molex
one company › a world of innovation

Brad™ SST Ethernet and Serial Communication Module for ControlLogix

SST-ESR2-CLX-RLL & SST-SR4-CLX-RLL

User Reference Guide

Document Edition: 2.3

Document #: 715-0097

Document Edition: 2.3

Date: January 2, 2014

This document applies to the SST-ESR2-CLX-RLL & SST-SR4-CLX-RLL Modules.

At Molex, we strive to ensure accuracy in our documentation. However, due to rapidly evolving products, software or hardware changes occasionally may not be reflected in our documents. If you notice any inaccuracies, please contact us (see Appendix F).

Molex Canada Ltd.
Industrial Products Business Unit
Integrated Products Division
216 Bathurst Drive
Waterloo, Ontario, Canada N2V 2L7

Hardcopies are not controlled.

Contents

Preface

Preface Sections:

- Purpose of this Guide
- Special Notation

Purpose of this Guide

This document is the User's Reference Guide for the SST ControlLogix (CLX) Ethernet/Serial Communication modules that comes in two variants:

- Ethernet / Serial Communication Modules (SST-ESR2-CLX-RLL)
- Serial Communication Module (SST-SR4-CLX-RLL)

The use of this guide will be beneficial to people responsible in the installation, programming or troubleshooting of Control Systems that uses Allen-Bradley CLX Processors.

This document contains helpful information in the installation, configuration and troubleshooting of either SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL communication modules in the Control System. It covers minimal information regarding the use and configuration of PLC's. It is assumed that user has the required knowledge to troubleshoot problems related to PLCs.

Other related document that applies to the configuration of the communication modules supported protocols is mentioned in this document. User will be directed to use the related document when applicable.

It is assumed that user has a basic understanding of the protocols that the modules will be used for.

Special Notation

The following special notations may be used throughout this guide:

Warning

Warning messages alert the reader to situations where personal injury may result. Warnings are accompanied by the symbol shown, and precede the topic to which they refer.

Caution

Caution messages alert the reader to situations where equipment damage may result. Cautions are accompanied by the symbol shown, and precede the topic to which they refer.

Note

A note provides additional information, emphasizes a point, or gives a tip for easier operation. Notes are accompanied by the symbol shown, and follow the text to which they refer.

1

System Requirements

Chapter Sections:

- Hardware Requirements
- Power Requirements
- Operating System Requirements
- Other Requirements

1.1 PC Hardware Requirements

- Intel Pentium 4 or equivalent processor minimum recommended
- 512 MB of RAM minimum recommended; more memory improves performance
- Minimum 100MB of Free Hard disk space
- 17”/19” 1280 x 1024 or higher –resolution monitor minimum recommended

  (XVGA 1024 x 768 minimum supported)
- CD-ROM drive or DVD-ROM drive
- Power cable, Ethernet cable, keyboard, mouse

1.2 PAC/PLC Power Requirements

Review your system’s power requirements to ensure that your chassis supports placement of the Module.

Note

The ESR2 consumes 850 mA @ 5VDC, and the SR4 consumes 1004mA @ 5VDC. Both consume 1.75 mA @ 24VDC.

For modular systems, calculate the total load on the power supply using the procedure described in the CLX 5000 Modular Style Installation & Operation Manual, Allen-Bradley Publication 1747-6.2.

1.3 PC Operating System Requirements

The Communication module configuration and diagnostic tools are tested to run in the following operating system only:

- Windows XP
- Windows Vista
- Windows 7
- Windows 8

1.4 Other Requirements

1.4.1 Rockwell Automation RSLogix 5000 Add-On-Instruction support Requirements

To use the communication module's AOI support it requires the following:

- Rockwell RSLogix 5000 Version 16 or later
- The following are available on the SST Backplane Communication Module Products CD v3.1or higher and from our website www.molex.com
- RSLogix 5000 Ladder sample code using AOI (Created in RSLogix 5000 Version 16)
    - Sample code using Generic Profile
        - CLX2000_Paging_AOI_using_Generic.L5X
        - CLX2000_Paging_AOI_using_Generic.ACD
        - CLX2000_Paging_and_Cyclic_Function_AOI_using_Generic.L5X
        - CLX2000_Paging_and_Cyclic_Function_AOI_using_Generic.ACD
    - Sample code using Add-On-Profile (AOP)
        - For SST-ESR2-CLX
            - CLX2000_Paging_AOI_using_AOP.ACD
            - CLX2000_Paging_AOI_using_AOP.L5X
            - Paging_And_Cyclic_Function_AOI_AOP.ACD
            - Paging_And_Cyclic_Function_AOI_AOP.L5X
        - For SST-SR4-CLX
            - CLX2000_Paging_AOI_for_SR4 using_AOP.ACD
            - CLX2000_Paging_AOI for SR4 using AOP.L5X
            - Paging_And_Cyclic_Function_AOI_SR4_AOP.ACD
            - Paging_And_Cyclic_Function_AOI_SR4_AOP.L5X

The module's firmware version must be 2.10.2 or higher to use the modules's AOI with Generic Profile. If using the modules's AOI with AOP profile then modoule's firmware 2.12.1.0 or higher is required. To check the firmware version, unplug and plug the module into the ControlLogix Rack, the module's LCD will display “BOOT” and shortly after will scroll the firmware version. If version is lower than 2.10.2, contact our technical support. Support information can be found in Support Section.

The BCMS (Brad Communication ™ SST Backplane Communication Module Software (BCMS) Installation version must be 1.9 or higher. The software installation package version can be verified from these locations:

Control Panel -> Program and Features if system is running Windows 7 OS.
Control Panel -> Add or Remove Programs if system is running Windows XP OS.

1.4.2 Rockwell Automation RSLogix 5000 Add-On-Profile support Requirements

To use the communication module's AOP support it requires the following:

- Rockwell RSLogix 5000 Version 15 or later

Note

The module's AOP only supports maximum input, output and status INT sizes (250, 248, 250). Use 1756-Module Generic Profile if smaller connections sizes are required.

The following are available on the SST Backplane Communication Module Products CD and from our website Molex Support and Download

- Rockwell's RSLogix 5000 AOP installation 1.21.1.0
- RSLogix 5000 Ladder sample code using AOP (Created in RSLogix 5000 Version 15)
    - Read_Write3048Words_v15_AOP.ACD
    - READ_WRITE_CYCLIC_FUNCTION_v15_AOP.ACD
    - GET_SET_SAMPLE_LADDER_AOP.ACD

The module's firmware version must be 2.3.0 or higher. To check the firmware version, unplug and plug the module into the ControlLogix Rack, the module's LCD will display "BOOT" and shortly after will scroll the firmware version. If version is lower than 2.3.0, contact our technical support. Support information can be found in Support Section.

The BCMS (Brad Communication ™ SST Backplane Communication Module Software (BCMS) CD version must be v1.7.0 or higher. The software installation package version can be verified from these locations:

Control Panel -> Program and Features if system is running Windows 7 OS.

Control Panel -> Add or Remove Programs if system is running Windows XP OS.

2

Package Description

Chapter Sections:

- Package Contents
- Hardware Features

2.1 Package Contents

Unpack the communications Module. Make sure that the contents include:

Table 1: SST-ESR2-CLX-RLL Package Contents

Qty	Package Content for SST-ESR2-CLX-RLL	Description
1	SST-ESR2-CLX-RLL	Communication Module
2	RJ45 Serial cable adapters to Db9 format	Connector to use for Modbus
1	CD with install package for Windows XP/VISTA/Win7 OS	Brad Communication ™ SST Backplane Communication Module Software (BCMS)
N/A	PDF version of this User Reference Guide	Included in the CD
N/A	PDF version of Siemens Industrial Ethernet Protocol Reference Guide	Included in the CD
N/A	PDF version of Modbus TCP Reference Guide	Included in the CD
N/A	PDF version of Modbus Serial Reference Guide	Included in the CD

Table 2: SST-SR4-RLL-CLX Package Contents

Qty	Package Content for SST-SR4-CLX-RLL	Description
1	SST-SR4-CLX-RLL	Communication Module
4	RJ45 Serial cable adapters to Db9 format	Connector to use for Modbus
1	CD with install package for Windows XP/VISTA/Win7 OS	Brad Communication ™ SST Backplane Communication Module Software (BCMS)
N/A	PDF version of this User Reference Guide	Included in the CD
N/A	PDF version of Siemens Industrial Ethernet Protocol Reference Guide	Included in the CD (not used)
N/A	PDF version of Modbus TCP Reference Guide	Included in the CD (not used)
N/A	PDF version of Modbus Serial Reference Guide	Included in the CD

2.2 Hardware Features

The LCD displays the Module’s Backplane firmware version at power-up (for 3 seconds), and connection status and errors during operation. The three LEDs display Module behavior.

Figure 1: Module Hardware Features

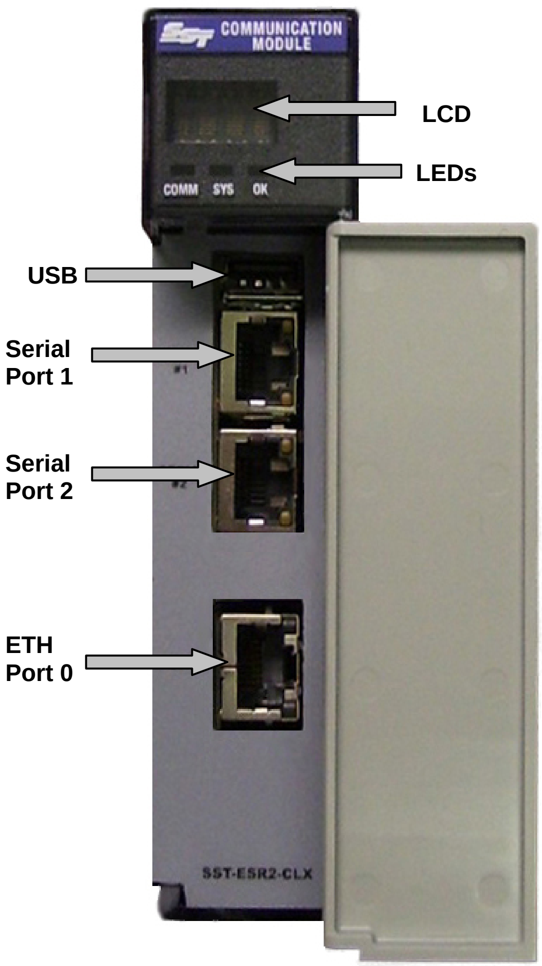
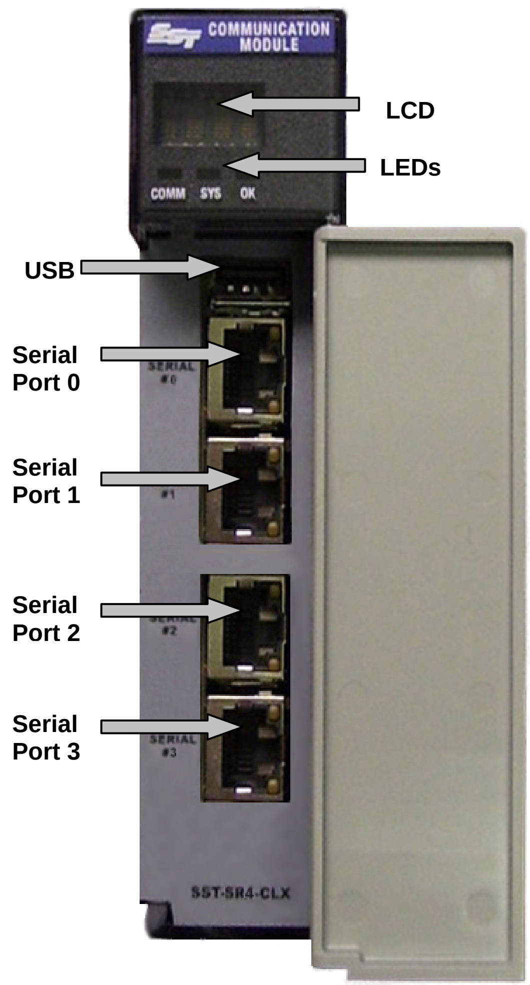
SST-ESR2-CLX-RLL	SST-SR4-CLX-RLL
	

Table 3: Description of Features

Feature	Description
Status LEDs	Display the communication and system status
Front label	Identifies the Module
RJ45 Serial port connector	For connecting to an RS232, RS422, or RS485 serial network
RJ45 Ethernet Connector	For connecting to the Ethernet network
Self-locking tabs	Secure the Module in the chassis slot
Side label (nameplate)	Provides Module information
Display	CLX Connection status faults, operation status IP Address and configuration name are displayed continuously during operation on SST-ESR2-CLX Configuration name is displayed continuously during operation on SST-SR4-CLX
USB configuration port	For downloading I/O configuration data and upgrading firmware

2.2.1 Status LEDs

There are three LEDs on the Module: the COMM LED, the SYS LED and the OK LED.
For detailed information, refer to LED and Display States section.

COMM LED

The COMM LED indicates the health of the Ethernet / Serial communications channels.
It turns green for 75ms when the Module has processed a valid cyclic function, and switches to red for 75ms when there are cyclic function errors on one or more channels. If no configured or active cyclic functions are present, the LED stays off. The LED is solid red if there is a configuration error.

SYS LED

The SYS LED indicates the health of the connection’s status, whether the Module is connected or the connection has been lost. At power-up, the SYS LED is amber until the first connection is made, at which point it turns green. If the connection is lost, the LED will turn red until the connection has been re-established.

OK LED

The OK LED indicates that initialization is complete and that the Module is OK. Green indicates that the Module has passed initialization, and if the LED is flashing, there has been a fatal error and an error code should be displaying. If after start up the LED is solid red, the Module has gone into a watchdog failure state and has ceased all communications with the CLX CPU and any network devices.

2.2.2 RJ45 Serial Connectors

The RJ45 Serial connectors are used to connect RS232, RS422, add RS485 devices to the Module. Their configuration depends on the configured protocols.

2.2.2.1 Serial Connectors

The top yellow LED blinks whenever a character is received, and the bottom green LED blinks every time the Module transmits.

2.2.3 RJ45 Ethernet Connector

The RJ45 Ethernet Connector connects to the configured industrial network.

2.2.3.1 Ethernet Connector LEDs

The top green LED indicates activity, and the bottom orange LED indicates that the link has been established.

2.2.4 USB Configuration Port

The USB Configuration port is used as an option to download configurations/firmware to the module.

3

Product Software

Chapter Sections:

- Installation Package
- Brad Communication ™ SST Backplane Communication Module Software (BCMS)
- Module’s Configuration Directory Structure

3.1 Installation Package

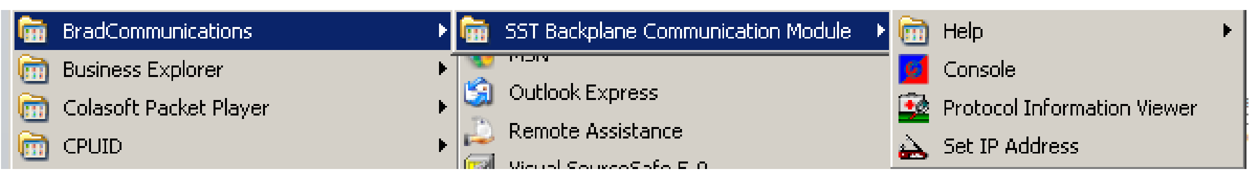
Installation package is available either from the CD or from our website. The CD always comes with the latest installation package. However, there are software updates that are released through our website and may not be available yet on CD. We recommend checking the website first before installing the software.

Note

Before installing a new version of the software, ensure that you uninstall any previous Backplane Console and reboot your PC.

From the Windows XP/Vista/7 installation CD, run the setup.exe file. This script has an option to install a set of tools for use with SST Backplane Modules. This option should be used to install the Configuration Tool and associated software.

After successful installation, go to All Programs and BradCommunications will be available as shown below.



The package will install the following:

- Help files
  - Modbus Ethernet TCP/IP Protocol Reference Guide
  - Modbus RTU/ASCII Master/Slave Reference Guide
  - Siemens Industrial Ethernet Protocol Reference Guide
  - SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL Reference Guide
- BCMS – (Brad Communication ™ SST Backplane Communication Module Software) is covered briefly in the next section.
- Sample ladder code projects.

3.2 Introduction to Brad Communication ™ SST Backplane Communication Module Software (BCMS)

The Brad Communication ™ SST Backplane Communication Module Software is comprised of multiple applications used to configure, connect and diagnose the communication module. This section briefly discuss the different components available from the BCMS.

Note

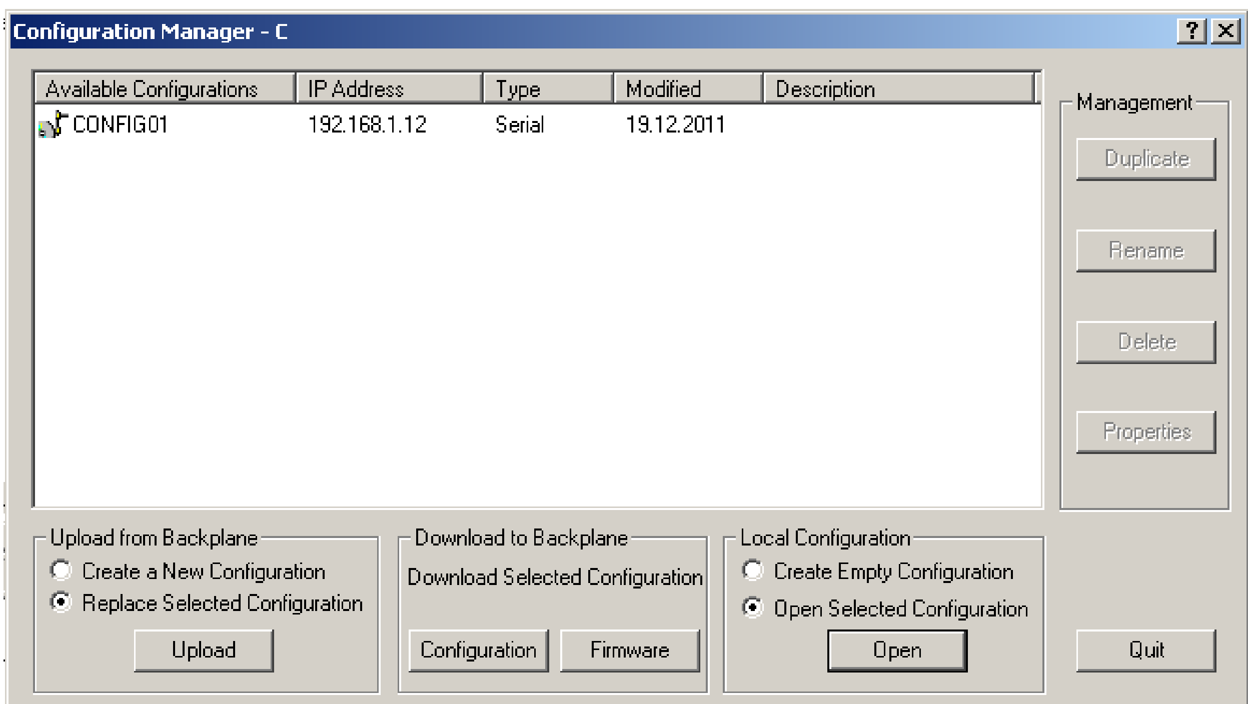
The detailed description of each BCMS component is covered in BCMS section of this user guide.

3.2.1 Configuration Manager

Configuration Manager provides different functionalities and will be discussed briefly in this section.

To run the Configuration Manager, go to the Start menu and click on . The main screen as shown below will be displayed.

Figure 2: Configuration Manager Main Screen



Note

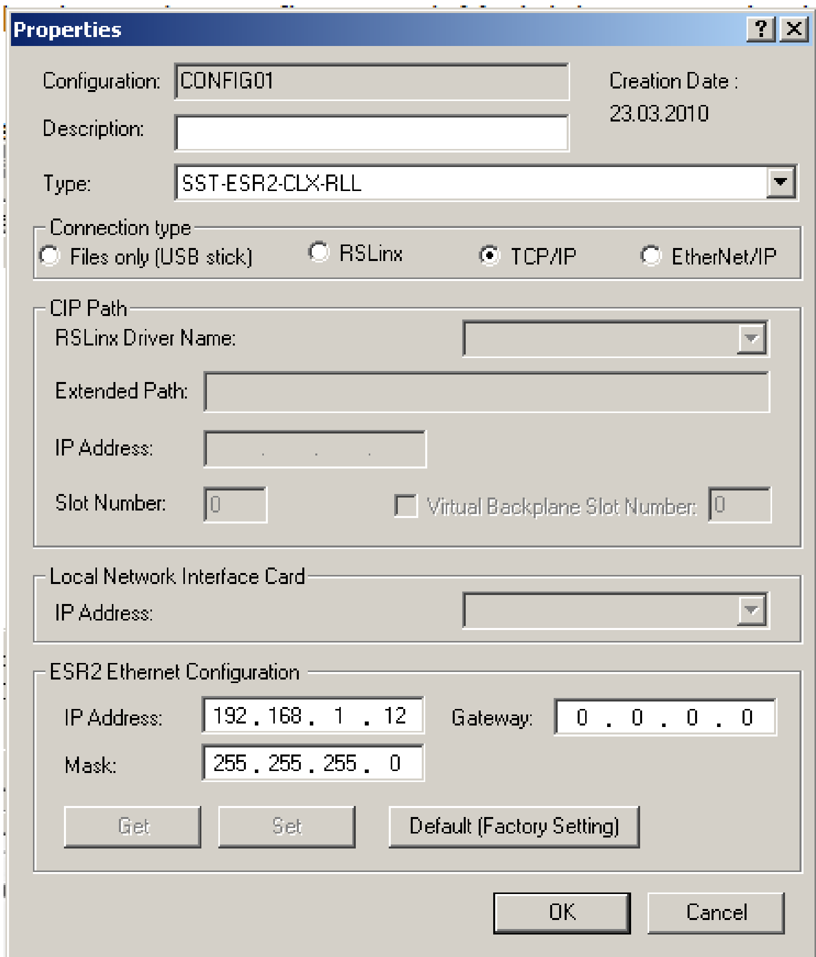
If the module being configured is SST-SR4-CLX-RLL, the type shown will be "Serial". If the module is SST-ESR2-CLX-RLL, the type shown will be "Eth/Ser

The Configuration Manager can be used to perform the following functions:

- List all configurations already present on the machine's hard drive
- Upload/Download configuration from/to the module.
- Create new configurations, either from scratch or by importing them from a previously configured module.
- Manage several directories corresponding to different configurations
- Download new firmware to the module
- Modify Configuration Property by: See Figure_3 for illustration
  - Changing connection type
  - Changing RsLinx Driver name
  - Changing ControlLogix Slot number.
  - Retrieve module's IP address
  - Update module's IP address

The Properties dialog is used to configure Module's communication channel:

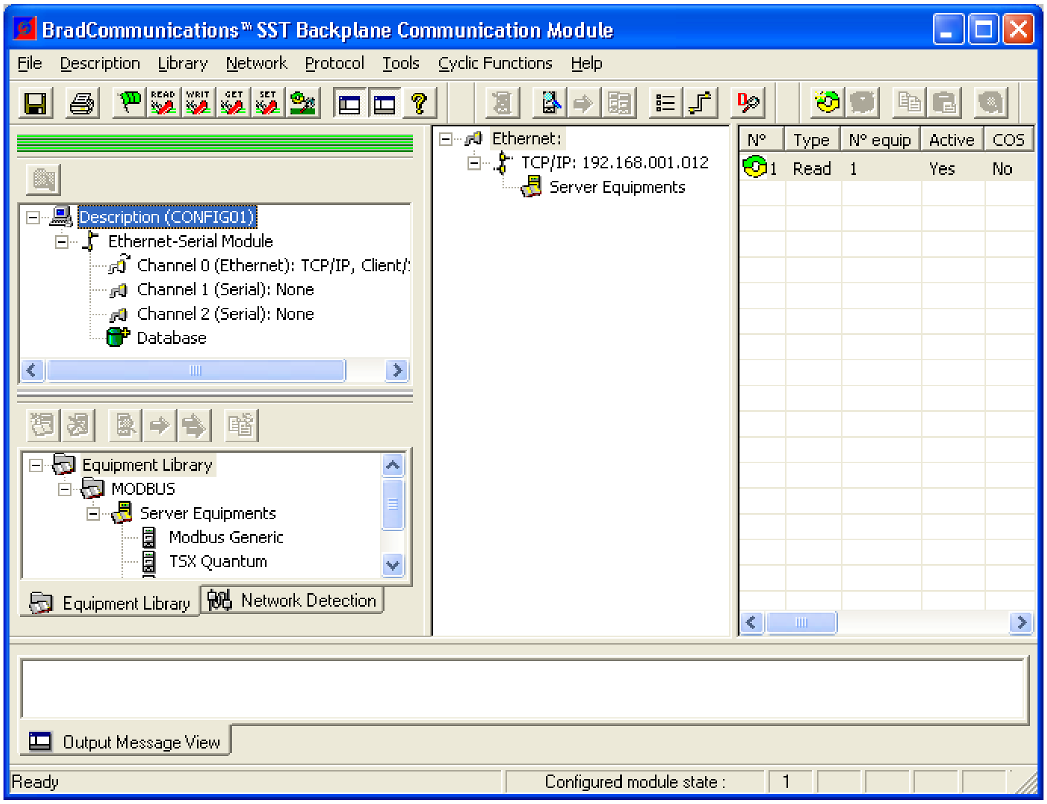
Figure 3: Properties Dialog



3.2.2 Console

When Console is opened, similar window will be displayed as shown below.

Figure 4: Console Application Main Screen



Console application provides links to various application listed below:

- PCInit – this application is used to download configuration to the module or initialize the module to a known working state.
- ReadWait – this application is used to send a read message to the server equipments. The type of read message sent depends on the server equipment configuration.
- WriteWait – this application is used to send a write message to the server equipments. The type of write message sent depends on the server equipment configuration.
- GetDB – this application is used to directly access (read) the module's database. The request sent to the module does not require any protocol or message type.

- SetDB - this application is used to directly access (write) the module's database. The request sent to the module does not require any protocol or message type.

- Network Diagnostic – is used to diagnose the Ethernet activity of the SST-ESR2-CLX-RLL module. This application does not support non-Ethernet port.

- Cyclic Function – is used to create cyclic function when module is running in Modbus Client, Modbus Master or Siemens S7/S5 Client configuration. See Cyclic Function section for detailed information on Cyclic Function.

- Visucyc – this application is used to monitor all the configured cylic function running in the module.

The Console (also referred as Backplane Console) is an integrated environment used to configure the Backplane and protocols for the communication module. It can perform the following functions:

- Configure multiple port configurations
- Configure multiple server/slave configuration
- Configure networks and connected devices
- Add devices to the configuration
- Select messaging protocol
- Configure protocol parameters
- Declare and configure cyclic functions for data exchange between the three-to-four networks
- Initialize the module while the configuration is being developed

Note

Once a configuration has been saved, it is also possible to initialize, carry out diagnostics and use the various operating tools independently of the Console.

3.3 Module's Installation Directory Location

3.3.1 Configurations

When running windows XP**:** The module's configuration is in

C:\Documents and Settings\All Users\Application Data\BradCommunications\SST Backplane Communication Module\config

When running in Windows 7: The module's configuration is in

C:\ProgramData\BradCommunication\SST Backplane Communication Module\config

3.3.2 AOI Files

When running windows XP: The AOI files are in

C:\Program Files\BradCommunications\SST Backplane Communication Module\Ladder Sample Code For ControlLogix\AOI

When running Windows 7*:* The AOI files are in

C:\Program Files (x86)\BradCommunications\SST Backplane Communication Module\Ladder Sample Code For ControlLogix\AOI

3.3.3 Module Firmware

When running windows XP: The module's firmware is in

C:\Documents and Settings\All Users\Application Data\BradCommunications\SST Backplane Communication Module\Firmware Update

When running in Windows 7: The module's configuration is in

C:\ProgramData\BradCommunication\SST Backplane Communication Module\Firmware Update

4

Product Features Overview

Chapter Sections:

- Communication Modules Variant
- Supported Protocols
- Module Capabilities
- Database Feature
- Add-On-Instruction Support
- Add-On-Profile Support
- Cyclic Functions
- CIP Messaging
- Operating Modes Supported
- Client or Server Configuration (Master or Slave)
- Module Version Information

4.1 Communication Modules Variant

The SST ControlLogix Communication Modules comes in two variants: SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL modules.

The SST-ESR2-CLX-RLL communication module features 1 Ethernet and 2 Serial communication ports with each port capable of running supported protocol messaging.

The SST-SR4-CLX-RLL communication module features 4 Serial communication ports with each port capable of running supported protocol messaging.

4.2 Supported Protocols

- SST-ESR2-CLX-RLL communication modules can support the following protocols.
    - Modbus TCP Messaging
    - Modbus Serial Messaging
    - Siemens S5 Messaging
    - Siemens S7 Messaging
- SST-SR4-CLX-RLL communication modules can support
    - Modbus Serial Messaging

4.2.1 Supported Modbus Function Codes and Equipment Types

Table 4: Modbus Supported Function codes

Device Variable Type	Data Size	Cyclic Function Type	Modbus Function Code	Equipment Type
word	>= 1	read	FC 3 : read multiple register	See Modbus Protocol Guide for supported equipment Guide
word	== 1	write	FC 6 : write single register	See Modbus Protocol Guide for supported equipment Guide
word	>= 1	write	FC16 : write multiple register	See Modbus Protocol

				Guide for supported equipment Guide
input word	n/a	read	FC 4 : read input register (not supported)	See Modbus Protocol Guide for supported equipment Guide
output word	n/a	read	FC 4 : Not supported	See Modbus Protocol Guide for supported equipment Guide
bit	>= 1	read	FC 1: read coil	See Modbus Protocol Guide for supported equipment Guide
input bit	n/a	read	FC 2: read input discrete (not supported)	See Modbus Protocol Guide for supported equipment Guide

4.2.2 Supported S5 PLC Variable Types

Table 5: Siemens S5 Supported PLC variable syntaxes

PLC Variable Type	Data Type	Cyclic Function Type	Function Code	Equipment Type
ABx/AWx	bits,byte, word,dword	Read/Write	n/a	S5
EBx/EWx	bits,byte, word,dword	Read	n/a	S5
MB0-127/MW0-127	bits,byte, word,dword	Read	n/a	S5
MB128-255/MW128-255	bits,byte, word,dword	Write	n/a	S5
DBx.DWx	bits,byte, word,dword	Read/Write	n/a	S5

4.2.3 Supported S7 PLC Variable Types

Table 6: Siemens S7 Supported PLC variable syntaxes

PLC Variable Type	Data Type	Cyclic Function Type	Function Code	Equipment Type
ABx/AWx	bits,byte, word,dword	Read/Write	n/a	S7-300 / S7-400 / S7-1200
EBx/EWx	bits,byte, word,dword	Read	n/a	S7-300 / S7-400 / S7-1200
MB0-127/MW0-127	bits,byte, word,dword	Read	n/a	S7-300 / S7-400 / S7-1200
MB128-255/MW128-255	bits,byte, word,dword	Write	n/a	S7-300 / S7-400 / S7-1200
DBx.DBx / DBx.DWx	bits,byte, word,dword	Read/Write	n/a	S7-300 / S7-400 / S7-1200

4.3 Module Capabilities

The features offered in each of the variant are summarized in the table below.

Table 7: Module Capabilities

Features	SST-ESR2-CLX-RLL	SST-SR4-CLX-RLL	Firmware Version Required
Direct Mapping of up to 30K of IO data to modules database	√	√	2.10.2 or higher
Variable IO and Status Size	√	√	2.10.2 or higher
Support to RA RSLogix 5000 AOI	√	√	2.10.2 or higher
S7 Client Configuration**	√		2.10.2 or higher
S7 Server Configuration	√		2.10.2 or higher
S5 Client Configuration	√		2.10.2 or higher
S5 Server Configuration	√		2.10.2 or higher

S7 Client/Server Configuration	√		2.10.2 or higher
S5 Client/Server Configuration	√		2.10.2 or higher
Gateway (Backplane <-> Supported Messaging Protocol)	√	√	All
Access up to 30K words of IO data using CIP messaging	√	√	All
Up to 496 bytes of Input and 496 bytes of Output data every RPI	√	√	All
15 Listen-Only connection plus 1 Output Connection	√	√	All
Unconnected CIP messages	√	√	All
Support to RA RSLogix 5000 Add-On-Profile	√	√	2.3.0
Support Remote Rack Configuration	√	√	All
Modbus Serial Master Configuration	√	√	All
Modbus Serial Slave Configuration	√	√	All
Simultaneous Modbus Master/Slave Configuration	√	√	All
255 Cyclic Function per channel	√	√	All
Dynamic Cyclic Function Management using CIP messages	√	√	All
Dynamic Cyclic Function Management using AOI	√	√	2.10.2 or higher
Configuration Upload/Download	√	√	All
Diagnostic Tool for remote slave devices	√	√	All
Firmware Update	√	√	All
Database Address Overlap Detection	√	√	2.10.2 or higher

Modbus TCP Client Configuration	√		All
Modbus TCP Server Configuration	√		All
Modbus TCP Client/Server Configuration	√		All
Supports up to a total of 128 Client/Server connections	√		All
Supports up to a total of 60 client connection when in server configuration	√		All

**Only support S7-1200 server configuration at this time.

4.4 Database Feature

The module's database supports two addressing mode. The difference between the two addressing mode will be discussed briefly in this section. Detailed discussion will be in Database Mapping section.

Default addressing mode is the original addressing mode supported by the communication modules. This type of addressing mode allows direct mapping of the ControlLogix IO table to the modules database. The INPUT table is mapped to address 2-249, OUTPUT table is mapped to address 250-497 and the STATUS table is mapped to address 500-699. Variables or data from address 700 and above can be accessed using CIP messaging. Sample Ladder logic projects are provided as part of our installation package. This ladder logic can be used as it is to access address 700 and above. It can also be easily modified and customized according to your network setup. See table below for database mapping illustration.

Note

If you are running in a 64-bit machine, ladder Logic can be found in

C:\Program Files (x86)\BradCommunications\SST Backplane Communication Module\Ladder Sample Code For ControlLogix

If you are running in a 32-bit machine, ladder logic can be found in

C:\Program Files\BradCommunications\SST Backplane Communication Module\Ladder Sample Code For ControlLogix.

Table 8: Default Database addressing mode configuration

SST_ESR2_CLX_RLL / SST_SR4_CLX_RLL Module <> CLX Mapping in Default Mode		
Communication Module	Database Offset / size	CLX Memory Location
Mapped Input Area	2 – 249 / (248 words)	INPUT Table
Mapped Output Area	250 – 497 / (248 words)	OUTPUT Table
Mapped Status Area	500 – 699 / (200 words)	STATUS Table
Unmapped Area	700 – 30719	Accessible by CIP Messaging

Extended addressing mode is the latest addition to our product features set. This type of database addressing mode provides flexibility to the user by allowing direct mapping of the module's entire database to the ControlLogix IO table. The mapped size can vary from 0-30K and can be customized according to your network setup. It can be mapped anywhere within the 30K of the module's database. Please see example below:

Table 9: Extended Database addressing mode configuration

SST_ESR2_CLX_RLL / SST_SR4_CLX_RLL Module <> CLX Mapping in Extended Mode		
Communication Module	Database Offset / size	CLX Memory Location
Mapped Input Area	Anywhere (512 – 28664 words)	CLX2000.DATABASE_DATA.INPUT_DATA Controller Tag
Mapped Output Area	Anywhere (512 – 28664 words)	CLX2000.DATABASE_DATA.OUTPUT_DATA Controller Tag
Mapped Status Area	Anywhere (0 – 1024 words)	CLX2000.DATABASE_DATA.STATUS_DATA Controller Tag
Cyclic Command Words	30720 (1024 words)	Reserved for Dynamic Triggering
Cyclic State Words	31744 (1024 words)	Reserved for Dynamic Triggering

Note

This feature is only supported from Firmware version 2.10.2 and higher.

4.5 Add-On-Instruction Support

The latest firmware version 2.10.2 supports Rockwell Automation RSLogix 5000 Add-on-Instruction when using Generic Profile (1756-Module). If using RSlogix 5000 Add-on-Instruction with Add-On-Profile (AOP) then backplane firmware version 2.12.1 or higher is required. The Add-On-Instruction feature allows direct mapping of the module's entire database to the ControlLogix IO table without running any additional CIP messages.

Note

When using the AOI with AOP (Add-On-Profile), the lowest RPI that can be set is 5ms.

4.6 Add-On-Profile Support

Firmware version 2.3 and above supports Rockwell Automation RSLogix 5000 Add-On-Profile programming.

4.7 Cyclic Functions

4.7.1 Concept

The **Backplane Console** is used to configure cyclic data exchange between master/slave or client/server connection. Data received from server is stored in the client database, while data sent to server originates from the client's database. The cyclic function data exchanges will take effect only if the configuration has been downloaded to the module.

The Console also includes commands for printing your configuration. Certain parameters must be entered for each cyclic function. If you save your work after configuring the functions, the Console archives your changes in the active configuration.

After making any changes to the configuration, you must download the modified configuration using the initialization utility, accessible via the Console.

4.7.2 Read Cyclic Function

A read cyclic allows reading of the server equipment data and stored into the modules database. The section of the database that stores the data read from the server equipment can be shared into the backplane by mapping it to the modules INPUT table or share it with other server equipment connected to the module. The database address used to store read data from the server equipment is the database address parameter configured when cyclic function was created.

A read cyclic function can be configured with different trigger option. Further detail is explained in Cyclic Function Triggering Mode section.

4.7.3 Write Cyclic Function

A write cyclic function allows writing of data to the server equipment. The data will come from backplane when the location of the data in the database is mapped into the OUTPUT table or from other server equipment connected to the module. The database address used as a source of data written to the server equipment is the database address parameter configured when cyclic function was created.

A write cyclic function can be configured with different trigger option. Further detail is explained in Cyclic Function Triggering Mode section.

4.8 CIP Messaging

Our sample ladder logic project supports different type of CIP messaging. e.g.: SET_WORD, GET_WORD, etc. User can use these CIP messages to the following:

- Read data from the modules database
- Write data to the modules database
- Dynamically triggers cyclic functions created in the configuration.

4.9 Operating Modes Supported

The Module's mode of operation follows that of the CLX PLC (Program, Test, Run).

Run Mode

When CLX Processor is set to “Run Mode”, the communication module’s database address that is mapped to the ControlLogix IO/Status Table is updated with IO and Status data every RPI (Request Packet Interval) that is configured for the module in your RSLogix 5000 program or Honeywell Plantscape Control program.

Program Mode

When CLX Processor is set to “Program Mode”, the communication module’s database address that is mapped to ControlLogix Output table may have either of the following values:

- Zero
- Last Output value before the CLX Processor mode has been changed to Program Mode.

When in Program Mode, the module can also be configured to stop all cyclic functions. This means that all network traffic related to the output table will be stopped.

Test Mode

This operating mode behaves similarly to Program Mode.

4.10 Client or Server Configuration (Master or Slave)

Communication Module as Client Equipment (Master)

The communication module channels or ports can be configured as client or master. This allows multiple networks running supported protocol communicates to a single module.

When there is an active backplane connection, ControlLogix IO data can be shared to the connected network equipment using cyclic function.

Communication Module as Server Equipment (Slave)

The communication module channel or ports can be configured as a server or slave. This allows multiple server / slave configuration / connection to a single or multiple client equipment.

When there is an active backplane connection, ControlLogix IO data can be shared to the connected network equipment using cyclic function.

Communication Module as Client and Server (Master/Slave)

Since each port or channel can be configured independently from each other, this allows a client/server (master/slave) configuration in a single module.

4.11 Module Version Information

BCMS provides different options to verify the firmware version running in the module. The communication modules contain two processors, each running different firmware.

The firmware running in NIOS processor which is primarily responsible of any backplane communications can be verified using the ***Firmware Updater application.***

The firmware running in PPC processor which is primarily responsibly in any network or protocol related communications can be verified using ***Firmware Updater Application*** and ***Apsym Application.***

5

Quick Start Guides

Chapter Sections:

- Introduction
- How to verify backplane communication to ESR2/SR4 module
- How to configure the module in RSLogix 5000 Software
- How to browse the module from the network
- How to make initial connection to the module
- How to initialize the module
- How to change database addressing mode

5.1 Introduction

This section will provide procedures how to use the communication module. Before proceeding to the subsequent section, make sure that you have all the required equipment/software ready. See the list below.

Table 10: Component Checklist

Qty	*Required Component*	*Description*
N/A	*Brad Communication ™ SST Backplane Communication Module Software (BCMS)*	*The communication module configuration tool*
N/A	*Rockwell Automation RSLogix Programming Software*	*Required for Backplane Configuration*
1	*Ethernet Cable*	*To connect the Module to the Modbus TCP or Industrial Ethernet network*
1	*SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL*	*Communication Modules*
1 - 4	*RJ45 Serial cable adapters to Db9 format*	*To connect to Modbus Serial network (4 if using SST-SR4-CLX-RLL)*
1	*L55, L6x or L7x*	*ControlLogix Processor for Backplane communication*
1 or 2	*1756-EN2T or 1756-ENBT*	*1756 10/100 Mbps Ethernet Bridge for Backplane communication (NOTE: 2 is required if configuring in remote rack)*
1 or 2	*1756 ControlLogix Rack*	*Tested to run in rack with chassis number of 4, 7, 10, 13*

5.2 How to Verify Backplane Communication to ESR2/SR4 Module

Warning

Before insertion or removal of module while under electrical power, ensure the environment is free from any explosive or hazardous contaminants, otherwise serious injury to persons or equipment could result.

If you are using an Ethernet Module, refer to Installing and Configuring the RSLinx Driver before proceeding. After successful installation of the Ethernet Module, you can proceed and skip procedures for installing an Ethernet/IP module.

The module has to be inserted to the rack properly. Below are procedures to install and remove the module from the ControlLogix rack.

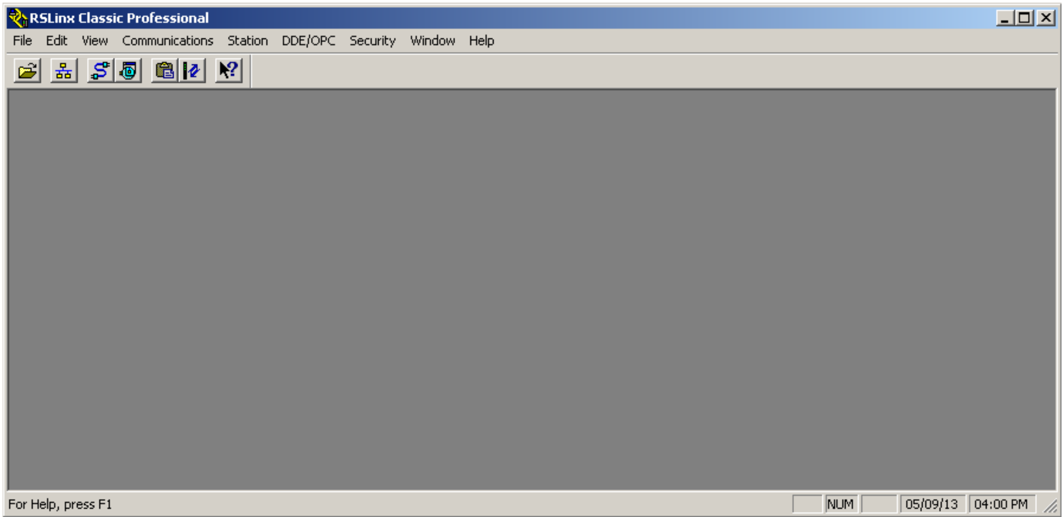
5.2.1 To install the module to the rack: follow these steps:

1. Disconnect power from the rack. This is an optional step as the module supports Removal and Insertion-Under-Power (RIUP)
2. Using the chassis card guides, align the full-sized circuit board.
3. Slide the Module into the chassis until the top and bottom latches catch.
4. Attach the network cables.
5. Apply connector termination as required.
6. Route the cable down and away from the Module.

To verify communication from the backplane, follow the steps below:

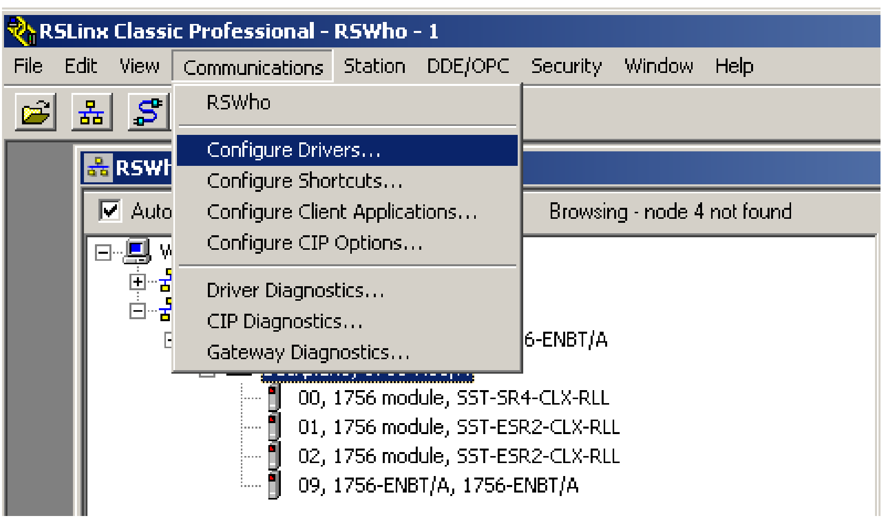
1. Insert a Ethernet/IP bridge into rack such as a 1756-ENBTor 1756-EN2T
2. Launch a licensed version of RSLinx. RSlinx will open similar to screenshot below.

Figure 5: RSLinx Classic Professional



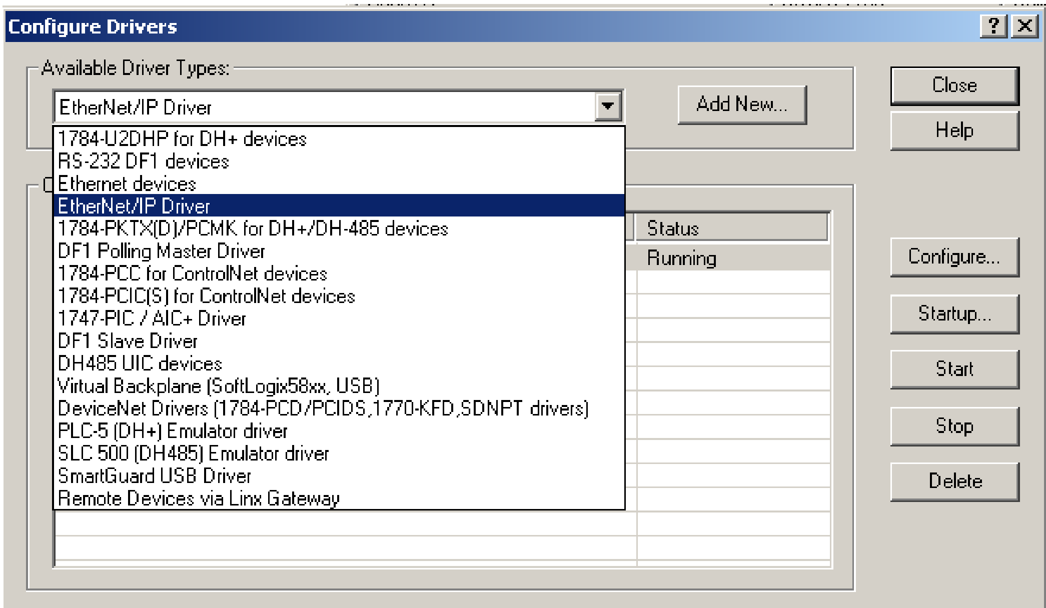
3. Configure the EtherNet/IP bridge, follow steps below:
    a) Locate the Communication tab in the screen. Click on Configure Drivers.

Figure 6: EtherNet/IP Bridge driver configuration



    b) The screen as displayed below will show up. Select EtherNet/IP driver then click "Add New"

Figure 7: Adding Ethernet Bridge



4. The machine where RSLogix Programming Software is installed may have multiple network interfaces. Choose the appropriate one with the same IP address segment as the EtherNet/IP module. If no IP address is of the same segment with the IP address displayed in the EtherNet/IP display, the IP address of the machine needs to be changed to be on the same IP address segment as the EtherNet/IP module to be able to communicate. Once communication to the module has been established, the IP address segment on the module can be changed to match the machine's IP address segment.

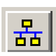
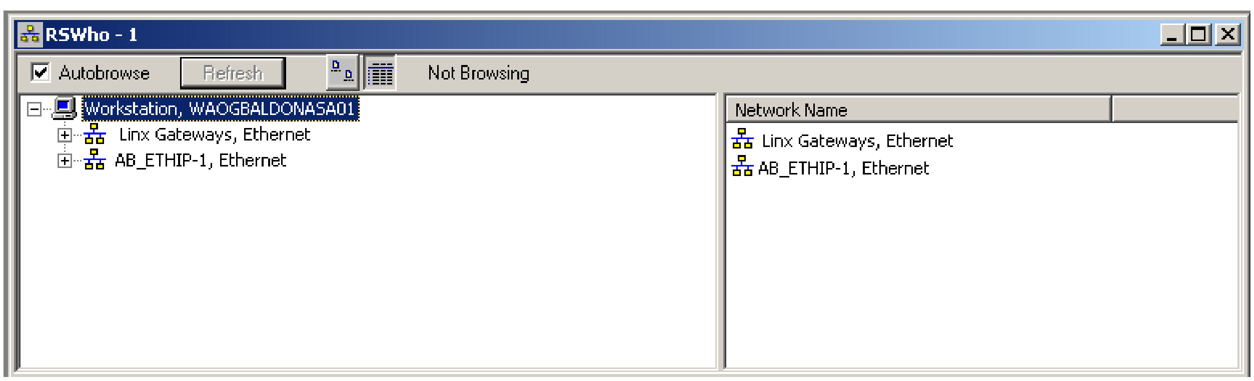
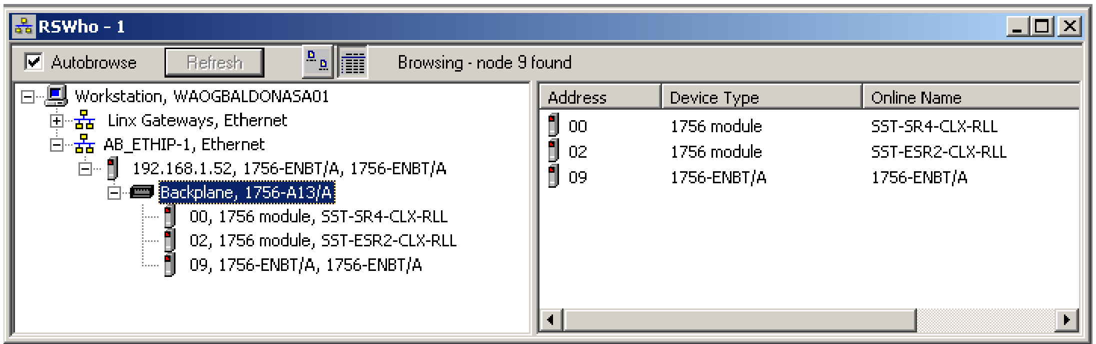
5. Click on the , the window similar below should show up in your screen.

Figure 8: RSLinx Classic Professional



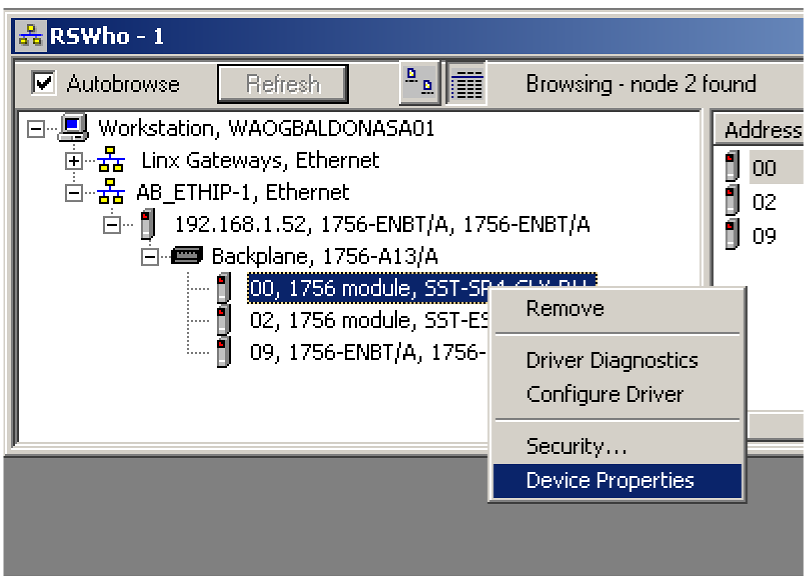
6. Expand the list that shows AB-ETHIP-1 and you see similar to what is displayed below.

Figure 9: RSWho Display



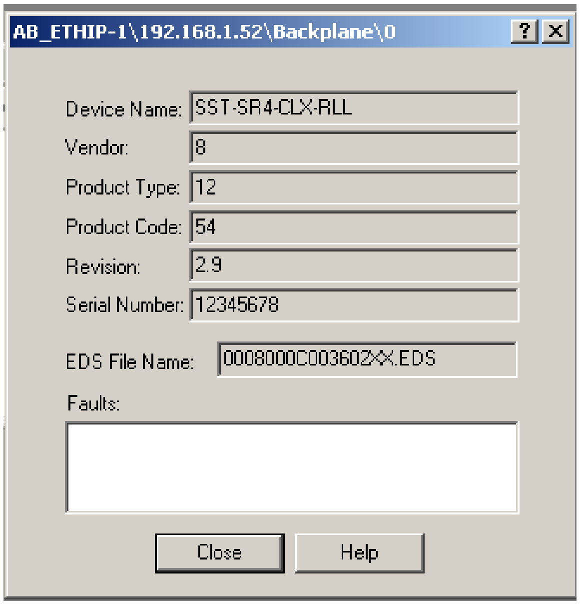
7. The display is showing 1 SST-SR4-CLX-RLL in slot 0 and 1 SST-ESR2-CLX-RLL in slot 2. Select the right product variant that is currently installed in your rack..
8. After selecting the right product variant, right click and the display below will be shown.

Figure 10: Checking Backplane Module Identity



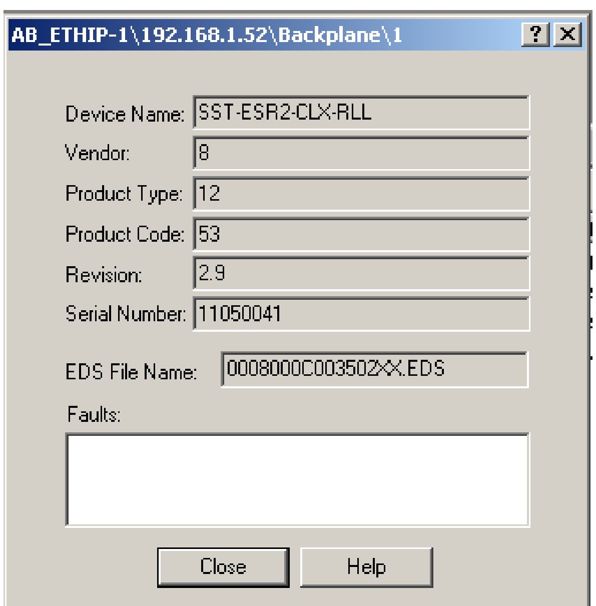
9. Right click on Device Properties and it will show up the diagram below if the SST-SR4-CLX-RLL is selected. The module may have a different firmware version. The firmware version displayed during "BOOT" is the firmware version displayed in the "Revision" in the display below.

Figure 11: SST-SR4-CLX-RLL Backplane Identity



10. If the above information is displayed successfully, the backplane communication with the module is working fine.
11. Select SST-ESR2-CLX-RLL, the module information will be as shown below.

Figure 12: SST-ESR2-CLX-RLL Backplane Identity



Note

Product code for SST-ESR2-CLX-RLL is 53 while SST-SR4-CLX-RLL is 54. If Revision is showing a different version than the one showed in the sample screenshot, your module is running a different firmware revision than what is running in the module where screen shot was taken. Please contact our support team if you wish to upgrade the module's firmware.

5.2.2 To remove the module from the rack: follow these steps:

Warning

Before insertion or removal of module while under electrical power, ensure the environment is free from any explosive or hazardous contaminants, otherwise serious injury to persons or equipment could result.

1. It is not necessarily need to disconnect the power, as the Module supports removal under power.
2. Remove all cabling from the Module.
3. Press the releases at the top and bottom of the Module and slide the Module out of the Module slot.

5.3 How to Configure the Module in RSLogix 5000 Software

There are two ways of configuring the SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL in RSLogix 5000; Generic 1756 Module type and AOP (Add-on-Profile). Generic 1756 Module is supported in all versions of RSLogix 5000. AOP is supported in RSLogix 5000 versions 15 and higher. AOP is the recommended method since it allows you to configure all options on the SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL via a GUI.

AOP generates data tags that are more specific to the module. Diagnostic tags are easier to identify and the I/O tags are separated from the command control registers. The configuration file is automatically updated when you select options in the AOP interface.

For customers with existing applications, the project can be upgraded to use AOP but it will be necessary to delete the Generic profile prior to using AOP. This includes updating ladder logic to use AOP tags instead of generic tags and then deleting the generic tags. AOP and generic sample ladder code is available.

If you need to reduce the size of the I/O tables you can use 1756-Module (Generic profile). I/O sizes are fixed at their maximum sizes in AOP.

Note

When you are configuring the Module, ensure that the CLX is in Program mode.

5.3.1 Using 1756 Add-On-Profile

Note

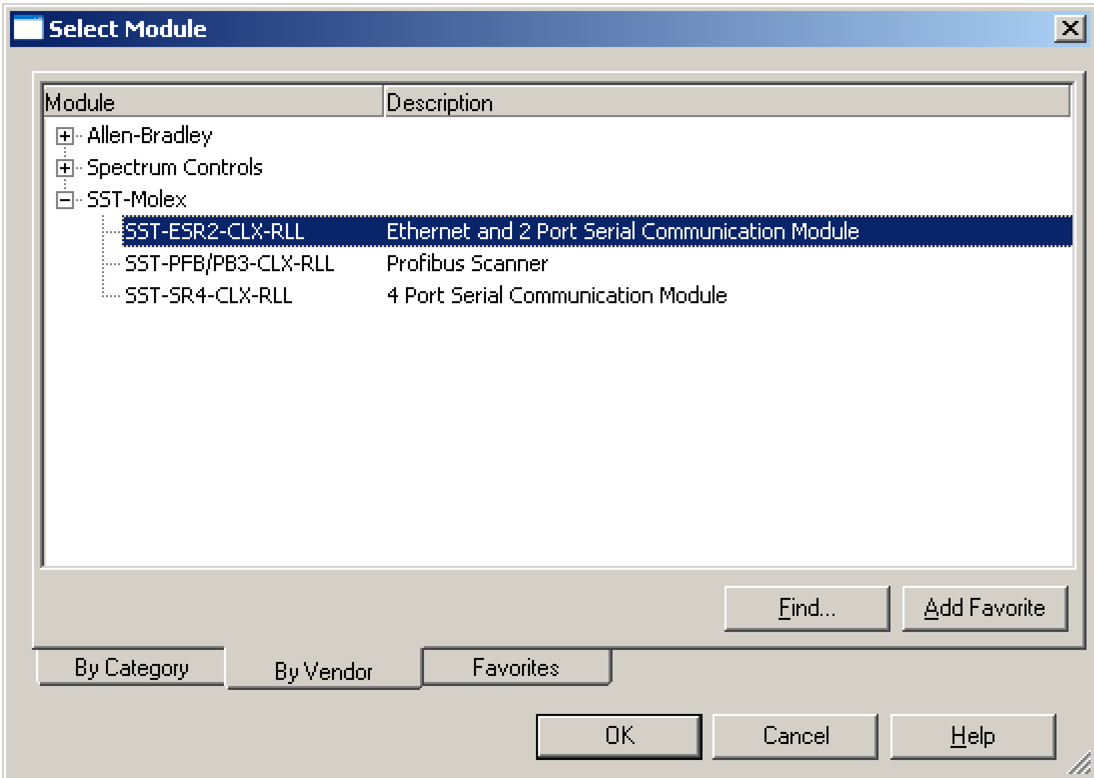
The module's AOP only supports maximum input, output and statu (250, 248, 250). Use 1756-Module Generic Profile if smaller conne are required.

1. From the SST Backplane Communication Module Products CD, run the SST Backplane Communication Install by going to Product Menu/SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL/. This will install the following components to configure the module for AOP:
    - o SST Backplane Communication console
    - o Firmware module esrclx.ss4, Default install directory is Program Files\BradCommunications\SST backplane Communication Module\Firmware Update
    - o RSLogix 5000 AOP sample ladder code. Default install directory is Program Files\BradCommunications\ SST backplane Communication Module \Ladder Sample Code for ControlLogix\AOP
2. From the SST Backplane Communication Module Products CD, run Rockwell's RSLogix 5000 AOP install and follow the instructions.
3. AOP requires the SST-ESR2/SR4-CLX-RLL to be loaded with firmware module esrclx.ss4 2.3 or higher. To determine if your module has this firmware version, view properties of module in RSLinx and see Revision. If you need to upgrade firmware see chapter 9 Upgrading Firmware Module.

5.3.1.1 To configure through RSLogix 5000, follow these steps:

1. In RSLogix 5000, from the project tree, right-click on 1756-Backplane and select New Module…
2. Scroll down until you see the Vendor Name "SST-Molex". Select SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL. The SST-ESR2-CLX-RLL is selected for this procedure. The module is also displayed under Category\Communications.

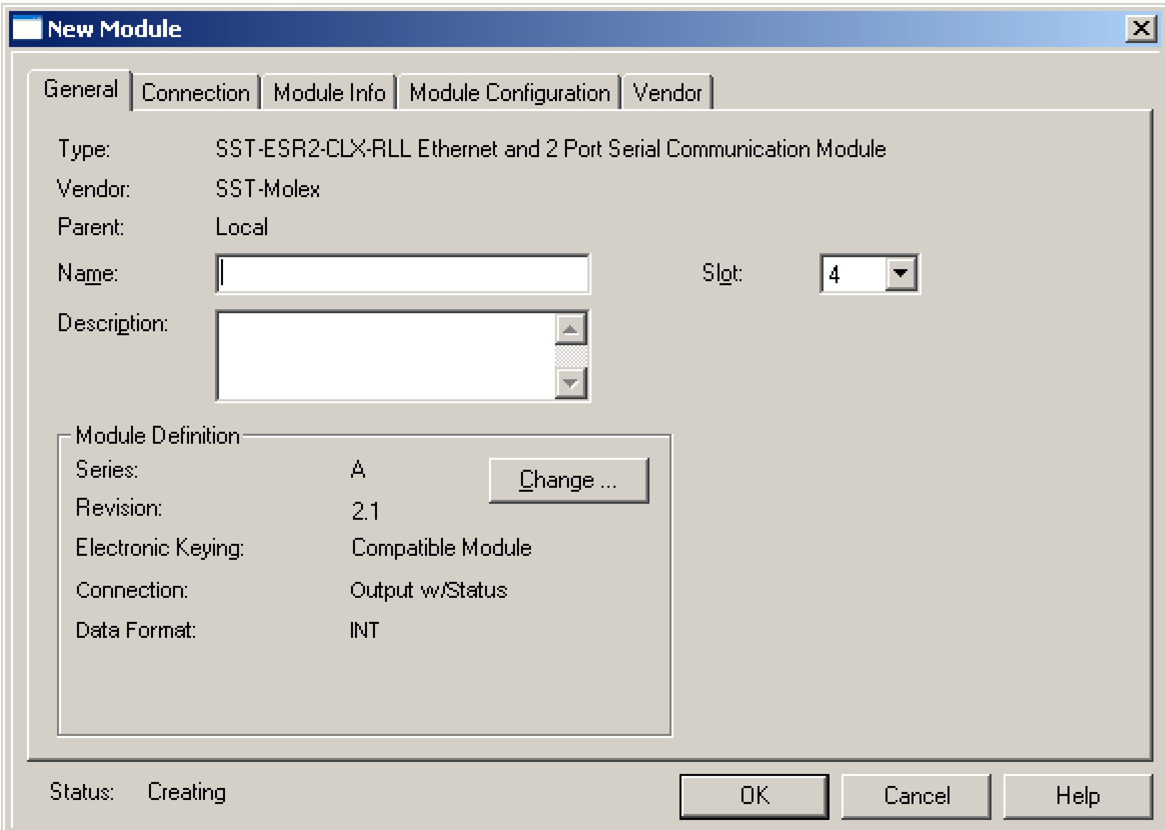
Figure 13: Adding Module to the IO Configuration



If there are more than one PLC communicating with the same SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL (output connection plus listen only connection), all must use the Generic Module type or AOP. Generic module and AOP types cannot be mixed.

3. Double click on SST-ESR2-CLX-RLL, The following window will appear.

Figure 14: Module Configuration Entry



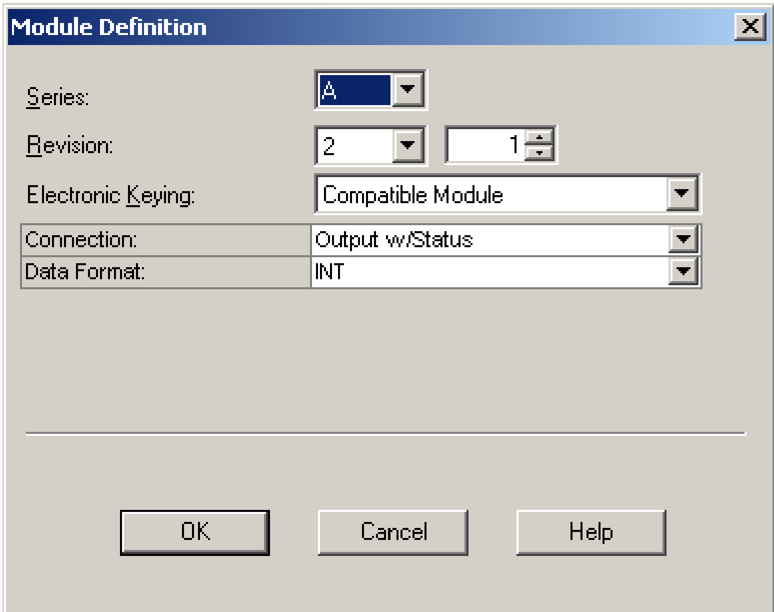
4. Enter a name in the Name field. This parameter must begin with a letter.

5. Enter a description. This is optional.

6. Click on the Down Arrow to configure the slot number of module.

7. If a different Connection type is required such as Listen-Only or different data format, click on the **Change…** button. The following Module Definition window will appear.

Figure 15: Module Definition Display



8. Series can only be A.

9. The Revision refers to the backplane firmware version on module. The revision contains a Major and Minor revision number. The defaults for Major and Minor are 2 and 1.This revision is displayed on the SST-ESR2/SR4-CLX-RLL LCD display at power-up and is displayed in RSLinx when viewing the properties of our module in RSWho.

10. Electronic keying is a feature that allows a consistency check to be done between the physical module and software configuration of the module. The consistency check involves checking attributes Vendor, Product Type, Catalog number, Major and Minor Revision. There are three modes available: Compatible, Exact Match, and Disable Keying. The default is compatible mode. In compatible mode, a COPN to the module is only successful when:

    - Module type and Catalog Number and Major revision must match

    - The Minor revision of the physical module must be equal to or greater than the one specified in the software.

    If the Exact Match option for the Electronic keying is selected, all of the attributes of the SST-ESR2-CLX-RLL module and the module created in software must match. If the module's firmware is upgraded later, the Revision setting must be updated so that it matches exactly.

    The "Disable Keying" option allows a COPN to the module without performing a consistency check. This option is not recommended.

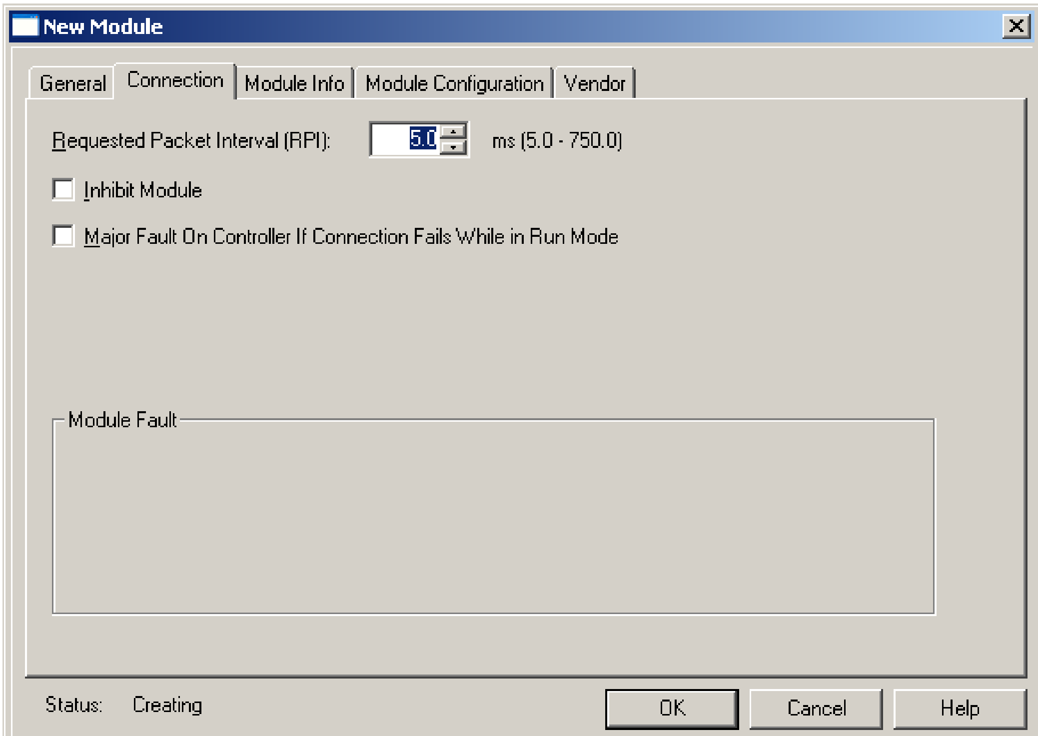
11. The default connection type is Output w/ Status. Listen-only connection is also available.

12. Select the data format required (INT, SINT, DINT). INT is the default.

13. Click OK to accept the changes.

14. If required, click on the Connection tab to modify the Rate Packet Interval (RPI) time. The default is 5ms. The valid range is 5ms to 750ms.

Figure 16: Connection Definition



15. Click on Module Configuration tab to configure the program mode output state. The default state is Zero-CyclicRuns.

Figure 17: Module Output State Configuration

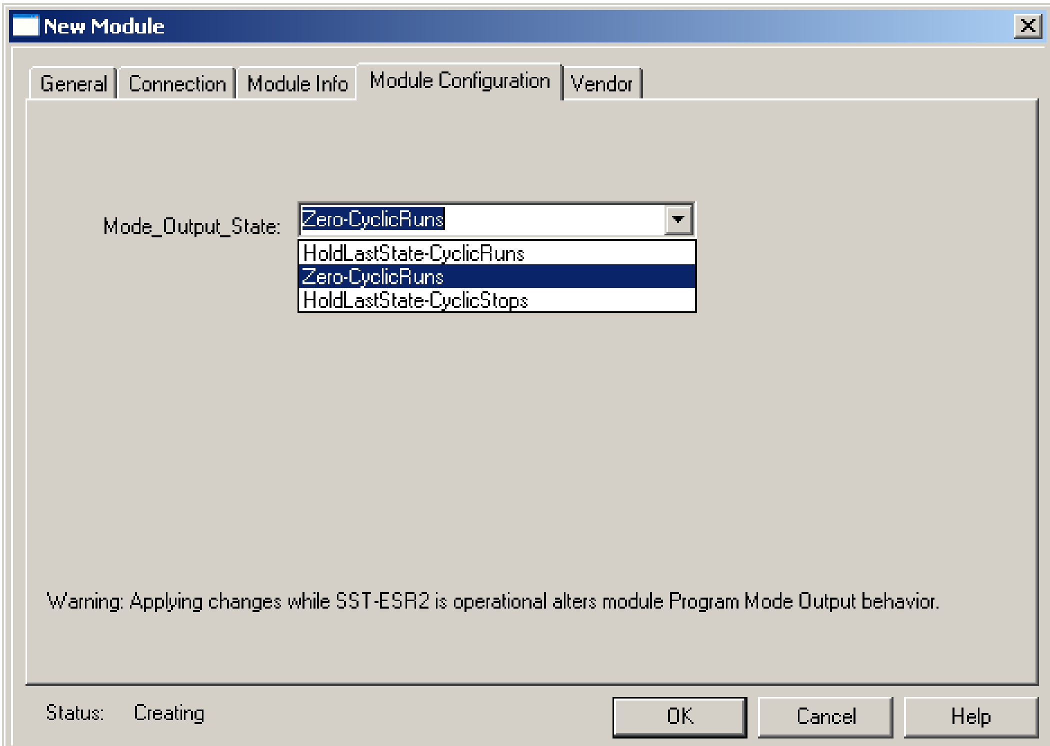


Table 11: AOP Setting

AOP Setting	Description
HoldLastState-CyclicRuns	When in Program mode, leaves outputs in their last state. During this state, cyclic functions continue to operate.
Zero-CyclicRuns	Zeroes outputs when in Program mode. During this state, cyclic functions continue to operate.
HoldLastState-CyclicStops	When in Program mode, leaves outputs in their last state. Cyclic functions related to the output table are stopped.

