ProSoft TECHNOLOGY

Where Automation Connects.

MVI56E-SMGC

ControlLogix™ Platform

Enhanced FMC722 TPU General Communication Module

May 26, 2022

USER MANUAL

Your Feedback Please

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

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

MVI56E-SMGC User Manual
For Public Use.

May 26, 2022

Content Disclaimer

This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither ProSoft Technology nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. Information in this document including illustrations, specifications and dimensions may contain technical inaccuracies or typographical errors. ProSoft Technology makes no warranty or representation as to its accuracy and assumes no liability for and reserves the right to correct such inaccuracies or errors at any time without notice. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of ProSoft Technology. All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components. When devices are used for applications with technical safety requirements, the relevant instructions must be followed. Failure to use ProSoft Technology software or approved software with our hardware products may result in injury, harm, or improper operating results. Failure to observe this information can result in injury or equipment damage.

Open Source Information

Open Source Software used in the product

The product contains, among other things, Open Source Software files, as defined below, developed by third parties and licensed under an Open Source Software license. These Open Source Software files are protected by copyright. Your right to use the Open Source Software is governed by the relevant applicable Open Source Software license conditions. Your compliance with those license conditions will entitle you to use the Open Source Software as foreseen in the relevant license. In the event of conflicts between other ProSoft Technology, Inc. license conditions applicable to the product and the Open Source Software license conditions, the Open Source Software conditions shall prevail. The Open Source Software is provided royalty-free (i.e. no fees are charged for exercising the licensed rights). Open Source Software contained in this product and the respective Open Source Software licenses are stated in the module webpage, in the link Open Source.

If Open Source Software contained in this product is licensed under GNU General Public License (GPL), GNU Lesser General Public License (LGPL), Mozilla Public License (MPL) or any other Open Source Software license, which requires that source code is to be made available and such source code is not already delivered together with the product, you can order the corresponding source code of the Open Source Software from ProSoft Technology, Inc. - against payment of the shipping and handling charges - for a period of at least 3 years since purchase of the product. Please send your specific request, within 3 years of the purchase date of this product, together with the name and serial number of the product found on the product label to:

ProSoft Technology, Inc.
Director of Engineering
9201 Camino Media, Suite 200
Bakersfield, CA 93311 USA

Warranty regarding further use of the Open Source Software

ProSoft Technology, Inc. provides no warranty for the Open Source Software contained in this product, if such Open Source Software is used in any manner other than intended by ProSoft Technology, Inc. The licenses listed below define the warranty, if any, from the authors or licensors of the Open Source Software. ProSoft Technology, Inc. specifically disclaims any warranty for defects caused by altering any Open Source Software or the product's configuration. Any warranty claims against ProSoft Technology, Inc. in the event that the Open Source Software contained in this product infringes the intellectual property rights of a third party are excluded. The following disclaimer applies to the GPL and LGPL components in relation to the rights holders:

"This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License and the GNU Lesser General Public License for more details."

For the remaining open source components, the liability exclusions of the rights holders in the respective license texts apply. Technical support, if any, will only be provided for unmodified software.

Important Safety Information

North America Warnings

- **A.** Warning – Explosion Hazard – When in hazardous locations, turn off power before replacing or wiring modules.
- **B.** Warning – Explosion Hazard – Substitution of Any Components May Impair Suitability for Class I, Division 2.
- **C.** Warning – Explosion Hazard – Do Not Disconnect Equipment Unless Power Has Been Switched Off Or The Area is Known To Be Non-Hazardous.
- **D.** Class 2 Power

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 ATEX Certified, tool-secured, 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.** Before operating the reset switch, be sure the area is known to be non-hazardous.
- **E.** If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

Agency Approvals & Certifications

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

For professional users in the European Union

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

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

Contents

1 Start Here

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

- **Rockwell Automation® Studio 5000 software:** launch the program, configure ladder logic, and transfer the ladder logic to the processor
- **Microsoft Windows®:** install and launch programs, execute menu commands, navigate dialog boxes, and enter data
- **Hardware installation and wiring:** install the module, and safely connect FMC722 TPU and ControlLogix devices to a power source and to the MVI56E-SMGC module's application port(s)

1.1 System Requirements

The MVI56E-SMGC module requires the following minimum hardware and software components:

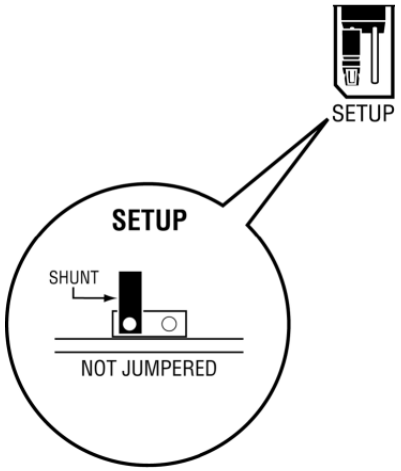
- Rockwell Automation ControlLogix® processor (firmware version 10 or higher) with compatible limited voltage power supply and one free slot in the rack for the MVI56E-SMGC module. The module requires 800mA of available 5 VDC and 3 mA of available 24 VDC power.
- Rockwell Automation Studio 5000 programming software
  - Version 16 or higher required for Add-On Instruction
  - Version 15 or lower must use Sample Ladder, available from www.prosoft-technology.com
- Rockwell Automation RSLinx® communication software version 2.51 or higher
- ProSoft Configuration Builder (PCB) (included)
- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- Supported operating systems:
  - Microsoft Windows 10 Pro x64
  - Microsoft Windows 7 Professional x64
  - Microsoft Windows 7 Professional x86
  - Microsoft Windows XP Pro SP2
- 100 Mbytes of free hard disk space (or more based on application requirements)

Note: The Hardware and Operating System requirements in this list are the minimum recommended to install and run software provided by ProSoft Technology®. Other third party applications may have different minimum requirements. Refer to the documentation for any third party applications for system requirements.

1.2 Setting Jumpers

The Setup Jumper acts as "write protection" for the module's firmware. In "write protected" mode, the Setup pins are not connected, and the module's firmware cannot be overwritten. The module is shipped with the Setup jumper OFF. Do not jumper the Setup pins together unless you are directed to do so by ProSoft Technical Support (or you want to update the module firmware).

The following illustration shows the MVI56E-SMGC jumper configuration with the Setup Jumper OFF.



Note: If you are installing the module in a remote rack, you may prefer to leave the Setup pins jumpered. You can update the module's firmware without requiring physical access to the module.

Security considerations:

Leaving the Setup pin jumpered leaves the module open to unexpected firmware updates.

You should consider segmenting the data flow for security reasons. Per IEC 62443-1-1, you should align with IEC 62443 and implement segmentation of the control system. Relevant capabilities are firewalls, unidirectional communication, DMZ. Oil and Gas customers should also see DNVGL-RP-G108 for guidance on partitioning.

You should practice security by design, per IEC 62443-4-1, including layers of security and detection. The module relies on overall network security design, as it is only one component of what should be a defined zone or subnet.

1.3 Installing the Module in the Rack

Make sure your ControlLogix processor and power supply are installed and configured, before installing the MVI56E-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 the MVI56E-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. An electrical arc can cause personal injury or property damage by sending an erroneous signal to the system's actuators. This can cause unintended machine motion or loss of process control. Electrical arcs may also cause an explosion when they happen in a hazardous environment. 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.

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

Note: When using the XT version (if applicable), you must use the 1756-A5XT or 1756-A7LXT chassis to uphold the XT specifications. In these chassis, modules are spaced further apart than in standard ControlLogix chassis. Spacers are inserted between active modules.

1.3.1 Before You Import the Add-On Instruction

Note: This section only applies if your processor is using Studio 5000 version 16 or higher.

An optional Add-On Instruction file is provided for the MVI56E-SMGC module. The main logic will be provided as a ladder logic file (not Add-On Instruction).

Download the files from www.prosoft-technology.com. Save them to a convenient location in your PC, such as *Desktop* or *My Documents*.

File Name	Description
MVI56ESMGC_Optional_AddOn_Rung_v1_x.L5X. A newer version may be available at: www.prosoft-technology.com	Optional L5X file containing additional Add-On Instruction with logic for changing Ethernet configuration and clock settings.

1.3.1.1 About the Optional Add-On Instruction

The Optional Add-On Instruction performs the following tasks:

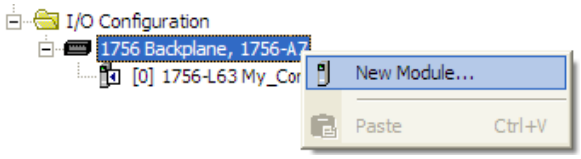
- **Read/Write Ethernet Configuration**. Allows the processor to read or write the module IP address, subnet mask, and network gateway IP address.
- **Read/Write Module Clock Value**. Allows the processor to read and write the module clock settings. The module's free-running clock also stores the last time that the Ethernet configuration was changed or the last time the module was restarted or rebooted. The date and time of the last change or restart is displayed on the scrolling LED during module power-up/start-up sequence.

Note: The date and time can also be configured in the module's webpage.

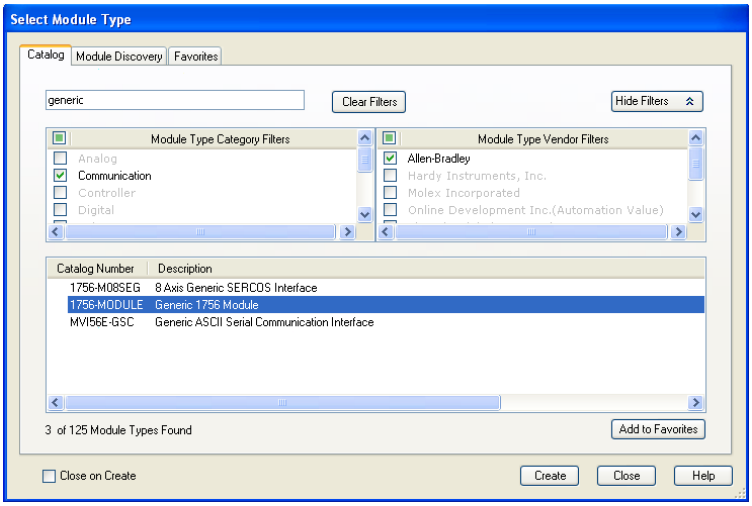
Important: The Optional Add-On Instruction supports only the two features listed above. Use the regular MVI56E-SMGC Add-On Instruction for all other features including backplane transfer and FMC722 TPU data communication.

1.3.2 Creating the Module

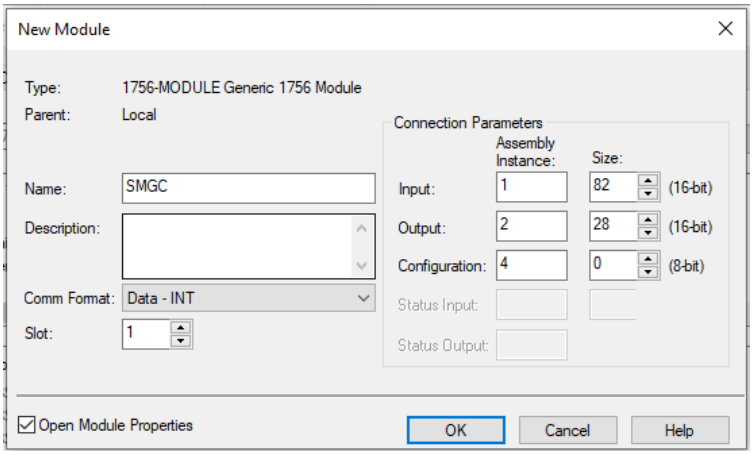
1 Add the MVI56E-SMGC module to the project. In the **CONTROLLER ORGANIZATION** window, select **I/O CONFIGURATION** and click the right mouse button to open a shortcut menu. On the shortcut menu, choose **NEW MODULE...**



This action opens the **SELECT MODULE** dialog box. Enter *generic* in the text box and select the **GENERIC 1756 MODULE**. If you're using a controller revision of 16 or less, expand **OTHER** in the **SELECT MODULE** dialog box, and then select the **GENERIC 1756 MODULE**.



2 Click **CREATE.** This action opens the **NEW MODULE** dialog box.



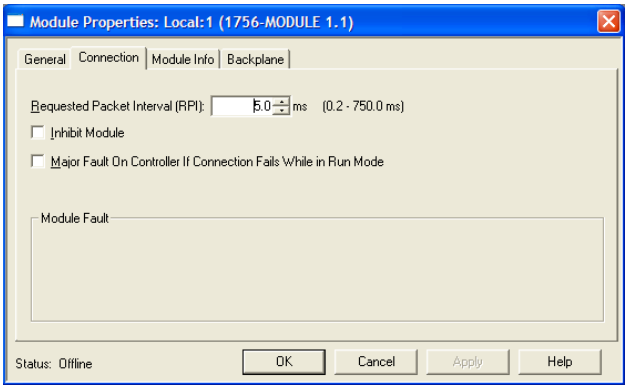
3 In the **NEW MODULE** dialog box, enter the following values.

Parameter	Value
Name	Enter a module identification string. **Example:** SMGC
Description	Enter a description for the module. **Example:** Enhanced SM General Communication Module
Comm Format	Select **DATA-INT**
Slot	Enter the slot number in the rack where the MVI56E-SMGC module is located
Input Assembly Instance	1
Input Size	82
Output Assembly Instance	2
Output Size	28
Configuration Assembly Instance	4
Configuration Size	0

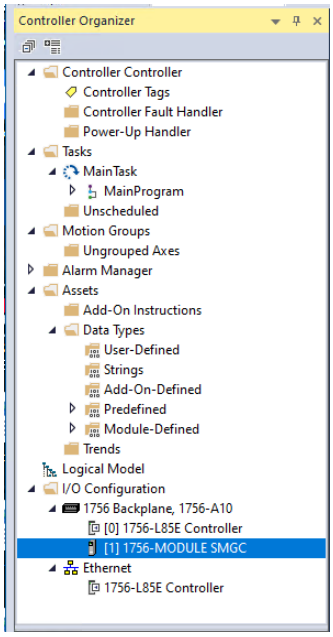
Important: You must select the **COMM FORMAT** as **DATA - INT** in the dialog box, otherwise the module will not communicate over the backplane of the ControlLogix rack.

4 Click **OK** to continue.

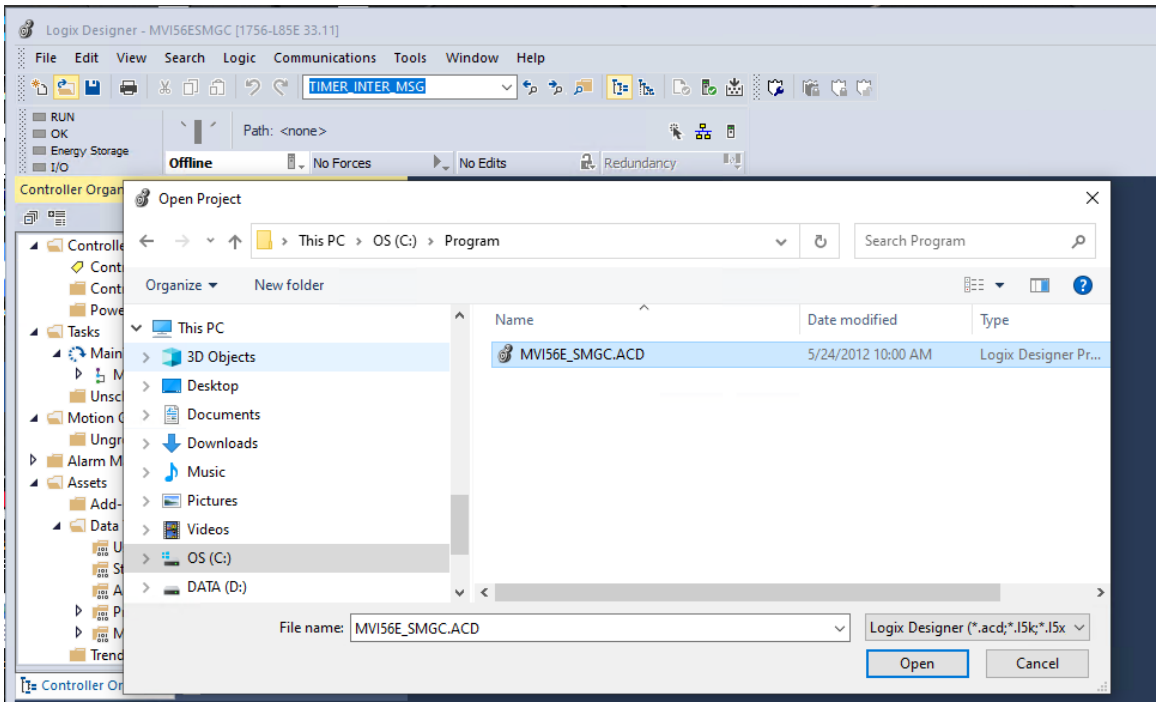
5 Edit the *Module Properties*. Select the *Requested Packet Interval* value for scanning the I/O on the module. This value represents the minimum frequency at which the module will handle scheduled events. This value should not be set to less than 1 millisecond. The default value is 5 milliseconds.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



6 Click **OK** to save the module and close the dialog box. Notice that the module now appears in the **CONTROLLER ORGANIZATION** window.

