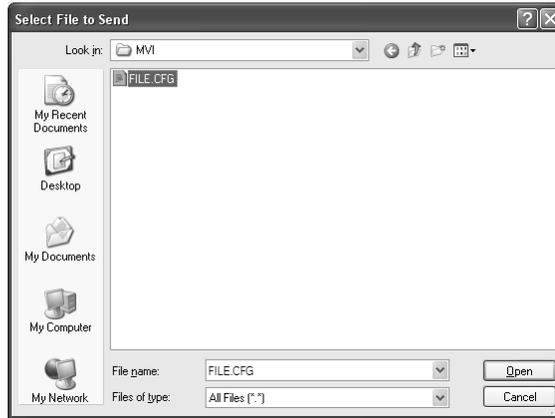
- From the **Transfer** menu in HyperTerminal, select **Send File**.



The Send File dialog appears.

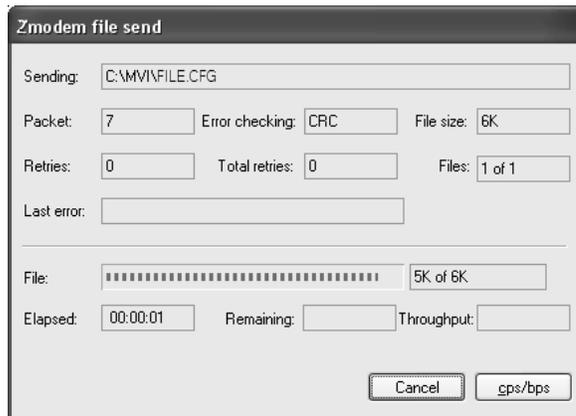


- 5 Use the Browse button to locate the configuration file your computer.

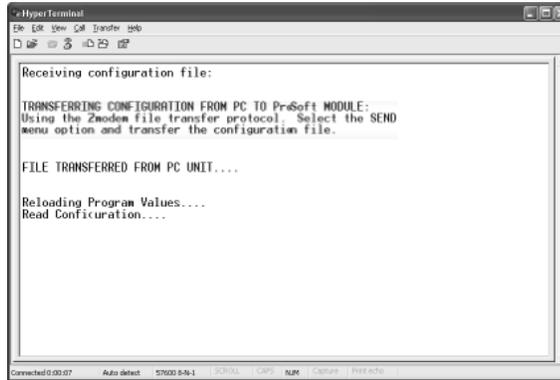


Note: This procedure assumes that you are uploading a newly edited configuration file from your PC to the module. However, configuration files are also available on the ProSoft Technology web site: www.prosoft-technology.com.

- 6 Select Zmodem as the protocol.
- 7 Click the Send button. This action opens the Zmodem File Send dialog box.



When the upload is complete, the screen indicates that the module has reloaded program values and displays information about the module.



8 Your module now contains the new configuration.

2.5 IP Address

In addition to the SMGC.CFG, the MVI56-SMGC module requires a second configuration file that identifies its Ethernet configuration. Without this configuration file, the module will not communicate properly on the network.

This file contains the Ethernet address information to be used by the module and may be transferred to and from the module from the **Network** command available on the debug port of the module. Please consult your network administrator for the correct settings for your network before placing this or any other Ethernet TCP/IP device upon your network.

Important: If the field "my_ip" does not exist, or if the wattcp.cfg file is corrupted or does not exist, the module will not function.

➤ To set the Module's IP Address

- 1 Locate the sample configuration files for your module.
- 2 Copy the configuration files and ladder to a location on your PC's hard drive. We recommend C:\temp.
- 3 After you move the files, right-click on each of the files, choose Properties, and clear the READ ONLY check box.
- 4 Start Notepad.exe, or any other editor that can save plain text files.
- 5 Open the file WATTCP.CFG. The following example shows the contents of a typical WATTCP.CFG file.

```
# ProSoft Technology
# Default private class 3 address
my_ip=192.168.0.100

# Default class 3 network mask
netmask=255.255.255.0
```

```
# The gateway I wish to use
gateway=192.168.0.1,192.168.0.0,255.255.255.0
```

- 6 Edit the file, using the IP addresses supplied by your network administrator.

Important: The module does not support DHCP (Dynamic Host Configuration Protocol) for obtaining an IP address from a server. This module must have its own static IP address that does not duplicate the IP address of any other device on the Ethernet network.

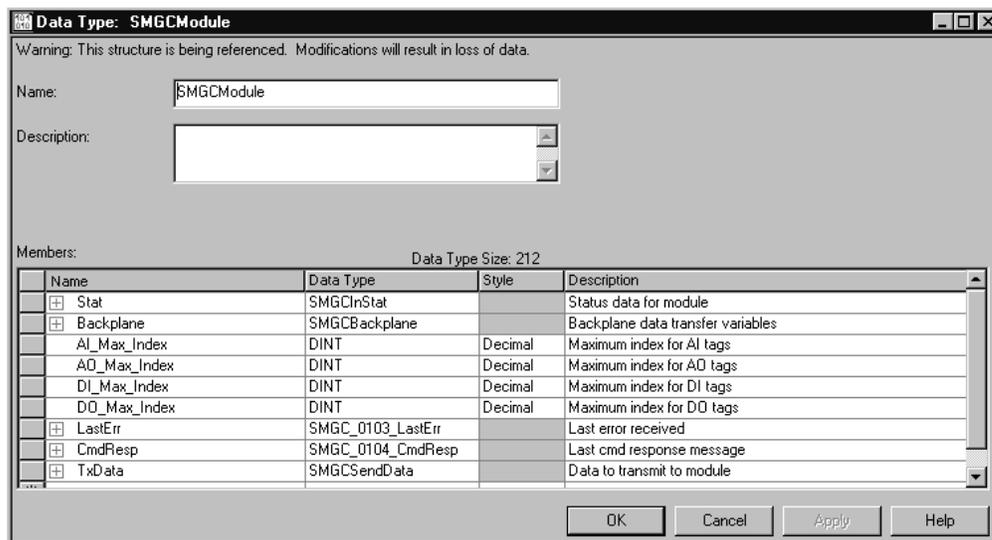
- 7 Save the file as WATTCP.CFG. You must now transfer the file to the module. Refer to [Transferring WATTCP.CFG to the module](#) (page 32, page 51) for the correct procedure.

2.6 Module Data

All data related to the MVI56-SMGC is stored in a user defined data types. An instance of the data types is required before the module can be used. This is accomplished by declaring a variable of the data type in the Controller Tags Edit Tags dialog box. Each of these object types is discussed in the following topics of the document.

2.6.1 Main Data Object (SMGCModule)

The SMGCModule object contains most of the data members and objects required by the MVI56-SMGC module. An instance of this object should be declared in the controller tags of the processor. The following figure displays this object:



The members of this object are discussed in the following topics.

Status Object (SMGCInStat)

This object views the status of the module. The **SMGCInStat** 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".

Data Type: SMGCInStat				
Name: SMGCInStat				
Description:				
Members: Data Type Size: 56 byte(s)				
Name	Data Type	Style	Description	
PassCnt	INT	Decimal	Program cycle counter for module	
Product	INT[2]	Hex	Product code for module (GSC)	
Rev	INT[2]	Hex	Revision level of module's code	
OP	INT[2]	Hex	Operating system version for module	
Run	INT[2]	Hex	Run number for module	
BlkErrs	SMGCBlkStat		Data block transfer statistics	
Client	SMGCClntStat[1]		Status for each client	

Within the SMGCInStat objects are objects containing the status information for the block transfer process and the client. Refer to [MVI56-SMGC Status Data Area](#) (page 79) for a complete listing of the data stored in this object.

Backplane Variables (SMGCBackplane)

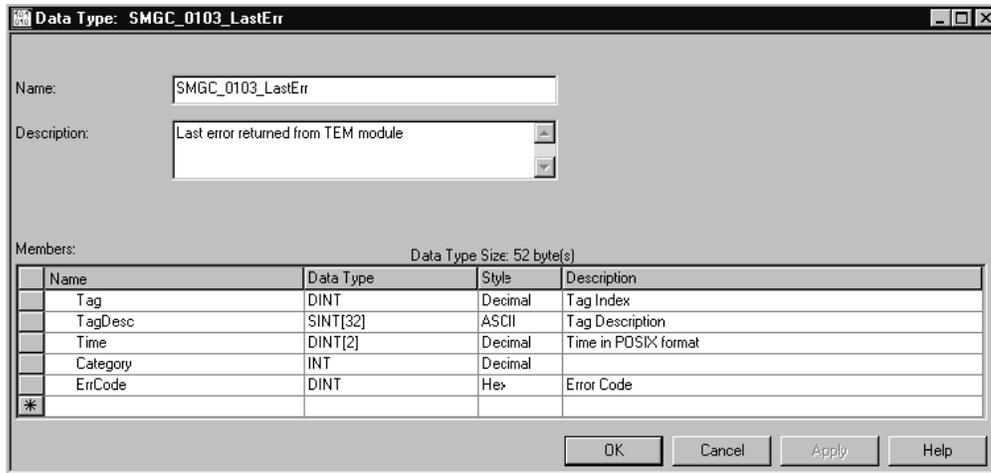
The SMGCBackplane object stores all the variables required for the data transfer operation between the module and the controller. The LastRead data member is used as the handshaking byte to indicate the arrival of new data from the module. The structure of this object is shown in the following illustration:

Data Type: SMGCBackplane				
Name: SMGCBackplane				
Description:				
Members: Data Type Size: 24 byte(s)				
Name	Data Type	Style	Description	
LastRead	INT	Decimal	Sequence number of last block read	
LastWriteCount	INT	Decimal	Last number of bytes written	
CurBlock	INT	Decimal	Sequence number for current block	
CurServer	INT	Decimal	Current server in read block	
RxLen	INT	Decimal	Length of message received	
DID_Rec	DINT	Decimal	Data block ID received	
TAG_ID	DINT	Decimal	Tag Index	
Offset	DINT	Decimal	Offset into current array	

The other members of the object are utilized in the ladder logic to assist in the data transfer operation.