16. Next set the configuration name. The configuration name comes from the configuration you created for the module using the Backplane communication console. By default, this is set to NULL and will allow any configuration to connect to module. In the first release of AOP, the configuration name is not configurable through the GUI. Go directly to configuration tag array under controller tags and find Local:Slot:C.ConfigName. Enter the ASCII characters for your configuration name.

Note

Our web address, online help and technical support phone numbers are available on the Vendor tab.

17. Click on OK. The SST-ESR2-CLX-RLL will appear in your I/O configuration.

Now begin creating your ladder code using the detailed tags created by AOP for the SST-ESR2-CLX-RLL. For a more detailed description of I/O and status array tags see Table 49 in Communication Module Status Information Section.

AOP allows making modifications to the configuration and apply them while the module has an open connection.

Warning

Be aware that applying changes while the SST-ESR2-CLX-RLL or SST-SR4-CLX-RLL is operational alters program mode output behavior.

To verify if the applied new configuration was successful, check the ReconfigStatus register in the module's status file.

5.3.1.2 ControlLogix Modes.

The Module's mode of operation follows that of the CLX PLC (Program, Test, Run).

- When the CLX is in Program mode, the Module outputs are either held at their last state, cleared, or all cyclic functions associated with the output table (word database address 250 – 498) are stopped. The Input table (word database addresses 2 – 249) continues to be updated when the Module is in Program mode.

- When the CLX is in Test mode, inputs are read and updated but outputs are either held at their last state, cleared, or all cyclic functions associated with the output table (word database address 250 – 498) are stopped.

When the CLX is in Run mode, the Module updates inputs and outputs and is updated at the RPI rate.

5.3.2 Using 1756 Generic Profile

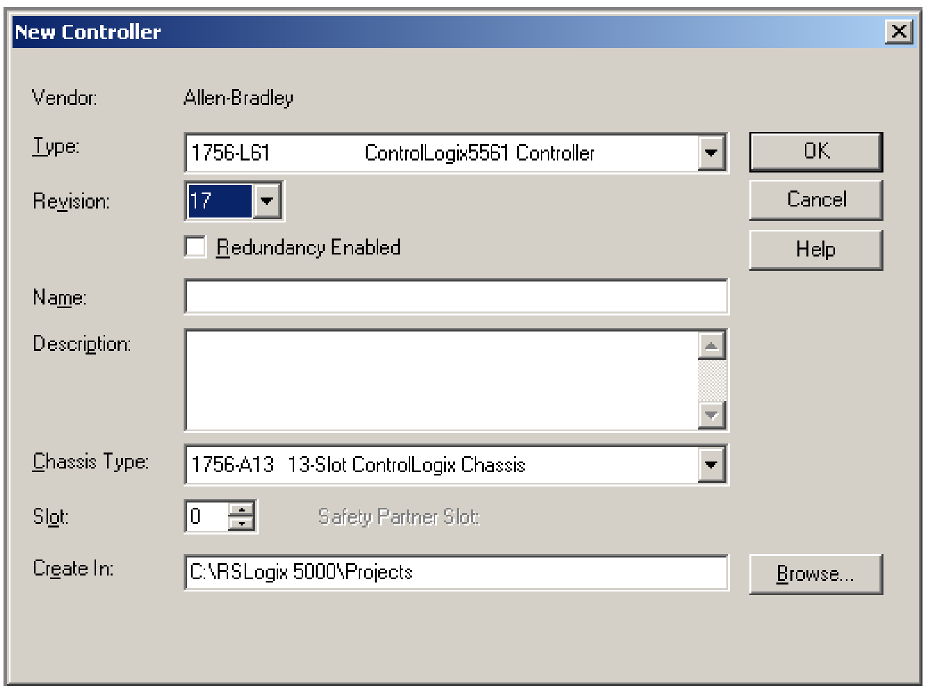
Note

When you are configuring the scanner, ensure that the ControlLogix CPU is in Program mode.

1. Make sure you have successfully completed the instructions from How to Verify Backplane Communication to ESR2/SR4 Module

2. Open RSLogix 5000 software and create a new program offline. A dialog box similar to what is shown below should appear in your screen

Figure 18: Select Controller Type Window

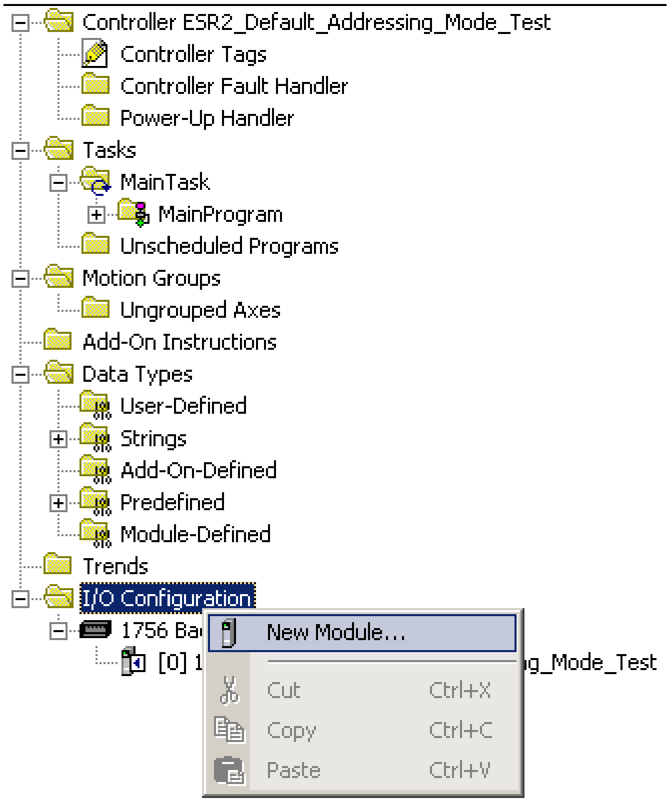


3. Select the 1756 CLX ControlLogix Controller as the Controller type and select applicable revision.

4. Select the correct Chassis type.

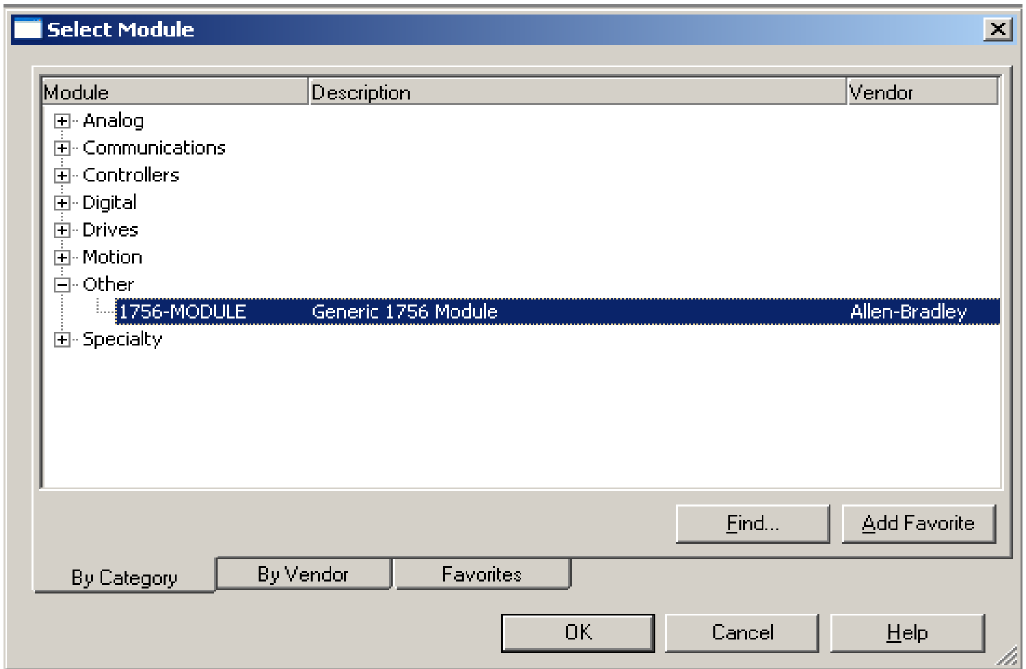
5. Fill in the controller name. Locate the ControlLogix slot number and fill it into the slot number field. Choose project location and click OK.

6. In the project tree, right-click on I/O Configuration and select New Module from the shortcut menu. See below for illustration.



7. Expand “Others” from the displayed window.

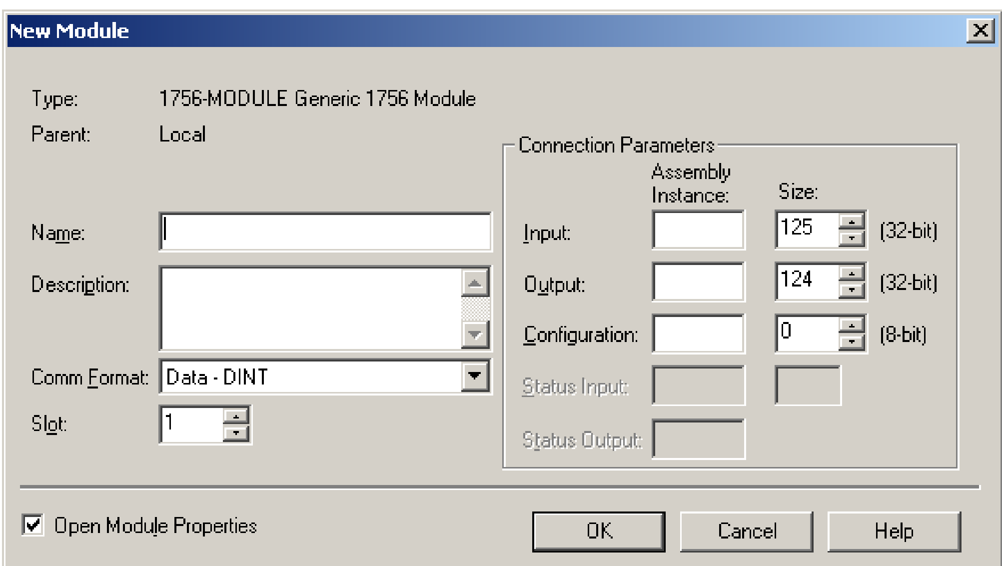
Figure 19: Select Module Type Window



8. From the Module Type list, select the Generic 1756 Module.

9. Click OK. The corresponding Module Properties window displays.

Figure 20: Module Properties Window



10. In the Name field, type in the Controller's new name.

11. Set the COMM Format to Data-INT-with status (recommended).

12. In the Slot field, set the correct slot number.

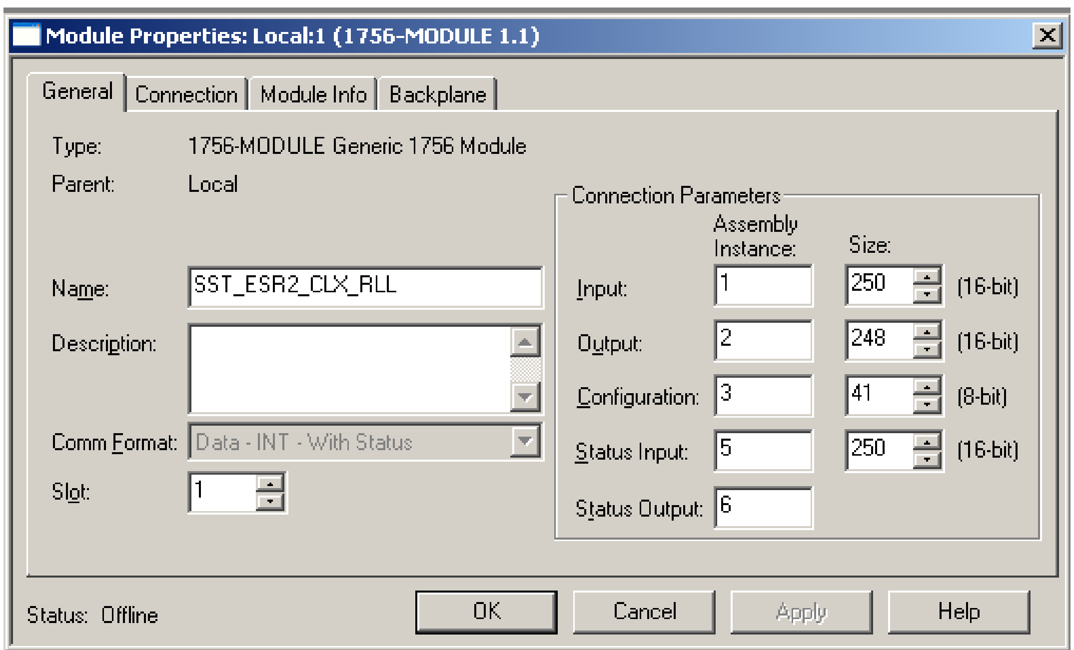
13. From the Connection Parameters frame, set the following values:

    - **Input** - Assembly Instance to 1 and Size to 250 16-bit elements
    - **Output** - Assembly Instance to 2 and Size to 248 16-bit elements
    - **Configuration** - Assembly Instance to 3 and Size to 41 8-bit elements (optional)
    - **Status Input** - Assembly Instance to 5 and Size to 250 16-bit elements
    - **Status Output** - Assembly Instance to 6

    An example screenshot below using Data INT – With Status

14. When OK is selected, the display will switch to Connection tab. Click General tab again, a similar display shown below.

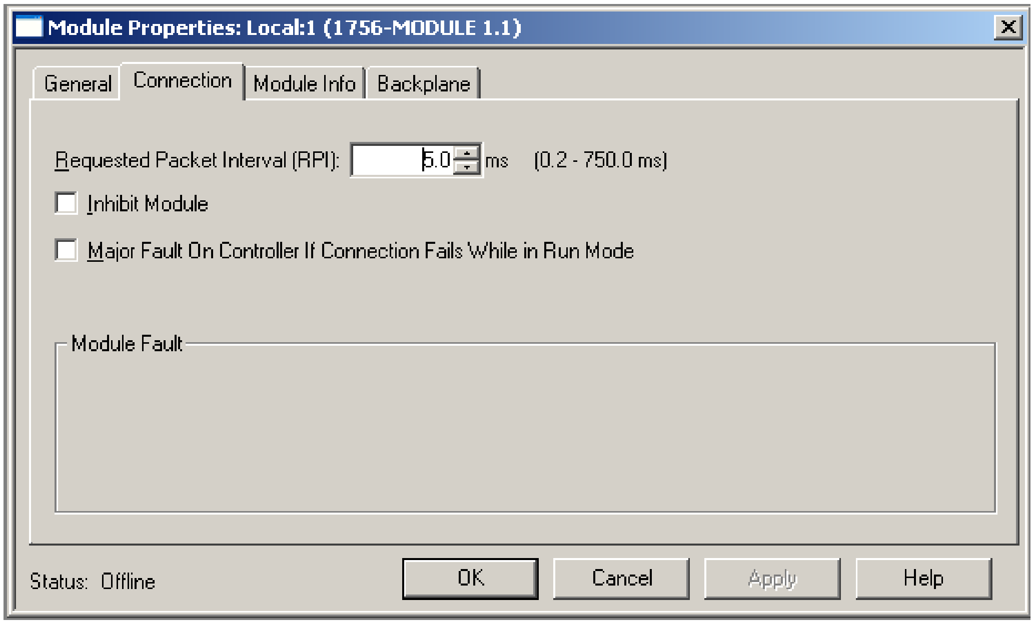
Figure 21: Module Properties Window



15. Click back to Connection tab to set the Requested Packet Interval (RPI) accordingly. This update rate should reflect the cyclic update rates set in the Console.

    e.g. if the cyclic update rates are set to 100ms each, an RPI lower than 100ms will read the same data from the database before the next 100ms cyclic update. The time between RPIs is used to allow diagnostics during runtime and to send generic CIP messages to read and write extended data from/to the database.

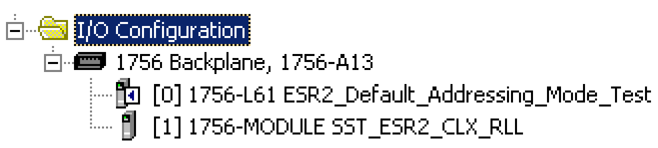
Figure 22: Module Properties, Control Program Setup



Note

An RPI value lower than 5ms may cause system performance issues.

16. Click OK and the ESR2 module is added into the IO Configuration tree.



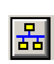
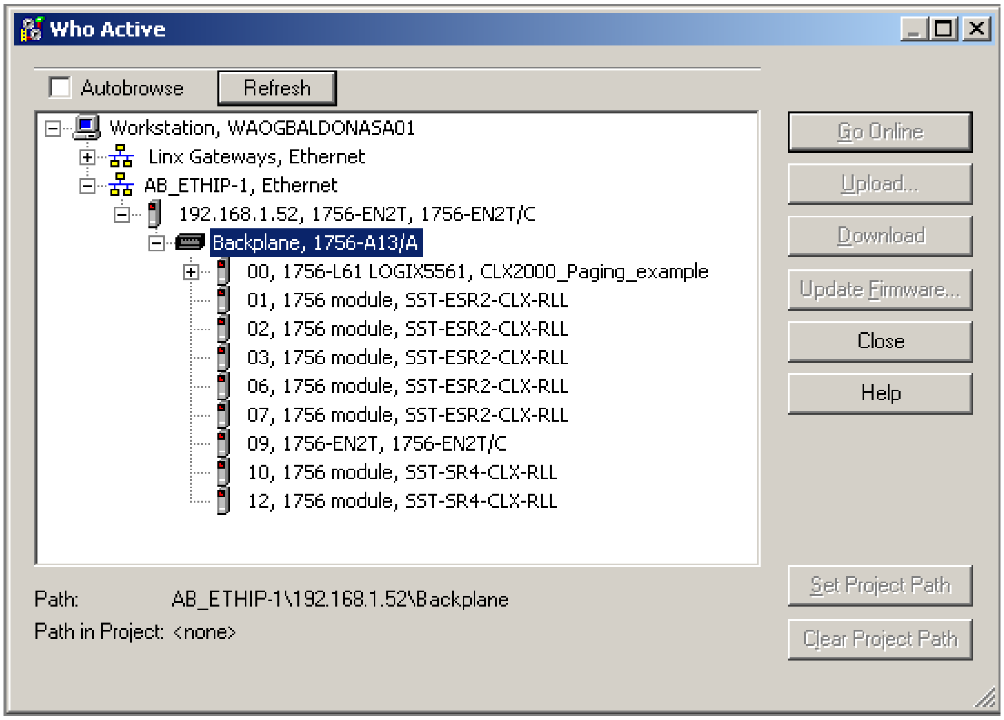
17. Next, properly configure the project path. To do this, click  in the RSLogix 5000 software, expand the Ethernet/IP driver which is configured in How to Verify Backplane Communication to our Module section. The ControllerLogix CPU which is located in slot 0 is also shown to be in slot 0 in the display.

Figure 23: RSWho Browsed Modules Display



18. Click on the slot where the ControlLogix is located, then click "Clear Project Path" if it has some value and then click the "Set Project Path".

19. Close the window.

20. Create the rest of Control program and all of the logic.

Note

You must include the sample RSLogix 5000 program, located on the CD, or refer to Section A.2, Startup Code Example, to interface to the Module.

21. Save the configuration file and download it to the CLX Module.

22. To check the IO table, double click on Controller Tags to have similar display as below.

Figure 24: Generic Profile Configuration Data Areas

− Local:1:C	{...}	{...}		AB:1756_MODUL...	
+ Local:1:C.Data	{...}	{...}	Hex	SINT[400]	ESR2 Configuration
− Local:1:I	{...}	{...}		AB:1756_MODUL...	
+ Local:1:I.Data	{...}	{...}	Decimal	INT[250]	ESR2 Input Data
− Local:1:O	{...}	{...}		AB:1756_MODUL...	
+ Local:1:O.Data	{...}	{...}	Decimal	INT[248]	ESR2 Output Data
− Local:1:S	{...}	{...}		AB:1756_MODUL...	
+ Local:1:S.Data	{...}	{...}	Decimal	INT[250]	ESR2 Satus

The above graphic shows the data areas created in the ControlLogix 5550 Controller as a result of the communication Module's Generic Profile configuration.

5.4 How to Browse the Module from the Network

This procedure is only applicable to SST-ESR2-CLX-RLL product variant.

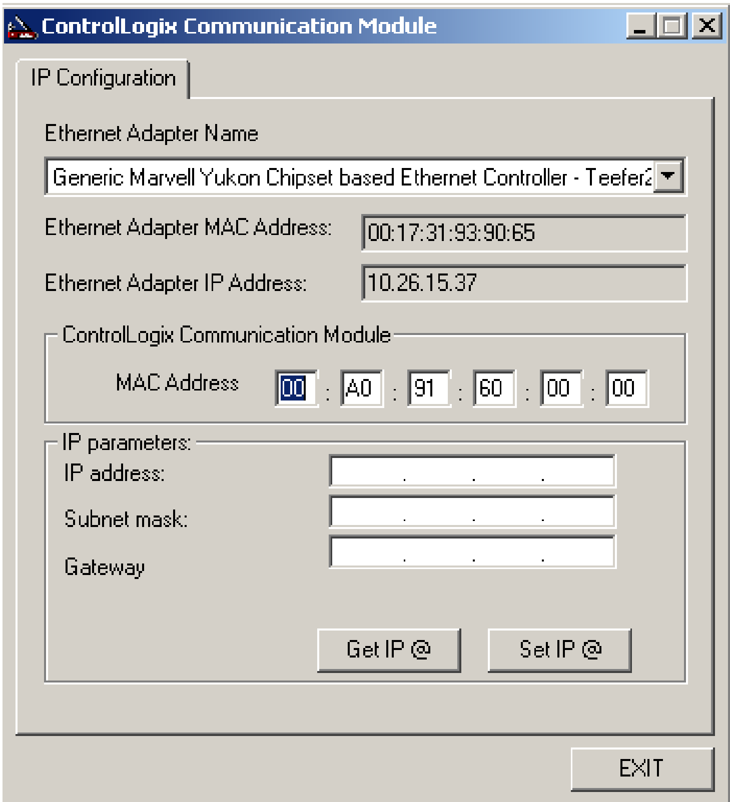
The module’s IP address must be configured before running any Ethernet communications. The following procedure must be carefully observed to have a successful initial communication to the module:

Note

The application will be sending broadcast packet to be able to retrieve the modules IP address. When the Windows Firewall in your machine is “ON”, this broadcast packets are dropped automatically, hence you will not be able to retrieve your module’s IP address and won’t be able to set it to your desired IP address. It is required that Firewall is disabled temporarily when you want to set or get the module’s IP address.

1. Go to BradCommunications > SST Backplane Communication Module > Set IP Address. The ControlLogix Communication Module dialog is displayed.

Figure 25: ControlLogix Communication Dialog



2. From the label on the side of SST-ESR2-CLX-RLL, record the MAC Address.
3. Insert the SST-ESR2-CLX-RLL into the rack and power up.
4. Connect one end of the RJ45 Ethernet cable to CH0 (bottom RJ45 Ethernet port on the card), and the other to the Ethernet card in the PC. Connect the Ethernet cable from the PC straight to the CH0 Ethernet port on the card without going through a switch or hub.
5. Click the “Get IP @” button.
6. Set the IP address to the desired value.
7. If the network gateway is used, enter a gateway address.
8. Click the “Set IP @” button.
9. Using the Windows “ping” command, check that the new IP address has been registered.

Note

To make the "ping" command work, check the PC and the SST-ESR2-CLX-RLL have IP addresses that are compatible with their respective sub-network masks. If necessary, contact the network administrator for more details.

5.5 How to Make Initial Connection to the Module

The SST Backplane Software Configuration provides different option to connect to your module. The following connection types are supported and can be used to download configuration to the module.

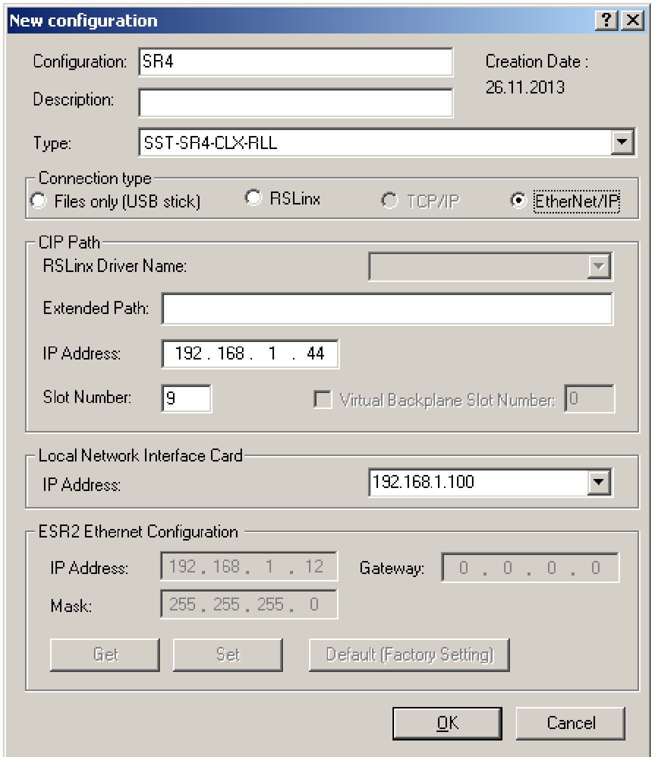
- RSLinx
- TCP/IP
- EtherNet/IP

Both SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL modules do support all connection types except for the TCP/IP connection type which only applies to the SST-ESR2-CLX-RLL variant.

5.5.1 Connecting to SST-SR4-CLX-RLL using EtherNet/IP

1. Launch the Console.
2. Configuration Manager Tool will show up as shown above using the other Connection Type.
3. Select Ethernet/IP as connection type.
4. Enter Extended path if the SST-SR4-CLX is located in a remote rack.
5. Enter the IP address of the 1756-ENBT when using the Ethernet/IP driver for local racks.
6. Enter the slot number where the SST-SR4-CLX module is located in rack.
7. Under Local Network Interface card, enter the IP address of the network interface card in PC that will be used to access the Ethernet network.
8. Click OK to save the configuration.
9. Connect to the Module by clicking the Open button under local configuration and selecting “Yes” to the message prompt that follows.
10. Refer to the sample diagram shown below that shows the configuration property when the connection type is using EtherNet/IP.

Figure 26: Configuration Property using EtherNet/IP connection type



5.5.2 Connecting to SST-ESR2-CLX-RLL using TCP/IP

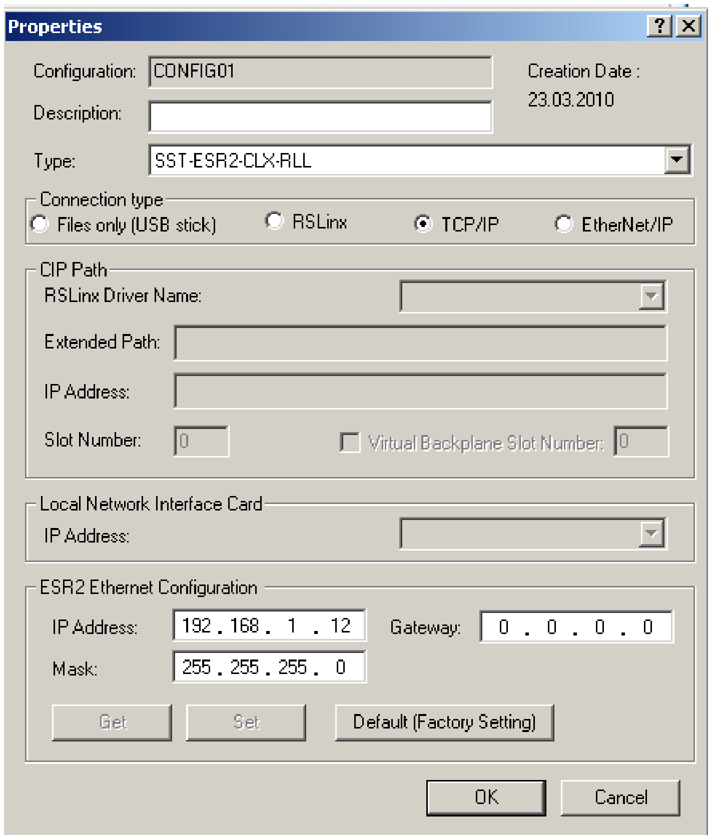
To initiate connection to the module, follow the steps provided below.

Note

We will only start a connection. We will use all default configuration values for now. It is assumed that before making a connection to the module that it has been browsed successfully in the network and the module is in the same network IP address range with the machine running BCMS.

1. Before proceeding, make sure that How to Browse the Module from the Network section has been completed successfully.
2. Launch the Console Application from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.
3. From the Configuration Manager, select a configuration and click “Property”
4. Configuration Property will show up as shown below.

Figure 27: Configuration Property using TCP/IP connection type



5. Connection type is TCP/IP

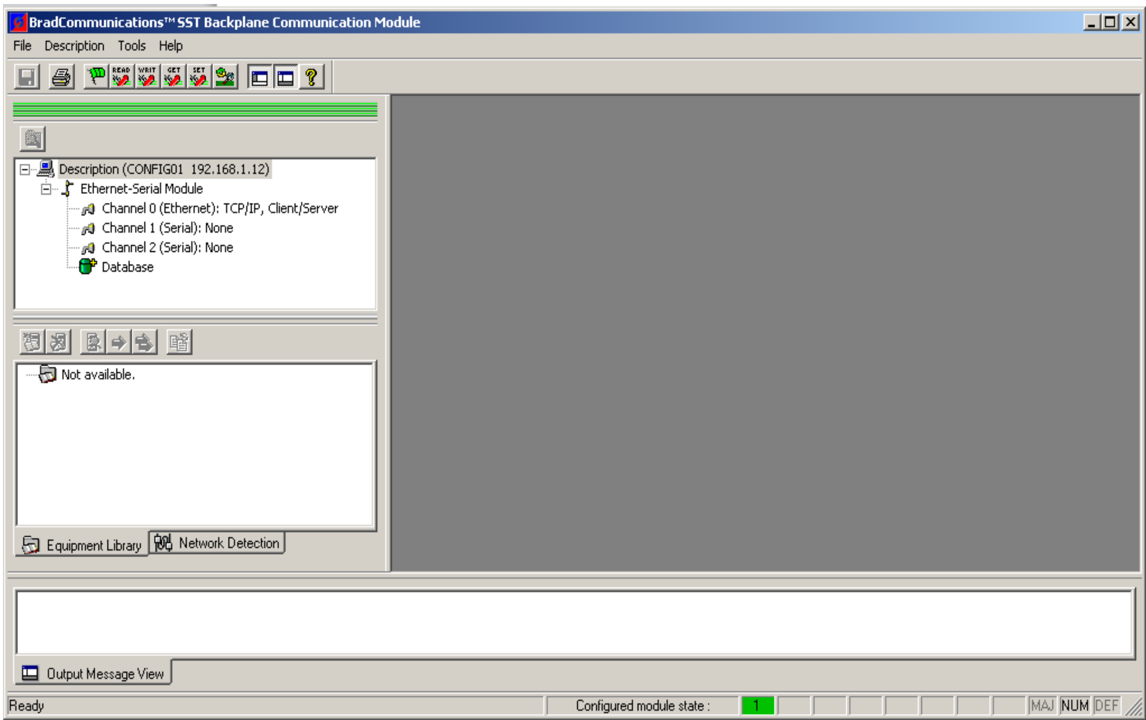
6. Change the IP address to the module's IP address if different. (Shown in the module's display)

7. Click "Ok" and then "Open"

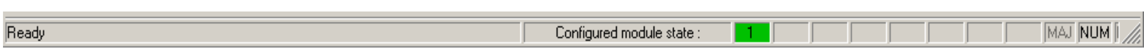
8. Click "Yes" when the window below comes out. If you're using Windows 7, the Warning message below will be displayed in a different dialog box format.

9. The Console Configuration Tool will be displayed next.

Figure 28: Console Main Screen using TCP/IP connection type



10. The Status Indicator Section should display either of the two shown below.



or

11. The module must be connected.

12. If Configured module state shows different color, the module is not connected properly. Check General Status Messages to understand the problem and follow suggested procedure to resolve it.

13. Close the application

5.5.3 Connecting to SST-SR4-CLX-RLL using RSLinx

To initiate connection to the module, follow the steps provided below.

Note

We will only start a connection. We will use all default configuration values for now.

1. Launch the Console Application from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

2. From the Configuration Manager, select a configuration and click "Property"

3. Fill in the configuration parameters as listed below:

    Configuration Name: SR4

    Communication Module Type: SST-SR4-CLX-RLL

    Connection Type: RSLinx

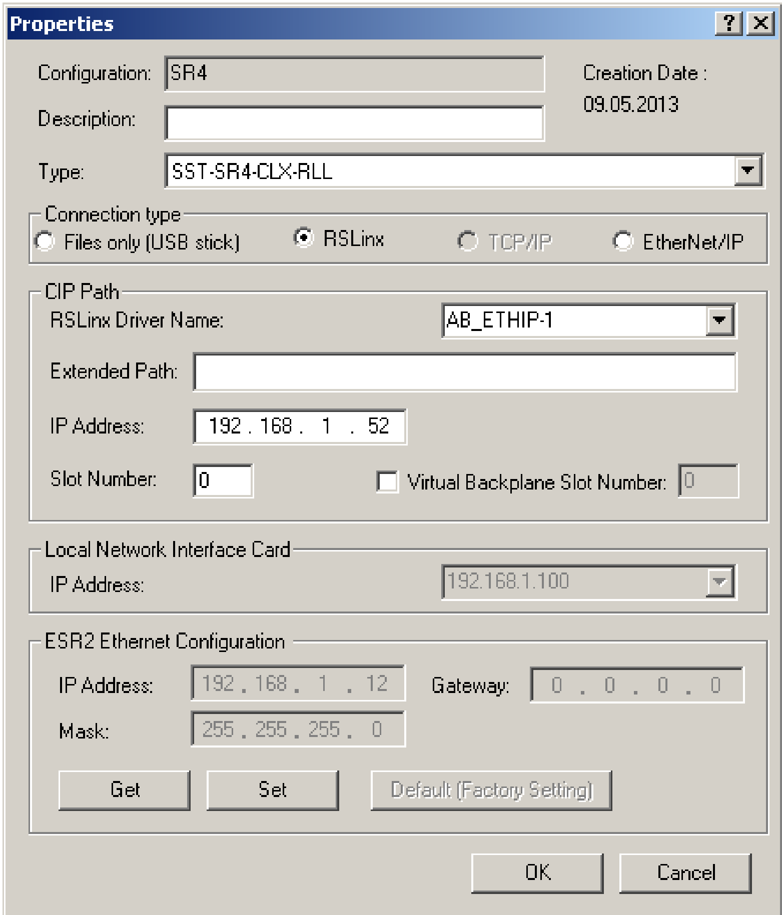
    RSLinx Driver Name: Configured in How to Verify Backplane Communication to ESR2/SR4 Module

    IP Address: (IP address of the EtherNet/IP Bridge)

    Slot Number: Slot number of SST-SR4-CLX-RLL module in the ControlLogix rack.

After filling in the entries, the values with the exception of the slot number and IP addresses will be as shown below.

Figure 29: Configuration Property using RSLinx connection type

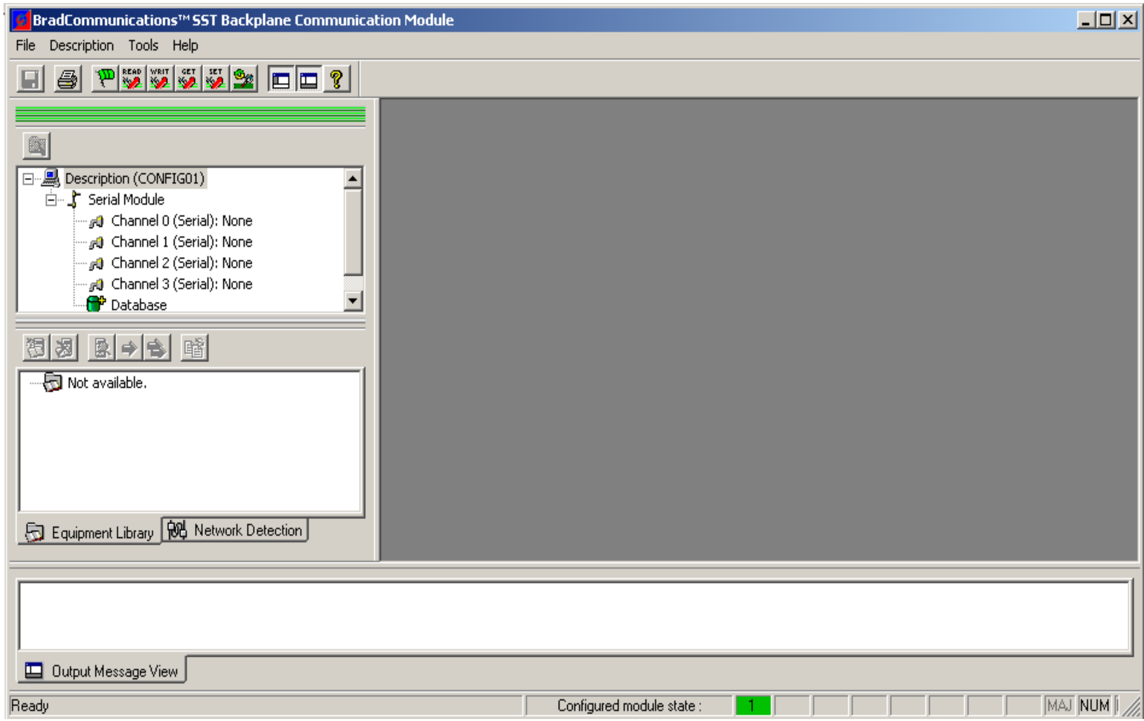


4. Click “Ok” and then “Open”

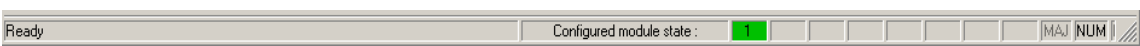
5. Click “Yes” when the window below comes out.

6. The Console Configuration Tool will be displayed next.

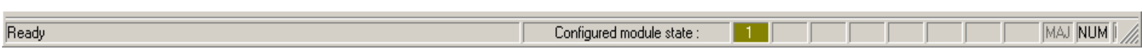
Figure 30: Console Main Screen using RSLinx connection type



7. The Status Indicator Section should display either of the two shown below.



or



8. The module must be connected.

9. If Configured module state shows different color, the module is not connected properly. Check General Status Messages to understand the problem and follow suggested procedure to resolve it.

10. Close the application.

5.6 How to Initialize the Module

The module uses the current configuration available in its system when it boots up. The configuration may be different to the active configuration selected from the configuration manager. To initialize the module with an active configuration, please follow the steps.

1. Ensure there is a connection to the module before starting initialization.

2. Check the Status Bar to see if it is connected. The module state may show the same color as shown below.

3. Locate the PCInit  application from the Console Application Menu Bar

4. Run the application. User will be prompted with dialog boxes to ensure user want to initialize the module. Initialization takes about 1-2 minutes if just downloading configuration files and using connection type TCP/IP. It takes more than 2 minutes to download configuration files when using RSLinx or EtherNet/IP connection types.

   If initialization is successful, the configured module state color indicator will be shown as below.

   If color remains the same as shown in Step 2, initialization is not successful.

5.7 How to Change Database Addressing Mode

The following sub-section provides instruction how to change the database addressing mode

Note

If there are cyclic functions created in the configuration, changing the database addressing mode will affect those cyclic functions.

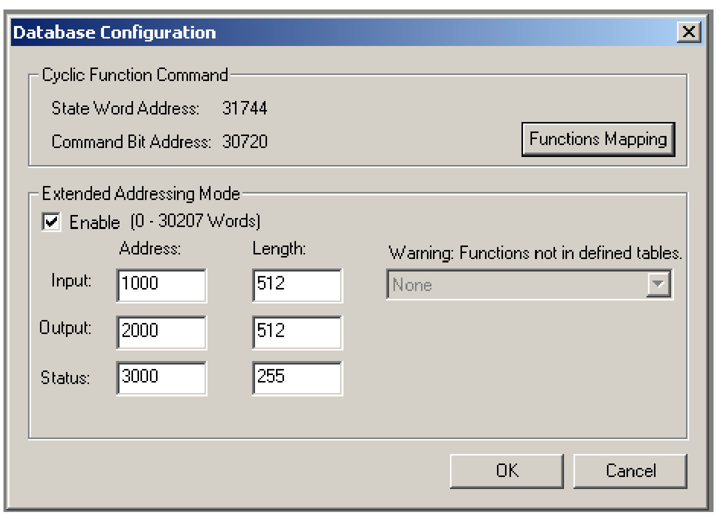
5.7.1 Changing to Default Addressing Mode

When a new configuration is created, the database is set to default addressing mode.

If the active configuration is configured in extended addressing mode, to change it to default addressing mode, follow the steps below.

1. Launch the Configuration Manager form Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.
2. Select the Configuration that you want to use and then click “Open”. When Figure 35 is displayed, go to the “Configuration Description Area”, locate and double-click Database .
3. The database configuration window will be displayed as shown below.

Figure 31: Changing Database from Extended to default



4. Uncheck the Extended Addressing Mode
5. Click OK and save the modification then download the configuration to re-initialize the communication module into default addressing mode.

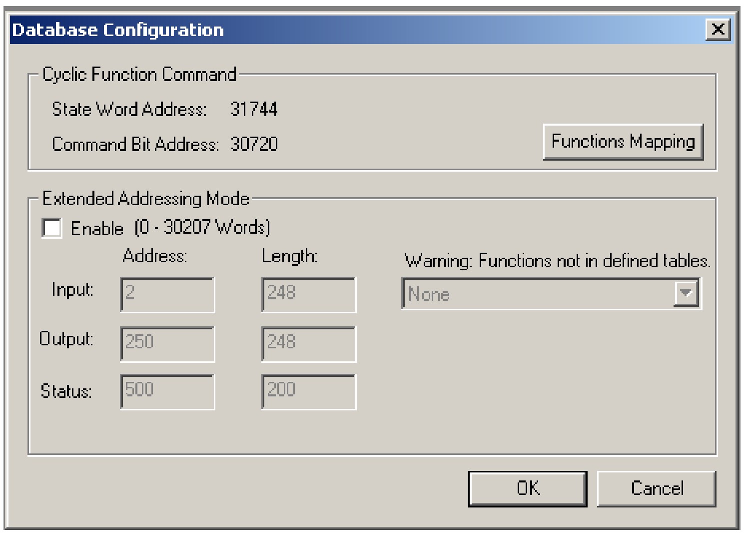
5.7.2 Changing to Extended Addressing Mode

1. Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

2. Select the Configuration that you want to use and then click “Open”. When Figure 35 is displayed go to the “Configuration Description Area”, locate and double-click Database .

3. The database configuration window will be displayed as shown below.

Figure 32: Changing Database from Default to Extended



4. Check the Extended Addressing Mode

5. Enter the database addressing parameters as indicated below. These are sample database configuration parameters, it can be changed later according to your requirements.

    a. INPUT Table address: 0; length: 1600

    b. OUTPUT Table address: 1600; length:1600

    c. STATUS Table Address: 3200; length: 1024

6. Save the modification, download the configuration to re-initialize the module using extended addressing mode

6

Configuration Examples

Chapter Sections:

- Module’s Configuration
- Siemens S7/S5 Messaging Configuration
- Modbus TCP Messaging Configuration
- Modbus Serial Messaging Configuration
- Configuration with Cyclic Function
- Configure the module in Listen-Only Mode
- Configure the module using the AOI
- Configure the module using the AOP
- Configure the module in Remote Rack via ControlNet Bridge
- Dynamic Triggering of Cyclic Function

The module should have a working configuration before it can run any protocol messaging.

6.1 Module's Configuration

Configuration is a set or group of module parameters saved into multiples files, downloaded into the module to be able to run specific functionality. Some of these files have default values and can be modified from the Console application.

Before configuring the network protocols, the type of messaging protocols run in the module must have been identified. The communication module variant to understand what connection type is available for the module has also been identified.

The following connection types are available to download configuration to the module.

- RSLinx
- TCP/IP
- EtherNet/IP
- USB Port

A configuration brings together:

- An IP address (for SST-ESR2-CLX-RLL only). For details see Section 5.4
- Protocol configuration parameters
- A definition of cyclic functions, which access the input/output data
- The configuration listings

It is represented on the hard disk as a set of files stored in a single directory, and identified by:

- Its name
- Its description
- The type of Backplane
- The IP address of the target Backplane or RSLinxs path, and the slot address

The active configuration is selected by running the Configuration Manager systematically before starting the Console.

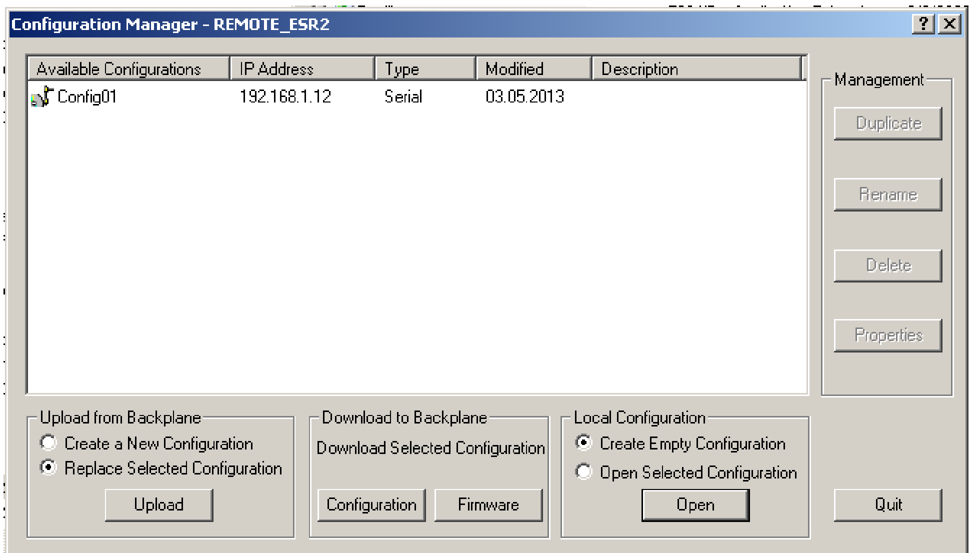
The following sub-sections describe how to create, load, download upload configuration using BCMS. The SST-ESR2-CLX-RLL variant is used in this section. The procedure can be followed similarly to configure a SST-SR4-CLX-RLL variant.

6.1.1 Creating a New Configuration for SST-ESR2-CLX-RLL

A configuration can be created either from scratch, copied from an existing configuration or uploaded from a communication module.

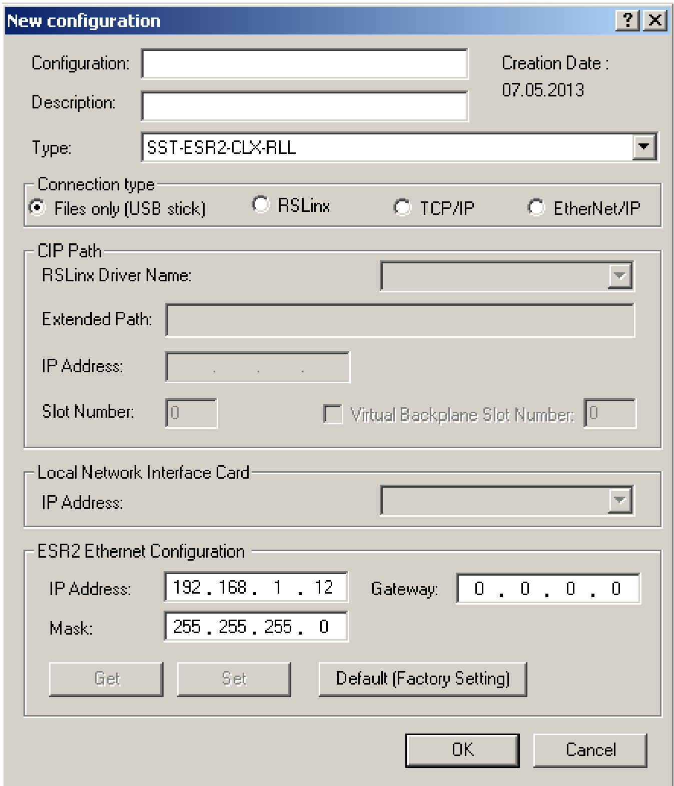
To start creating a new configuration, launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console. The diagram below will show up.

Figure 33: Configuration Manager Main Screen



1. In Local Configuration, select Create Empty Configuration and then click the Open button. Another screen will show up similar to the one shown below.

Figure 34: Configuration Property



2. Fill in the necessary parameters as indicated below:

    a. Configuration Name: ESR2

    b. Description (optional): 1st Configuration

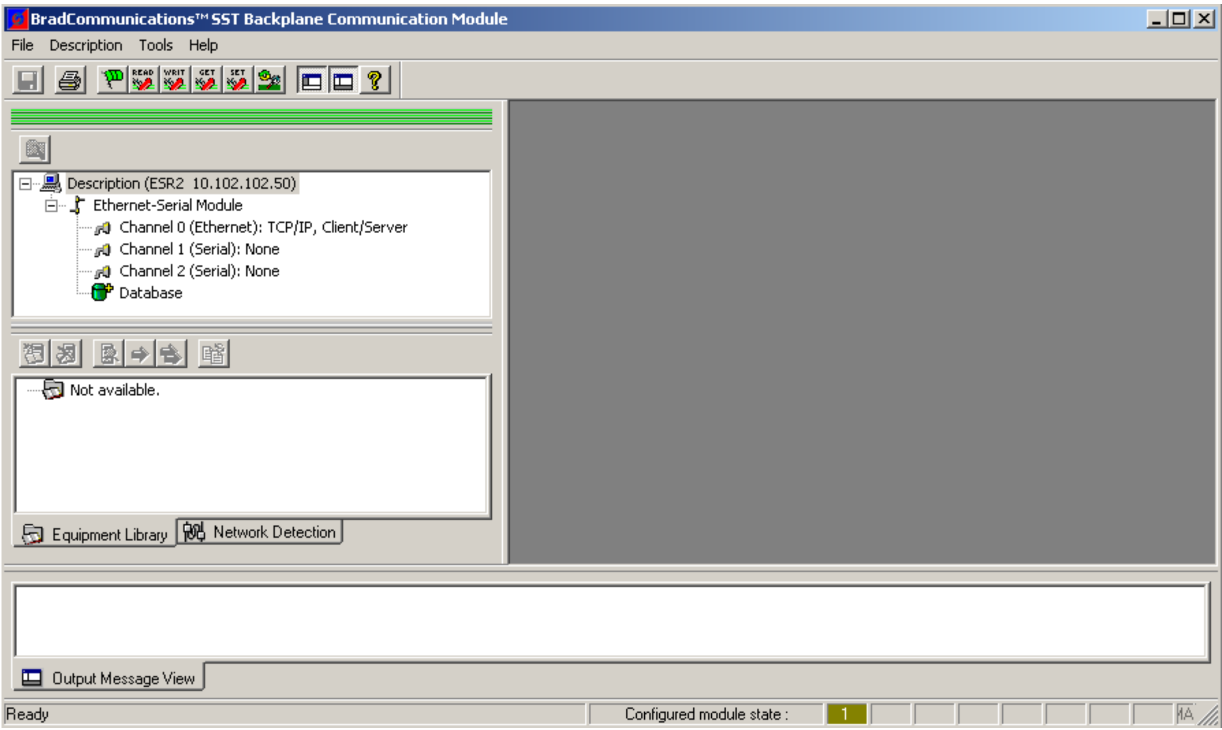
    c. Type (Choose the right module variant): SST-ESR2-CLX-RLL

    d. Choose the connection type (this procedure is using TCP/IP connection)- TCP/IP

    e. Change the IP address to the module's IP address 10.102.102.50 (IP address could have been modified in How to Browse the Module from the Network). To check the IP address, verify it from the module's display. The LCD alternately display the module configuration name and its IP address)

3. Click OK. The Console application will start as shown below.

Figure 35: Console Application with newly created configuration



At this point, the changes made to the configuration are related to connection parameters only.

4. Close the application.

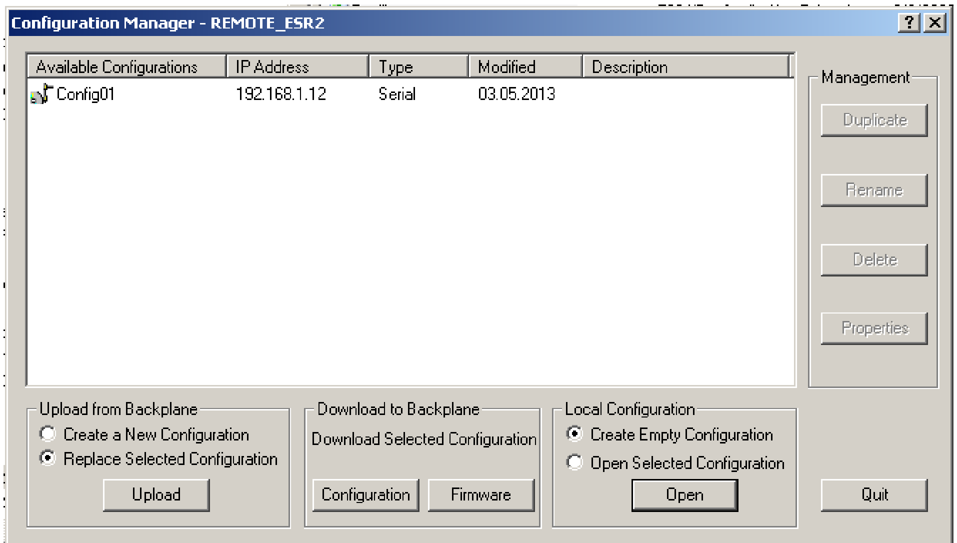
6.1.2 Creating a New Configuration for SST-SR4-CLX-RLL

A configuration can be created either from scratch, copied from an existing configuration or uploaded from a communication module.

To start creating a new configuration, launch the Configuration Manager from

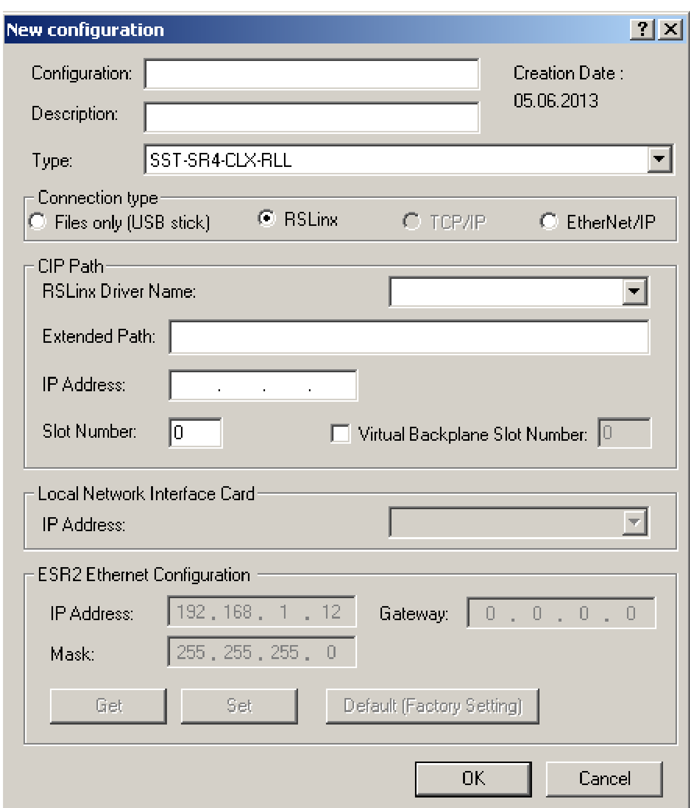
Start->All Programs->BradCommunications->SST Backplane Communication Module->Console. The diagram below will show up.

Figure 36: Configuration Manager Main Screen



1. In Local Configuration, select Create Empty Configuration and then click the Open button. Another screen will show up similar to the one shown below.

Figure 37: Configuration Property



2. Fill in the configuration parameters as listed below:

   Configuration Name: SR4

   Communication Module Type: SST-SR4-CLX-RLL

   Connection Type: RSLinx

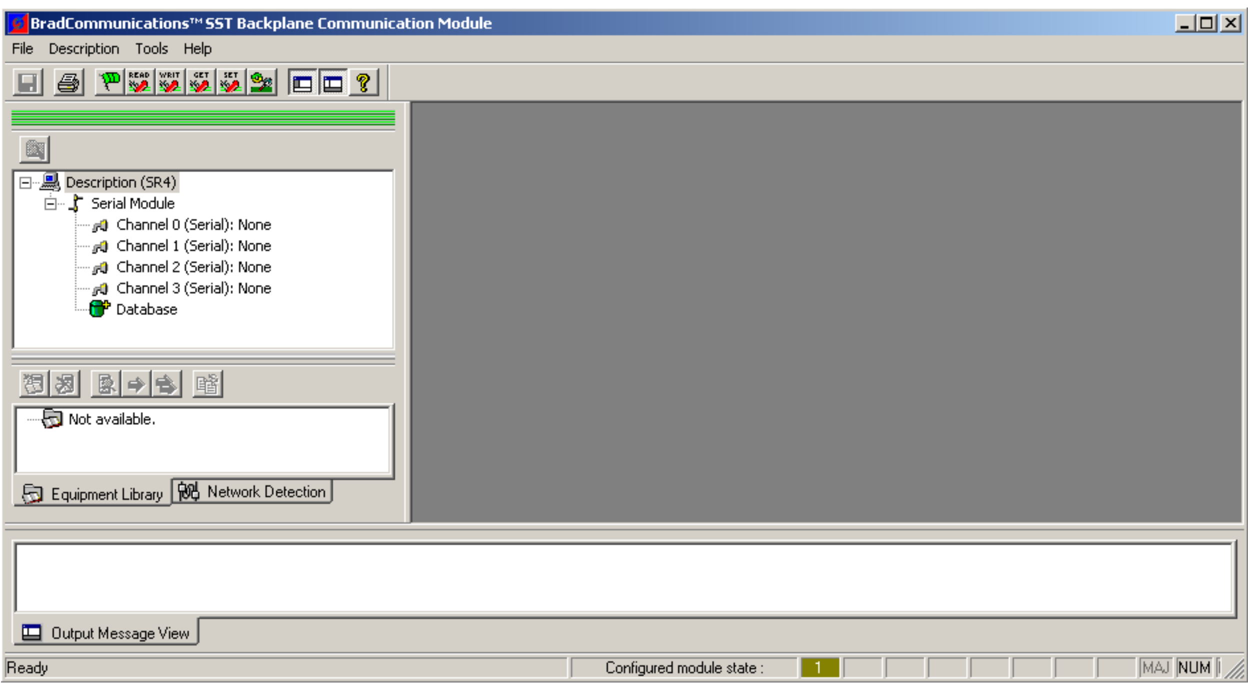
   RSLinx Driver Name: Configured in How to Verify Backplane Communication to ESR2/SR4 Module

   IP Address: 192.168.1.44 (IP address of the EtherNet/IP Bridge)

   Slot Number: 10 (Slot number of SST-SR4-CLX-RLL module in the ControlLogix rack).

3. Click OK. The Console application will start as shown below.

Figure 38: Console Application with newly created configuration



At this point the changes made to the configuration are related to connection parameters only.

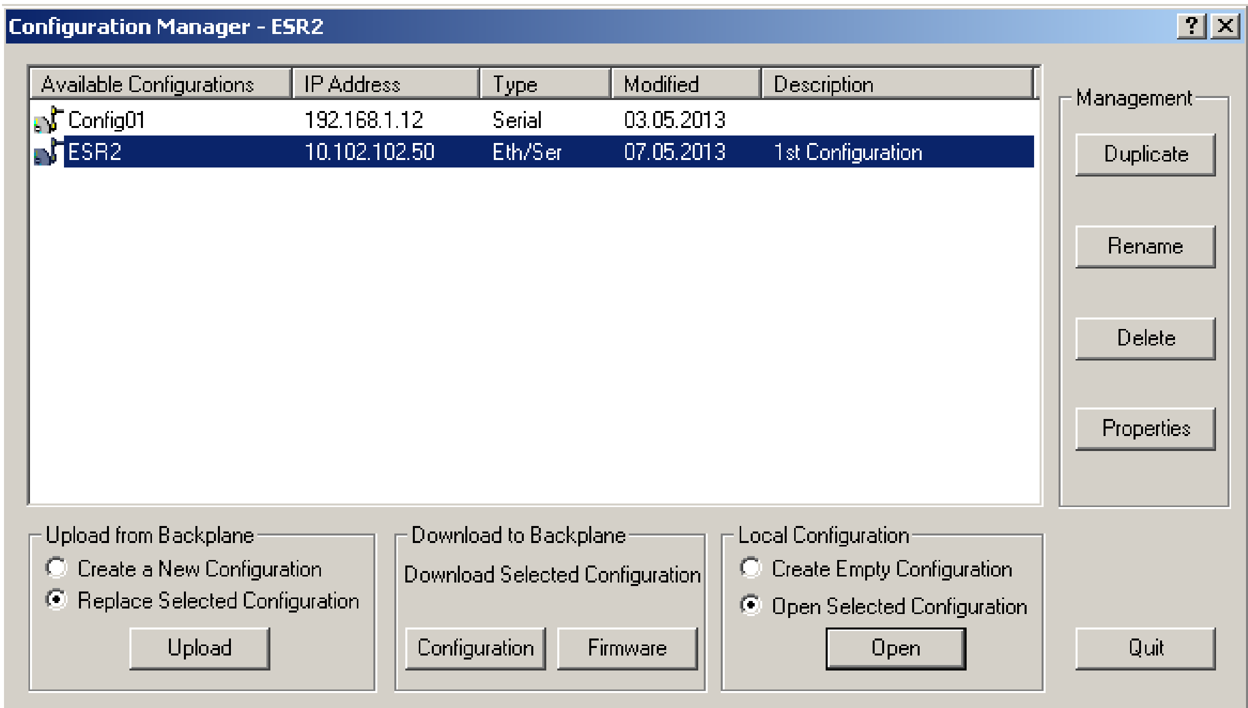
4. Close the application.

6.1.3 Loading an existing configuration

Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

The new configuration created in Creating a New Configuration for ESR2-CLX-RLL will be listed as shown below. If the created one for SST-SR4-CLX-RLL following instruction from Creating a New Configuration for SST-SR4-CLX-RLL then module type will be "Serial".

Figure 39: Configuration Manager with New Configuration Displayed



Select the new configuration created and click the "Open" button to launch the console application using the selected configuration. The diagram shown in Figure 38 will be displayed.

A loaded configuration means the console application has loaded the selected configuration and will be updated when changes is made from the console application tools. The module will have to be upgraded with the active configuration to be able to run the new configuration parameters.

Upgrading the modules configuration can be done in two ways: (a) Using PCInit and (b) USB port. See the next two sections for the procedure.

6.1.4 Downloading Configuration to the Module

6.1.4.1 Using Configuration Manager

Complete the steps from either Section 6.1.1 or Section 6.1.2. You can proceed with the download by following the steps below:

1. Ensure the module is connected before downloading the configuration.

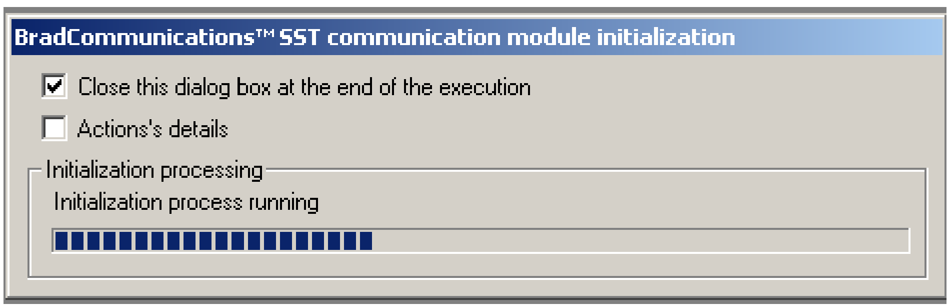
2. Check the Status Bar to see if it is connected. The module state must show the same color as shown below.

3. Locate the PCInit  application from the Console Application Menu Bar

4. Run the application. User will be prompted with dialog boxes to ensure if continuing to initialize the module.

5. Click OK. As download starts, the console will show the progress of the download similar to the screen displayed below.

Figure 40: Download Progress Window



6. Initialization takes about 1-2 minutes if just downloading configuration files and using connection type TCP/IP. It takes more than 2 minutes to download configuration files when using RSLinx or EtherNet/IP connection types.

   If initialization is successful, the configured module state color indicator will be shown as below.

   If color remains the same as shown in Step 2, initialization is not successful.

6.1.4.2 Using USB Port

To upgrade the module configuration using the USB configuration port, follow these steps:

Note

If the module does not currently support USB, the backplane and protocol firmware on the card needs to be updated before it can update the module configuration via USB. To do this, update the .ss4 firmware and the protocol firmware as per section 9.2.

1. Confirm the USB stick is using FAT32 file system. If not then use another USB stick that is or backup existing one and reformat it as FAT32.

2. Create a relevant configuration using the Configuration Manager, or select an existing configuration from CommonAppDataFolder\BradCommunications\SST Backplane Communication Module\config.

   If a new configuration is created, it can be found in the same directory. This configuration should be different from the one that's running.

   Note

   In Windows XP, the CommonAppDataFolder is Documents and Settings > All Users >Application Data folder

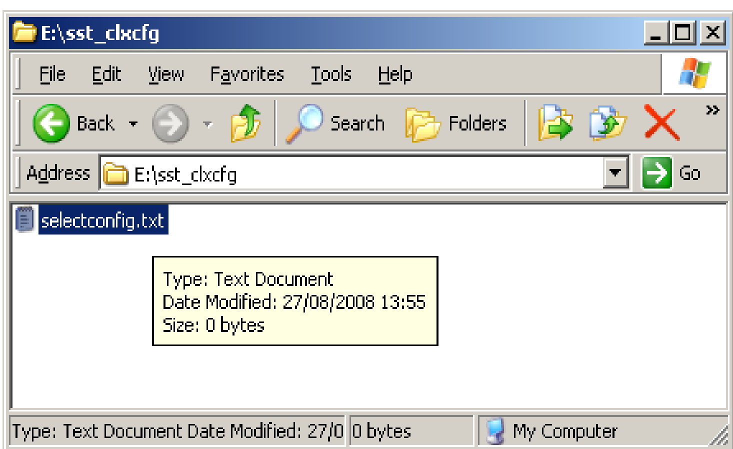
   In Windows 7, the CommonAppDataFolder is the ProgramData folder

3. On the USB stick, do the following:

   - Check if the "sst_clxcfg" directory exists. If not, create it and ensure that it doesn't contain a folder named "**oldconfig**".

   - Inside the directory, check if selectconfig.txt is available. If not, create it using Notepad.

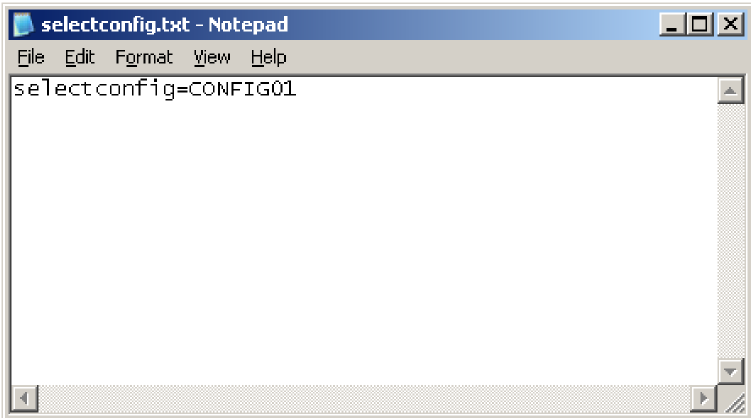
Figure 41: SelectConfig.txt, Inside Sst_clxcfg Folder



4. In selectconfig.txt, insert "selectconfig=xxxxx", where "xxxxx" represents the name of the configuration folder to be downloaded to the module.

   There should be NO escape character after the configuration name (e.g., "\n" for new line, or "\r" for carriage return).

Figure 42: SelectConfig.txt, with CONFIG01 as Selected Configuration



5. Copy the configuration into the sst_clxcfg folder and ensure that the folder name matches that in the selectconfig entry.

Figure 43: Configuration Directory

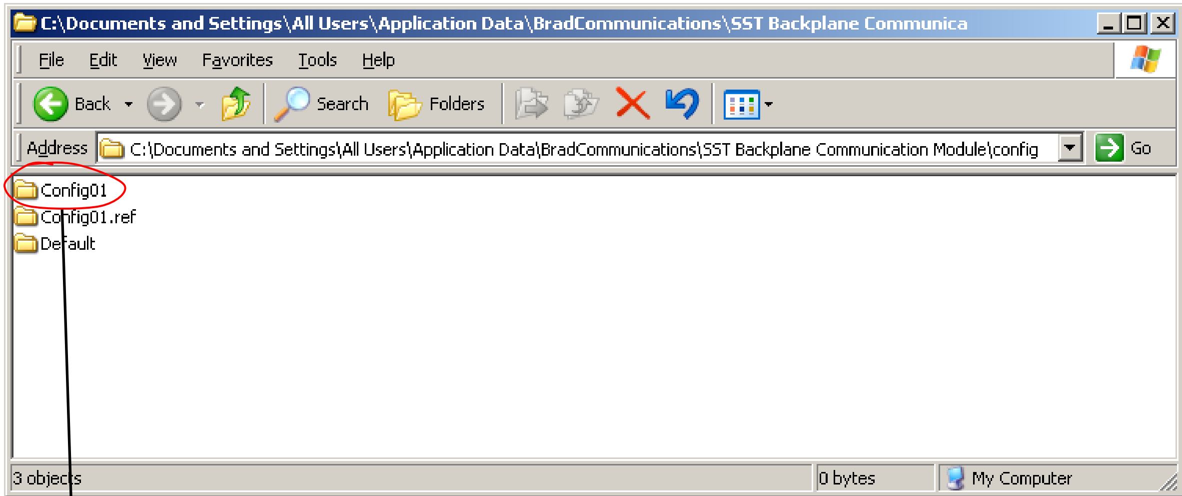
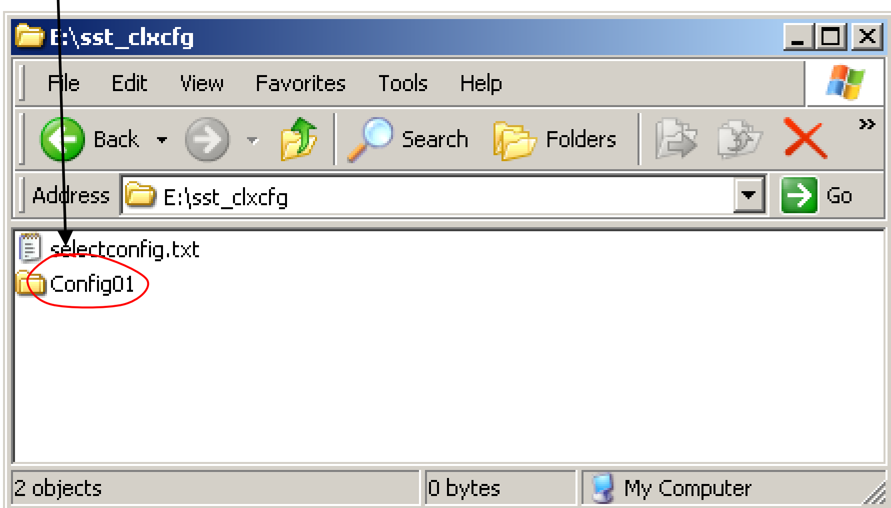


Figure 44: Selected Configuration, Coped into SST_clxcfg



6. Ensure that the configuration to be downloaded matches the module card type.

7. Plug the USB stick into the configuration port. A message “Reset the card to start update ” will appear on LCD display. Reset the card. The message “reading files…” will appear and then transition to “backup in progress…”. The existing configuration will be copied into a directory named “**oldconfig**” inside sst_clxcfgbefore the upgrade starts.

8. Ensure that “backup in progress…” and “update in progress…” are displayed during the backup and update, respectively. If the module is connected to a controller while the update is executing, the messages will only be displayed as the process begins.

   The display will change to CONNBLK until the update completes. When the update is complete the message “Update complete, remove USB stick then reset the card” is displayed. Remove the USB stick and reset the card.

If Protocol firmware or backplane firmware existed in configuration directory, their extensions will be renamed to UPD instead of OUT to indicate all task files were updated on module.

9. Remove the USB stick.

6.1.5 Backup the module's configuration

6.1.5.1 Using Configuration Manager

1. Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

2. When Figure 39 is shown in the screen, locate the Upload from Backplane selection, select between Create or replace a configuration. Be careful when using this selection as the selected configuration may get overwritten. Use a different configuration name with description to associate it with a configuration backup.

6.1.5.2 Using USB port

To back up of the module's running configuration using USB configuration port, follow these steps:

1. On the USB stick, do the following:

   - Check if the "sst_clxcfg" directory is available. If not, create it and ensure that it doesn't contain a folder named "**oldconfig**".