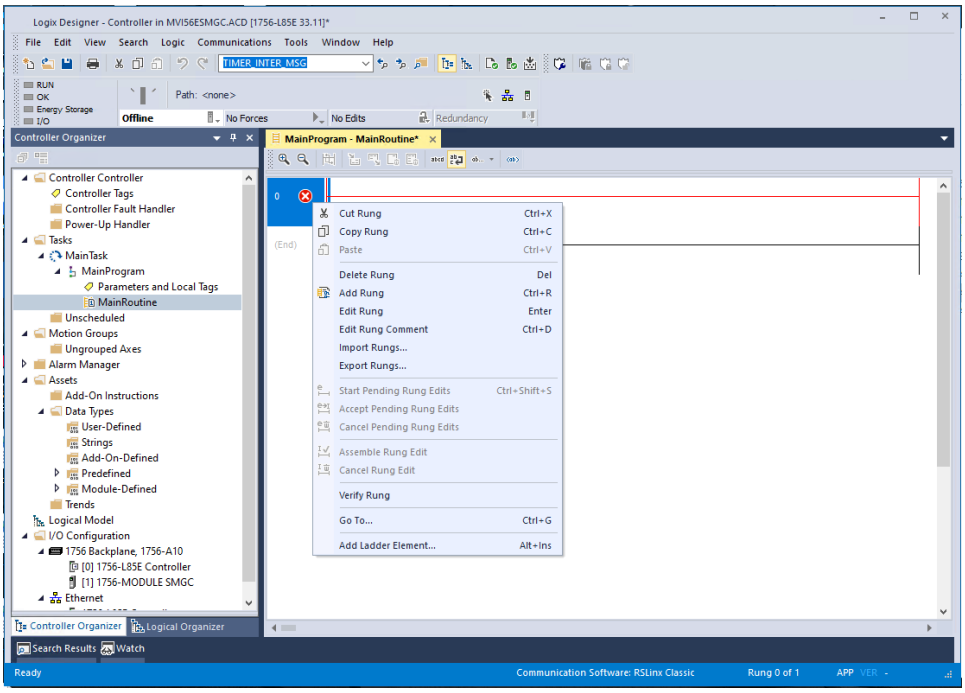


7 In order to open the sample ladder logic, click **FILE/OPEN…**

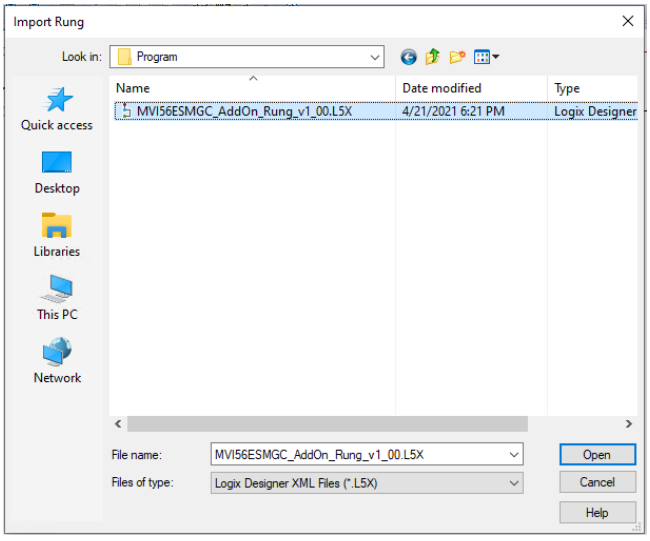


1.3.3 Importing the Optional Add-On Instruction

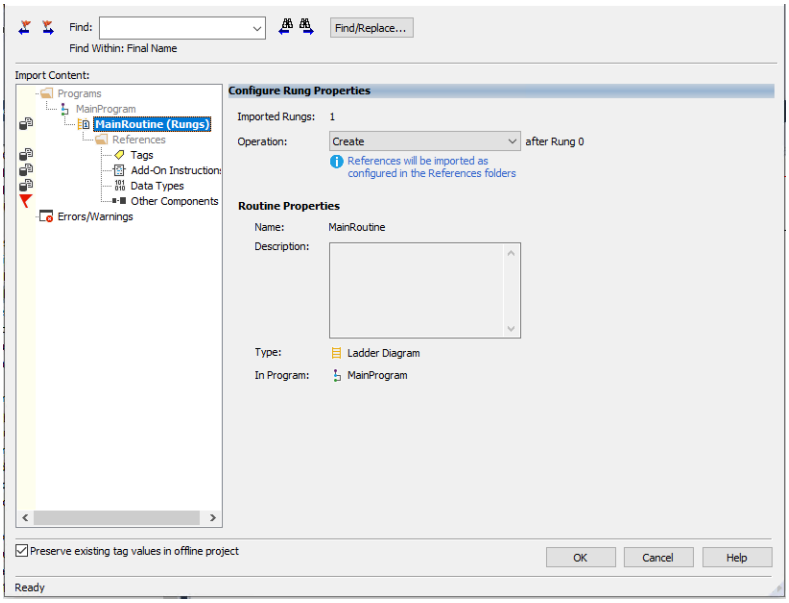
1. In the *Controller Organization* window, expand the *Tasks* folder and subfolders until you reach the *MainProgram* folder.
2. In the *MainProgram* folder, double-click to open the *MainRoutine* ladder. Select an empty rung in the routine, and then click the right mouse button to open a shortcut menu. On the shortcut menu, choose **IMPORT RUNGS…**



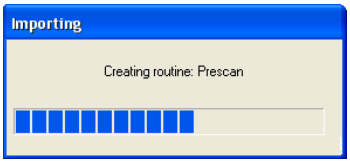
3. Navigate to the location on your PC where you saved the Add-On Instruction (for example, *My Documents* or *Desktop*). Select the .L5X file.



4 This action opens the **IMPORT CONFIGURATION** dialog box. Click **TAGS** under **MAINROUTINE** to show the controller tags that will be created. Note that if you are using a controller revision number of 16 or less, the **IMPORT CONFIGURATION** dialog box does not show the **IMPORT CONTENT** tree.



5 Click **OK** to confirm the import.



6 When the import is completed, the new rung with the Add-On Instruction appears as shown in the following illustration.

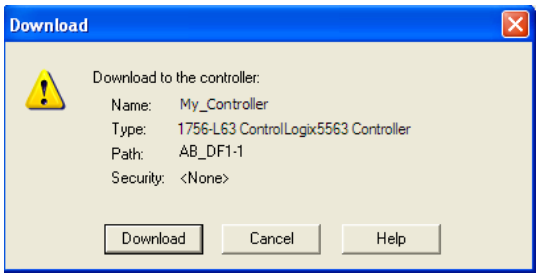
7 The procedure also imports the new User Defined Data Types, Controller Tags, and the Add-On instruction for your project.

8 Save the application and then download the sample ladder logic to the processor.

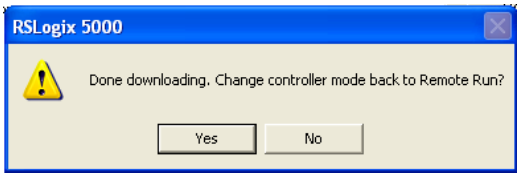
1.4 Downloading the Sample Program to the Processor

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

1. If you are not already online with the processor, in Studio 5000 open the *Communications* menu, and then choose **DOWNLOAD**. Studio 5000 will establish communication with the processor. You may download through any available network connection.
2. When communication is established, Studio 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



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



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

2 MVI56E-SMGC Configuration

2.1 Using ProSoft Configuration Builder Software

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

Note: During startup and initialization, the MVI56E-SMGC module receives its protocol and backplane configuration information from the installed Personality Module (Compact Flash). Use *ProSoft Configuration Builder* to configure module settings and to download changes to the Personality Module.

2.1.1 Installing ProSoft Configuration Builder

Use the ProSoft Configuration Builder (PCB) software to configure the module. You can find the latest version of the ProSoft Configuration Builder (PCB) on our web site: www.prosoft-technology.com. The installation filename contains the PCB version number. For example, **PCB_4.3.4.5.0238.EXE**.

1. Open a browser window and navigate to www.prosoft-technology.com.
2. Perform a search for 'pcb' in the Search bar. Click on the ProSoft Configuration Builder search result.
3. On the PCB page, click the download link for ProSoft Configuration Builder, and save the file to your Windows desktop.
4. After the download completes, double-click the file to install. If you are using Windows 7, right-click the PCB installation file and then choose **RUN AS ADMINISTRATOR**. Follow the instructions that appear on the screen.
5. If you want to find additional software specific to your MVI56E-SMGC, enter the model number into the ProSoft website search box and press the **ENTER** key.

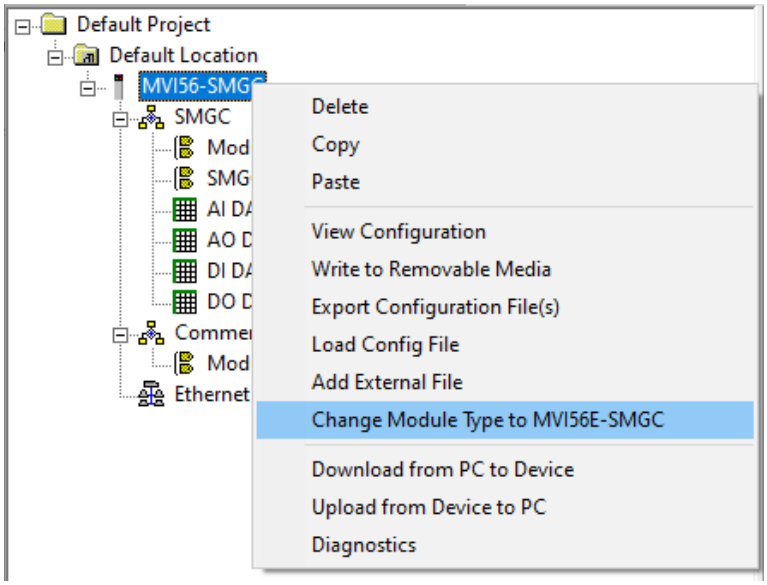
2.1.2 Upgrading from MVI56-SMGC in ProSoft Configuration Builder

MVI56E-SMGC modules are fully backward-compatible with MVI56-SMGC modules. However, you will need to convert your MVI56-SMGC configuration in *ProSoft Configuration Builder* to a form that your new MVI56E-SMGC module will accept when you download it.

ProSoft Configuration Builder version 2.2.2 or later has an upgrade option that easily performs this conversion, while preserving all your configuration settings and any name you may have given your module.

Important: For this procedure, you need to have *ProSoft Configuration Builder* version 2.2.2 or later installed on your PC. You can download the latest version from www.prosoft-technology.com.

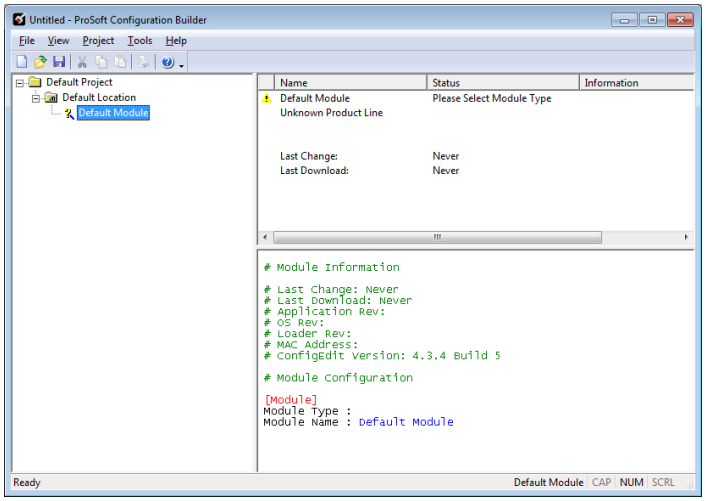
1 In *ProSoft Configuration Builder's* tree view, click the **MODULE** icon and right-click to open a shortcut menu.



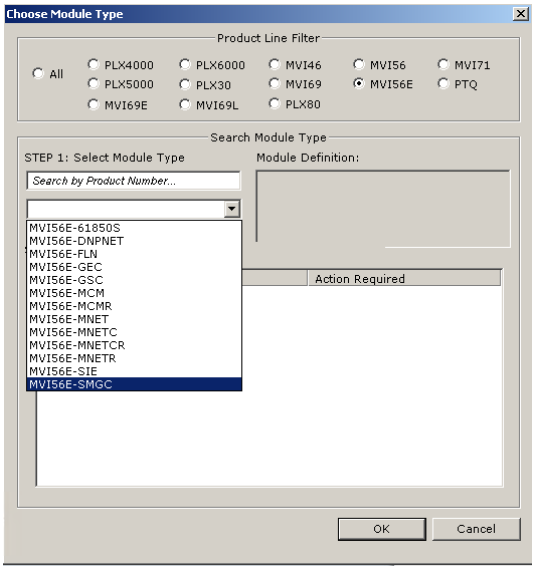
2 On the shortcut menu, select Change Module Type to MVI56E-SMGC.

2.1.3 Setting Up the Project

If you have used other Windows configuration tools before, you will find the screen layout familiar. PCB's window consists of a tree view on the left, and an information pane and a configuration pane on the right side of the window. The following illustration shows the PCB window with a new project.



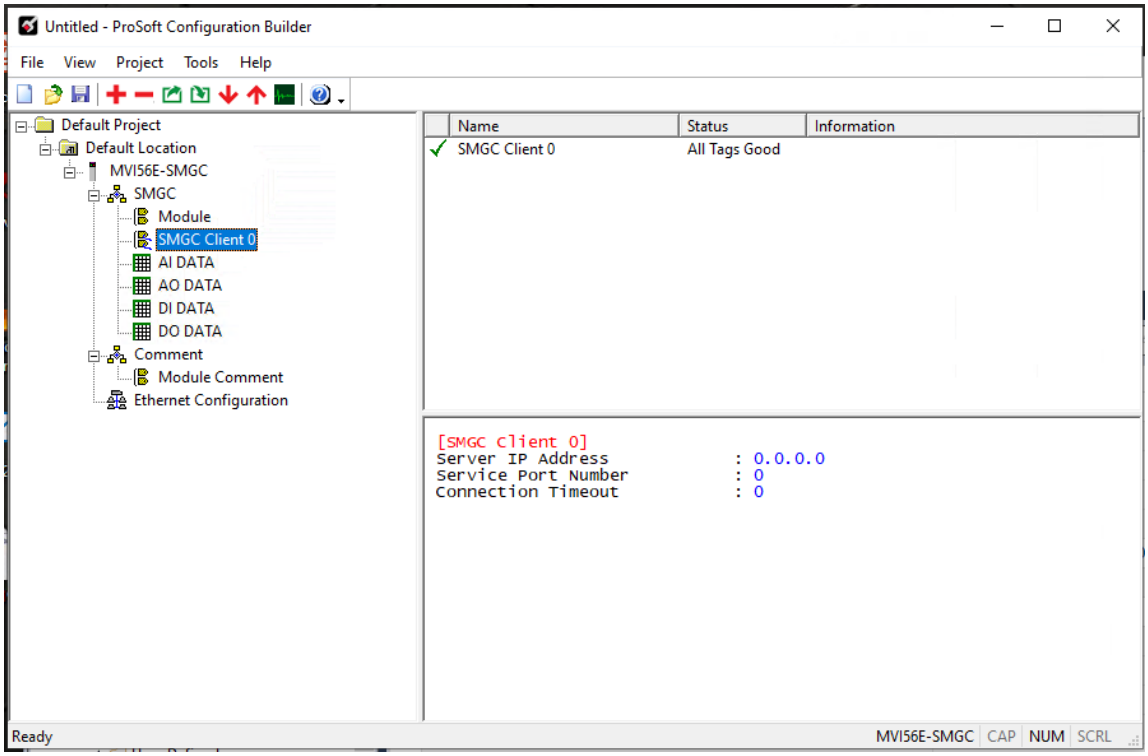
1 In PCB, right-click **DEFAULT MODULE** in the tree view and then select **CHOOSE MODULE TYPE**. This opens the *Choose Module Type* dialog box.



2 In the *Product Line Filter* area of the dialog box, select **MVI56E**. In the *Select Module Type* dropdown list, select **MVI56E-SMGC**, and then click **OK** to save your settings and return to the *ProSoft Configuration Builder* window.

2.1.4 Setting Module Parameters

Notice that the contents of the information pane and the configuration pane changed when you added the MVI56E-SMGC module to the project.



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

2.1.4.1 Renaming an Object

1. Right-click the object and then choose **RENAME.**
2. Type the name to assign to the object.
3. Click away from the object to save the new name.

2.1.4.2 Configuring Module Parameters

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

2.1.4.3 Printing a Configuration File

1. In the main PCB window, right-click the **MVI56E-SMGC MODULE** icon and then choose **VIEW CONFIGURATION**.
2. In the *View Configuration* dialog box, click the **FILE** menu and then click **PRINT**.
3. In the *Print* dialog box, choose the printer to use from the drop-down list, select the printing options, and then click **OK**.

2.2 Connecting Your PC to the Module

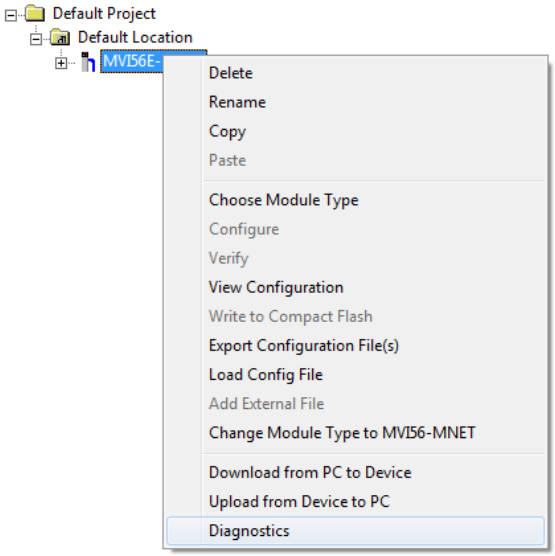
2.2.1 Using CIPconnect to Connect to the Module

You can use CIPconnect® to connect a PC to the MVI56E-SMGC module over Ethernet using Rockwell Automation's 1756-ENBT EtherNet/IP® module. This allows you to configure the MVI56E-SMGC module and network, upload and download files, and view network and module diagnostics from a PC. RSLinx is not required when you use CIPconnect. All you need are:

- The IP addresses and slot numbers of any 1756-ENBT modules in the path
- The ControlNet node numbers and slot numbers of any 1756-CNBx ControlNet Bridge modules in the path
- The slot number of the MVI56E-SMGC in the destination ControlLogix chassis (the last ENBT/CNBx and chassis in the path).

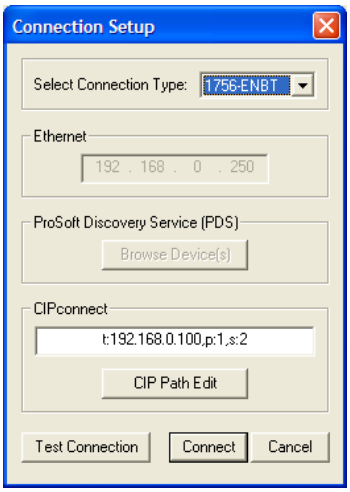
To use CIPconnect, follow these steps.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** icon and then choose **DIAGNOSTICS.**

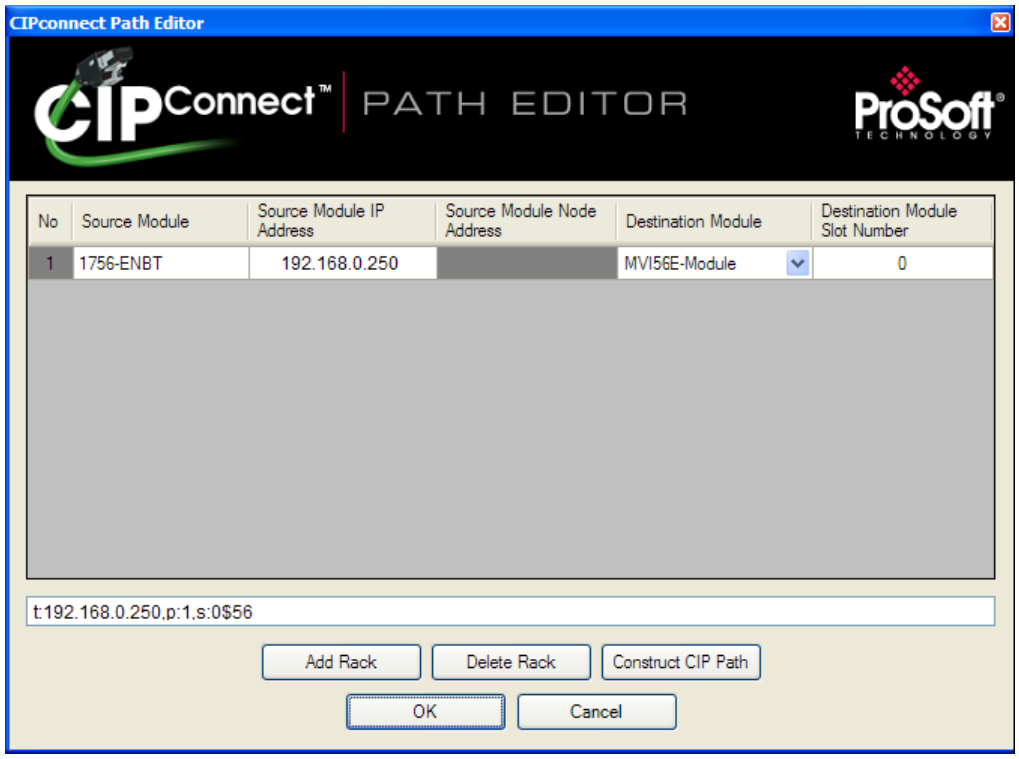


2. In the *Diagnostics* window, click the **SET UP CONNECTION** button.

3 In the *Select Connection Type* dropdown list, choose **1756-ENBT**. The default path appears in the text box, as shown in the following illustration.



4 Click **CIP PATH EDIT** to open the *CIPconnect Path Editor* dialog box. The *CIPconnect Path Editor* allows you to define the path between the PC and the MVI56E-SMGC module. The first connection from the PC is always a 1756-ENBT (Ethernet/IP) module.



Each row corresponds to a physical rack in the CIP path.

- If the MVI56E-SMGC module is located in the same rack as the first 1756-ENBT module, select **RACK NO. 1** and configure the associated parameters.
- If the MVI56E-SMGC is available in a remote rack (accessible through ControlNet or Ethernet/IP), include all racks (by using the **ADD RACK** button).