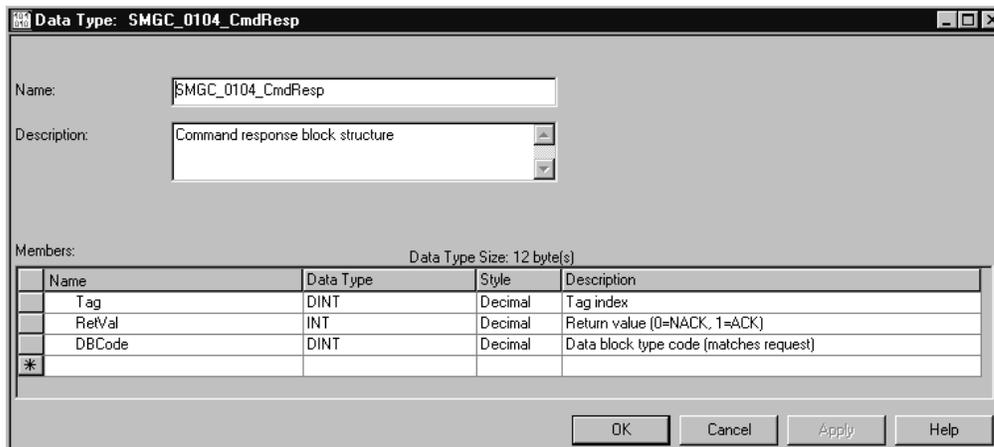
Last Error Data (SMGC_0103_LastErr)

The SMGC_0103_LastErr object stores the information received in the last error block received from the server. When this block is received, ladder logic copies the information into this object. The structure of the object is shown in the following screen:



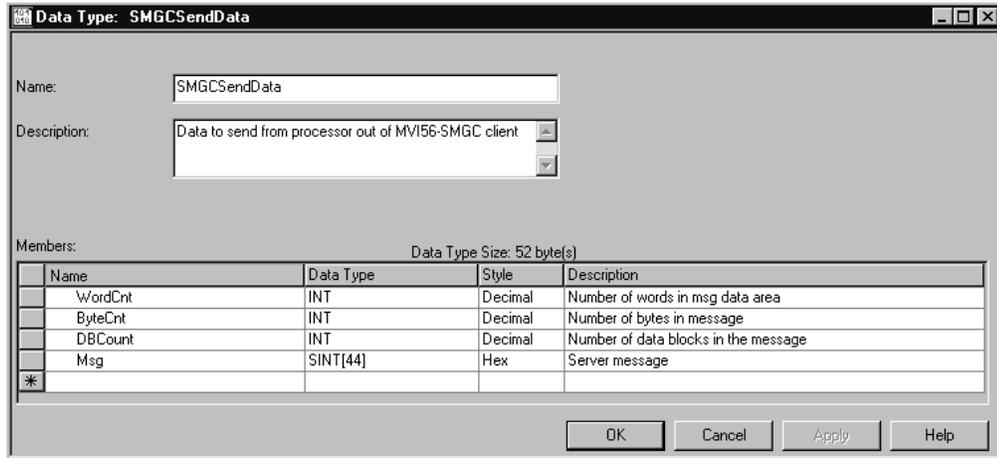
Command Response Data (SMGC_0104_CmdResp)

The SMGC_0104_CmdResp object stores the information received in the last command response block from the server. The structure of this block is shown in the following illustration:



Transmit Data (SMGCSendData)

The SMGCSendData object is utilized to build messages to send to the module that will be sent to the server. The structure of this object is such that the data can be directly copied into a write block built for the module. The structure of this object is shown in the following screen:

Other support objects

Other support objects are defined in the main data object for the module to hold the size information for each of the data types used in the SMGC protocol. The following table defines these members:

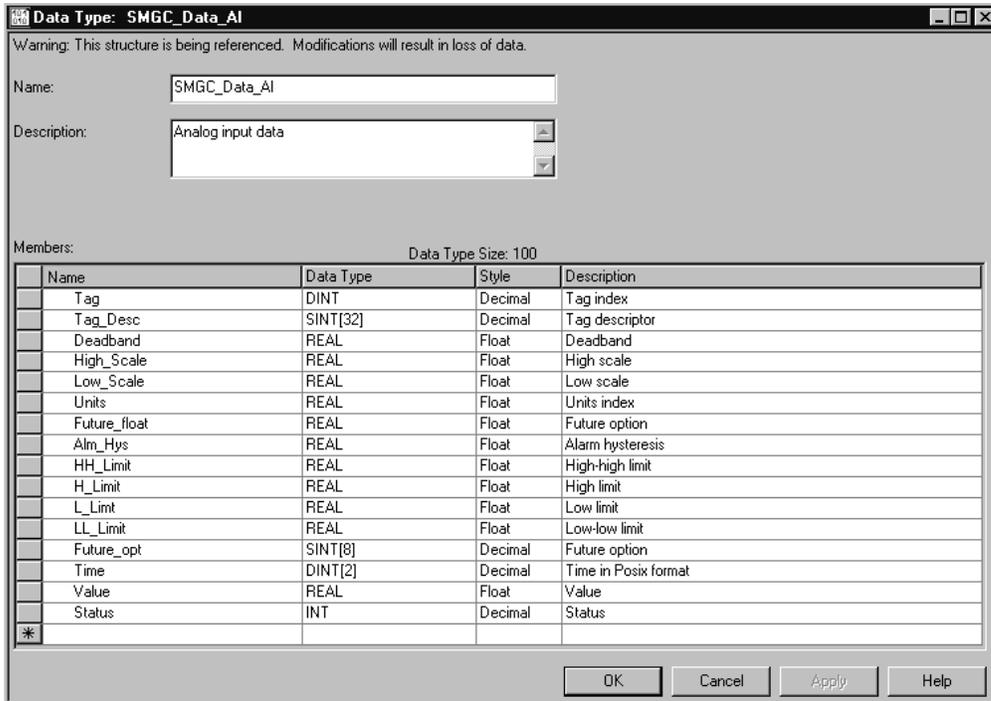
Object	Description
AI_Max_Index	Maximum index for AI array
AO_Max_Index	Maximum index for AO array
DI_Max_Index	Maximum index for DI array
DO_Max_Index	Maximum index for DO array

Tag index values are assigned to work with the member values set in the object and greatly simplify the ladder logic. For example, analog input points in the server are defined with the following tag index values: 5000 to 9999. For example, the analog input data can be set up as an array of 100 analog points (AI_Max_Index = 100) with tag indexes assigned 5000 to 5099. These data are stored in an array of type SMGC_Data_AI with array indices of [0] to [99]. The start tag index (fixed for the specific data type) can be used to determine the offset into an array based on the tag index received in a data block from the server and the maximum index can be used to check its validity.

2.6.2 Database Values

The data values received from the server (response data) and control data sent to the server (request data) are stored in arrays. There is a separate array set up for each of the four data types utilized by the SMGC protocol. Make certain the size of each array is as large as or larger than the value set for the associated xx_Max_Index member.

Data for analog inputs is stored in the following data type:



Data for the analog outputs is stored in the following data type:

Warning: This structure is being referenced. Modifications will result in loss of data.

Name:

Description:

Members: Data Type Size: 88 byte(s)

Name	Data Type	Style	Description
Tag	DINT	Decimal	Tag index
Tag_Desc	SINT[32]	Decimal	Tag descriptor
High_Scale	REAL	Float	High scale
Low_Scale	REAL	Float	Low scale
Units	REAL	Float	Units index
Future_op	SINT[8]	Decimal	Future option
Time	DINT[2]	Decimal	Time in Posix format
Step	DINT	Decimal	Readback step
Value	REAL	Float	Readback value
Status	INT	Decimal	Readback status
Check_Back	DINT	Decimal	Check back
Op_mode	INT	Decimal	Operation mode to transmit
Setpoint	REAL	Float	Setpoint value to transmit

Buttons: OK, Cancel, Apply, Help

Data for the digital inputs is stored in the following data type:

Warning: This structure is being referenced. Modifications will result in loss of data.

Name:

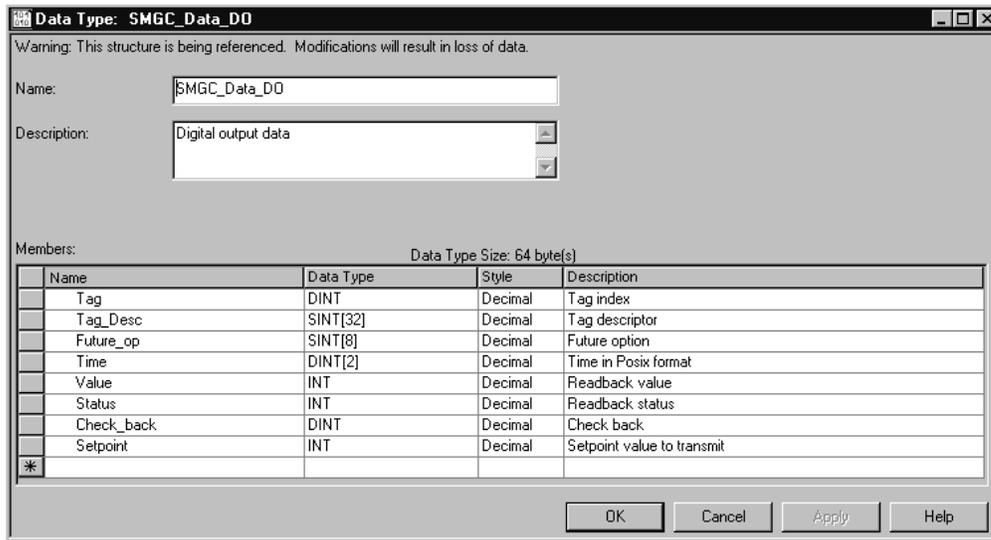
Description:

Members: Data Type Size: 56 byte(s)

Name	Data Type	Style	Description
Tag	DINT	Decimal	Tag index
Tag_Desc	SINT[32]	Decimal	Tag descriptor
Future_op	SINT[8]	Decimal	Future option
Time	DINT[2]	Decimal	Time in Posix format
Value	INT	Decimal	Value
Status	INT	Decimal	Status

Buttons: OK, Cancel, Apply, Help

Data for the digital outputs is stored in the following data types:



2.6.3 Helper Objects

Other data types can be defined in the ladder logic to help build request messages or to help parse response messages. Because the MVI56-SMGC application is written to interface to all SMGC protocol servers with any configuration, it is the responsibility of the application engineer to construct the other data types required for their application. The sample ladder logic contains examples of such structures. Refer to the SMGC protocol specification for a complete understanding of each data block that can be transmitted by the protocol.