   - Inside the directory, check if selectconfig.txt is available. If not, create it using Notepad.

2. In selectconfig.txt, "selectconfig=" should be left blank, and there should be NO escape character after the equal sign (e.g., "\n" for new line, or "\r" for carriage return).

3. Plug the USB stick into the configuration port, and reset. The running configuration will be copied into the USB stick, and "backing up config" and will be displayed.

4. When "backup done" is displayed, remove the USB stick.

Note

Only back up/update with the USB stick when the system goes into reset.

6.2 Siemens S7/S5 Messaging Configuration

6.2.1 Configure the Module as Siemens Server Equipment.

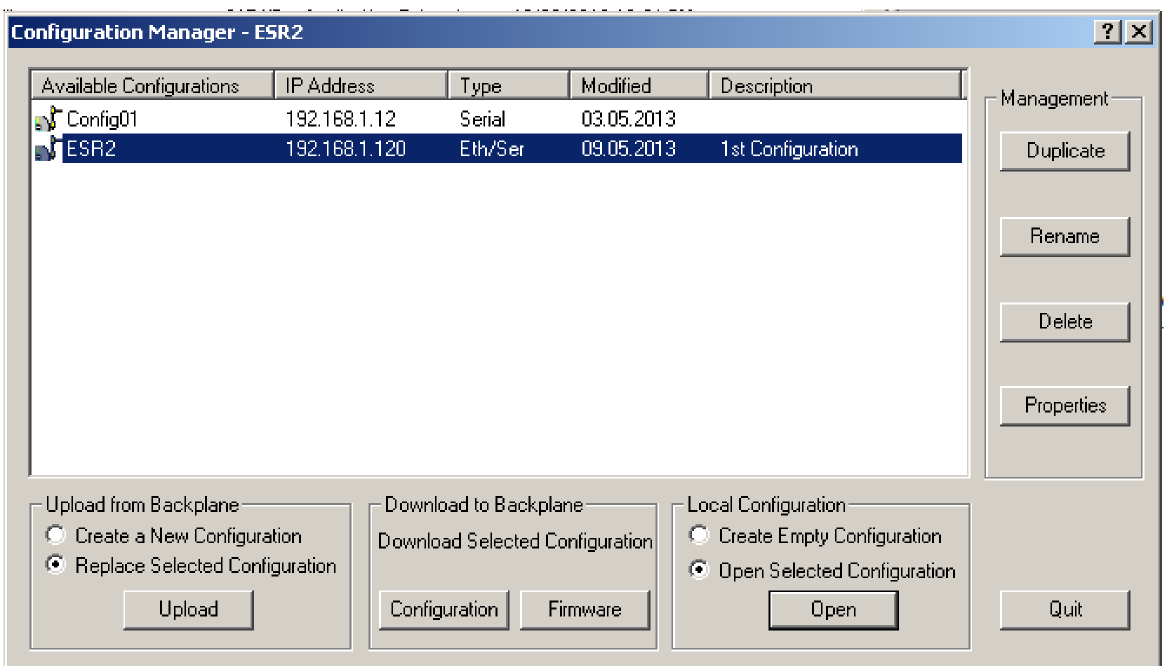
This product feature is only available in the SST-ESR2-CLX-RLL variant. Before proceeding in this section, the module must have a proper IP address configuration, it can be connected using the procedure in How to Browse the Module from the Network section. The module by default is configured in server configuration. To define what protocol messaging that the TCP port runes, follow the steps below:

1. In this sample configuration, the module will be configured as server equipment to Siemens PLCs running in client configuration.

2. Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

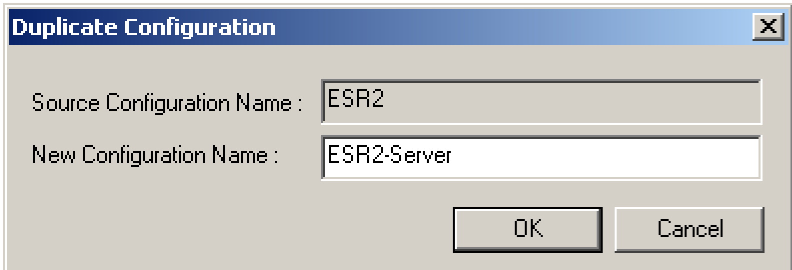
3. Select the newly created configuration from Creating a New Configuration section, and copy the created configuration into a different configuration name so the first configuration can be used as reference.

Figure 45: Configuration Manager



4. Click on “Duplicate” and name the new configuration as ESR2-Server

Figure 46: Duplicate Configuration

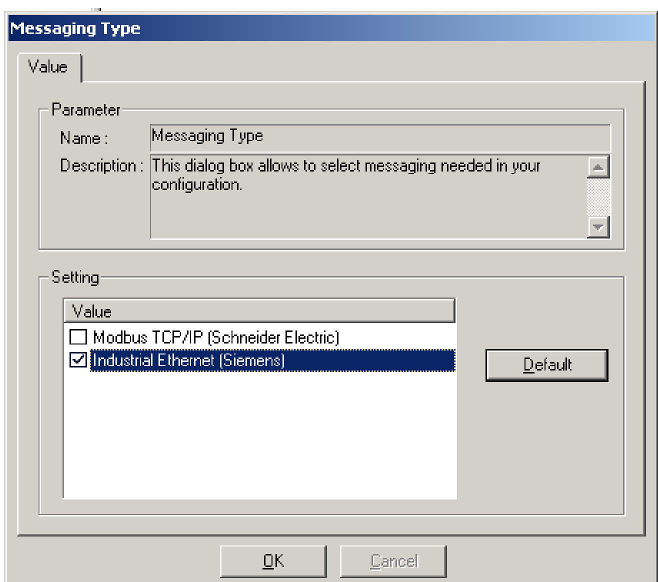


5. Click OK and select the newly duplicated configuration and click “Open”. The Console will use the new configuration and the new configuration name will be displayed in the “Configuration Description Area”.

6. Make sure the module is connected. Refer to Console Status Bar Information section which shows the status information of the communication to the module.

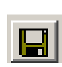
7. In the “Configuration Description Area”, locate then right click the Channel 0 (Ethernet):TCP/IP, Client/Server. The Messaging Type dialog box appears.

Figure 47: Messaging Type Selection Window



8. Select “Industrial Ethernet (Siemens)”

9. Confirm the modification by clicking OK.

10. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

11. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

12. If download is successful, the module is now ready to communicate to the Siemens PLC.

6.2.2 Configure the Module as Siemens Client Equipment

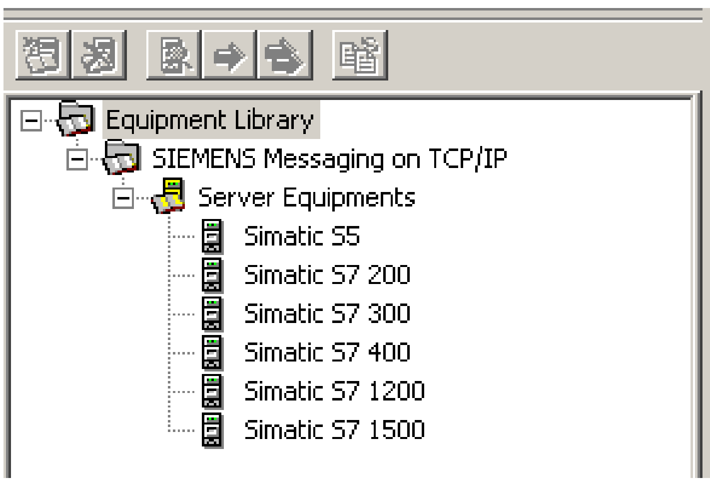
This product feature is only available in the SST-ESR2-CLX-RLL variant. To configure the module as Client Equipment, follow the procedure listed below:

1. Complete the procedure 2-4 from Configure the Module as Siemens Server Equipment. Name the new configuration to “ESR2-Client”.

2. Then continue procedure 5-9 from Configure the Module as Siemens Server Equipment.

3. The Console application Equipment Library area will have similar display as shown below.

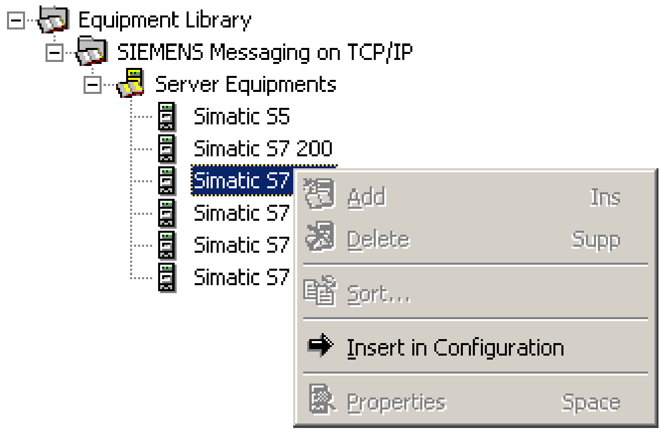
Figure 48: Siemens Equipment Library Selection



4. The supported Siemens PLCs are listed above.

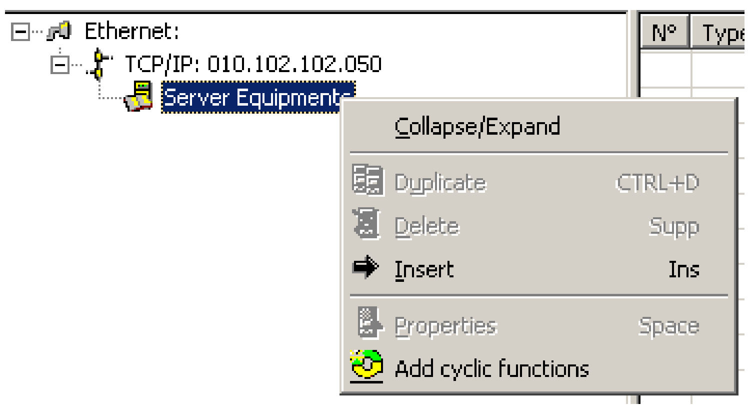
5. Add S7 300 Equipment, click on the selected PLC and choose any of the following methods.

a. Right click the selected PLC and click on “Insert in Configuration”, as shown here



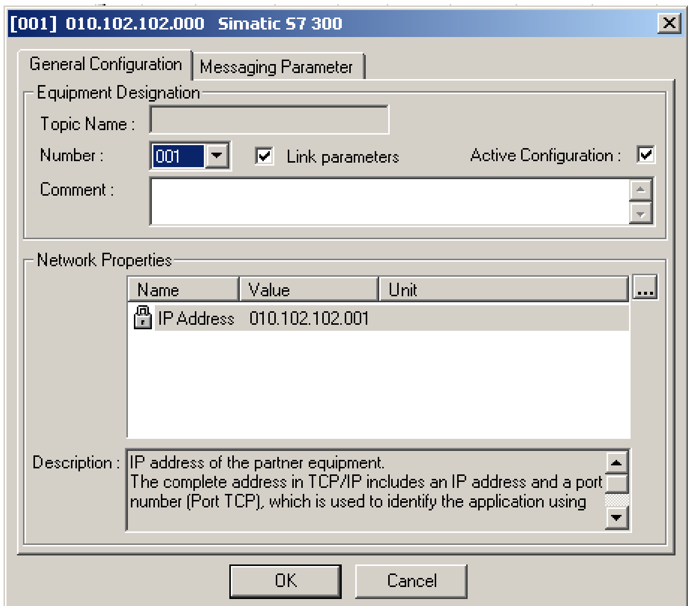
b. Click on the 

c. In the “Protocol View area”, right click the Server Equipment and click “Insert”, as shown here.



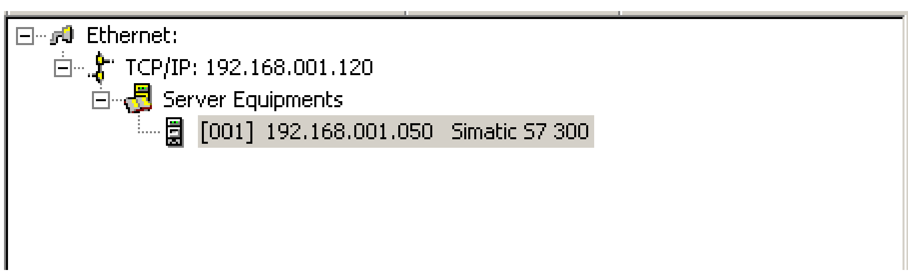
Any of the described method will display the screen below.

Figure 49: Server Equipment Configuration parameter



6. Change the IP address to Siemens PLC running as Server equipment IP address (PLC's IP address is 192.168.1.50)

7. Click OK. The protocol view area as shown below.

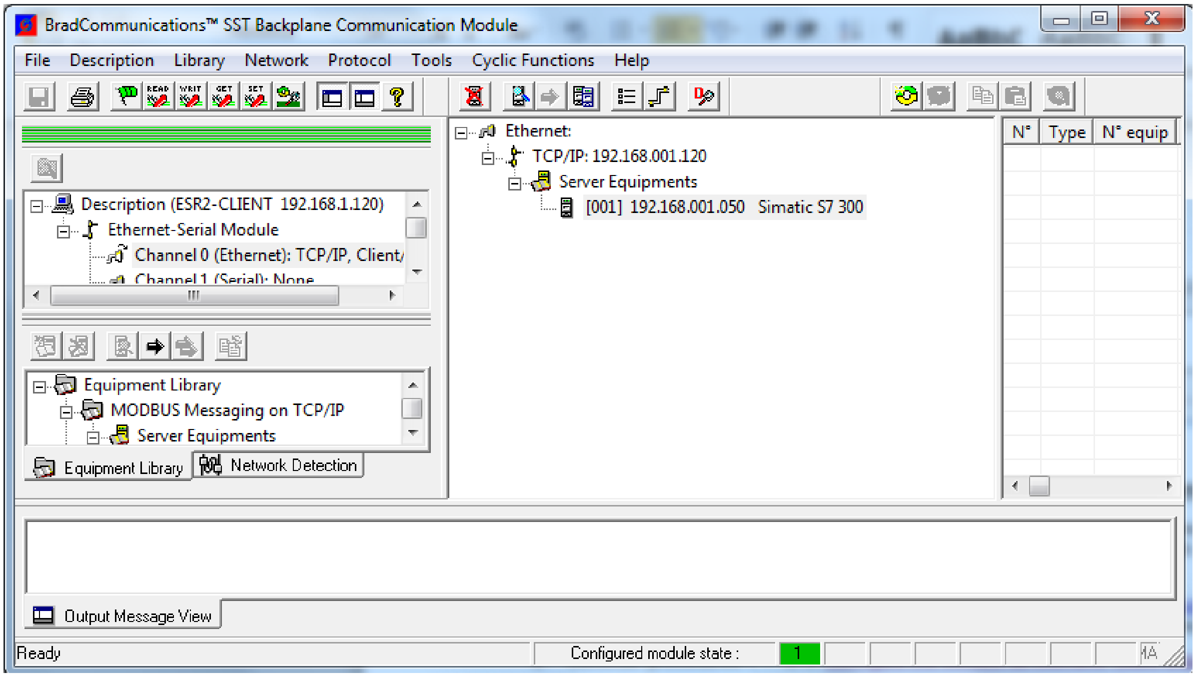


8. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

9. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

10. The module is now running a Siemens S7 Messaging in Client configuration. See how console application main screen will look like when running client equipment with server equipment added.

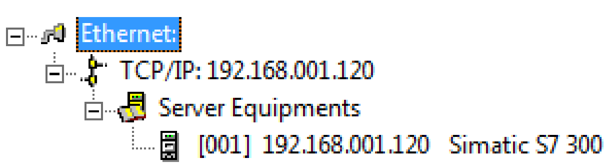
Figure 50: Console Main Screen In Client Configuration



6.2.3 Configure the Module as Siemens Client/Server Equipment

This product feature is only available in the SST-ESR2-CLX-RLL variant. The same procedure in Configure the Module as Siemens Client Equipment will be used in this section with some modification for server equipment.

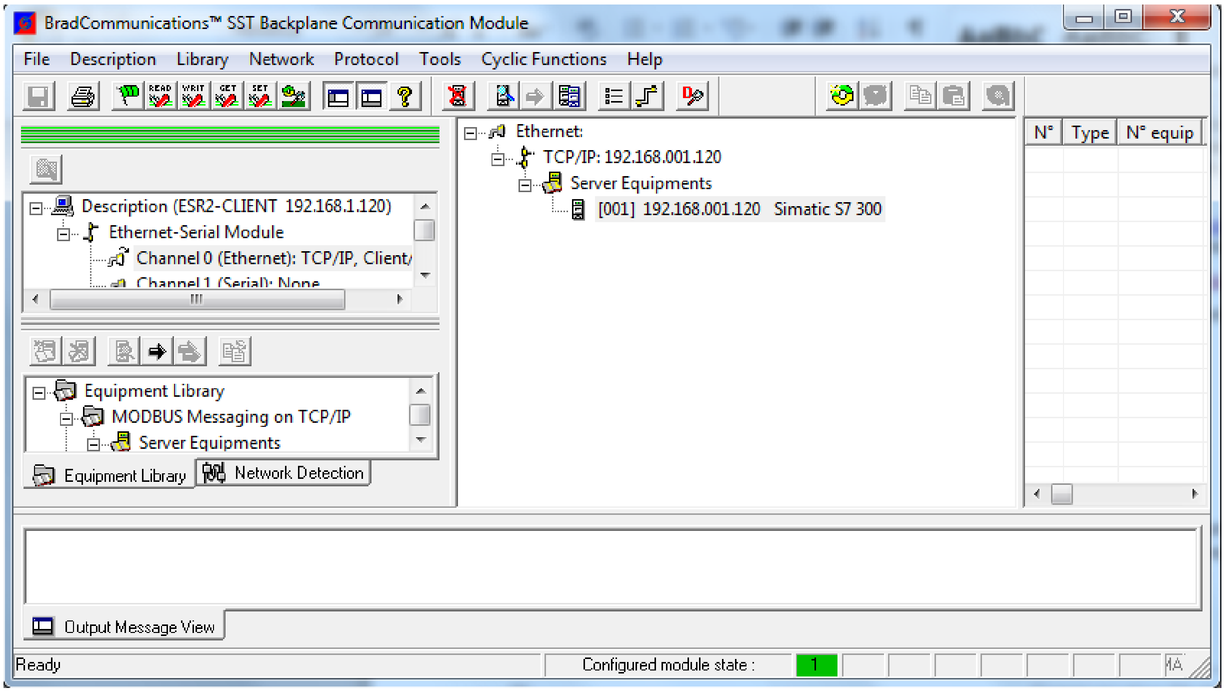
1. Complete procedures 1 - 6 in Configure the Module as Siemens Client Equipment section. When changing server equipment parameter in step 6, use the module's IP address.
2. Continue procedures 7 – 10 in Configure the Module as Siemens Client Equipment.
3. After adding the server equipment, it should look similar to what is displayed below.



4. Save configuration and download configuration, click  and . The configuration download takes about 2-3 minutes.
5. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

6. The module is now running a Siemens S7 Messaging in Client / Server configuration. See how console application main screen will look like when running client / server equipment in one module.

Figure 51: Console Main Screen In Client / Server Configuration



6.3 Modbus TCP Messaging Configuration

6.3.1 Configure the Module as Modbus TCP Server Equipment

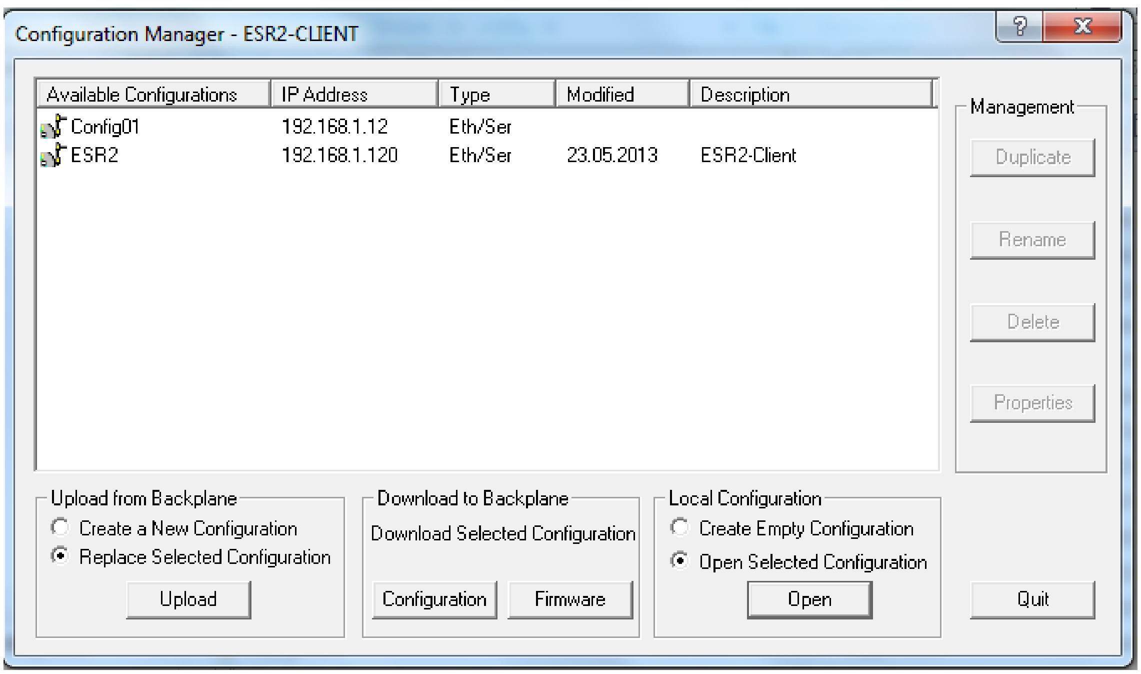
This product feature is only available in the SST-ESR2-CLX-RLL variant. Before proceeding in this section, the module must have a proper IP address configuration and it is able to be connected using the procedure in How to Browse your Module from the Network section. The module by default is configured in server configuration. To define what protocol messaging that TCP port runs. Follow the steps below:

1. In this sample configuration, the module will be configured as an server equipment to Modbus TCP PLCs running in client configuration.

2. Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

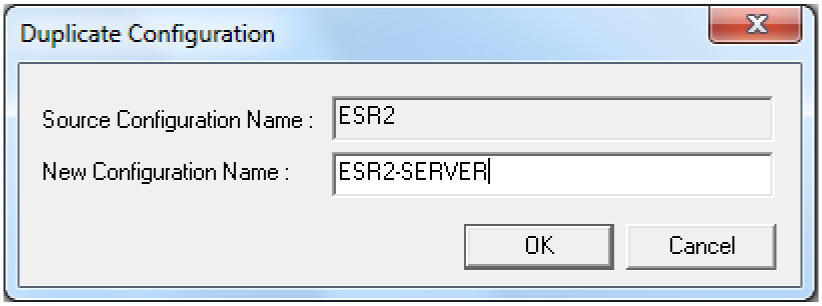
3. Select the newly created configuration from Creating a New Configuration section. Copy the created configuration into a different configuration name so the first configuration can be used as reference.

Figure 52: Configuration Manager



4. Click on “Duplicate” and name the new configuration as ESR2-Server

Figure 53: Duplicate Configuration

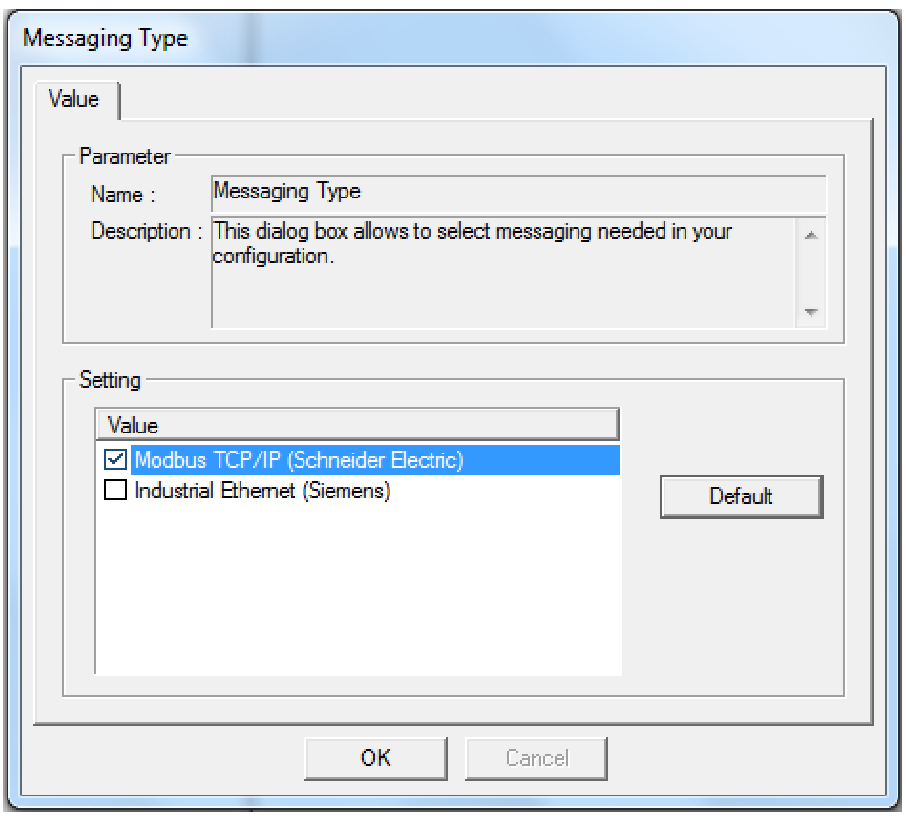


5. Click OK and select the newly duplicated configuration and click “Open”. The Console will use the new configuration and the new configuration name will be displayed in the “Configuration Description Area”.

6. Make sure the module is connected. Refer to Console Status Bar Information section which shows the status indicator of the module.

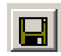
7. In the “Configuration Description Area”, locate then right click the Channel 0 (Ethernet):TCP/IP, Client/Server. The Messaging Type dialog appears.

Figure 54: Messaging Type Selection Window



8. Select “Modbus TCP/IP (Schneider Electric)”

9. Confirm the modification by clicking OK.

10. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

11. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

12. If the download is successful, the module is now ready to communicate to the Modbus TCP PLC

6.3.2 Configure the Module as Modbus TCP Client Equipment

This product feature is only available in the SST-ESR2-CLX-RLL variant. Procedures in Configure the Module as Modbus TCP Server Equipment is applicable to this section. To configure the module as Modbus TCP client, follow the listed steps below.

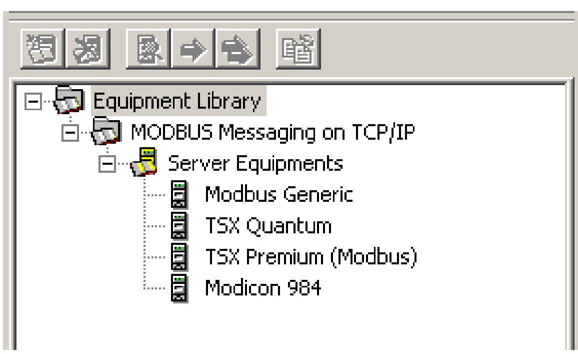
1. In this sample configuration, the module will be configured as Modbus TCP client equipment.

2. Complete the procedure 2-4 from Configure the Module as Modbus TCP Server Equipment. Name the new configuration to "ESR2-Client".

3. Then follow procedure 5-8 from Configure the Module as Modbus TCP Server Equipment

4. Confirm the modification by clicking OK.

5. The Console application Equipment Library area will have similar display as shown below.



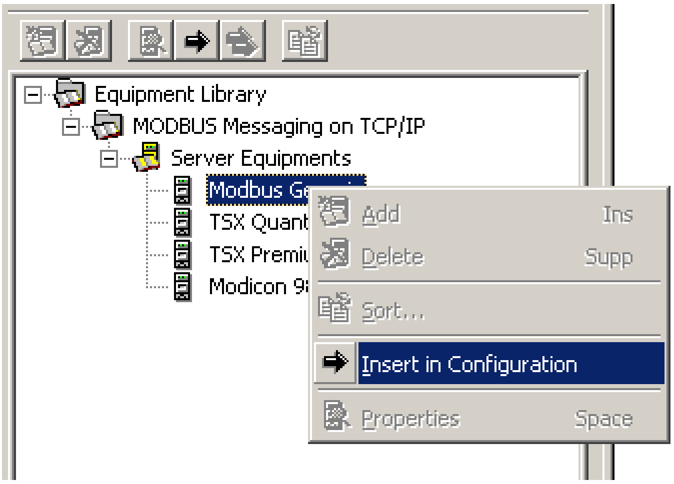
Note

When module is configured as server equipment, the equipment library area will be displayed. YOU DON'T NEED to do anything in those equipment when running in server configuration

6. The supported Modbus PLC's are listed above.

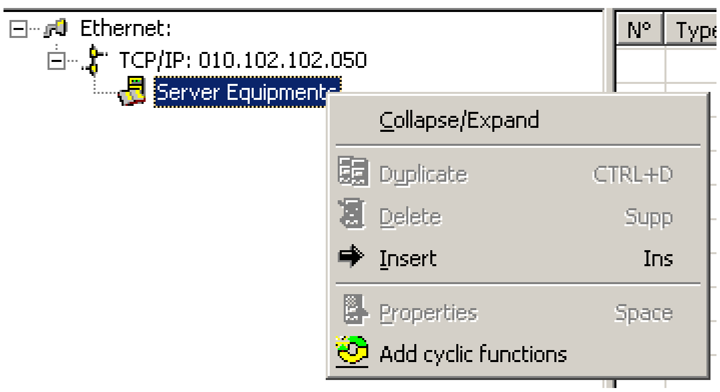
7. To add Modbus Generic, click on the selected PLC and choose any of the following methods.

   a. Right click the selected PLC and click on "Insert in Configuration", as shown here



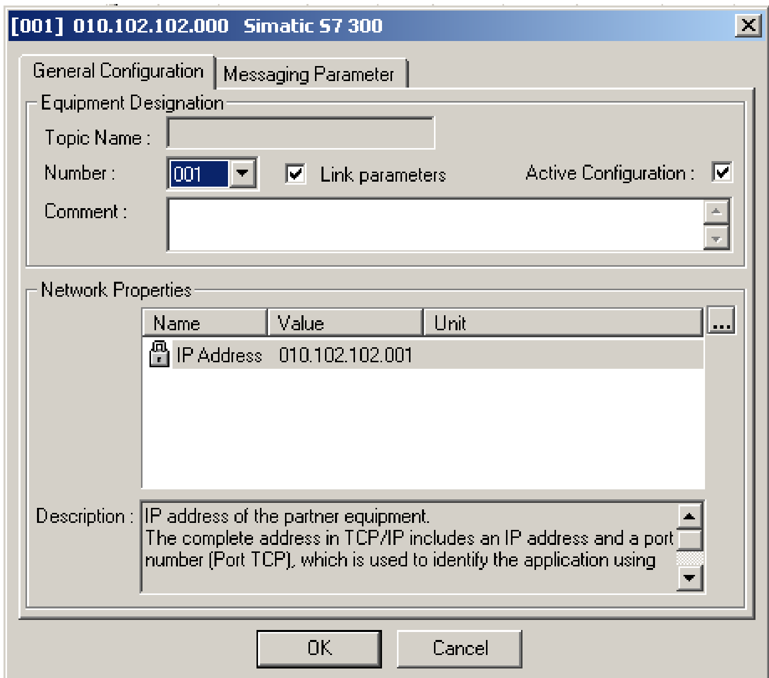
   b. Click on the 

   c. In the "Protocol View area", right click the Server Equipment and click "Insert", as shown here.



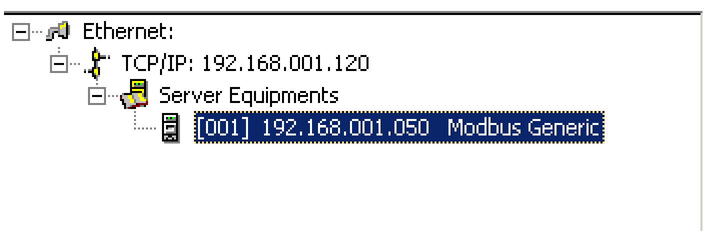
Any of the described method will display the screen below.

Figure 55: Server Equipment Configuration parameter



8. Change the IP address to Modbus PLC's IP address (PLC's IP address is 192.168.1.50)

9. Click OK. The protocol view area as shown below.



10. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

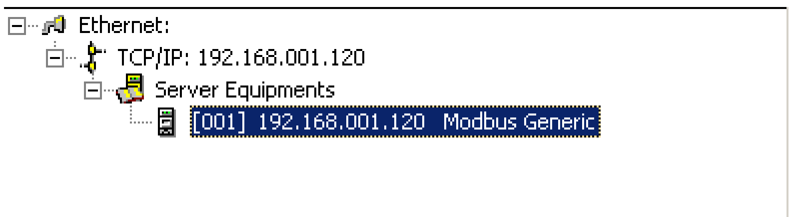
11. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information section.

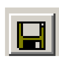
12. The module is now running Modbus TCP in Client configuration.

6.3.3 Configure the Module as Modbus TCP Client/Server Equipment

This product feature is only available in the SST-ESR2-CLX-RLL variant. To create client / server configuration, follow the steps below:

1. Complete procedures 1 - 8 in Configure the Module as Modbus TCP Client Equipment section. When changing server equipment parameter in step 6, you will have to use the module's IP address.
2. Continue procedures 7 – 10 in Configure the Module as Siemens Client Equipment.
3. Click OK. The module listed as an server equipment in the protocol view area is shown below.



4. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.
5. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information section.
6. The module is now running Modbus TCP in client / server configuration.

6.4 Modbus Serial Configuration

6.4.1 Configure the Module as Modbus Serial Slave Equipment

This product feature is available in both SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL variants. The connection type will vary depending in which module you have:

- If the module is SST-ESR2-CLX-RLL, all connection type can be used to configure the module.
- If the module is SST-SR4-CLX-RLL, only RSLinx and EtherNet/IP can be used.

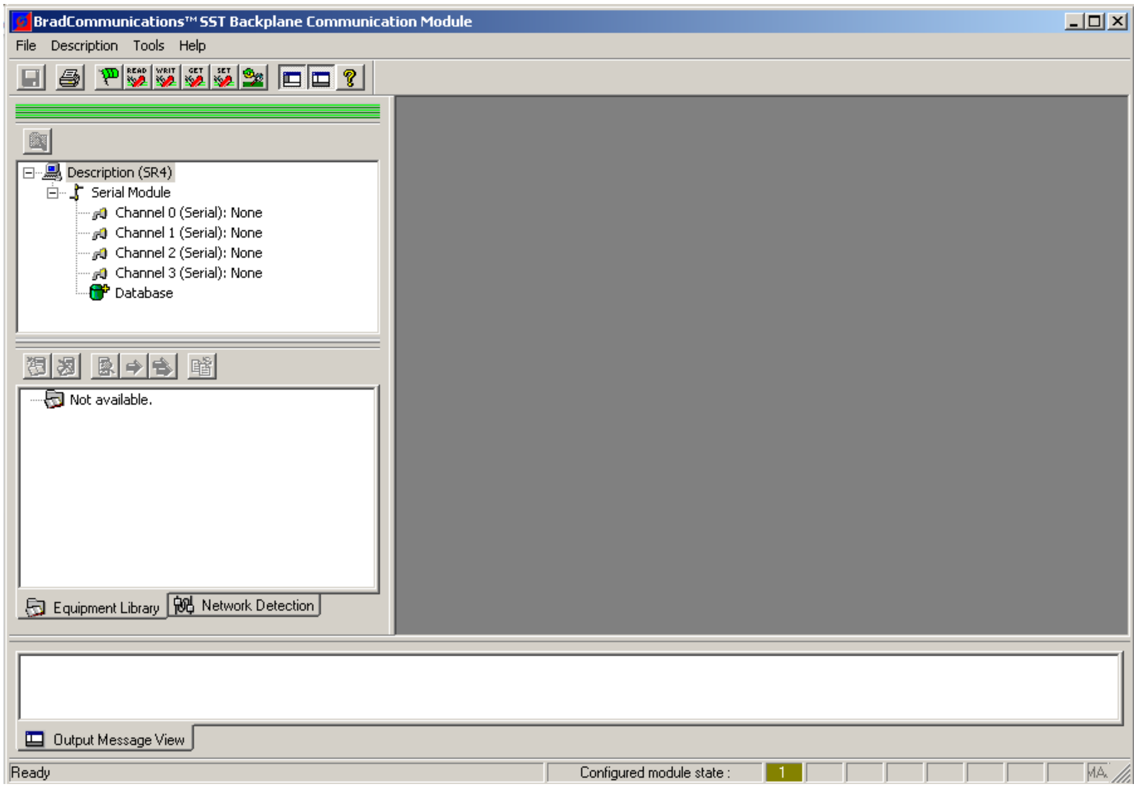
In this sample configuration, the SSR-SR4-CLX-RLL variant is used.

Note

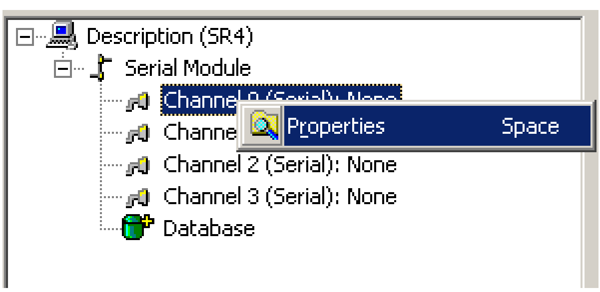
To use RsLinx connection type, the Rockwell Automation RSLinx must have been successfully installed with license.

1. Make sure the module has a backplane connection before starting to create the configuration.
2. In this sample configuration, the module will be configured as server equipment to a Modbus Serial PLC client equipment.
3. Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.
4. After completing the instruction from Creating a New Configuration for SST-SR4-CLX-RLL, the screen should show similar display as below.

Figure 56: Console Main of SR4 with RSLinx Connection Type

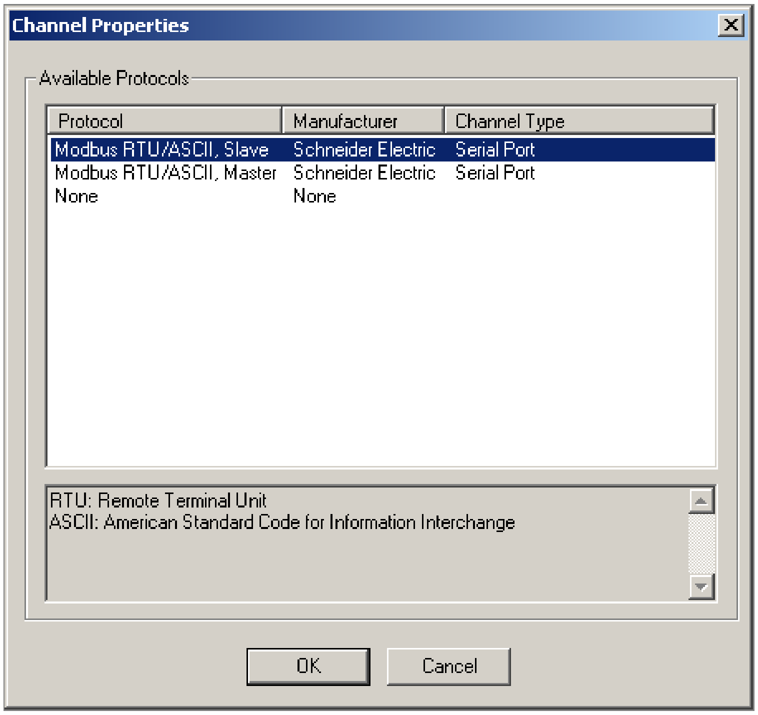


5. In the “Configuration Description Area”, locate Channel 0 (Serial): None and right click to access the channel property, as shown below. NOTE: any available channel can be selected.



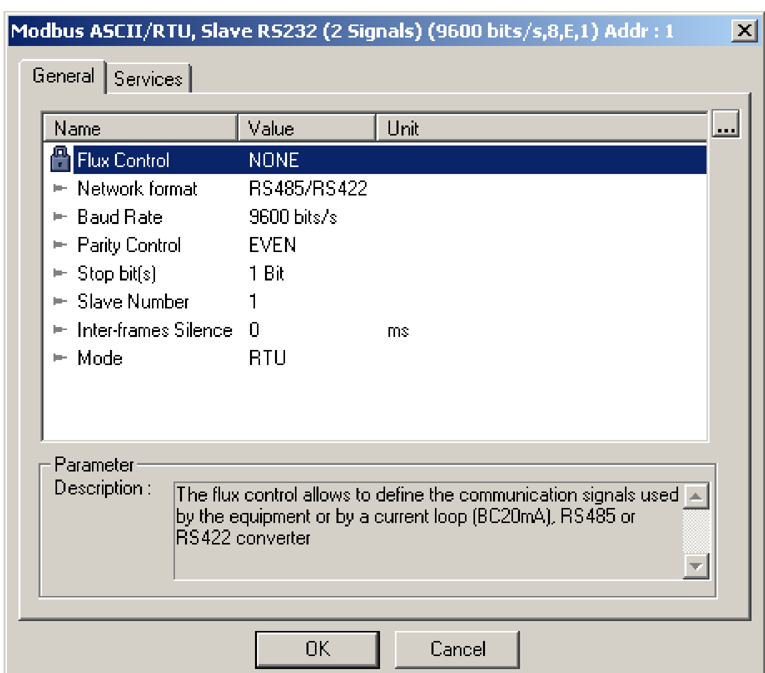
6. Click on “Properties” and select Modbus RTU/ASCII, Slave” as shown below.

Figure 57: Serial Protocol Selection



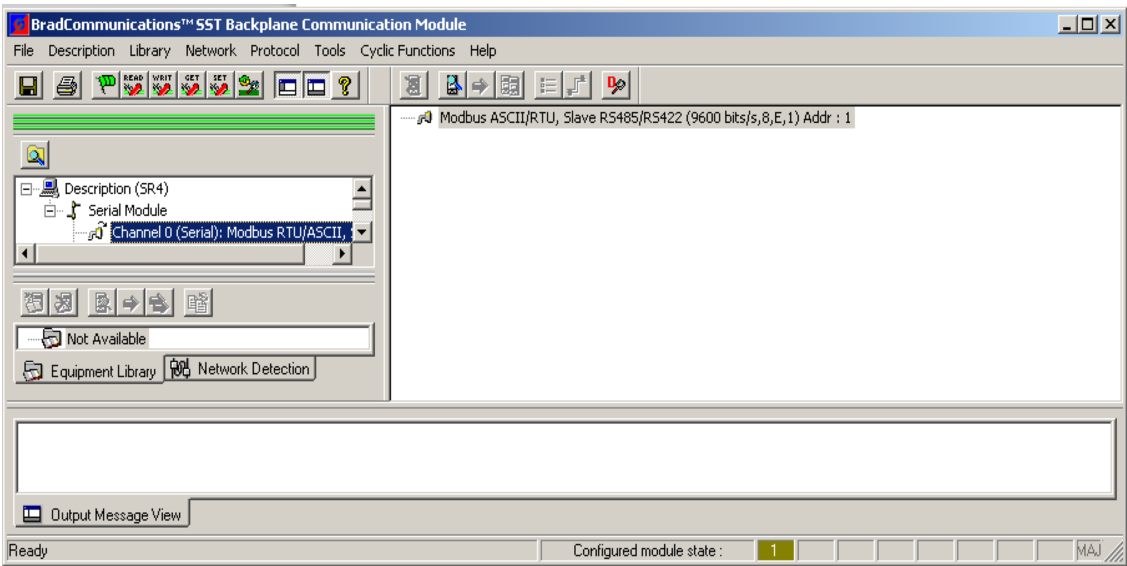
7. Click “Ok” and then configure the channel based on the requirement. The diagram below is just provided for illustration.

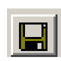
Figure 58: Serial Port Configuration



8. Click OK and you see this entry in the Protocol View Area. If a different network format and baud rate are selected, the display maybe different.

Figure 59: Console Main of SR4 with Channel 0 in Slave Configuration



9. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

10. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information section.

11. If download is successful, the module is now running Modbus serial in server configuration.

6.4.2 Configure the Module as Modbus Serial Master Equipment

This product feature is available in both SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL variants. The connection type will vary depending in which module you have:

- If the module is SST-ESR2-CLX-RLL, all connection type can be used to configure the module.
- If the module is SST-SR4-CLX-RLL, only RSLinx and EtherNet/IP can be used.

In this sample configuration, the SSR-SR4-CLX-RLL variant is used in RTU mode.

Note

To use RsLinx connection type, the Rockwell Automation RSLinx must have been successfully installed with license.

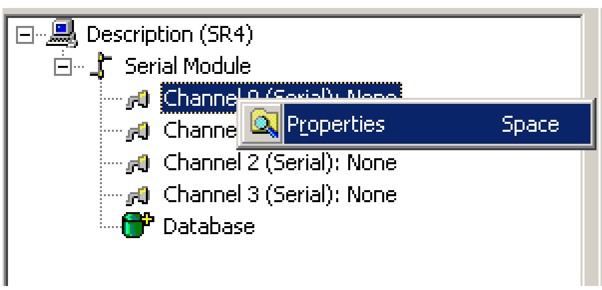
1. Make sure the module is successfully connected before starting to create the configuration.

2. In this sample configuration, the module will be configured as server equipment to a Modbus Serial PLC client equipment.

3. Launch the Configuration Manager from Start->All Programs->BradCommunications->SST Backplane Communication Module->Console.

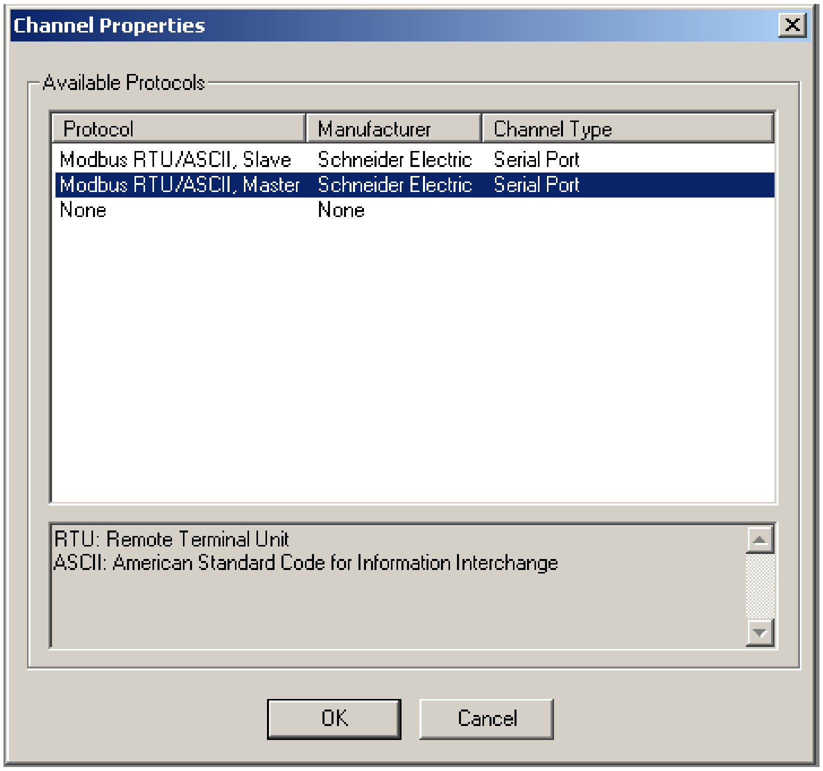
4. After completing the instruction from Creating a New Configuration for SST-SR4-CLX-RLL, the screen should show similar display as in Figure_56.

5. In the “Configuration Description Area”, locate Channel 0 (Serial): None and right click to access the channel property, as shown below. Any available channel can be selected.



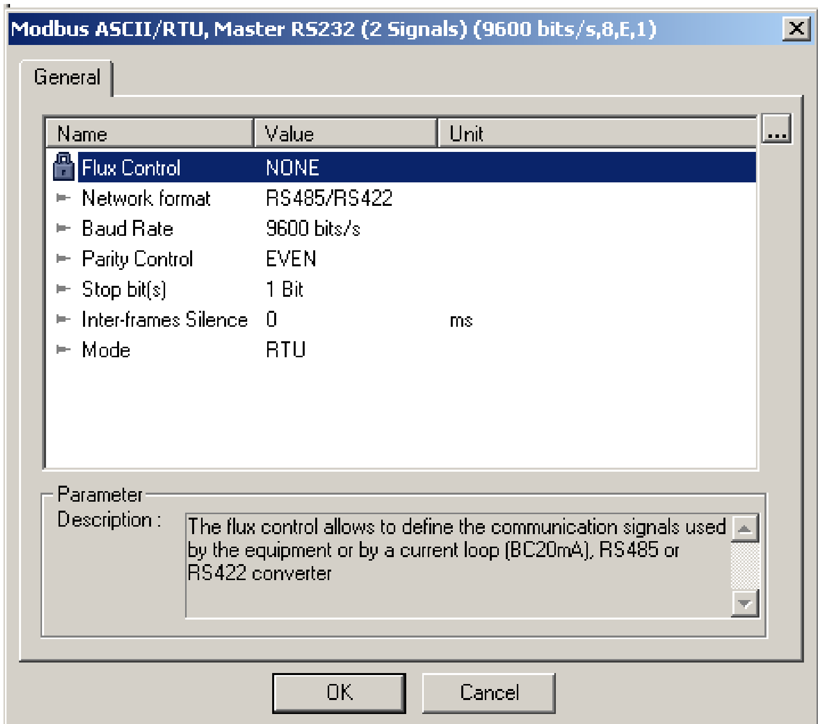
6. Click on “Properties” and select Modbus RTU/ASCII, Master” as shown below.

Figure 60: Selecting Master Equipment from Channel Property



7. Click OK, the screen below should show up and then click ok again. This diagram below is only used for illustration.

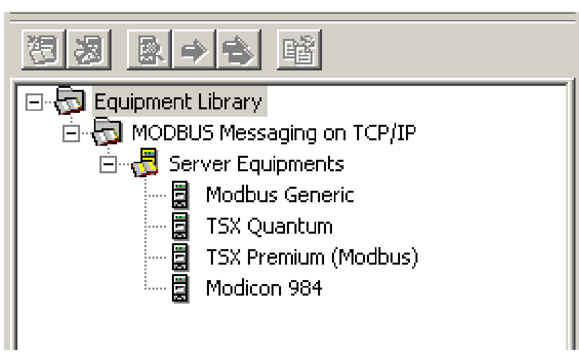
Figure 61: Configuring the Master Equipment



Note

Similar configuration can be followed but using ASCII instead of RTU mode.

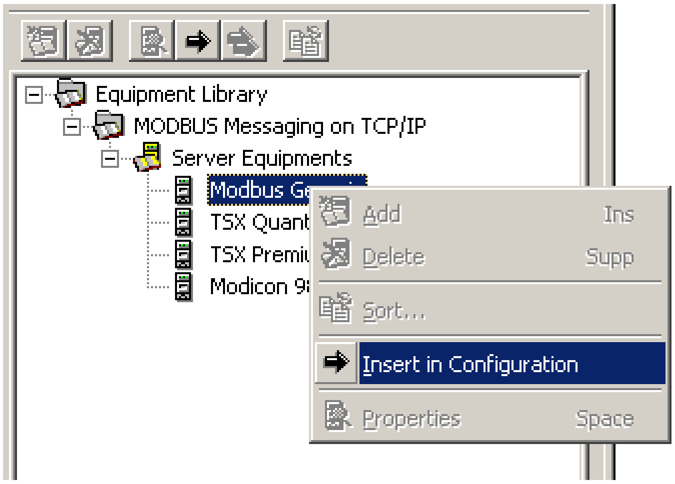
8. The Console application Equipment Library area will have similar display as shown below.



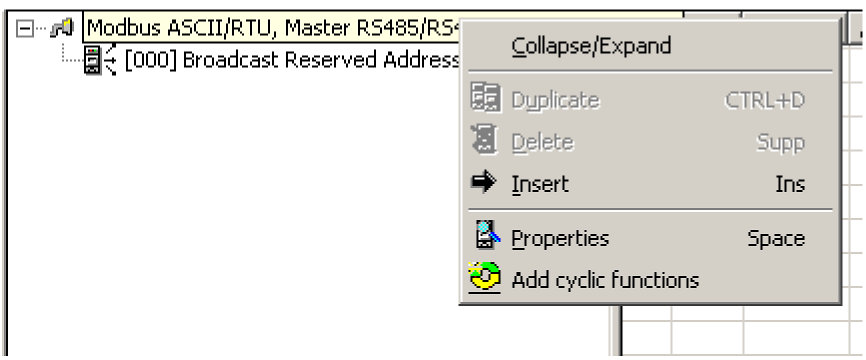
9. The supported Modbus PLC's are listed above.

10. The Modbus Generic is used in this sample configuration. To add Modbus Generic, click on the selected PLC and choose any of the following methods.

    a. Right click the selected PLC and click on “Insert in Configuration”, as shown here

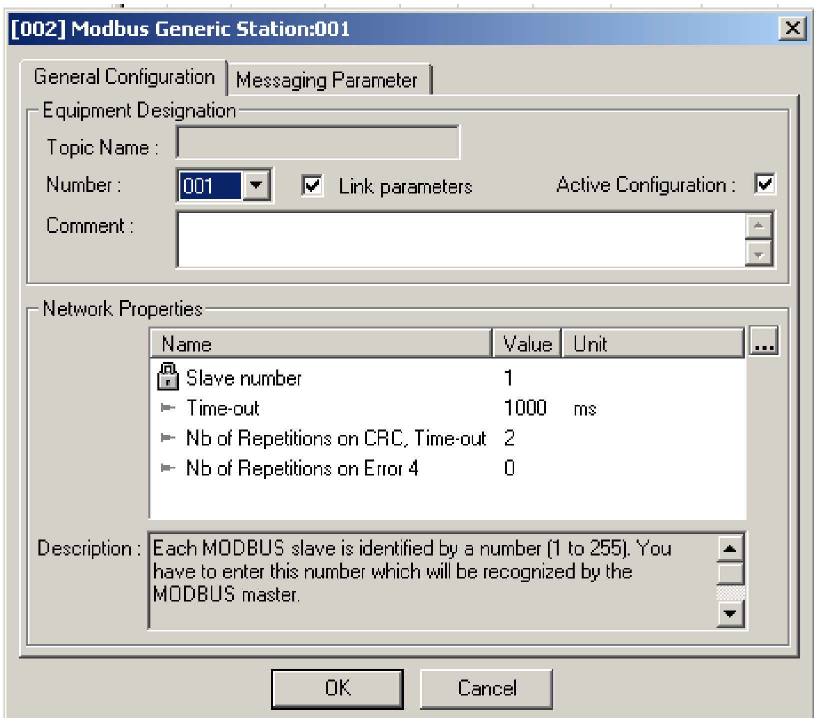


    b. Click on the 

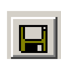


11. Any option chosen will display the next window. Fill in the correct value in the Slave number parameter.

Figure 62: Adding Slave Equipment and its Parameter



12. Click OK and the Protocol View Area will show the newly added PLC.

12. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

13. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information section.

14. If download is successful, the module is now running Modbus serial in client configuration.

6.4.2.1 Configure the Module as Modbus Serial Master/Slave Configuration

Depending on the requirement, the following combination can be configured:

- Configure 1 channel as Modbus Serial Master Equipment and 3 channels as Modbus Serial Slave equipment.
- Configure 2 channels as Modbus Serial Master Equipment and 2 channels as Modbus Serial Slave equipment.
- Configure 3 channels as Modbus Serial Master Equipment and 1 channel as Modbus Serial Slave equipment.

To configure the module in master/slave configuration, follow the procedure below:

1. Run procedures in Configure the Module as Modbus Serial Slave Equipment in the channel to configure slave.
2. Run procedures in Configure the Module as Modbus Serial Master Equipment in the channel to configure master
3. It is very important that using the right slave number in the slave number parameter when adding the slave equipment to the master equipment.

6.5 Configuration with Cyclic Function

This section will provide sample configurations with cyclic function configured in the following supported Messaging Type:

- Modbus TCP Messaging
  - Modbus TCP in Default Addressing Mode [Section 6.5.1.1]
  - Modbus TCP in Extended Addressing Mode [Section 6.5.1.2]
- Modbus Serial Messaging
  - Modbus Serial in Default Addressing Mode [Section 6.5.2.1]
  - Modbus Serial in Extended Addressing Mode [Section 6.5.2.2]
- Siemens S7/S5 Messaging
  - Siemens S7/S5 in Default Addressing Mode [Section 6.5.3.1]
  - Siemens S7/S5 in Extended Addressing Mode [Section 6.5.3.2]

6.5.1 Modbus TCP Messaging.

6.5.1.1 Modbus TCP in Default Addressing Mode

When module is configured in default addressing mode, the modules database address 0-699 is mapped by default to ControlLogix INPUT, OUTPUT and STATUS Table.

No Backplane Connection:

To be able to demonstrate successfully this sample configuration, a Modbus TCP Client equipment and Modbus TCP Server equipment is needed, and the client and server equipment must have an active connection to be able to show data exchanges configured through cyclic function.

Using SST-ESR2-CLX-RLL product, an easy following procedure will be described on how to create a cyclic function.

Using two available modules of this product variant, one can be configured as a Modbus TCP Client, and the other one as Modbus TCP Server.

If one module is only available, then a single communication module configured both as Modbus TCP Client and Modbus TCP Server will be used. Cyclic functions can only be created when module is running in client equipment configuration mode.

Note

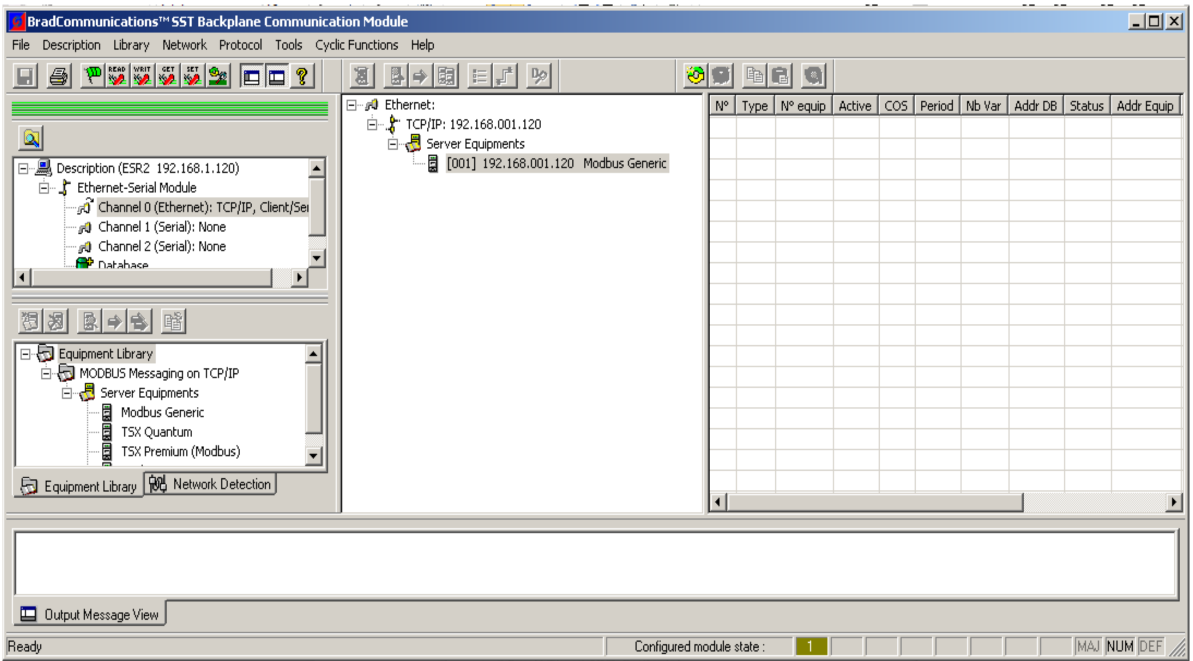
Since client and server equipment is sharing the same database, make sure different addresses are used for the source and destination address of the data.

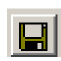
The procedures below assume only a single communication module for both client and server configuration will be used.

1. Complete procedures in Configure the Module as Modbus TCP Client/Server Equipment

2. After completing the steps in Configure the Module as Modbus TCP Client/Server Equipment, the Console application main screen should be displayed with Port 0 (TCP/IP) configured as Client with Modbus TCP server equipment added with IP address 192.168.1.120. See Figure 63 below.

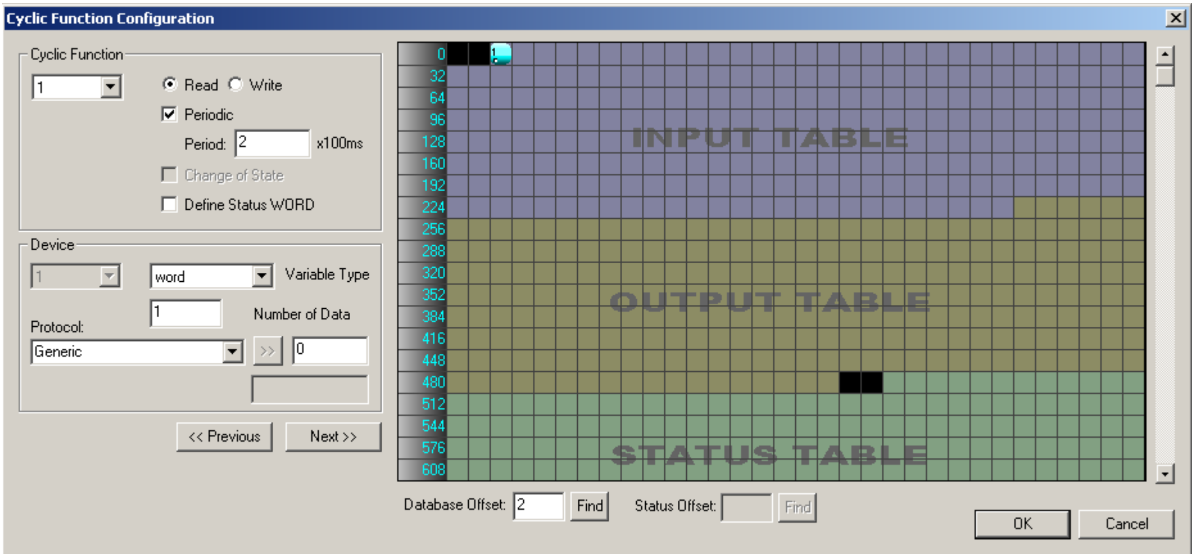
Figure 63: Console Application Main Screen



3. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.
4. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.
5. Creation of cyclic functions can be started by clicking . The created cyclic function will provide automatic data exchange between the client and server equipment.

   A read and write cyclic function will be created to automatically read and write data from/to the server equipment variable addresses. The function code as well as the size of the data to be read is part of the cyclic function parameters.

Figure 64: Cyclic Function Dialog Window

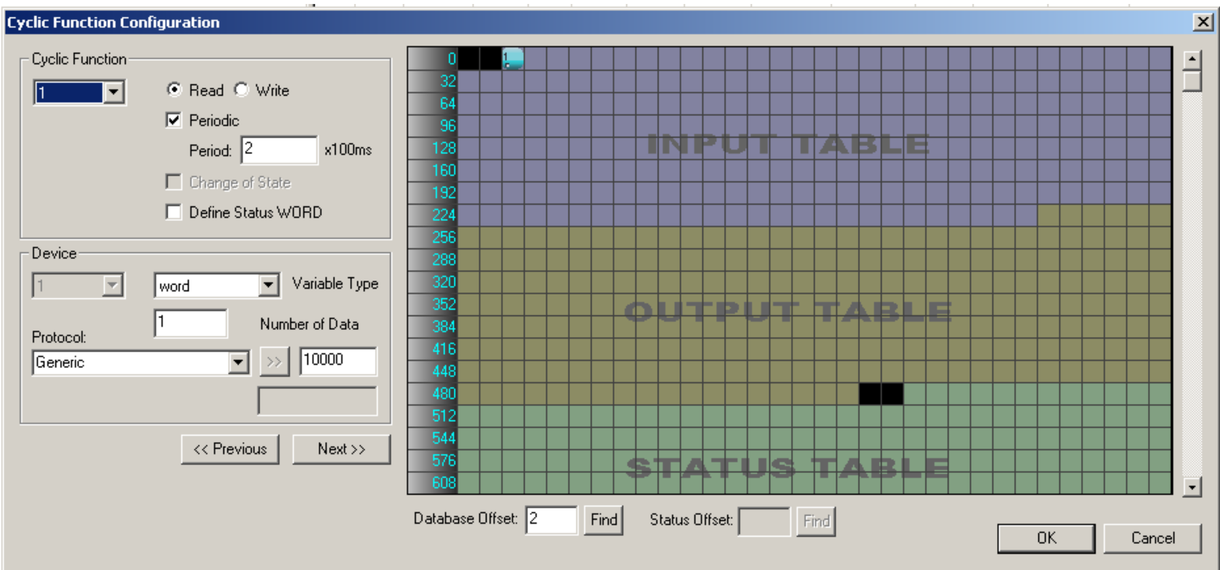


The detail of each parameter in discussed in Cyclic Function Configuration Parameters and Table 38.

a. Create cyclic function #1 with the parameter below.
   i. Use default cyclic function number provided (1 should be the default if this the first cyclic function)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Generic
   vii. Device First variable address is 10000 (server equipment address)
   viii. Database offset is 2 (client equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below. See Figure 65.

Figure 65: Cyclic Function Dialog View after adding successfully a Cyclic Function



b. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

c. Create cyclic function #2 with the parameter below

   i. Use default cyclic function number provided (2 should be the default now)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Generic
   vii. Device First variable address is 11000 (server equipment address)
   viii. Database offset is 1000 (client equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.

The difference between the first and second cyclic function is the database offset location in the client.

1st cyclic function database address is located in the mapped IO table address 0-699

2nd cyclic function database address is located outside of the mapped IO table address

e. Create cyclic function #3 with the parameters below:

   i. Use default cyclic function number provided (3 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Generic

vii. Device First variable address is 12000 (server equipment address)

viii. Database offset is 250 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view

g. Create cyclic function #4 with the parameter below

i. Use default cyclic function number provided (4 should be the default now)

ii. Select Write cyclic function type

iii. Check the Periodic setting

iv. Use default period of 2 x 10ms = 20ms

v. Number of data is 1

vi. Protocol is Generic

vii. Device First variable address is 13000 (server equipment address)

viii. Database offset is 2000 (client equipment address)

h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

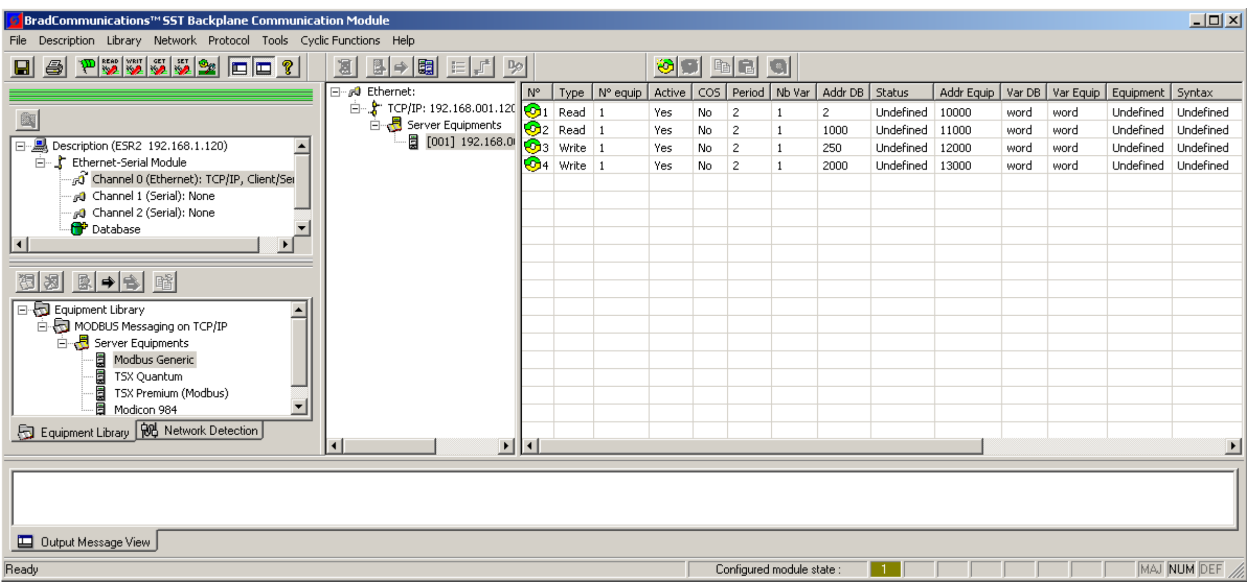
The difference between the first and second cyclic function is the database offset location in the client.

3rd cyclic function database address is located in the mapped IO table address 0-699

4th cyclic function database address is located outside of the mapped IO table address

6. After creating all the cyclic function your screen should have a similar display as below.

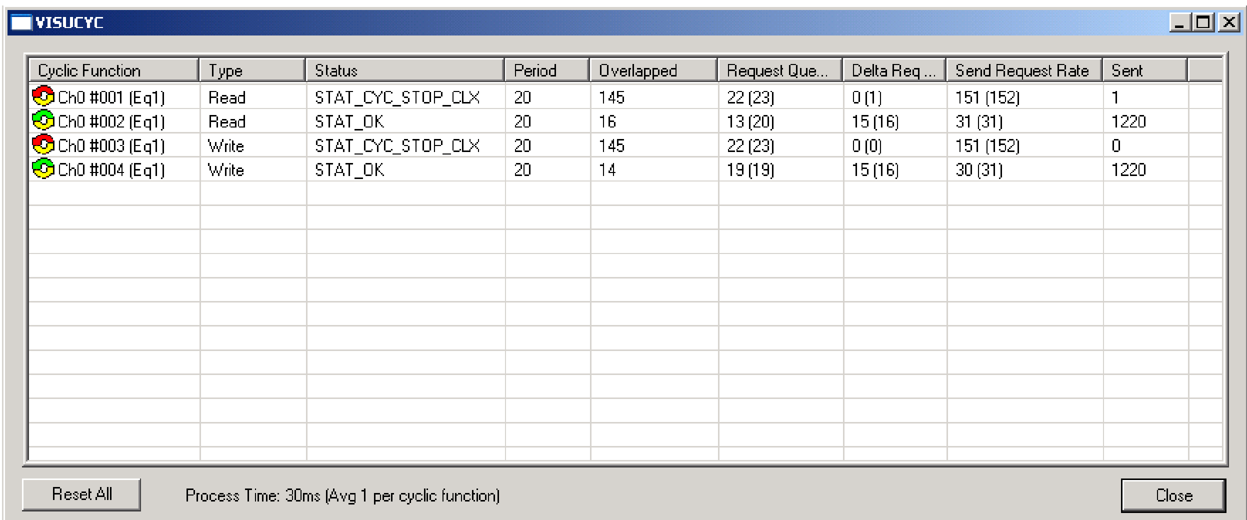
Figure 66: Cyclic Function List View



7. Download the configuration to the module by clicking  first and then .

8. When download has completed, (Refer to Console Status Bar Information to check if module is properly initialized), run the visual cyclic function application by clicking  from the console menu bar.
9. The Visucyc application should now be running similar to screen shot below.

Figure 67: VisuCyc Application Window with Cyclic Function Status



*Both 1st and 3rd cyclic function status is 135 – **not active / not running***

*Both 2nd and 4th cyclic function status is 0 – **active / running***

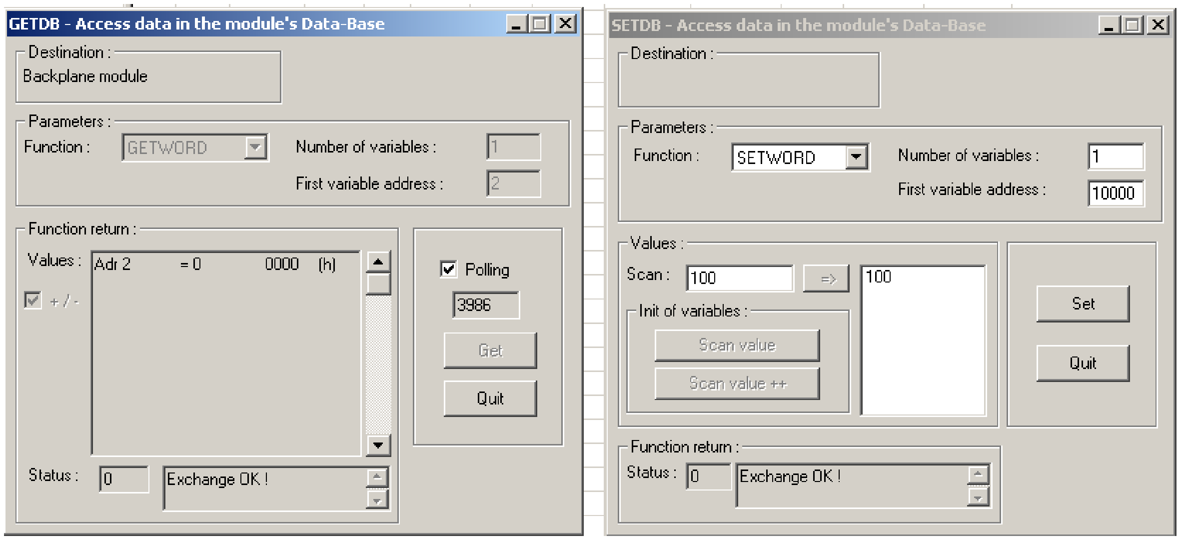
Note

The module will not run any cyclic function configured in the database address mapped to INPUT, OUTPUT and STATUS Table when there is no backplane connection.