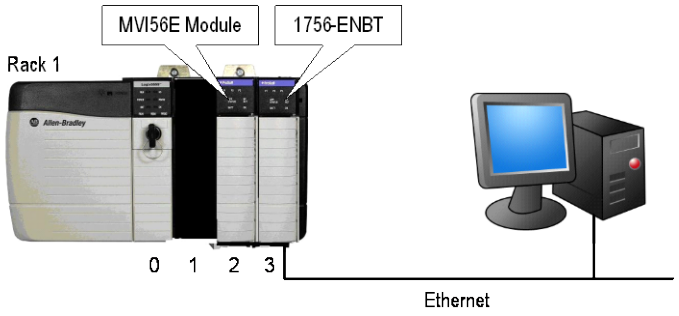
Parameter	Description
Source Module	Source module type. This field is automatically selected depending on the destination module of the last rack (1756-CNB or 1756-ENBT).
Source Module IP Address	IP address of the source module (only applicable for 1756-ENBT)
Source Module Node Address	Node address of the source module (only applicable for 1756-CNB)
Destination Module	Select the destination module associated to the source module in the rack. The connection between the source and destination modules is performed through the backplane.
Destination Module Slot Number	The slot number where the destination MVI56E-SMGC module is located.

To use the CIPconnect Path Editor, follow these steps.

1. Configure the path between the 1756-ENBT connected to your PC and the MVI56E-SMGC module.
2. If the module is located in a remote rack, add more racks to configure the full path.
3. The path can only contain ControlNet or Ethernet/IP networks.
4. The maximum number of supported racks is six.
5. Click **CONSTRUCT CIP PATH** to build the path in text format.
6. Click **OK** to confirm the configured path.

2.2.1.1 Example 1: Local Rack Application

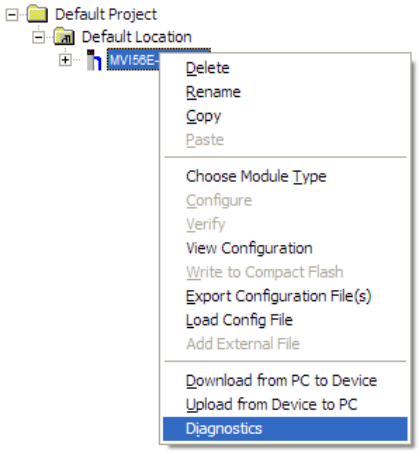
For this example, the MVI56E-SMGC module is located in the same rack as the 1756-ENBT that is connected to the PC.



Rack 1

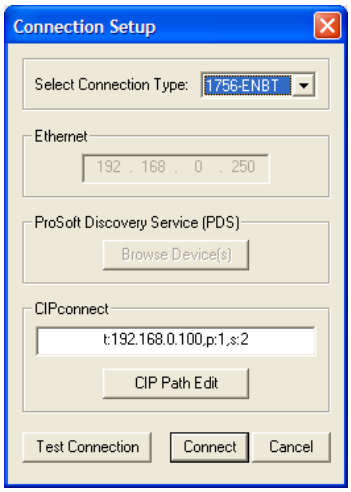
Slot	Module	Network Address
0	ControlLogix Processor	-
1	Any	-
2	MVI56E-SMGC	-
3	1756-ENBT	192.168.0.100

1 In *ProSoft Configuration Builder*, right-click the MVI56E-SMGC icon and then choose **DIAGNOSTICS.**

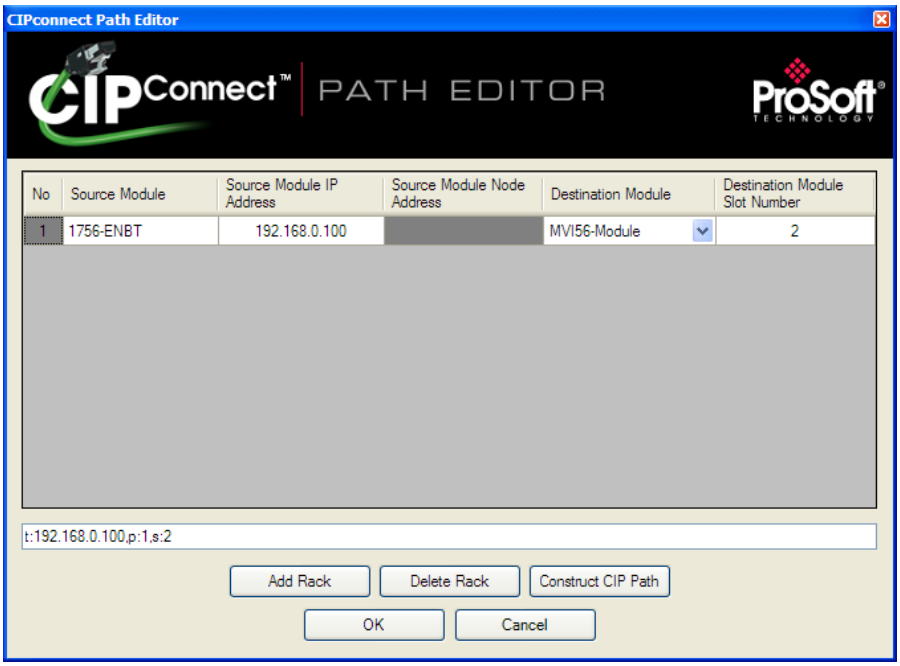


2 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

3 In the *Select Connection Type* dropdown list, choose **1756-ENBT**. The default path appears in the text box, as shown in the following illustration.

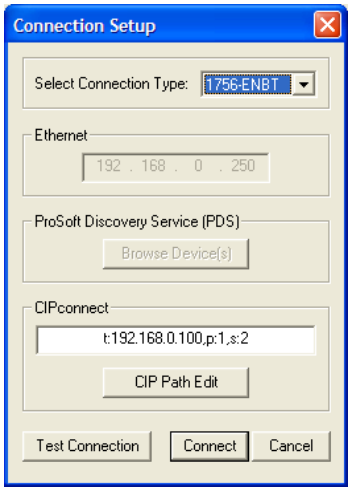


4 Configure the path as shown in the following illustration, and click **CONSTRUCT CIP PATH** to build the path in text format.

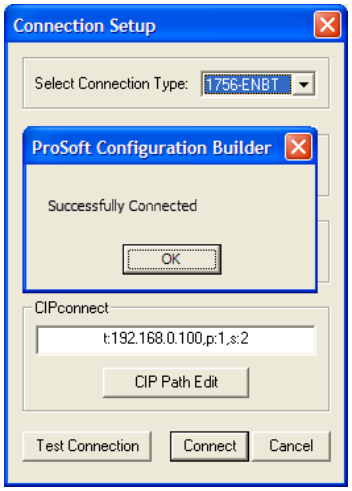


5 Click **OK** to close the *CIPconnect Path Editor* and return to the *Connection Setup* dialog box.

6 Check the new path in the *Connection Setup* dialog box.



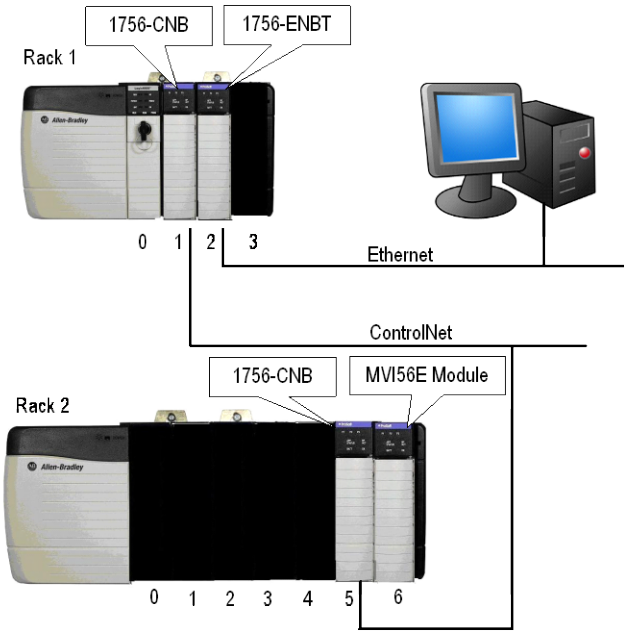
7 Click **TEST CONNECTION** to verify that the physical path is available. The following message should be displayed upon success.



8 Click **OK** to close the *Test Connection* pop-up and then click **CONNECT** to close the *Connection Set* up dialog box. The Diagnostics menu is now connected through CIPconnect.

2.2.1.2 Example 2: Remote Rack Application

For this example, the MVI56E-SMGC module is located in a remote rack accessible through ControlNet, as shown in the following illustration.



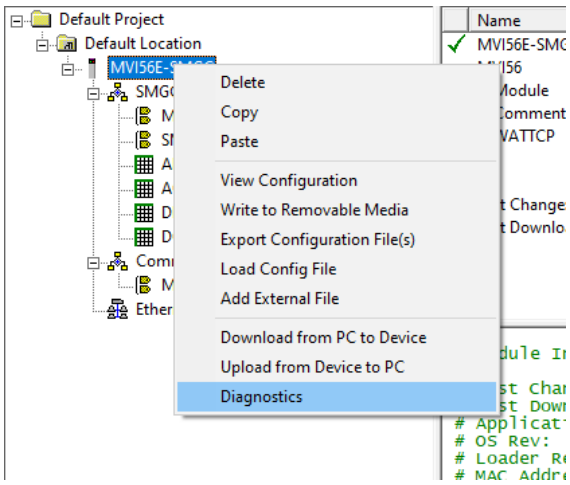
Rack 1

Slot	Module	Network Address
0	ControlLogix Processor	-
1	1756-CNB	Node = 1
2	1756-ENBT	192.168.0.100
3	Any	-

Rack 2

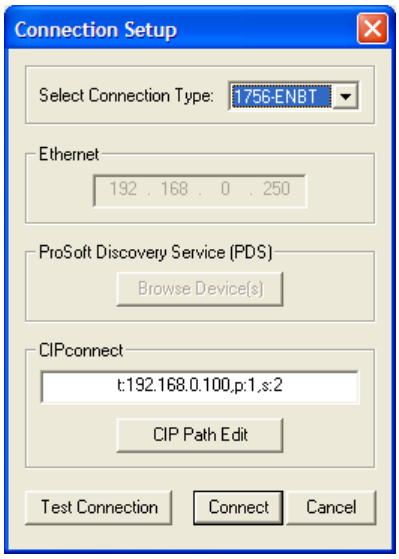
Slot	Module	Network Address
0	Any	-
1	Any	-
2	Any	-
3	Any	-
4	Any	-
5	1756-CNB	Node = 2
6	MVI56E-SMGC	-

1 In *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** icon and then choose **DIAGNOSTICS.**

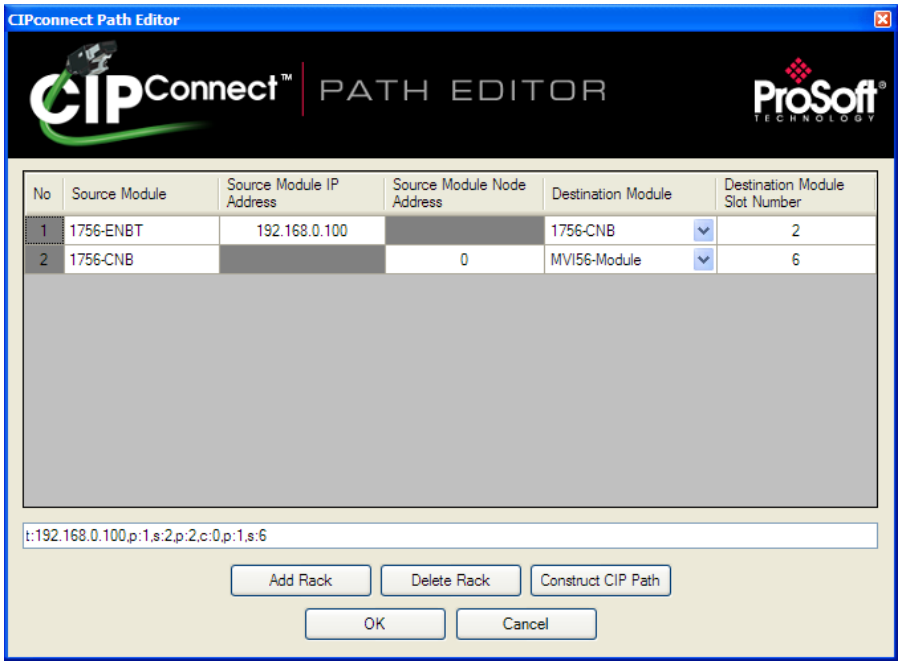


2 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

3 In the *Select Connection Type* dropdown list, choose **1756-ENBT**. The default path appears in the text box, as shown in the following illustration.

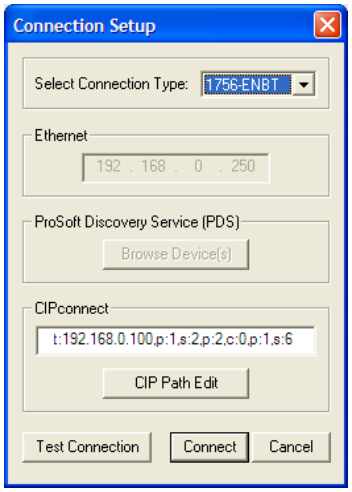


4 Configure the path as shown in the following illustration, and click **CONSTRUCT CIP PATH** to build the path in text format.

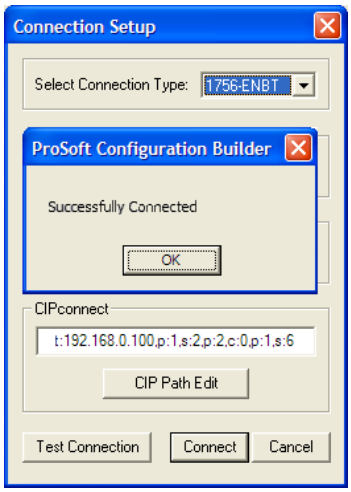


5 Click **OK** to close the *CIPconnect Path Editor* and return to the *Connection Setup* dialog box.

6 Check the new path in the *Connection Setup* dialog box.



7 Click **TEST CONNECTION** to verify that the physical path is available. The following message should be displayed upon success.

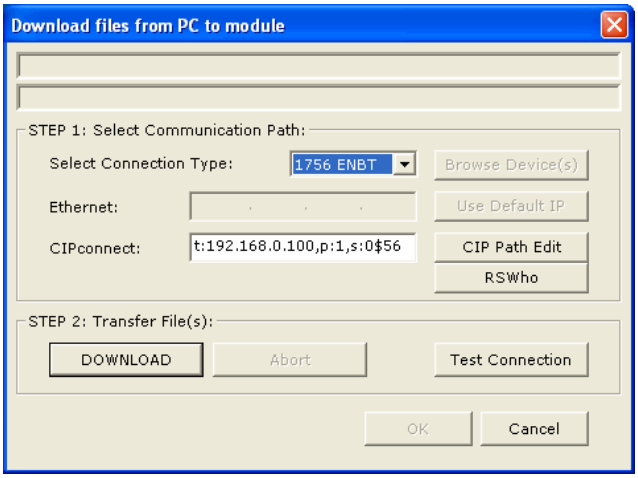


8 Click **OK** to close the *Test Connection* pop-up and then click **CONNECT** to close the *Connection Set* up dialog box. The Diagnostics menu is now connected through CIPconnect.

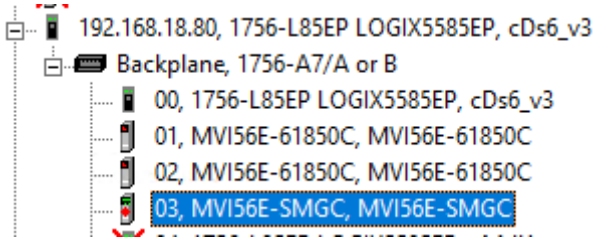
2.2.2 Using RSWho to Connect to the Module

Note: RSLinx must be installed on your PC to use this feature. An ENBT module must also be configured in the rack.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** module.
2. From the shortcut menu, choose **DOWNLOAD FROM PC TO DEVICE**.
3. In the *Download* dialog box, choose **1756 ENBT** from the *Select Connection Type* dropdown box.



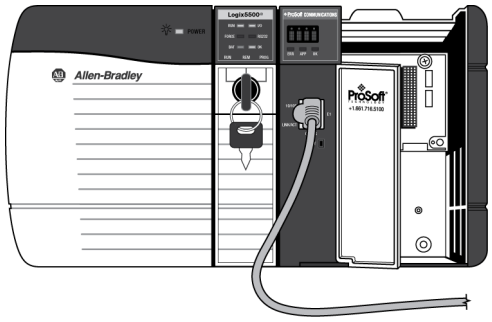
4. Click **RSWHO** to display modules on the network. The MVI56E-SMGC module will automatically be identified on the network.



5. Select the module, and then click **OK**.

2.2.3 Connecting Your PC to the Module's Ethernet Port

With the module securely mounted, connect one end of the Ethernet cable to the *Config (E1)* Port, and the other end to an Ethernet hub or switch accessible from the same network as your PC. You can also connect directly from the Ethernet Port on your PC to the *Config (E1)* Port on the module by using an Ethernet crossover cable (not included).

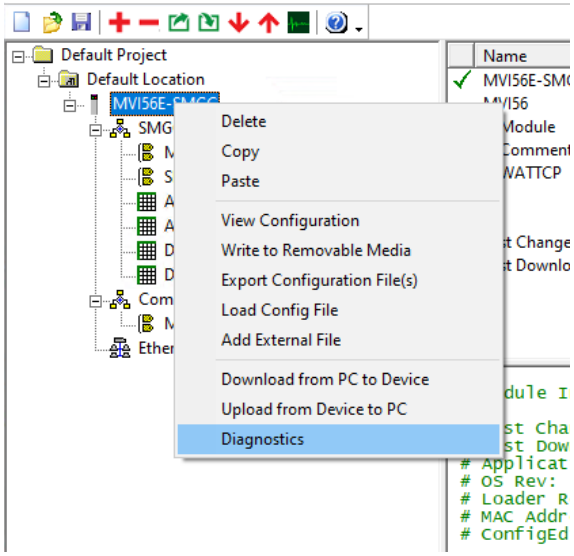


2.2.3.1 Setting Up a Temporary IP Address

Important: *ProSoft Configuration Builder* locates MVI56E-SMGC modules through UDP broadcast messages. These messages may be blocked by routers or layer 3 switches. In that case, *ProSoft Discovery Service* will be unable to locate the modules.

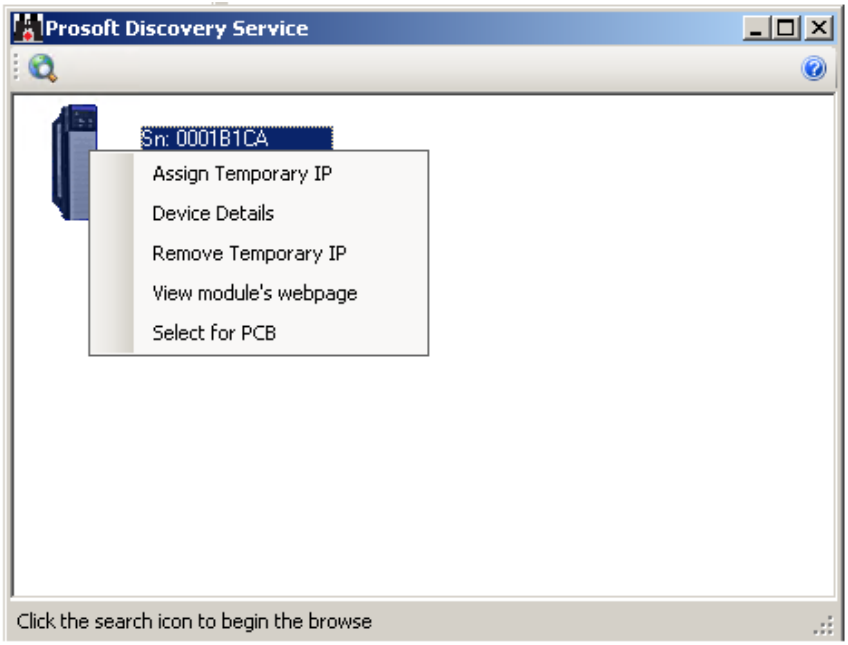
To use *ProSoft Configuration Builder*, arrange the Ethernet connection so that there is no router/ layer 3 switch between the computer and the module OR reconfigure the router/ layer 3 switch to allow routing of the UDP broadcast messages.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** icon to open a shortcut menu.
2. On the shortcut menu, choose **DIAGNOSTICS.**

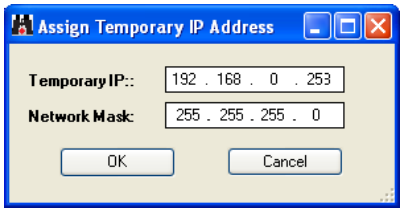


3 In the *Diagnostics* window, click the **SET UP CONNECTION** button.

4 In the *Connection Setup* dialog box, click the **BROWSE DEVICE(S)** button to open the *ProSoft Discovery Service*. Select the module, then right-click and choose **ASSIGN TEMPORARY IP**.