3 Ladder Logic

Ladder logic is required for application of the MVI56-SMGC module. 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.

4 Diagnostics and Troubleshooting - MVI & PTQ

In This Chapter

- Reading Status Data From the Module 45
- LED Status Indicators 52

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

- Status data values are transferred from the module to the processor.
- Data contained in the module can be viewed through the Configuration/Debug port attached to a terminal emulator.
- LED status indicators on the front of the module provide information on the module's status.

4.1 Reading Status Data From the Module

The MVI56-SMGC module returns a status data set to the ControlLogix processor in each read block. This data is transferred to the ControlLogix processor continuously with each read block.

The Configuration/Debug port provides the following functionality:

- Full view of the module's configuration data
- View of the module's status data
- Version Information
- Control over the module (warm boot and cold boot)
- Facility to upload and download the module's configuration file

4.1.1 The Configuration/Debug Menu

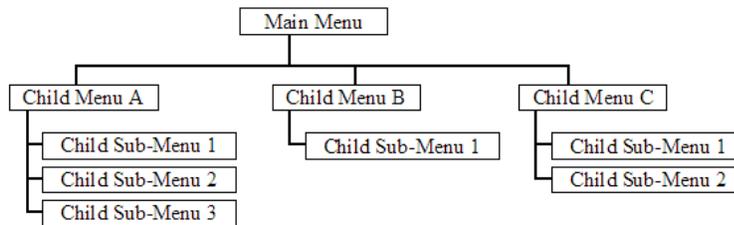
The Configuration and Debug menu for this module is arranged as a tree structure, with the Main Menu at the top of the tree, and one or more sub-menus for each menu command. The first menu you see when you connect to the module is the Main menu.

Because this is a text-based menu system, you enter commands by typing the command letter from your computer keyboard in the terminal application (for example, HyperTerminal). 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.

Navigation

All of the sub-menus for this module contain commands to redisplay the menu or return to the previous menu. You can always return from a sub-menu 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 you the menus available for this module, and briefly discusses the commands available to you.

Keystrokes

The keyboard commands on these menus are almost always non-case sensitive. You can enter most commands in lower case or capital 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][/]**.

Also, take care to distinguish capital letter **[I]** from lower case letter **[i]** (L) and number **[1]**; likewise for capital letter **[O]** and number **[0]**. Although these characters look nearly the same on the screen, they perform different actions on the module.

4.1.2 Required Hardware

You can connect directly from your computer's serial port to the serial port on the module to view configuration information, perform maintenance, and send (upload) or receive (download) configuration files.

ProSoft Technology recommends the following minimum hardware to connect your computer to the module:

- 80486 based processor (Pentium preferred)
- 1 megabyte of memory
- At least one UART hardware-based serial communications port available. USB-based virtual UART systems (USB to serial port adapters) often do not function reliably, especially during binary file transfers, such as when uploading/downloading configuration files or module firmware upgrades.
- A null modem serial cable.

4.1.3 Required Software

In order to send and receive data over the serial port (COM port) on your computer to the module, you must use a communication program (terminal emulator).

A simple communication program called HyperTerminal is pre-installed with recent versions of Microsoft Windows operating systems. If you are connecting from a machine running DOS, you must obtain and install a compatible communication program. The following table lists communication programs that have been tested by ProSoft Technology.

DOS	ProComm, as well as several other terminal emulation programs
Windows 3.1	Terminal
Windows 95/98	HyperTerminal
Windows NT/2000/XP	HyperTerminal

The module uses the Ymodem file transfer protocol to send (upload) and receive (download) configuration files from your module. If you use a communication program that is not on the list above, please be sure that it supports Ymodem file transfers.

4.1.4 Using the Configuration/Debug Port

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

- 1 Connect your computer to the module's port using a null modem cable.
- 2 Start the communication program on your computer and configure the communication parameters with the following settings:

Baud Rate	57,600
Parity	None
Data Bits	8
Stop Bits	1
Software Handshaking	None

- 3 Open the connection. When you are connected, press the [?] key on your keyboard. If the system is set up properly, you will see a menu with the module name followed by a list of letters and the commands associated with them.

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

- 1 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.
- 2 Verify that RSLinx is not controlling the COM port. Refer to [Disabling the RSLinx Driver for the Com Port on the PC](#) (page 77).
- 3 Verify that your communication software is using the correct settings for baud rate, parity and handshaking.
- 4 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, you can contact ProSoft Technology Technical Support for further assistance.

4.1.5 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 on your terminal screen:

```
MVI56-SMGC COMMUNICATION MODULE MENU
?=Display Menu
B=Block Transfer Statistics
C=Module Configuration
R=Transfer Configuration from PC to MVI Unit
S=Transfer Configuration from MVI Unit to PC
U=Version Information
Status: 1=Client 0
Cfg : 6=Client 0 7=AI Tags 8=AO Tags 9=DI Tags 0=DO Tags
@=Network Menu      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 failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. 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.

Viewing Module Configuration

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

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

Transferring the Configuration File from PC to MVI56 module

Press **[R]** to send (upload) the configuration file from your PC to the module and store the file on the module's Compact Flash Disk.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, the module will restart the program and load the new configuration information. Review the new configuration using menu commands **[6]** and **[0]** to be certain the module is configured correctly.

Transferring the Configuration File from MVI56 module to PC

Press **[S]** to receive (download) the configuration file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully downloaded, you can open and edit the file to change the module's configuration.

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.

Viewing Port Communication Status

Press **[1]** from the Main Menu to view the port communication status for the application port.

Use this command to view communication status and statistics for the selected port. This information can be informative when trouble-shooting communication problems.

Viewing Port Configuration

Press **[6]** from the Main Menu to view configuration information for the application port.

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

7, 8, 9, 0 = xx Tags

These options display the tags configured for the associated data points. Select the '?' key to display the menu associated with the option. After selecting the 'S' option, the following will be displayed:

```
01 TAG LIST (0 to 19)
ARRAY
INDEX TAG DESCRIPTION TAG INDEX
0 ET-06003A 5000
1 ET-06004A 5001
2 ET-06005A 5002
3 ET-06006A 5003
4 ET-06007A 5004
5 ET-06008A 5005
6 ET-06002A 5006
7 ET-06001A 5007
8 ET-06009A 5008
9 ET-06010A 5009
10 ET-06011A 5010
11 ET-06012A 5011
12 ET-06013A 5012
13 ET-06014A 5013
14 PT-07564A 5014
15 PT-07563A 5015
16 PT-07562A 5016
17 PT-07561A 5017
18 PT-07531A 5018
19 TT-07531A 5019
```

Opening the Network Menu

Press [@] to open the network menu. The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP, gateway and other network specification information. You can find more information about the commands on this menu in the Network Menu (page 51) section.

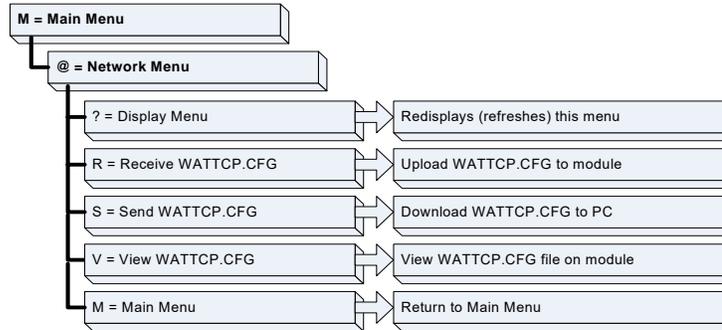
Exiting the Program

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 failures. Only use these commands if you are specifically directed to do so by ProSoft Technology Technical Support staff. Some of these command keys are not listed on the menu, but are active nevertheless. Please be careful when pressing keys so that you do not accidentally execute an unwanted command.

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 ROM to configure the module.

4.1.6 Network Menu

The network menu allows you to send, receive and view the WATTCP.CFG file that contains the IP and gateway addresses, and other network specification information.



Transferring WATTCP.CFG to the module

Press **[R]** to transfer a new WATTCP.CFG file from the PC to the module. Use this command to change the network configuration for the module (for example, the module's IP address).

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

Transferring WATTCP.CFG to the PC

Press **[S]** to transfer the WATTCP.CFG file from the module to your PC.

Press **[Y]** to confirm the file transfer, and then follow the instructions on the terminal screen to complete the file transfer process.

After the file has been successfully transferred, you can open and edit the file to change the module's network configuration.

Viewing the WATTCP.CFG file on the module

Press **[V]** to view the module's WATTCP.CFG file. Use this command to confirm the module's current network settings.