10. To check if active cyclic function is running properly, follow this brief diagnostic below:

    a. Locate and run  and  application.

    b. To test cyclic function # 1, fill in the values as shown below.

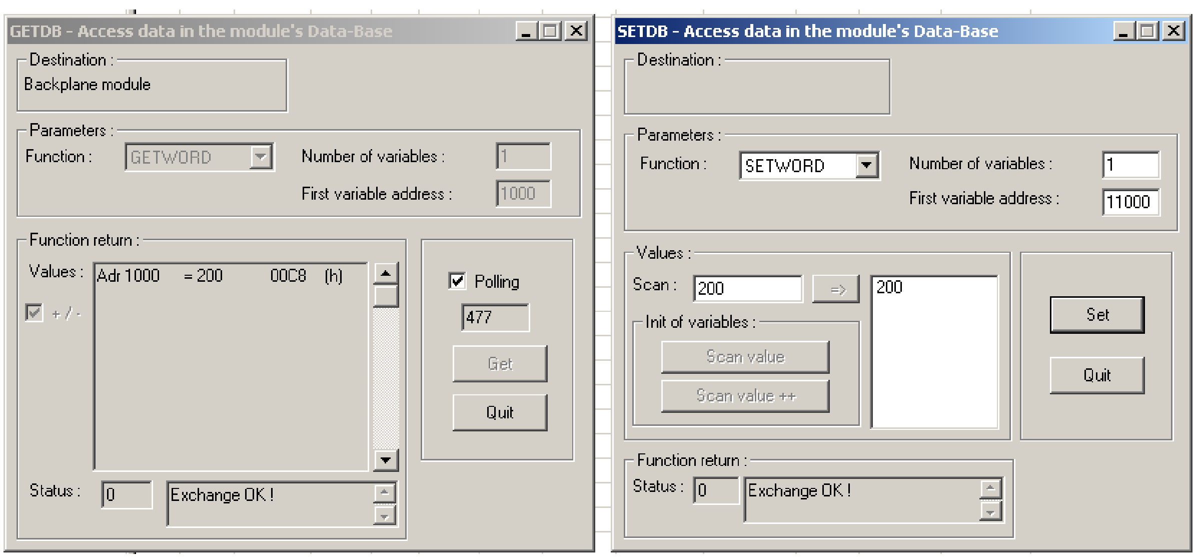


SetDB sets the server equipment address 10000 with value 100.

GetDB reads the client equipment database address 2 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 2.

c. To test cyclic function # 2, fill in values as shown below.

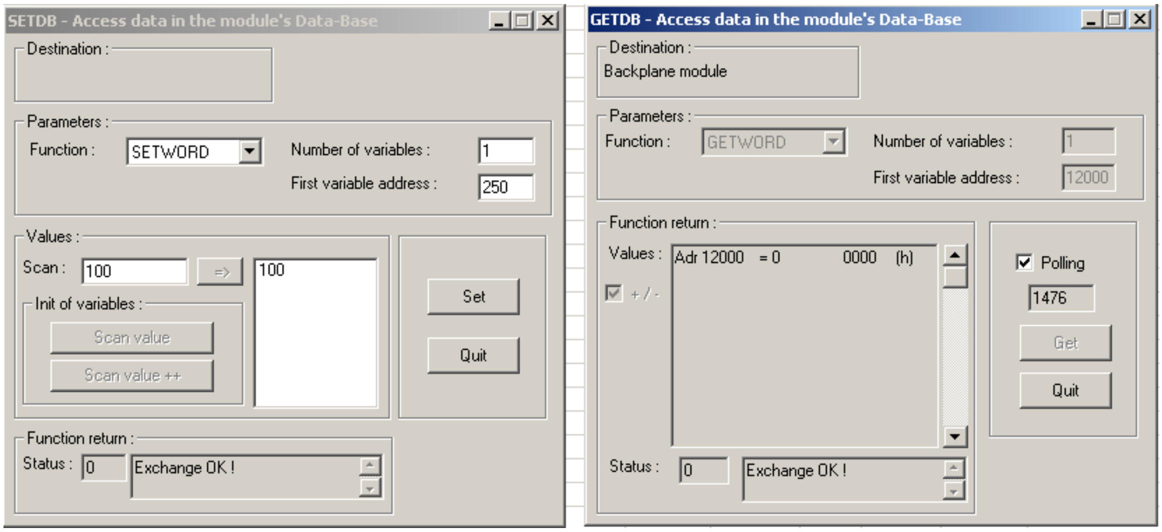


SetDB sets the server equipment address 11000 with value 200.

GetDB reads the client equipment database address 1000 with value shown as 200.

If the data is shown correctly then the setup is working properly.

d. To test cyclic function # 3, fill in the values as show below.



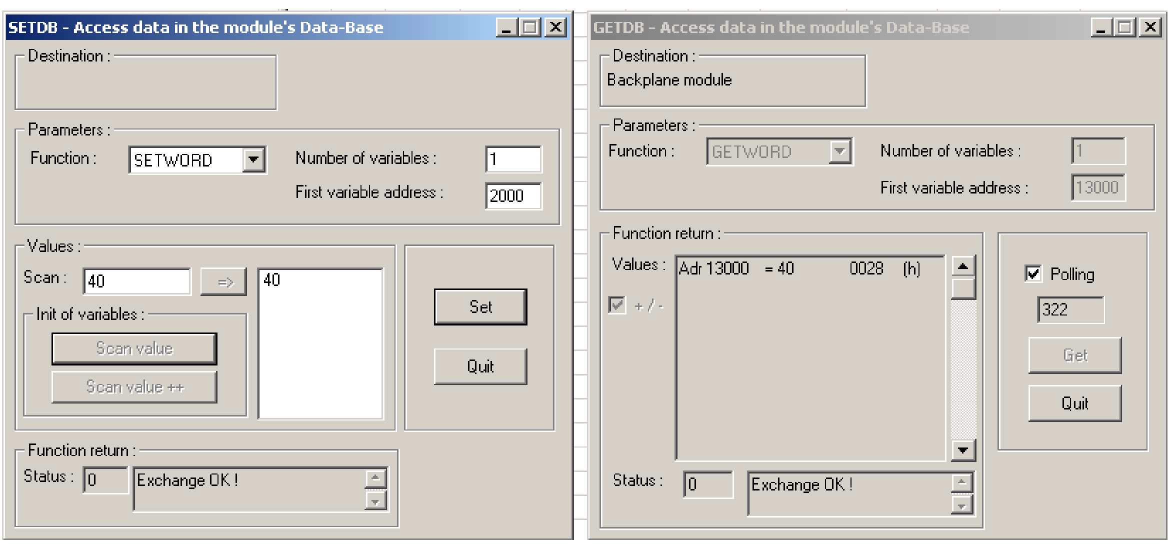
SetDB sets the client database address 250 with value 100.

GetDB reads the server equipment database address 12000 with value shown as 0.

If the data is shown correctly then the setup is working properly.

As indicated in the cyclic function status from Visucyc application, cyclic function #3 is ***not active***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 250.

e. To test cyclic function #4, fill in the values as shown below.



SetDB sets the client equipment address 2000 with value 40.

GetDB reads the server equipment address 13000 with value shown as 40.

If the data is shown correctly then the setup is working properly.

In the next set of procedures, the module will communicate with the ControlLogix CPU. When backplane connection becomes active, all cyclic function with **"not active / not running"** state will switch to "**active/running**".

With Backplane Connection:

To be able to demonstrate successfully this sample configuration, it is assumed that the previous sets of procedures (from **No Backplane Connection**) were completed successfully.

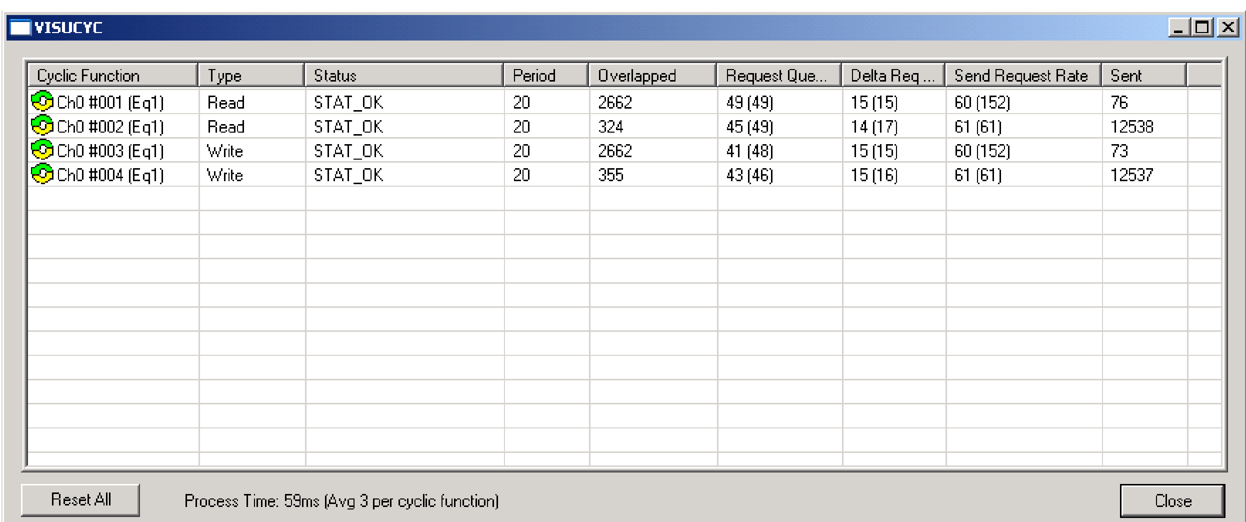
1. Create a backplane connection by following procedures from Using 1756 Generic Profile section.

2. Make sure using the right Ethernet/IP address, the correct slot location of the module, and it is able to make it into "Run Mode". See below.

3. At this point, all the LEDs on the module are in Green color, and "COPN" alternately displayed with the module's IP address and Configuration name, configured in step #1.

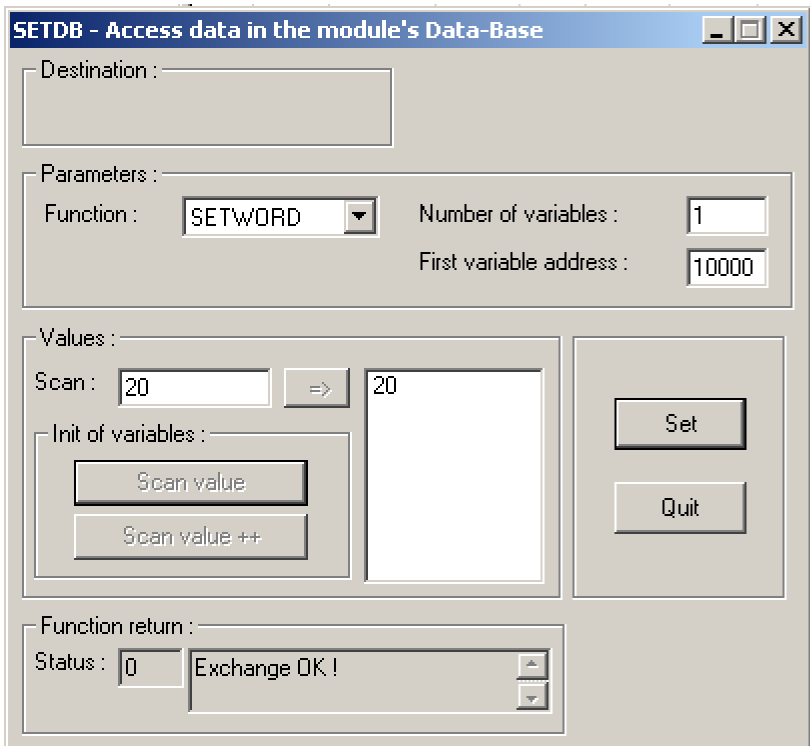
4. Launch the VisuCyc and all the cyclic functions are in active status. See below.

Figure 68: VisuCyc Application Window with all Cyclic Function running OK



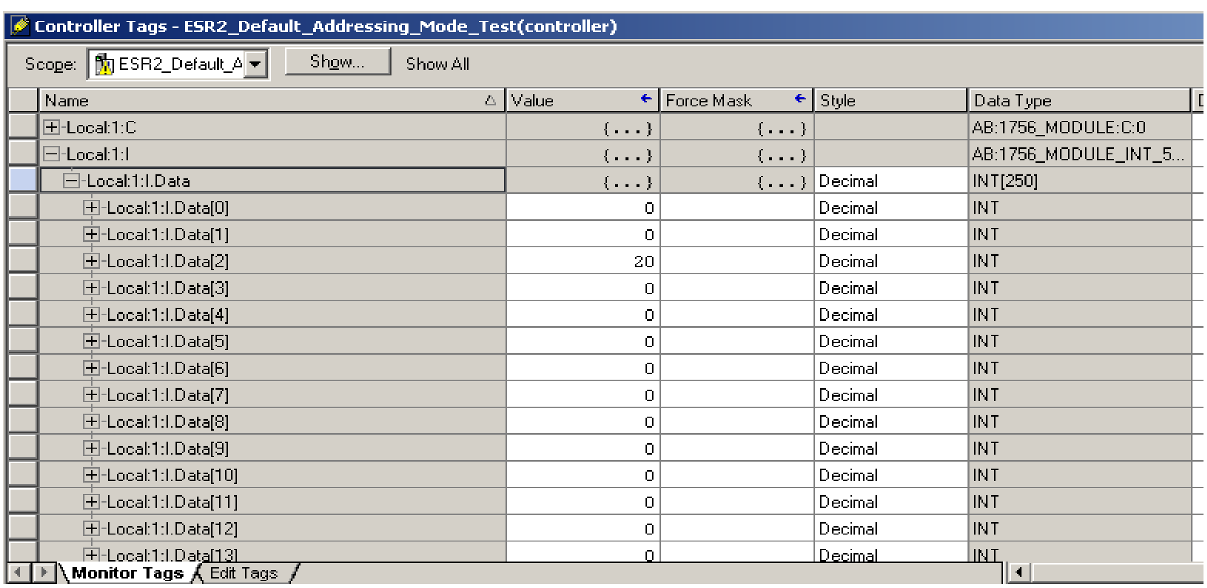
5. Test cyclic function 1 & 3, check if values written and read from the mapped database address are exchanged properly.

6. To test cyclic function #1, fill in the values as shown below.



SetDB sets the server equipment address 10000 with value 20.

7. In RSLogix 5000 Software, expand the INPUT Table in the Controller Tags, the value 20 is in offset 2. See below.


Controller Tags - ESR2_Default_Addressing_Mode_Test(controller)

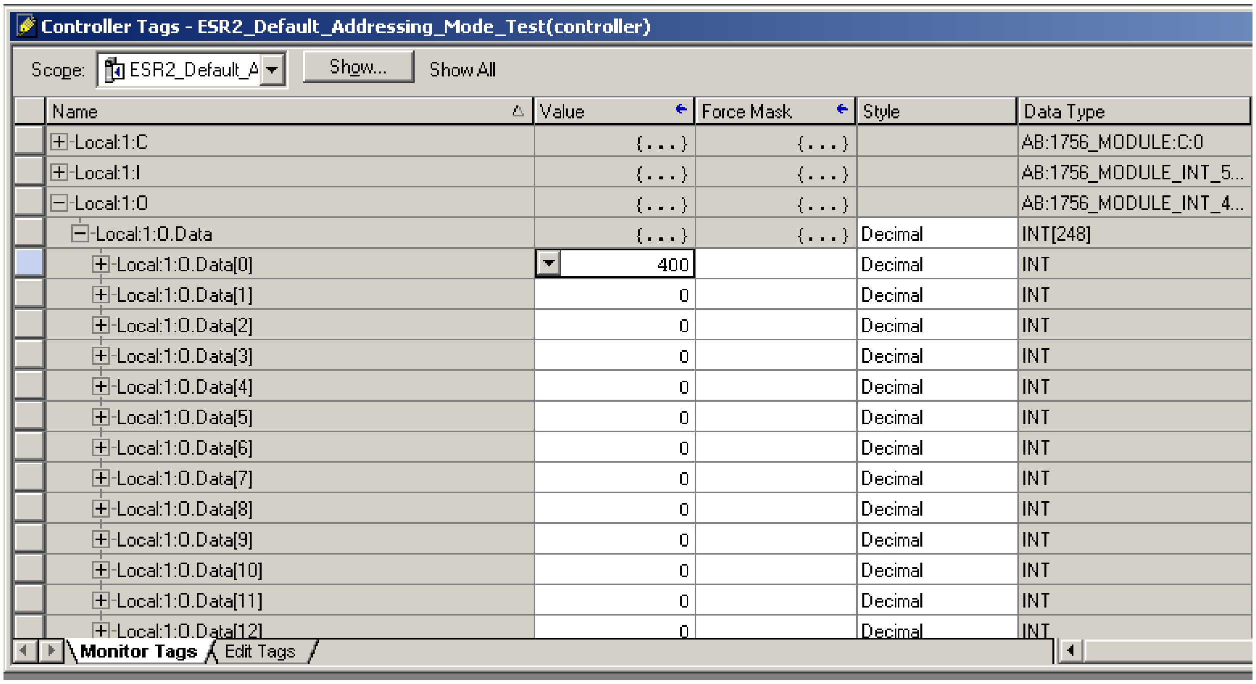
Scope: ESR2_Default_A Show... Show All

Name	Value	Force Mask	Style	Data Type
+-Local:1:C	{...}	{...}		AB:1756_MODULE:C:0
-Local:1:I	{...}	{...}		AB:1756_MODULE_INT_5...
-Local:1:I.Data	{...}	{...}	Decimal	INT[250]
+-Local:1:I.Data[0]	0		Decimal	INT
+-Local:1:I.Data[1]	0		Decimal	INT
+-Local:1:I.Data[2]	20		Decimal	INT
+-Local:1:I.Data[3]	0		Decimal	INT
+-Local:1:I.Data[4]	0		Decimal	INT
+-Local:1:I.Data[5]	0		Decimal	INT
+-Local:1:I.Data[6]	0		Decimal	INT
+-Local:1:I.Data[7]	0		Decimal	INT
+-Local:1:I.Data[8]	0		Decimal	INT
+-Local:1:I.Data[9]	0		Decimal	INT
+-Local:1:I.Data[10]	0		Decimal	INT
+-Local:1:I.Data[11]	0		Decimal	INT
+-Local:1:I.Data[12]	0		Decimal	INT
+-Local:1:I.Data[13]	0		Decimal	INT

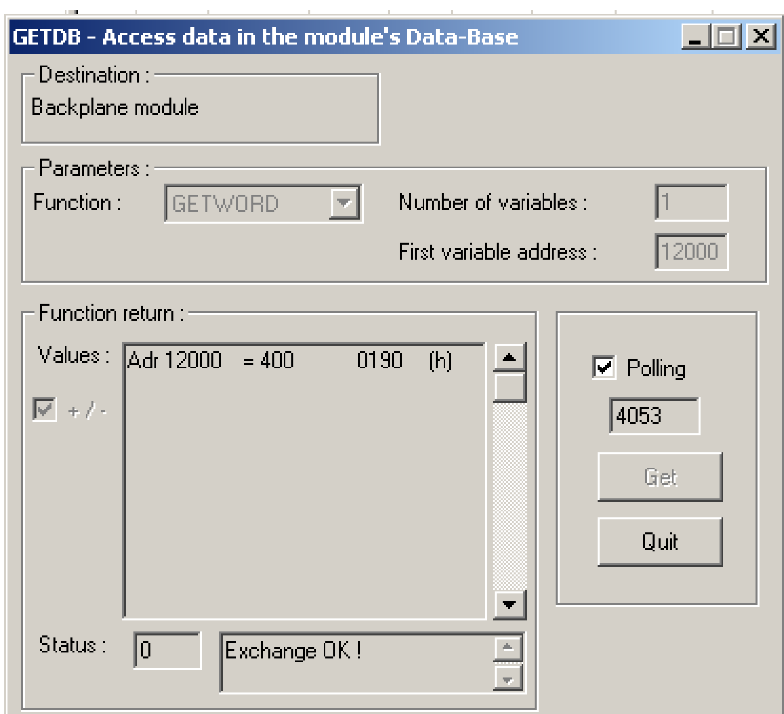
Monitor Tags / Edit Tags

8. If the values are correct, the backplane configuration is working properly.

9. To test cyclic function #3, expand Output Table and write 400 into offset 0 (offset 250 in database)



Run ***GetDB*** and read the values as offset 12000.



10. If the values are correct, the backplane configuration is working properly.

11. Start adding the test of desired configuration.

6.5.1.2 Modbus TCP in Extended Addressing Mode

When module is configured in extended addressing mode, the entire address defined in Input Table, Output Table and Status Table from the Console Database Configuration are accessible from the backplane using our AOI (Add-On-Instruction).

In this configuration, map 1600 words of INPUT, 1600 words of OUTPUT and 1024 words of STATUS of the module's database to the backplane. The database configuration uses extended addressing mode to map the database larger than the maximum INPUT and OUTPUT table connection size allowed in the ControlNet.

No Backplane Connection:

To be able to demonstrate successfully this sample configuration, Modbus TCP Client equipment and Modbus TCP Server equipment are used, and they must have an active connection to be able to show data exchanges configured through cyclic function.

Using SST-ESR2-CLX-RLL product, and create cyclic functions with easy to follow procedures.

If there are two available modules of this product variant, configure one as Modbus TCP Client and the other one as Modbus TCP Server.

If there is only one module, use a single communication module configured both as Modbus TCP Client and Modbus TCP Server. Cyclic functions can only be created when module is running in client equipment configuration.

Note

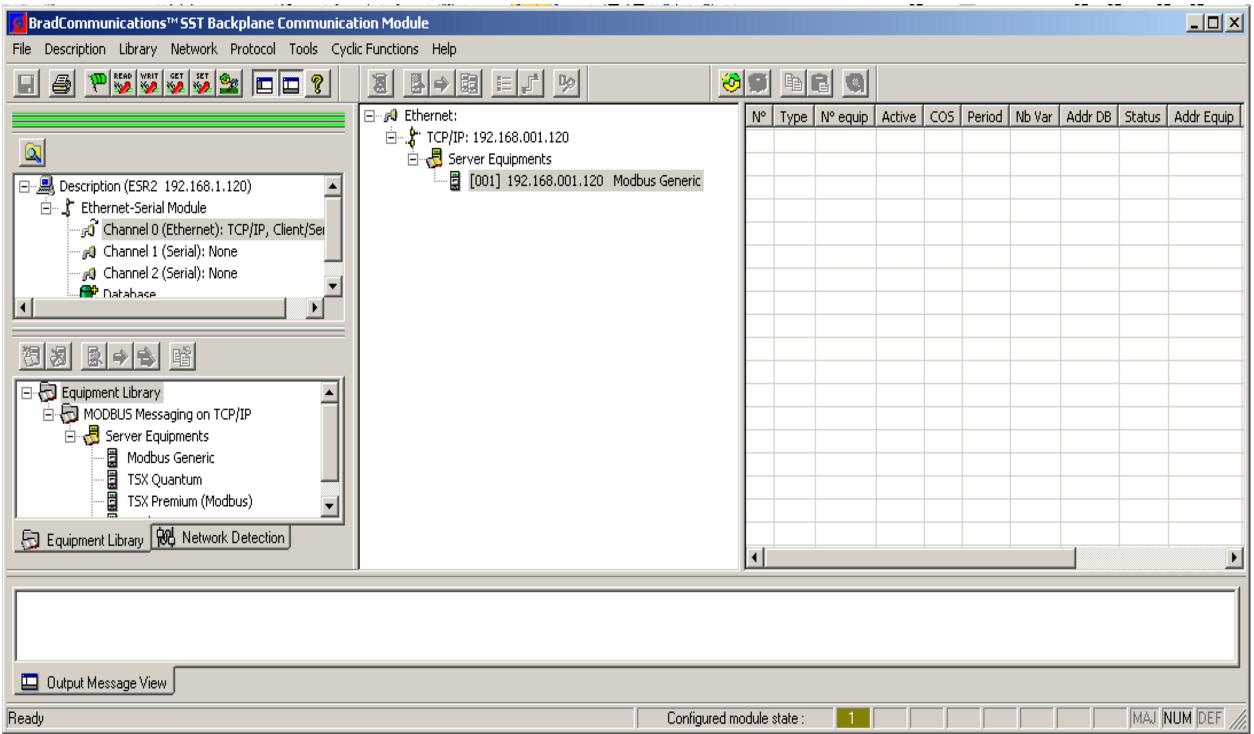
Since client and server equipment is sharing the same database, make sure different addresses are used for the source and destination address of the data.

The procedures below use a single communication module for both client and server configuration.

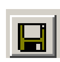
The firmware version should be 2.10.2 or higher to be able to use this feature. Follow the steps below to configure the module in extended addressing mode.

1. Complete procedures in Configure the Module as Modbus TCP Client/Server Equipment
2. After completing the steps in Configure the Module as Modbus TCP Client/Server Equipment section, the Console application main screen should be displayed with Port 0 (TCP/IP) configured as Client and Modbus TCP server equipment with IP address 192.168.1.120 is added. See Figure_69.

Figure 69: Console Main Screen with Client / Server Equipment Configuration.



3. Complete the instruction in Changing to Extended Addressing Mode section.

4. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

5. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

6. Start creating cyclic function by clicking . The cyclic function created will provide automatic data exchange between the client and server equipment. create both read and write cyclic function to automatically read and write data from/to the server equipment variable addresses. The function code as well as the size of the data to be read is part of the cyclic function parameters

   The detail of each parameter in discussed in Cyclic Function Configuration Parameters and Table 37.

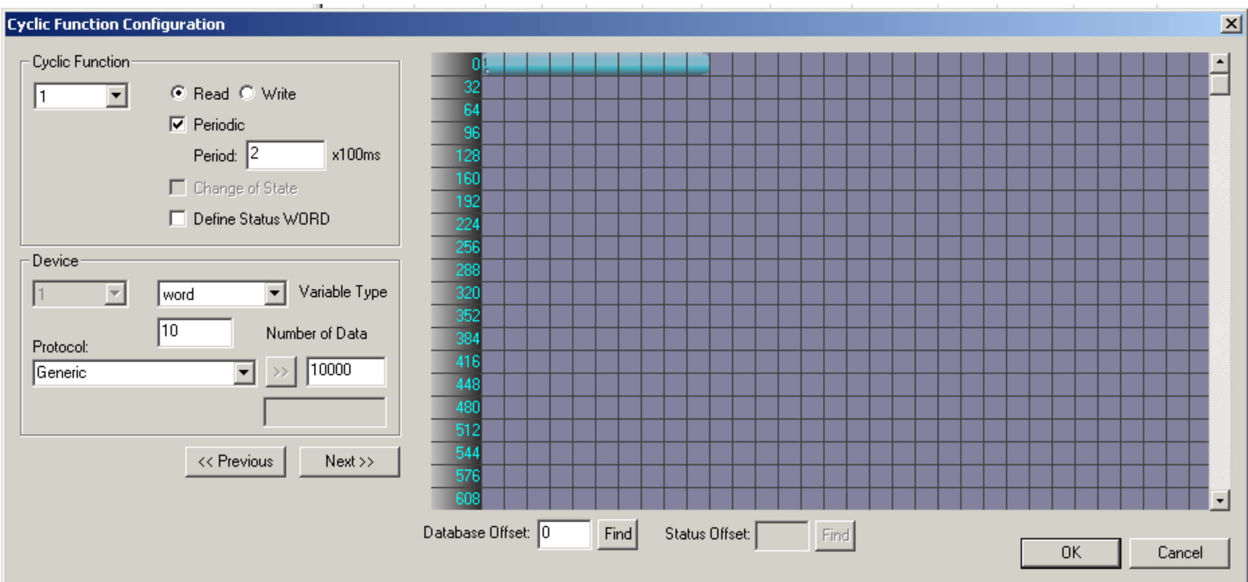
   a. Create cyclic function #1 with the parameter below.
      i. Use default cyclic function number provided (1 should be the default if this the first cyclic function)
      ii. Select Read cyclic function type
      iii. Check the Periodic setting
      iv. Use default period of 2 x 10ms = 20ms
      v. Number of data is 10

vi. Protocol is Generic

vii. Device First variable address is 10000 (server equipment address)

viii. Database offset is 0 (client equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below.



b. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

c. Create cyclic function #2 with the parameter below

i. Use default cyclic function number provided (2 should be the default now)

ii. Select Read cyclic function type

iii. Check the Periodic setting

iv. Use default period of 2 x 10ms = 20ms

v. Number of data is 10

vi. Protocol is Generic

vii. Device First variable address is 11000 (server equipment address)

viii. Database offset is 5000 (client equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.

The difference between the first and second cyclic function is the database offset location in the client.

<u>1st cyclic function database address is located in the mapped Input table address 0-1599</u>

<u>2nd cyclic function database address is located outside of Input and Output mapped table address</u>

e. Create cyclic function #3 with the parameters below:

i. Use default cyclic function number (3 should be the default now)

ii. Select Write cyclic function type
iii. Check the Periodic setting
iv. Use default period of 2 x 10ms = 20ms
v. Number of data is 10
vi. Protocol is Generic
vii. Device First variable address is 12000 (server equipment address)
viii. Database offset is 1600 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
g. Create cyclic function #4 with the parameter below
   i. Use default cyclic function number (4 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 10
   vi. Protocol is Generic
   vii. Device First variable address is 13000 (server equipment address)
   viii. Database offset is 6000 (client equipment address)

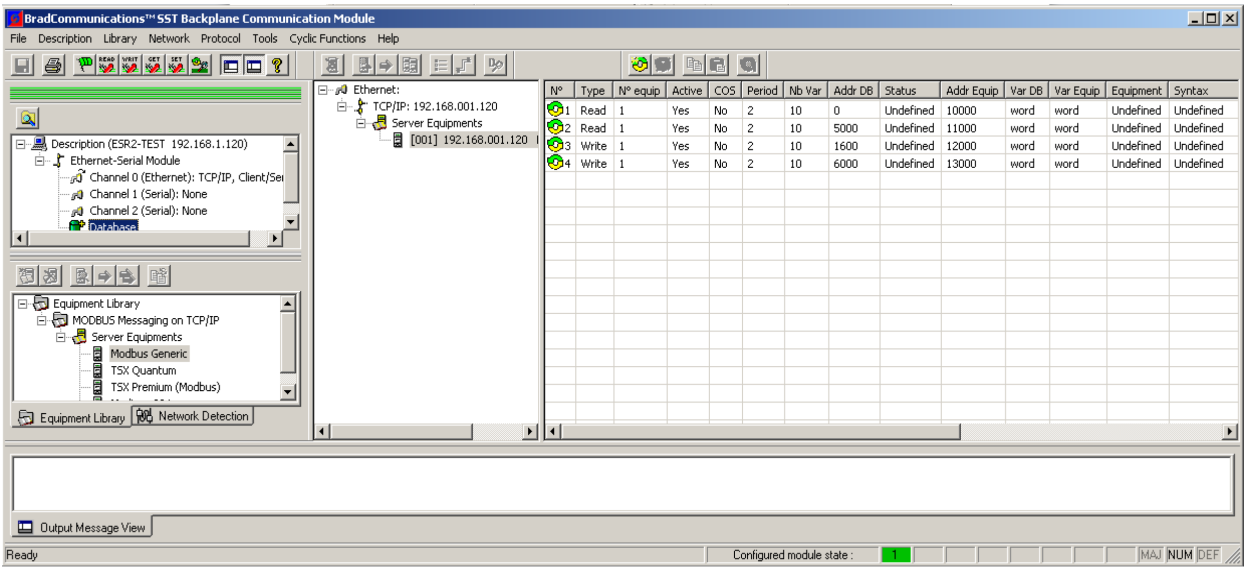
h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

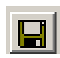
   The difference between the first and second cyclic function is the database offset location in the client.

   3rd cyclic function database address is located in the mapped Output table address 1600-3199

   4th cyclic function database address is located outside of the mapped Input and Output table address

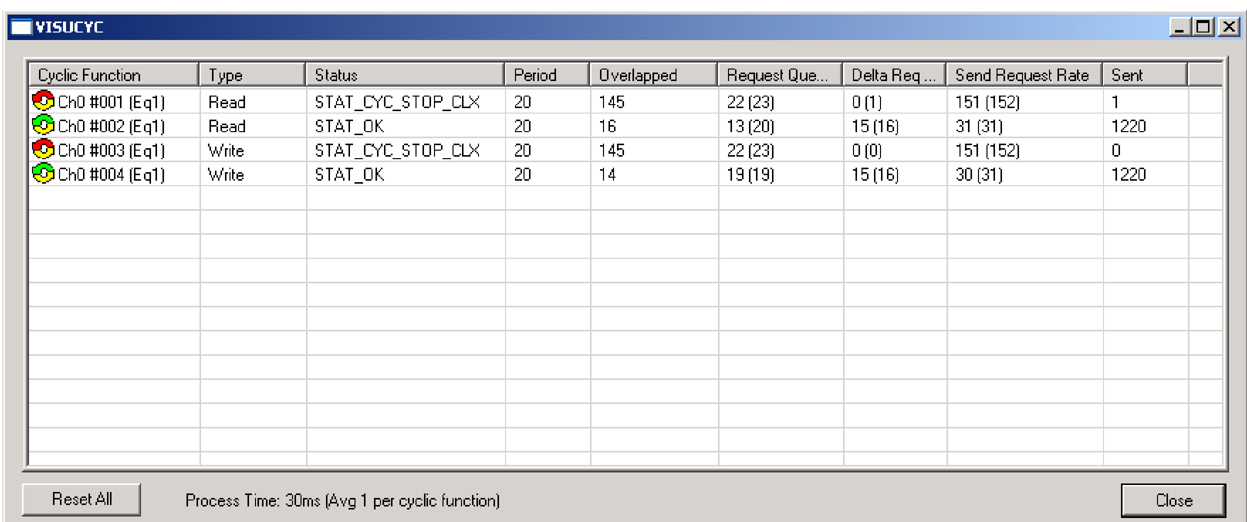
7. After creating all the cyclic function you console window should have similar display as below.



8. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

9. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

10. If initialization is successful, run the Visucyc application to check the status of each cyclic function. You should have similar display as below.



*Both 1st and 3rd cyclic function status is 135 – **Not active / not running***

*Both 2nd and 4th cyclic function status is 0 – **active / running***

Note

The module will not run any cyclic function configured in the database address mapped to INPUT, OUTPUT and STATUS Table when there is no backplane connection.

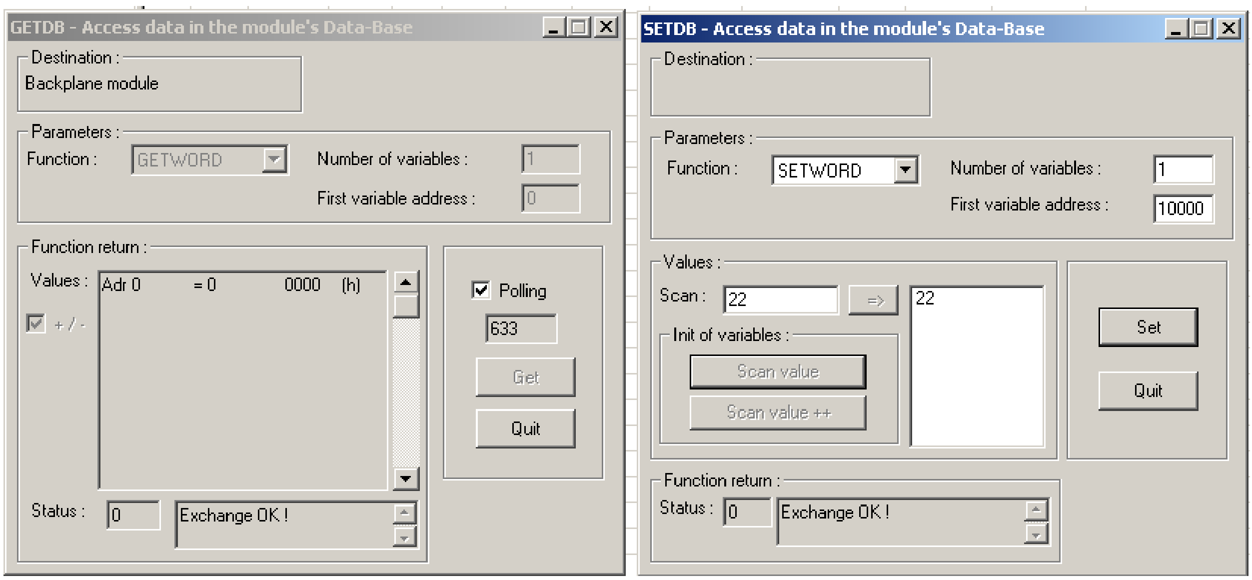
11. To check if active cyclic function is running properly, follow this brief diagnostic below.

   a. Locate and run 

and 

application.

   b. To test cyclic function # 1, fill in the values as shown below.

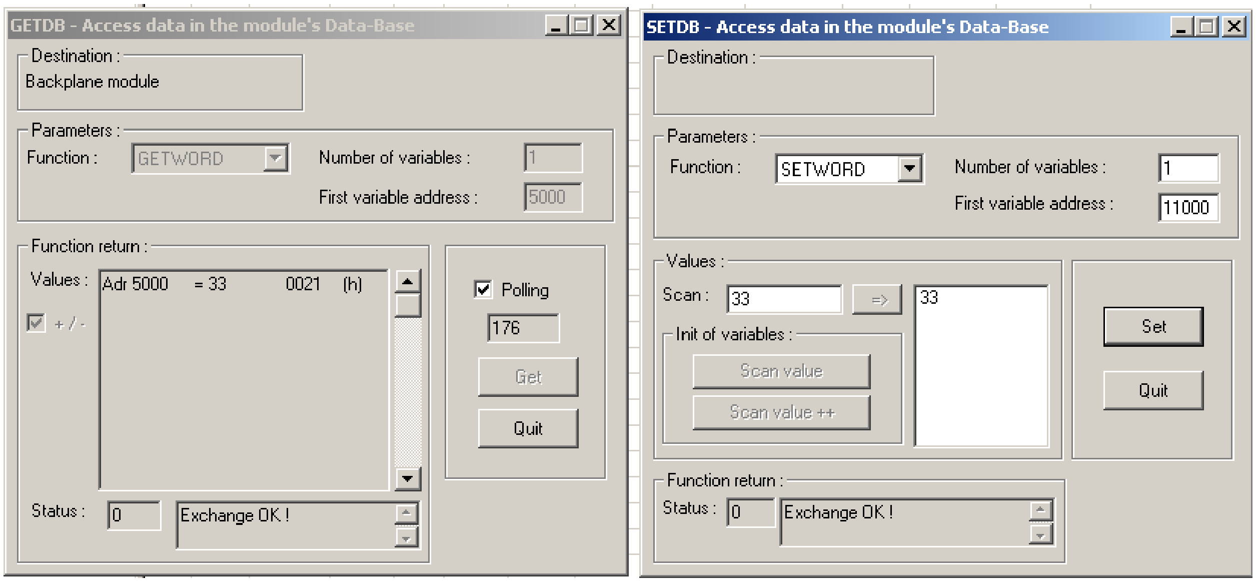


SetDB sets the server equipment address 10000 with value 22.

GetDB reads the client equipment database address 0 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 0.

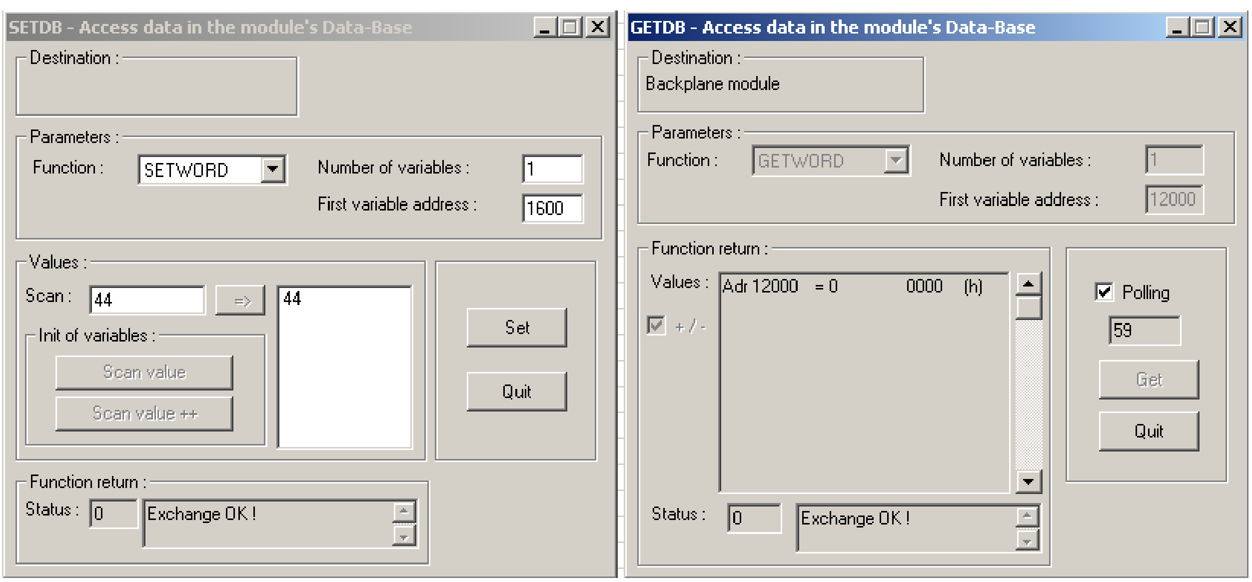
   c. To test cyclic function # 2, fill in values as shown below.



SetDB sets the server equipment address 11000 with value 33.

GetDB reads the client equipment database address 5000 with value shown as 33. If the data shown is correctly, the setup is working properly.

d. To test cyclic function # 3, fill in the values as shown below.

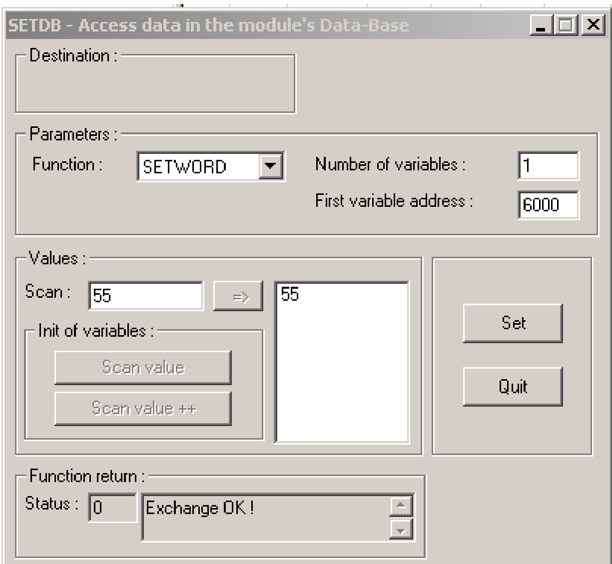


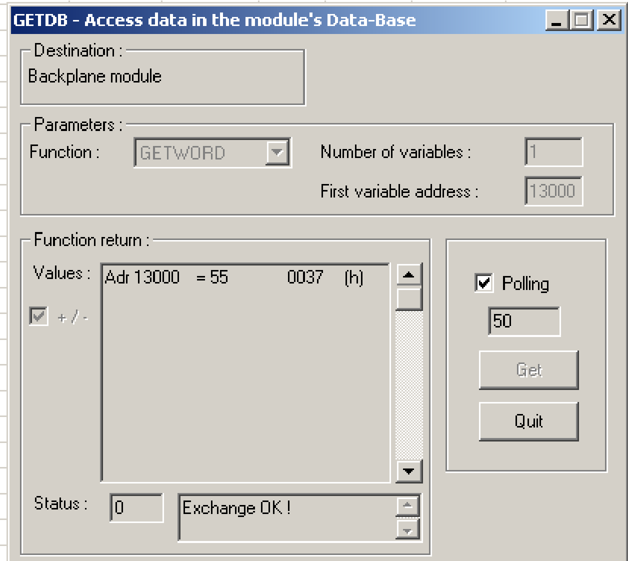
SetDB sets the client database address 1600 with value 44.

GetDB reads the server equipment database address 12000 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 12000.

e. To test cyclic function #4, fill in the values shown below.





SetDB sets the client database address 6000 with value 55.

GetDB reads the server equipment database address 13000 with value shown as 55. If the data shows the correct value, then the setup is working properly.

In the next set of procedures, the module will communicate with the ControlLogix CPU. When backplane connection becomes active, all cyclic function with **"not active / not running"** state will switch to "**active/running**".

With Backplane Connection:

Multiple sample configurations using different parameters will be provided in the following examples:

Example 1: Transfer 1 word of data for each write and read cyclic function using the default mapping defined in the AOI.

1. If the previous set of procedures in **No Backplane Connection were skipped,** going back and configuring the same cyclic functions 1-4 will be required. Run diagnostics until successfully.

2. Follow instruction 1-19 in Using 1756 Generic Profile section.

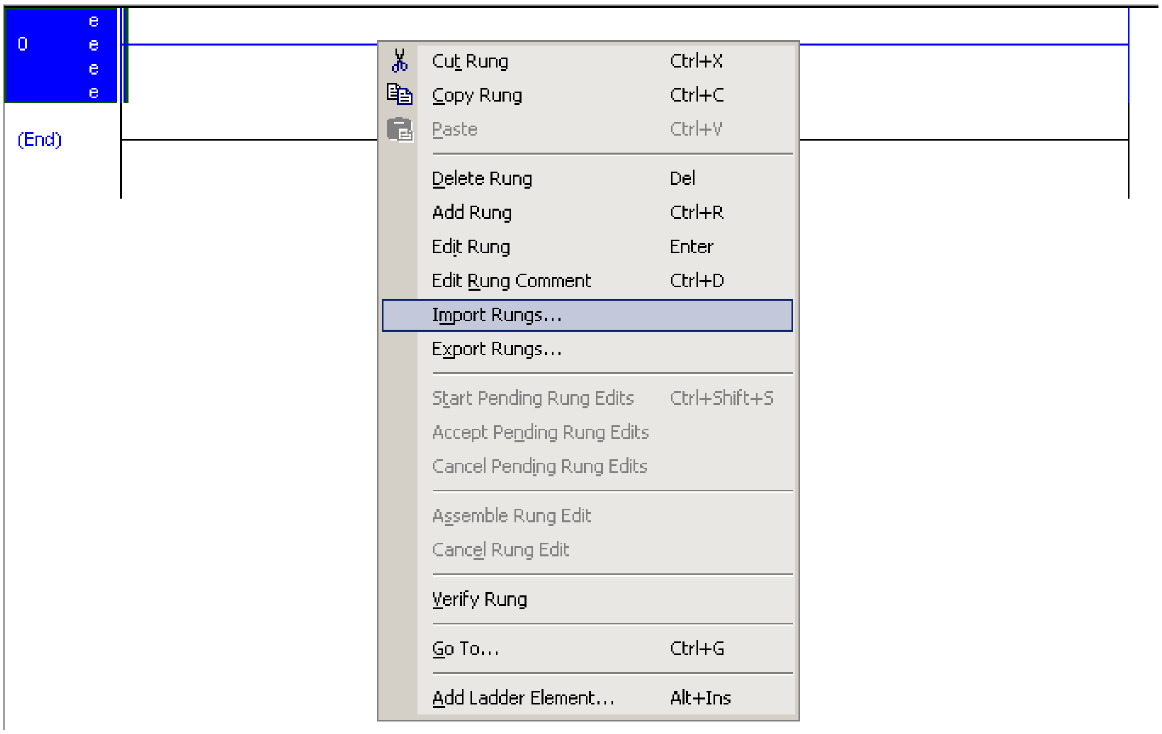
   Note

   For first time user of this feature, the console configuration should be done first prior to configuring ControlLogix connection because of the mapping of the database.

3. If smaller input and output sizes are required, please refer to section Changing the I/O connection size.

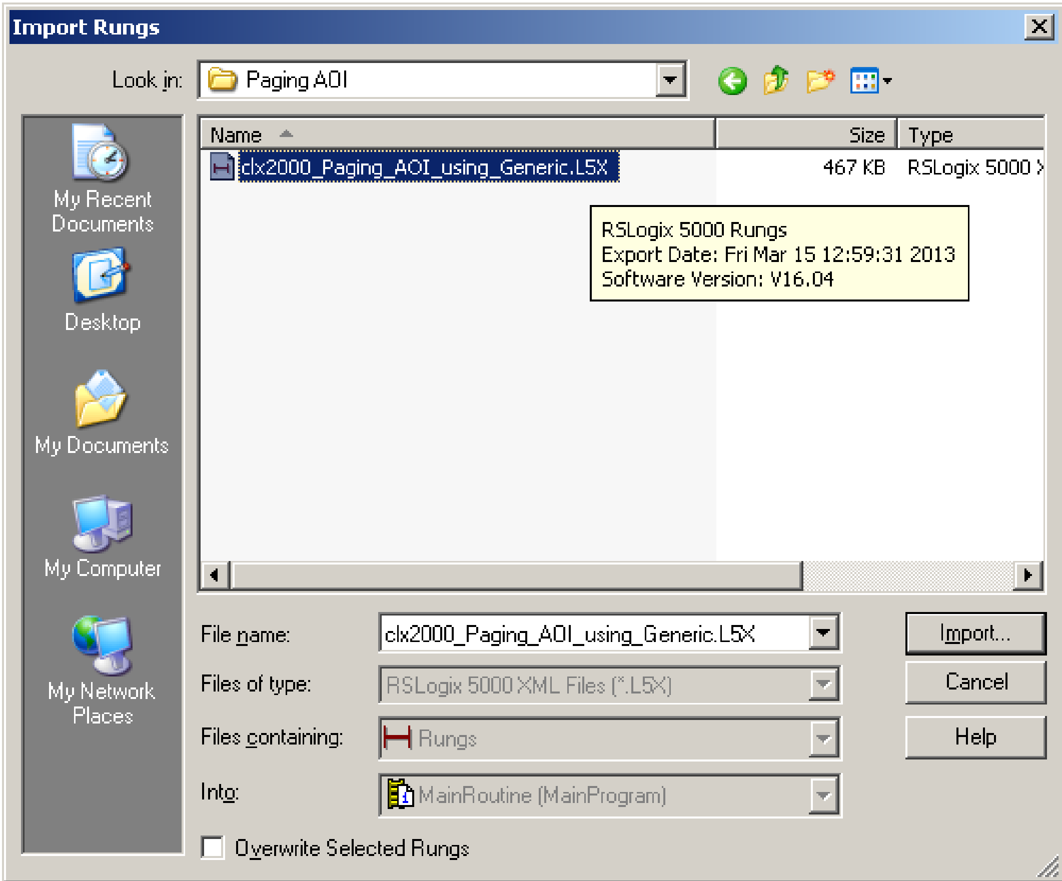
4. In RSLogix 5000 Task->MainTask, double click on MainRoutine. See below.

5. Select a rung as below and right-click on rung and select “Import Rung…”



6. Browse for the ….L5X file as below and select it and select Import.... Refer to Module's Installation Directory Location section for the location of the AOI files.

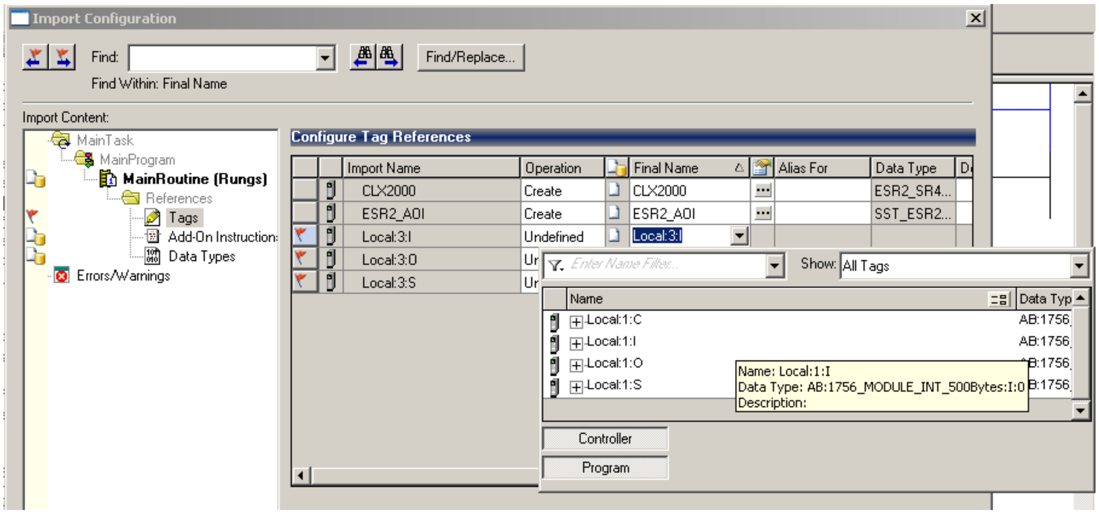
Figure 70: Adding the Paging AOI L5X file.



7. The import Configuration dialog box will appear as shown in Figure 71. For the tags that have X beside it, you will need to select the Input and Output tags for the module added in your I/O configuration.

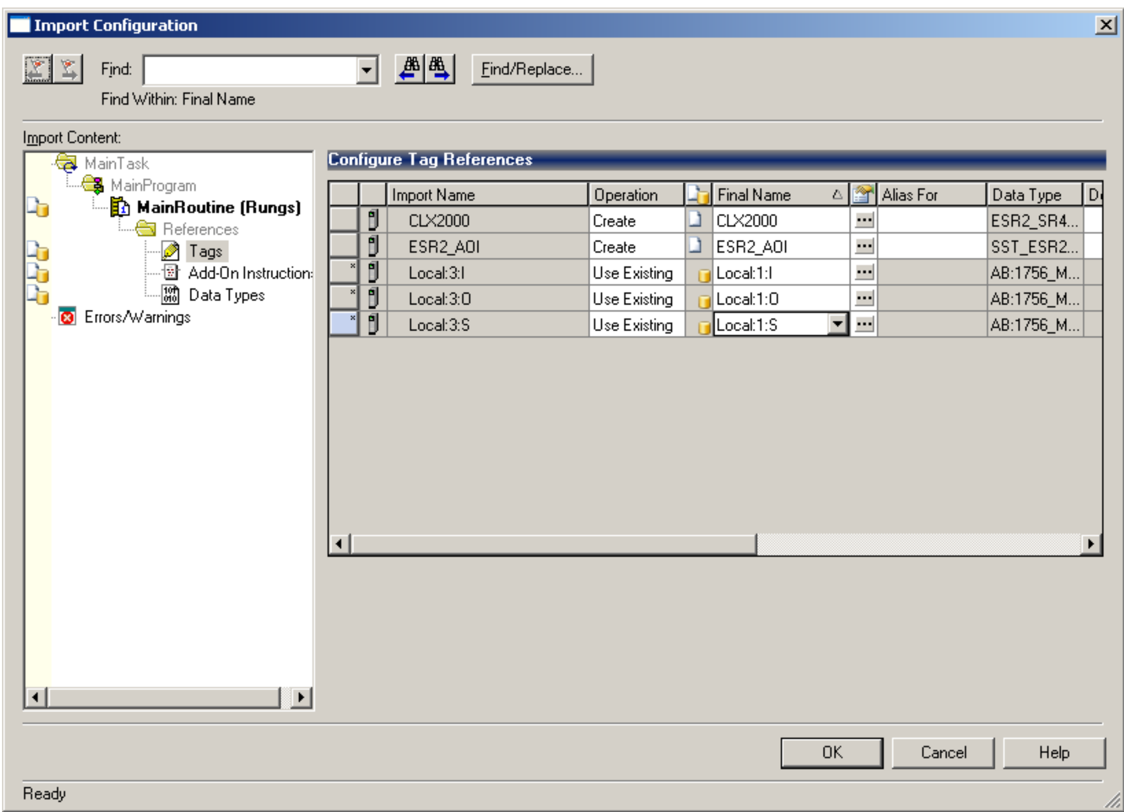
8. In this example the module was configured at slot 1 but the AOI was expecting slot 3. Update Tags Local:3:I, Local:3:O,Local:3:S to reference slot 1.

Figure 71: Changing the Slot location from the AOI.

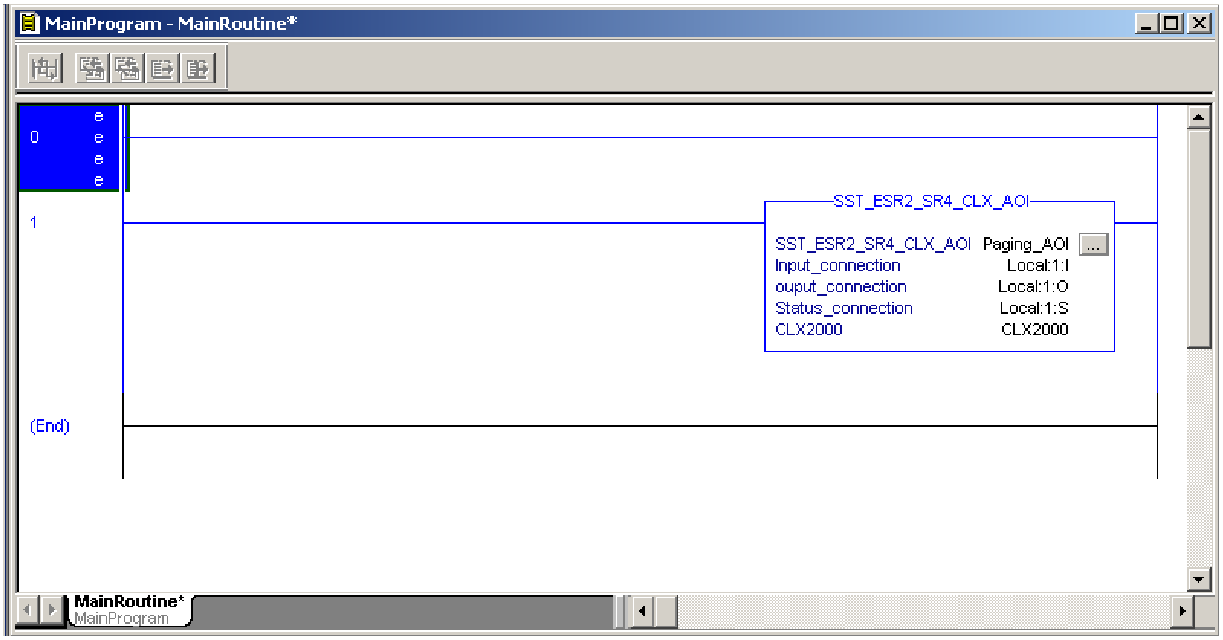


9. After correcting the last three tags click ok to begin the import.

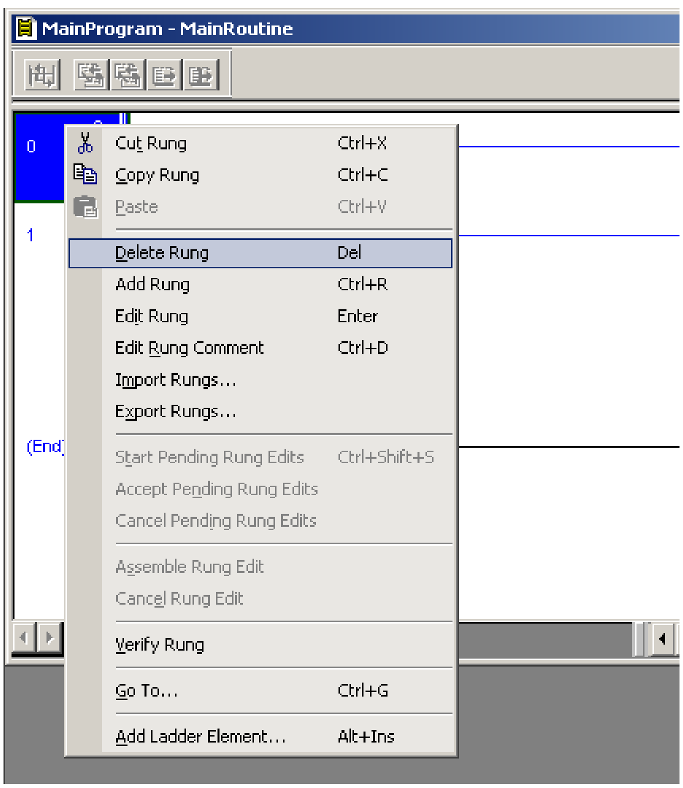
Figure 72: AOI with Changed Slot location of the module.



10. After the import is complete you should see similar display below.



11. Delete the empty rung as shown below.



12. Under Controller Tags, go to the CLX2000 tag. Expand CLX2000.CONFIGURATION as below.

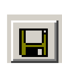
INPUT Table Start Address	= 0
INPUT Table size	= 1600
OUTPUT Table Start Address	= 1600
OUTPUT Table Size	= 1600
STATUS Table Start Address	= 3200
STATUS Table Size	= 1024

We will use our default mapped address in the AOI in this configuration.

Figure 73: AOI Controller Tags

Name	Value	Force Mask	Style	Data Type
⊟-CLX2000	{...}	{...}		ESR2_SR4_MODULE
⊟-CLX2000.CONFIGURATION	{...}	{...}		DATABASE_CONFIG
⊞-CLX2000.CONFIGURATION.Input_Table_Start_Ad...	0		Decimal	INT
⊞-CLX2000.CONFIGURATION.Input_Table_Size	1600		Decimal	INT
⊞-CLX2000.CONFIGURATION.Output_Table_Start_A...	1600		Decimal	INT
⊞-CLX2000.CONFIGURATION.Output_Table_Size	1600		Decimal	INT
⊞-CLX2000.CONFIGURATION.Status_Table_Start_A...	3200		Decimal	INT
⊞-CLX2000.CONFIGURATION.Status_Table_Size	1024		Decimal	INT
⊟-CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
⊞-CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊞-CLX2000.DATABASE_DATA.STATUS	{...}	{...}	Decimal	INT[1024]
⊟-CLX2000.NUMBER_OF_BLKS	{...}	{...}		Pages_Control
⊞-CLX2000.NUMBER_OF_BLKS.Number_Input_Blocks	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Number_Output_Blo...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Number_Status_Blo...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Input_Remainder	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Output_Remainder	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Status_Remainder	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Input_Connection_S...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Output_Connection_...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Status_Connection...	0		Decimal	INT

13. INPUT table address and size, OUTPUT table address and size and STATUS table address and size should match with the database settings for INPUT, OUTPUT and STATUS in console configuration. To check that it matches in the Console Configuration Description Area, locate and double click . Change to the values shown above if it's different. (Refer to Changing to Extended Addressing Mode) for more details.

14. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

Note

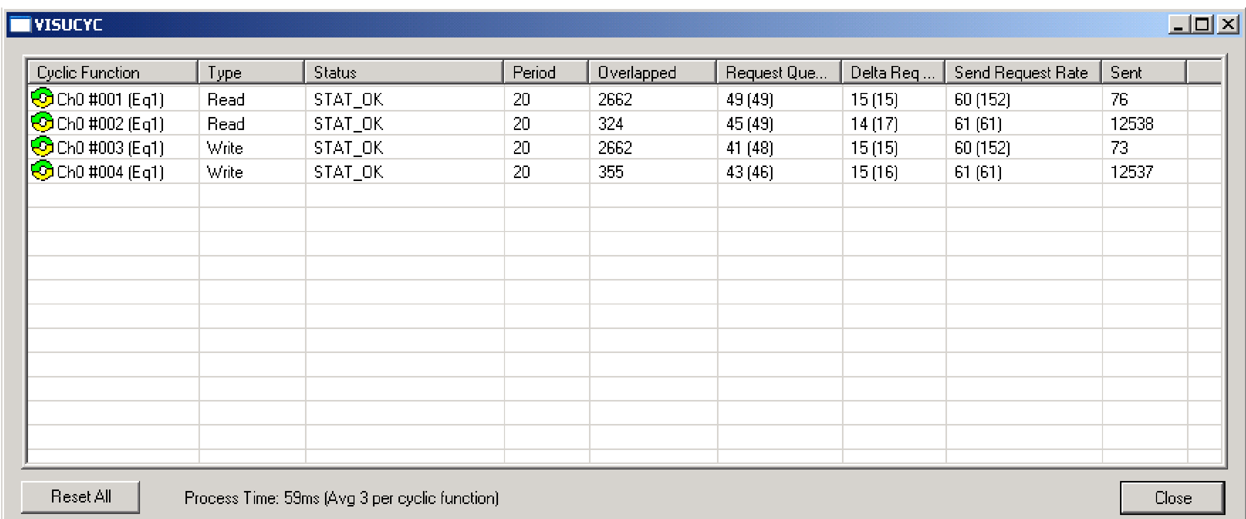
The AOI instruction will not run if there is a mismatch between the CONFIGURATION tags and the database settings in console configuration. A mismatch is indicated when Tag CLX2000.VALID_CONFIGURATION = 0. Expand Tag CLX2000 and it's the very last member of this structure. The AOI Tag CLX2000.VALID_CONFIGURATION = 1 condition must be met before the AOI will execute.

15. To see the input, output and status tags, expand the CLX2000.DATABASE_DATA as shown in Figure_73. These are the tags the AOI will use.

16. Save the configuration file and download it to the CLX Module.

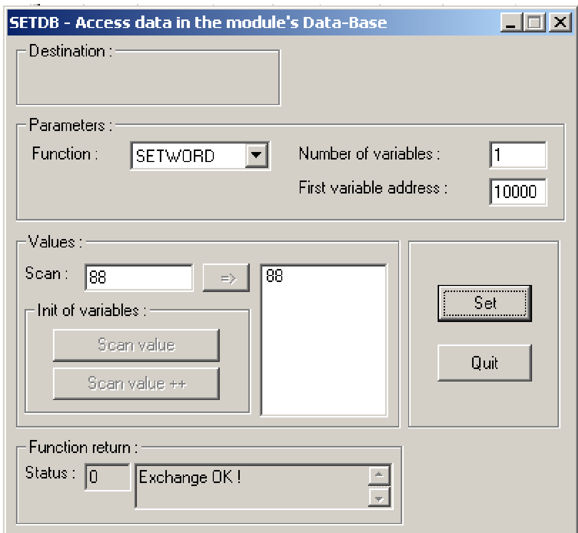
17. From the Console Application window, locate and launch VisuCyc, all the cyclic functions are in active status. See below.

Figure 74:   Visual Cyclic Function Display After Backplane Connection



Cyclic Function	Type	Status	Period	Overlapped	Request Que...	Delta Req ...	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_OK	20	2662	49 (49)	15 (15)	60 (152)	76
Ch0 #002 (Eq1)	Read	STAT_OK	20	324	45 (49)	14 (17)	61 (61)	12538
Ch0 #003 (Eq1)	Write	STAT_OK	20	2662	41 (48)	15 (15)	60 (152)	73
Ch0 #004 (Eq1)	Write	STAT_OK	20	355	43 (46)	15 (16)	61 (61)	12537

18. Test cyclic function 1 & 3 to check if values written and read from the mapped database address are exchanged properly. To test cyclic function #1, fill in the values as shown below.



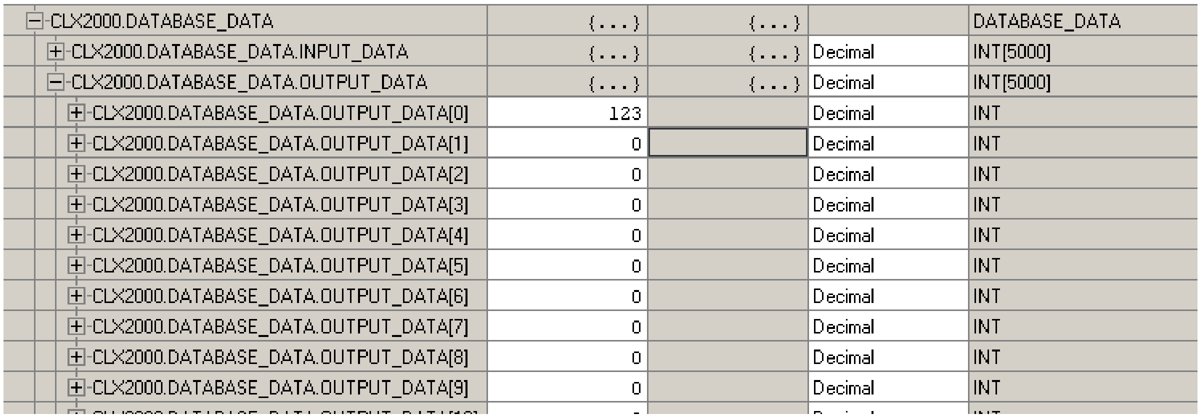
SetDB sets the server equipment address 10000 with value 88.

19. In RSLogix 5000 Software, expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value set in the server equipment address 10000 using SetDB should appear. See below.

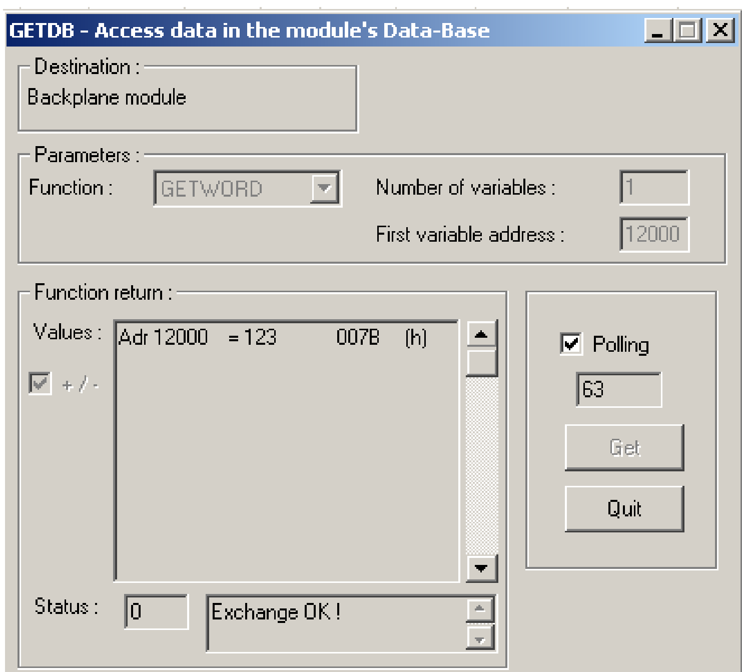
CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.INPUT_DATA[0]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[2]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[3]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[4]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[5]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[6]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[7]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[8]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[9]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[10]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[11]	0		Decimal	INT

20. If the values are correct, the backplane configuration is working properly.

21. To test cyclic function #3, expand CLX2000.DATABASE_DATA.OUTPUT_DATA and write 123 into offset 0 (offset 5000 in database)



Run ***GetDB*** and read the values as offset 12000.



22. If the values are correct, the backplane configuration is working properly.
23. Start adding the test of the desired configuration.

If Modification to this AOI is required to suit the required application, then the following details are available. The first 4 words of the ControlLogix controller output (Local:Slot:O.Data[0-3]) as shown below are reserved for handshaking between the ControlLogix controller and module.

The remaining output data Local:Slot:O:Data[4 – 247] is for transferring data to the module at the specified database offset which would be offset 1600 in the example below. The examples below use maximum connections sizes (250 INT Input, 248 INT Output, 250 INT Status)

Output Table:

Local:2:O.Data	{...}	{...}	Decimal	INT[248]
Local:2:O.Data[0]	128		Decimal	INT
Local:2:O.Data[1]	0		Decimal	INT
Local:2:O.Data[2]	1600		Decimal	INT
Local:2:O.Data[3]	1024		Decimal	INT

The following Output parameters are as follows:

Local:Slot:O.Data[0] is used for sending commands to the module. (128 = I/O command)Local:Slot:O:Data[1] is used to tell the module what area of Database Input area to map to the controller input file. The database input area starts at 0 and has a size of 1600.

Local:Slot:O:Data[2] is used tell the module what area of the Database output area will be used to to write to from the controller output file. The database output area starts at 1600 and has a size of 1600.

Local:Slot:O:Data[3] is used to tell the module what area of the database status area to map to the controller status file. The database status area starts at 3200 and has a size of 1024.

The first 3 words in input file (Local:Slot:I.Data[0-2] are reserved for returning the module command, database input and output offset to the ControlLogix controller to indicate that the module is currently updating data at the these current database offsets.

An example below indicates that the module is in Data Exchange mode, the controller input table is being updated from database offset 0 – 245, the controller output data is being written to the module's database output area starting as address 1600.

Input Table:

Local:2:I.Data	{...}	{...}	Decimal	INT[250]
Local:2:I.Data[0]	128		Decimal	INT
Local:2:I.Data[1]	0		Decimal	INT
Local:2:I.Data[2]	1600		Decimal	INT
Local:2:I.Data[3]	0		Decimal	INT

The following Input parameters are as follows:

Local:Slot:I.Data[0] is used for indicating when the module has processed the command sent to it from Local:Slot:Slot:O.Data[0]). When this = 128 the module is in Data Exchange Mode.

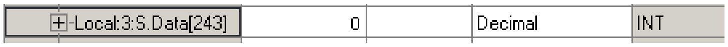
Local:Slot:I:Data[1] is used to indicate the database input offset that is currently mapped into the controller input table.