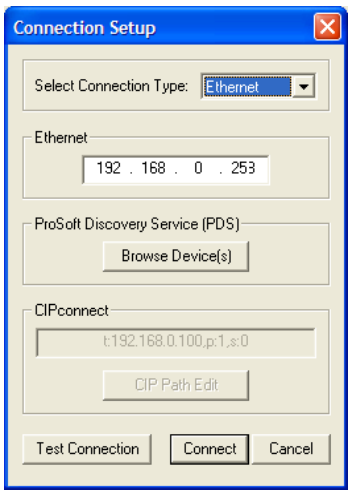


5 The module's default IP address is 192.168.0.250. Choose an unused IP within your subnet, and then click **OK**.

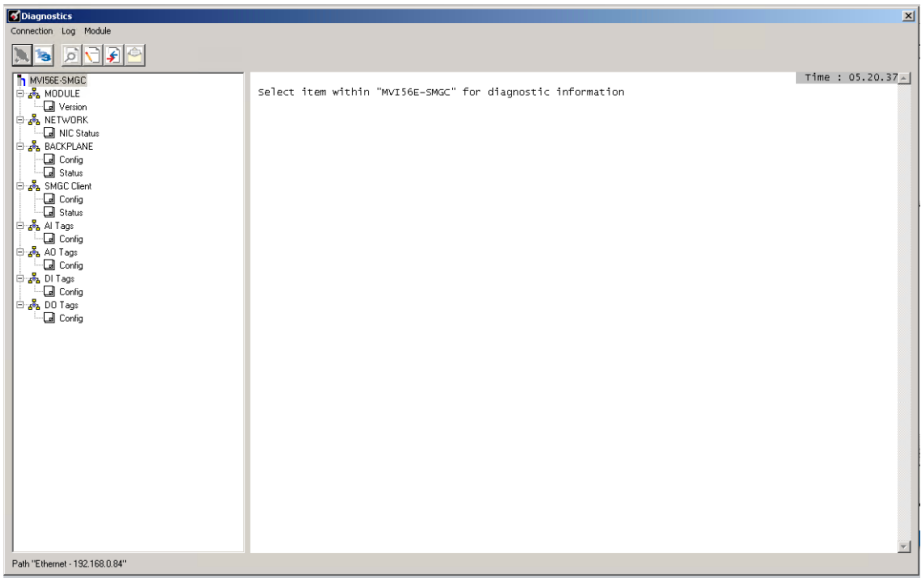


Important: The temporary IP address is only valid until the next time the module is initialized.

6 Close the *ProSoft Discovery Service* window. Enter the temporary IP in the Ethernet address field of the *Connection Setup* dialog box, then click the **TEST CONNECTION** button to verify that the module is accessible with the current settings.



7 If the *Test Connection* is successful, click **CONNECT**. The *Diagnostics* menu will display in the *Diagnostics* window.

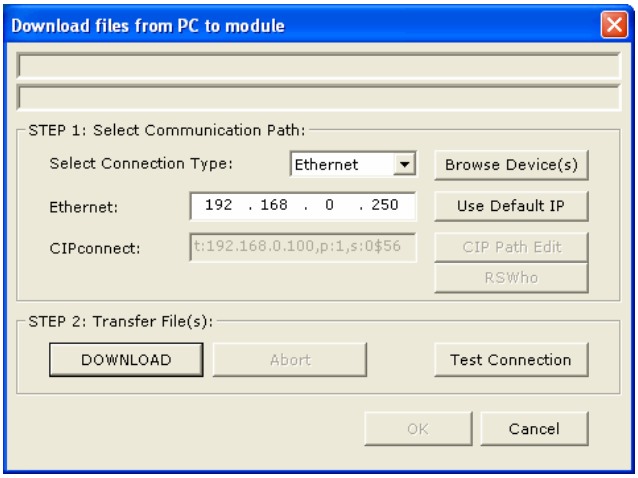


2.3 Downloading the Project to the Module

In order for the module to use the settings you configured, you must download the updated Project file from your PC to the module.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** icon and then choose **DOWNLOAD FROM PC TO DEVICE.** This opens the *Download* dialog box.
2. In the *Download* dialog box, choose the connection type in the *Select Connection Type* dropdown box:
   - Choose **ETHERNET** if you are connecting to the module through the Ethernet cable.
   - Choose **1756 ENBT** if you are connecting to the module through CIPconnect or RSWho.

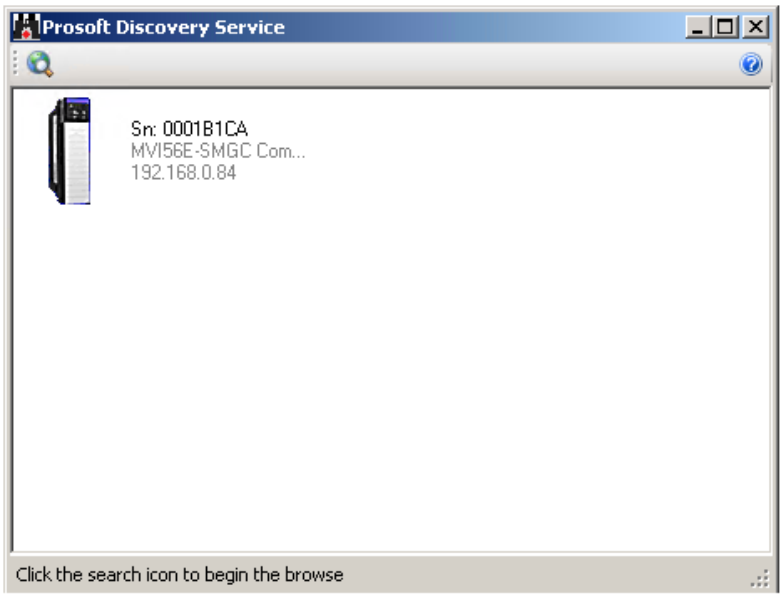
Note: If you connected to the module using an Ethernet cable and set a temporary IP address, the Ethernet address field contains that temporary IP address. ProSoft Configuration Builder uses this temporary IP address to connect to the module.



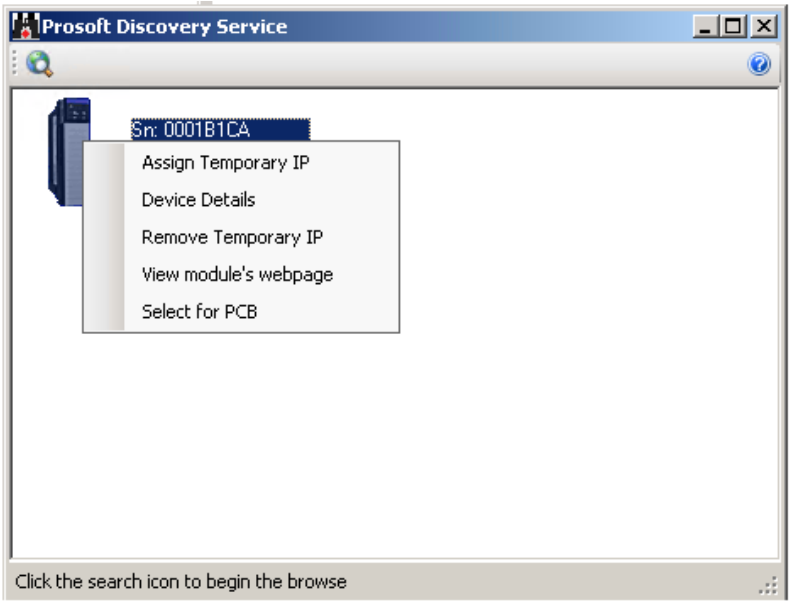
3. Click **TEST CONNECTION** to verify that the IP address allows access to the module.
4. If the connection succeeds, click **DOWNLOAD** to transfer the Ethernet configuration to the module.

If the *Test Connection* procedure fails, you will see an error message. To correct the error, follow these steps.

1. Click **OK** to dismiss the error message.
2. In the *Download* dialog box, click **BROWSE DEVICE(S)** to open *ProSoft Discovery Service*.



3. Right-click the module and then choose **SELECT FOR PCB**.



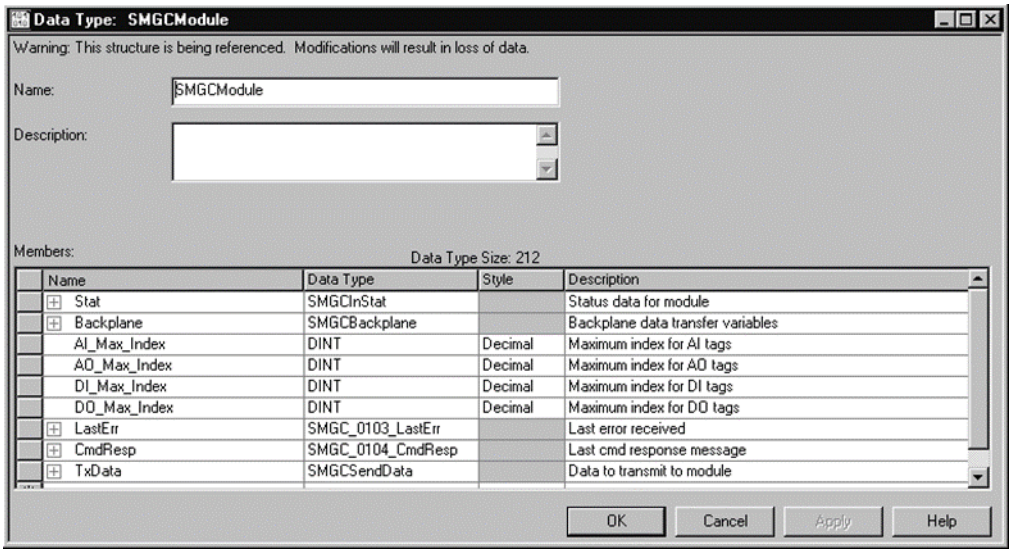
4. Close *ProSoft Discovery Service*.
5. Click **DOWNLOAD** to transfer the configuration to the module.

3 Using Controller Tags

All data related to the MVI56E-SMGC is stored in 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.

3.1 Main Data Object (SMGCModule)

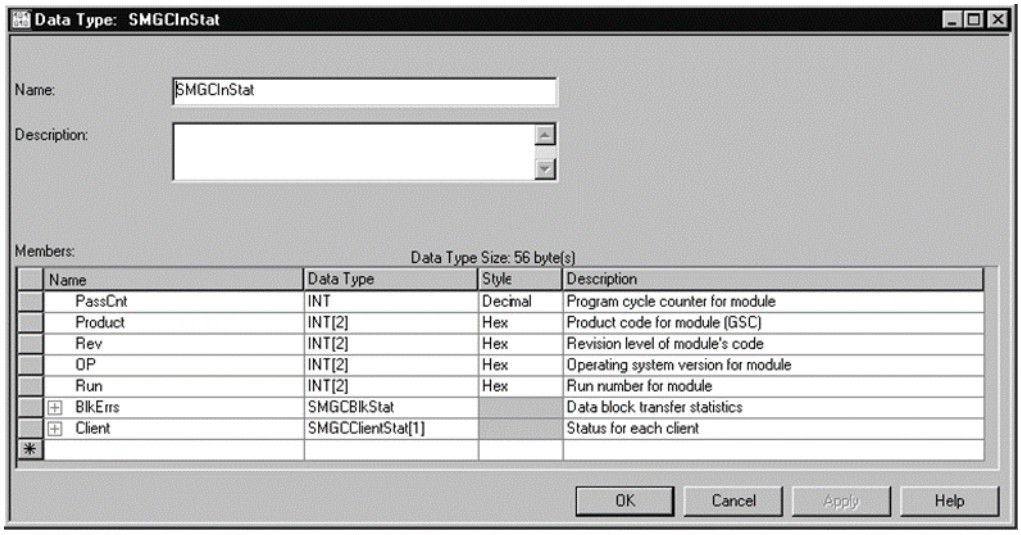
The *SMGCModule* object contains most of the data members and objects required by the MVI56E-SMGC module. An instance of this object should be declared in the controller tags of the processor. The following figure displays this object:



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

3.1.1 Status Object (SMGCInStat)

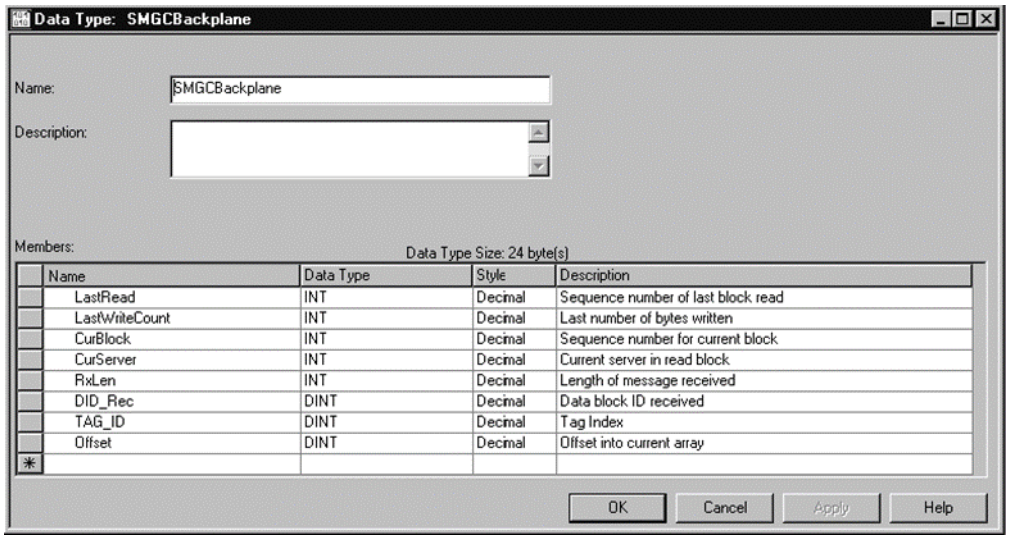
This object views the status of the module. The *SMGCInStat* object is updated each time a read block is received by the processor. Use this data to monitor the state of the module at a "real-time rate".



Within the *SMGCInStat* objects are objects containing the status information for the block transfer process and the client.

3.1.2 Backplane Variables (SMGCBackplane)

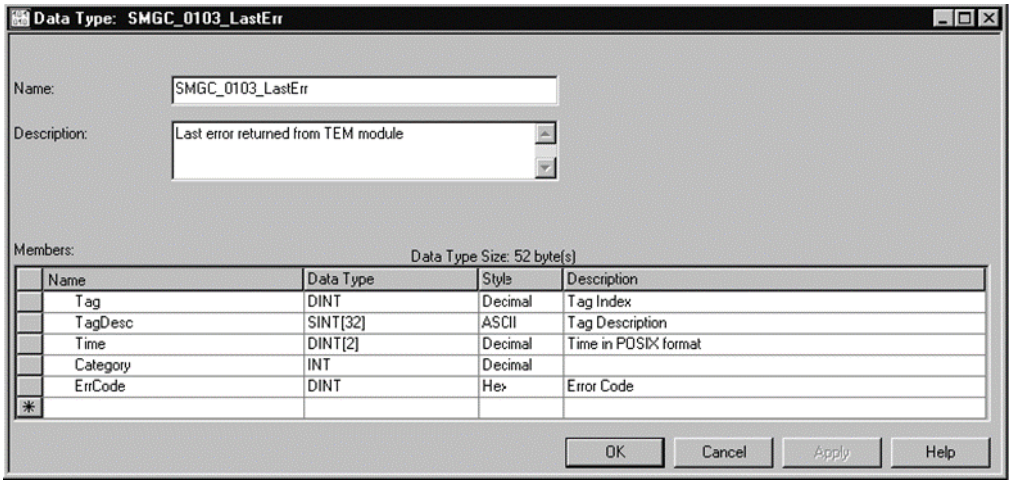
The *SMGCBackplane* object stores all the variables required for the data transfer operation between the module and the controller. The *LastRead* data member is used as the handshaking byte to indicate the arrival of new data from the module. The structure of this object is shown in the following illustration:



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

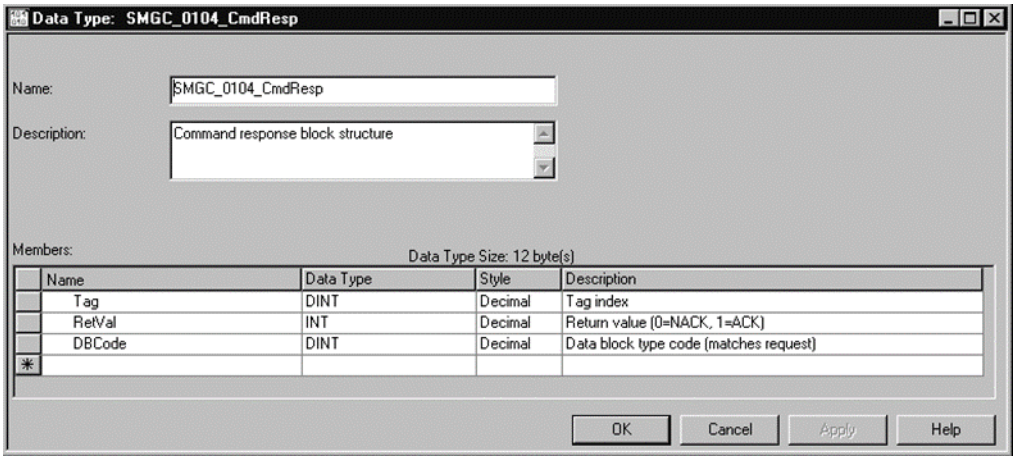
3.1.3 Last Error Data (SMGC_0103_LastErr)

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:



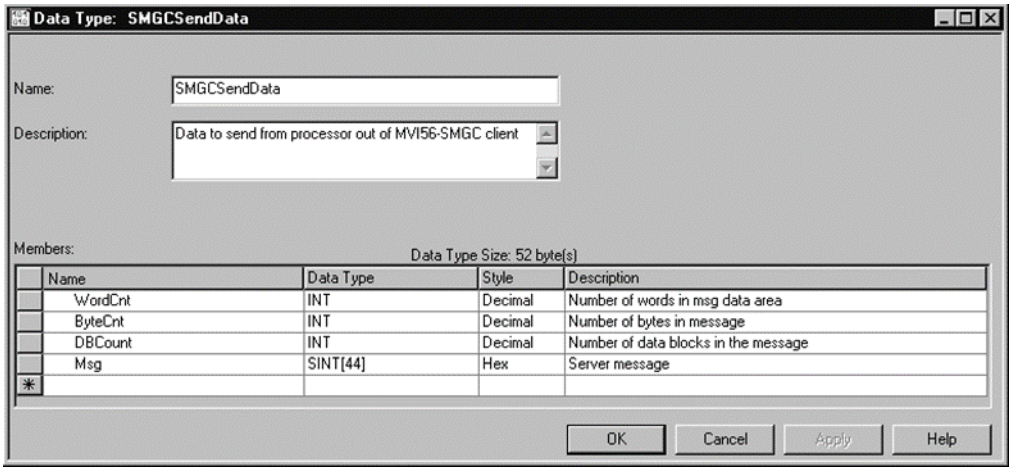
3.1.4 Command Response Data (SMGC_0104_CmdResp)

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:



3.1.5 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:



3.1.6 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:

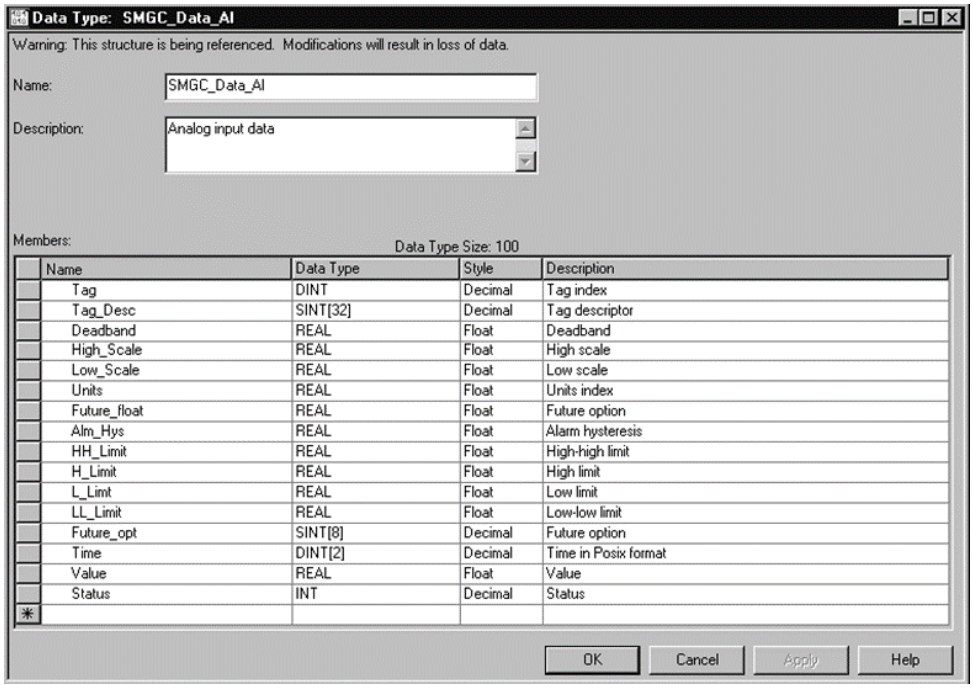
Object	Description
AI_Max_Index	Maximum index for AI array
AO_Max_Index	Maximum index for AO array
DI_Max_Index	Maximum index for DI array
DO_Max_Index	Maximum index for DO array

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.