```

Network Menu Selected
WATTCP.CFG FILE:
# ProLinx Communication Gateways, Inc.
# Default private class 3 address
# my_ip=192.168.0.135
# Default class 3 network mask
# netmask=255.255.255.0
# The gateway I wish to use
# gateway=192.168.0.1
#Parameters used by the ProLinx Communication Gateways, Inc. module
#Local_Domain_Name=mycompany.com
#Password=PASSWORD
  
```

Returning to the Main Menu

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

4.2 LED Status Indicators

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

ProSoft Module	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	Port not used
		Off	Port not used
P2	Green	On	Port not used
		Off	Port not used
APP Status	Amber	Off	The MVI56-SMGC is working normally.
		On	The MVI56-SMGC module program has recognized a communication error.
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 the factory, as this is not a user serviceable item.

4.2.1 Ethernet LED Indicators

LED	State	Description
Data	Off	No activity on the port.
	Green Flash	The port is either actively transmitting or receiving data.
Link	Off	No connection to hub or network is detected.
	Green Solid	Connected to hub or network correctly. This is the normal operating state.

4.2.2 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 the configuration data being transferred to the module from the ControlLogix processor.

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 Support.

4.2.3 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. Verify that the slot in the rack configuration has been set up correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. Verify that all modules in the rack are configured in the ladder logic.

Module Errors

Problem Description	Steps to take
BP ACT LED remains off or blinks slowly	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: <ul style="list-style-type: none"> ▪ The processor is in Run mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write block data transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is configured in the processor.
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 the card in the rack, and then restore power to the rack.

5 Reference

In This Chapter

- Product Specifications..... 55
- Functional Overview..... 57
- Cable Connections..... 75
- MVI56-SMGC Status Data Area..... 79
- MVI56-SMGC Server Communication Module Configuration 81

5.1 Product Specifications

The MVI56 SM General Communication Protocol Module allows ControlLogix I/O compatible processors to interface easily with SM General Communication Protocol devices over an Ethernet network.

5.1.1 Features and Benefits

The MVI56-SMGC module interfaces an Ethernet communication device using the SM General Communication Protocol with the ControlLogix processor. A single client is present in the module to interface with a single server. The client permits both the reception and transmission of data between the ControlLogix processor and attached device.

5.1.2 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

5.1.3 Hardware Specifications

Specification	Description
Backplane Current Load	800 mA @ 5 V DC; 3mA @ 24V DC
Operating Temperature	0 to 60°C (32 to 140°F)
Storage Temperature	-40 to 85°C (-40 to 185°F)
Shock	30g Operational 50g non-operational Vibration: 5 g from 10 to 150 Hz
Relative Humidity	5% to 95% (non-condensing)
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 115,200 baud, depending on protocol 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 500V Optical isolation from backplane
Shipped with Unit	RJ45 to DB-9M cables for each port 6-foot RS-232 configuration cable

5.1.4 Functional Specifications

- One client to initiate a connection to a server and to receive and/or transmit data. This client will also set the tag indexes for all data points to be monitored and controlled based upon the user configuration
- The module makes the TCP/IP Connection automatically to the remote server
- Data exchange between the ControlLogix processor and the MVI module is optimized for use on a Control Net network
- 10/100 Base-T Ethernet compatible interface
- Configurable parameters for the client include:
 - Server IP address
 - Service Port number
 - Connection timeout
 - Tag lists for each data type

5.2 Functional Overview

This section provides an overview of how the MVI56-SMGC module transfers data using the SMG protocol. You should understand the important concepts in this chapter before you begin installing and configuring the module.

5.2.1 General Concepts

The following discussion explains several concepts that are important for understanding the operation of the MVI56-SMGC module.

Module Power Up

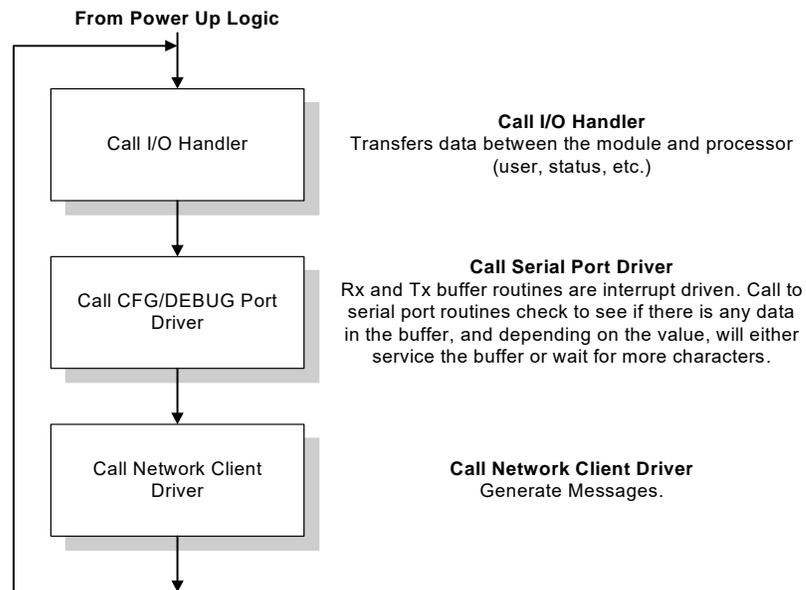
On power up, the module begins performing the following logical functions:

- 1 Initialize hardware components
 - Initialize ControlLogix backplane driver
 - Test and clear all RAM
 - Initialize the serial communication ports
- 2 Initialize the TCP/IP stack and Ethernet interface
- 3 Read configuration for module from SMGC.CFG file on Compact Flash Disk
- 4 Initialize the client
- 5 Set up the serial communication interface for the debug/configuration port

After the module has received the configuration, the module will attempt to make a connection to the specified server. After the connection is established, the module will transfer all the tag indexes to the server and obtain the configuration information. When this process is complete, the module will begin receiving and transmitting messages. Messages transmitted by the module are those constructed in ladder logic and passed to the module. Messages received by the module from the server are passed to the ladder logic for parsing.

Main Logic Loop

Upon completing the power up configuration process, the module enters an infinite loop that performs the functions shown in the following diagram.



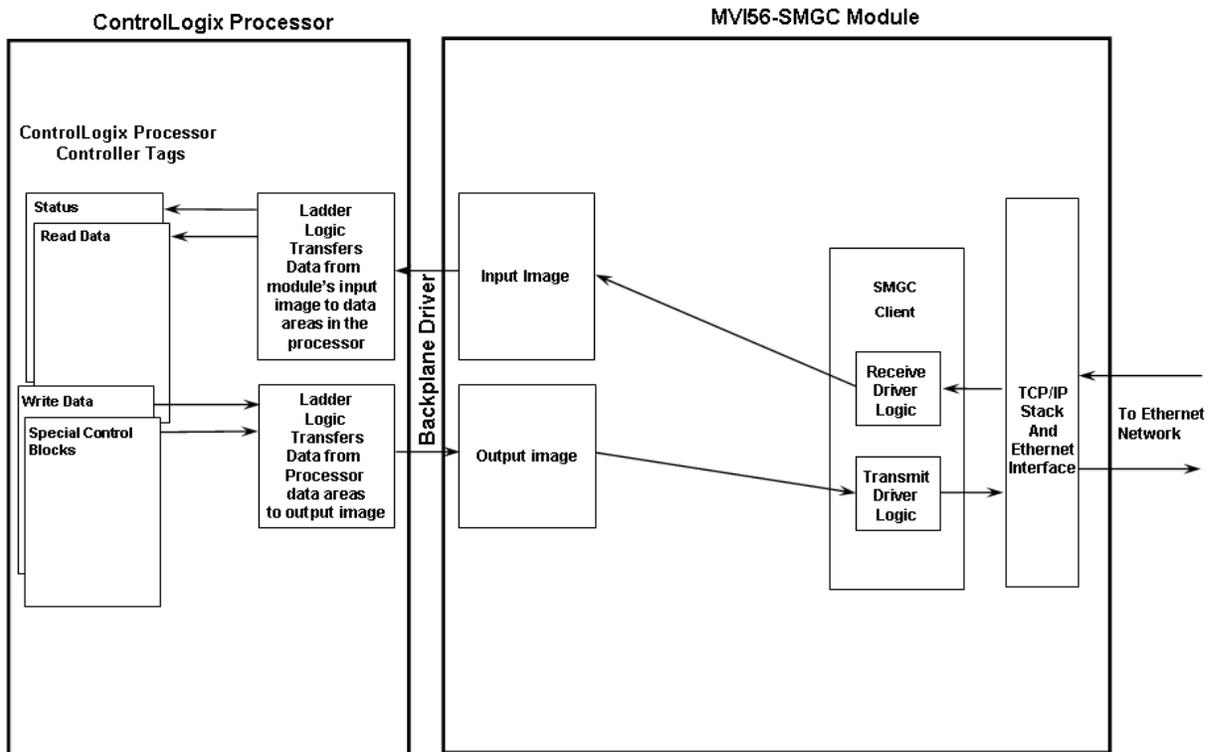
Backplane Data Transfer

The MVI56-SMGC module communicates directly over the ControlLogix backplane. Data travels between the module and the ControlLogix processor across the backplane using the module's input and output images. The update frequency of the data is determined by the 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.

Data received by the module is placed in the module's input image. This data is processed by the ladder logic in the ControlLogix processor. The input image for the module is set to 164 bytes. This data area permits fast throughput between the module and the processor and permits space for the maximum response message.

The processor inserts data in the module's output image to transfer to the module. The module's program extracts the data and transmits the data out to the Ethernet network. Each message is directed to the client that is connected to the remote server. The output image for the module is set to 56 bytes. This data area permits fast throughput of data from the processor to the module and is large enough to hold the largest request message.

The following illustration shows the data transfer method used to move data between the ControlLogix processor, the MVI56-SMGC module, and the Ethernet network.



As shown in the previous diagram, 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 defined in the controller tags. The user is responsible for handling and interpreting all data received on the application ports and transferred to the input image. Additionally, the user is responsible for constructing messages to be transferred out of the module by building the messages in the output image of the module

Normal Data Transfer

Normal data transfer includes the transferring of data received by or to be transmitted from the module and the status data. These data are transferred through read (input image) and write (output image) blocks. Refer to Module Configuration for a description of the data objects used with the blocks and the ladder logic required. The following topics discuss the structure and function of each block.

Read Block

These blocks of data transfer information from the module to the ControlLogix processor. The structure of the input image used to transfer this data is shown in the following table:

	Word Offset	Description
	0	Client Number for data received. If the word contains a value of -1, no receive data is present. This version of the software supports a single client. The module will only return of value of 0 when data is ready for this client.
Received Data	1	Number of characters (0 to 102) in the data block being passed to the processor contained in the data area (2 to 52). In order to keep the ladder logic processing simple, the module will only pass a single data block to the processor in each transaction.
	2 to 52	102 bytes of data area to hold the received data block.
Module Status	53 to 79	This data area contains the status data block. Each block transferred to the processor contains this set of information.
	80	Number of characters processed from last write block or error code.
	81	Block Sequence Number (Bumped each scan by module)

The Block Sequence Number (word 81) is an index value used to signal to the ControlLogix processor that a new block is ready for processing. The ladder logic must recognize a change in this value and process the data encapsulated in the input image. If data is available, a block containing the data received and the complete status data area is passed to the processor. The value at word 0 in the block determines if data from the client is present in the block. If word 0 is set to a value of -1, no data has been received on the connected client that must be transferred to the processor. If word 0 contains a value of 0, the read block contains a single data block received by the client. The following table lists the relationship between the values in word zero and their meaning:

Value	Description
-1	No receive data in block
0	Client 0 data in block

Word 1 of the input image determines the number of bytes in the data area of the block. This word only has significance if word 0 of the block is set to zero. This word will have a value less than or equal to 102, the maximum number of bytes that can be contained in a single data block. The data set is located in the block starting at word offset 2. The ladder logic must handle data received by the module. The structure of each block received by the client is well defined in the SMGC protocol specification. If the client receives more than one data block, it will send each block individually to the processor. This simplifies the ladder logic, as it only needs to handle a single data block type and tag index for each read block. The module will send the processor all the data blocks received until the last one is sent.

The module status data begins at word offset 53. A full discussion of this data set can be found in the **Module Configuration** section. The ladder logic should use the state value of the client to determine if an open connection (state value of 1) is available. Messages can only be sent from the ladder logic when the connection is open.

Word 80 of the message informs the ladder logic of the number of bytes sent in the last write block that was processed or if an error occurred when processing the write block. Ladder logic can check to make sure all bytes sent to the module are processed. If not, data may be lost. The following table lists the values returned in this word:

Value	Definition
-6	Client is not connected to server
-5	Invalid block count set in request
-4	Invalid Data Block ID specified in request
-3	Write overflow
-2	Write request >400 bytes
-1	Client busy with previous write request
0	No data requested to be written
>0	Number of characters written

The last word of the input image (word 81) is the block sequence number. This word's value is changed each time the module has processed the previous output image or is sending new information to the processor. This value should be copied to the first word of the output image (write block sequence number). Using this sequence number, the arrival of new data can easily be recognized by both the processor and the module.

Write Block

These blocks of data transfer information from the ControlLogix processor to the module. The structure of the output image used to transfer this data is shown in the following table.

	Word Offset	Description
	0	Block Sequence Number (Read block number as set by module)
Transmit Data	1	Client Number for data to transmit. If the word contains a value of -1, no transmit data is present. This version of the software only supports a single client. This word will be set to zero when data is ready for this client.
	2	Total number of characters contained in the data area (4 to 25) to be used to build data blocks. Each write block can transmit up to 44 bytes of data. This parameter has a valid range of 0 to 44. If the parameter is set to zero, no data will be transmitted.
	3	Number of data blocks contained in this write block. This parameter specifies the number of individual data blocks present in this block. Each of these blocks will be parsed and built for transmission to the TEM server. This parameter has a range of 0 to the maximum number of data blocks that can be built in the 44 character data area (words 4 to 25). For example, the user can send this parameter to 1 when sending SetTagIndex data blocks (44 bytes/block) or to 3 when requesting Signature data blocks (14 bytes/block).
	4 to 25	Data to transmit on specified server or client.
	26	Client[0] Control Word
	27	Module Control Word contains a value of 0, -1 or -2.

The Block Sequence Number is that received on the last read block transfer through the input image on the module. The ladder logic should copy this value from word 81 of the input image to word 0 of output image in the ladder logic. This is the last operation performed when constructing the write block. The module's program will trigger the process write block function when a new value is recognized in word 0 of the output image.

Word 1 of the block defines if the block contains data to write to the Ethernet network. If the word is set to a value of -1, there is no data in the message. If the word is set to 0, the block contains one or more data blocks to send out the client. The following table lists the relationship between word 1 values and their meaning:

Value	Description
-1	No transmit data in block
0	Client 0 data in block

Word 2 of the block defines the length of all data blocks in the write block. Up to 44 bytes of data can be sent in each block starting at word offset 4 in the block. It is the responsibility of the ladder logic to set word 2 to the correct message length and to populate the data area (words 4 to 25) with the correct data.

Word 3 of the block defines the number of SMGC protocol data blocks contained in the write block. For simple application of the module, this word should be set to 1 and the block should only contain data for a single data block. If more complexity is required, the ladder logic can be written to assemble several data blocks in a message (up to 44 bytes) and set word 3 to the number of data blocks contained in the message.

Word 26 closes the TCP/IP connection on the client. The following table lists the values recognized by the module for these words:

Code	Definition
0	No operation to perform
1	Close the client socket
2	Abort the client socket

Word 27 is utilized to control the module. The module as displayed in the following table recognizes two commands:

Code	Definition
0	No operation to perform
-1	Warm boot module
-2	Cold boot module

The module will perform the action specified in the command control word.

Network Client Driver

In order for data to be transferred between the module and another device, a TCP/IP connection must be made between the remote server and the client in the module. When no TCP/IP connection is established between the client and the remote server, the module will constantly attempt to make the connection. After the connection is established, either device can send and receive data.

The IP and service port set in the connection request must be valid for the connection process to succeed and are configured in the SMGC.CFG file developed by the user. If either of these two values is set incorrectly, the module will not be able to communicate.

The client has a state value in the status data area. This value is utilized by the ladder logic to determine if a connection is present on the client. The following table defines the state status values used by each client:

State Value	Definition
-1	Client is preparing to open a connection with the server.
0	The client is waiting to establish the connection with the server.
1	The client has established a connection with the server and can send and receive data.
10	The client is sending the AI tagdata to the server.
11	The client is waiting for the response to the tag index set request.
20	The client is sending the AO tag data to the server.
21	The client is waiting for the response to the tag index set request.
30	The client is sending the DI tagdata to the server.

State Value	Definition
31	The client is waiting for the response to the tag index set request.
40	The client is sending the DO tag data to the server.
41	The client is waiting for the response to the tag index set request.
50	The client is performing the configuration data request of the server to obtain all the configuration data.
1000	The client has initiated a close operation on the connection.
1001	The client is waiting for the close on the connection to complete.
1002	The client is issuing an abort socket is forced closed. (reset) on the connection. The
2000	Delay state after a failure. In this state, the module will delay for 10 seconds before trying to connect to the server.

Ladder logic should only direct messages to the client when it has a state status value of 1. The module will ignore all messages sent to the client with any other state value.

Ladder logic can request the client to close or abort a connection at any time. This is done by setting the client control word (26) in the output image to a specified value. The following table describes the values recognized by the module:

Code	Definition
0	No operation to perform
1	Close the client socket
2	Abort the client socket

In addition to the state value, the module contains a connection value that is also returned in the status data area. The table following lists the definition of each value:

State Value	Definition
-3	Server closed connection for client or server is not available.
-2	Unable to open connection with specified server (check IP and service port values).
-1	Unable to open connection with specified server because of an invalid IP address.
0	The client is idle and not connected.
1	The client is set to connect to the server and waiting for the connection to establish.
2	The client is connected to the server and can transfer data.