Local:Slot:I:Data[2] is used to indicate the database output that is currently mapped to the controller output table.

Local:Slot:I:Data[3] is a reserved address and cannot be used.

The following status address is used to indicate the database status offset that is currently mapped to the controller status table at offsets 0 - 199.

Status Table:



Example 2: Transfer 100 words of data per cyclic function, to configure 5 write cyclic function and 5 read cyclic function using the default mapping defined in the AOI.

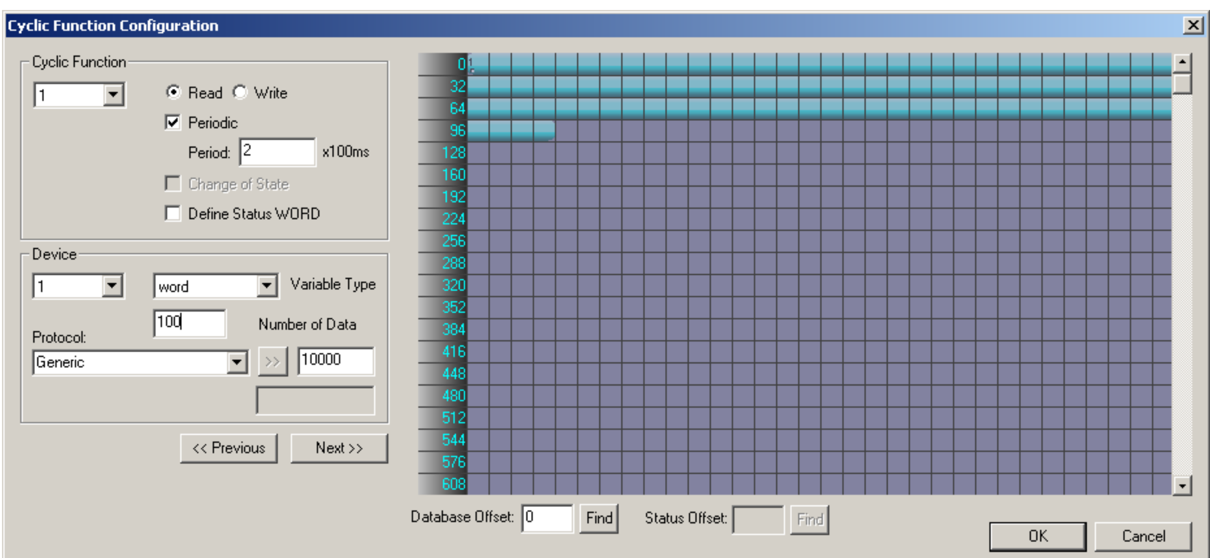
1. Example 2 is a continuation of Example 1, therefore all console configurations, backplane connection and the quick test are done successfully.

2. From the console application window:

    a. Modify cyclic function #1 with the parameter below:

        i. default cyclic function number provided
        ii. Select Read cyclic function type
        iii. Check the Periodic setting
        iv. Use default period of 2 x 10ms = 20ms
        v. Number of data is 100
        vi. Protocol is Generic
        vii. Device First variable address is 10000 (server equipment address)
        viii. Database offset is 0 (client equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below.



    b. Click “OK”, and the first cyclic function will be shown in the cyclic function list view
    c. Modify cyclic function #2 with the parameter below:

        i. Use default cyclic function number (2 should be the default now)

ii. Select Read cyclic function type
iii. Check the Periodic setting
iv. Use default period of 2 x 10ms = 20ms
v. Number of data is 100
vi. Protocol is Generic
vii. Device First variable address is 11000 (server equipment address)
viii. Database offset is 100 (client equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.
e. Modify cyclic function #3 with the parameters below:
   i. Use default cyclic function number (3 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 100
   vi. Protocol is Generic
   vii. Device First variable address is 12000 (server equipment address)
   viii. Database offset is 1600 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
g. Modify cyclic function #4 with the parameter below
   i. Use default cyclic function number (4 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 100
   vi. Protocol is Generic
   vii. Device First variable address is 13000 (server equipment address)
   viii. Database offset is 1700 (client equipment address)

h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

3. Add 6 more cyclic functions to this configuration.
   a. Create cyclic function #5 with the parameter below.
      i. default cyclic function number
      ii. Select Read cyclic function type
      iii. Check the Periodic setting
      iv. Use default period of 2 x 10ms = 20ms
      v. Number of data is 100

vi. Protocol is Generic
vii. Device First variable address is 14000 (server equipment address)
viii. Database offset is 1000 (client equipment address)

b. Click “OK”, and the first cyclic function will be shown in the cyclic function list view
c. Create cyclic function #6 with the parameter below
    i. Use default cyclic function number
    ii. Select Read cyclic function type
    iii. Check the Periodic setting
    iv. Use default period of 2 x 10ms = 20ms
    v. Number of data is 100
    vi. Protocol is Generic
    vii. Device First variable address is 15000 (server equipment address)
    viii. Database offset is 1100 (client equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.
e. Create cyclic function #7 with the parameters below:
    i. Use default cyclic function number
    ii. Select Read cyclic function type
    iii. Check the Periodic setting
    iv. Use default period of 2 x 10ms = 20ms
    v. Number of data is 100
    vi. Protocol is Generic
    vii. Device First variable address is 16000 (server equipment address)
    viii. Database offset is 1200 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
g. Create cyclic function #8 with the parameter below
    i. Use default cyclic function number
    ii. Select Write cyclic function type
    iii. Check the Periodic setting
    iv. Use default period of 2 x 10ms = 20ms
    v. Number of data is 100
    vi. Protocol is Generic
    vii. Device First variable address is 17000 (server equipment address)
    viii. Database offset is 2500 (client equipment address)

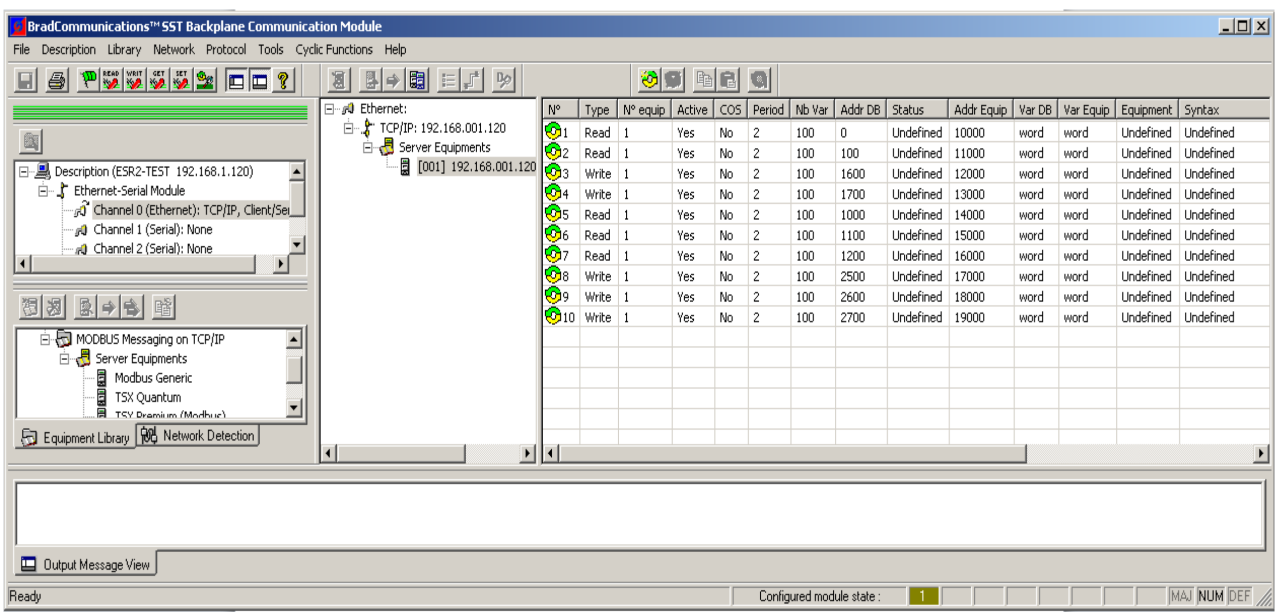
h. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
i. Create cyclic function #9 with the parameter below
    i. Use default cyclic function number

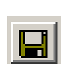
ii. Select Write cyclic function type
iii. Check the Periodic setting
iv. Use default period of 2 x 10ms = 20ms
v. Number of data is 100
vi. Protocol is Generic
ix. Device First variable address is 18000 (server equipment address)
vii. Database offset is 2600 (client equipment address)

j. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
k. Create cyclic function #10 with the parameter below
i. Use default cyclic function number
ii. Select Write cyclic function type
iii. Check the Periodic setting
iv. Use default period of 2 x 10ms = 20ms
v. Number of data is 100
vi. Protocol is Generic
vii. Device First variable address is 19000 (server equipment address)
viii. Database offset is 2700 (client equipment address)

4. After all the cyclic functions are created, you should have similar display below.

Figure 75: Cyclic Function List View

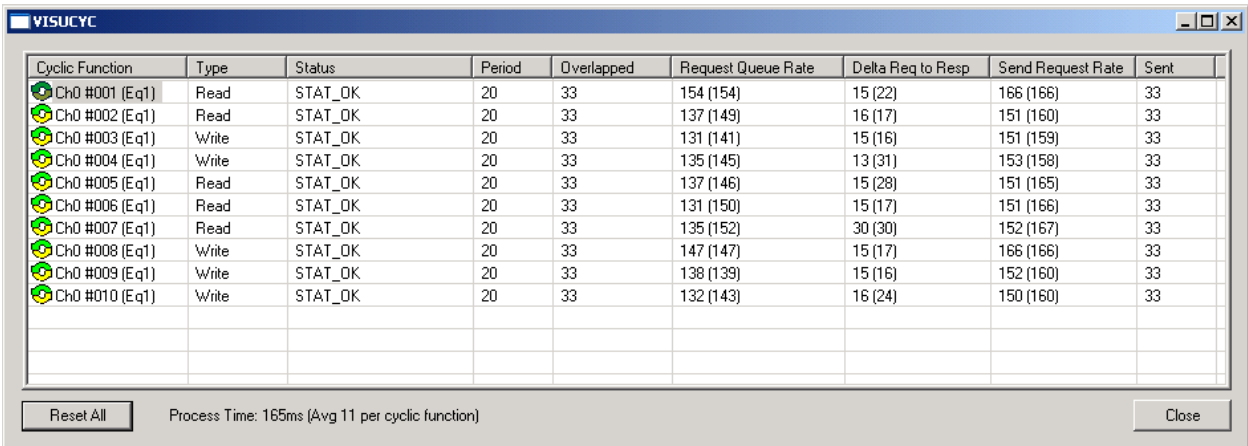


5. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

6. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

7. If initialization is successful, run the Visucyc application to check the status of each cyclic function. A similar display as below.

Figure 76: Visucyc Display with all Cyclic Function Running OK



Cyclic Function	Type	Status	Period	Overlapped	Request Queue Rate	Delta Req to Resp	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_OK	20	33	154 (154)	15 (22)	166 (166)	33
Ch0 #002 (Eq1)	Read	STAT_OK	20	33	137 (149)	16 (17)	151 (160)	33
Ch0 #003 (Eq1)	Write	STAT_OK	20	33	131 (141)	15 (16)	151 (159)	33
Ch0 #004 (Eq1)	Write	STAT_OK	20	33	135 (145)	13 (31)	153 (158)	33
Ch0 #005 (Eq1)	Read	STAT_OK	20	33	137 (146)	15 (28)	151 (165)	33
Ch0 #006 (Eq1)	Read	STAT_OK	20	33	131 (150)	15 (17)	151 (166)	33
Ch0 #007 (Eq1)	Read	STAT_OK	20	33	135 (152)	30 (30)	152 (167)	33
Ch0 #008 (Eq1)	Write	STAT_OK	20	33	147 (147)	15 (17)	166 (166)	33
Ch0 #009 (Eq1)	Write	STAT_OK	20	33	138 (139)	15 (16)	152 (160)	33
Ch0 #010 (Eq1)	Write	STAT_OK	20	33	132 (143)	16 (24)	150 (160)	33

8. Since we already have an active communication with the backplane, all cyclic functions are in "**active**" state.

Note

In this example, all cyclic functions are located in the mapped portion of the database.

9. Start testing all read cyclic function; cyclic function 1,2,5,6 & 7.

10. From the Console application window, locate and launch SetDB.

11. Enter 100 as the Number of variables, enter 10000 as the First variable address.

12. In Scan enter 1, then click Scan value ++ and then OK, setting values in server equipment address 10000-10099 with values 1-100.

13. Repeat step 11 but use First variable address 11000.

14. Repeat step 12 with value 100, then click Scan value ++ and then OK, setting values in server equipment address 11000-11099 with values 100-199.

15. Repeat step 11 but use First variable address 14000.

16. Repeat step 12 with value 50, then click 

and then OK, setting values in server equipment address 14000-14099 with values 50.

17. Repeat step 11 but use First variable address 15000.

18. Repeat step 12 with value 60, then click 

and then OK, setting values in server equipment address 15000-15099 with values 60.

19. Repeat step 11 but use First variable address 16000.

20. Repeat step 12 with value 70, then click 

and then OK, setting values in server equipment address 16000-16099 with values 70.

21. To summarize the cyclic function used address in both client and server equipment for the Read Cyclic functions, see table below.

Table 12: Read Cyclic Function Test Summary

Cyclic Function #	CPU INPUT Table Address	Values	Client Equipment Database Start Address	Server Equipment Database Start Address
1	0	0-100	0	10000
2	100	100-199	100	11000
5	1000	50	1000	14000
6	1100	60	1100	15000
7	1200	70	1200	16000

24. In RSLogix 5000 Software, expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value set in the server equipment address using SetDB. See below.

For cylic function # 1:

CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5
CLX2000.DATABASE_DATA.INPUT_DATA[0]	1		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[2]	3		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[3]	4		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[4]	5		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[5]	6		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[6]	7		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[7]	8		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[8]	9		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[9]	10		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[10]	11		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[11]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[12]	13		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[13]	14		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[14]	15		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[15]	16		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[16]	17		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[17]	18		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[18]	19		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[19]	20		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[20]	21		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[21]	22		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[22]	23		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[23]	24		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[24]	25		Decimal	INT

For cylic function # 2

CLX2000.DATABASE_DATA.INPUT_DATA[99]	100		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[100]	100		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[101]	101		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[102]	102		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[103]	103		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[104]	104		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[105]	105		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[106]	106		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[107]	107		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[108]	108		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[109]	109		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[110]	110		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[111]	111		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[112]	112		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[113]	113		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[114]	114		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[115]	115		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[116]	116		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[117]	117		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[118]	118		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[119]	119		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[120]	120		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[121]	121		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[122]	122		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[123]	123		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[124]	124		Decimal	INT :

For cyclic function # 5

CLX2000.DATABASE_DATA.INPUT_DATA[1000]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1001]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1002]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1003]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1004]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1005]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1006]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1007]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1008]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1009]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1010]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1011]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1012]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1013]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1014]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1015]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1016]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1017]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1018]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1019]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1020]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1021]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1022]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1023]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1024]	50		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1025]	50		Decimal	INT

For cyclic function # 6

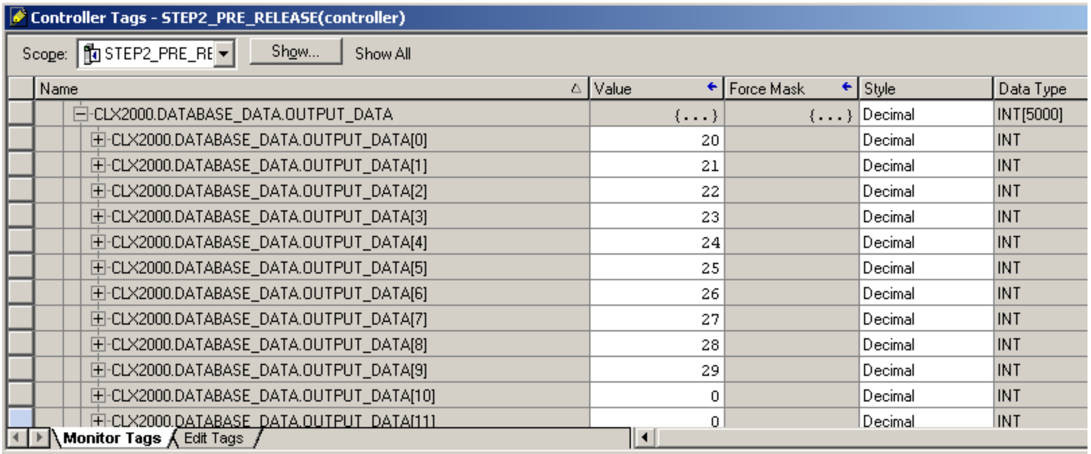
CLX2000.DATABASE_DATA.INPUT_DATA[1100]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1101]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1102]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1103]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1104]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1105]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1106]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1107]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1108]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1109]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1110]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1111]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1112]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1113]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1114]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1115]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1116]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1117]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1118]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1119]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1120]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1121]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1122]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1123]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1124]	60		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1125]	60		Decimal	INT

For cyclic function # 7

CLX2000.DATABASE_DATA.INPUT_DATA[1200]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1201]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1202]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1203]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1204]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1205]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1206]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1207]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1208]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1209]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1210]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1211]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1212]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1213]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1214]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1215]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1216]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1217]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1218]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1219]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1220]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1221]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1222]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1223]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1224]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1225]	70		Decimal	INT

25. If the same values as shown in the INPUT table, the setup is working properly
26. Now test the Write cyclic function.
27. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

For Cyclic Function #3:



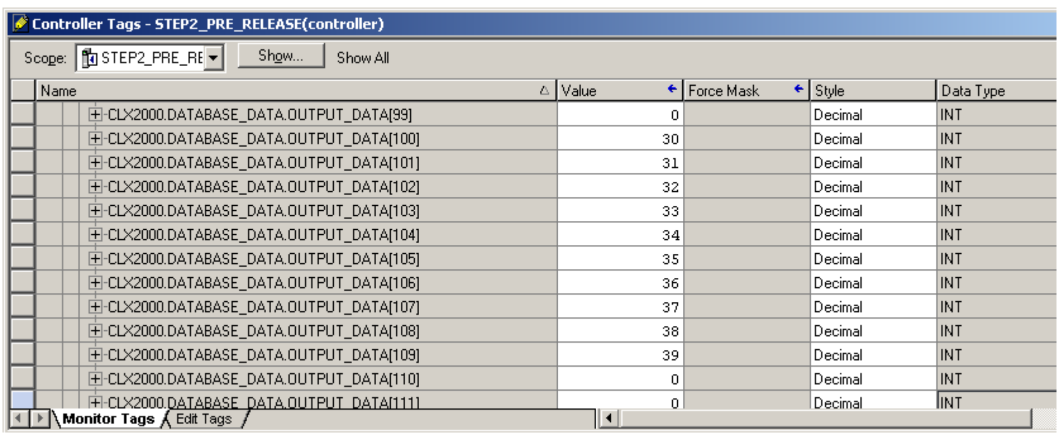
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.OUTPUT_DATA[0]	20		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1]	21		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[2]	22		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[3]	23		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[4]	24		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[5]	25		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[6]	26		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[7]	27		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[8]	28		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[9]	29		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[10]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[11]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function # 4



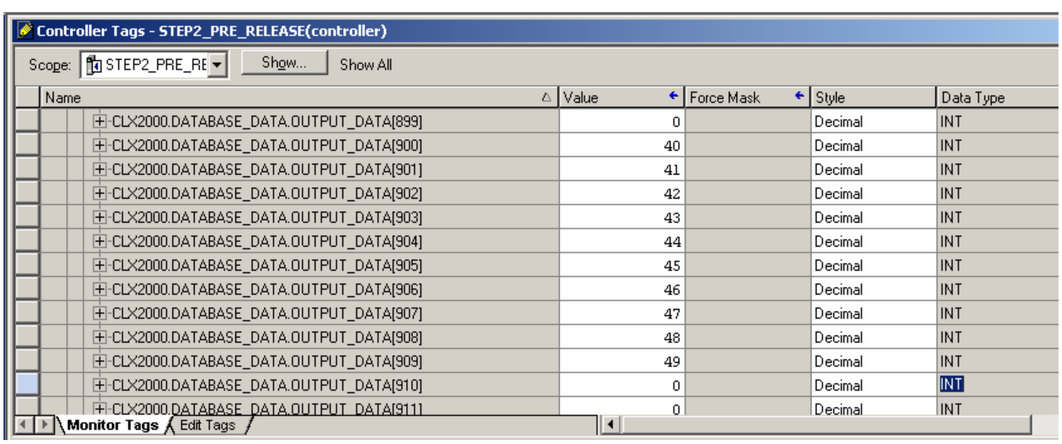
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[99]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[100]	30		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[101]	31		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[102]	32		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[103]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[104]	34		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[105]	35		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[106]	36		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[107]	37		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[108]	38		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[109]	39		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[110]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[111]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function #8:



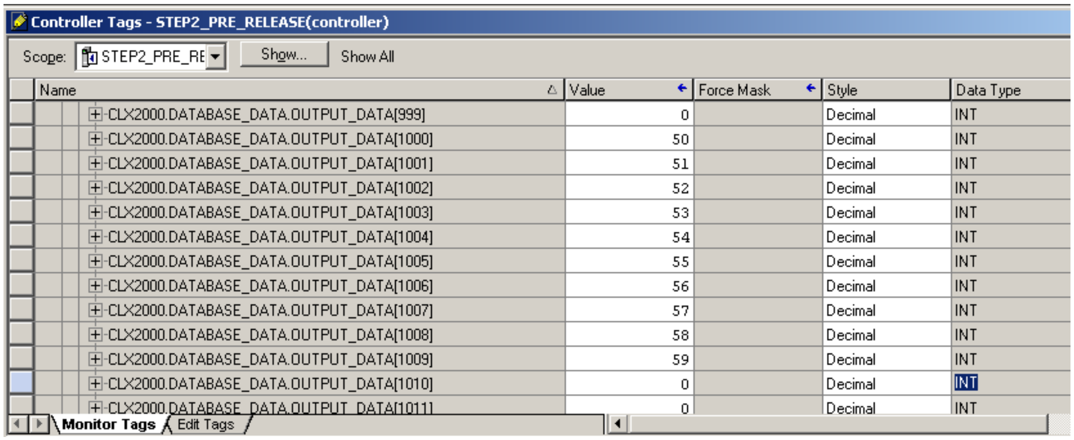
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[899]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[900]	40		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[901]	41		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[902]	42		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[903]	43		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[904]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[905]	45		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[906]	46		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[907]	47		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[908]	48		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[909]	49		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[910]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[911]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function # 9:



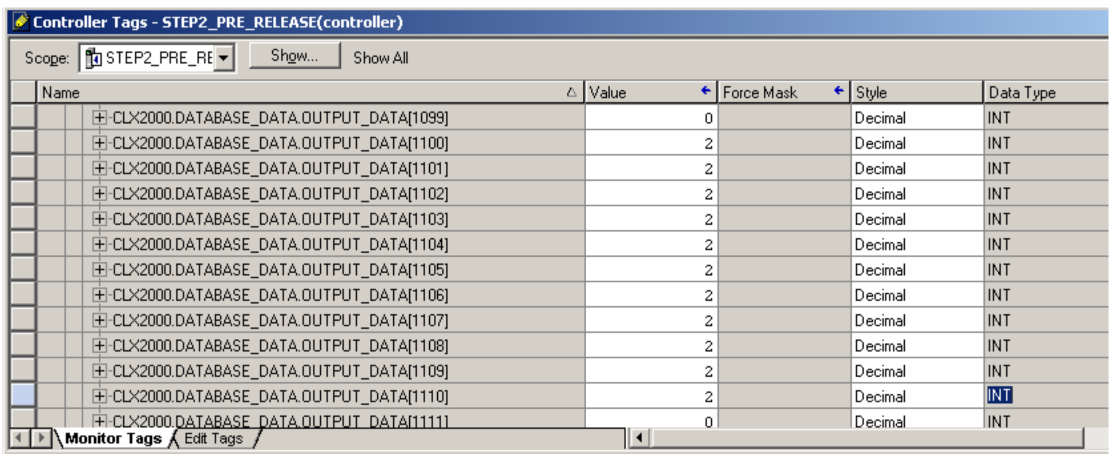
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[999]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1000]	50		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1001]	51		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1002]	52		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1003]	53		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1004]	54		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1005]	55		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1006]	56		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1007]	57		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1008]	58		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1009]	59		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1010]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1011]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function # 10:



Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[1099]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1100]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1101]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1102]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1103]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1104]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1105]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1106]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1107]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1108]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1109]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1110]	2		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1111]	0		Decimal	INT

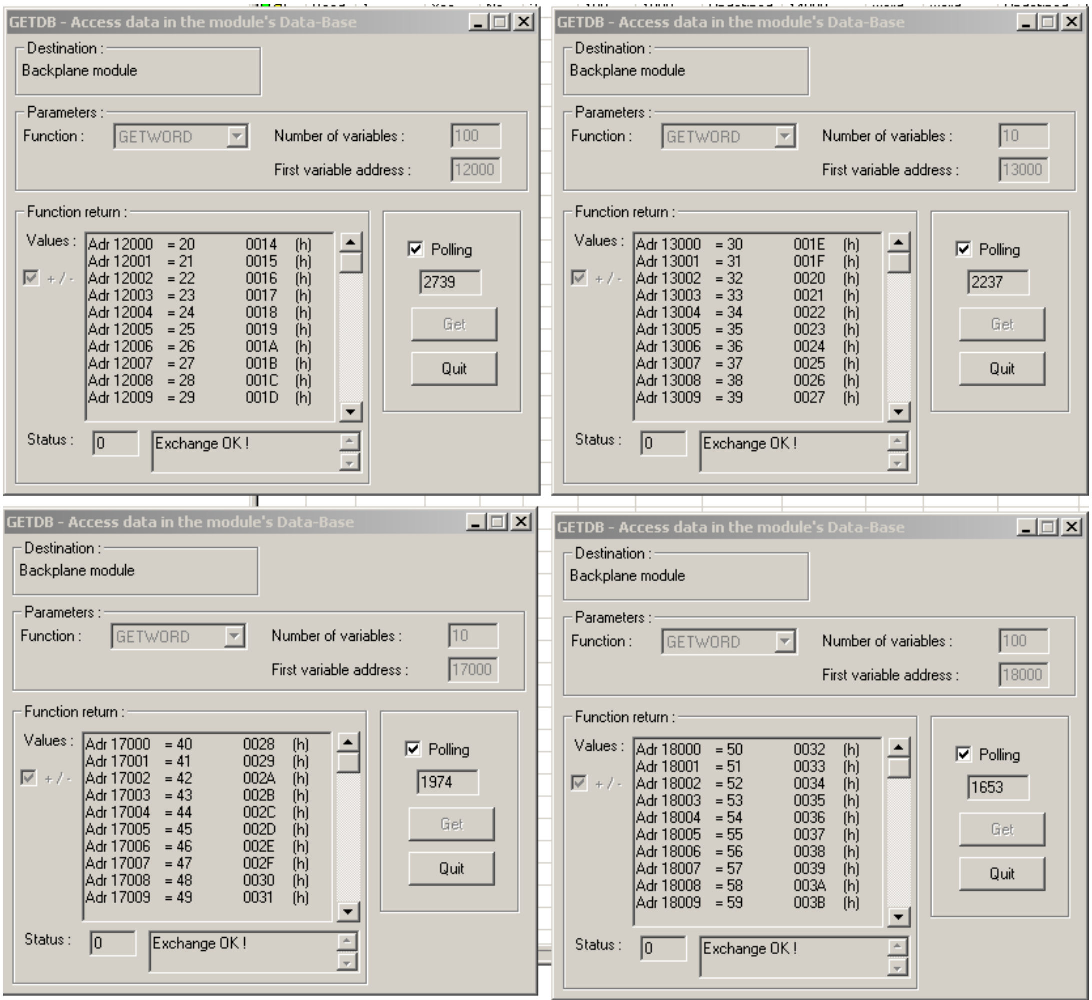
Monitor Tags / Edit Tags

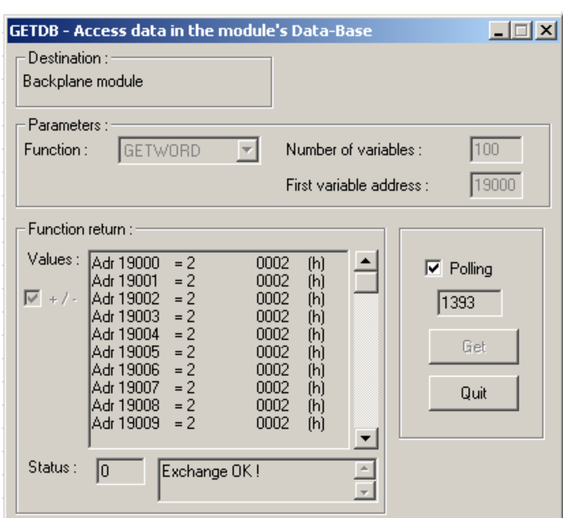
Table 13: Write Cyclic Function Test Summary

Cyclic Function #	CPU OUTPUT Table address	Values	Client Equipment Database Start Address	Server Equipment Database Start Address
3	0	20-29	1600	12000
4	100	30-39	1700	13000
8	900	40-49	2500	17000
9	1000	50-59	2600	18000
10	1100	2	2700	19000

28. Read the server equipment addresses as indicated in the table above. See result below. The results should be the same. Write 10 words for each write cyclic function, so there are only 10 non-zero values shown using GetDB for each server equipment address.

29. From the Console application window, locate and launch GetDB.





30. If the values are correct, the setup is working properly.

Example 3: Changing the database mapping size.

This sample configuration will only change the database mapping size. This will be a continuation from Example 2.

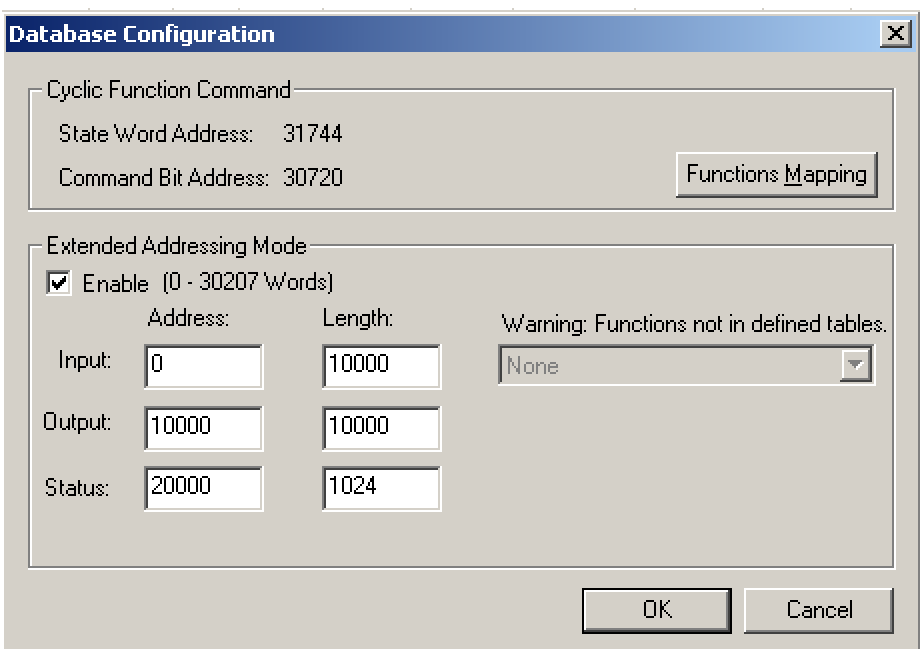
1. In your RsLogix 5000 software, change the connection state to Offline.

2. From the console application window, run Database and change the INPUT, OUTPUT and STATUS table addresses and size.

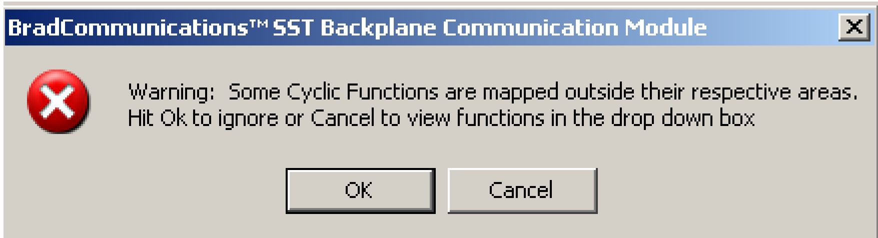
Warning

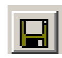
This is just a demonstration of how to change the AOI mapping address. The values that will be used here will cause the cyclic function to stop working since addresses between client and server in the same module maybe overlapping

Figure 77: Database Configuration Window



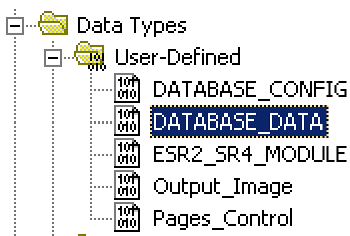
When the values are entered above, the message below will pop up, click OK to ignore,



3. TO save and download configuration, click  and . The configuration download takes about 2-3 minutes.

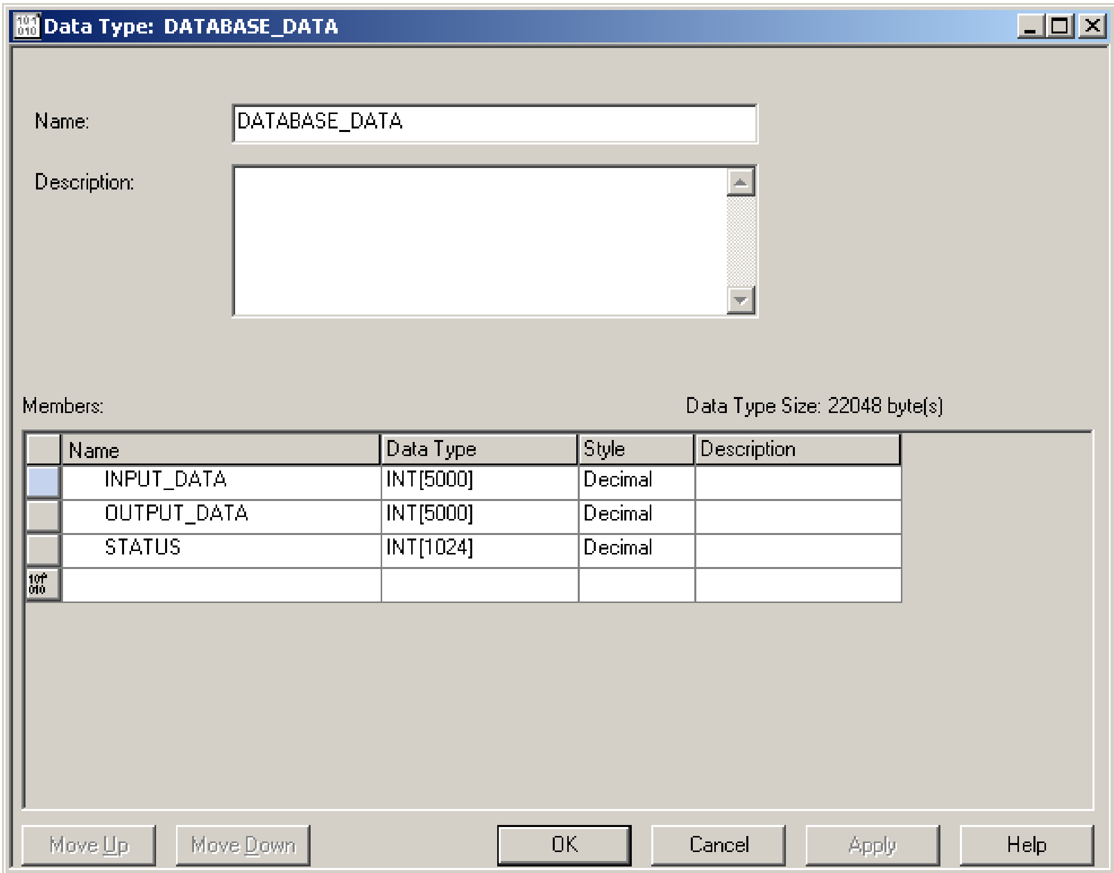
4. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

5. From the Rslogix 5000, double click DATABASE_DATA under Data Types, see below.



6. The window below should be displayed on the screen

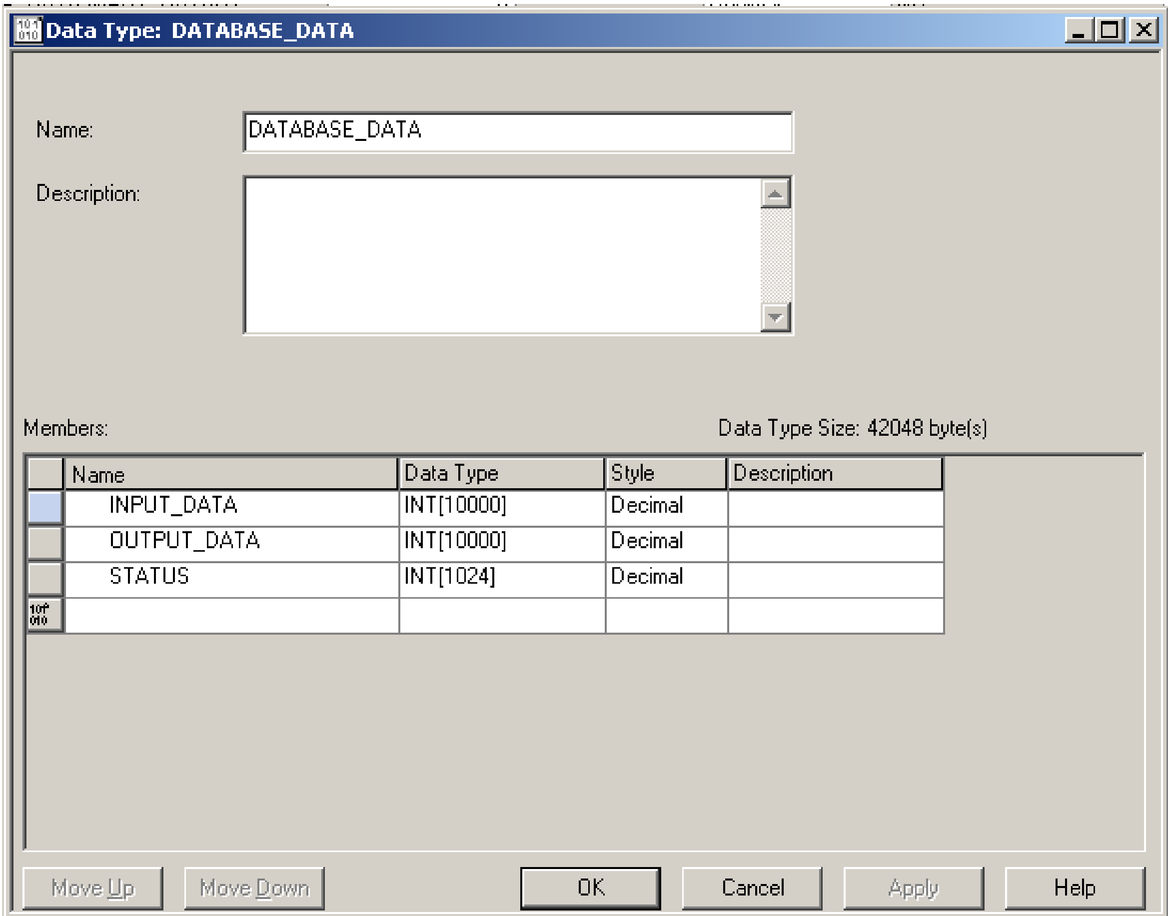
Figure 78: Database Array Size Declaration



The current AOI configuration allows up 5000 direct mapping of INPUT and OUTPUT table. Before changing the value read some important notes below:

- If the desired mapping size fits in this size, there is no need to change anything from the backplane configuration. The database configuration from the console only needs to be changed.
- If the desired mapping is greater than what is currently defined, modify it with desired values.

7. Since the INPUT and OUTPUT mapping size of 10000 words each, these sizes need to be increased. After the change, the display should be similar as below.



8. Click OK and check if CLX2000.DATABASE_DATA.INPUT_DATA and CLX2000.DATABASE_DATA.OUTPUT_DATA size arrays did change.

9. The new array size should now be 0-9999.

10. Save the configuration file and download it to the CLX Module.

6.5.2 Modbus Serial Messaging

6.5.2.1 Modbus Serial in Default Addressing Mode

When module is configured in default addressing mode, the modules database address 0-699 is mapped by default to ControlLogix INPUT, OUTPUT and STATUS Table.

No Backplane Connection:

To be able to demonstrate successfully this sample configuration, Modbus Serial Master Equipment and Modbus Serial Slave equipment are used, and they must have an active connection to be able to show data exchanges configured through cyclic function.

The SST-SR4-CLX-RLL product is used in this sample, to create cyclic function with easy to follow procedure. If there are two available modules of this product variant, configure one as Modbus Serial Master and the other one as Modbus Serial Slave. If there is only one module, use a single communication module configured both as Modbus Serial Master and Modbus Serial Slave. Cyclic functions can only be created when module is running a master equipment configuration.

Note

Since master and slave equipment is sharing the same database, make sure different addresses are used for the source and destination address of the data.

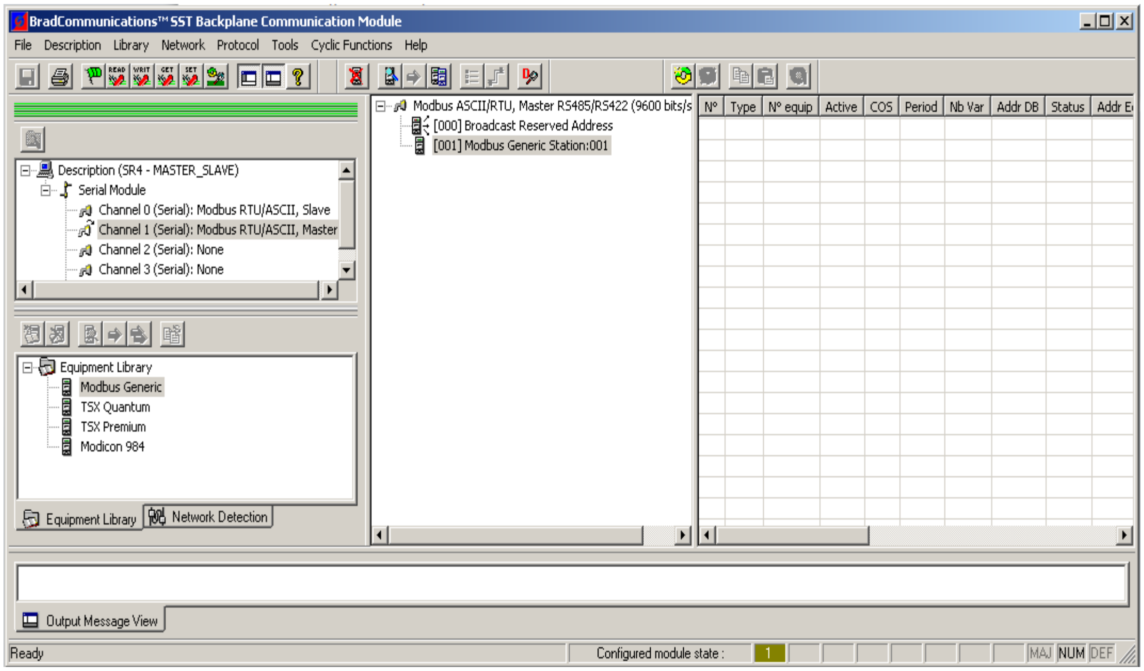
The procedures is using a single communication module for both client and server configuration.

1. Configure Channel 0 using procedures in Configure the Module as Modbus Serial Slave Equipment.

2. Configure Channel 1 using procedures in Configure the Module as Modbus Serial Master Equipment.

3. Make sure using the equipment number which is configured for the slave in Channel 0 when adding the slave equipment in the master equipment in Channel 1.

4. When procedures in Step 1 and Step 2 are complete, the console application main screen display similar to what is shown below.

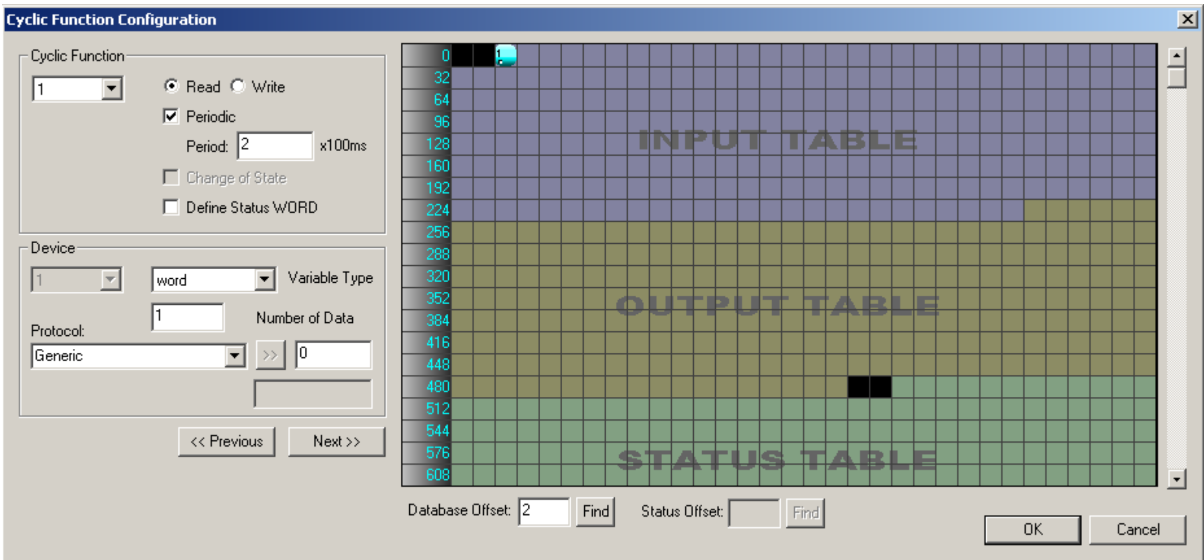


5. To save and download the configuration, click  and . The configuration download takes about 2-3 minutes.

6. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

7. Creating cyclic function by clicking . The cyclic function will provide automatic data exchange between the master (configured in channel 1) and slave (configured in channel 0) equipment. Creating some read and write cyclic functions that will allow automatic reading and writing from/to from the slave equipment.

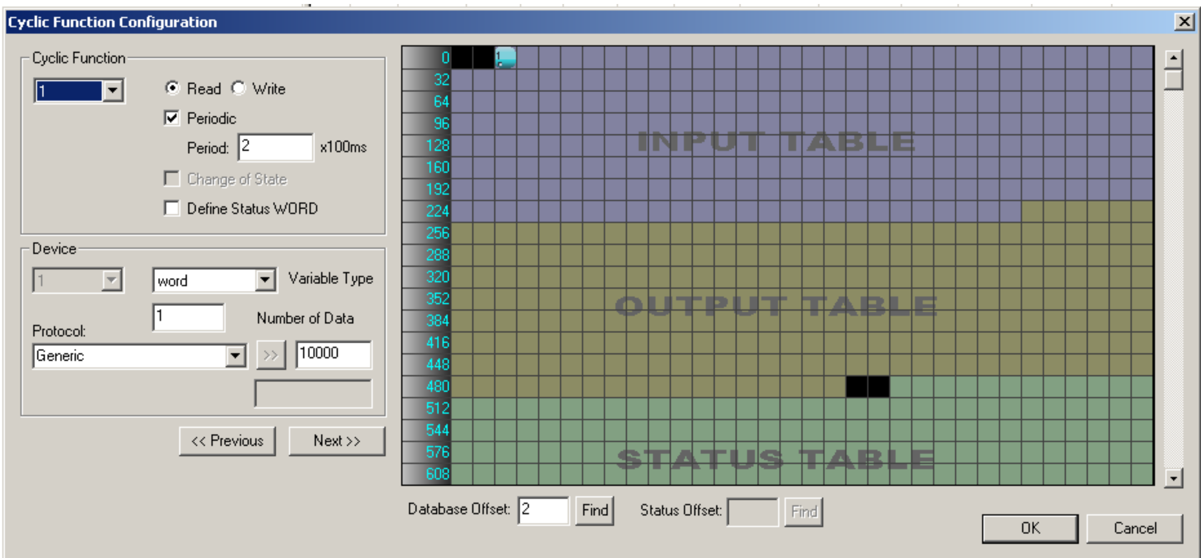
Figure 79: Cyclic Function Window



The detail of each parameter in discussed in Cyclic Function Configuration Parameters and Table 38.

i. Create cyclic function #1 with the parameter below.
   i. Use default cyclic function number provided (1 should be the default if this the first cyclic function)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Generic
   vii. Device First variable address is 10000 (slave equipment address)
   viii. Database offset is 2 (master equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below.



j. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

k. Create cyclic function #2 with the parameter below
   i. Use default cyclic function number provided (2 should be the default now)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Generic
   vii. Device First variable address is 11000 (slave equipment address)
   viii. Database offset is 1000 (master equipment address)

l. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.

The difference between the first and second cyclic function is the database offset location in the client.

1st cyclic function database address is located in the mapped IO table address 0-699

2nd cyclic function database address is located outside of the mapped IO table address

m. Create cyclic function #3 with the parameters below:
   i. Use default cyclic function number provided (3 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Generic
   vii. Device First variable address is 12000 (slave equipment address)

viii. Database offset is 250 (master equipment address)

n. Click "OK", and the third cyclic function will be shown in the cyclic function list view

o. Create cyclic function #4 with the parameter below

i. Use default cyclic function number provided (4 should be the default now)
ii. Select Write cyclic function type
iii. Check the Periodic setting
iv. Use default period of 2 x 10ms = 20ms
v. Number of data is 1
vi. Protocol is Generic
vii. Device First variable address is 13000 (slave equipment address)
viii. Database offset is 2000 (master equipment address)

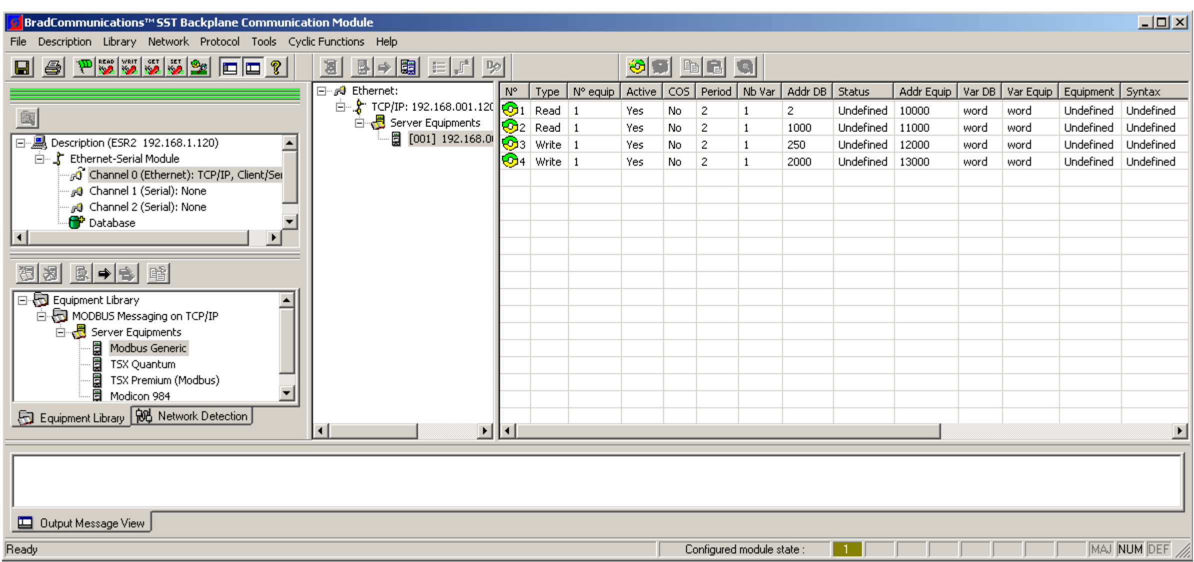
p. Click "OK", and the fourth cyclic function will be shown in the cyclic function list view

The difference between the first and second cyclic function is the database offset location in the client.

3rd cyclic function database address is located in the mapped IO table address 0-699

4th cyclic function database address is located outside of the mapped IO table address

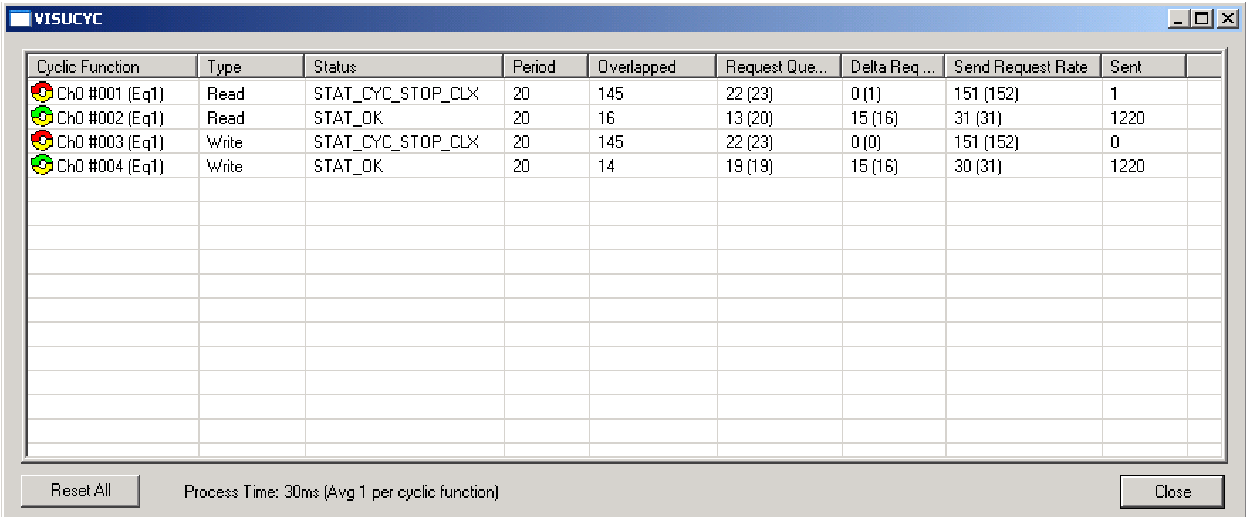
11. After creating all the cyclic function your screen should have a similar display as below.



12. To save and download the configuration to the module by clicking  and .

13. When download has completed, (Refer to Console Status Bar Information to check if the module is properly initialized), run the visual cyclic function application by clicking  from the main console screen.

The similar display is shown below.



Cyclic Function	Type	Status	Period	Overlapped	Request Que...	Delta Req ...	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_CYC_STOP_CLX	20	145	22 (23)	0 (1)	151 (152)	1
Ch0 #002 (Eq1)	Read	STAT_OK	20	16	13 (20)	15 (16)	31 (31)	1220
Ch0 #003 (Eq1)	Write	STAT_CYC_STOP_CLX	20	145	22 (23)	0 (0)	151 (152)	0
Ch0 #004 (Eq1)	Write	STAT_OK	20	14	19 (19)	15 (16)	30 (31)	1220

*Both 1st and 3rd cyclic function status is 135 – **not active / not running***

*Both 2nd and 4th cyclic function status is 0 – **active / running***

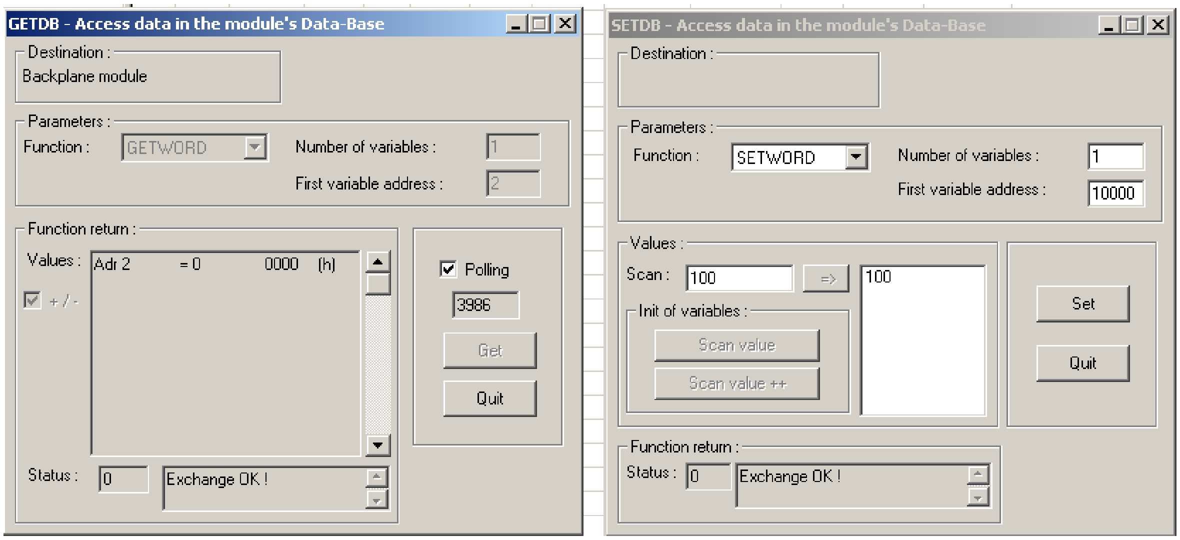
Note

The module will not run any cyclic function configured in the database address mapped to INPUT, OUTPUT and STATUS Table when there is no backplane connection.

14. To check if active cyclic function is running properly, follow this brief diagnostic below:

   f. Locate and run  and  application.

   g. To test cyclic function # 1, fill in the values as shown below.

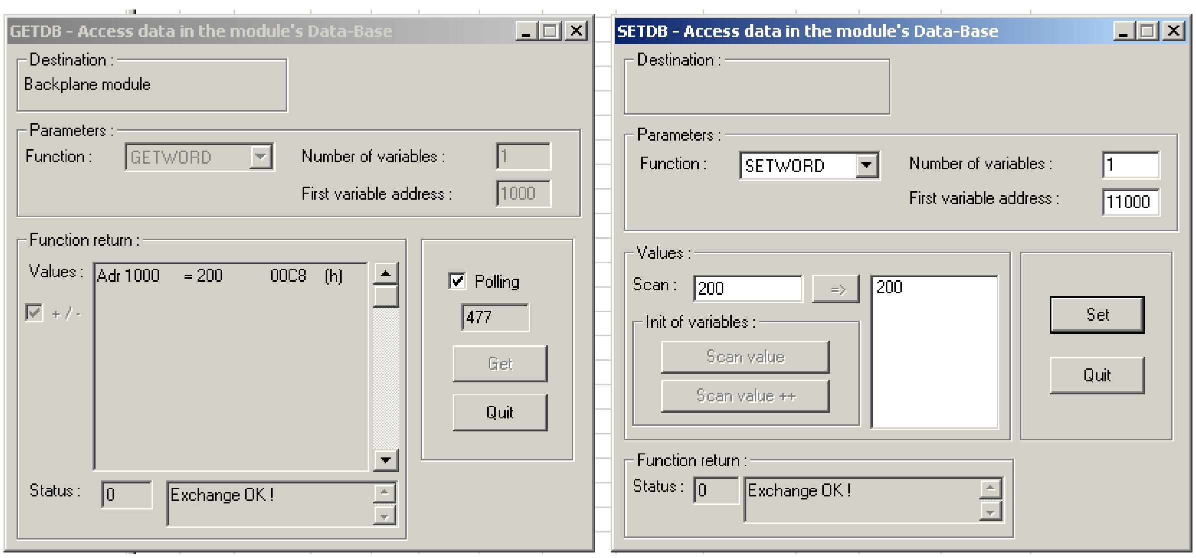


SetDB sets the server equipment address 10000 with value 100.

GetDB reads the client equipment database address 2 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 2.

h. To test cyclic function # 2, fill in values as shown below.

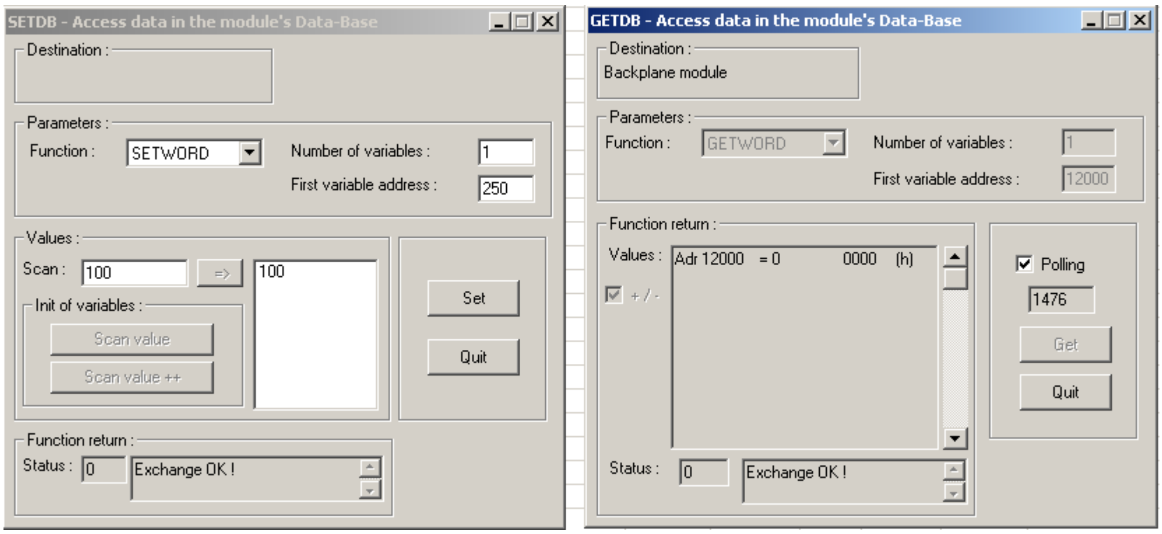


SetDB sets the server equipment address 11000 with value 200.

GetDB reads the client equipment database address 1000 with value shown as 200.

If the data is correct, it means the setup is working properly.

i. To test cyclic function # 3, fill in the values as show below.



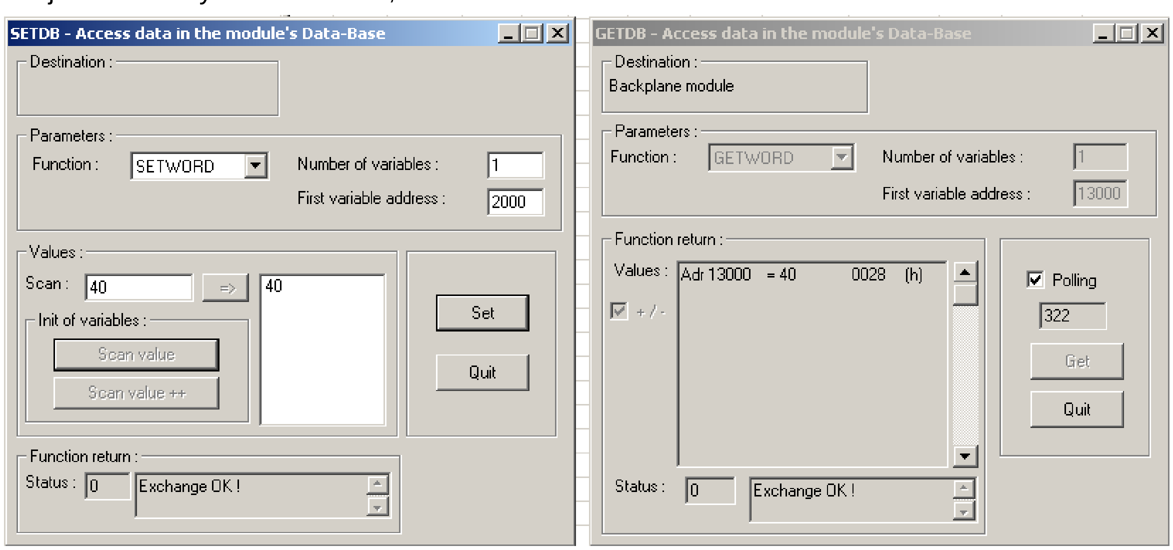
SetDB sets the client database address 250 with value 100.

GetDB reads the server equipment database address 12000 with value shown as 0.

If the data is correct, it means the setup is working properly.

As indicated in the cyclic function status from Visucyc application, cyclic function #3 is ***not active***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 250.

j. To test cyclic function #4, fill in the values as shown below.



SetDB sets the client equipment address 2000 with value 40.

GetDB reads the server equipment address 13000 with value shown as 40.

If the data is correct, it means the setup is working properly.

In the next set of procedures, the module will communicate with the ControlLogix CPU. When backplane connection becomes active, all cyclic function with **"not active / not running"** state will switch to "**active/running**".

With Backplane Connection:

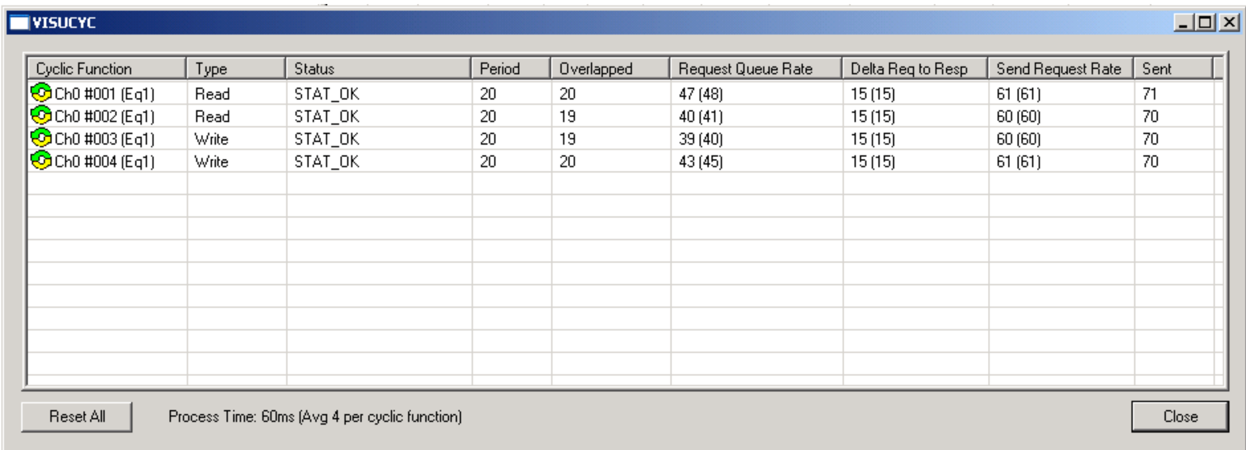
To be able to demonstrate successfully this sample configuration, it is assumed that the previous sets of procedures (from **No Backplane Connection**) were completed successfully.

1. Create a backplane connection by following procedures from Using 1756 Generic Profile section.

2. Make sure using the right Ethernet/IP address, the correct slot location of the module, and make it into "Run Mode". See below.

3. At this point, all the LEDs on the module are in Green color, and "COPN" alternately displayed with the module's IP address and Configuration name, configured in step #1.

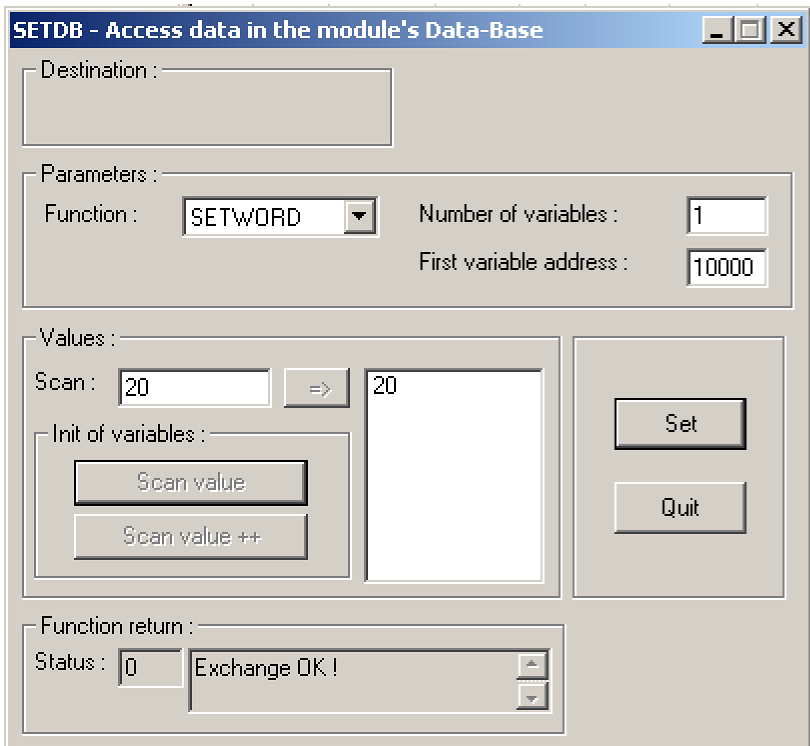
4. Launch the VisuCyc and all the cyclic functions are in active status. See below.



Cyclic Function	Type	Status	Period	Overlapped	Request Queue Rate	Delta Req to Resp	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_OK	20	20	47 (48)	15 (15)	61 (61)	71
Ch0 #002 (Eq1)	Read	STAT_OK	20	19	40 (41)	15 (15)	60 (60)	70
Ch0 #003 (Eq1)	Write	STAT_OK	20	19	39 (40)	15 (15)	60 (60)	70
Ch0 #004 (Eq1)	Write	STAT_OK	20	20	43 (45)	15 (15)	61 (61)	70

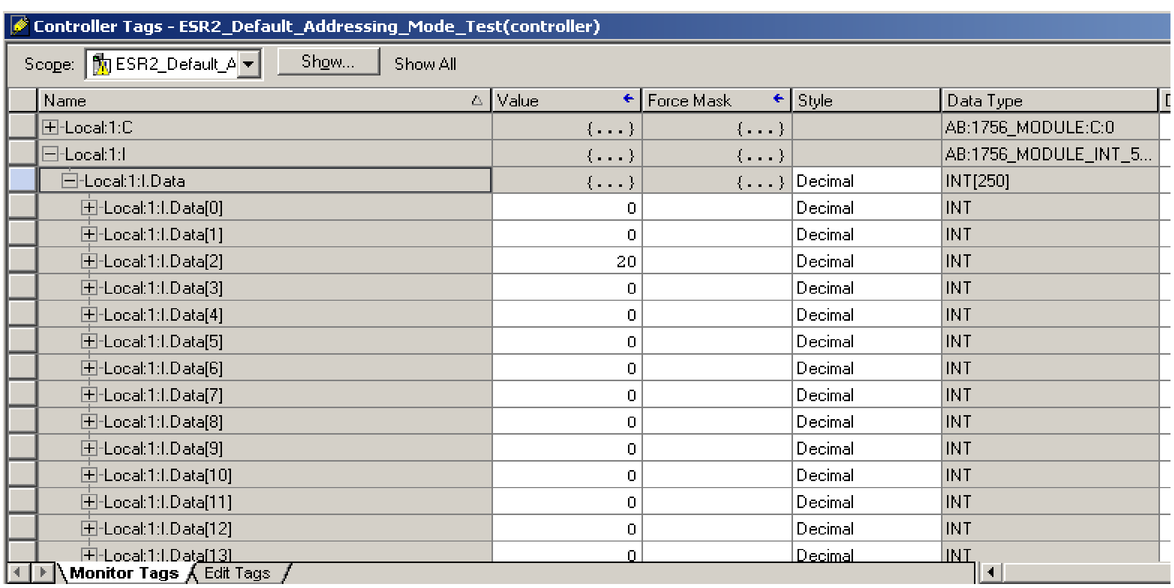
5. Using cyclic function 1 & 3 to check if values written and read from the mapped database address are exchanged properly.

6. To test cyclic function #1, fill in the values as shown below.



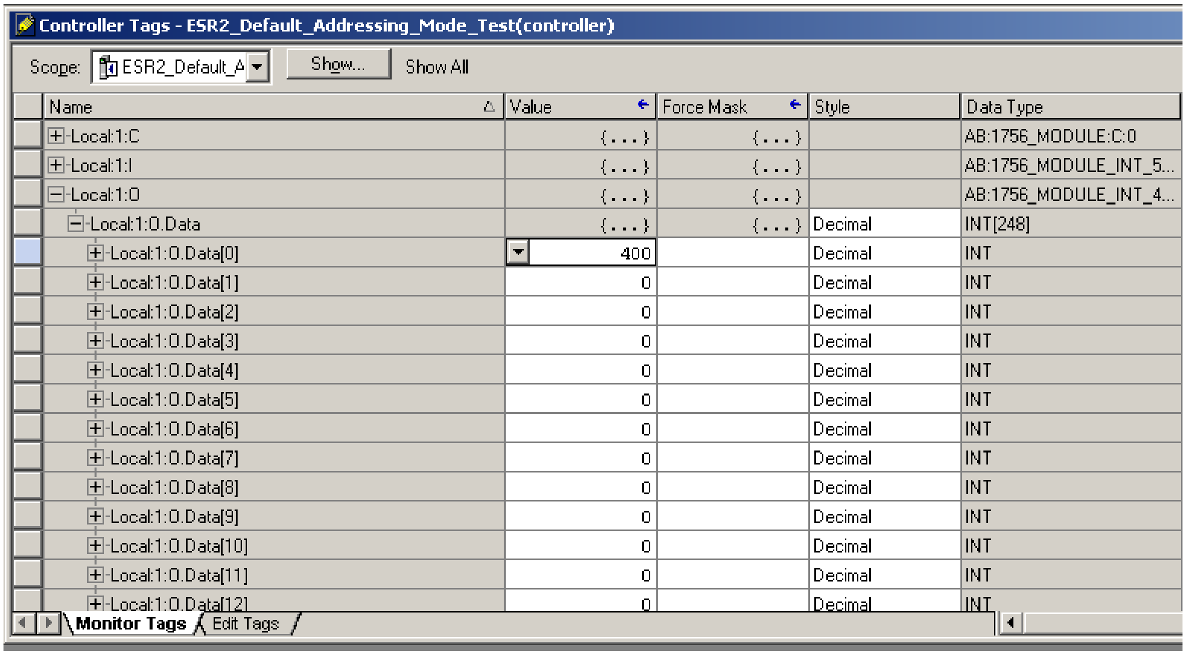
SetDB sets the server equipment address 10000 with value 20.