3.2 Database Values

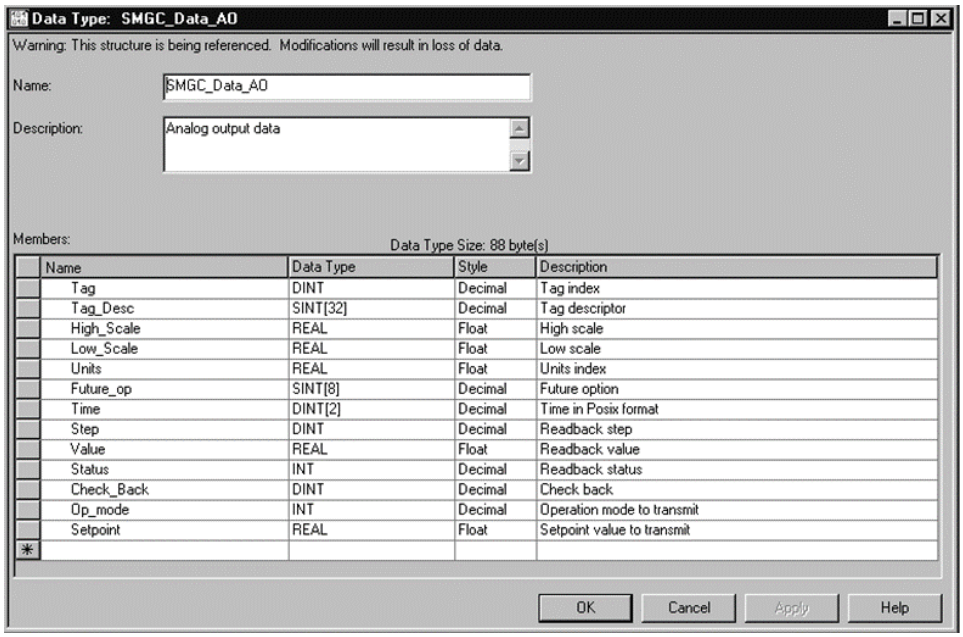
The data values received from the server (response data) and control data sent to the server (request data) are stored in arrays. There is a separate array set up for each of the four data types utilized by the SMGC protocol. Make certain the size of each array is as large as or larger than the value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

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



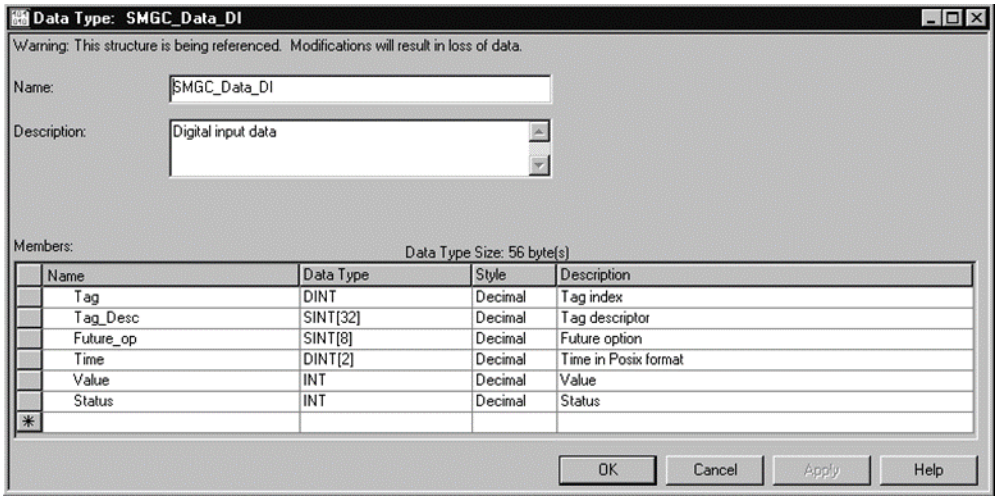
Name	Data Type	Style	Description
Tag	DINT	Decimal	Tag index
Tag_Desc	SINT[32]	Decimal	Tag descriptor
Deadband	REAL	Float	Deadband
High_Scale	REAL	Float	High scale
Low_Scale	REAL	Float	Low scale
Units	REAL	Float	Units index
Future_float	REAL	Float	Future option
Alm_Hys	REAL	Float	Alarm hysteresis
HH_Limit	REAL	Float	High-high limit
H_Limit	REAL	Float	High limit
L_Limt	REAL	Float	Low limit
LL_Limit	REAL	Float	Low-low limit
Future_opt	SINT[8]	Decimal	Future option
Time	DINT[2]	Decimal	Time in Posix format
Value	REAL	Float	Value
Status	INT	Decimal	Status

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

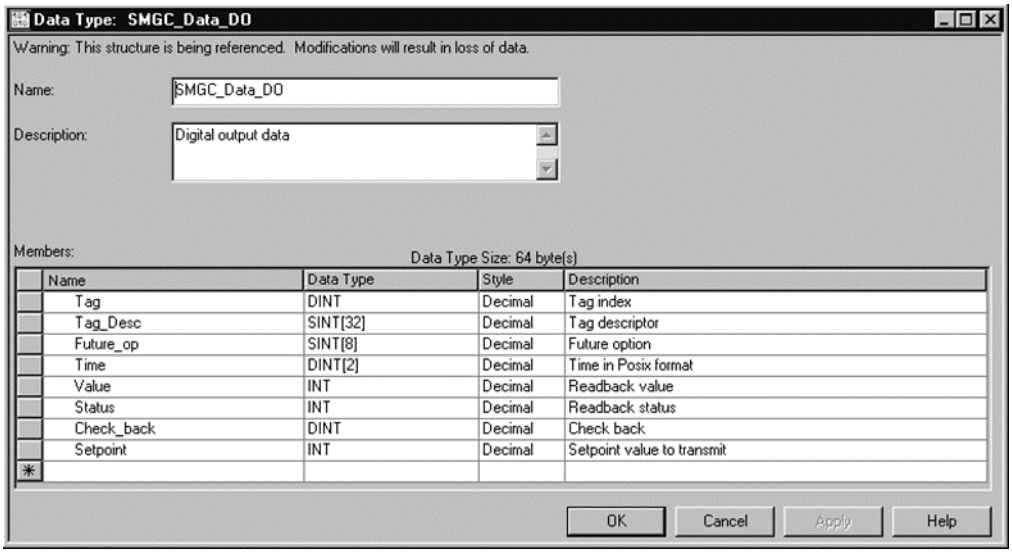


Name	Data Type	Style	Description
Tag	DINT	Decimal	Tag index
Tag_Desc	SINT[32]	Decimal	Tag descriptor
High_Scale	REAL	Float	High scale
Low_Scale	REAL	Float	Low scale
Units	REAL	Float	Units index
Future_op	SINT[8]	Decimal	Future option
Time	DINT[2]	Decimal	Time in Posix format
Step	DINT	Decimal	Readback 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

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



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



3.3 Helper Objects

Other data types can be defined in the ladder logic to help build request messages or to help parse response messages. Since the MVI56E-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 Enhanced FMC722 TPU General Communication Module specification for a complete understanding of each data block that can be transmitted by the protocol.

4 Diagnostics and Troubleshooting

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

- LED status indicators on the front of the module provide information on the module's status.
- Status data contained in the module can be viewed in *ProSoft Configuration Builder* through the Ethernet port.
- Status data values are transferred from the module to the processor.

4.1 Ethernet LED Indicators

The Ethernet LEDs indicate the module's Ethernet port status as follows:

LED	State	Description
Data	OFF	Ethernet connected at 10Mbps duplex speed
	AMBER Solid	Ethernet connected at 100Mbps duplex speed
Link	OFF	No physical network connection is detected. No Ethernet communication is possible. Check wiring and cables.
	GREEN Solid or Blinking	Physical network connection detected. This LED must be ON solid for Ethernet communication to be possible.

4.1.1 Scrolling LED Status Indicators

The scrolling LED display indicates the module's operating status as follows:

Initialization Messages

Code	Message
Boot	Module is initializing
Waiting for Processor Connection	Module is waiting to connect to processor.
MVI56E-SMGC vXX.YY.ZZZ Last Config: <date>	The last date when the module changed its IP address.

Operation Messages

After the initialization step, the following message pattern will be repeated.

```
<Backplane Status> <IP Address> <Backplane Status> <Port Status>
```

Code	Message
<Backplane Status>	OK: Module is communicating with processor ERR: Module is unable to communicate with processor. For this scenario, the <Port Status> message above is replaced with "Processor faulted or is in program mode".
<IP Address>	Module IP address

4.1.2 Non-Scrolling LED Status Indicators

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

LED	Color	Status	Indication
APP	Red or Green	OFF	No connections established over Ethernet
		GREEN	At least one connection established
		RED	During operation, this LED will remain RED in case all connections previously established have been terminated. In case a single connection is established then the LED will be set ON/GREEN. This LED will also be temporarily set as ON/RED during one of the following conditions: The firmware is initializing during startup. The module is rebooting due to a cold boot or warm boot request from the ladder logic or Debug Menu.
OK	Red or Green	OFF	The module is not receiving adequate power or is not securely plugged into the rack.
		GREEN	The module is operating normally
		RED	The module has detected an internal error or is being initialized. If the LED remains RED for over 10 seconds, the module is not working.
ERR	Red or Green	OFF	Not Used

4.2 Clearing a Fault Condition

Typically, if the OK LED on the front of the module remains RED for more than ten seconds, a hardware problem has been detected 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 it is installed correctly.
5. Re-insert the card in the rack and turn the power back on.
6. Verify correct configuration data is being transferred to the module from the ControlLogix controller.
7. If the module's OK LED does not turn GREEN, verify that the module is inserted completely into the rack. If this does not cure the problem, contact ProSoft Technology Technical Support.

4.3 Troubleshooting the LEDs

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

Processor Errors

Problem Description	Steps to take
Processor Fault	Verify the module is securely plugged into the slot that has been configured for the module in the I/O Configuration of RSLogix. Verify the slot location in the rack has been configured correctly in the ladder logic.
Processor I/O LED flashes	This indicates a problem with backplane communications. A problem could exist between the processor and any installed I/O module, not just the MVI56E-SMGC. Verify all modules in the rack are configured correctly.

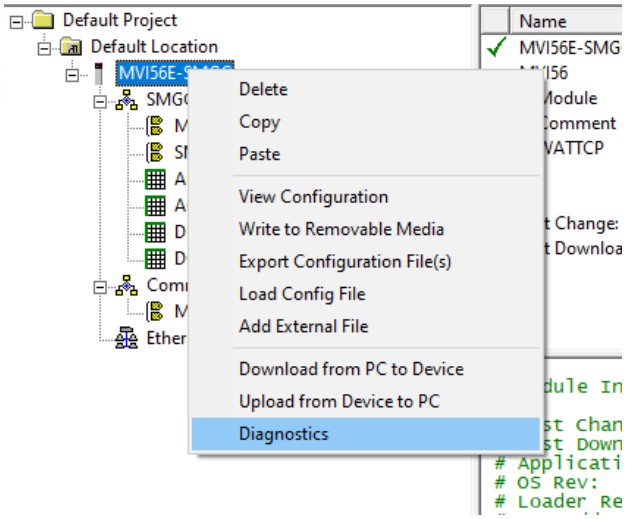
Module Errors

Problem Description	Steps to take
Module Scrolling LED display: <Backplane Status> condition reads ERR	This indicates that backplane transfer operations are failing. Connect to the module's Configuration/Debug port to check this. To establish backplane communications, verify the following items: ▪ The processor is in RUN or REM RUN mode. ▪ The backplane driver is loaded in the module. ▪ The module is configured for read and write data block transfer. ▪ The ladder logic handles all read and write block situations. ▪ The module is properly configured in the processor I/O configuration and ladder logic.
OK LED remains RED	The program has halted or a critical error has occurred. Connect to the communication 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.

4.4 Using the Diagnostics Menu in ProSoft Configuration Builder

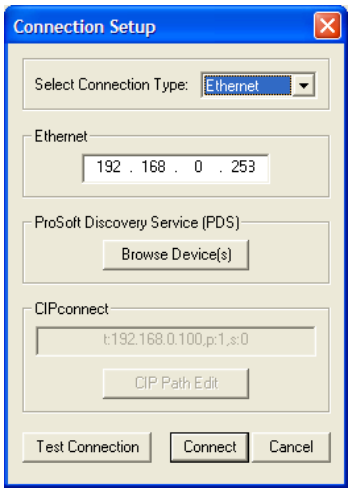
Tip: Multiple ProSoft Configuration Builder *Diagnostics* windows can be open for multiple modules at a time.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** icon to open a shortcut menu.
2. On the shortcut menu, choose **DIAGNOSTICS.**

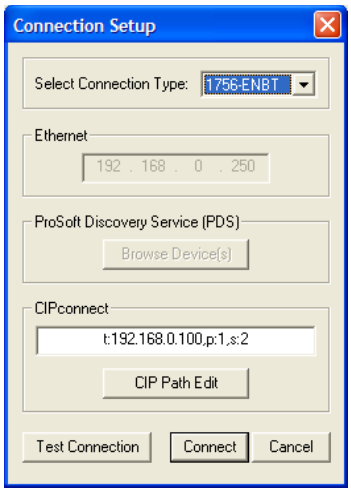


3. In the *Diagnostics* window, click the **SET UP CONNECTION** button to browse for the module's IP address.

4 In the *Connection Setup* dialog box, click the **TEST CONNECTION** button to verify that the module is accessible with the current settings.



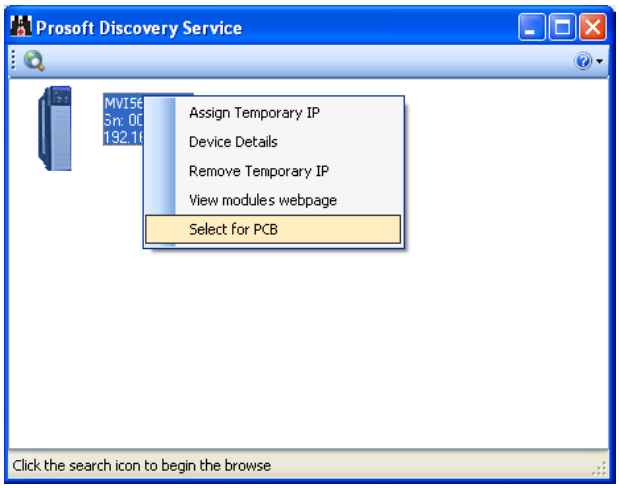
5 You can also use CIPconnect® to connect to the module through a 1756-ENBT card.



6 If the *Test Connection* is successful, click **CONNECT**.

If *PCB* is unable to connect to the module:

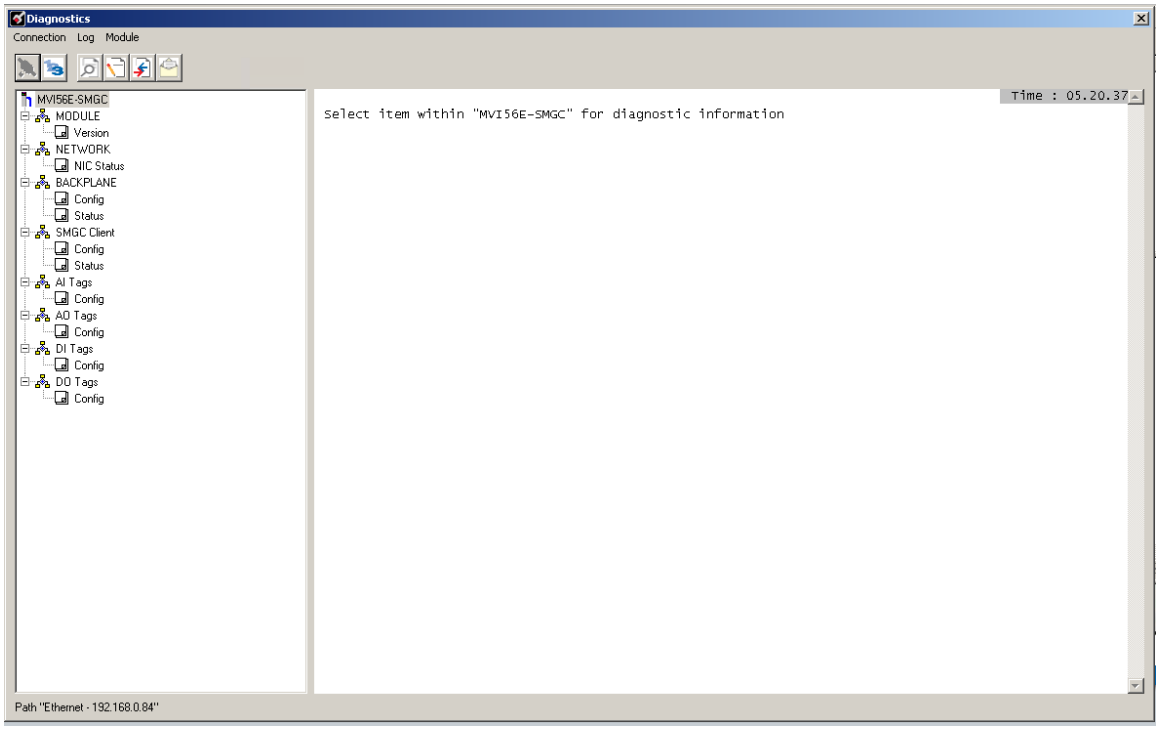
1. Click the **BROWSE DEVICE(S)** button to open the *ProSoft Discovery Service*. Select the module, then right-click and choose **SELECT FOR PCB**.