3	The connection is being closed by the client.

If a TCP/IP connection cannot be established with the selected remote server, use the value returned for this status value to determine the source of the error.

After the connection with the server is established and the tag index and configuration information are transferred, the client and server can freely send messages. The SMGC protocol supports the messages listed in the following table:

Blk ID	Length	Description	TX	RX
Command Data Blocks				
0x00000101	8	Configuration Request	X	
0x00000102	8	Ping Request/Response	X	X
0x00000103	54	Error Response		X
0x00000104	14	Command Response		X
0x00000105	26	Time Synchronization Request	X	
0x00000106	14	Signature Request	X	
0x00000107	70	Signature Response		X
0x00000108	44	Set Tag Index Request	X	
Analog Input Data Blocks				
0x00000201	102	AI Configuration		X*
0x00000202		AI To I/O Request		
0x00000203	22	AI From I/O Response		X*
Digital Input Data Blocks				
0x00000801	60	DI Configuration		X*
0x00000802		DI To I/O Request		
0x00000803	20	DI From I/O Response		X*
Analog Output Data Blocks				
0x00000401	82	AO Configuration		X*
0x00000402	14	AO To I/O Request	X	
0x00000403	30	AO From I/O Response		X*
Digital Output Data Blocks				
0x00001001	64	DO Configuration		X*
0x00001002	10	DO To I/O Request	X	
0x00001003	24	DO From I/O Response		X*

* - These blocks are transferred to the module on change without a request.

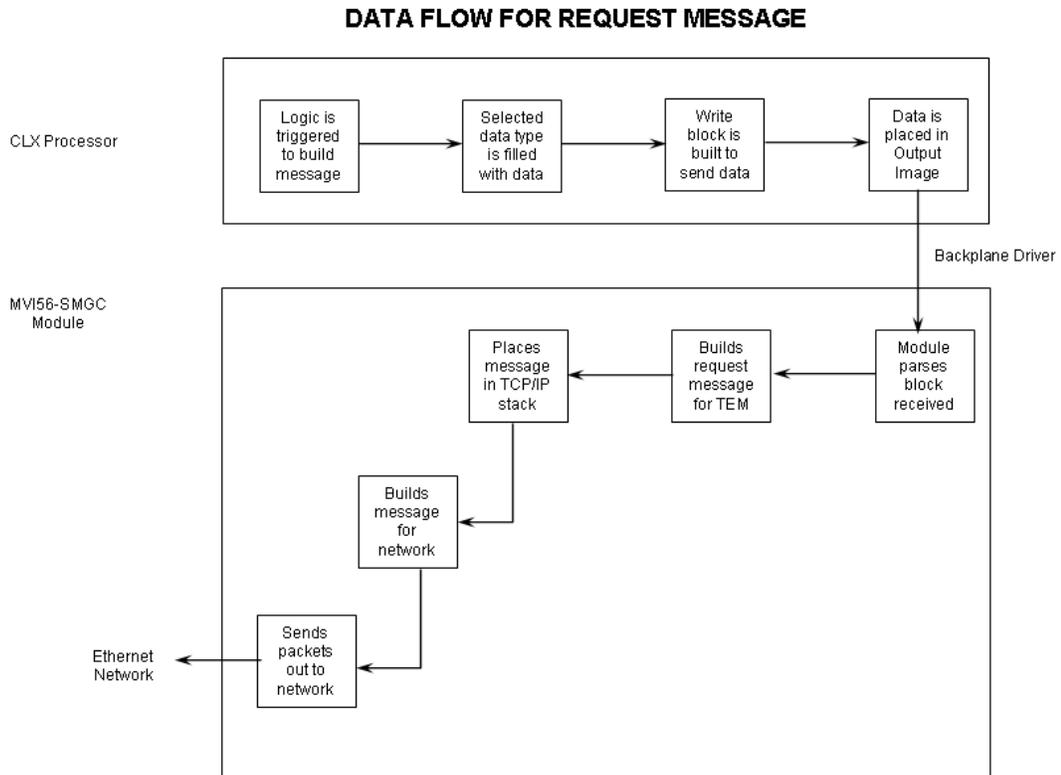
The blocks indicated in the TX column are those transmitted from the module to the server. Ladder logic must be constructed to build these data blocks. These messages are usually sent based on some trigger in the ladder logic and are usually not sent on a continuous basis. The ladder logic builds these data blocks in the output image of the module. The client driver in the module builds the SMGC protocol messages and sends it out on the Ethernet network.

Blocks indicated in the RX column are those received from the server. These blocks are either sent in a response to a request from the module or as a result of change in the server's database (unsolicited response) without a request message. Ladder logic must be developed to parse these blocks as they are sent from the module to the ControlLogix processor. The data received in these blocks is stored in the controller tags of the processor and are available for use by other processors.

Data Flow

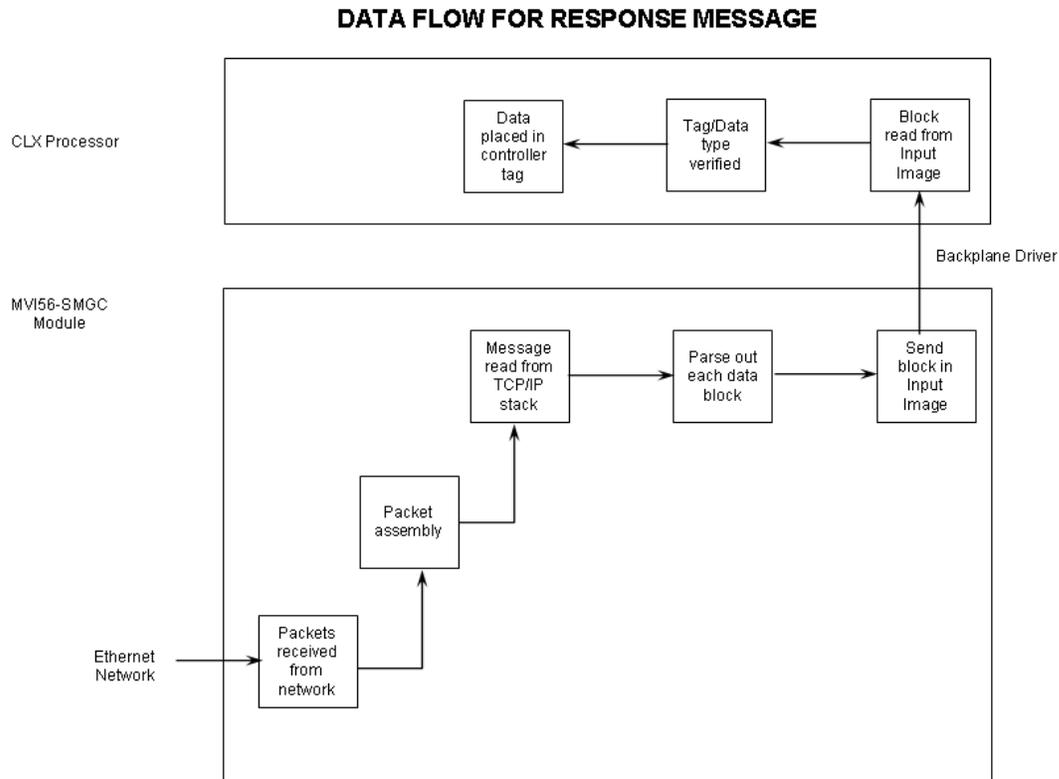
This section describes the flow of data blocks between the ControlLogix processor and the remote server. Ladder logic in the ControlLogix processor is responsible for generating requests and parsing messages received from the server. The parsed data from the received messages are placed in the controller tags of the processor.

The following illustration shows the data flow for a request message:



The ladder logic determines when a request message is to be built. This request can be triggered by an external or internal event in the processor. It is the responsibility of the application engineer to construct ladder logic for each request message to be sent to the remote server.

The following illustration shows the data flow for a response message:



The remote server unit generates the response message. It can be generated in response to a request message sent from the ladder logic or unsolicited because of a change in the database of the remote server. Ladder logic is responsible for parsing each data block received over the backplane and for placing the data in the correct controller tags.

Data Storage

All data utilized by the module is stored in the ControlLogix processor. Arrays of each data type are constructed as controller tags in the ladder logic. Each data type utilized by the module is discussed in the following topics. Refer to the SM General Communication Protocol Specification, April 6, 2002, for a complete discussion of the protocol.

Analog Input Data

Analog input data is stored in a controller tag with the following data structure:

Byte Count	Data Type	Description
4	DINT	Tag Index
32	SINT[32]	Tag Descriptor
4	REAL	Deadband
4	REAL	High Scale

Byte Count	Data Type	Description
4	REAL	Low Scale
4	REAL	Units Index
4	REAL	Future Option
4	REAL	Alarm Hysteresis
4	REAL	High-High Limit
4	REAL	High Limit
4	REAL	Low Limit
4	REAL	Low-Low Limit
8	SINT[8]	Future Option
8	DINT[2]	Time in POSIX Format
4	REAL	Out.Value
2	INT	Out.Status

This data is acquired from the server using data blocks 0x201 and 0x203. The format of the 0x201 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Analog Input Cfg Response data block code 0x00000201
4 to 5	4	DINT	Tag Index
6 to 21	32	SINT[32]	Tag Descriptor
22 to 23	4	REAL	Deadband
24 to 25	4	REAL	High Scale
26 to 27	4	REAL	Low Scale
28 to 29	4	REAL	Units Index
30 to 31	4	REAL	Future Option
32 to 33	4	REAL	Alarm Hysteresis
34 to 35	4	REAL	High-High Limit
36 to 37	4	REAL	High Limit
38 to 39	4	REAL	Low Limit
40 to 41	4	REAL	Low-Low Limit
42 to 45	8	SINT[8]	Future Option
46 to 49	8	DINT[2]	Time in POSIX format
50 to 51	4	REAL	Out.Value
52	2	INT	Out.Status

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when requested by the client (request data block 0x101) or when the associated data changes in the server.

The format of the 0x203 block is as follows (read block from the module):

Word Offset	Byte count	Data Type	Description
2 to 3	4	DINT	Analog Input Response data block code 0x00000203
4 to 5	4	DINT	Tag Index
6 to 9	8	DINT[2]	Time in POSIX format
10 to 11	4	REAL	Out.Value
12	2	INT	Out.Status

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when the associated data changes in the server.

Digital Input Data

Digital input data is stored in a controller tag with the following data structure:

Byte Count	Data Type	Description
4	DINT	Tag Index
32	SINT[32]	Tag Descriptor
8	SINT[8]	Future Option
8	DINT[2]	Time in POSIX Format
2	INT	OutD.Value
2	INT	OutD.Status

This data is acquired from the server using data blocks 0x801 and 0x803. The format of the 0x801 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Digital Input CfgResponse data block code 0x00000801
4 to 5	4	DINT	Tag Index
6 to 21	32	SINT[32]	Tag Descriptor
22 to 25	8	SINT[8]	Future Option
26 to 29	8	DINT[2]	Time in POSIX format
30	2	INT	OutD.Value
31	2	INT	OutD.Status

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when requested by the client (request data block 0x101) or when the associated data changes in the server.

The format of the 0x803 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Digital Input Response data block code 0x00000803
4 to 5	4	DINT	Tag Index

Word Offset	Byte Count	Data Type	Description
6 to 9	8	DINT[2]	Time in POSIX format
10	2	INT	OutD.Value
11	2	INT	OutD.Status

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when the associated data changes in the server.

Analog Output Data

Analog output data is stored in a controller tag with the following data structure:

Byte Count	Data Type	Description
4	DINT	Tag Index
32	SINT[32]	Tag Descriptor
4	REAL	High Scale
4	REAL	Low Scale
4	REAL	Units Index
8	SINT[8]	Future Option
8	DINT[2]	Time in POSIX Format
4	DINT	Readback Step
4	REAL	Readback Value
2	INT	Readback Status
4	DINT	Check Back
2	INT	Operation Mode
4	REAL	Setpoint

This data is acquired from the server using data blocks 0x401 and 0x403 and set by other devices or the ladder logic. The format of the 0x401 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	AnalogOutput Cfg Response data block code 0x00000401
4 to 5	4	DINT	Tag Index
6 to 21	32	SINT[32]	Tag Descriptor
22 to 23	4	REAL	High Scale
24 to 25	4	REAL	Low Scale
26 to 27	4	REAL	Units Index
28 to 31	8	SINT[8]	Future Option
32 to 35	8	DINT[2]	Time in POSIX format
36 to 37	4	DINT	Readback Step
38 to 39	4	REAL	Readback Value
40	2	INT	Readback Status
41 to 42	4	DINT	Check Back

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when requested by the client (request data block 0x101) or when the associated data changes in the server.

The format of the 0x403 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	AnalogOutput Response data block code 0x00000403
4 to 5	4	DINT	Tag Index
6 to 9	8	DINT[2]	Time in POSIX format
10 to 11	4	DINT	Readback Step
12 to 13	4	REAL	Readback Value
14	2	INT	Readback Status
15 to 16	4	DINT	Check Back

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when the associated data changes in the server.

The last two members of the controller tag data structure alter the value in the server and control the analog output point using a data block 0x402. These data are passed from the ladder logic to the module in a write block with the following format:

Word Offset	Byte Count	Data Type	Description
4 to 5	4	DINT	Analog Output Request data block code 0x00000402
6 to 7	4	DINT	Tag Index
8	2	INT	Operation Mode
9 to 10	4	REAL	Setpoint

Operation Mode:

0 = Set

1 = Setpoint

2 = Step (0=stop, >0=move open, <0=Move closed)

The ladder logic should be programmed to send this message when a new setpoint value is required for a specific tag in the server.

Digital Output Data

Digital output data is stored in a controller tag with the following data structure:

Byte Count	Data Type	Description
4	DINT	Tag Index
32	SINT[32]	Tag Descriptor
8	SINT[8]	Future Option

Byte Count	Data Type	Description
8	DINT[2]	Time in POSIX Format
2	INT	Readback Value
2	INT	Readback Status
4	DINT	Check Back
2	INT	Setpoint