7. In RSLogix 5000 Software, expand the INPUT Table in the Controller Tags. the value 20 in offset 2. See below.

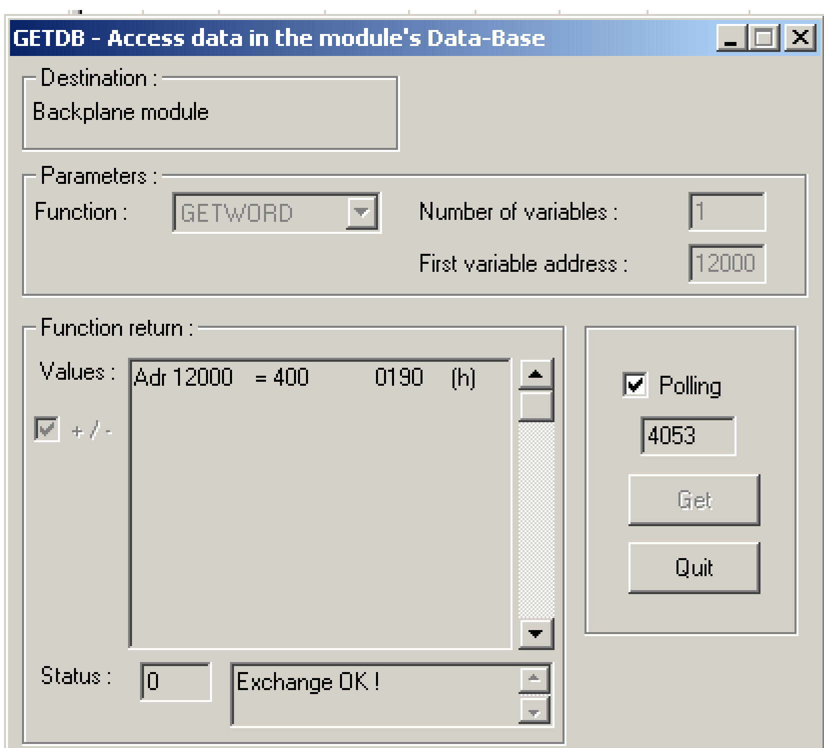


8. If the data is correct values, the backplane configuration is working properly.

9. To test cyclic function #3, expand Output Table and write 400 into offset 0 (offset 250 in database)



Run ***GetDB*** and read the values as offset 12000.



10. If the data is correct values, the backplane configuration is working properly.
11. Now start adding the test of the desired configuration.

6.5.2.2 Modbus Serial in Extended Addressing Mode

When the module is configured in extended addressing mode, the entire address defined in Input Table, Output Table and Status Table from the Console Database Configuration are accessible from the backplane using our AOI (Add-On-Instruction).

In this configuration, map 1600 words of INPUT, 1600 words of OUTPUT and 1024 words of STATUS of the module's database to the backplane. The database configuration needs to use extended addressing mode to be able to map the database larger than the maximum INPUT and OUTPUT table connection size allowed in the ControlNet.

No Backplane Connection:

To be able to demonstrate successfully this sample configuration, Modbus Serial Master Equipment and Modbus Serial Slave equipment are used and they must have an active connection to be able to show data exchanges configured through cyclic function.

The SST-SR4-CLX-RLL product is used in this sample configuration. If there are two available modules of this product variant, configure one as Modbus Serial Master and the other one as Modbus Serial Slave. If there is only one module, use a single communication module configured both as Modbus Serial Master and Modbus Serial Slave. Cyclic functions can only be created when module is running a master equipment configuration.

Note

Since client and server equipment is sharing the same database, make sure different addresses are used for the source and destination address of the data.

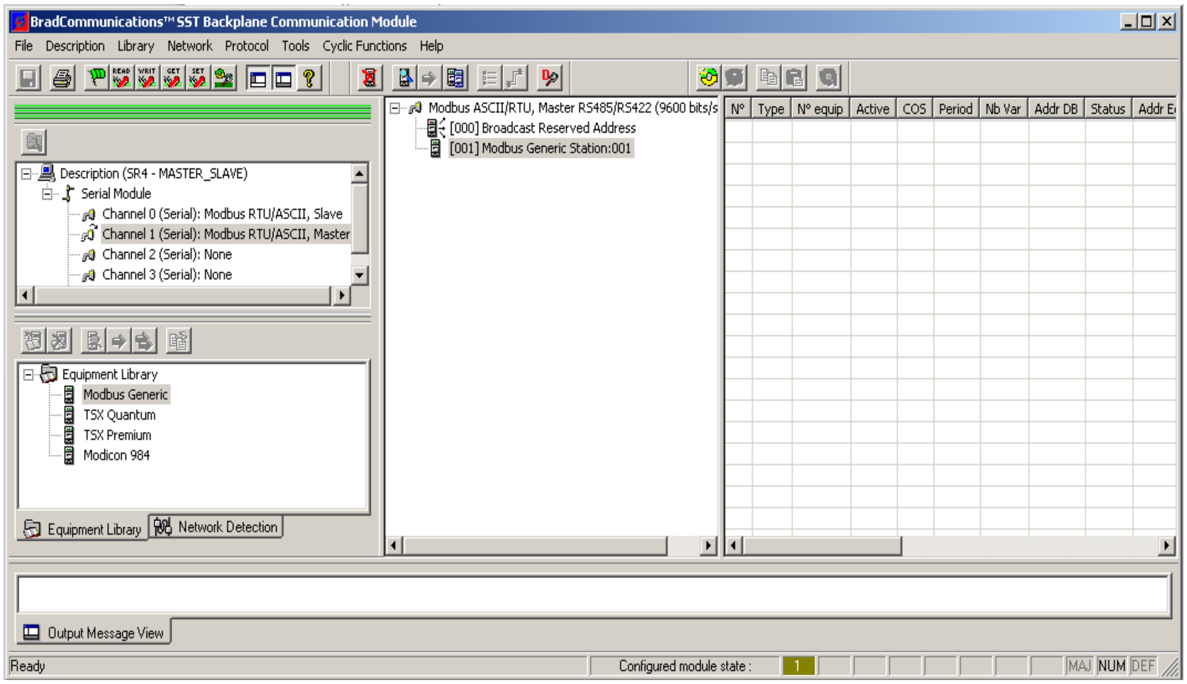
The firmware version should be 2.10.2 or higher to be able to use this feature. The procedures below assumed a single communication module is used for both client and server configuration.

1. Configure Channel 0 using procedures in Configure the Module as Modbus Serial Slave Equipment.

2. Configure Channel 1 using procedures in Configure the Module as Modbus Serial Master Equipment.

3. Make sure using the equipment number that is configured for the slave in Channel 0 when adding the slave equipment in the master equipment in Channel 1.

4. When procedures in Step 1 and Step 2 are complete, the console application main screen with display similar to what is shown below.



5. Complete the instruction in Changing to Extended Addressing Mode section.

6. To save and download the configuration, click  and . The configuration download takes about 2-3 minutes.

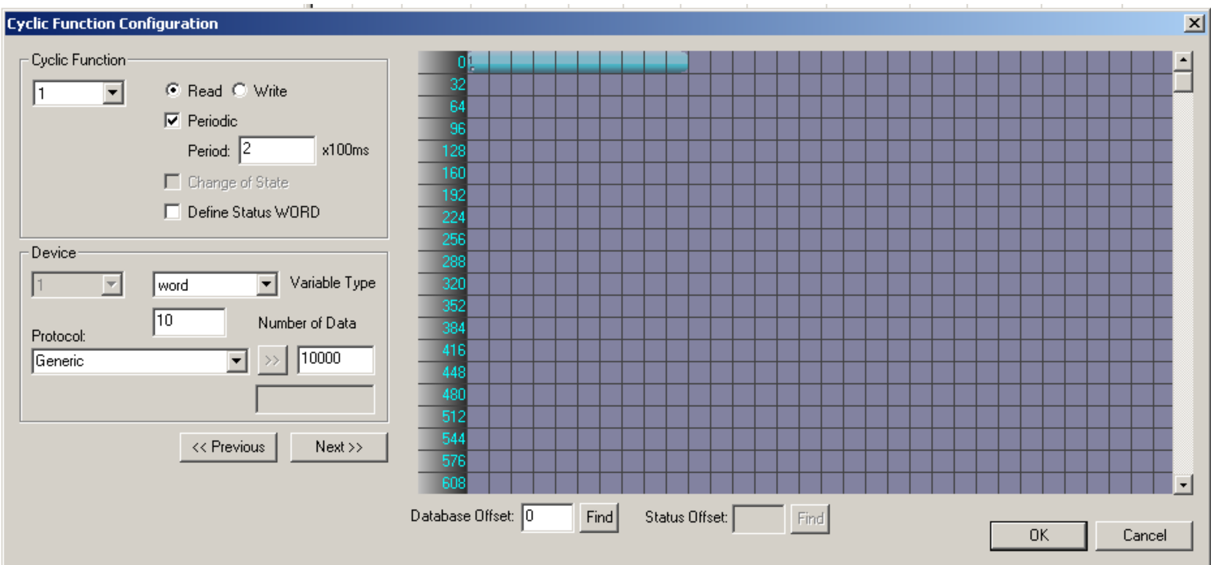
7. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

8. Start creating cyclic function by clicking . The cyclic function created will provide automatic data exchange between the master (configured in channel 1) and slave (configured in channel 0) equipment. To create read and write cyclic functions that will allow automatic reading and writing from/to from the slave equipment.

   The detail of each parameter in discussed in Cyclic Function Configuration Parameters and Table 38.

   a. Create cyclic function #1 with the parameter below.
      i. Use default cyclic function number (1 should be the default if this the first cyclic function)
      ii. Select Read cyclic function type
      iii. Check the Periodic setting
      iv. Use default period of 2 x 10ms = 20ms
      v. Number of data is 10
      vi. Protocol is Generic
      vii. Device First variable address is 10000 (slave equipment address)
      viii. Database offset is 0 (master equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below.



b. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

c. Create cyclic function #2 with the parameter below

   i. Use default cyclic function number (2 should be the default now)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 10
   vi. Protocol is Generic
   vii. Device First variable address is 11000 (slave equipment address)
   viii. Database offset is 5000 (master equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.

The difference between the first and second cyclic function is the database offset location in the client.

1st cyclic function database address is located in the mapped Input table address 0-1600

2nd cyclic function database address is located outside of Input and Output mapped table address

e. Create cyclic function #3 with the parameters below:

   i. Use default cyclic function number (3 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 10

vi. Protocol is Generic
vii. Device First variable address is 12000 (slave equipment address)
viii. Database offset is 1600 (master equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
g. Create cyclic function #4 with the parameter below
   i. Use default cyclic function number (4 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 10
   vi. Protocol is Generic
   vii. Device First variable address is 13000 (slave equipment address)
   viii. Database offset is 6000 (master equipment address)

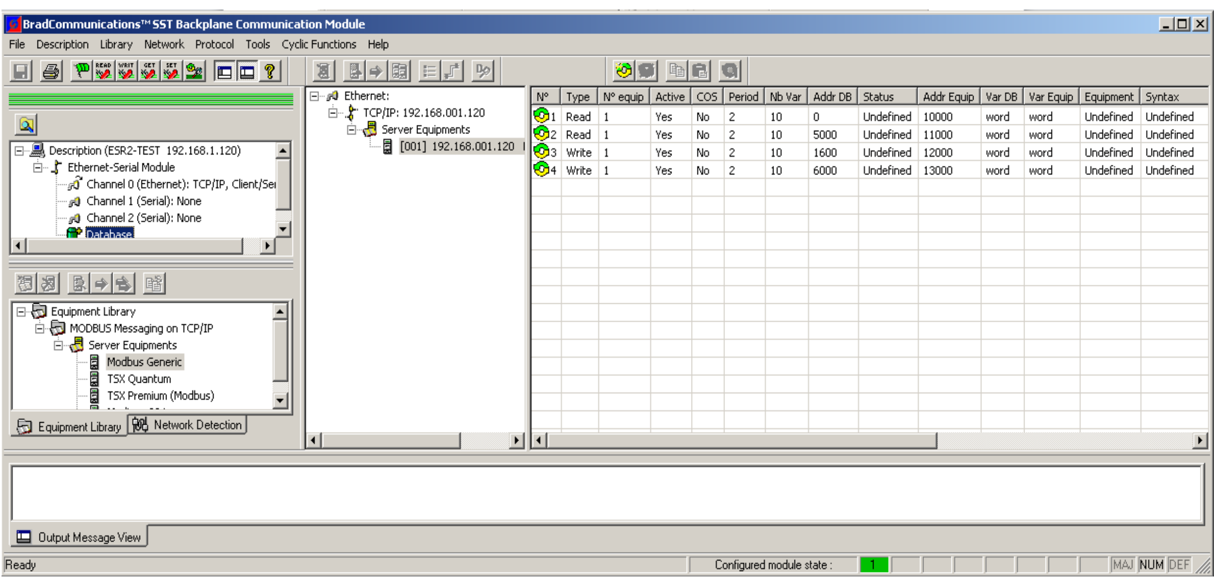
h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

   The difference between the first and second cyclic function is the database offset location in the client.

   3rd cyclic function database address is located in the mapped Output table address 1600-3199

   4th cyclic function database address is located outside of the mapped Input and Output table address

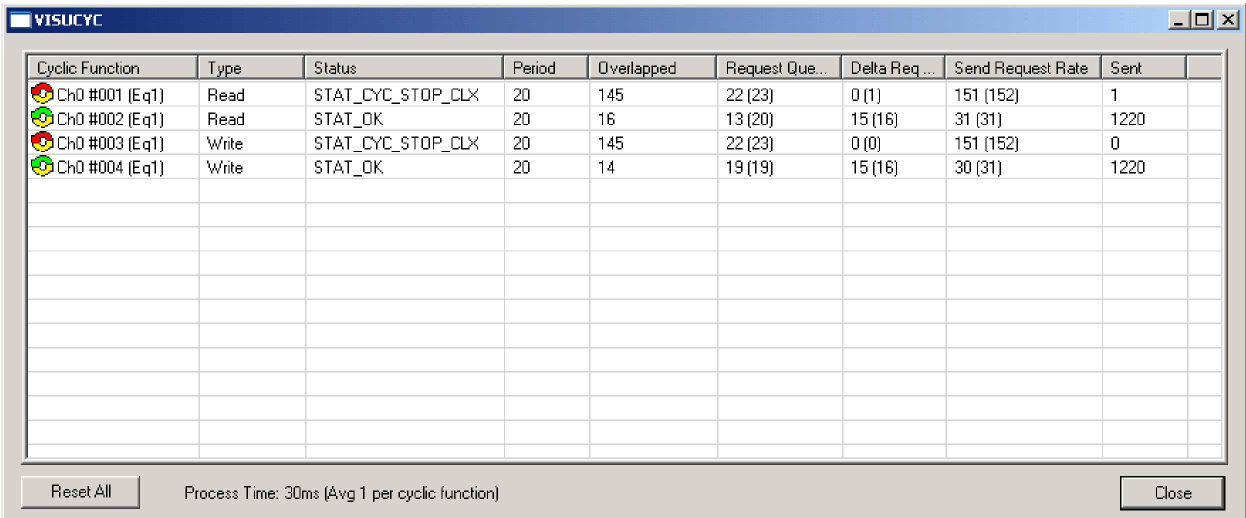
9. After creating all the cyclic function you console window should have similar display as below.



10. To save configuration and download configuration, click  and . The configuration download takes about 2-3 minutes.

11. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

12. If initialization is successful, run the Visucyc application to check the status of each cyclic function. It displays as below.



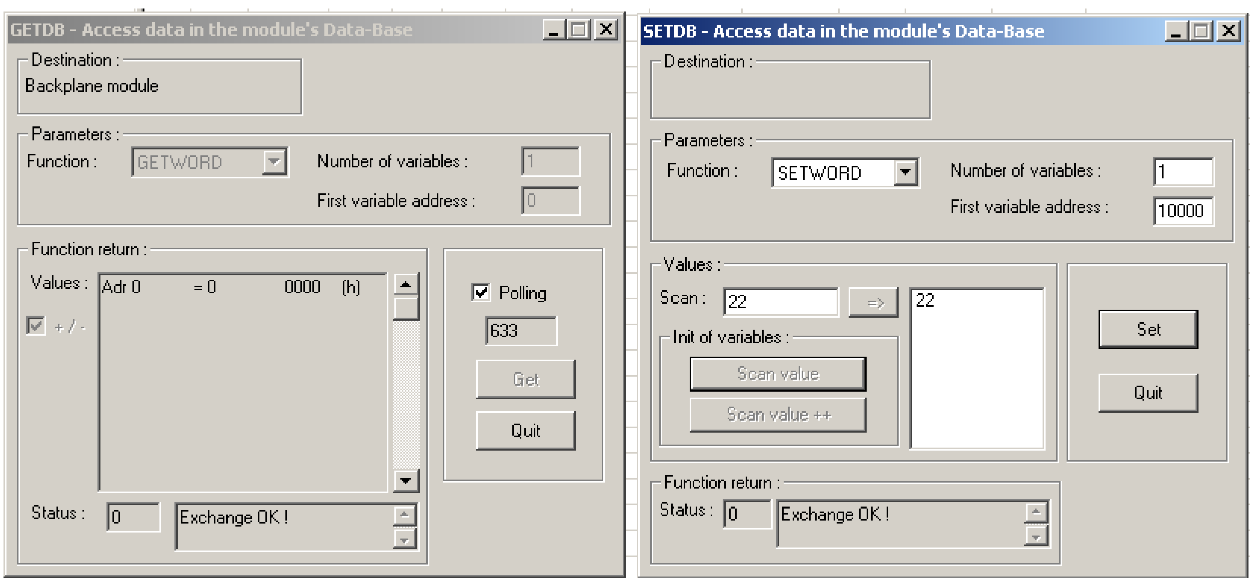
*Both 1st and 3rd cyclic function status is 135 – n**ot active /not running***

*Both 2nd and 4th cyclic function status is 0 – **active / running***

12. To check if active cyclic function is running properly, follow this brief diagnostic below.

f. Locate and run  and  application.

g. To test cyclic function # 1, fill in the values as shown below.

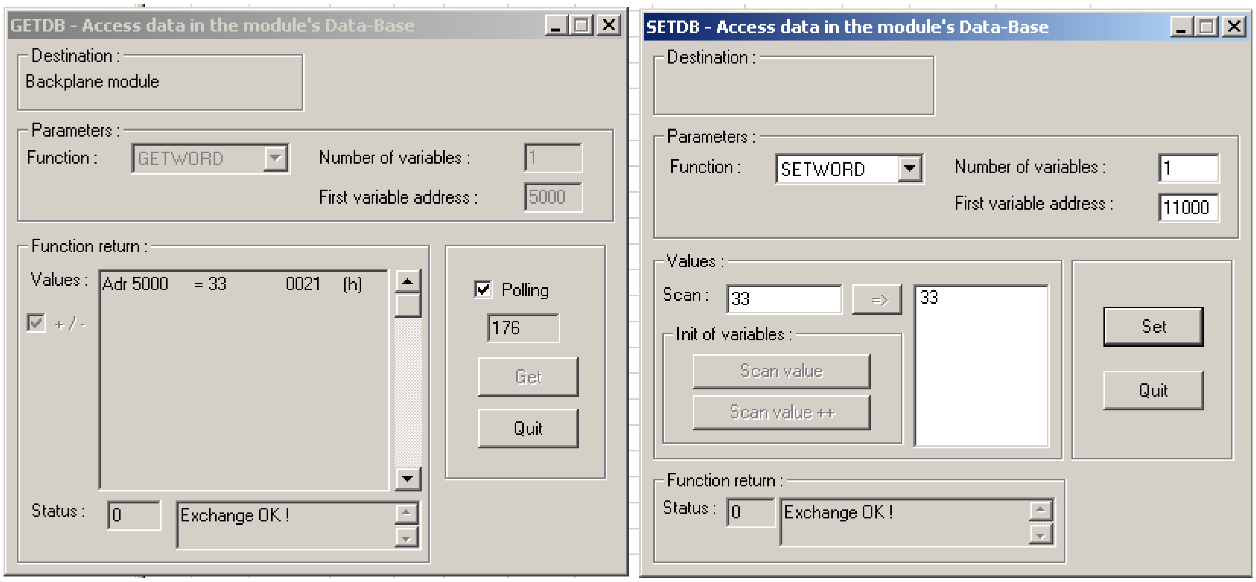


SetDB sets the server equipment address 10000 with value 22.

GetDB reads the client equipment database address 0 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 0.

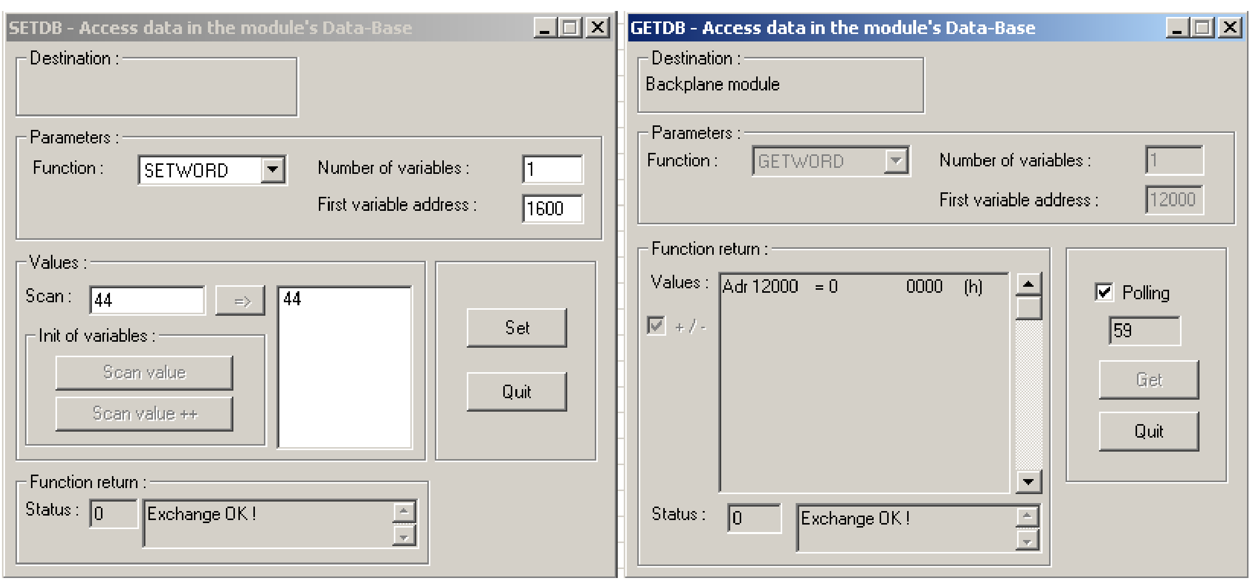
h. To test cyclic function # 2, fill in values as shown below.



SetDB sets the server equipment address 11000 with value 33.

GetDB reads the client equipment database address 5000 with value shown as 33.

i. To test cyclic function # 3, fill in the values as shown below.

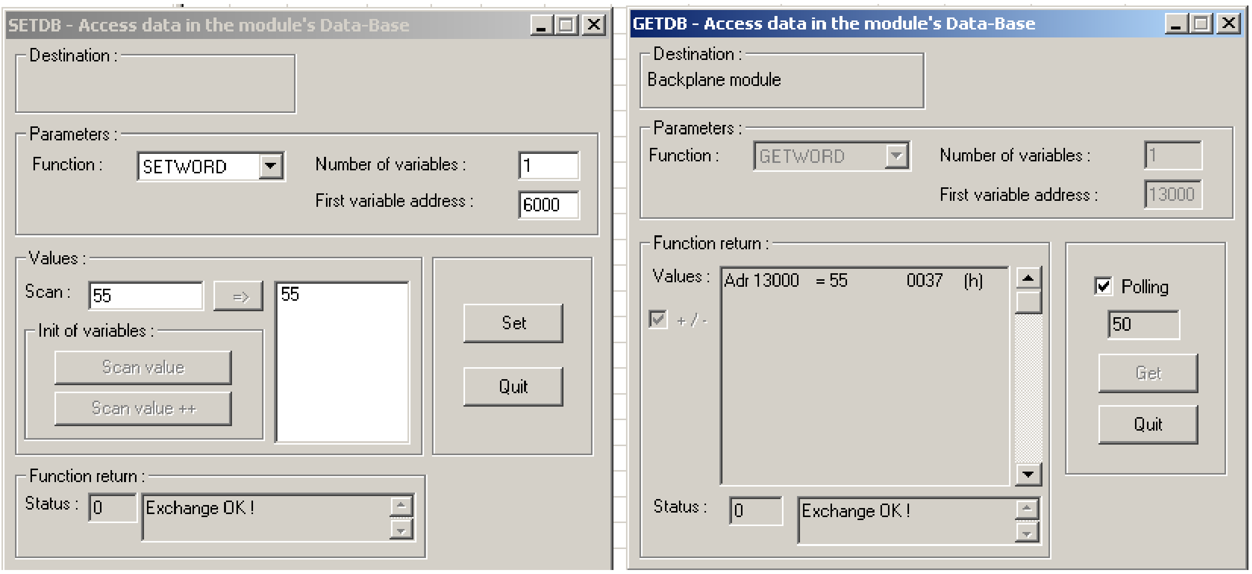


SetDB sets the client database address 1600 with value 44.

GetDB reads the server equipment database address 12000 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 12000.

j. To test cyclic function #4, fill in the values shown below.



SetDB sets the client database address 6000 with value 55.

GetDB reads the server equipment database address 13000 with value shown as 55.

In the next set of procedures, the module will communicate with the ControlLogix CPU. When backplane connection becomes active, all cyclic function with **"not active / not running"** state will switch to "**active/running**".

With Backplane Connection:

The following provide multiple sample configurations with different parameters.

Example 1: Transfer 1 word of data for each write and read cyclic function using the default mapping defined in the AOI.

1. If the previous set of procedures in **No Backplane Connection** is skipped, please go back and configure the same cyclic functions 1-4 and run all diagnostic successfully.

2. Follow instruction 1-19 in Using 1756 Generic Profile section.

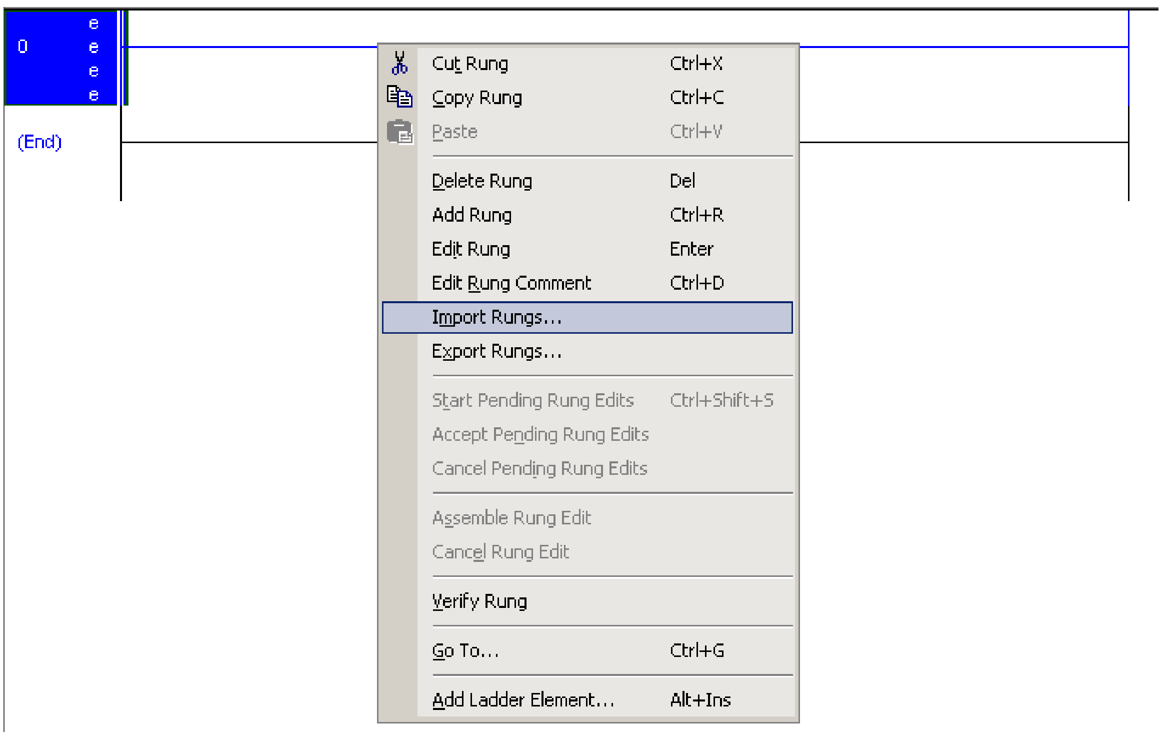
Note

For first time user of this feature, the console configuration should be done first prior to configuring ControlLogix connection because of the mapping of the database.

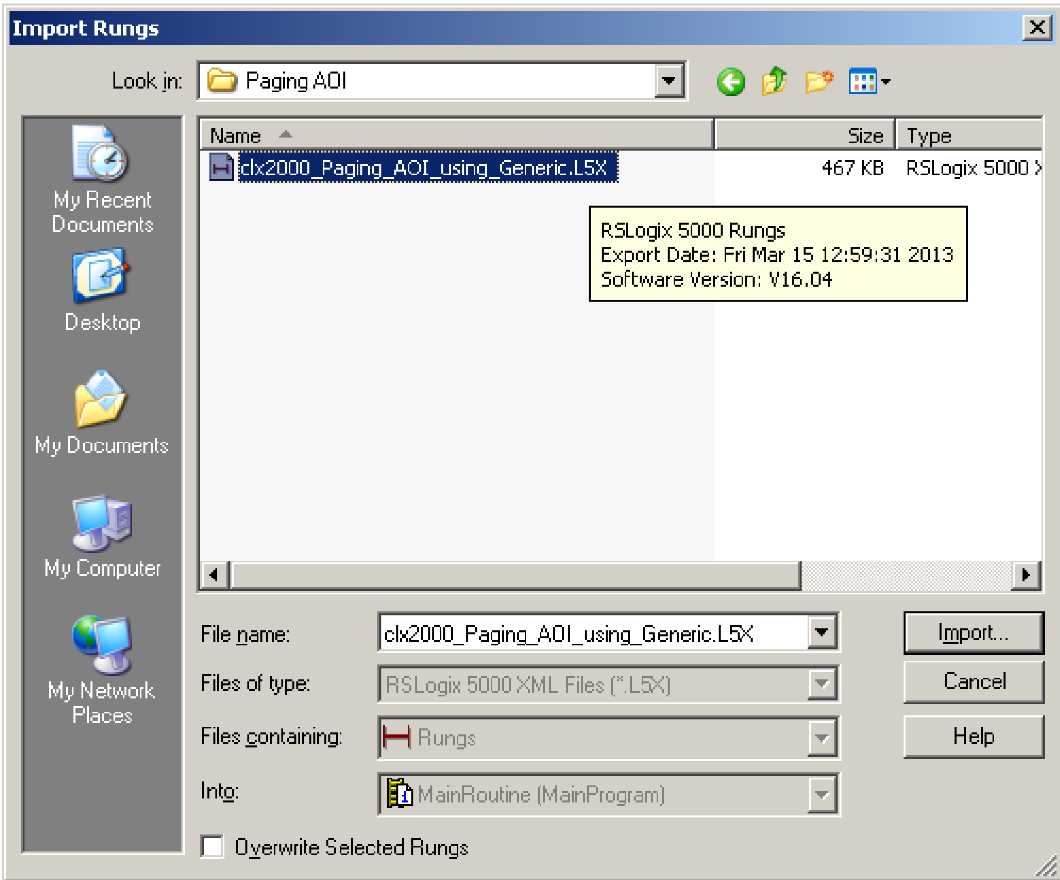
3. If smaller input and output sizes are required, please refer to section Changing the I/O connection size.

4. In RSLogix 5000 Task->MainTask, double click on MainRoutine. See below.

5. Select a rung as below and right-click on rung and select “Import Rung…”

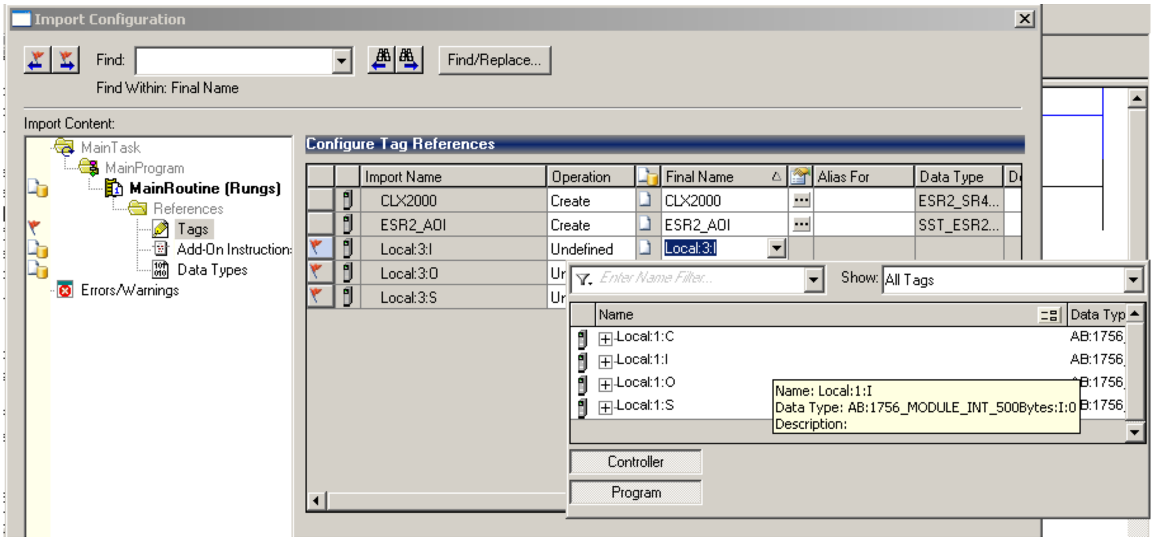


6. Browse for the ….L5X file as below and select it and select Import.... Refer to Module's Installation Directory Location section for the location of the AOI files.

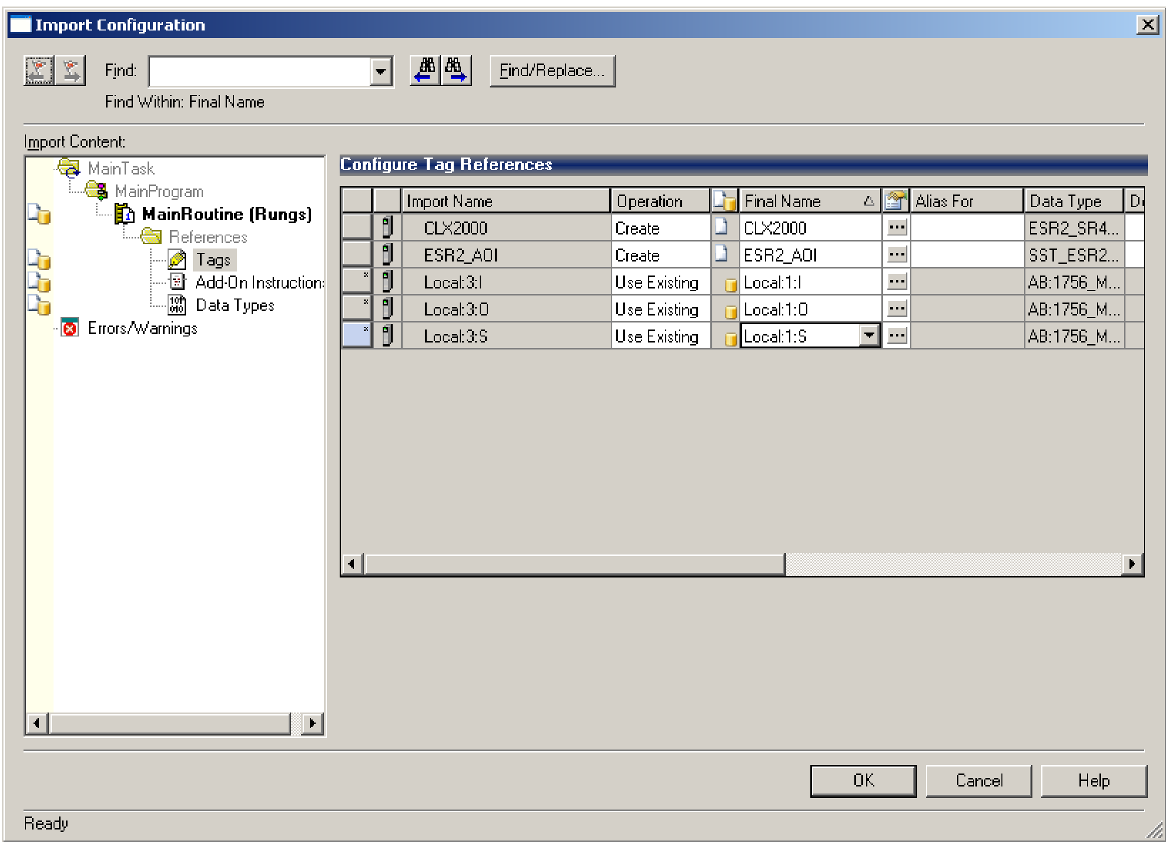


7. The import Configuration dialog box will appear as shown in below. For the tags that have X beside it, select the Input and Output tags for the module added in the I/O configuration.

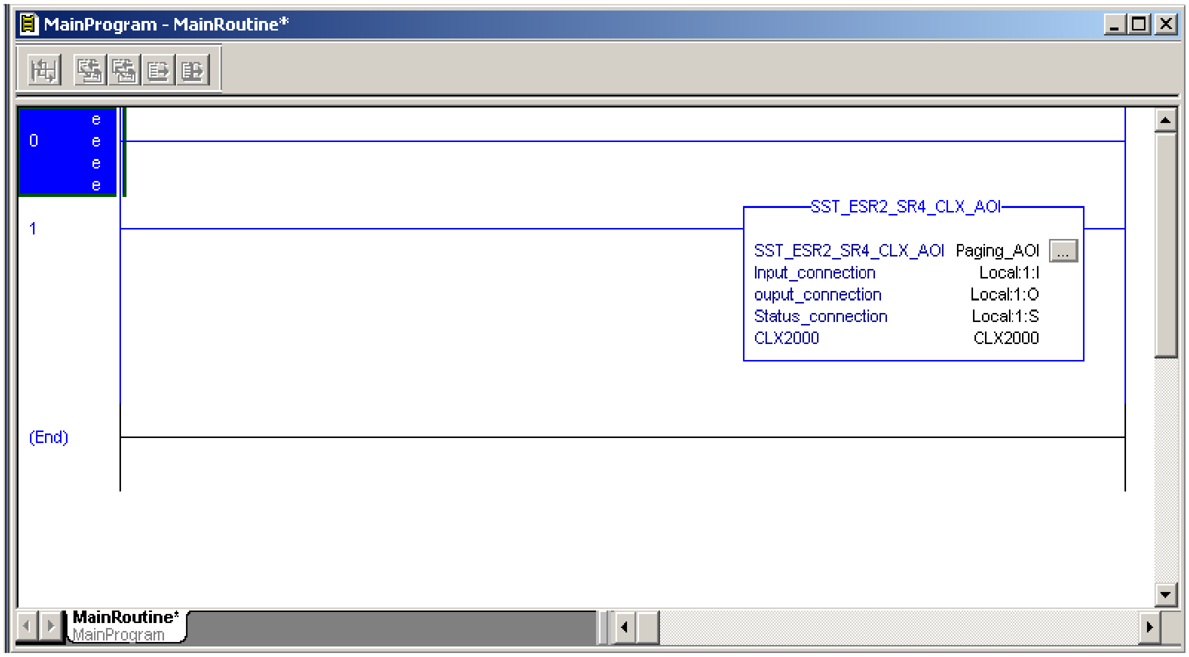
8. In this example the module was configured at slot 1 but the AOI was expecting slot 3. Update Tags Local:3:I, Local:3:O,Local:3:S to reference slot 1.



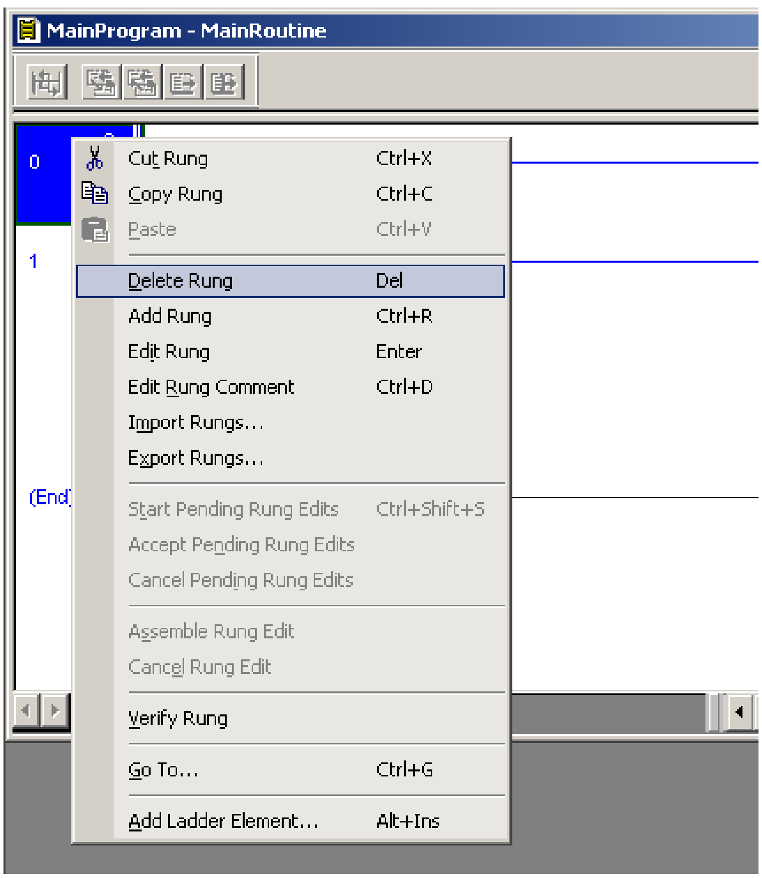
9. After correcting the last three tags click ok to begin the import.



10. After the import is complete, the display shown below.



11. Delete the empty rung as shown below.



12. Under Controller Tags, go to the CLX2000 tag. Expand CLX2000.CONFIGURATION as below.

    INPUT Table Start Address = 0

    INPUT Table size = 1600

    OUTPUT Table Start Address = 1600

    OUTPUT Table Size = 1600

    STATUS Table Start Address = 3200

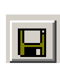
    STATUS Table Size = 1024

    The default mapped address is used in the AOI in this configuration.

Figure 80: AOI Controller Tags

Name	Value	Force Mask	Style	Data Type
⊟-CLX2000	{...}	{...}		ESR2_SR4_MODULE
⊟-CLX2000.CONFIGURATION	{...}	{...}		DATABASE_CONFIG
⊞-CLX2000.CONFIGURATION.Input_Table_Start_Ad...	0		Decimal	INT
⊞-CLX2000.CONFIGURATION.Input_Table_Size	1600		Decimal	INT
⊞-CLX2000.CONFIGURATION.Output_Table_Start_A...	1600		Decimal	INT
⊞-CLX2000.CONFIGURATION.Output_Table_Size	1600		Decimal	INT
⊞-CLX2000.CONFIGURATION.Status_Table_Start_A...	3200		Decimal	INT
⊞-CLX2000.CONFIGURATION.Status_Table_Size	1024		Decimal	INT
⊟-CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
⊞-CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊞-CLX2000.DATABASE_DATA.STATUS	{...}	{...}	Decimal	INT[1024]
⊟-CLX2000.NUMBER_OF_BLKS	{...}	{...}		Pages_Control
⊞-CLX2000.NUMBER_OF_BLKS.Number_Input_Blocks	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Number_Output_Blo...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Number_Status_Blo...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Input_Remainder	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Output_Remainder	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Status_Remainder	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Input_Connection_S...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Output_Connection_...	0		Decimal	INT
⊞-CLX2000.NUMBER_OF_BLKS.Status_Connection...	0		Decimal	INT

13. INPUT Table address and size, OUTPUT Table address and size and STATUS Table address and size should match with the database settings for INPUT, OUTPUT and STATUS in console configuration. To check that it match, in Console Configuration Description Area, locate and double click . Change to the values shown above if it's different. (Refer to Changing to Extended Addressing Mode) for more details.

14. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

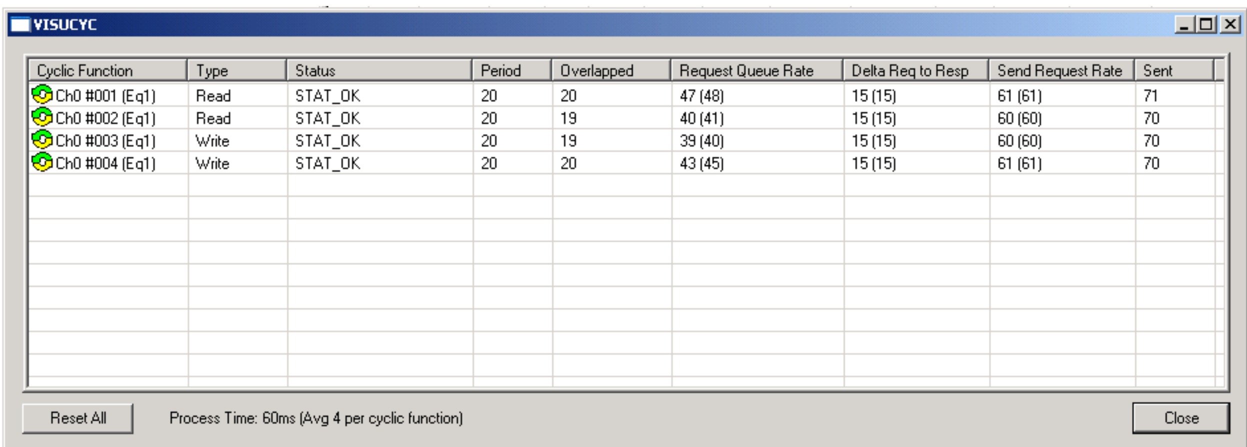
Note

The AOI instruction will not run if there is a mismatch between the CONFIGURATION tags and the database settings in console configuration.

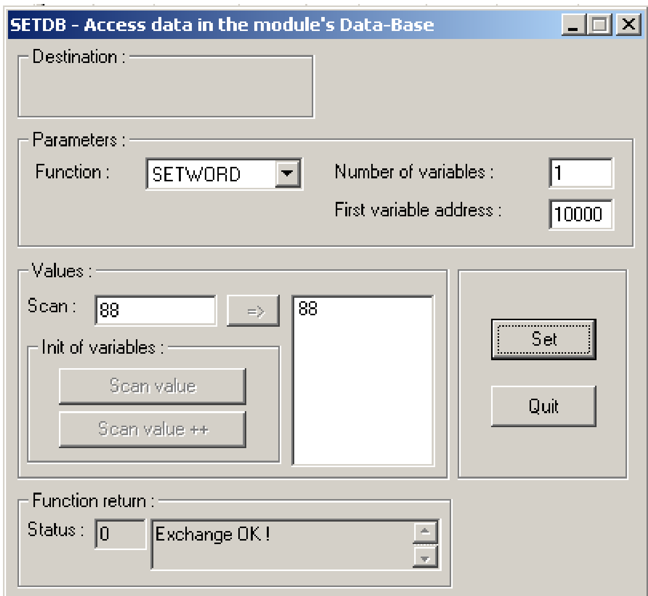
15. To see the input, output and status tags, expand the CLX2000.DATABASE_DATA. These are the tags that AOI will use.

16. Save the configuration file and download it to the CLX Module.

17. From the Console Application window, locate and launch VisuCyc, all the cyclic functions are in active status. See below.

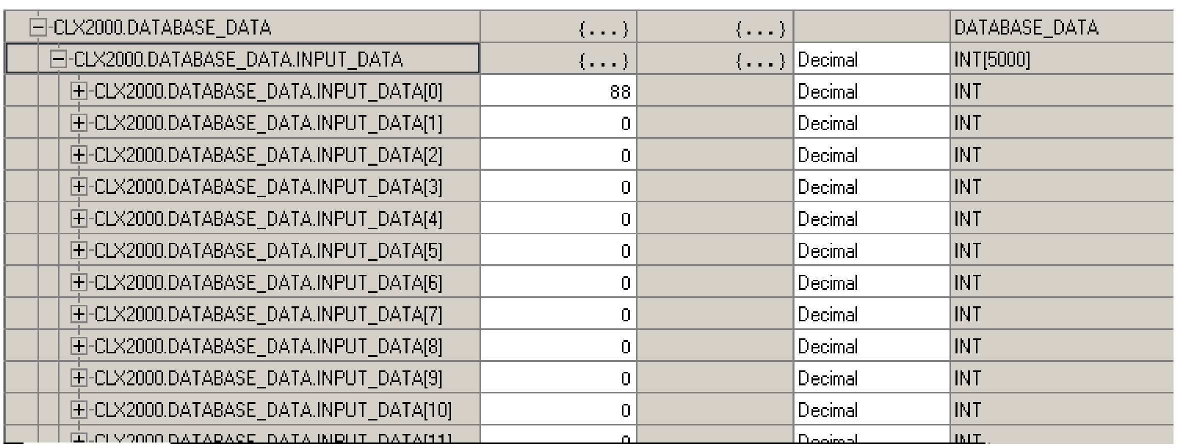


18. Test cyclic function 1 & 3 to check if values written and read from the mapped database address are exchanged properly. To test cyclic function #1, fill in the values as shown below.

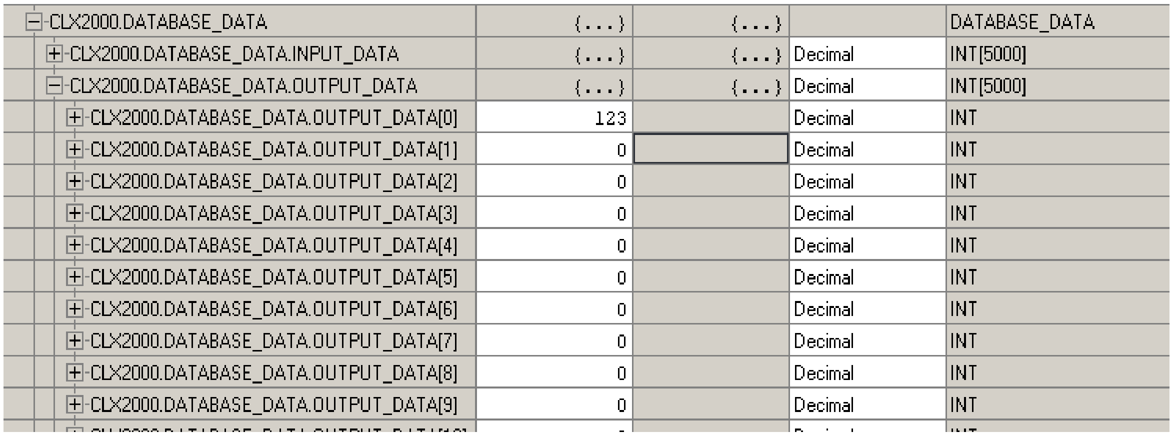


SetDB sets the server equipment address 10000 with value 88.

19. In RSLogix 5000 Software, expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value that is set in the server equipment address 10000 using SetDB should be displayed. See below.

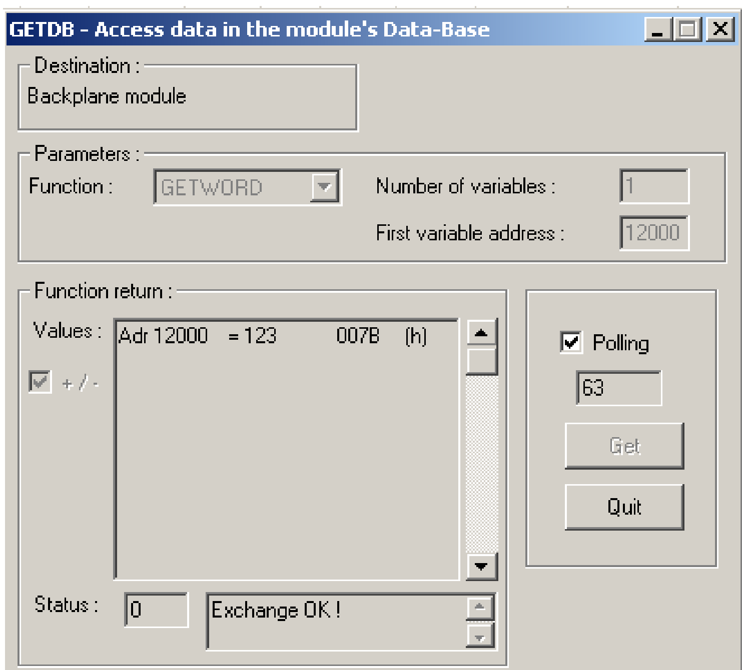


20. To test cyclic function #3, expand CLX2000.DATABASE_DATA.OUTPUT_DATA and write 123 into offset 0 (offset 5000 in database)



CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.OUTPUT_DATA[0]	123		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[2]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[3]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[4]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[5]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[6]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[7]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[8]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[9]	0		Decimal	INT

Run ***GetDB*** and read the values as offset 12000.



21. Now start adding the test of the desired configuration.

Example 2: Transfer 100 words of data per cyclic function. Configuring 5 write cyclic function and 5 read cyclic function using the default mapping defined in the AOI.

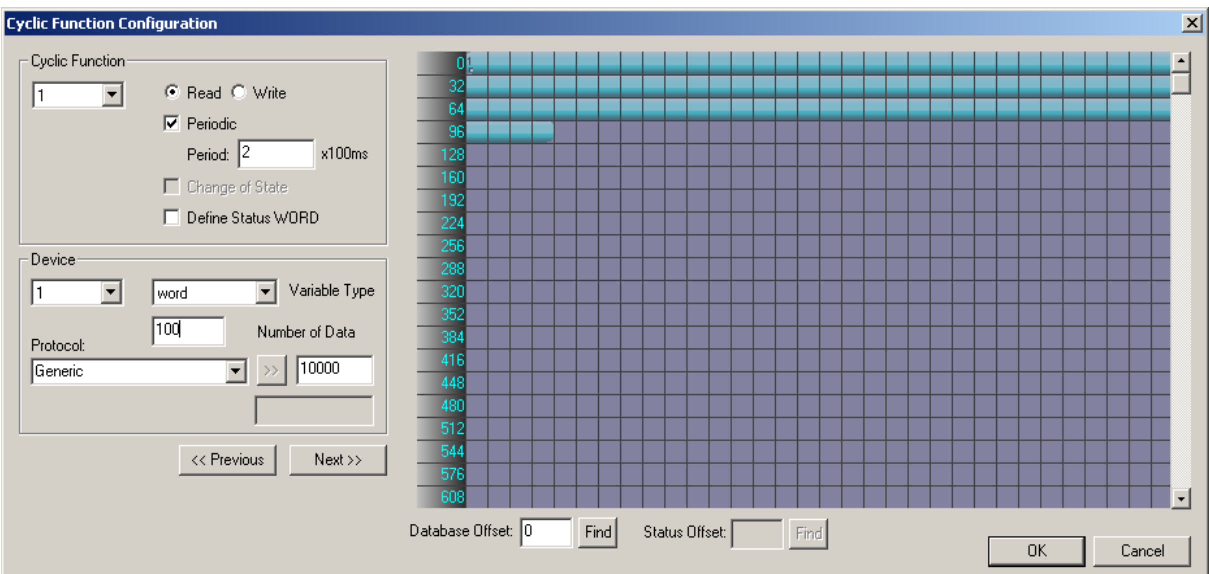
1. It is assume that Example 2 is a continuation of Example 1 therefore all console configurations, backplane connection and the quick test are done successfully.

2. From the console application window:

    i. Modify cyclic function #1 with the parameter below:

        i. default cyclic function number provided
        ii. Select Read cyclic function type
        iii. Check the Periodic setting
        iv. Use default period of 2 x 10ms = 20ms
        v. Number of data is 100
        vi. Protocol is Generic
        vii. Device First variable address is 10000 (slave equipment address)
        viii. Database offset is 0 (master equipment address)

    After filling up the parameter the cyclic function will be similar to what is shown below.



    j. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

    k. Modify cyclic function #2 with the parameter below:

        ix. Use default cyclic function number (2 should be the default now)
        x. Select Read cyclic function type
        xi. Check the Periodic setting
        xii. Use default period of 2 x 10ms = 20ms
        xiii. Number of data is 100
        xiv. Protocol is Generic

xv. Device First variable address is 11000 (slave equipment address)
xvi. Database offset is 100 (master equipment address)

l. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.
m. Modify cyclic function #3 with the parameters below:
   ix. Use default cyclic function number (3 should be the default now)
   x. Select Write cyclic function type
   xi. Check the Periodic setting
   xii. Use default period of 2 x 10ms = 20ms
   xiii. Number of data is 100
   xiv. Protocol is Generic
   xv. Device First variable address is 12000 (slave equipment address)
   xvi. Database offset is 1600 (master equipment address)

n. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
o. Modify cyclic function #4 with the parameter below
   ix. Use default cyclic function number (4 should be the default now)
   x. Select Write cyclic function type
   xi. Check the Periodic setting
   xii. Use default period of 2 x 10ms = 20ms
   xiii. Number of data is 100
   xiv. Protocol is Generic
   xv. Device First variable address is 13000 (slave equipment address)
   xvi. Database offset is 1700 (master equipment address)

p. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

3. Add 6 more cyclic functions to this configuration.
   l. Create cyclic function #5 with the parameter below.
      ix. default cyclic function number
      x. Select Read cyclic function type
      xi. Check the Periodic setting
      xii. Use default period of 2 x 10ms = 20ms
      xiii. Number of data is 100
      xiv. Protocol is Generic
      xv. Device First variable address is 14000 (slave equipment address)
      xvi. Database offset is 1000 (master equipment address)

   m. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

n. Create cyclic function #6 with the parameter below
    ix. Use default cyclic function number
    x. Select Read cyclic function type
    xi. Check the Periodic setting
    xii. Use default period of 2 x 10ms = 20ms
    xiii. Number of data is 100
    xiv. Protocol is Generic
    xv. Device First variable address is 15000 (slave equipment address)
    xvi. Database offset is 1100 (master equipment address)

o. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.
p. Create cyclic function #7 with the parameters below:
    ix. Use default cyclic function number
    x. Select Read cyclic function type
    xi. Check the Periodic setting
    xii. Use default period of 2 x 10ms = 20ms
    xiii. Number of data is 100
    xiv. Protocol is Generic
    xv. Device First variable address is 16000 (slave equipment address)
    xvi. Database offset is 1200 (master equipment address)

q. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
r. Create cyclic function #8 with the parameter below
    x. Use default cyclic function number
    xi. Select Write cyclic function type
    xii. Check the Periodic setting
    xiii. Use default period of 2 x 10ms = 20ms
    xiv. Number of data is 100
    xv. Protocol is Generic
    xvi. Device First variable address is 17000 (slave equipment address)
    xvii. Database offset is 2500 (master equipment address)

s. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
t. Create cyclic function #9 with the parameter below
    viii. Use default cyclic function number
    ix. Select Write cyclic function type
    x. Check the Periodic setting
    xi. Use default period of 2 x 10ms = 20ms
    xii. Number of data is 100
    xiii. Protocol is Generic

xviii. Device First variable address is 18000 (slave equipment address)

xiv. Database offset is 2600 (master equipment address)

u. Click "OK", and the third cyclic function will be shown in the cyclic function list view

v. Create cyclic function #10 with the parameter below

ix. Use default cyclic function number

x. Select Write cyclic function type

xi. Check the Periodic setting

xii. Use default period of 2 x 10ms = 20ms

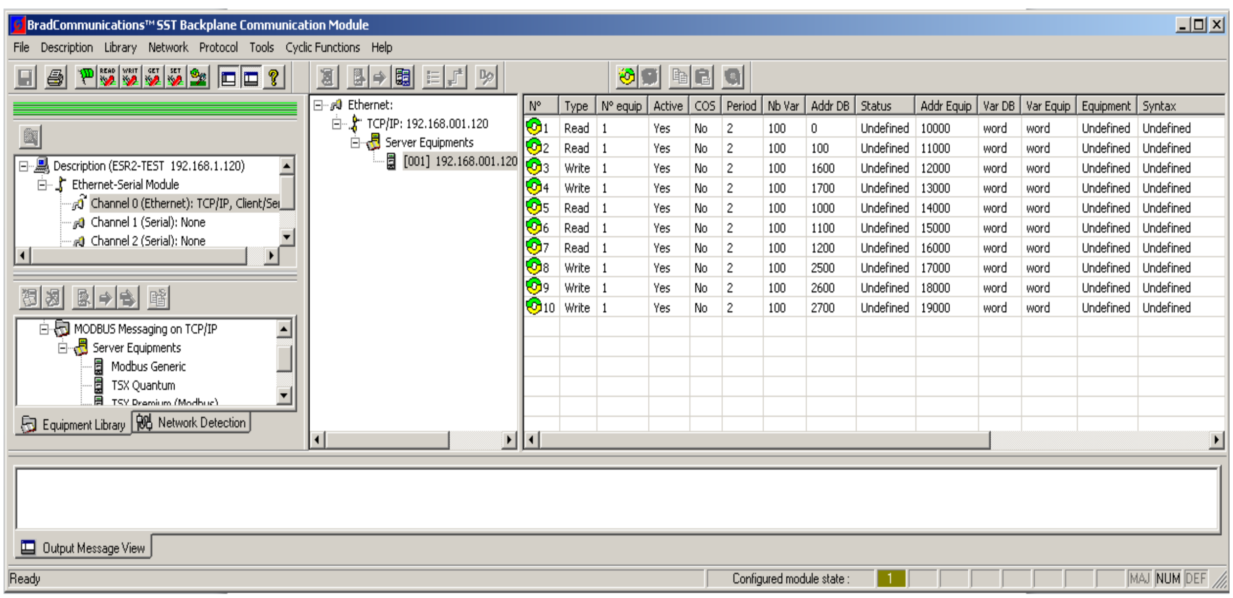
xiii. Number of data is 100

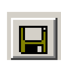
xiv. Protocol is Generic

xv. Device First variable address is 19000 (slave equipment address)

xvi. Database offset is 2700 (master equipment address)

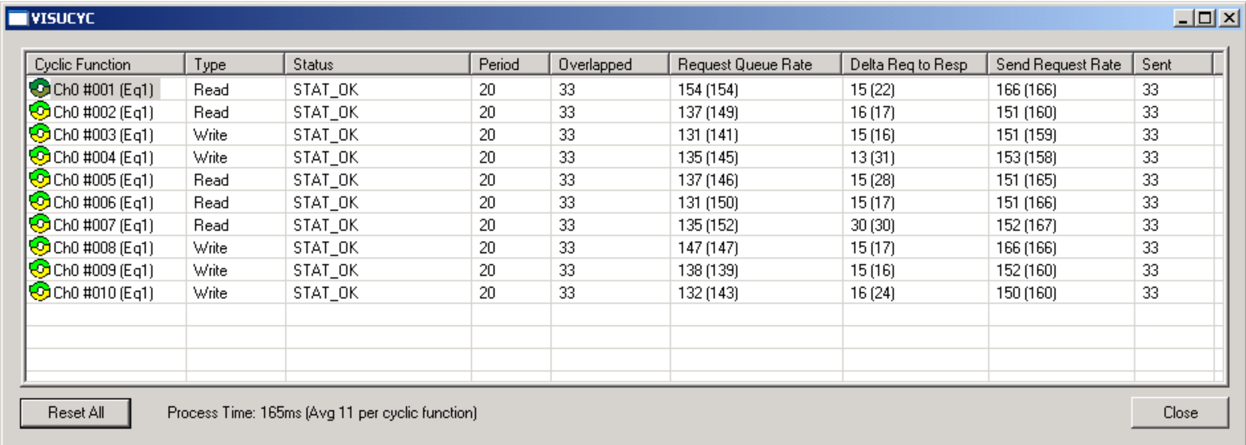
4. After all the cyclic functions are created, you should have similar display below.



5. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

6. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to [Console Status Bar Information] for status information on the communication to the module.

7. If initialization is successful, run the Visucyc application to check the status of each cyclic function. The similar display as below.


VISUCYC

Cyclic Function	Type	Status	Period	Overlapped	Request Queue Rate	Delta Req to Resp	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_OK	20	33	154 (154)	15 (22)	166 (166)	33
Ch0 #002 (Eq1)	Read	STAT_OK	20	33	137 (149)	16 (17)	151 (160)	33
Ch0 #003 (Eq1)	Write	STAT_OK	20	33	131 (141)	15 (16)	151 (159)	33
Ch0 #004 (Eq1)	Write	STAT_OK	20	33	135 (145)	13 (31)	153 (158)	33
Ch0 #005 (Eq1)	Read	STAT_OK	20	33	137 (146)	15 (28)	151 (165)	33
Ch0 #006 (Eq1)	Read	STAT_OK	20	33	131 (150)	15 (17)	151 (166)	33
Ch0 #007 (Eq1)	Read	STAT_OK	20	33	135 (152)	30 (30)	152 (167)	33
Ch0 #008 (Eq1)	Write	STAT_OK	20	33	147 (147)	15 (17)	166 (166)	33
Ch0 #009 (Eq1)	Write	STAT_OK	20	33	138 (139)	15 (16)	152 (160)	33
Ch0 #010 (Eq1)	Write	STAT_OK	20	33	132 (143)	16 (24)	150 (160)	33

Reset All Process Time: 165ms (Avg 11 per cyclic function) Close

8. Since the communication with the backplane is active, all cyclic functions are in “**active**” state.

Note

In this example, all cyclic functions are located in the mapped portion of the database.

9. Test all read cyclic function; cyclic function 1,2,5,6 & 7.
10. From the Console application window, locate and launch SetDB.
11. Enter 100 in Number of variables, enter 10000 in First variable address.
12. In Scan enter 1, then click [Scan value ++] and then OK, you’re setting values in server equipment address 10000-10099 with values 1-100.
13. Repeat step 11 but use First variable address 11000.
14. Repeat step 12 with value 100, then click [Scan value ++] and then OK, the values in server equipment address 11000-11099 are set with values 100-199.
15. Repeat step 11 but use First variable address 14000.
16. Repeat step 12 with value 50, then click [Scan value] and then OK, the values in server equipment address 14000-14099 are set with values 50.
17. Repeat step 11 but use First variable address 15000.
18. Repeat step 12 with value 60, then click [Scan value] and then OK, the values in server equipment address 15000-15099 are set with values 60.

19. Repeat step 11 but use First variable address 16000.

20. Repeat step 12 with value 50, then click Scan value and then OK, the values in server equipment address 16000-16099 are set with values 70.

21. To summarize the cyclic function used address in both client and server equipment for the Read Cyclic functions, see table below.

Table 14: Read Cyclic Function Test Summary

Cyclic Function #	CPU INPUT Table Address	Values	Client Equipment Database Start Address	Server Equipment Database Start Address
1	0	0-100	0	10000
2	100	100-199	100	11000
5	1000	50	1000	14000
6	1100	60	1100	15000
7	1200	70	1200	16000

22. In RSLogix 5000 Software, expand the CLX2000.DATABASE_DATA.INPUT_DATA, the values that are set in the server equipment address using SetDB are displayed as below.