2. Close *ProSoft Discovery Service*, and click the **CONNECT** button again.
3. If these troubleshooting steps fail, verify that the Ethernet cable is connected properly between your computer and the module, either through a hub or switch (using the grey cable) or directly between your computer and the module (using the red cable).
4. If you are still not able to establish a connection, contact ProSoft Technology for assistance.

4.5 Diagnostics Menu

The Diagnostics menu 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.

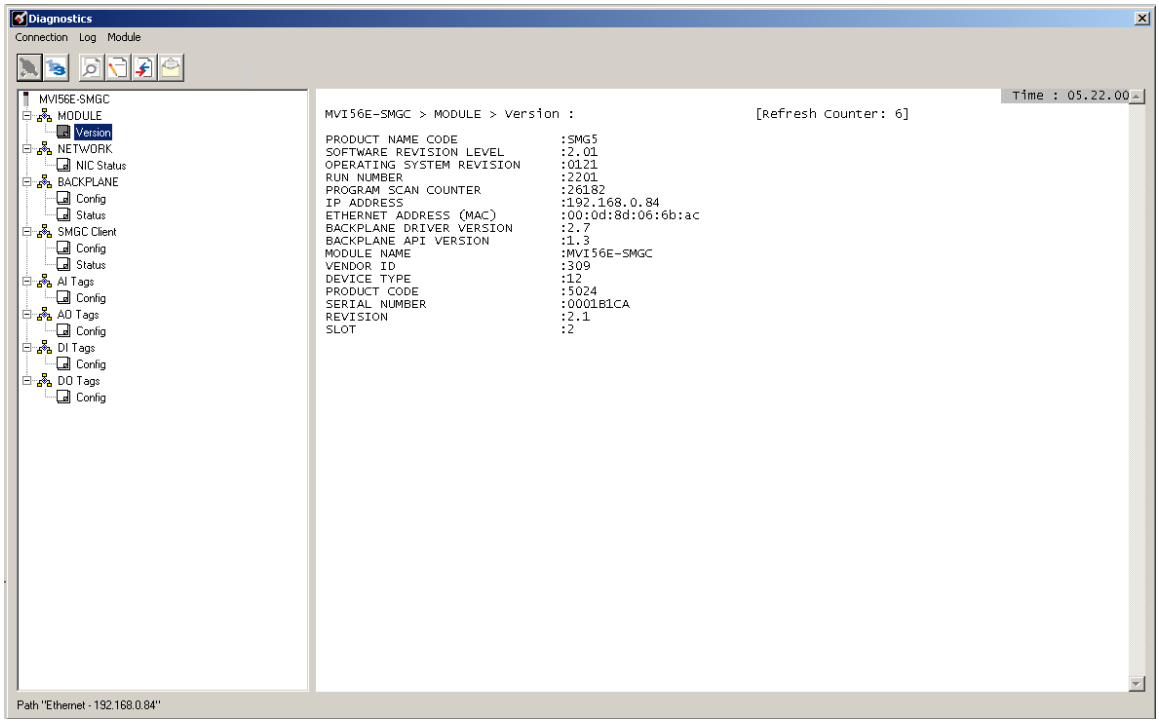


4.5.1 Module Information

Use the *MODULE* menu to view configuration and hardware information for the MVI56E-SMGC module's backplane and Ethernet application port.

4.5.1.1 Version

Use the *Version* menu to view module hardware and firmware information.

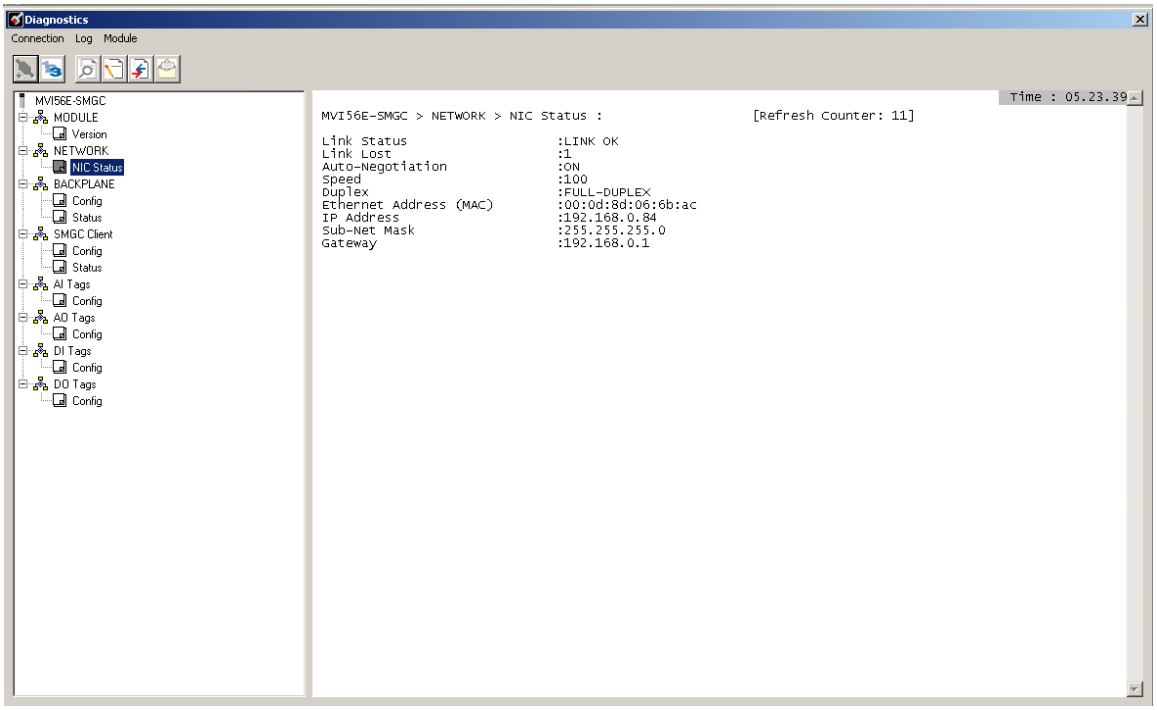


4.5.2 Network Information

4.5.2.1 NIC Status

Use the *NIC Status* (Network Interface Card) menu to view configuration and status information for the MVI56E-SMGC module's Ethernet application port.

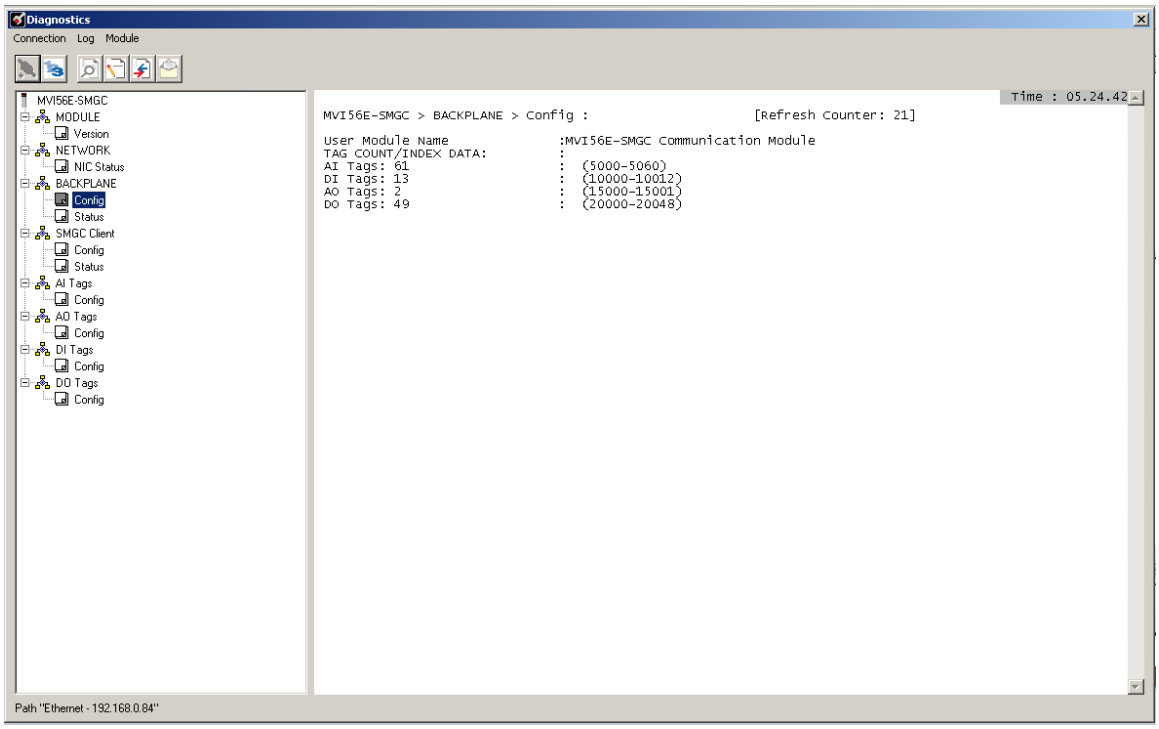
The information on this menu is useful for troubleshooting Ethernet network connectivity problems.



4.5.3 Backplane Information

Use the *BACKPLANE* menu to view the backplane status information for the MVI56E-SMGC module.

4.5.3.1 Configuration



4.5.3.2 Backplane Status

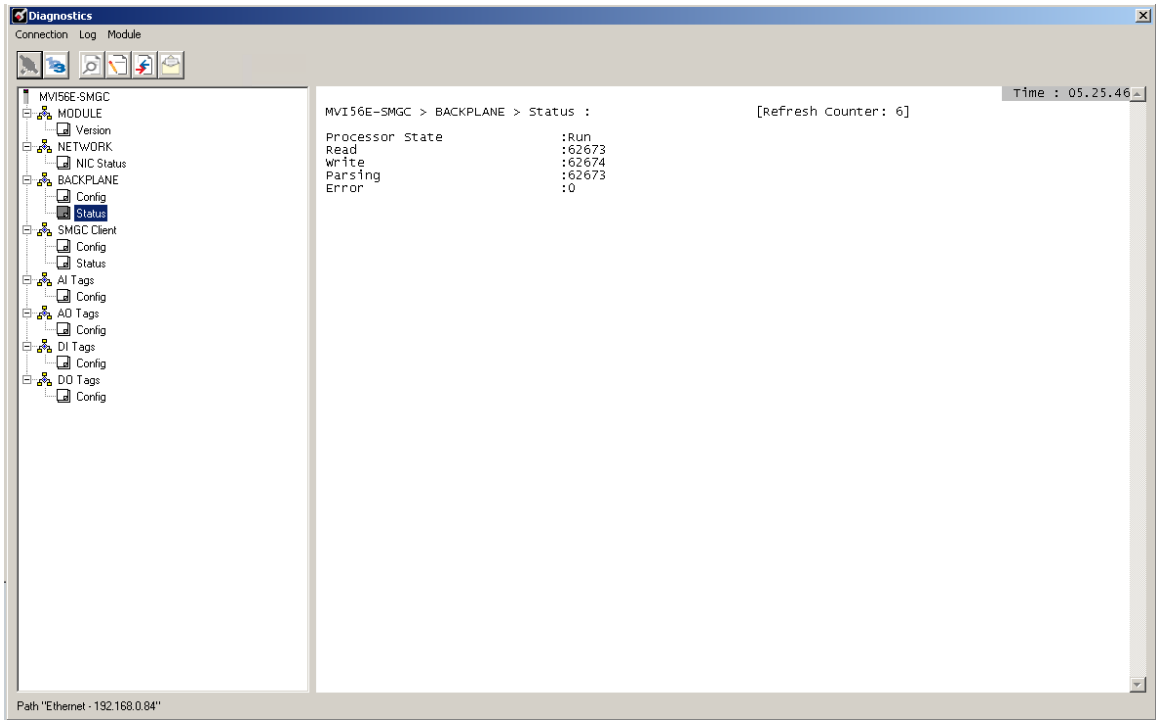
Use the *Status* menu to view current backplane status, including

- Number of retries
- Backplane status
- Fail count
- Number of words read
- Number of words written
- Number of words parsed
- Error count
- Event count
- Command count

During normal operation, the read, write, and parsing values should increment continuously, while the error value should not increment.

The status values on this menu correspond with members of the Status Data Definition.

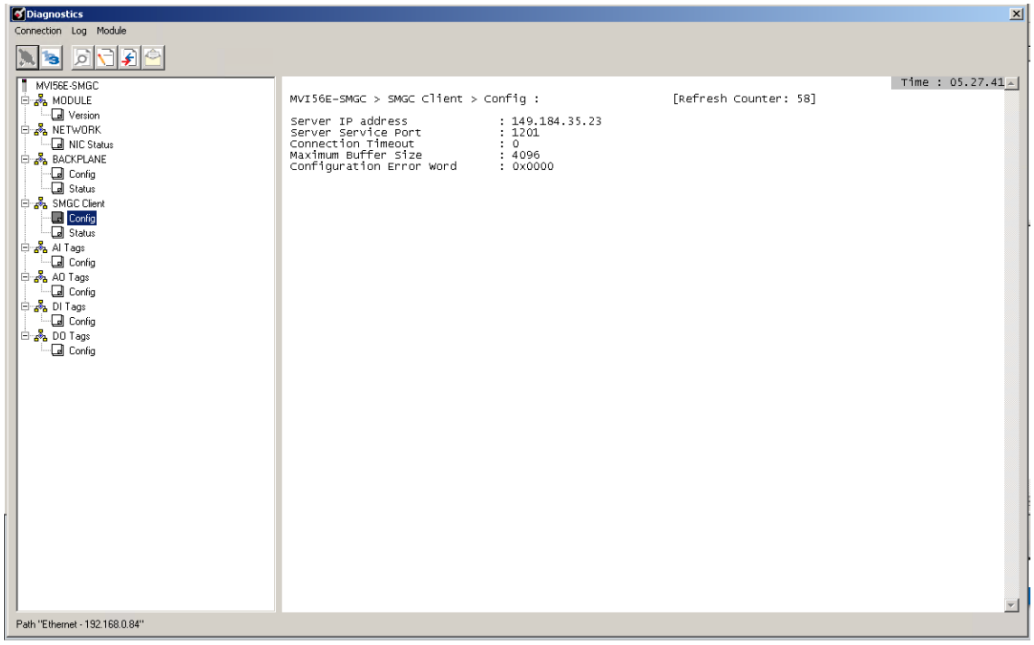
BACKPLANE Status



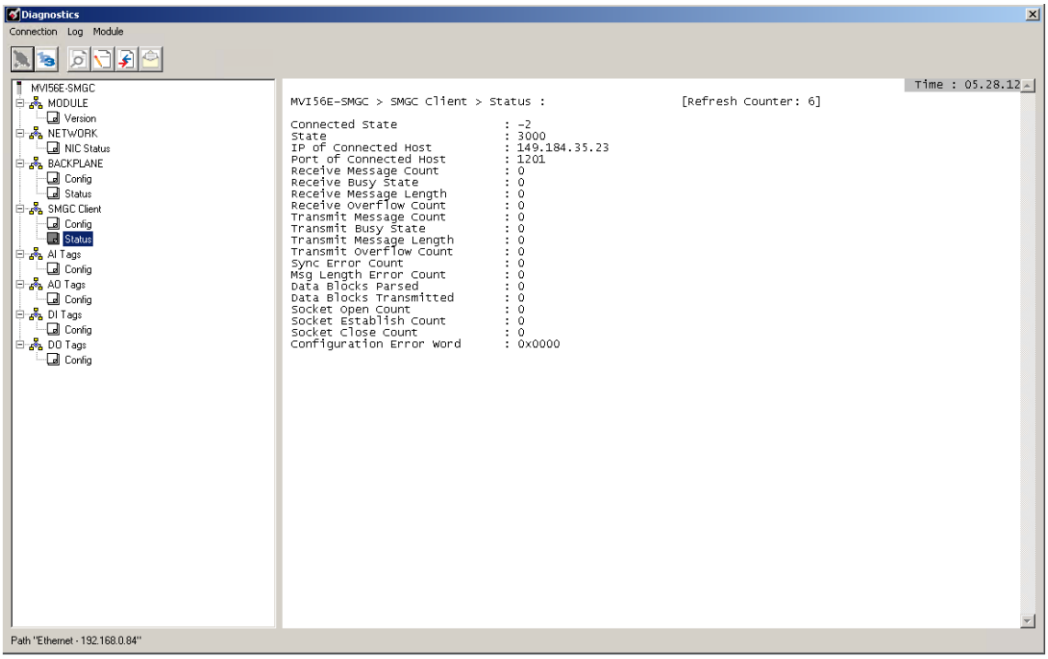
4.5.4 SMGC Client

Use the *SMGC Client* menu to view the SMGC Client configuration and status of the MVI56E-SMGC module.

4.5.4.1 SMGC Client Configuration



4.5.4.2 SMGC Client Status

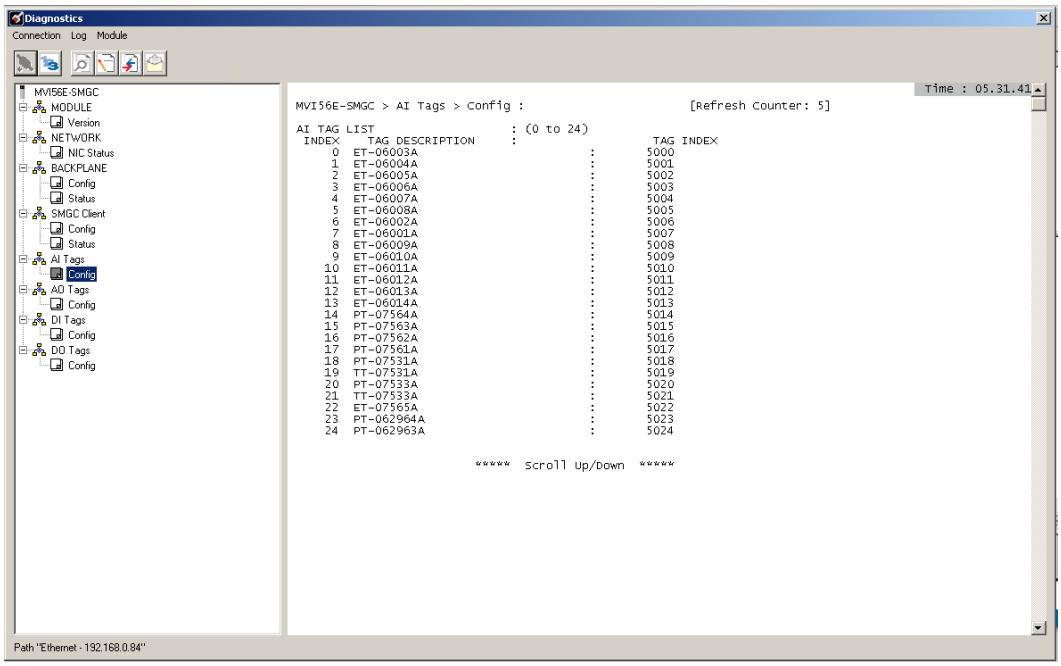


4.6 I/O Tag Configuration

Use the Tag configuration menus to view the configuration for the MVI56E-SMGC client.