This data is acquired from the server using data blocks 0x1001 and 0x1003 and set by other devices or the ladder logic. The format of the 0x1001 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Digital Output Cfg Response data block code 0x00001001
4 to 5	4	DINT	Tag Index
6 to 21	32	SINT[32]	Tag Descriptor
22 to 25	8	SINT[8]	Future Option
26 to 29	8	DINT[2]	Time in POSIX format
30	2	INT	Readback Value
31	2	INT	Readback Status
32 to 33	4	DINT	Check Back

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when requested by the client (request data block 0x101) or when the associated data changes in the server.

The format of the 0x1003 block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Digital Output Response data block code 0x00001003
4 to 5	4	DINT	Tag Index
6 to 9	8	DINT[2]	Time in POSIX format
10	2	INT	Readback Value
11	2	INT	Readback Status
12 to 13	4	DINT	Check Back

When the module passes one of these blocks to the ControlLogix processor, the ladder logic should use the Tag Index value to determine which array index to use for placing the data into the controller tag database. These blocks are passed to the module from the server when the associated data changes in the server.

The last member of the controller tag data structure alters the value in the server and control the digital output point using a data block 0x1002. These data are passed from the ladder logic to the module in a write block with the following format:

Word Offset	Byte Count	Data Type	Description
4 to 5	4	DINT	Digital Output Request data block code 0x00001002
6 to 7	4	DINT	Tag Index
8	2	INT	Setpoint

The ladder logic should be programmed to send this message when a new setpoint value is required for a specific tag in the server.

Command Data Blocks

Command data blocks are utilized by the SMGC protocol to perform specific functions or to indicate status of the messaging. For a complete discussion of each command data block, refer to the protocol specification. A discussion of each function is given in the following topics:

Configuration Request

This data block requests the configuration information from a single or all configured tags in the server. The format of the message to request configuration information is as follows (write block to the module):

Word Offset	Byte Count	Data Type	Description
4 to 5	4	DINT	Configuration data block code 0x00000101
6 to 7	4	DINT	Tag Index (0xFFFFFFFF=all tags)

If a specific tag is required, enter its tag index into the tag index member of the block. The module will automatically issue this request after all the tag indexes are set by the module. This process occurs each time the module makes a TCP/IP connection to the server.

Ping Request/Response

This data block is used to ping the server and have the server respond with the same message sent. This message can be sent to check that the server is still connected to the client and to keep the TCP/IP connection open. The format of the request message is as follows (write block to the module):

Word Offset	Byte Count	Data Type	Description
4 to 5	4	DINT	Ping data block code (0x00000102)
6 to 7	4	DINT	Tag Index (unused)

The server will respond with a block with the following format (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Ping data block code (0x00000102)
4 to 5	4	DINT	Tag Index (unused)

Error Response

Data blocks of this type are sent from the server to the module when an error is recognized. The format of this block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Error code data block 0x00000103
4 to 5	4	DINT	Tag Index
6 to 21	32	SINT[32]	Tag description
22 to 25	8	DINT[2]	Time in POSIX format
26	2	INT	Category
27 to 28	4	DINT	Error code value

If Tag Index = 0, general error and not tag related Else, Tag Index is equal to the tag related to the response error.

Category Codes: 1=Error, 2=Warning, 3=Information

Error Code List:

0xF0000001 → Unknown tag description in set tag index data block

(0x00000108) 0xF0000002 → Not initialized tag index in the xx Configuration data block (send set tag index)

0xF0000003 →

0xF0000004 → Tag index is already set for the object

0xF0000005 →

0xF0000006 →

0xF0000007 → Command on wrong object type

0xF0000008 →

0xF0000009 → Tag index already exists

0xF000000A → No operation before configuration data sent

0xF000000B → No operation before all configuration data sent

Command Response

Data blocks of this type are sent from the server to the module in response to a request. The format of this block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Command Response data block code 0x00000104
4 to 5	4	DINT	Tag Index
6	2	INT	Return Value
7 to 8	4	DINT	Data block type code matching this block

Return Value --> 0=NACK, 1=ACK

Time Synchronization Request

This data block is sent from the ladder logic to the server to synchronize the clocks of the units. The format of this block is as follows (write block to the module):

Word Offset	Byte Count	Data type	Description
4 to 5	4	DINT	Time synchronization data block code 0x00000105
6 to 7	4	DINT	TagIndex (unused)
8	2	INT	Time format code (1 or 2)
9 to 16	16		Time data in the specified format

Time Format: 1=POSIX, 2=Siemens

Word Offset	Byte Count	Data type	Description
POSIX Time Format (number of seconds since Jan, 1970):			
	seconds	4 bytes	unsigned long
	nanoseconds	4 bytes	unsigned long
Siemens Time Format:			
Byte #	Range	Name	
0	0 to 59	Second	
1		Decimal seconds (Not used)	
2	0 to 23	Hour	
3	0 to 59	Minutes	
4	1 to 31	Date	
5		Weekday (Not Used)	
6	0 to 99	Year	
7	1 to 12	Month	

Signature Request

This data block requests signature data from a sensor or actuator. The format of this block is as follows (write block to the module):

Word Offset	Byte Count	Data Type	Description
4 to 5	4	DINT	Request for signature data block code 0x00000106
6 to 7	4	DINT	Tag Index
8	2	INT	Operation Code
9	2	INT	Frequency
10	2	INT	Number of Samples

Operation Code:

0 = Open actuator signature

1 = Close actuator signature

2 = Sensor signature

Frequency: Number of milliseconds between samples for sensor signature requests

Number of Samples:

Number of samples to be taken for a sensor signature request.

Max number of samples is 400 if MAX_BUFFER is sufficient.

If Operation Code = 0 or 1, set to 120 for valve signature

Signature Response

This data block holds the information from the server in response to a signature request. The format of this block is as follows (read block from the module):

Word Offset	Byte Count	Data Type	Description
2 to 3	4	DINT	Signature Response data block code 0x00000107
4 to 5	4	DINT	Tag Index
6	2	INT	Operation Code
7 to 10	8	DINT[2]	Time in POSIX format
11	2	INT	Frequency
12	2	INT	Number of Samples

Word Offset	Byte Count	Data Type	Description
13	2	INT	Message Number
14	2	INT	Total Number of Sensors
15 to 16	4	DINT	Tag Index
17	2	INT	Units Index
18 to 33	32	SINT[32]	Tag Description
34 to 35	4	DINT	Value
36	2	INT	Status

This block will only hold the value and status for the first sample. All other samples received will not be passed to the processor.

Set Tag Index Request

This data block specifies the tag index value for a specific tag element of a selected data type in the server. Use of tag index values greatly reduces the bandwidth on the network required to transfer tag information. The format of this block is as follows (write block to the module):

Word Offset	Byte Count	Data Type	Description
4 to 5	4	DINT	Set Tag Index Request data block code 0x00000108
6 to 7	4	DINT	Tag Index
8 to 23	32	SINT[32]	Tag Descriptor
24 to 25	4	DINT	Configuration Data Block Type

Configuration Data Block Type List: AI = 0x00000201
AO = 0x00000401
DI = 0x00000801
DO = 0x00001001

The module will automatically issue this command for each tag configured by the user in the SMGC.CFG file for each data type. This process is required in order to set the tag index values in the server. The ladder logic and controller tags defined in the ControlLogix processor must be set up to match the tags defined in each data type.

5.3 Cable Connections

The MVI56-SMGC module has the following communication connections on the module:

- One Ethernet port (RJ45 connector)
- One RS-232 Configuration/Debug port (RJ45 connector)

5.3.1 Ethernet Connection

The MVI56-SMGC module has an RJ45 port located on the front of the module labeled "Ethernet", for use with the TCP/IP network. The module is connected to the Ethernet network using an Ethernet cable between the module's Ethernet port and an Ethernet switch or hub.

Note: Depending on hardware configuration, you may see more than one RJ45 port on the module. The Ethernet port is labeled "Ethernet".

Warning: The MVI56-SMGC module is NOT compatible with Power Over Ethernet (IEEE802.3af / IEEE802.3at) networks. Do NOT connect the module to Ethernet devices, hubs, switches or networks that supply AC or DC power over the Ethernet cable. Failure to observe this precaution may result in damage to hardware, or injury to personnel.

Important: The module requires a static (fixed) IP address that is not shared with any other device on the Ethernet network. Obtain a list of suitable IP addresses from your network administrator BEFORE configuring the Ethernet port on this module.

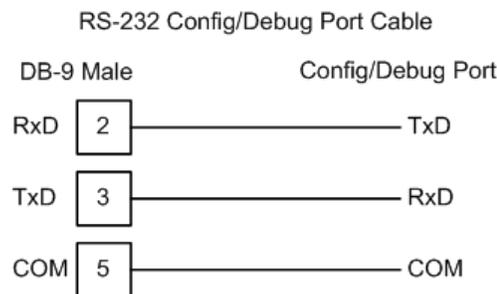
Ethernet Port Configuration: wattcp.cfg

The wattcp.cfg file must be set up properly in order to use a TCP/IP network connection. You can view the current network configuration using an ASCII terminal by selecting "@" (Network Menu) and "V" (View) options when connected to the Debug port.