For cylic function # 1:

CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5
CLX2000.DATABASE_DATA.INPUT_DATA[0]	1		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[2]	3		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[3]	4		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[4]	5		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[5]	6		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[6]	7		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[7]	8		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[8]	9		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[9]	10		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[10]	11		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[11]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[12]	13		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[13]	14		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[14]	15		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[15]	16		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[16]	17		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[17]	18		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[18]	19		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[19]	20		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[20]	21		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[21]	22		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[22]	23		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[23]	24		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[24]	25		Decimal	INT

For cylic function # 2

CLX2000.DATABASE_DATA.INPUT_DATA[99]	100		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[100]	100		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[101]	101		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[102]	102		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[103]	103		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[104]	104		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[105]	105		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[106]	106		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[107]	107		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[108]	108		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[109]	109		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[110]	110		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[111]	111		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[112]	112		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[113]	113		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[114]	114		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[115]	115		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[116]	116		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[117]	117		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[118]	118		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[119]	119		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[120]	120		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[121]	121		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[122]	122		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[123]	123		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[124]	124		Decimal	INT

For cyclic function # 5

		CLX2000.DATABASE_DATA.INPUT_DATA[1000]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1001]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1002]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1003]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1004]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1005]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1006]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1007]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1008]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1009]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1010]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1011]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1012]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1013]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1014]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1015]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1016]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1017]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1018]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1019]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1020]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1021]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1022]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1023]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1024]	50		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1025]	50		Decimal	INT

For cyclic function # 6

		CLX2000.DATABASE_DATA.INPUT_DATA[1100]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1101]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1102]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1103]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1104]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1105]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1106]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1107]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1108]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1109]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1110]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1111]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1112]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1113]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1114]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1115]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1116]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1117]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1118]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1119]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1120]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1121]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1122]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1123]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1124]	60		Decimal	INT
		CLX2000.DATABASE_DATA.INPUT_DATA[1125]	60		Decimal	INT

For cyclic function # 7

CLX2000.DATABASE_DATA.INPUT_DATA[1200]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1201]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1202]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1203]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1204]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1205]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1206]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1207]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1208]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1209]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1210]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1211]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1212]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1213]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1214]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1215]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1216]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1217]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1218]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1219]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1220]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1221]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1222]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1223]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1224]	70		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1225]	70		Decimal	INT

23. Now test the write cyclic function.
24. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

For Cyclic Function #3:

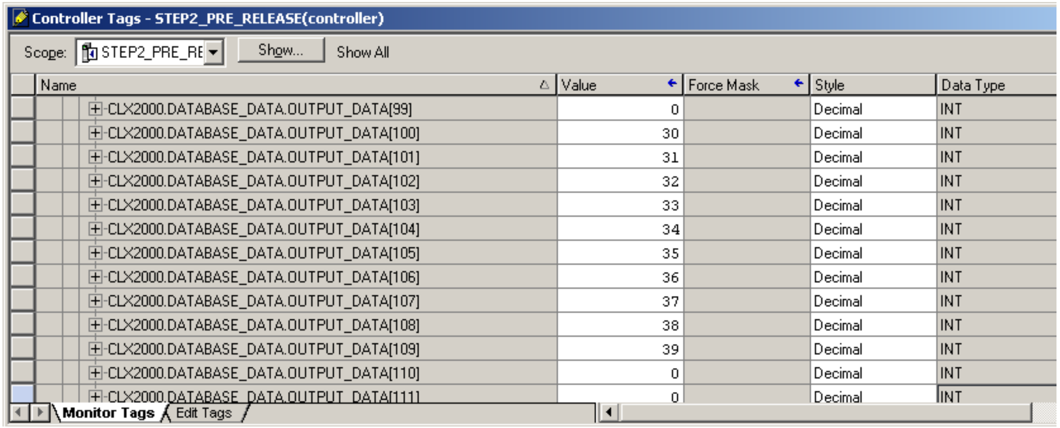
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.OUTPUT_DATA[0]	20		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1]	21		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[2]	22		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[3]	23		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[4]	24		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[5]	25		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[6]	26		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[7]	27		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[8]	28		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[9]	29		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[10]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[11]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function # 4



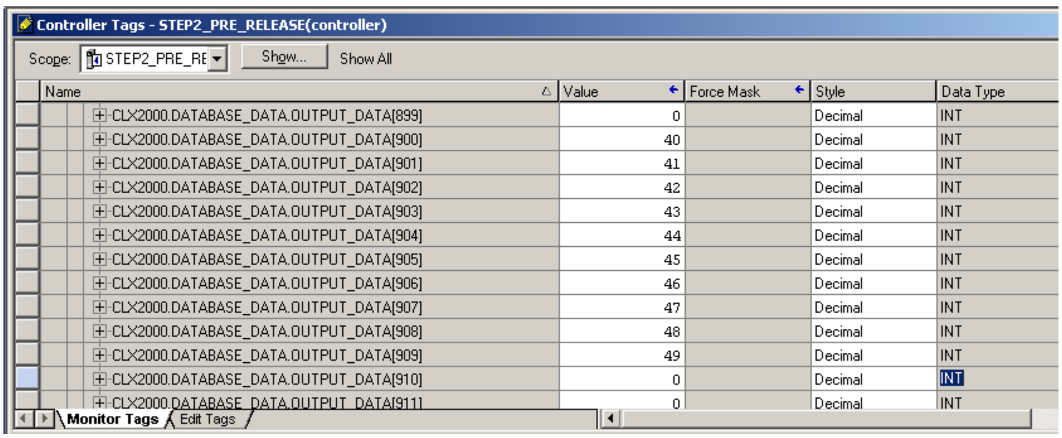
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[99]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[100]	30		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[101]	31		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[102]	32		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[103]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[104]	34		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[105]	35		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[106]	36		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[107]	37		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[108]	38		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[109]	39		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[110]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[111]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function #8:



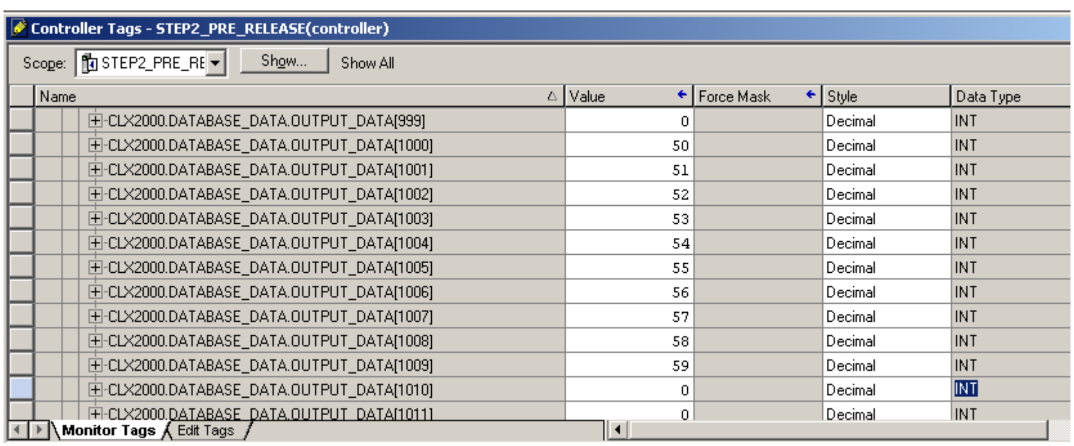
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[899]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[900]	40		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[901]	41		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[902]	42		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[903]	43		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[904]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[905]	45		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[906]	46		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[907]	47		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[908]	48		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[909]	49		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[910]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[911]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function # 9:



Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[999]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1000]	50		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1001]	51		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1002]	52		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1003]	53		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1004]	54		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1005]	55		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1006]	56		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1007]	57		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1008]	58		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1009]	59		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1010]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1011]	0		Decimal	INT

Monitor Tags / Edit Tags

For Cyclic Function # 10:

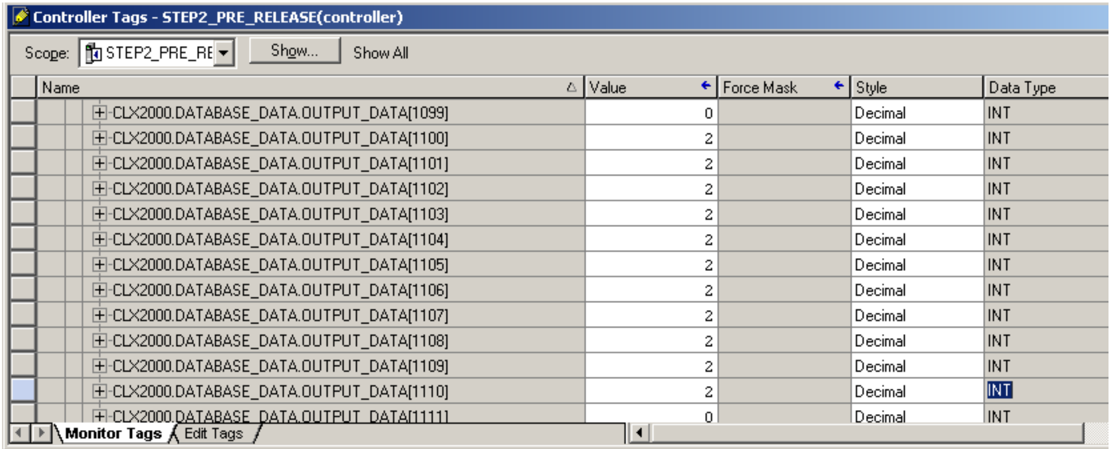
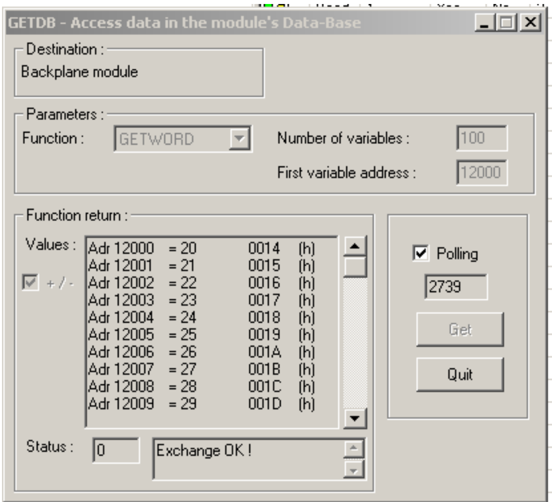


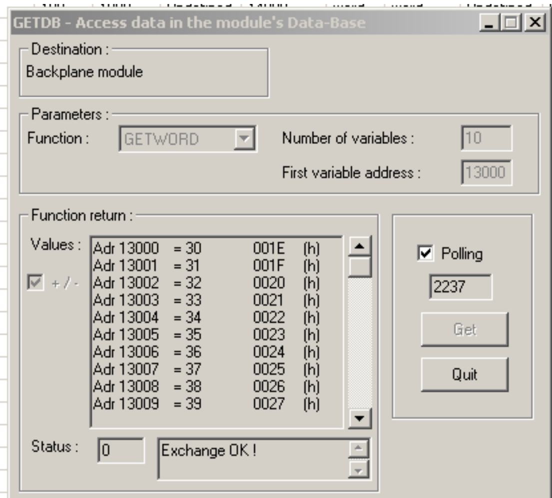
Table 15: Write Cyclic Function Test Summary

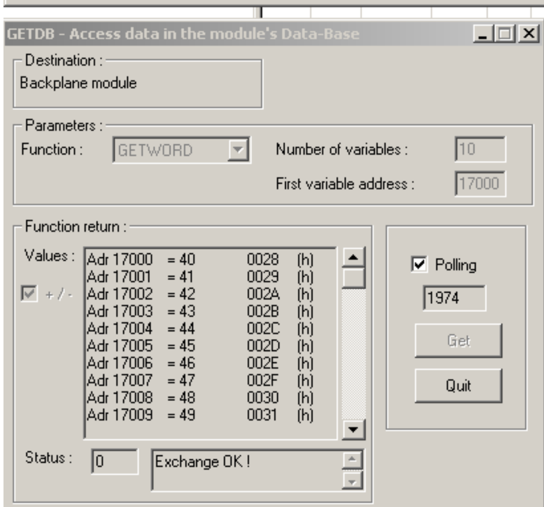
Cyclic Function #	CPU OUTPUT Table address	Values	Client Equipment Database Start Address	Server Equipment Database Start Address
3	0	20-29	1600	12000
4	100	30-39	1700	13000
8	900	40-49	2500	17000
9	1000	50-59	2600	18000
10	1100	2	2700	19000

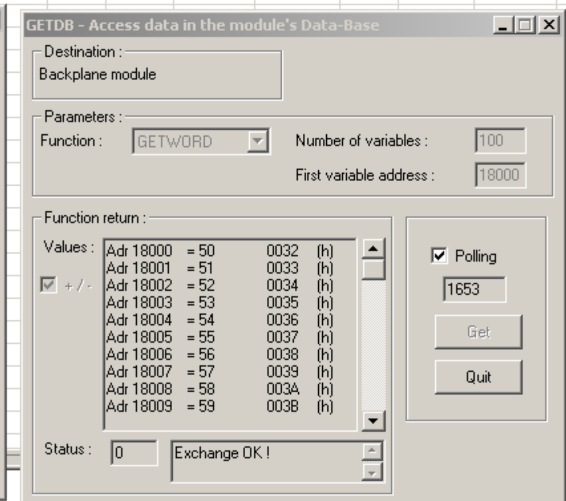
25. Read the server equipment addresses as indicated in the table above. See result below. The results should be same as the values in the above table. Write 10 words for each write cyclic function, so can only see 10 non-zero values using GetDB for each server equipment address.

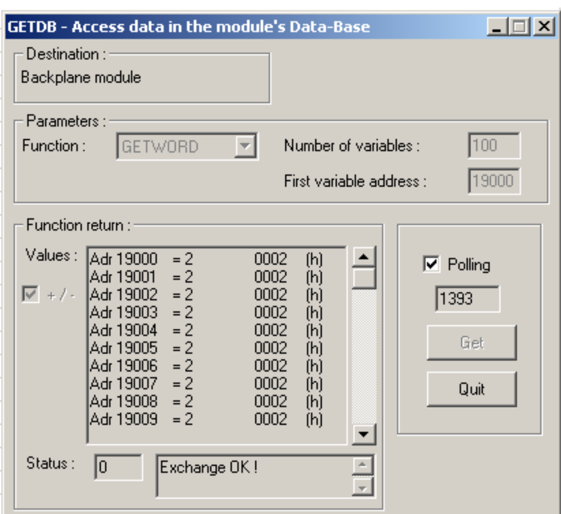
26. From the Console application window, locate and launch GetDB  .


GETDB - Access data in the module's Data-Base
Destination :
Backplane module
Parameters :
Function : GETWORD
Number of variables : 100
First variable address : 12000
Function return :
Values :
+ / -
Adr 12000 = 20 0014 (h)
Adr 12001 = 21 0015 (h)
Adr 12002 = 22 0016 (h)
Adr 12003 = 23 0017 (h)
Adr 12004 = 24 0018 (h)
Adr 12005 = 25 0019 (h)
Adr 12006 = 26 001A (h)
Adr 12007 = 27 001B (h)
Adr 12008 = 28 001C (h)
Adr 12009 = 29 001D (h)
Polling
2739
Get
Quit
Status : 0
Exchange OK !


GETDB - Access data in the module's Data-Base
Destination :
Backplane module
Parameters :
Function : GETWORD
Number of variables : 10
First variable address : 13000
Function return :
Values :
+ / -
Adr 13000 = 30 001E (h)
Adr 13001 = 31 001F (h)
Adr 13002 = 32 0020 (h)
Adr 13003 = 33 0021 (h)
Adr 13004 = 34 0022 (h)
Adr 13005 = 35 0023 (h)
Adr 13006 = 36 0024 (h)
Adr 13007 = 37 0025 (h)
Adr 13008 = 38 0026 (h)
Adr 13009 = 39 0027 (h)
Polling
2237
Get
Quit
Status : 0
Exchange OK !


GETDB - Access data in the module's Data-Base
Destination :
Backplane module
Parameters :
Function : GETWORD
Number of variables : 10
First variable address : 17000
Function return :
Values :
+ / -
Adr 17000 = 40 0028 (h)
Adr 17001 = 41 0029 (h)
Adr 17002 = 42 002A (h)
Adr 17003 = 43 002B (h)
Adr 17004 = 44 002C (h)
Adr 17005 = 45 002D (h)
Adr 17006 = 46 002E (h)
Adr 17007 = 47 002F (h)
Adr 17008 = 48 0030 (h)
Adr 17009 = 49 0031 (h)
Polling
1974
Get
Quit
Status : 0
Exchange OK !


GETDB - Access data in the module's Data-Base
Destination :
Backplane module
Parameters :
Function : GETWORD
Number of variables : 100
First variable address : 18000
Function return :
Values :
+ / -
Adr 18000 = 50 0032 (h)
Adr 18001 = 51 0033 (h)
Adr 18002 = 52 0034 (h)
Adr 18003 = 53 0035 (h)
Adr 18004 = 54 0036 (h)
Adr 18005 = 55 0037 (h)
Adr 18006 = 56 0038 (h)
Adr 18007 = 57 0039 (h)
Adr 18008 = 58 003A (h)
Adr 18009 = 59 003B (h)
Polling
1653
Get
Quit
Status : 0
Exchange OK !


GETDB - Access data in the module's Data-Base
Destination :
Backplane module
Parameters :
Function : GETWORD
Number of variables : 100
First variable address : 19000
Function return :
Values :
+ / -
Adr 19000 = 2 0002 (h)
Adr 19001 = 2 0002 (h)
Adr 19002 = 2 0002 (h)
Adr 19003 = 2 0002 (h)
Adr 19004 = 2 0002 (h)
Adr 19005 = 2 0002 (h)
Adr 19006 = 2 0002 (h)
Adr 19007 = 2 0002 (h)
Adr 19008 = 2 0002 (h)
Adr 19009 = 2 0002 (h)
Polling
1393
Get
Quit
Status : 0
Exchange OK !

Example 3: Changing the database mapping size.

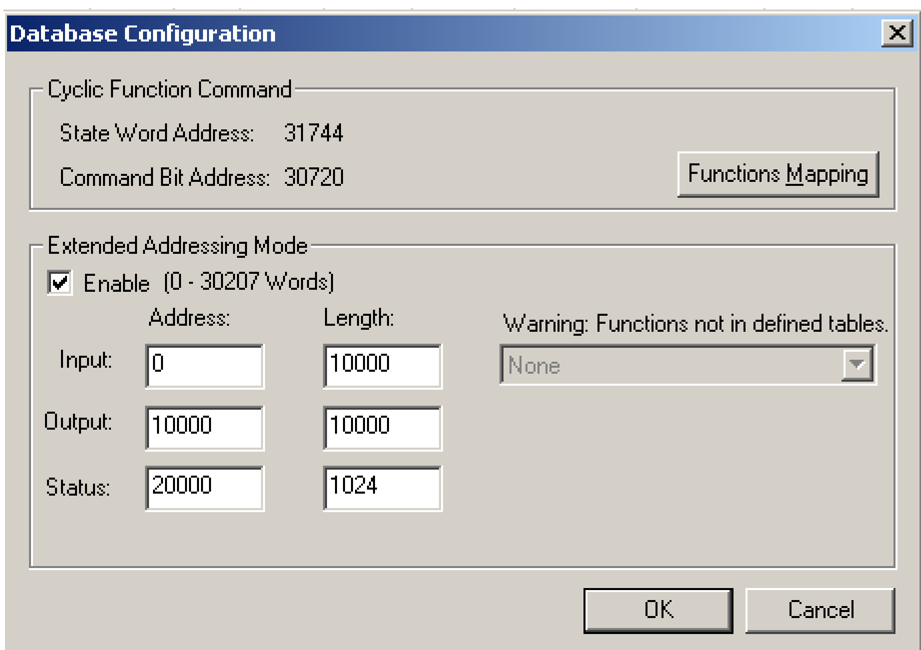
This sample configuration will only change the database mapping size. This will be a continuation from Example 2.

1. In the RsLogix 5000 software, change the connection state to Offline.

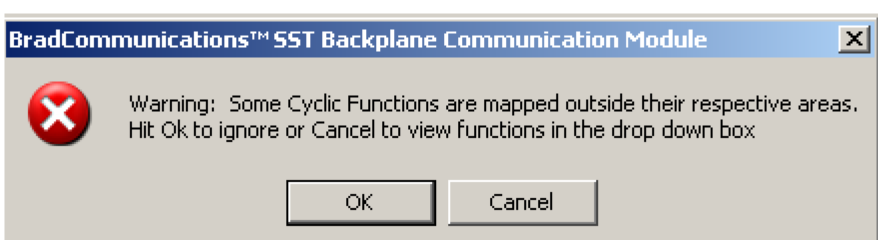
2. From the console application window, run Database and change the INPUT, OUTPUT and STATUS table addresses and size.

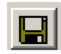
Warning

This is just a demonstration of how to change the AOI mapping address. The values that will be used here will cause the cyclic function to stop working since addresses between client and server in the same module are now overlapping



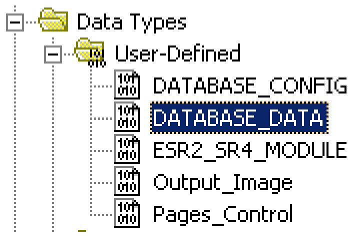
When the values are entered above, the message below will pop up, click ok to continue.



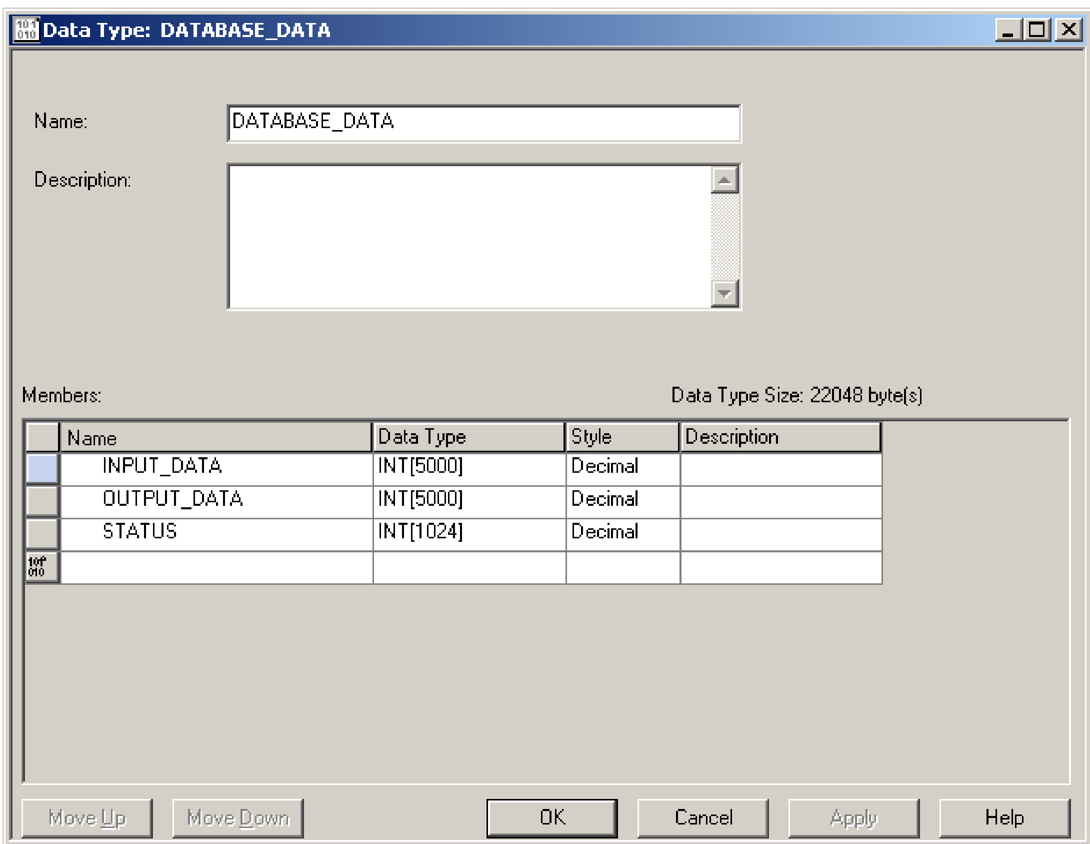
3. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

4. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

5. From the Rslogix 5000, double click DATABASE_DATA under Data Types, see below.



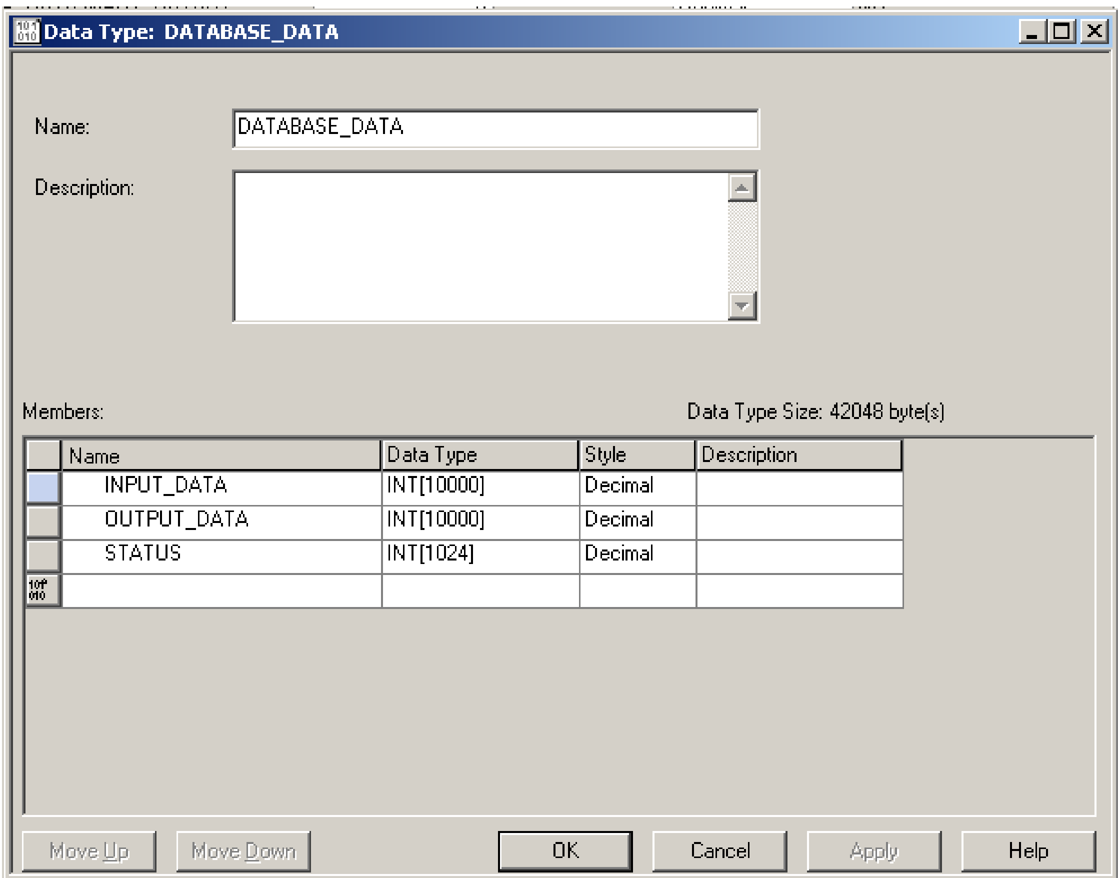
6. The window below should be displayed on the screen



The current AOI configuration allows up 5000 direct mapping of INPUT and OUTPUT table. Before changing the value read some important notes below:

- If the desired mapping size fits in this size, there is no need to change anything from the backplane configuration. Only the database configuration from the console needs to be changed.
- If the desired mapping is greater than what is currently defined, change it to the desired value.

7. Since the INPUT and OUTPUT mapping sizes are 10000 words each, the sizes need to be increased here as well. After the change, the display should be similar as below.



8. Click OK and check if CLX2000.DATABASE_DATA.INPUT_DATA and CLX2000.DATABASE_DATA.OUTPUT_DATA size arrays did change.
9. The new array size should now be 0-9999.
10. Save the configuration file and download it to the CLX Module.

6.5.3 Siemens S7/S5 Messaging

6.5.3.1 Siemens S7/S5 in Default Addressing Mode

When the module is configured in extended addressing mode, the entire address defined in Input Table, Output Table and Status Table from the Console Database Configuration are accessible from the backplane using our AOI (Add-On-Instruction).

In this configuration, 1600 words of INPUT, 1600 words of OUTPUT and 1024 words of STATUS of the module's database are mapped to the backplane. The database configuration needs to use extended addressing mode to be able to map the database larger than the maximum INPUT and OUTPUT table connection size allowed in the ControlNet.

A quick description of how the Siemens PLC Variable addressing is mapped to the modules database configured in default addressing mode when running as server equipment is shown here.

For more detailed information on the Siemens PLC addressing, please refer to the Industrial Ethernet Protocol Guide.

Table 16: Siemens Supported PLC Variables and Database mapping

Supported requests	Database data type	Database Address	Corresponding S7 addresses
Read Inputs (EW)	Word	250-496	EW0 – EW246
Read Outputs (AW)	Word	0-246	AW0 - AW246
Write Outputs (AW)	Word	0-246	AW0 - AW246
Read Memo (MW)	Word	768-1023	MW0 – MW255
Write Memo (MW)	Word	768-1023	MW0 – MW255
Read Data Blocks (DBW)	Word	1024 – 29952	DB4.DBW0 – DB116.DBW255
Write Data Blocks (DBW)	Word	1024 - 29952	DB4.DBW0 – DB116.DBW255

No Backplane Connection:

To be able to demonstrate successfully this sample configuration, Siemens S7/S5 Client equipment and Siemens S7/S5 Server equipment are needed and they must have an active connection to be able to show data exchanges configured through cyclic function.

The SST-ESR2-CLX-RLL product is used in this sample. If there are two available modules of this product variant, configure one as Siemens S7/S5 Client and the other one as Siemens S7/S5 Server equipment. If there is only one module, the single communication module will be configured both as a Siemens S7/S5 Client and a Siemens S7/S5 Server. Cyclic functions can only be created when module is running in client equipment configuration.

Note

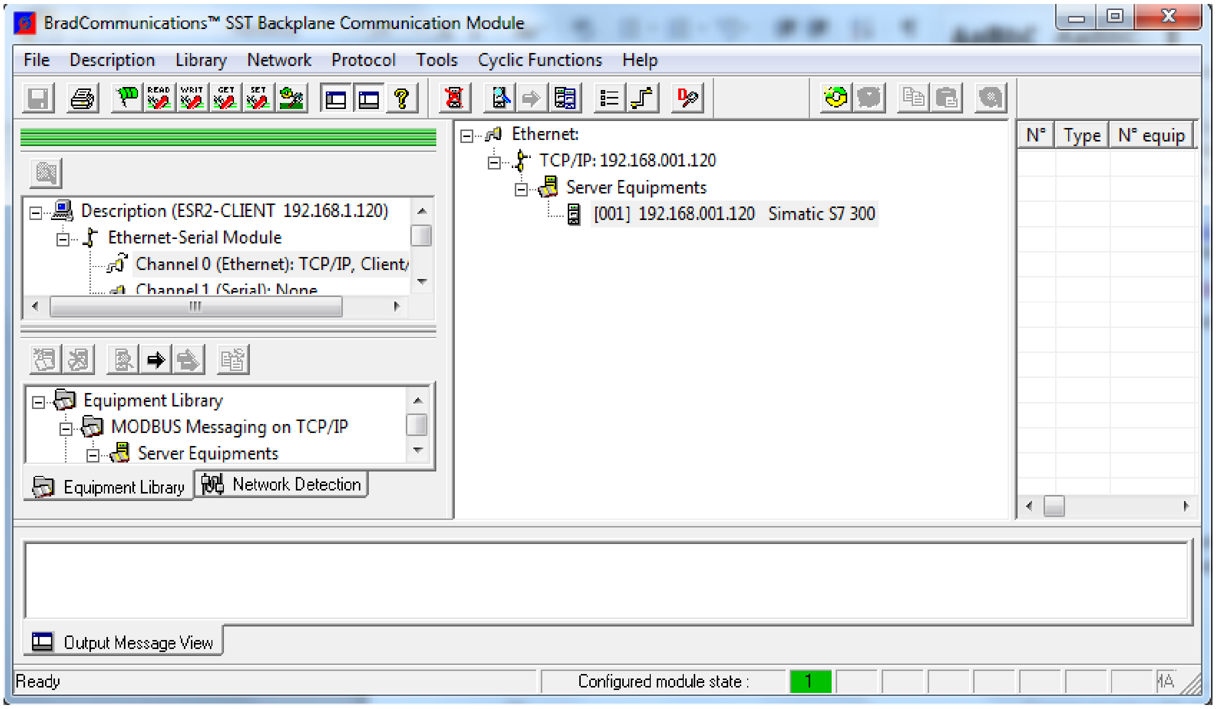
Since client and server equipment is sharing the same database, make sure different addresses are used for the source and destination address of the data.

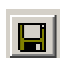
The procedures below is based on using a single communication module for both client and server configuration.

The firmware version should be 2.10.2 or higher to be able to use this feature. Follow the steps below to configure the module in extended addressing mode.

1. Complete procedures in Configure the Module as Siemens Client/Server Equipment.

2. After completing the steps in Configure the Module as Siemens Client/Server Equipment section, The Console application main screen should be displayed with Port 0 (TCP/IP) configured as Client and Siemens S7-300 server equipment with IP address 192.168.1.120 is added, shown below.

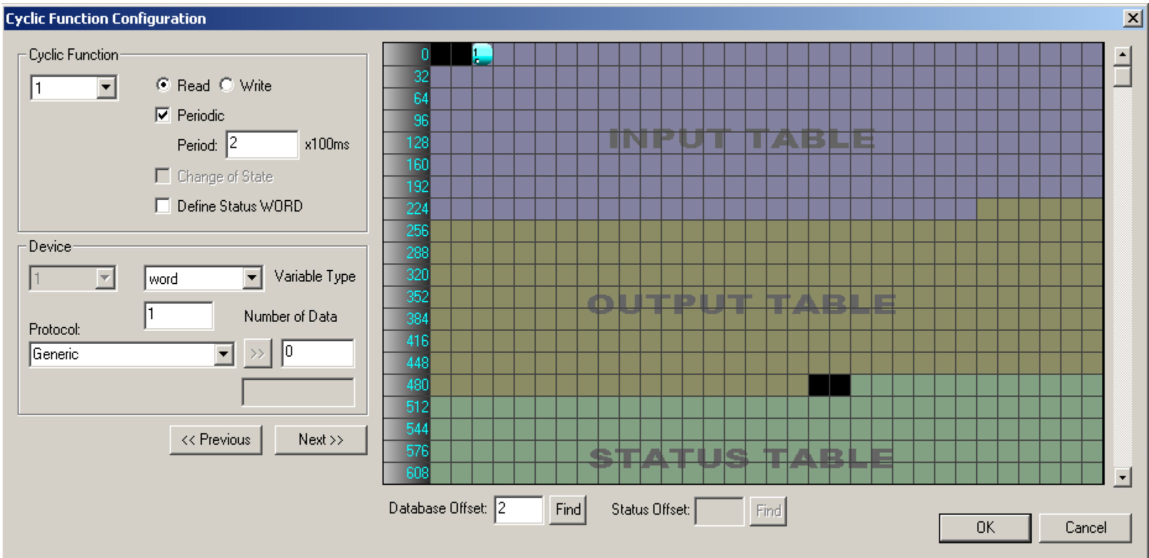


3. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

4. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to [Console Status Bar Information] for status information on the communication to the module.

5. Start creating cyclic function by clicking . The cyclic function created will provide automatic data exchange between the client and server equipment. Creating both read and write cyclic functions to automatically read and write data from/to the server equipment variable addresses. The function code as well as the size of the data to be read is part of the cyclic function parameters.

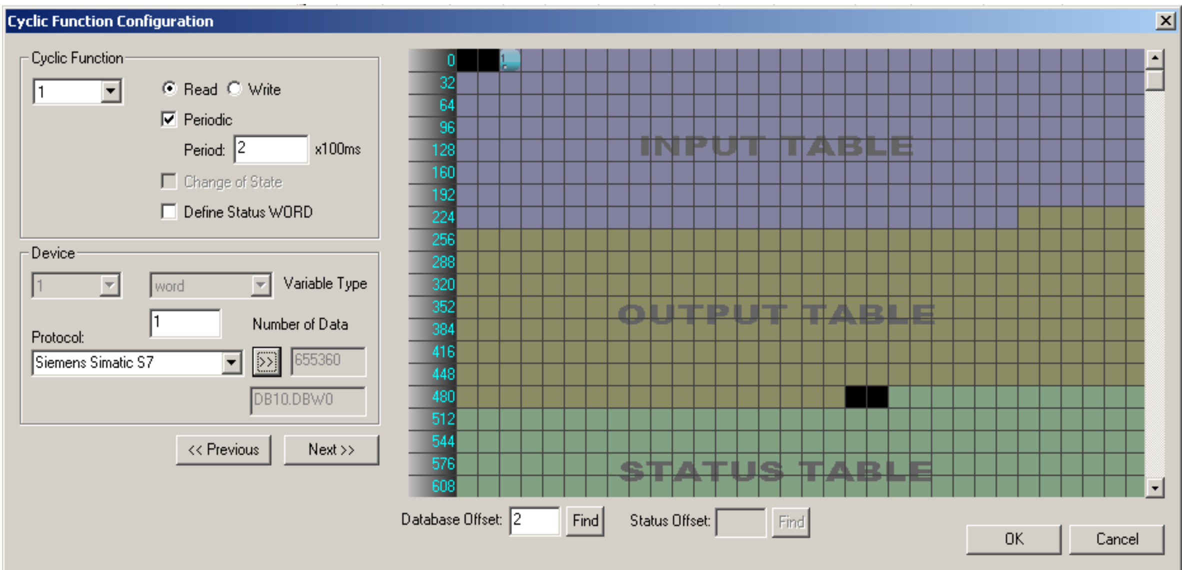
Figure 81: Cyclic Function Dialog Window



The detail of each parameter in discussed in Cyclic Function Configuration Parameters and Table 38.

a. Create cyclic function #1 with the parameter below.
   i. Use default cyclic function number provided (1 should be the default if this the first cyclic function)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Siemens Simatic S7
   vii. PLC Variable Syntax DB10.DW0 (server equipment address:2560)
   viii. Database offset is 2 (client equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below.



b. Click "OK", and the first cyclic function will be shown in the cyclic function list view
c. Create cyclic function #2 with the parameter below
    i. Use default cyclic function number provided (2 should be the default now)
    ii. Select Read cyclic function type
    iii. Check the Periodic setting
    iv. Use default period of 2 x 10ms = 20ms
    v. Number of data is 1
    vi. Protocol is Siemens Simatic S7
    vii. PLC Variable Syntax DB11.DW0 (server equipment address:2816)
    viii. Database offset is 1000 (client equipment address)

d. Click "OK", and the second cyclic function will be shown in the cyclic function list view.
The difference between the first and second cyclic function is the database offset location in the client.

1st cyclic function database address is located in the mapped IO table address 0-699
2nd cyclic function database address is located outside of the mapped IO table address

e. Create cyclic function #3 with the parameters below:
    i. Use default cyclic function number provided (3 should be the default now)
    ii. Select Write cyclic function type
    iii. Check the Periodic setting
    iv. Use default period of 2 x 10ms = 20ms
    v. Number of data is 1
    vi. Protocol is Siemens Simatic S7

vii. PLC Variable Syntax DB12.DW0 (server equipment address:3072)

viii. Database offset is 250 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view

g. Create cyclic function #4 with the parameter below

i. Use default cyclic function number provided (4 should be the default now)

ii. Select Write cyclic function type

iii. Check the Periodic setting

iv. Use default period of 2 x 10ms = 20ms

v. Number of data is 1

vi. Protocol is Generic

vii. PLC Variable Syntax DB13.DW0 (server equipment address:3328)

viii. Database offset is 2000 (client equipment address)

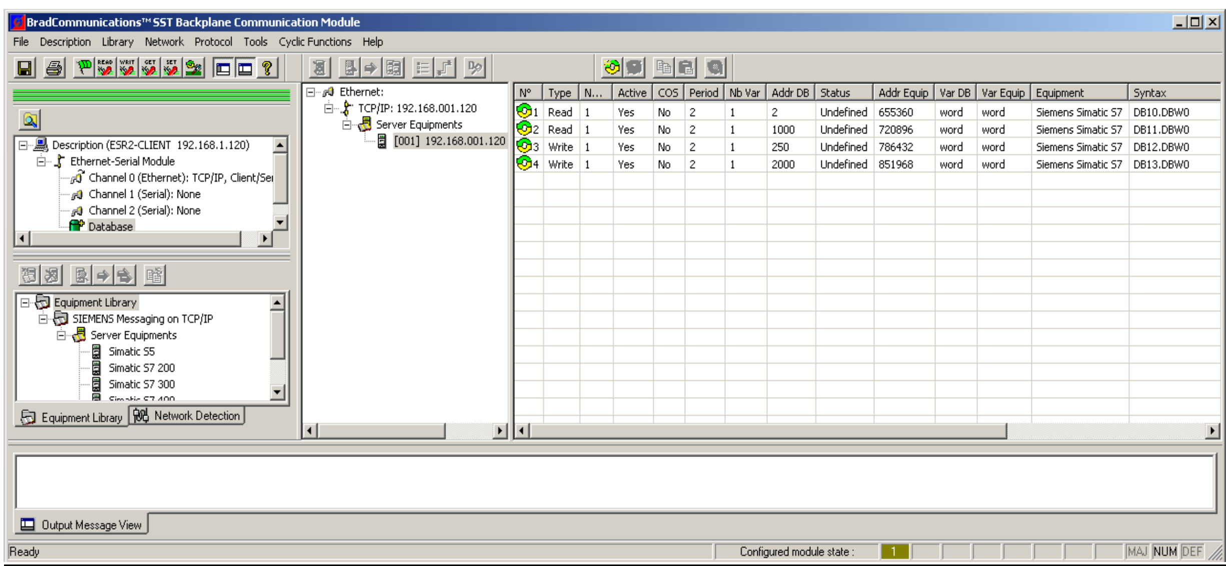
h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

The difference between the first and second cyclic function is the database offset location in the client.

3rd cyclic function database address is located in the mapped IO table address 0-699

4th cyclic function database address is located outside of the mapped IO table address

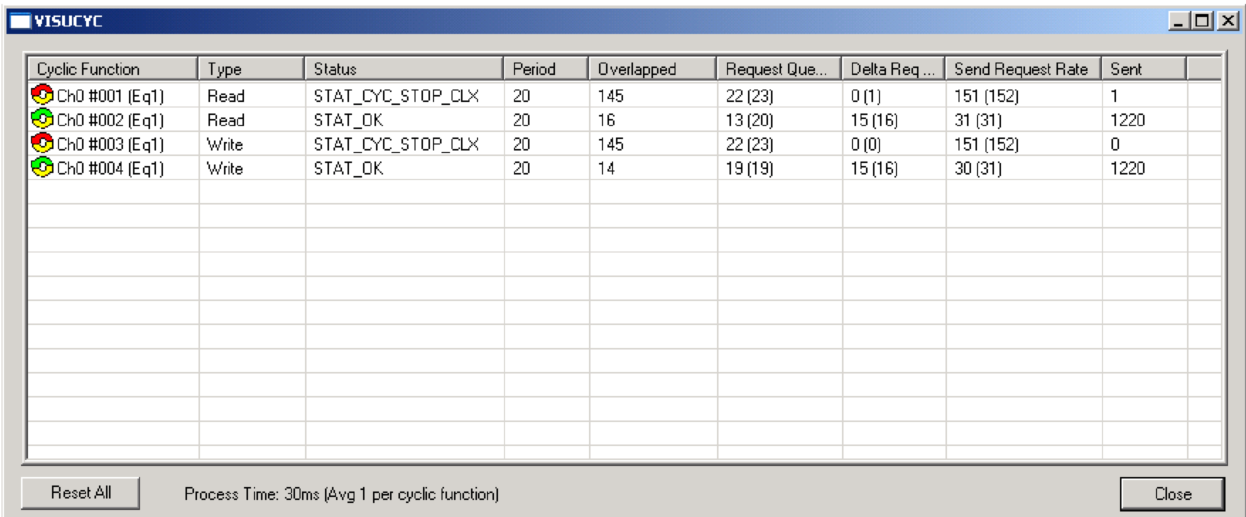
6. After creating all the cyclic function your screen should have a similar display as below.



7. Download the configuration to the module by clicking  and .

8. When download has completed, (Refer to Console Status Bar Information to check if module is properly initialized), run the visual cyclic function application by clicking  from the main console screen.

   The display is shown below.



VISUCYC

Cyclic Function	Type	Status	Period	Overlapped	Request Que...	Delta Req ...	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_CYC_STOP_CLX	20	145	22 (23)	0 (1)	151 (152)	1
Ch0 #002 (Eq1)	Read	STAT_OK	20	16	13 (20)	15 (16)	31 (31)	1220
Ch0 #003 (Eq1)	Write	STAT_CYC_STOP_CLX	20	145	22 (23)	0 (0)	151 (152)	0
Ch0 #004 (Eq1)	Write	STAT_OK	20	14	19 (19)	15 (16)	30 (31)	1220

Reset All   Process Time: 30ms (Avg 1 per cyclic function)   Close

*<u>Both 1st and 3rd cyclic function status is 135 – **not active / not running**</u>*

*<u>Both 2nd and 4th cyclic function status is 0 – **active / running**</u>*

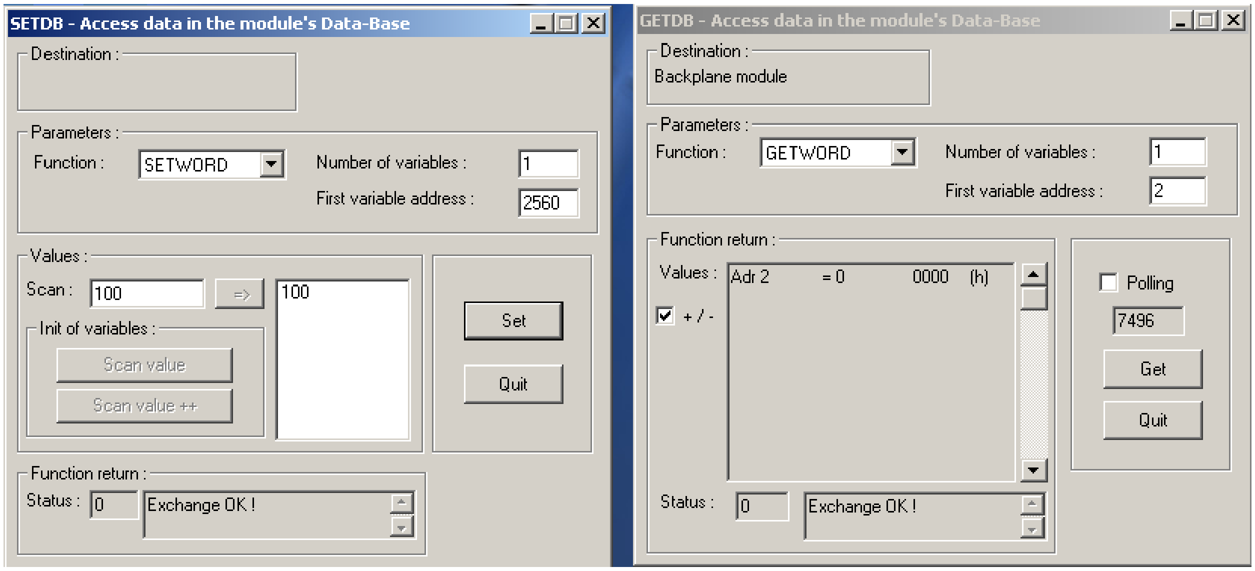
Note

The module will not run any cyclic function configured in the database address mapped to INPUT, OUTPUT and STATUS Table when there is no backplane connection.

9. To check if active cyclic function is running properly, follow this brief diagnostic below:

   a. Locate and run  and  application.

   b. To test cyclic function # 1, in the SetDB application fill in the First Variable address with value 2560 (DB10.DW0). In Scan Edit box, fill in with value 100, click on Scan value then Set.

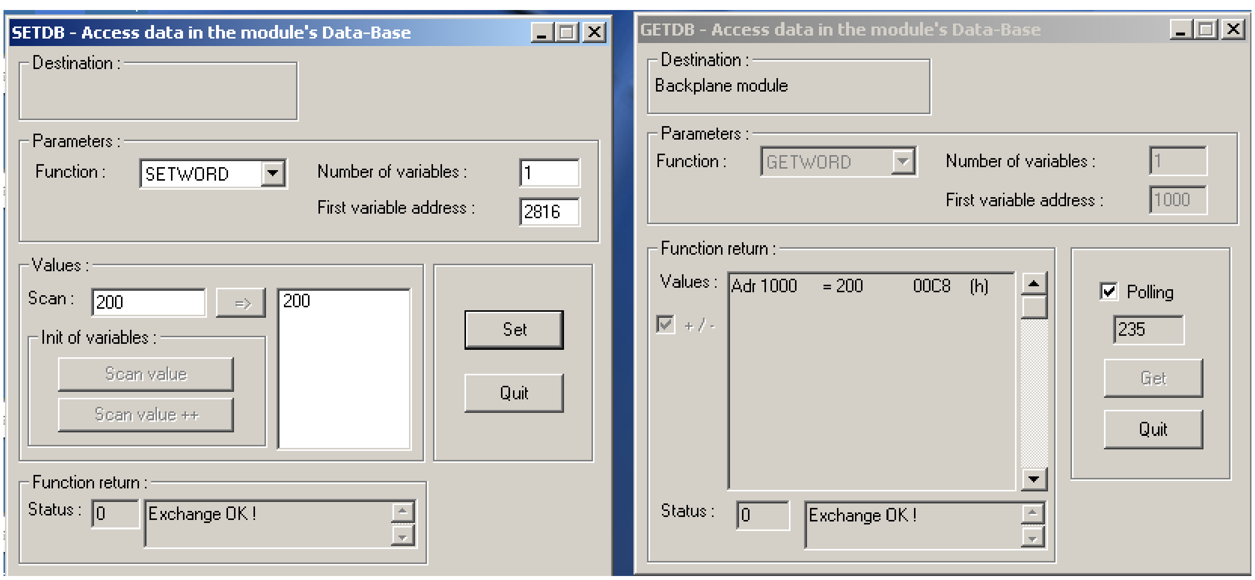


SetDB sets the server equipment address 2560 (DB10.DW0) with value 100.

GetDB reads the client equipment database address 2 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 2.

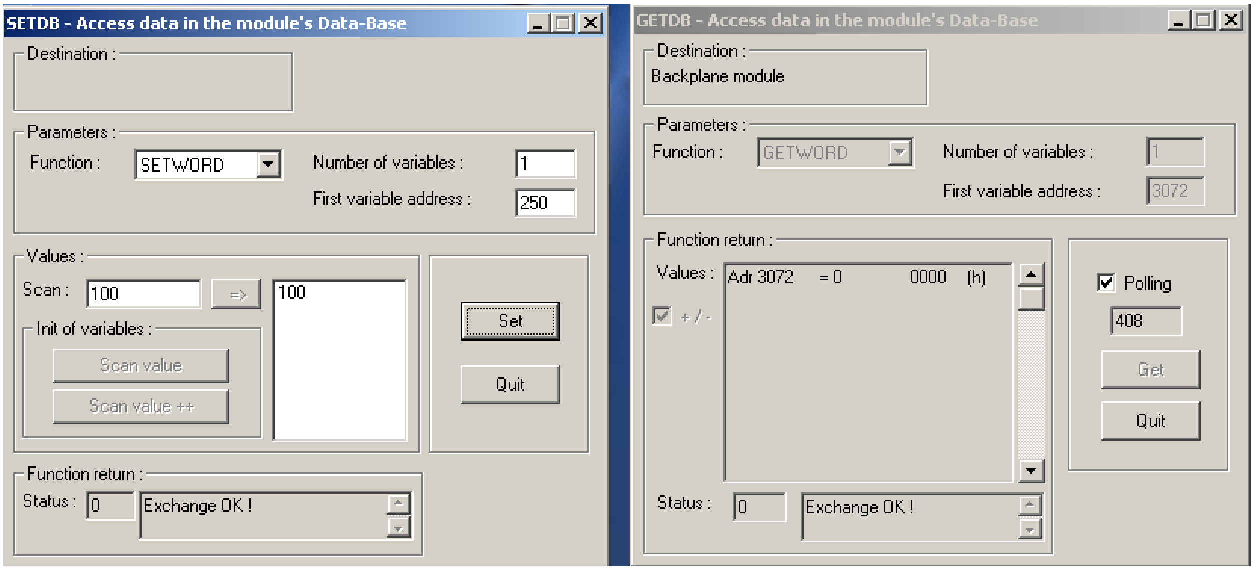
c. To test cyclic function # 2, fill in values as shown below.



SetDB sets the server equipment address 2816 (DB11.DW0) with value 200.

GetDB reads the client equipment database address 1000 with value shown as 200.

d. To test cyclic function # 3, fill in the values as show below.

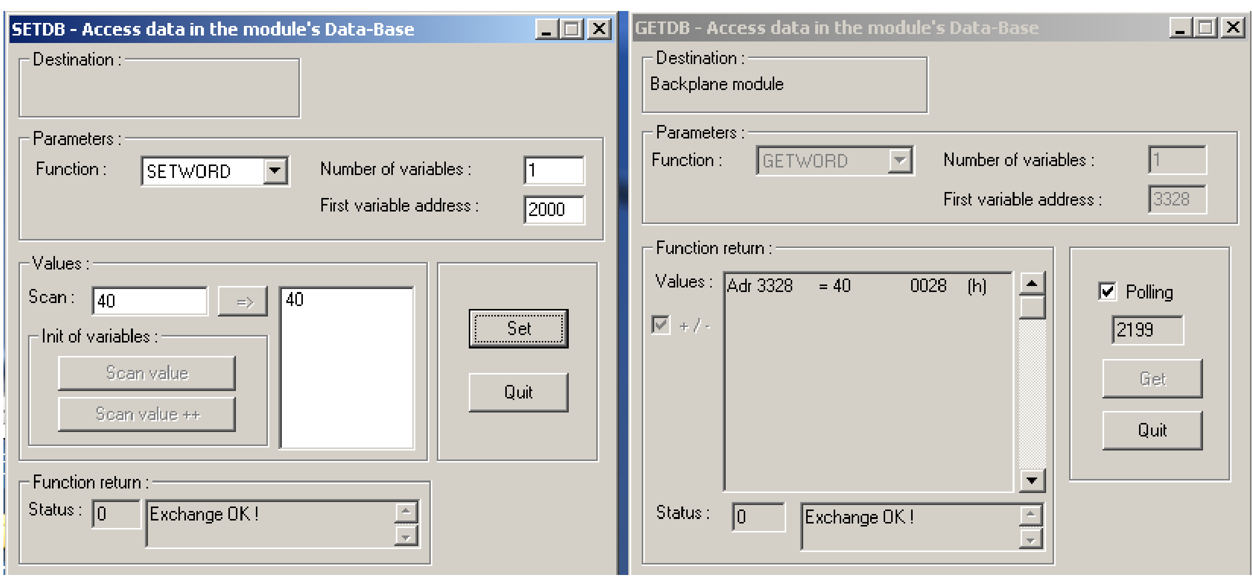


SetDB sets the client database address 250 with value 100.

GetDB reads the server equipment database address 3072 (DB12.DW0) with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #3 is ***not active***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 250.

e. To test cyclic function #4, fill in the values as shown below.



SetDB sets the client equipment address 2000 with value 40.

GetDB reads the server equipment address 3328 (DB13.DW0) with value shown as 40.

In the next set of procedures, The module communicates with the ControlLogix CPU. When backplane connection becomes active, all cyclic function with **"not active / not running"** state will switch to "**active/running**".

With Backplane Connection:

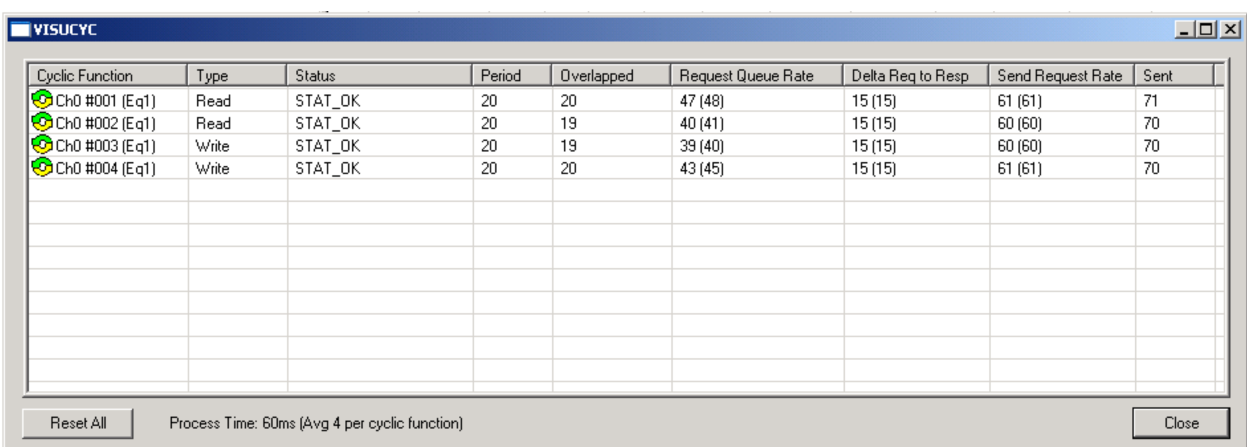
To be able to demonstrate successfully this sample configuration, it is assumed that the previous sets of procedures (from **No Backplane Connection**) were completed successfully.

1. Create a backplane connection by following procedures from Using 1756 Generic Profile section.

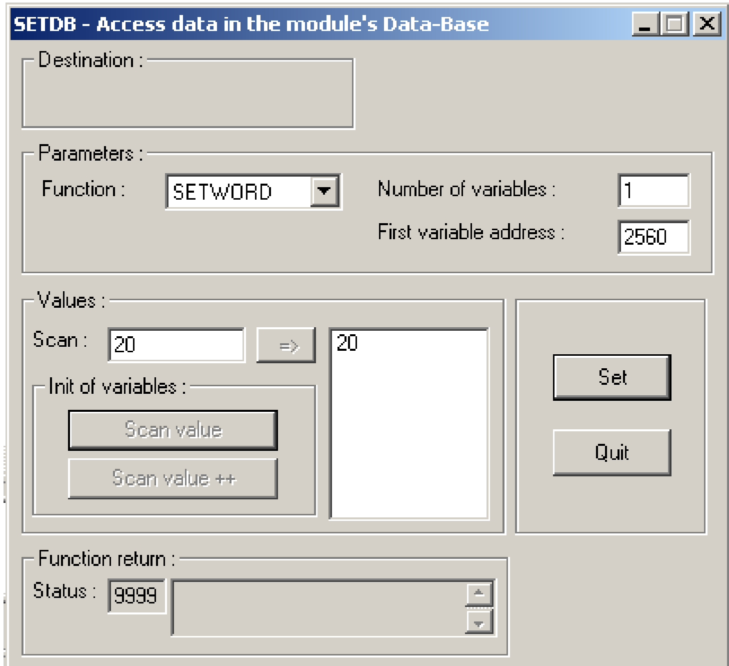
2. Make sure using the right Ethernet/IP address, the correct slot location of the module, and make it into "Run Mode". See below.

3. At this point, all the LEDs on the module are in Green color, and "COPN" alternately displayed with the module's IP address and Configuration name, configured in step #1.

4. Launch the VisuCyc and all the cyclic functions are in active status. See below.

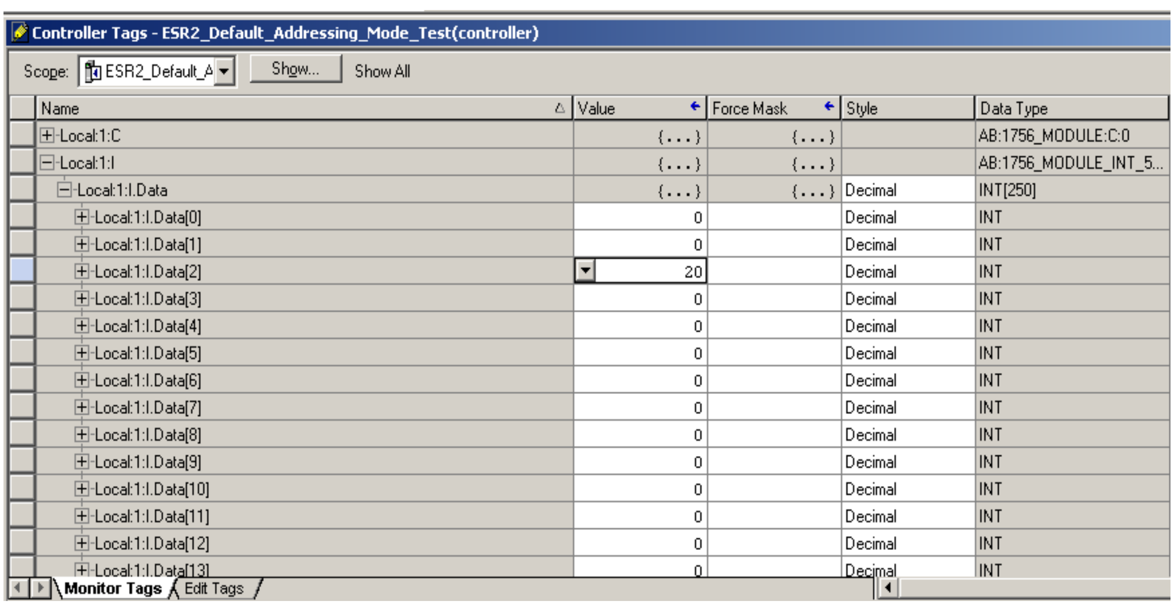


5. Test cyclic function 1 & 3 to check if values written and read from the mapped database addresses are exchanged properly. To test cyclic function #1, fill in the values as shown below.

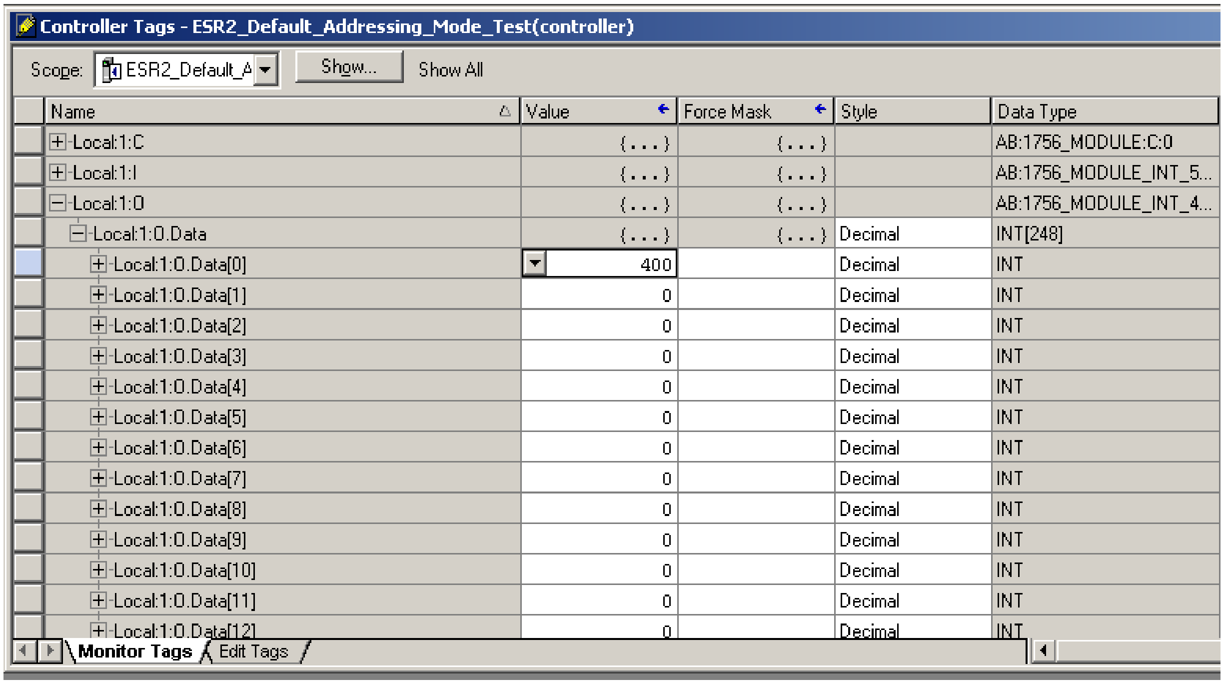


SetDB sets the server equipment address 2560 (DB10.DW0) with value 20.

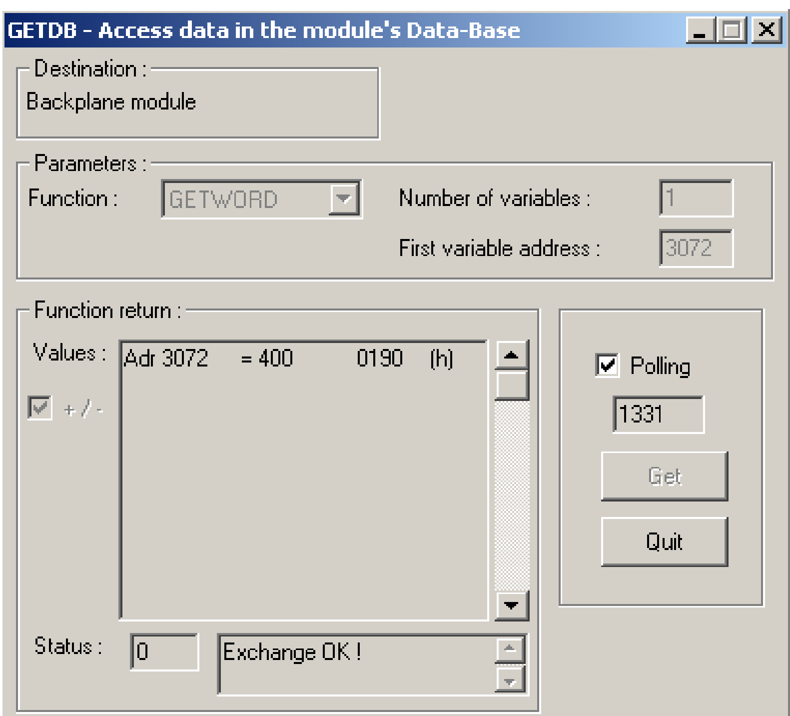
6. In RSLogix 5000 Software, expand the INPUT Table in the Controller Tags. The value 20 appears in offset 2. See below.



7. To test cyclic function #3, expand Output Table and write 400 into offset 0 (offset 250 in database)



Run ***GetDB*** and read the values as offset 3072 (DB12.DW0).



8. Now start adding the test of the desired configuration.

6.5.3.2 Siemens S7/S5 in Extended Addressing Mode

When module is configured in extended addressing mode, the entire address defined in Input Table, Output Table and Status Table from the Console Database Configuration are accessible from the backplane using our AOI (Add-On-Instruction).

In this configuration, 1600 words of INPUT, 1600 words of OUTPUT and 1024 words of STATUS of the module's database are mapped to the backplane. The database configuration needs to use extended addressing mode to be able to map the database larger than the maximum INPUT and OUTPUT table connection size allowed in the ControlNet.

A quick description of how the Siemens PLC Variable addressing is mapped to the modules database configured in extended addressing when running as server equipment is shown here. For more detailed information on using Siemens PLC addressing, please refer to the Industrial Ethernet Protocol Guide.

Table 17: Siemens Supported PLC Variables and Database mapping

Supported requests	Database data type	Database Address	Corresponding S7 addresses
Read Inputs (EW)	Word	*v + ***y***	EW***y*** (y:0-255)
Read Outputs (AW)	Word	*w + ***y***	AW***y*** (y:0-255)
Write Outputs (AW)	Word	*w + ***y***	AW***y*** (y:0-255)
Read Memo (MW)	Word	*v+256 + ***a***	MW***a*** (y:0-127)
Write Memo (MW)	Word	*w+128+ ***b***	MW***b*** (b:128-255)
Read Data Blocks (DBW)	Word	**256 – 29952(word)	DB1.DBW0 – DB116.DBW255
Write Data Blocks (DBW)	Word	**256 – 29952(word)	DB1.DBW0 – DB116.DBW255

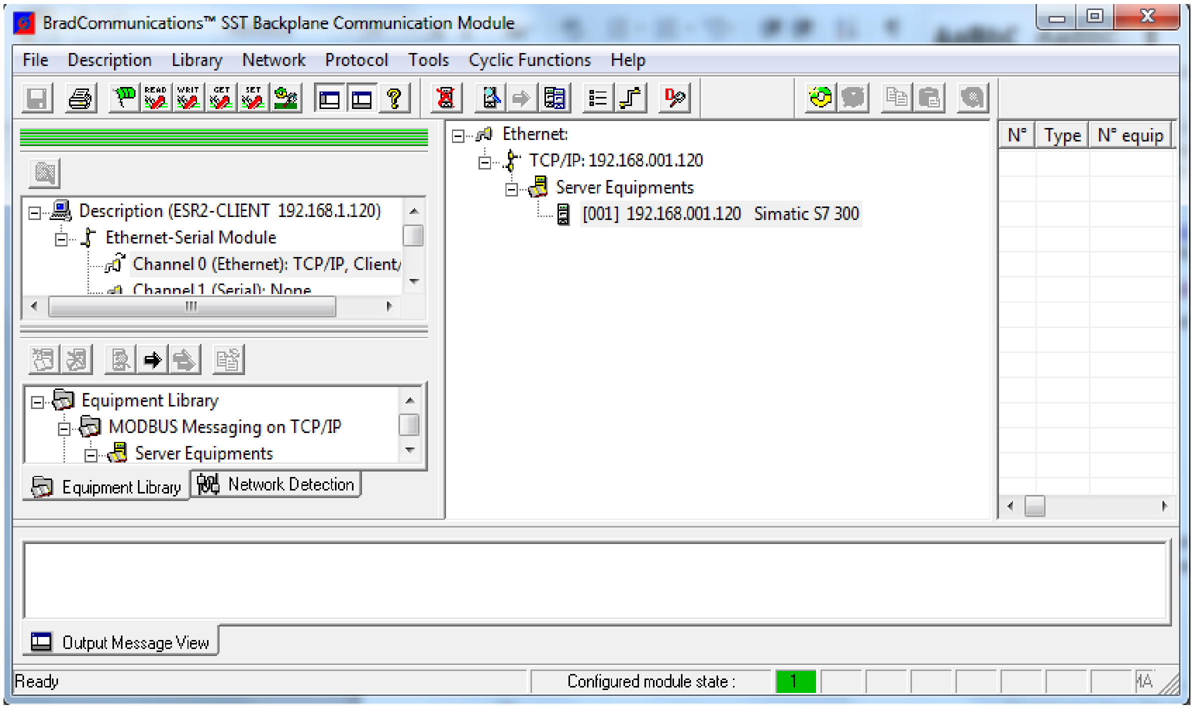
w = start address of the input table (0-29952)

v = start address of the output table (0-29952)

The firmware version should be 2.10.2 or higher to be able to use this feature. Follow the steps below to configure the module in extended addressing mode.

1. Complete procedures in Configure the Module as Siemens Client/Server Equipment

2. After completing the steps in Configure the Module as Siemens Client/Server Equipment section, the Console application main screen should be displayed with Port 0 (TCP/IP) configured as Client and Siemens S7-300 server equipment with IP address 192.168.1.120 is added.



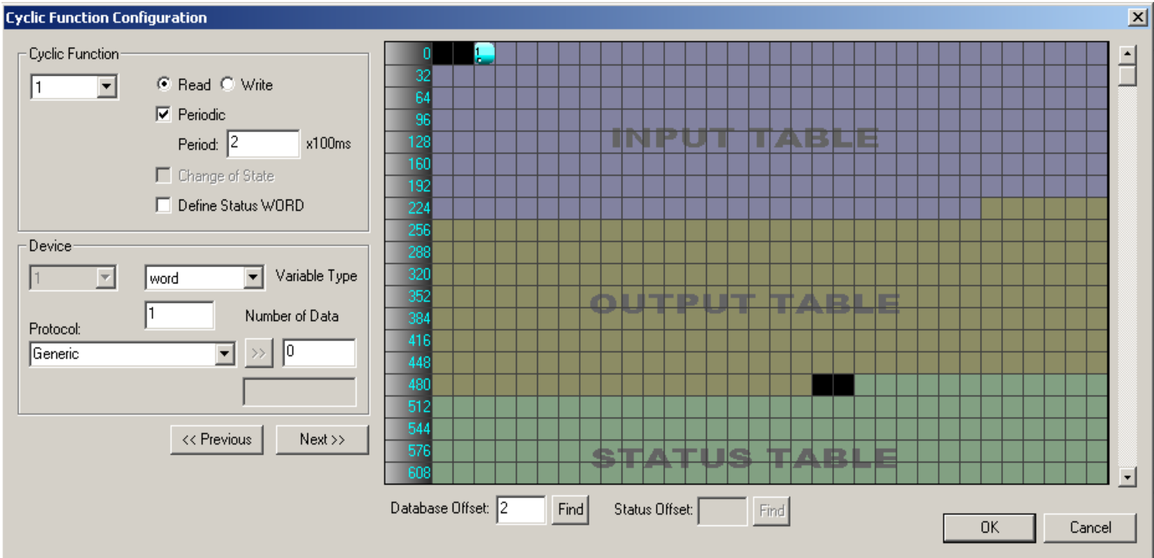
3. Complete the instruction in Changing to Extended Addressing Mode section.

4. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

5. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

6. Creating cyclic function by clicking . The cyclic function created will provide automatic data exchange between the client and server equipment. Creating both read and write cyclic function to automatically read and write data from/to the server equipment variable addresses. The function code as well as the size of the data to be read is part of the cyclic function parameters.

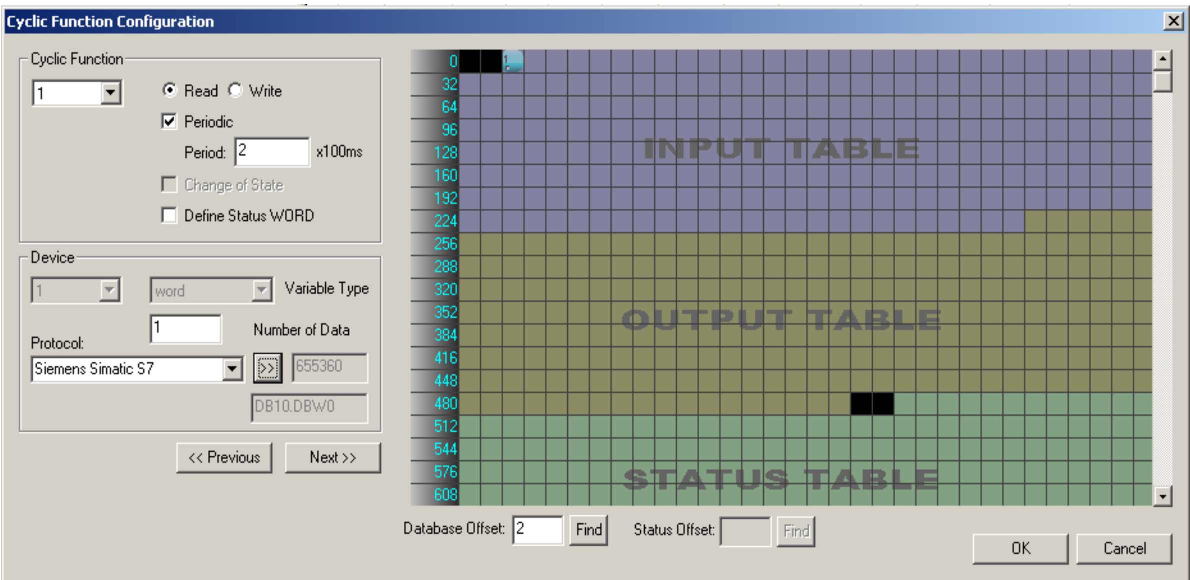
Figure 82: Cyclic Function Dialog Window



The detail of each parameter in discussed in Cyclic Function Configuration Parameters and Table 38.

a. Create cyclic function #1 with the parameter below.
   i. Use default cyclic function number provided (1 should be the default if this the first cyclic function)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Siemens Simatic S7
   vii. PLC Variable Syntax DB20.DW0 (server equipment address:5120)
   viii. Database offset is 0 (client equipment address)

After filling up the parameter the cyclic function will be similar to what is shown below.



b. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

c. Create cyclic function #2 with the parameter below

   i. Use default cyclic function number provided (2 should be the default now)
   ii. Select Read cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Siemens Simatic S7
   vii. PLC Variable Syntax DB21.DW0 (server equipment address:5376)
   viii. Database offset is 5000 (client equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.

The difference between the first and second cyclic function is the database offset location in the client.