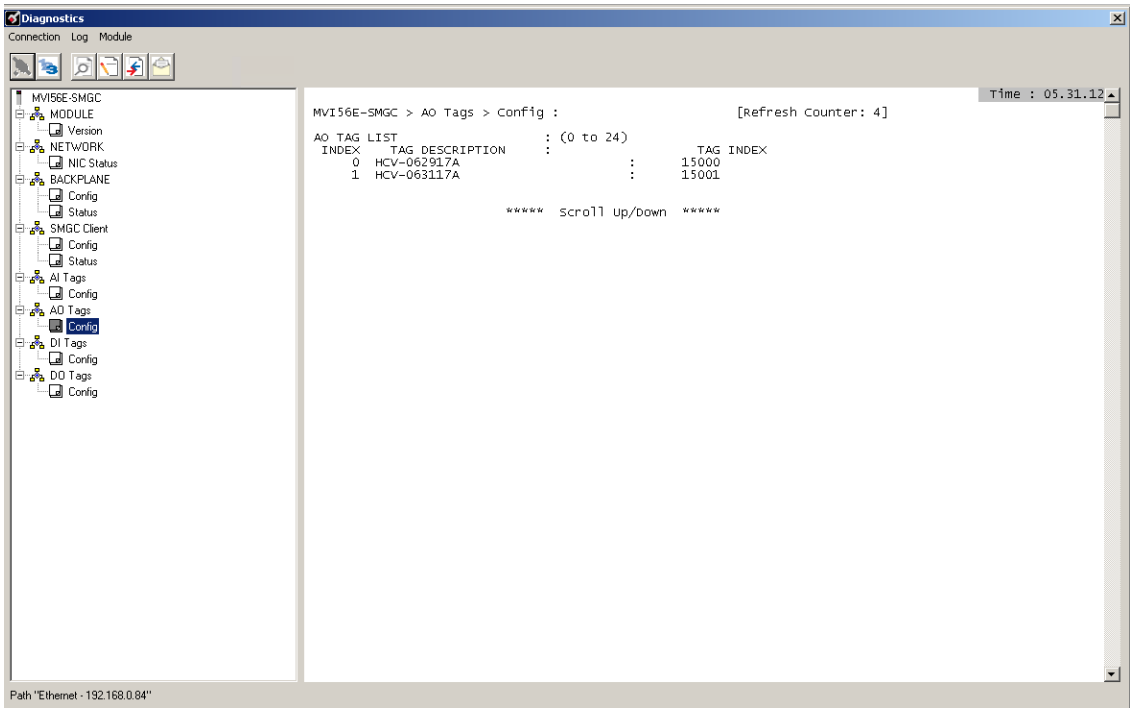
4.6.1 AI Tags

Use the *AI Tags* menu to display the Analog Input tag configuration.



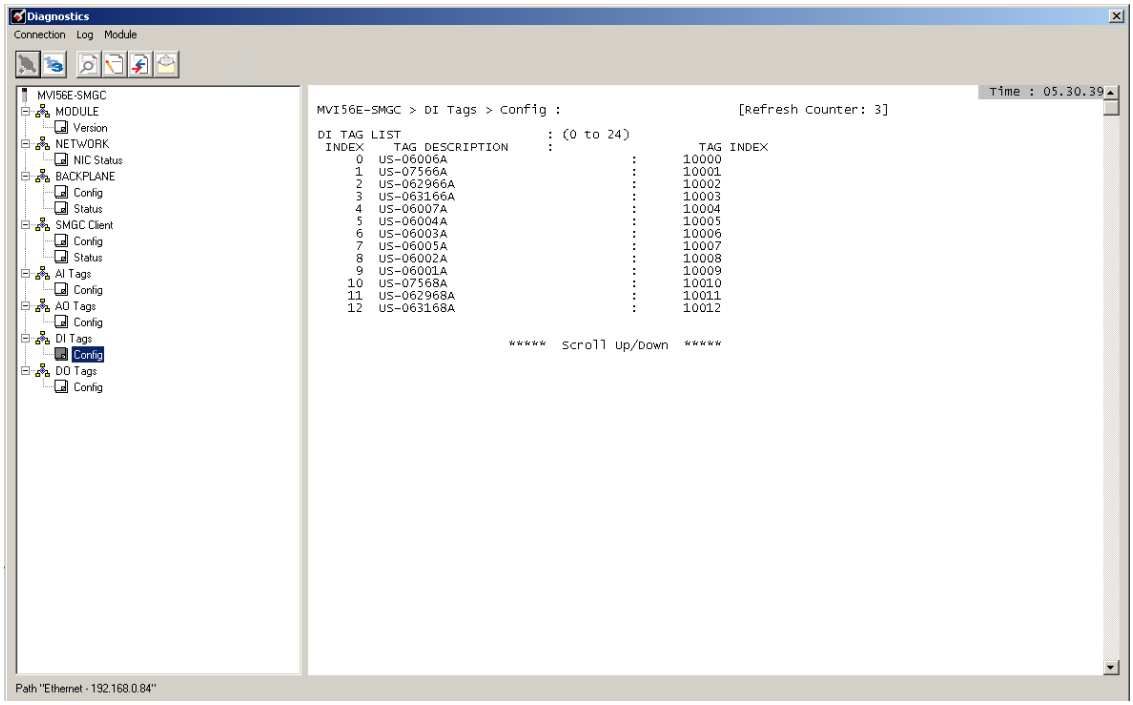
4.6.2 AO Tags

Use the *AO Tags* menu to display the Analog Output tag configuration.



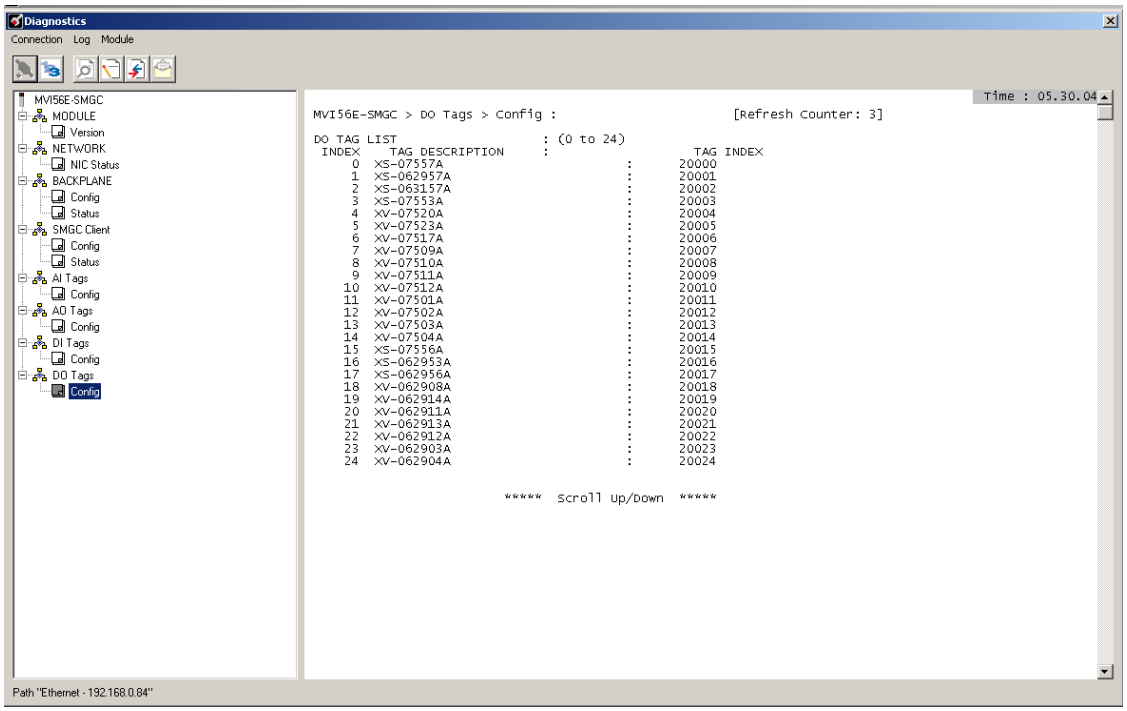
4.6.3 DI Tags

Use the *DI Tags* menu to display the Digital Input tag configuration.



4.6.4 DO Tags

Use the *DO Tags* menu to display the Digital Output tag configuration.



4.7 Reading Status Data from the Module

Module status information is useful for troubleshooting and can be accessed in several different ways. The following is a listing of the data contained in the MVI56E-SMGC *statusdata* object.

Object in SMGCInStat	Block Offset Start	Description
PassCnt	53	Program cycle counter
Product	54	Product name
Rev	56	Revision level
OP	58	Operating system level
Run	60	Run number
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 syncbytes for the start of a message
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.

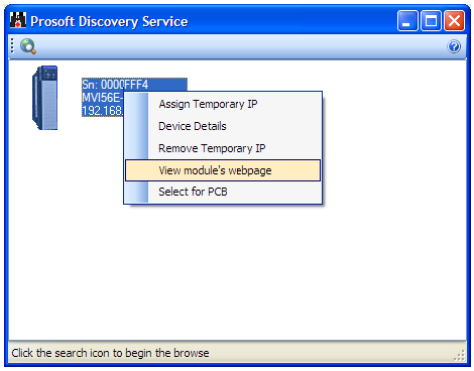
The configuration error word has the following format:

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	

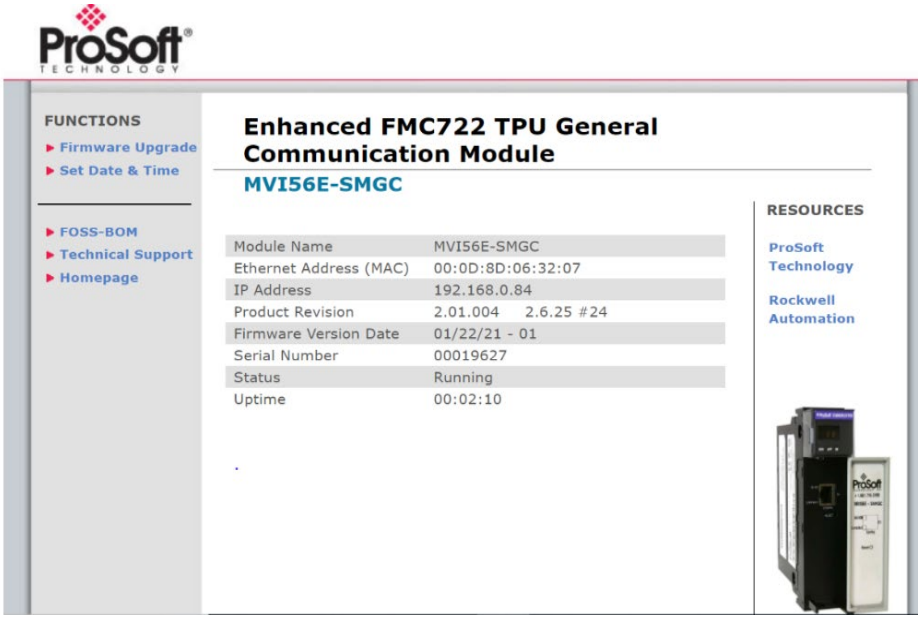
4.8 Connecting to the Module's Webpage

The module's internal webserver provides access to general product information, firmware download link, and links to the ProSoft Technology's website.

1. In the tree view in *ProSoft Configuration Builder*, right-click the **MVI56E-SMGC** icon and then choose **DOWNLOAD FROM PC TO DEVICE**.
2. In the *Download* dialog box, choose the connection type in the *Select Connection Type* dropdown box:
   - Choose **ETHERNET** if you are connecting to the module through the Ethernet cable.
   - Choose **1756 ENBT** if you are connecting to the module through CIPconnect or RSWho.
3. In the *Download files from PC to module* dialog box, click **BROWSE DEVICE(S)**.
4. In *ProSoft Discovery Service*, right-click the **MVI56E-SMGC** icon and then choose **VIEW MODULE'S WEBPAGE**.

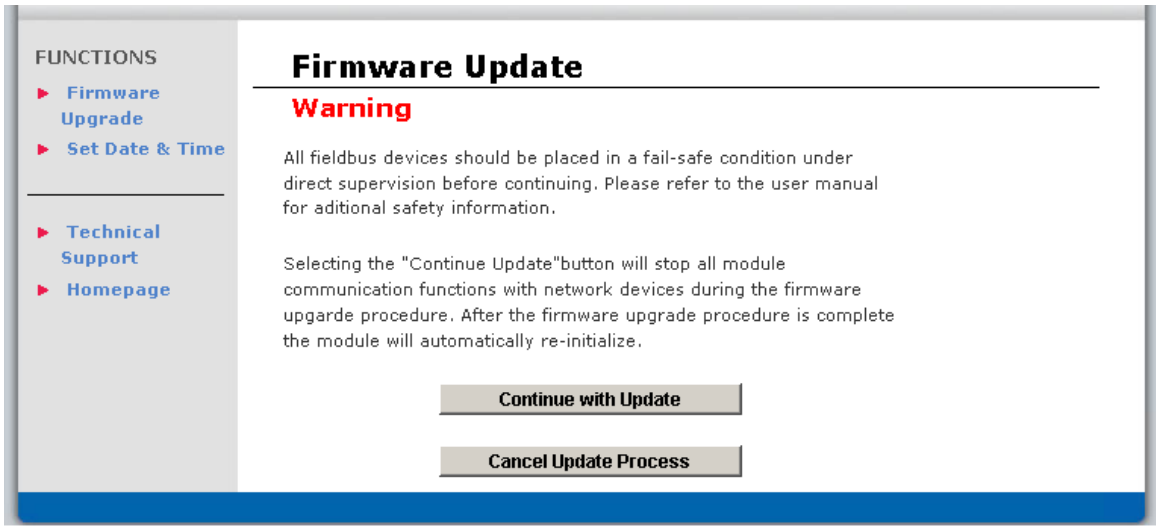


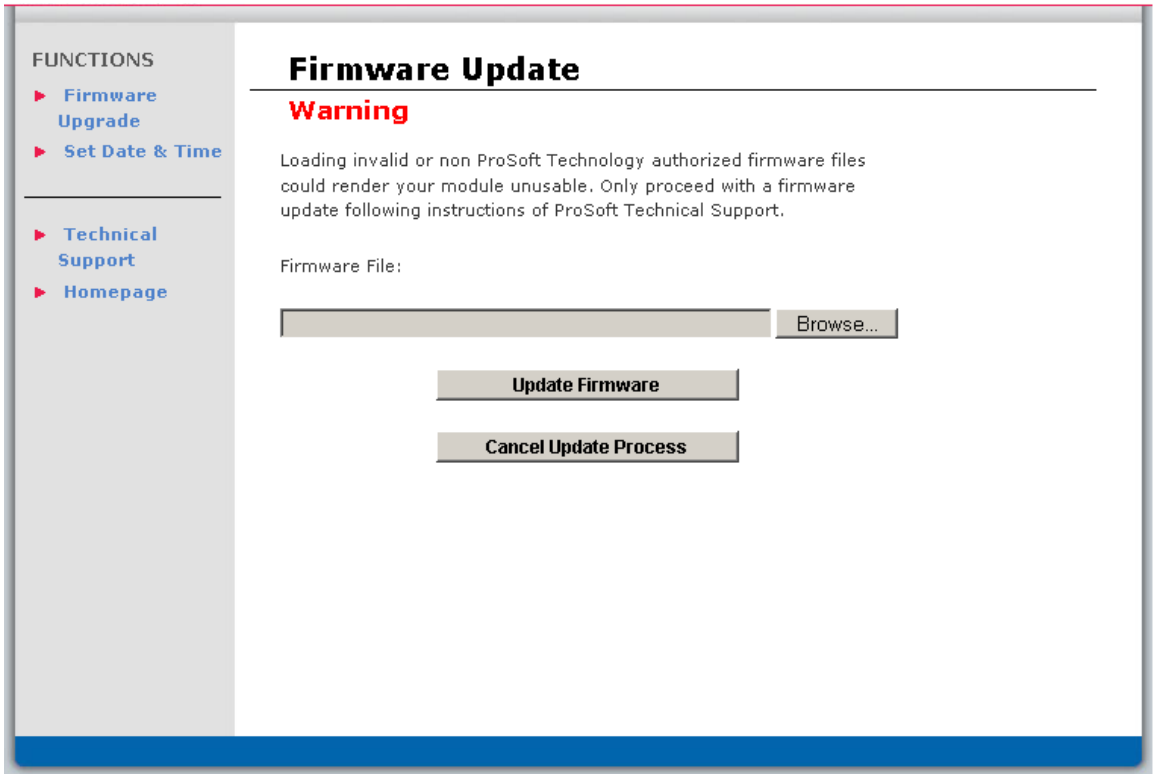
5. This displays the module webpage.



4.8.1 Firmware Upgrade

Firmware upgrade is available through the module's webpage. Follow the prompts to complete the upgrade.





5 Reference

5.1 Product Specifications

The MVI56E-SMGC Enhanced SM General Communication Module allows ControlLogix I/O compatible processors to interface easily with SM General Communication Protocol devices over an Ethernet network.

The MVI56E-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.

5.1.1 Hardware Specifications

Specification	Description
Dimensions	Standard 1756 ControlLogix® single-slot module
Backplane current load	800 mA @ 5 VDC 3 mA @ 24 VDC
Operating temperature	0°C to 60°C (32°F to 140°F)
Storage temperature	-40°C to 85°C (-40°F to 185°F)
Shock	30 g operational 50 g non-operational
Vibration	5 g from 10 to 150 Hz
Relative humidity	5% to 95% (with no condensation)
LED indicators	Battery Status (ERR) Application Status (APP) Module Status (OK)
4-character, scrolling, alphanumeric LED display	Shows module, version, IP, application port setting, port status, and error information
Ethernet port	10/100 Base-T, RJ45 Connector, for CAT5 cable Link and Activity LED indicators Auto-crossover cable detection

5.2 Functional Overview

This section provides an overview of how the MVI56E-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 MVI56E-SMGC module.

5.2.1.1 Module Power Up

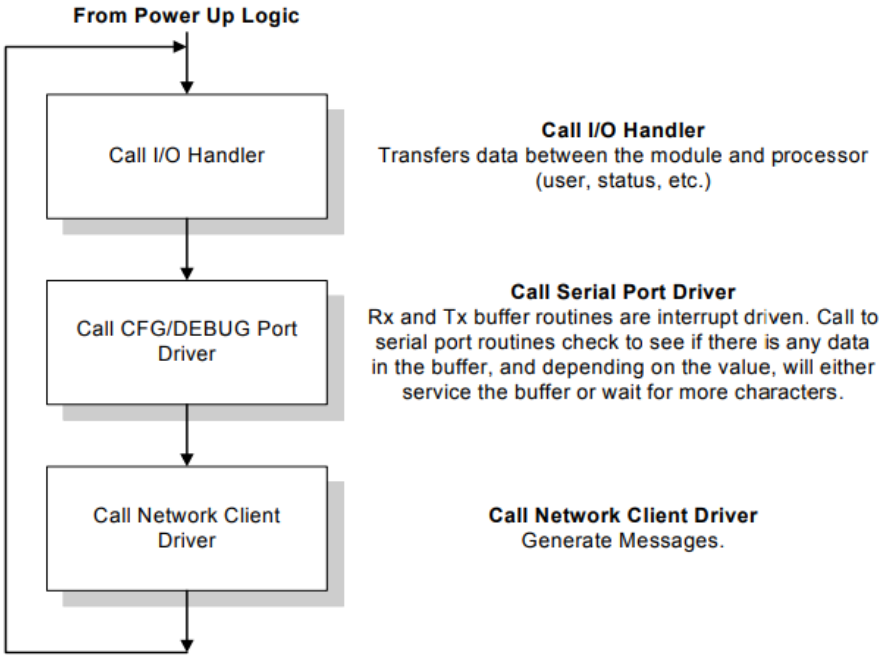
On power up, the module begins performing the following logical functions:

1. Initialize hardware components
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

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.

5.2.1.2 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.