```
# WATTCP.CFG FILE:  
# Prosoft Technology.  
my_ip=192.168.0.100  
# Default class 3 network mask  
netmask=255.255.255.0  
# The gateway I wish to use  
gateway=192.168.0.1
```

5.3.2 RS-232 Configuration/Debug Port

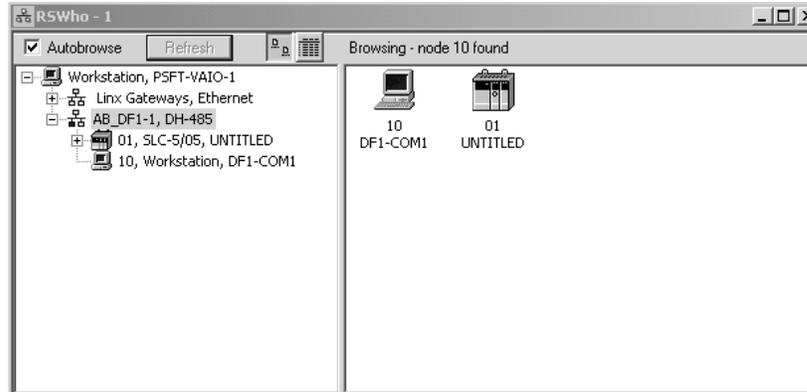
This port is physically an RJ45 connection. An RJ45 to DB-9 adapter cable is included with the module. This port permits a PC based terminal emulation program to view configuration and status data in the module and to control the module. The cable for communications on this port is shown in the following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



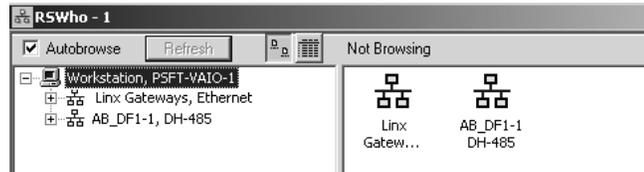
Disabling the RSLinx Driver for the Com Port on the PC

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 HyperTerminal or a similar 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:

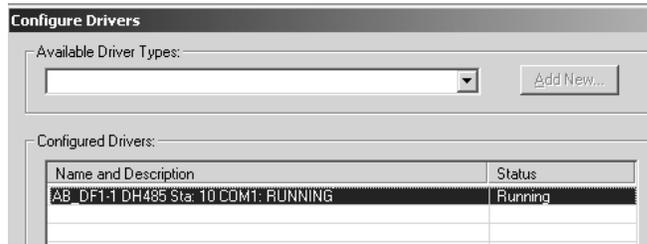


Branches are displayed or hidden by clicking on the  or the  icons.



- 4 When you have verified that the driver is not being browsed, go to **Communications>Configure Drivers**

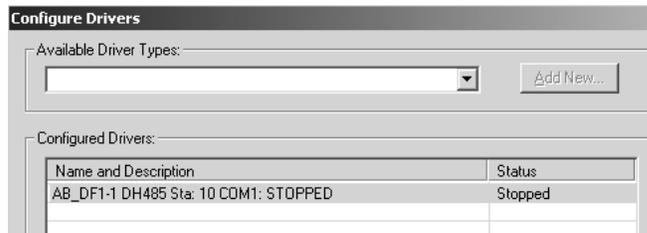
You may see something like this:



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" on the side of the window:



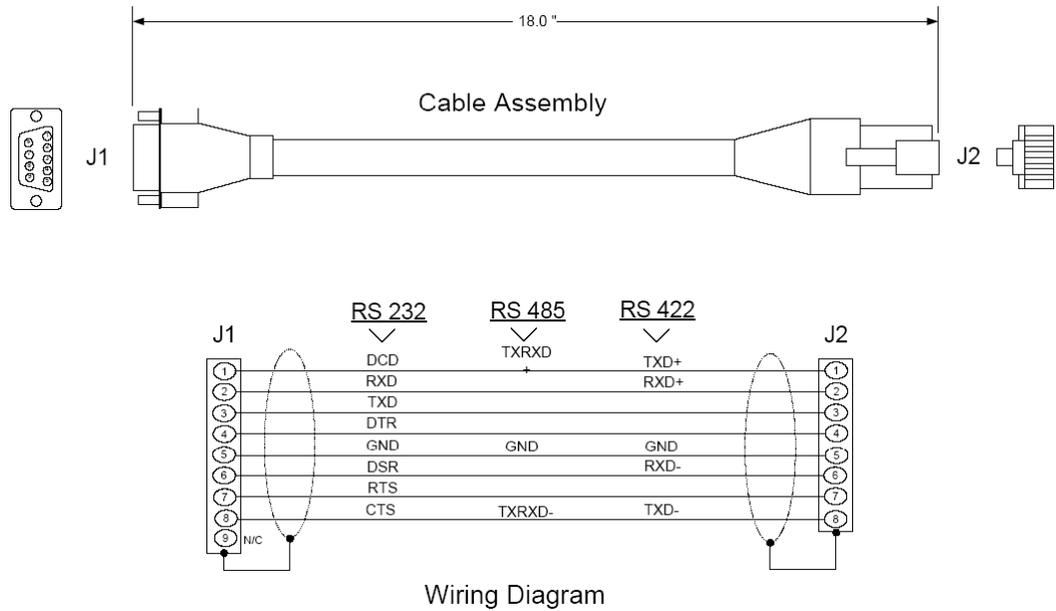
5 After you have stopped the driver you will see the following:



6 Upon seeing this, you may now use that com port to connect to the 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 not open, and you still cannot stop the driver, then reboot your PC.

5.3.3 DB9 to RJ45 Adaptor (Cable 14)



5.4 MVI56-SMGC Status Data Area

This section contains a listing of the data contained in the MVI56-SMGC status data object.

Object in SMGCInStat	Block Offset Start	Description
PassCnt	53	Program cycle counter
Product	54	Product name as ASCII string
Rev	56	Revision level as ASCII string
OP	58	Operating system level as ASCII string
Run	60	Run number as ASCII string
BlkErrs.Read	62	Number of blocks transferred from module to processor
BlkErrs.Write	63	Number of blocks transferred from processor to module
BlkErrs.Parse	64	Number of blocks parsed by module
BlkErrs.Err	65	Number of block errors in module
Client[0].Connect	66	Connection state of Client 0
Client[0].State	67	State machine value
Client[0].Open	68	Number of times socket open function called
Client[0].Estab	69	Number of times socket established
Client[0].Close	70	Number of times socket closed
Client[0].SyncErr	71	Total number of errors in trying to find the sync bytes for the start of a message.

Object in SMGCInStat	Block Offset Start	Description
Client[0].DBPCount	72	Total number of data blocks parsed.
Client[0].DBTCount	73	Total number of data blocks built.
Client[0].CfgWord	74	Configuration error word
Client[0].RxCount	75	This status value contains the total number of messages received by the client.
Client[0].RxOverflow	76	This status value contains the total number of messages received that exceed the specified buffer size for the client.
Client[0].TxCount	77	This status value contains the total number of messages transmitted by the client.
Client[0].TxOverflow	78	This status value contains the total number of transmit messages that exceeded the specified maximum buffer size for the client.
Client[0].LenErr	79	Total number of messages with a length setting larger than MAX_BUFFER.

The configuration error word has the following format:

Bit Position	Description
Bit 0	0x0001 Invalid server IP address specified.
Bit 1	0x0002 Service port zero assigned. This is not valid.
Bit 2	0x0004 A value less than 5000 is entered for the Connection Timeout parameter other than 0.
Bit 3	0x0008
Bit 4	0x0010
Bit 5	0x0020
Bit 6	0x0040
Bit 7	0x0080
Bit 8	0x0100
Bit 9	0x0200
Bit 10	0x0400
Bit 11	0x0800
Bit 12	0x1000
Bit 13	0x2000
Bit 14	0x4000
Bit 15	0x8000

5.5 MVI56-SMGC Server Communication Module Configuration

[Section]/Item	Value	Range	Description
[MODULE]			Module parameter definitions
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.
Password:		Up to 20 chars	This parameter sets the password for the module. If the parameter is not set, the module will not be password protected. If a password is present, it will be required for FTP access. The module interprets the password as a case-sensitive string. Do not include spaces or special characters in the password. Only alpha and numeric characters should be used.

[Section]/Item	Value	Range	Description
[SMGC Client 0]			Client definition
Server IP Address:		Up to 20 Chars.	This parameter specifies the IP address in dotted notation (That is, 192.168.0.100) of the server to interface with. If this parameter is set incorrectly, the client will never connect to the server and no communications will be possible.
Service Port Number:		1 to 65535	This parameter sets the TCP/IP service port on the server to connect to. If this parameter is set incorrectly, the client will never connect to the server and no communications will be possible.
Connection Timeout:		0 or 5000 to 65535	This parameter specifies the number of milliseconds the client will permit the socket to be inactive after a connection is made before closing the socket. This timeout period is reset on each read or write packet. If the parameter is set to 0, the connection will not timeout.

[Section]/Item	Value	Range	Description
[AI DATA]			Section containing list of AI tag names

[Section]/Item	Value	Range	Description
[AO DATA]			Section containing list of AO tag names

[Section]/Item	Value	Range	Description
[DI DATA]			Section containing list of DI tag names

[Section]/Item	Value	Range	Description
[DO DATA]			Section containing list of DO tag names

6 Support, Service & Warranty

6.1 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>.

6.2 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