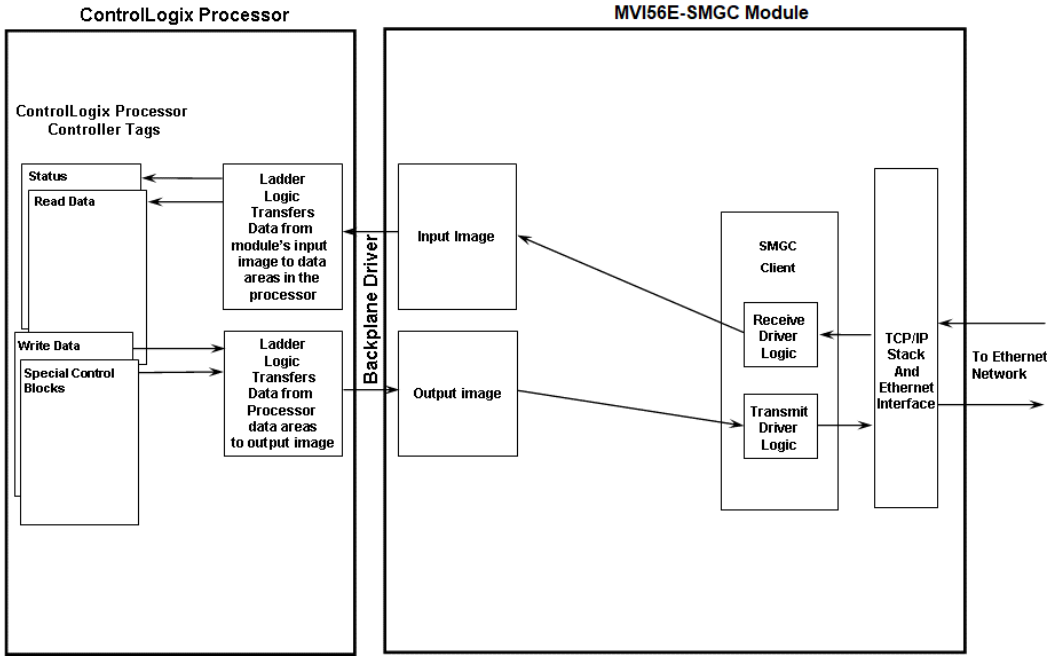
5.2.1.3 Backplane Data Transfer

The MVI56E-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 MVI56E-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.

5.2.1.4 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	Description
-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 sent 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	Description
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	Description
0	No operation to perform
-1	Warm boot module
-2	Cold boot module

5.2.1.5 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	Description
-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.
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 upon connection.
2000	The Delay state after a failure. In this state, the module will delay for 10 seconds before trying to connect to the server.
3000	The client is issuing the ARP command request and waiting for the response.
3001	The client has received the ARP response and try to connect the socket.

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	Description
0	No operation to perform
1	Close the client socket
2	Abor 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	Description
-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.

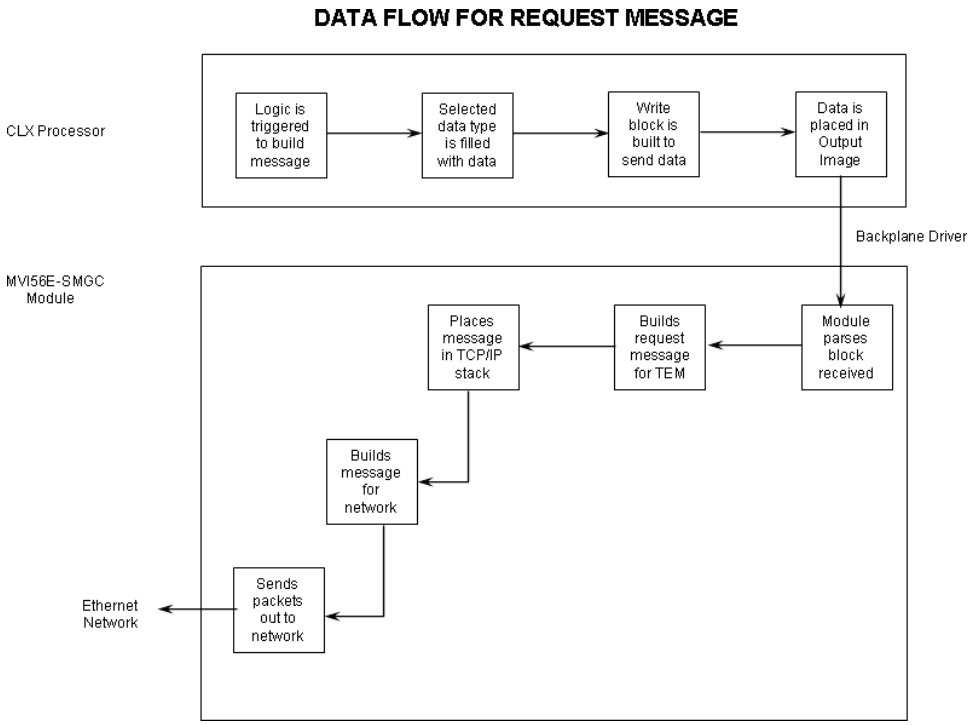
The blocks indicated in the TX column are those transmitted from the module to the server. Ladder logic must be constructed to build these data blocks. These messages are usually sent based on some trigger in the ladder logic and are usually not sent on a continuous basis. The ladder logic builds these 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.

5.2.1.6 Data Flow

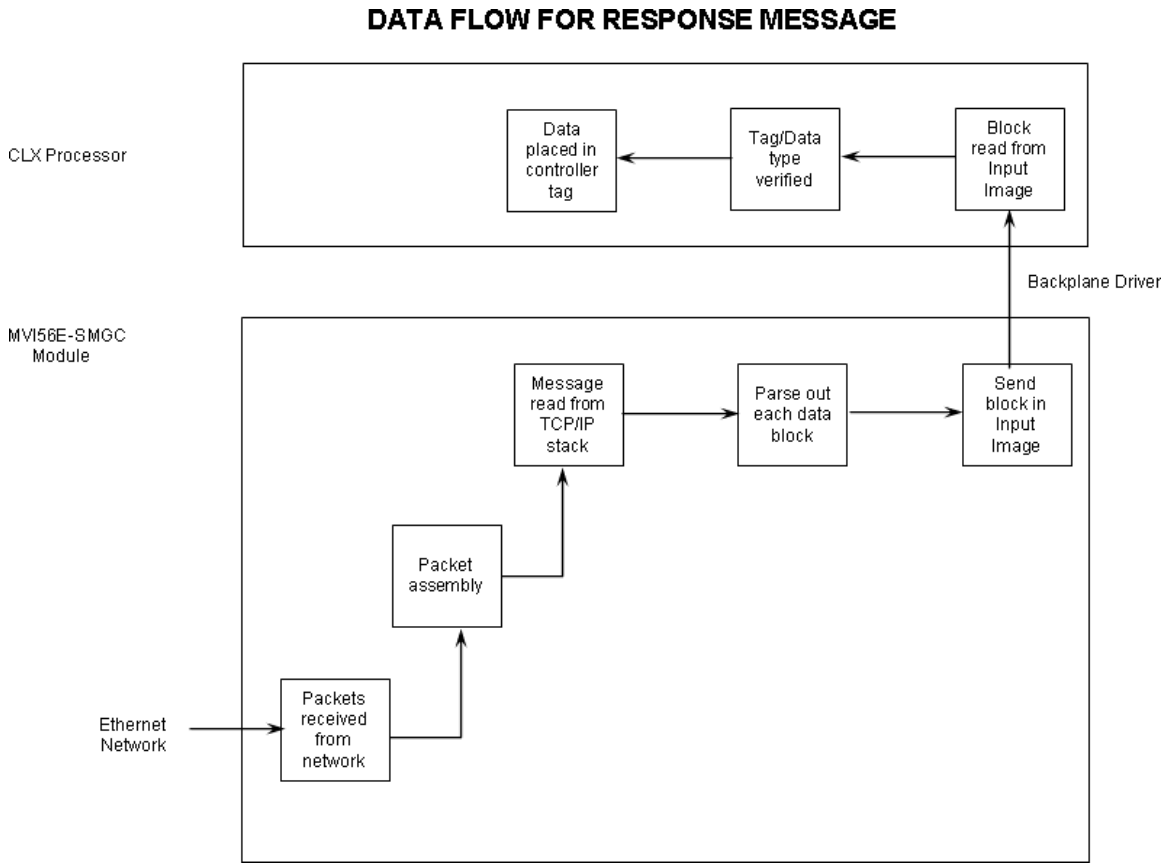
This section describes the flow of data blocks between the ControlLogix processor and the remote server. Ladder logic in the ControlLogix processor is responsible for generating requests and parsing messages received from the server. The parsed data from the received messages are placed in the controller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

5.2.1.7 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
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. Theformat 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
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
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.

5.2.1.8 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 theblock. The module will automatically issue this request after all the tag indexesare 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 thesame 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 ofthe 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

If Tag Index = non-zero, the Tag Index is equal to the tag related to the response error

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

Error Code	Description
0xF0000001	Unknown tag description in set tag index data block
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 arequest. 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 theclocks 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

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
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 Ethernet Cable Specifications

The recommended cable is Category 5 or better. A Category 5 cable has four twisted pairs of wires, which are color-coded and cannot be swapped. The module uses only two of the four pairs.

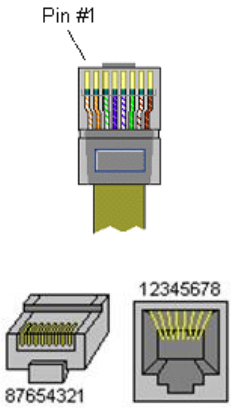
The Ethernet port or ports on the module are Auto-Sensing. You can use either a standard Ethernet straight-through cable or a crossover cable when connecting the module to an Ethernet hub, a 10/100 Base-T Ethernet switch, or directly to a PC. The module detects the cable type and uses the appropriate pins to send and receive Ethernet signals.

Some hubs have one input that can accept either a straight-through or crossover cable, depending on a switch position. In this case, you must ensure that the switch position and cable type agree.

5.3.1 Ethernet Cable Configuration

Note: The standard connector view shown is color-coded for a straight-through cable.

Crossover cable			Straight- through cable	
RJ-45 PIN	**RJ-45 PIN**		**RJ-45 PIN**	**RJ-45 PIN**
1 Rx+	3 Tx+		1 Rx+	1 Tx+
2 Rx-	6 Tx-		2 Rx-	2 Tx-
3 Tx+	1 Rx+		3 Tx+	3 Rx+
6 Tx-	2 Rx-		6 Tx-	6 Rx-



5.3.2 Ethernet Performance

Ethernet performance in the MVI56E-SMGC module can be affected in the following way:

- Accessing the web interface (refreshing the page, downloading files, and so on) may affect performance
- Also, high Ethernet traffic may impact performance, so consider one of these options:
- Use managed switches to reduce traffic coming to module port
- Use CIPconnect for these applications and disconnect the module Ethernet port from the network

5.4 Using the Sample Program

If your processor uses Studio 5000 version 15 or earlier, you will not be able to use the Add-On Instruction for your module. Follow the steps below to obtain and use a sample program for your application.

5.4.1 Opening the Sample Program in Studio 5000

The sample program for your MVI56E-SMGC module includes custom tags, data types and ladder logic for data I/O, status and command control. For most applications, you can run the sample program without modification, or, for advanced applications, you can incorporate the sample program into your existing application.

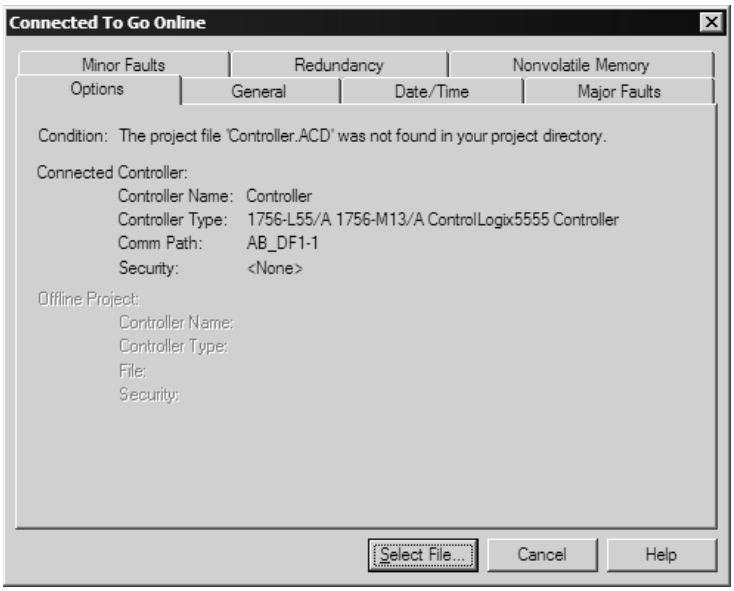
You can always download the latest version of the sample ladder logic and user manuals for the MVI56E-SMGC module from the ProSoft Technology website, at www.prosoft-technology.com

From that link, navigate to the download page for your module and choose the sample program to download for your version of Studio 5000 and your processor.

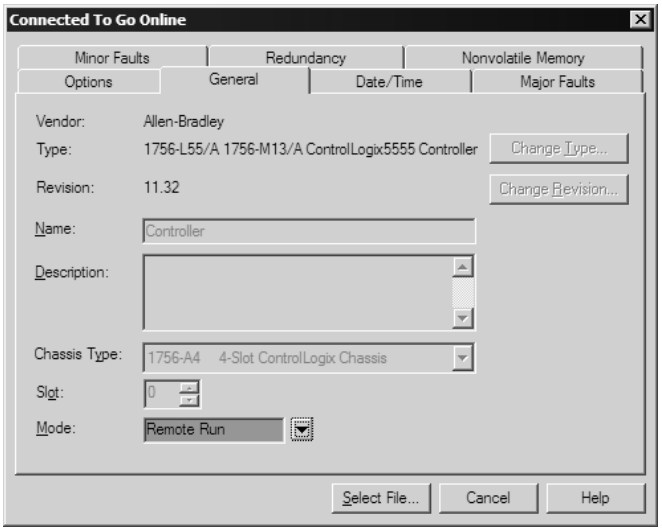
To determine the firmware version of your processor

Important: The RSLinx service must be installed and running on your computer in order for Studio 5000 to communicate with the processor. Refer to your RSLinx and Studio 5000 documentation for help configuring and troubleshooting these applications.

1. Start Studio 5000 and close any existing project that may be loaded.
2. Open the **COMMUNICATIONS** menu and choose **GO ONLINE**. Studio 5000 will establish communication with the processor. This may take a few moments.
3. When Studio 5000 has established communication with the processor, the *Connected To Go Online* dialog box will open.



4 In the *Connected To Go Online* dialog box, click the **GENERAL** tab. This tab shows information about the processor, including the Revision (firmware) version. In the following illustration, the firmware version is 11.32

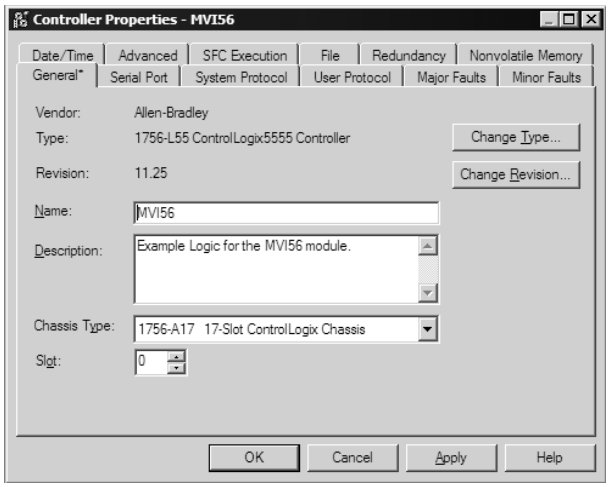


5 Select the sample ladder logic file for your firmware version.

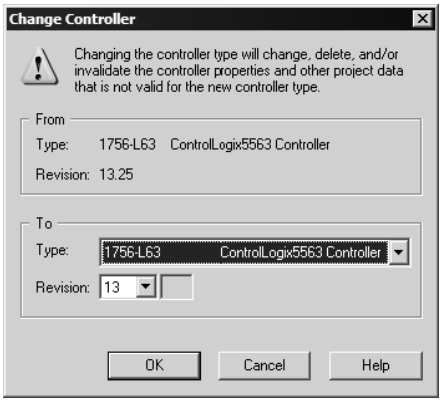
5.4.2 Selecting the Controller Type

The sample application is for a 1756-L63 ControlLogix 5563 Controller. If you are using a different model of the ControlLogix processor, you must configure the sample program to use the correct processor model.

1. In the *Controller Organizer* list, right-click the folder for the controller and then choose **PROPERTIES.** This action opens the *Controller Properties* dialog box.



2. Click the **CHANGE TYPE** or **CHANGE CONTROLLER** button. This action opens the *Change Controller* dialog box.



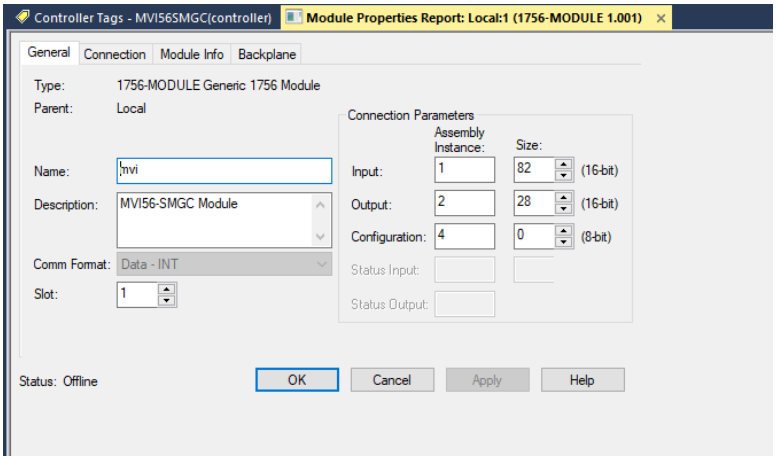
3. Open the **TYPE** dropdown list, and then select your ControlLogix controller.
4. Select the correct firmware revision for your controller, if necessary.
5. Click **OK** to save your changes and return to the previous window.

5.4.3 Selecting the Slot Number for the Module

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

To change the slot number

1. In the *Controller Organizer* list, right-click the module and then choose **PROPERTIES.** This action opens the *Module Properties* dialog box.

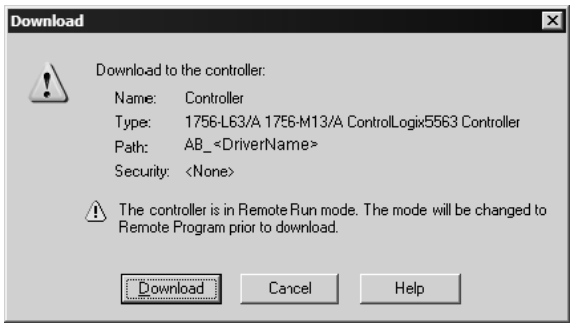


2. In the **SLOT** field, use the up and down arrows on the right side of the field to select the slot number where the module will reside in the rack, and then click **OK.**
3. RSLogix will automatically apply the slot number change to all tags, variables and ladder logic rungs that use the MVI56E-SMGC slot number for computation.

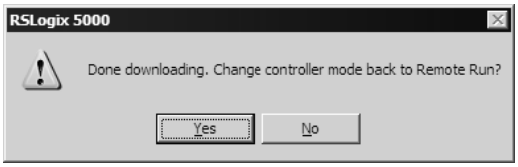
5.4.4 Downloading the Sample Program to the Processor

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.** Studio 5000 will establish communication with the processor.
2. When communication is established, Studio 5000 will open a confirmation dialog box. Click the **DOWNLOAD** button to transfer the sample program to the processor.



3. Studio 5000 will compile the program and transfer it to the processor. This process may take a few minutes.
4. When the download is complete, Studio 5000 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 Studio 5000 documentation to interpret and correct the error.

5.4.5 Adding the Sample Ladder to an Existing Application

1. Copy the Controller Tags from the sample program.
2. Copy the User-Defined Data Types from the sample program.
3. Copy the Ladder Rungs from the sample program.
4. Save and Download the new application to the controller and place the processor in RUN mode.

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

Note: For technical support calls within the United States, a 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