1st cyclic function database address is located in the mapped IO table address 0-1599
2nd cyclic function database address is located outside of the mapped IO table address

e. Create cyclic function #3 with the parameters below:

   i. Use default cyclic function number provided (3 should be the default now)
   ii. Select Write cyclic function type
   iii. Check the Periodic setting
   iv. Use default period of 2 x 10ms = 20ms
   v. Number of data is 1
   vi. Protocol is Siemens Simatic S7

vii. PLC Variable Syntax DB22.DW0 (server equipment address:5632)
viii. Database offset is 1600 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view

g. Create cyclic function #4 with the parameter below

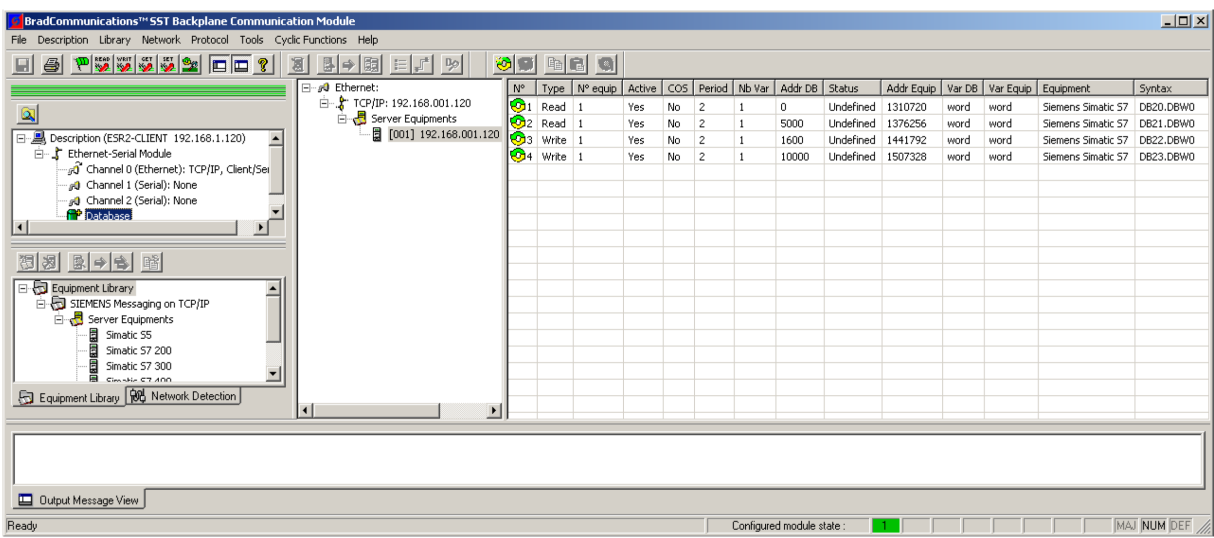
i. Use default cyclic function number provided (4 should be the default now)
ii. Select Write cyclic function type
iii. Check the Periodic setting
iv. Use default period of 2 x 10ms = 20ms
v. Number of data is 1
vi. Protocol is Siemens Simatic S7
vii. PLC Variable Syntax DB23.DW0 (server equipment address:5888)
viii. Database offset is 10000 (client equipment address)

h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

The difference between the first and second cyclic function is the database offset location in the client.

3rd cyclic function database address is located in the mapped IO table address 0-1599
4th cyclic function database address is located outside of the mapped IO table address

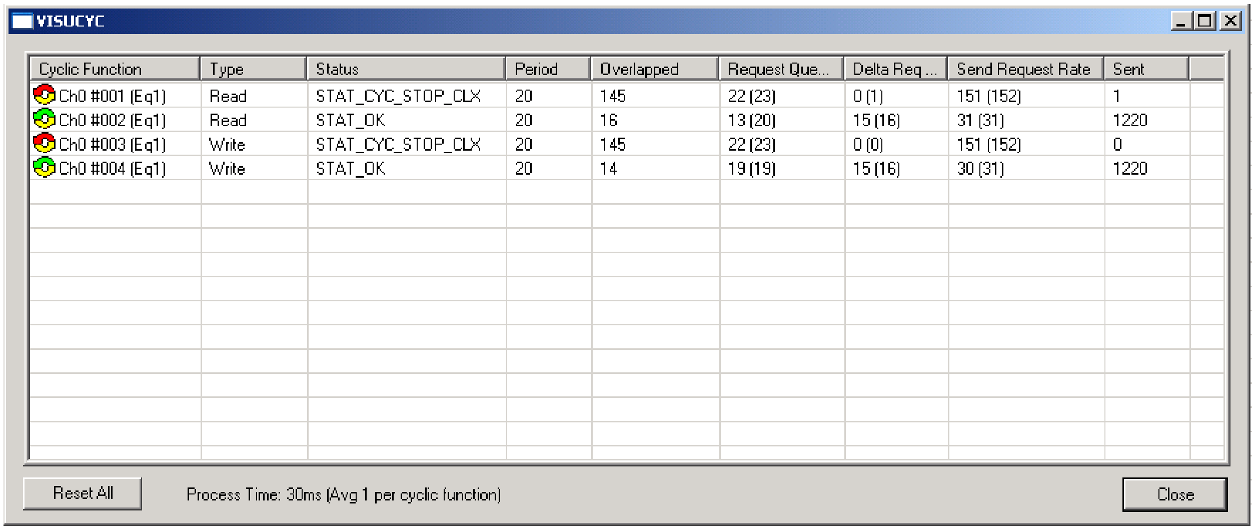
7. After creating all the cyclic function your screen should have a similar display as below.



8. Download the configuration to the module by clicking  and .

9. When download has completed, (Refer to Console Status Bar Information to check if module is properly initialized), run the visual cyclic function application by clicking from the main console screen.

   The screen display as below:



Cyclic Function	Type	Status	Period	Overlapped	Request Que...	Delta Req ...	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_CYC_STOP_CLX	20	145	22 (23)	0 (1)	151 (152)	1
Ch0 #002 (Eq1)	Read	STAT_OK	20	16	13 (20)	15 (16)	31 (31)	1220
Ch0 #003 (Eq1)	Write	STAT_CYC_STOP_CLX	20	145	22 (23)	0 (0)	151 (152)	0
Ch0 #004 (Eq1)	Write	STAT_OK	20	14	19 (19)	15 (16)	30 (31)	1220

*Both 1st and 3rd cyclic function status is 135 – **not active / not running***

*Both 2nd and 4th cyclic function status is 0 – **active / running***

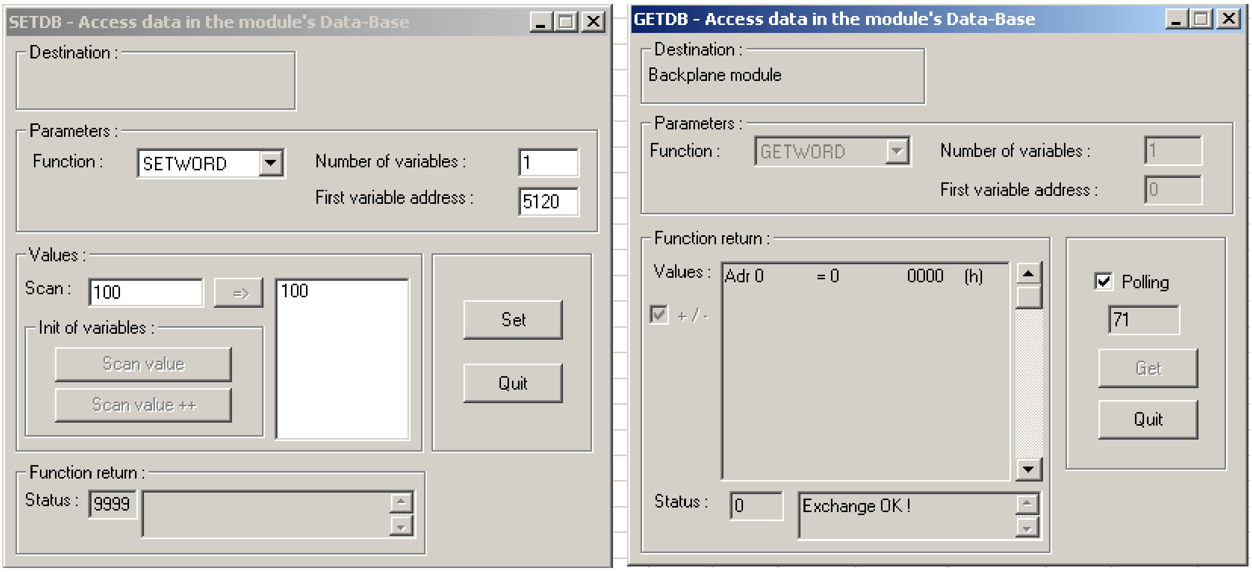
Note

The module will not run any cyclic function configured in the database address mapped to INPUT, OUTPUT and STATUS Table when there is no backplane connection.

To check if active cyclic function is running properly, follow this brief diagnostic:

a. Locate and run and application.

b. To test cyclic function # 1, fill in the values as shown below.

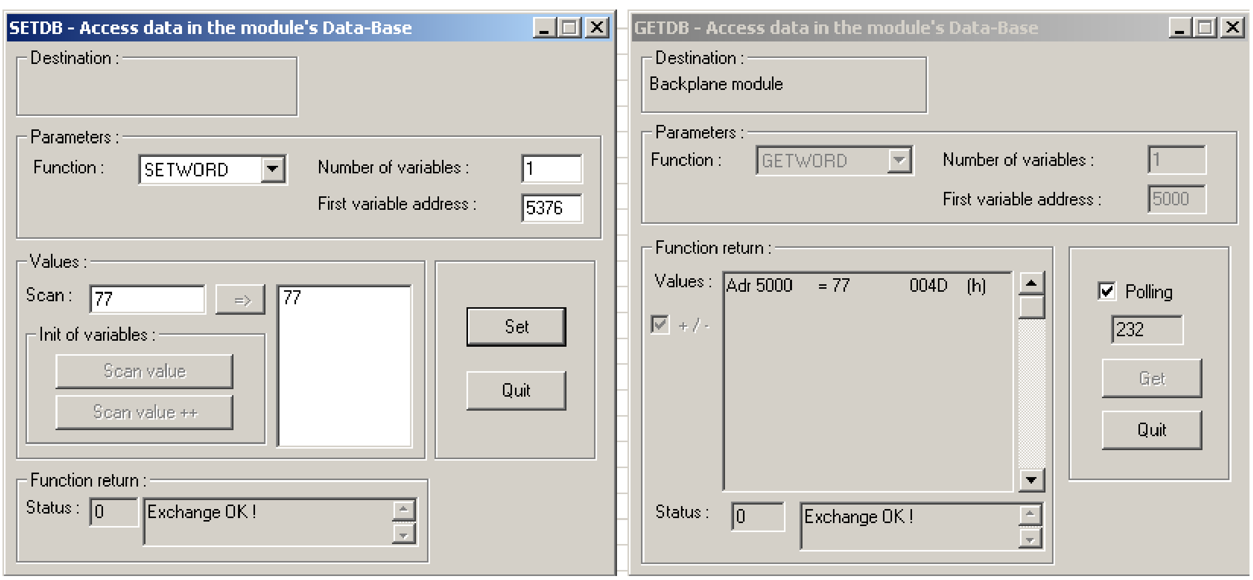


SetDB sets the server equipment address 5120 (DB20.DW0) with value 100.

GetDB reads the client equipment database address 0 with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active/not running***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 0.

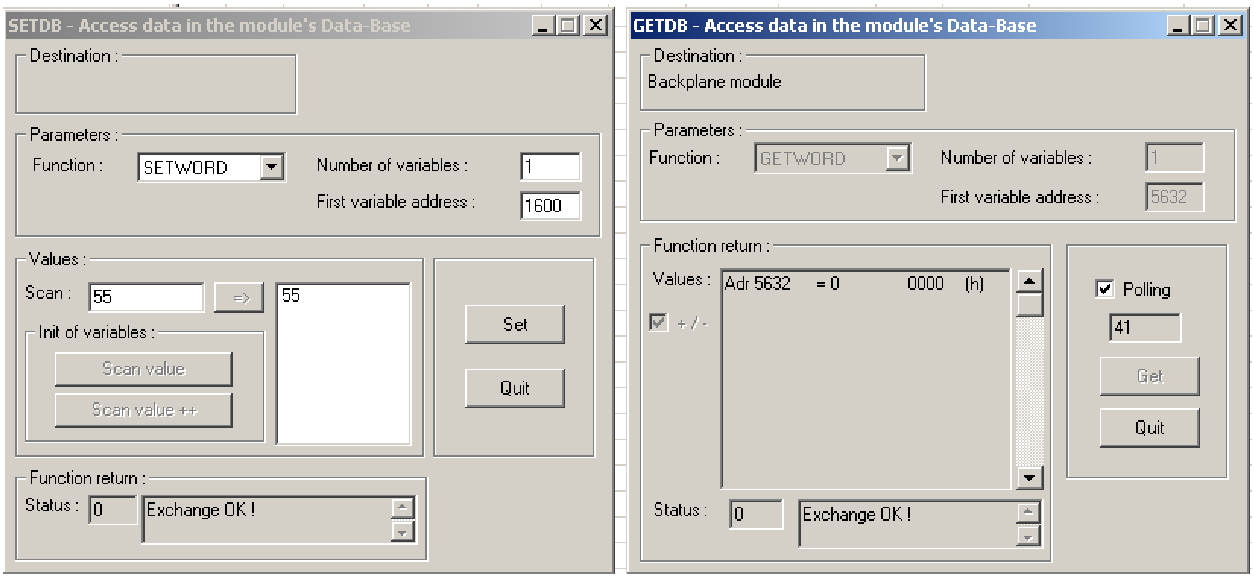
c. To test cyclic function # 2, fill in values as shown below.



SetDB sets the server equipment address 5376 (DB21.DW0) with value 77.

GetDB reads the client equipment database address 5000 with value shown as 77. If you are getting the data correctly your setup is working properly.

d. To test cyclic function # 3, fill in the values as shown below.

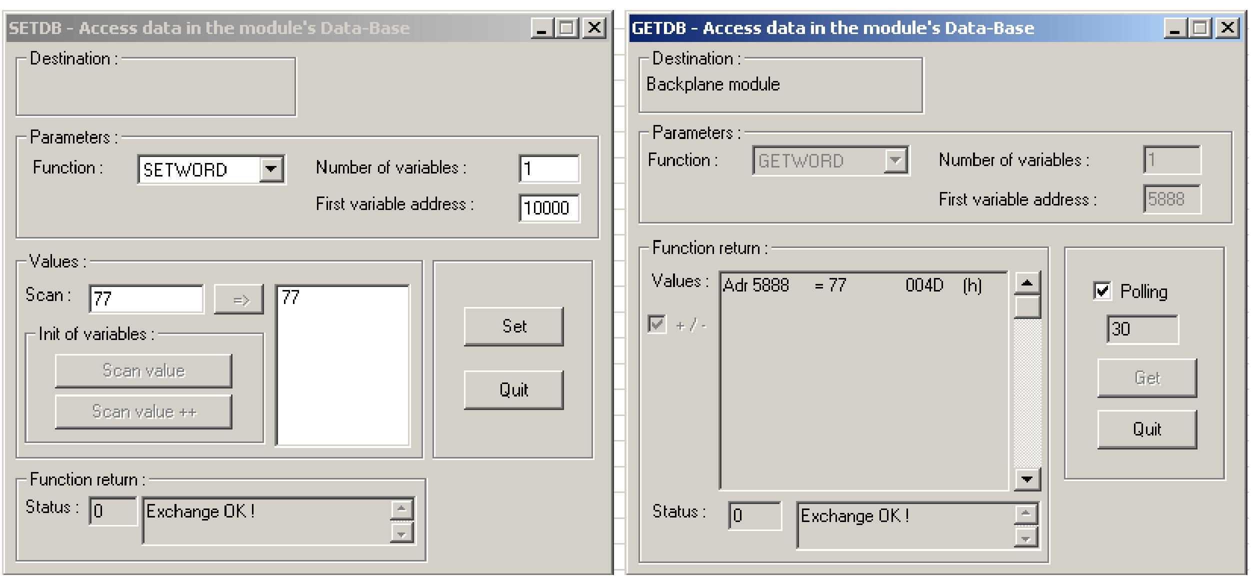


SetDB sets the client database address 1600 with value 55.

GetDB reads the server equipment database address 5632 (DB22.DW0) with value shown as 0.

As indicated in the cyclic function status from Visucyc application, cyclic function #1 is ***not active***, therefore ***GetDB*** reads 0 (initial value in the module's database) in address 5632.

e. To test cyclic function #4, fill in the values shown below.



SetDB sets the client database address 10000 with value 77.

GetDB reads the server equipment database address 5888 (DB23.D20) with value shown as 77.

In the next set of procedures, The module will communicate with the ControlLogix CPU. When backplane connection becomes active, all cyclic function with **"not active / not running"** state will switch to "**active/running**".

With Backplane Connection:

Multiple sample configurations with different parameters are provided.

Example 1: Transfer 1 word of data for each write and read cyclic function using the default mapping defined in the AOI.

1. If the previous set of procedures in **No Backplane Connection** is skipped, please go back and configure the same cyclic functions 1-4 and run all diagnostic successfully.

2. Follow instruction 1-19 in Using 1756 Generic Profile section.

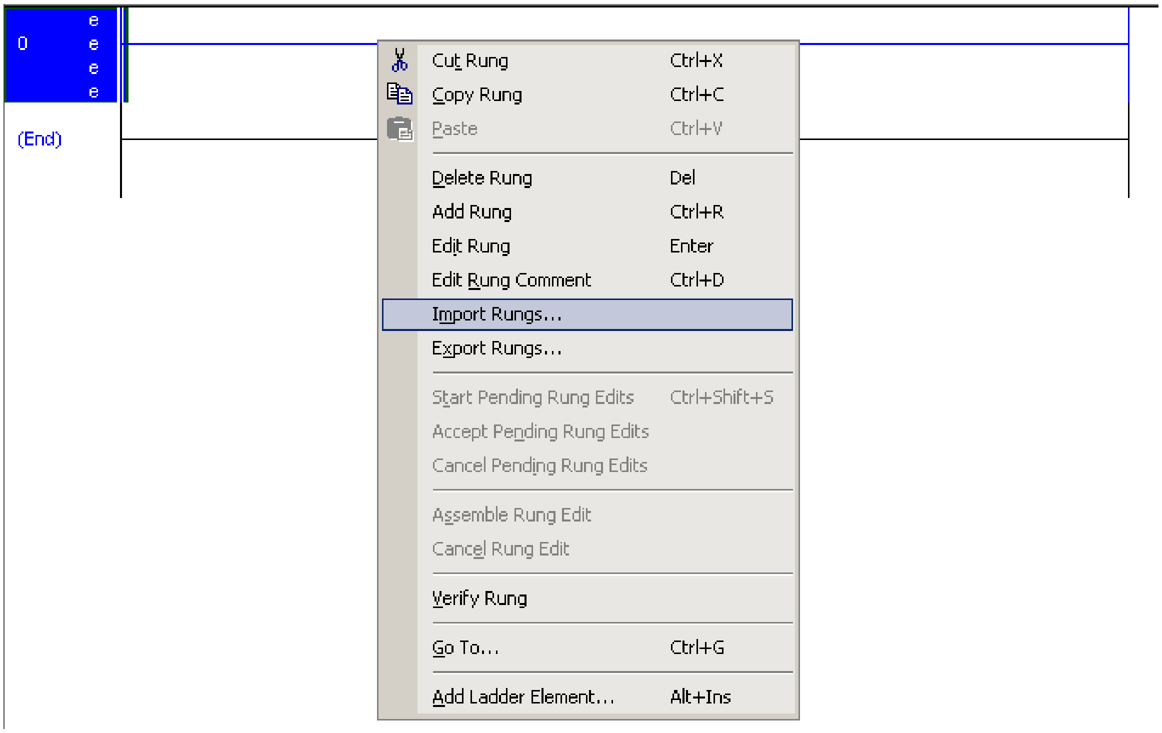
Note

For first time user of this feature, the console configuration should be done first prior to configuring ControlLogix connection because of the mapping of the database.

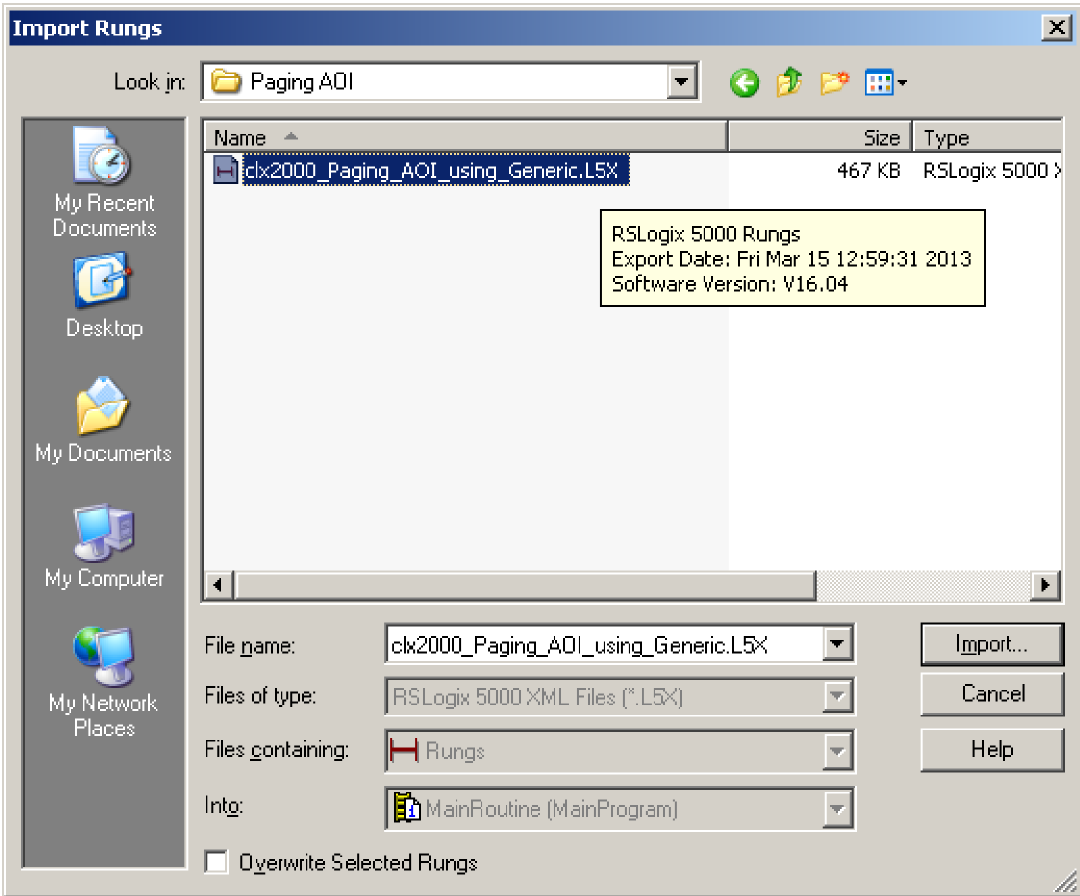
3. If smaller input and output sizes are required, please refer to section Changing the I/O connection size.

4. In RSLogix 5000 Task->MainTask, double click on MainRoutine. See below.

5. Select a rung as below and right-click on rung and select "Import Rung…"

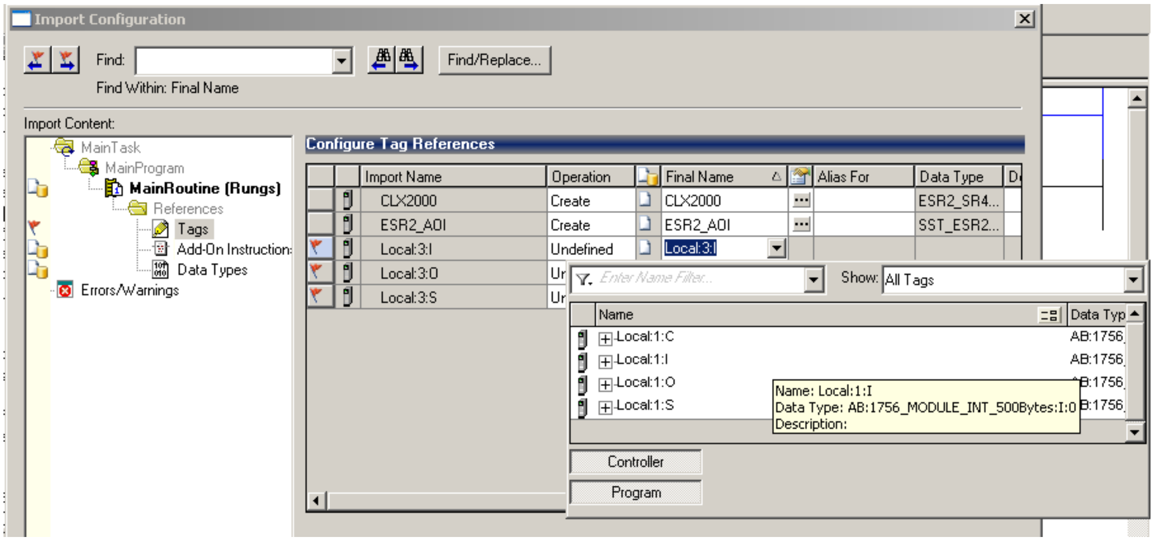

0
(End)
Cut Rung Ctrl+X
Copy Rung Ctrl+C
Paste Ctrl+V
Delete Rung Del
Add Rung Ctrl+R
Edit Rung Enter
Edit Rung Comment Ctrl+D
Import Rungs...
Export Rungs...
Start Pending Rung Edits Ctrl+Shift+S
Accept Pending Rung Edits
Cancel Pending Rung Edits
Assemble Rung Edit
Cancel Rung Edit
Verify Rung
Go To... Ctrl+G
Add Ladder Element... Alt+Ins

6. Browse for the ….L5X file as below and select it and select Import.... Refer to Module's Installation Directory Location section for the location of the AOI files.

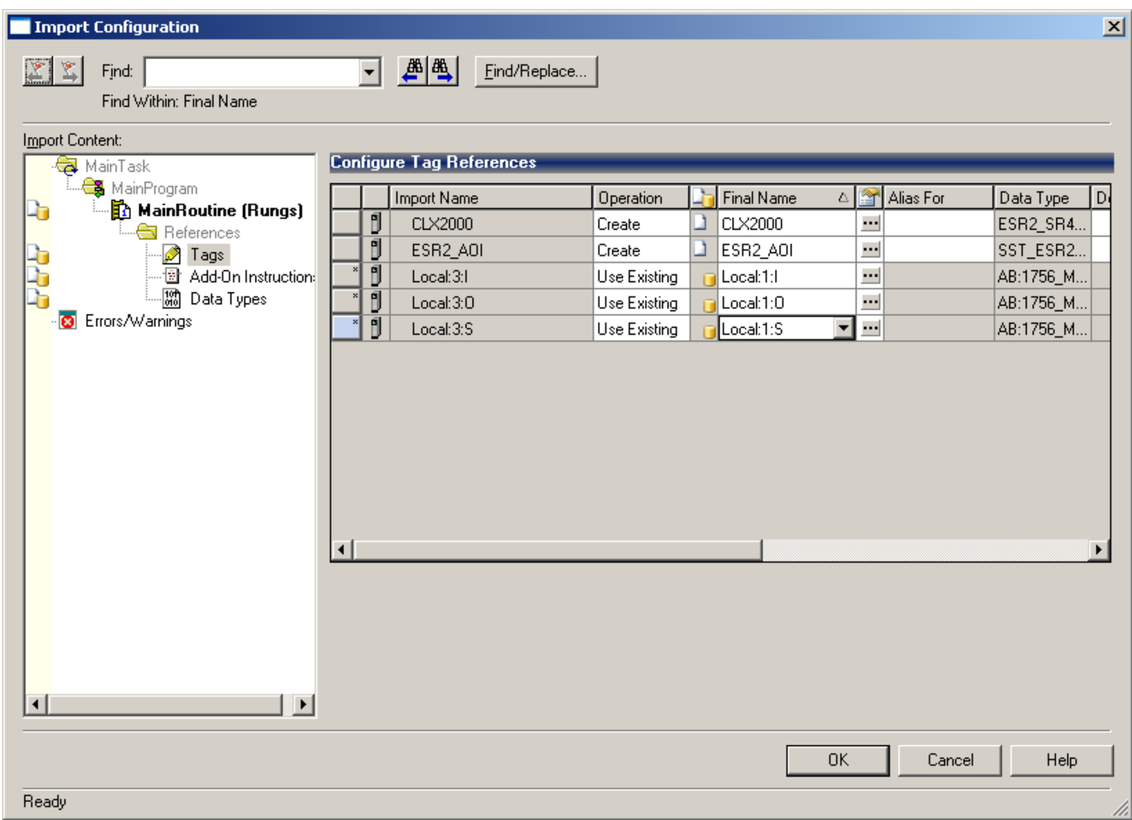


7. The import Configuration dialog box will appear as shown in below. For the tags that have X beside it, select the Input and Output tags for the module added in the I/O configuration.

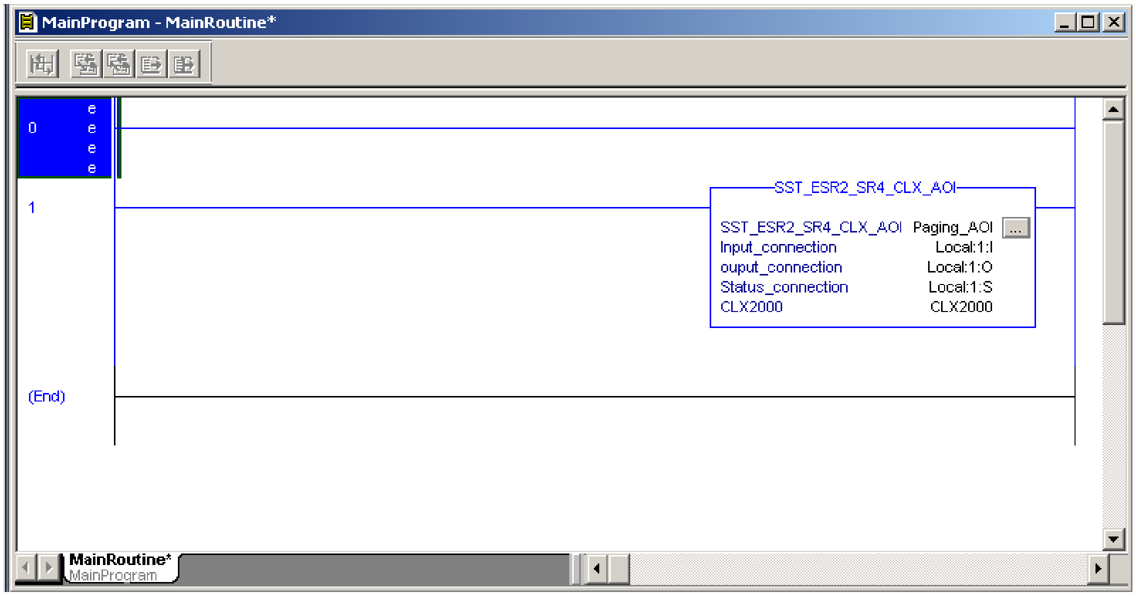
8. In this example the module was configured at slot 1 but the AOI was expecting slot 3. Update Tags Local:3:I, Local:3:O,Local:3:S to reference slot 1.



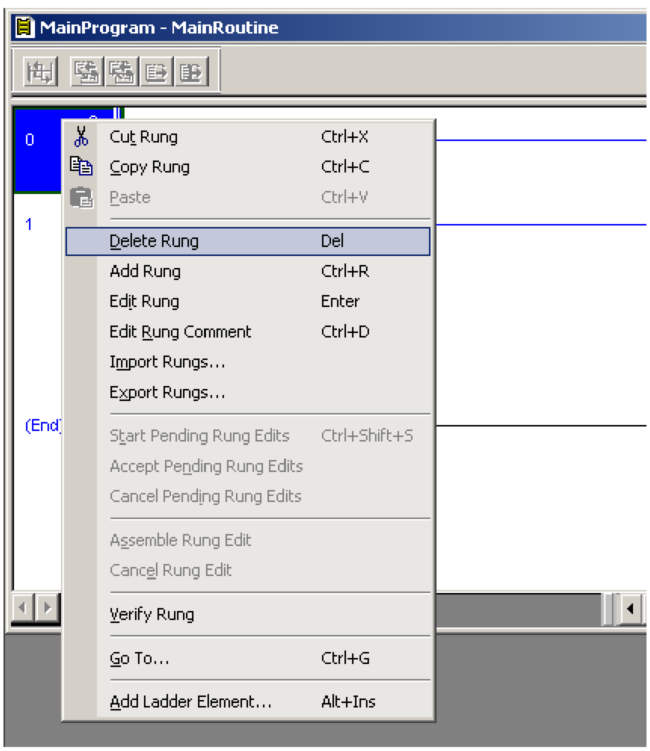
9. After correcting the last three tags click ok to begin the import.



10. After the import is complete, a similar display shown below.



11. Delete the empty rung as shown below.



12. Under Controller Tags, go to the CLX2000 tag. Expand CLX2000.CONFIGURATION as below.

INPUT Table Start Address	= 0
INPUT Table size	= 1600
OUTPUT Table Start Address	= 1600

OUTPUT Table Size	= 1600
STATUS Table Start Address	= 3200
STATUS Table Size	= 1024

The default mapped address is used in the AOI in this configuration.

Figure 83: AOI Controller Tags:

CLX2000	{...}	{...}		ESR2_SR4_MODULE
CLX2000.CONFIGURATION	{...}	{...}		DATABASE_CONFIG
CLX2000.CONFIGURATION.Input_Table_Start_Address	0		Decimal	INT
CLX2000.CONFIGURATION.Input_Table_Size	1600		Decimal	INT
CLX2000.CONFIGURATION.Output_Table_Start_Address	1600		Decimal	INT
CLX2000.CONFIGURATION.Output_Table_Size	1600		Decimal	INT
CLX2000.CONFIGURATION.Status_Table_Start_Address	3200		Decimal	INT
CLX2000.CONFIGURATION.Status_Table_Size	1024		Decimal	INT
CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
CLX2000.NUMBER_OF_BLKS	{...}	{...}		Pages_Control
CLX2000.NUMBER_OF_BLKS.Number_Input_Blocks	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Number_Output_Blocks	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Number_Status_Blocks	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Input_Remainder	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Output_Remainder	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Status_Remainder	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Input_Connection_Size	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Output_Connection_Size	0		Decimal	INT
CLX2000.NUMBER_OF_BLKS.Status_Connection_Size	0		Decimal	INT
CLX2000.CONFIGURATION_STATUS	0		Decimal	INT
CLX2000.VALID_CONFIGURATION	0		Decimal	BOOL

13. INPUT Table address and size, OUTPUT Table address and size and STATUS Table address and size should match with the database settings for INPUT, OUTPUT and STATUS in console configuration. To check that it match, in Console Configuration Description Area, locate and double click . Change to the values shown above if it's different. (Refer to Changing to Extended Addressing Mode) for more details.

14. To save and download configuration, click and . The configuration download takes about 2-3 minutes.

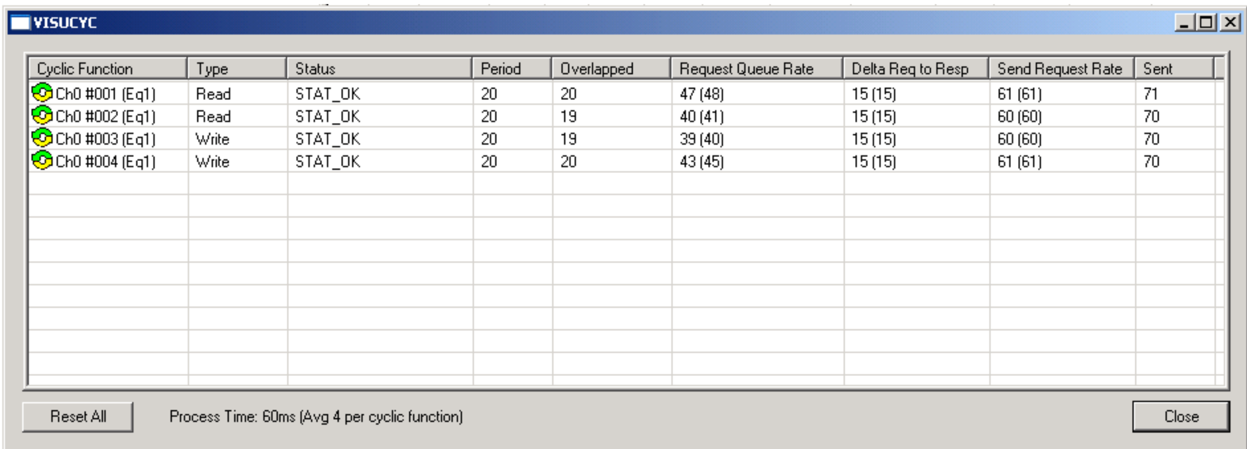
Note

The AOI instruction will not run if there is a mismatch between the CONFIGURATION tags and the database settings in console configuration.

15. To see the input, output and status tags, expand the CLX2000.DATABASE_DATA as shown in Figure 83 . These are the tags the AOI will use.

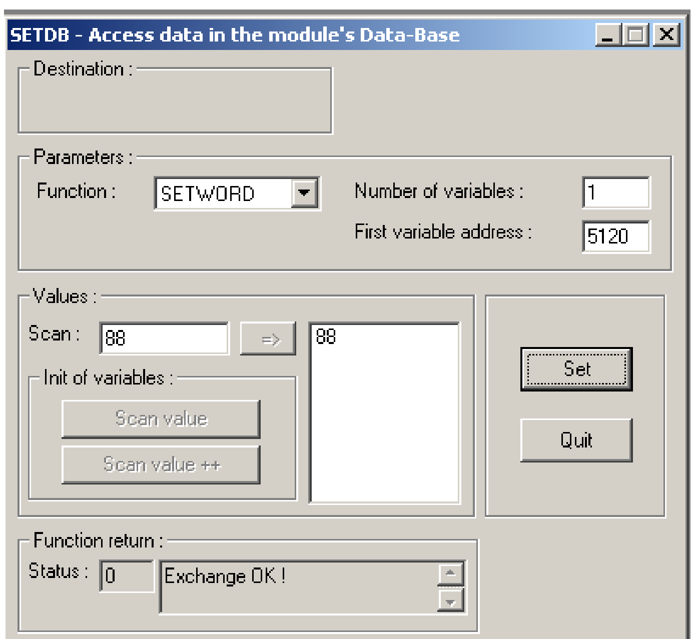
16. Save the configuration file and download it to the CLX Module.

17. From the Console Application window, locate and launch VisuCyc and all the cyclic functions are in active status. See below.



18. Test cyclic function 1 & 3 to check if values written and read from the mapped database address are exchanged properly.

19. To test cyclic function #1, fill in the values as shown below.



SetDB sets the server equipment address 5120 (DB20.DW0) with value 88.

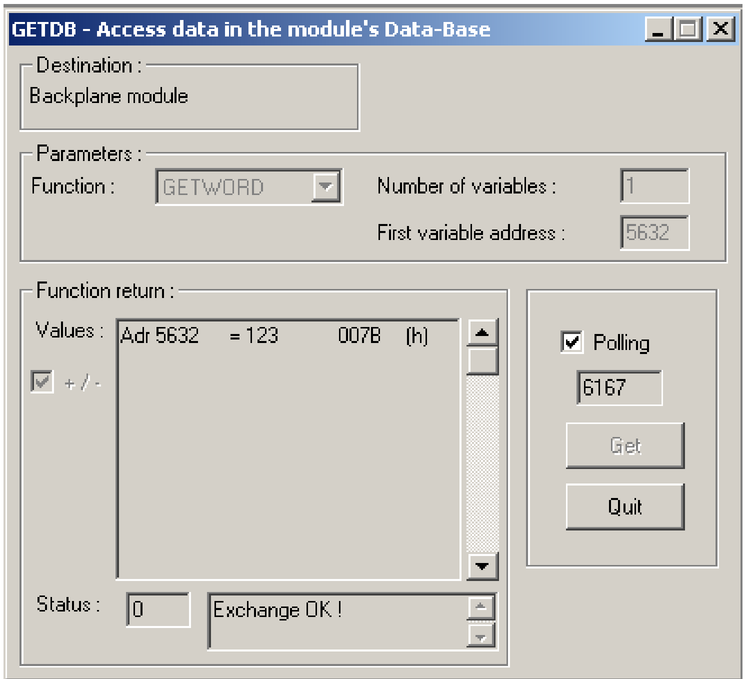
20. In RSLogix 5000 Software, expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value 88 which is set in the server equipment address 5120 using SetDB appears in table. See below.

⊟-CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
⊟-CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[0]	88		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[1]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[2]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[3]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[4]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[5]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[6]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[7]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[8]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[9]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[10]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.INPUT_DATA[11]	0		Decimal	INT

21. To test cyclic function #3, expand CLX2000.DATABASE_DATA.OUTPUT_DATA and write 123 into offset 0 (offset 1600 in database)

⊟-CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
⊞-CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊟-CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[0]	123		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[1]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[2]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[3]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[4]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[5]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[6]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[7]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[8]	0		Decimal	INT
⊞-CLX2000.DATABASE_DATA.OUTPUT_DATA[9]	0		Decimal	INT

Run ***GetDB*** and read the values as offset 5632 (DB22.DW0).



22. Now start adding the test of the desired configuration.

Example 2: Transfer 100 words of data per cyclic function. To create 5 write cyclic functions and 5 read cyclic functions using the default mapping defined in the AOI.

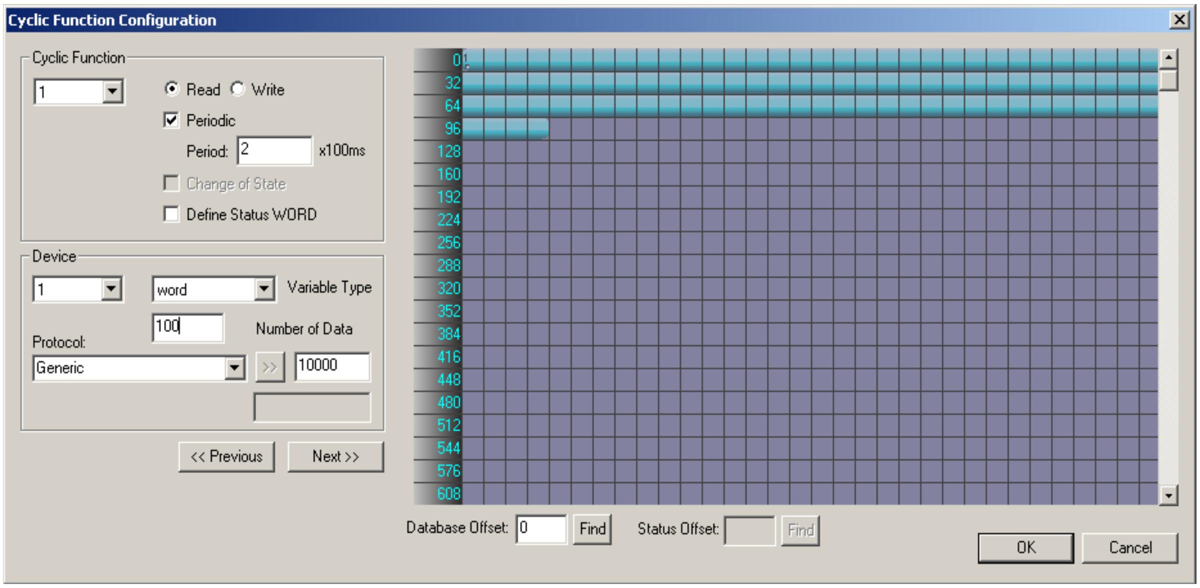
1. It is assume that Example 2 is a continuation of Example 1 therefore all console configurations, backplane connection and the quick test are done successfully.

2. From the console application window:

   a. Modify cyclic function #1 with the parameter below:

      ix. default cyclic function number provided
      x. Select Read cyclic function type
      xi. Check the Periodic setting
      xii. Use default period of 2 x 10ms = 20ms
      xiii. Number of data is 10
      xiv. Protocol is Siemens Simatic S7
      xv. Device First variable address is DB20.DW0 (server equipment address:5120)
      xvi. Database offset is 0 (client equipment address)

   After filling up the parameter the cyclic function will be similar to what is shown below.



   b. Click "OK", and the first cyclic function will be shown in the cyclic function list view

   c. Modify cyclic function #2 with the parameter below:

      xvii. Use default cyclic function number provided
      xviii. Select Read cyclic function type
      xix. Check the Periodic setting
      xx. Use default period of 2 x 10ms = 20ms
      xxi. Number of data is 10
      xxii. Protocol is Siemens Simatic S7

xxiii. Device First variable address is DB21.DW0 (server equipment address)
xxiv. Database offset is 100 (client equipment address)

d. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.
e. Modify cyclic function #3 with the parameters below:
   xvii. Use default cyclic function number provided
   xviii. Select Write cyclic function type
   xix. Check the Periodic setting
   xx. Use default period of 2 x 10ms = 20ms
   xxi. Number of data is 10
   xxii. Protocol is Siemens Simatic S7
   xxiii. Device First variable address is DB22.DW0 (server equipment address:5632)
   xxiv. Database offset is 1600 (client equipment address)

f. Click “OK”, and the third cyclic function will be shown in the cyclic function list view
g. Modify cyclic function #4 with the parameter below
   xvii. Use default cyclic function number provided
   xviii. Select Write cyclic function type
   xix. Check the Periodic setting
   xx. Use default period of 2 x 10ms = 20ms
   xxi. Number of data is 10
   xxii. Protocol is Siemens Simatic S7
   xxiii. Device First variable address is DB23.DW0 (server equipment address:5888)
   xxiv. Database offset is 1700 (client equipment address)

h. Click “OK”, and the fourth cyclic function will be shown in the cyclic function list view

3. Add 6 more cyclic functions to this configuration.

   w. Create cyclic function #5 with the parameter below.
      xvii. default cyclic function number provided
      xviii. 306lect Read cyclic function type
      xix. Check the Periodic setting
      xx. Use default period of 2 x 10ms = 20ms
      xxi. Number of data is 10
      xxii. Protocol is Siemens Simatic S7
      xxiii. Device First variable address is EW50 (server equipment address:1650)
      xxiv. Database offset is 1000(client equipment address)

   x. Click “OK”, and the first cyclic function will be shown in the cyclic function list view

y. Create cyclic function #6 with the parameter below
   xvii. Use default cyclic function number provided
   xviii. Select Read cyclic function type
   xix. Check the Periodic setting
   xx. Use default period of 2 x 10ms = 20ms
   xxi. Number of data is 10
   xxii. Protocol is Siemens Simatic S7
   xxiii. Device First variable address is MW50 (server equipment address:1906)
   xxiv. Database offset is 1100 (client equipment address)

z. Click “OK”, and the second cyclic function will be shown in the cyclic function list view.

aa. Create cyclic function #7 with the parameters below:
   xvii. Use default cyclic function number provided
   xviii. Select Read cyclic function type
   xix. Check the Periodic setting
   xx. Use default period of 2 x 10ms = 20ms
   xxi. Number of data is 10
   xxii. Protocol is Siemens Simatic S7
   xxiii. Device First variable address is AW50 (server equipment address:50)
   xxiv. Database offset is 1200 (client equipment address)

bb. Click “OK”, and the third cyclic function will be shown in the cyclic function list view

cc. Create cyclic function #8 with the parameter below
   xix. Use default cyclic function number provided
   xx. Select Write cyclic function type
   xxi. Check the Periodic setting
   xxii. Use default period of 2 x 10ms = 20ms
   xxiii. Number of data is 10
   xxiv. Protocol is Siemens Simatic S7
   xxv. Device First variable address is AW100 (server equipment address:100)
   xxvi. Database offset is 1800 (client equipment address)

dd. Click “OK”, and the third cyclic function will be shown in the cyclic function list view

ee. Create cyclic function #9 with the parameter below
   xv. Use default cyclic function number provided
   xvi. Select Write cyclic function type
   xvii. Check the Periodic setting
   xviii. Use default period of 2 x 10ms = 20ms
   xix. Number of data is 10
   xx. Protocol is Siemens Simatic S7

xxvii. Device First variable address is MW150 (server equipment address:278)

xxi. Database offset is 1900 (client equipment address)

ff. Click “OK”, and the third cyclic function will be shown in the cyclic function list view

gg. Create cyclic function #10 with the parameter below

xvii. Use default cyclic function number provided

xviii. Select Write cyclic function type

xix. Check the Periodic setting

xx. Use default period of 2 x 10ms = 20ms

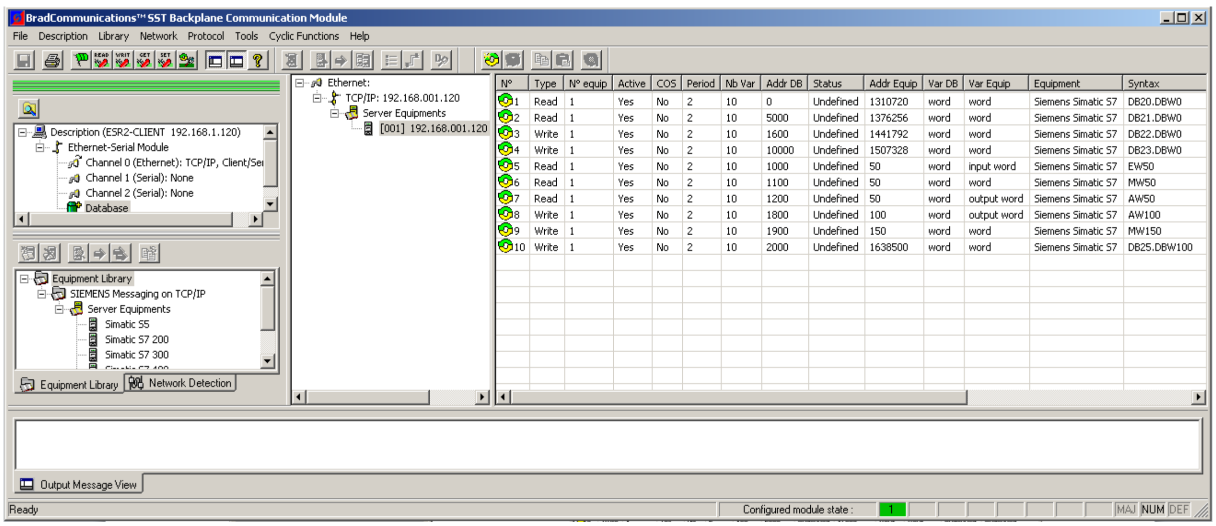
xxi. Number of data is 10

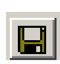
xxii. Protocol is Siemens Simatic S7

xxiii. Device First variable address is DB25.DW100 (server equipment address:6500)

xxiv. Database offset is 2000 (client equipment address)

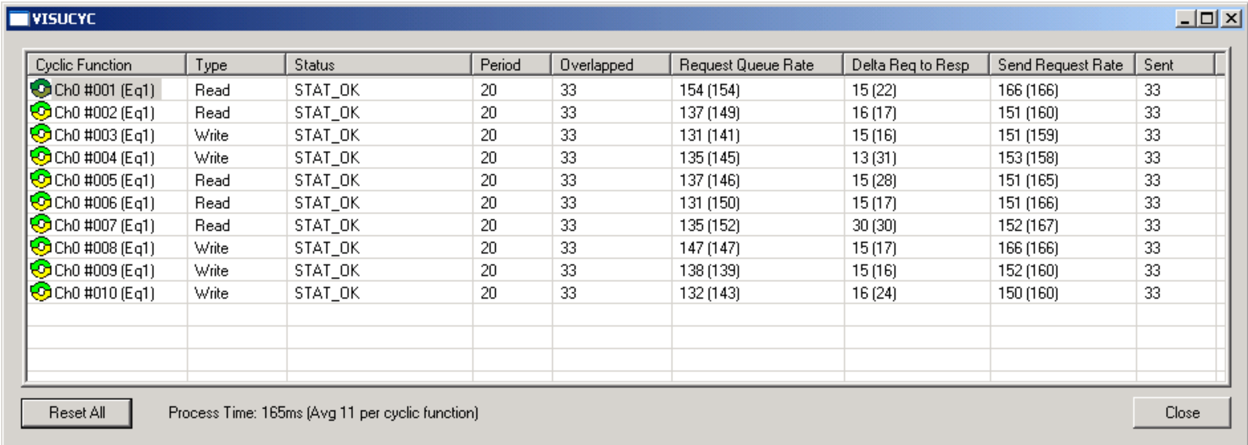
4. After all the cyclic functions are created, the similar display is shown as below.



5. To save and download configuration, click  and . The configuration download takes about 2-3 minutes.

6. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to Console Status Bar Information for status information on the communication to the module.

7. If initialization is successful, run the Visucyc application to check the status of each cyclic function. The similar display is shown as below.


VISUCYC

Cyclic Function	Type	Status	Period	Overlapped	Request Queue Rate	Delta Req to Resp	Send Request Rate	Sent
Ch0 #001 (Eq1)	Read	STAT_OK	20	33	154 (154)	15 (22)	166 (166)	33
Ch0 #002 (Eq1)	Read	STAT_OK	20	33	137 (149)	16 (17)	151 (160)	33
Ch0 #003 (Eq1)	Write	STAT_OK	20	33	131 (141)	15 (16)	151 (159)	33
Ch0 #004 (Eq1)	Write	STAT_OK	20	33	135 (145)	13 (31)	153 (158)	33
Ch0 #005 (Eq1)	Read	STAT_OK	20	33	137 (146)	15 (28)	151 (165)	33
Ch0 #006 (Eq1)	Read	STAT_OK	20	33	131 (150)	15 (17)	151 (166)	33
Ch0 #007 (Eq1)	Read	STAT_OK	20	33	135 (152)	30 (30)	152 (167)	33
Ch0 #008 (Eq1)	Write	STAT_OK	20	33	147 (147)	15 (17)	166 (166)	33
Ch0 #009 (Eq1)	Write	STAT_OK	20	33	138 (139)	15 (16)	152 (160)	33
Ch0 #010 (Eq1)	Write	STAT_OK	20	33	132 (143)	16 (24)	150 (160)	33

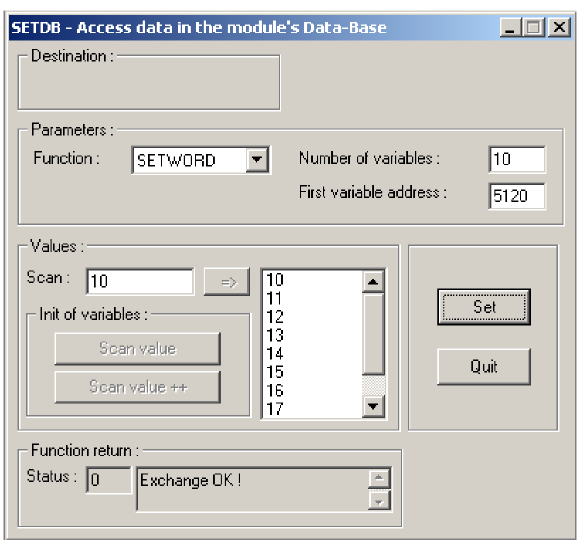
Reset All | Process Time: 165ms (Avg 11 per cyclic function) | Close

8. Since the active communication with the backplane, all cyclic functions are in “**active**” state.

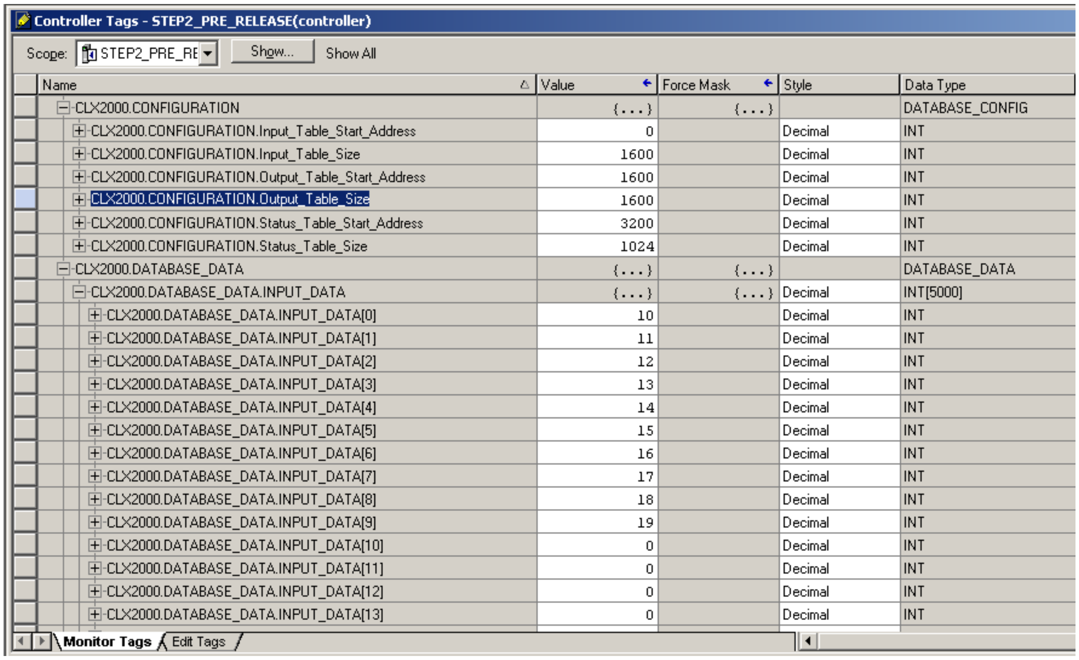
Note

In this example, all cyclic functions are located in the mapped portion of the database.

9. Nnow start testing the cyclic function.
   a. Cyclic Function # 1:
      i. From the Console application window, locate and launch SetDB..
      ii. Fill in the parameters as shown below.



      iii. In RSLogix 5000 Software, expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value which is set in the server equipment address 5120 (DB20.DW0) using SetDB appears in the tableSee below.



Controller Tags - STEP2_PRE_RELEASE(controller)

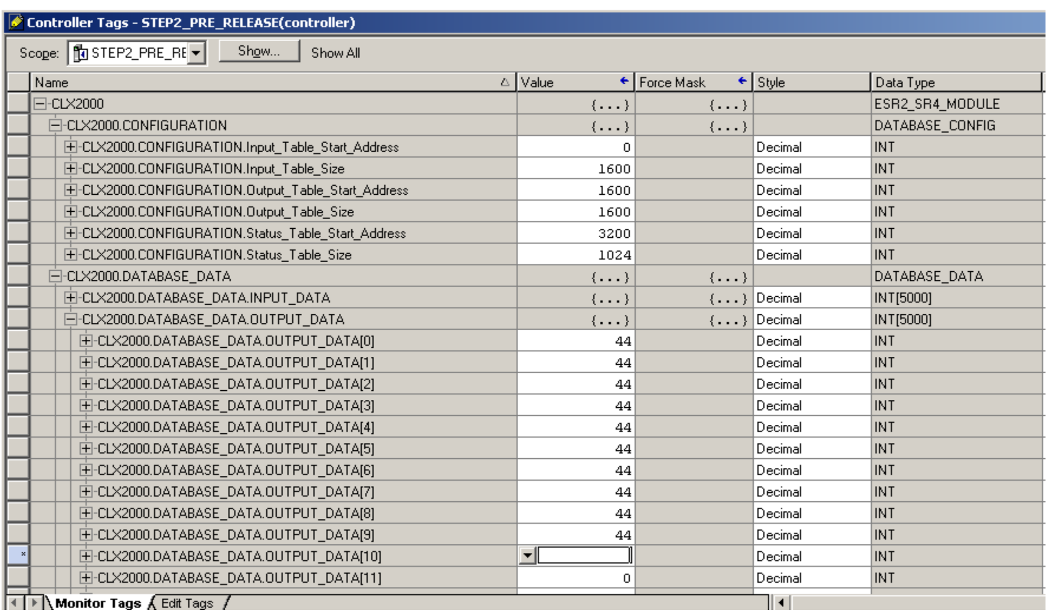
Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.CONFIGURATION	{...}	{...}		DATABASE_CONFIG
CLX2000.CONFIGURATION.Input_Table_Start_Address	0		Decimal	INT
CLX2000.CONFIGURATION.Input_Table_Size	1600		Decimal	INT
CLX2000.CONFIGURATION.Output_Table_Start_Address	1600		Decimal	INT
CLX2000.CONFIGURATION.Output_Table_Size	1600		Decimal	INT
CLX2000.CONFIGURATION.Status_Table_Start_Address	3200		Decimal	INT
CLX2000.CONFIGURATION.Status_Table_Size	1024		Decimal	INT
CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.INPUT_DATA[0]	10		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1]	11		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[2]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[3]	13		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[4]	14		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[5]	15		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[6]	16		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[7]	17		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[8]	18		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[9]	19		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[10]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[11]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[12]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[13]	0		Decimal	INT

Monitor Tags / Edit Tags

b. Cyclic Function # 3:

i. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.



Controller Tags - STEP2_PRE_RELEASE(controller)

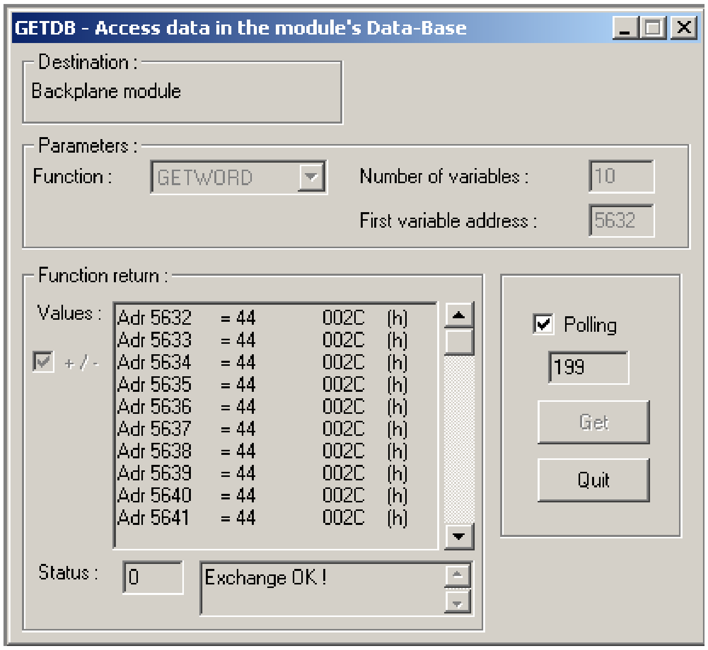
Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000	{...}	{...}		ESR2_SR4_MODULE
CLX2000.CONFIGURATION	{...}	{...}		DATABASE_CONFIG
CLX2000.CONFIGURATION.Input_Table_Start_Address	0		Decimal	INT
CLX2000.CONFIGURATION.Input_Table_Size	1600		Decimal	INT
CLX2000.CONFIGURATION.Output_Table_Start_Address	1600		Decimal	INT
CLX2000.CONFIGURATION.Output_Table_Size	1600		Decimal	INT
CLX2000.CONFIGURATION.Status_Table_Start_Address	3200		Decimal	INT
CLX2000.CONFIGURATION.Status_Table_Size	1024		Decimal	INT
CLX2000.DATABASE_DATA	{...}	{...}		DATABASE_DATA
CLX2000.DATABASE_DATA.INPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.OUTPUT_DATA	{...}	{...}	Decimal	INT[5000]
CLX2000.DATABASE_DATA.OUTPUT_DATA[0]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[1]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[2]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[3]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[4]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[5]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[6]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[7]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[8]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[9]	44		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[10]			Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[11]	0		Decimal	INT

Monitor Tags / Edit Tags

ii. From the Console application window, locate and launch GetDB. .

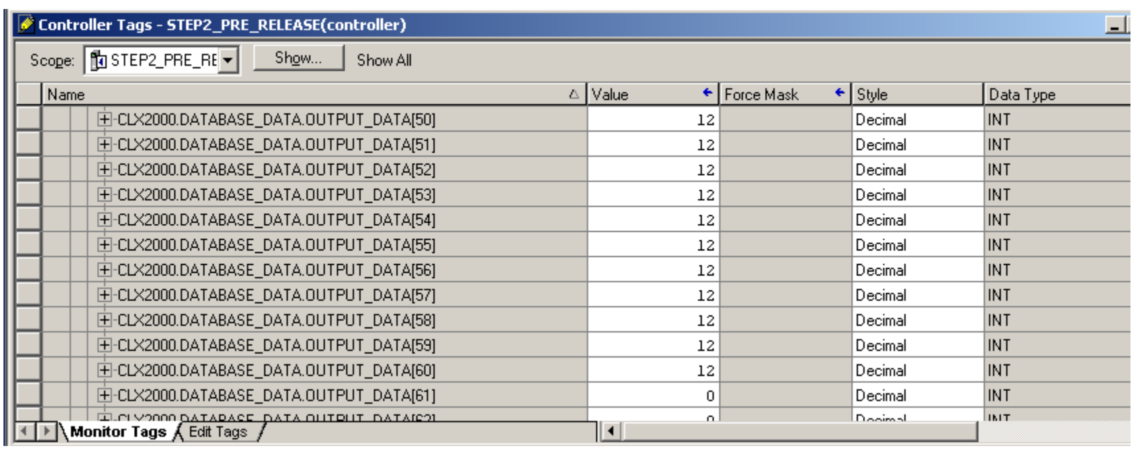
iii. Fill in the parameters as shown below.



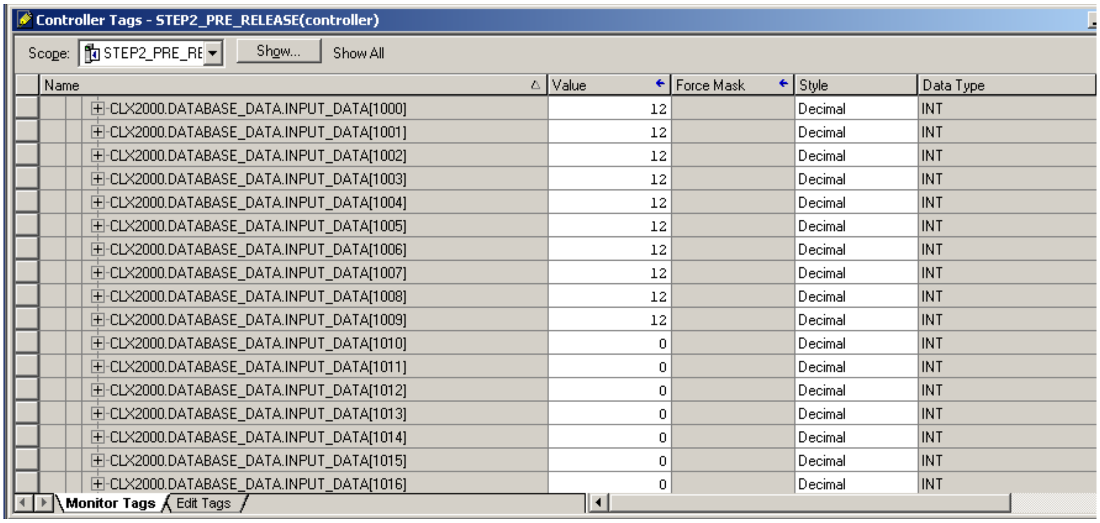
c. Cyclic Function # 5:

i. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

ii. The cylic function is configured to read EW50 and write to 1000, EW0-255 is mapped in the first 256 Words of the Output table.



iii. Expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value which is set in the Output table offset 50 as shown below.


Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

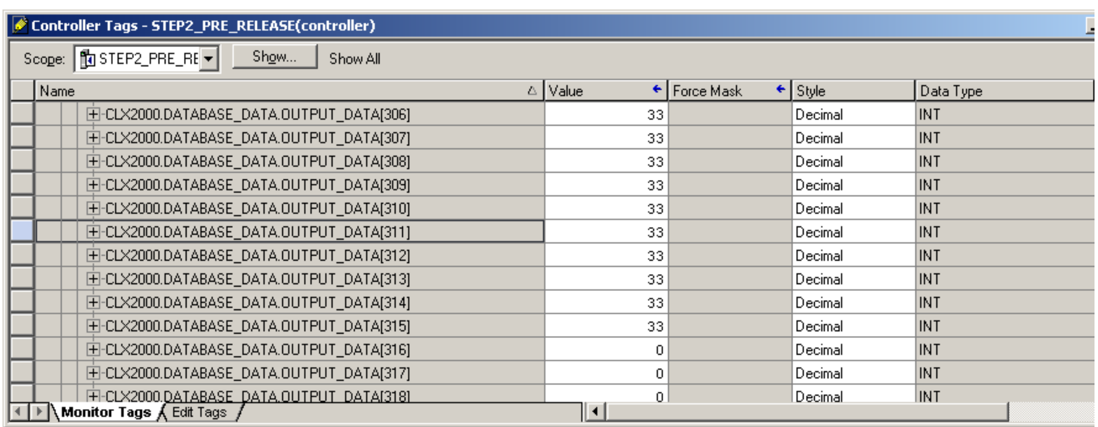
Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.INPUT_DATA[1000]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1001]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1002]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1003]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1004]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1005]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1006]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1007]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1008]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1009]	12		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1010]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1011]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1012]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1013]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1014]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1015]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1016]	0		Decimal	INT

Monitor Tags / Edit Tags

d. Cyclic Function # 6:

   i. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

   ii. The cylic function is configured to read MW50 and write to 1100, MW0-127 is mapped to the Output Table after the EW.

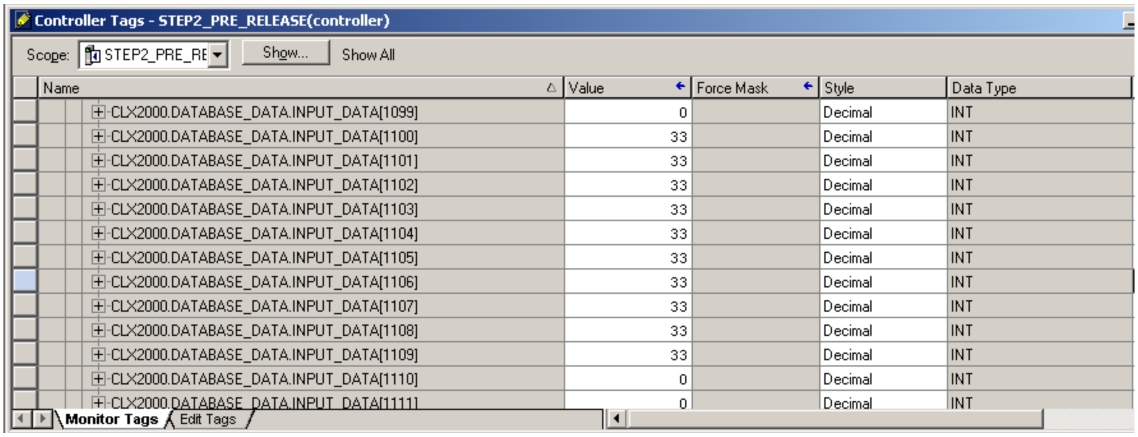

Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[306]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[307]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[308]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[309]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[310]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[311]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[312]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[313]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[314]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[315]	33		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[316]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[317]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[318]	0		Decimal	INT

Monitor Tags / Edit Tags

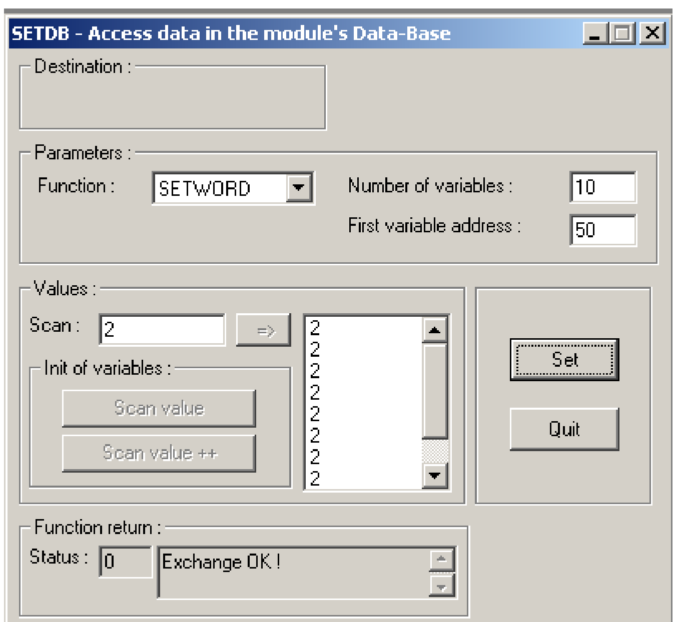
   iii. Expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value which is set in the Output table offset 306 as shown below.

25 

e. Cyclic Function # 7:

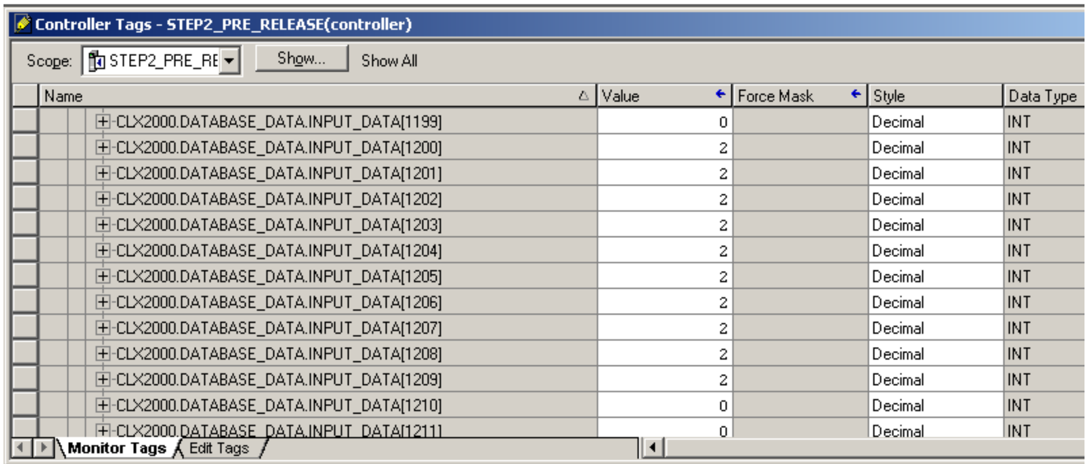
i. From the Console application window, locate and launch SetDB. .

ii. Fill in the parameters as shown below.



iv. The cyclic function is configured to read data from AW50 and write to 1200.

v. Expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value which is set in database offset 50. AW0-A255 is mapped to the first 256 words of the Input Table.



Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

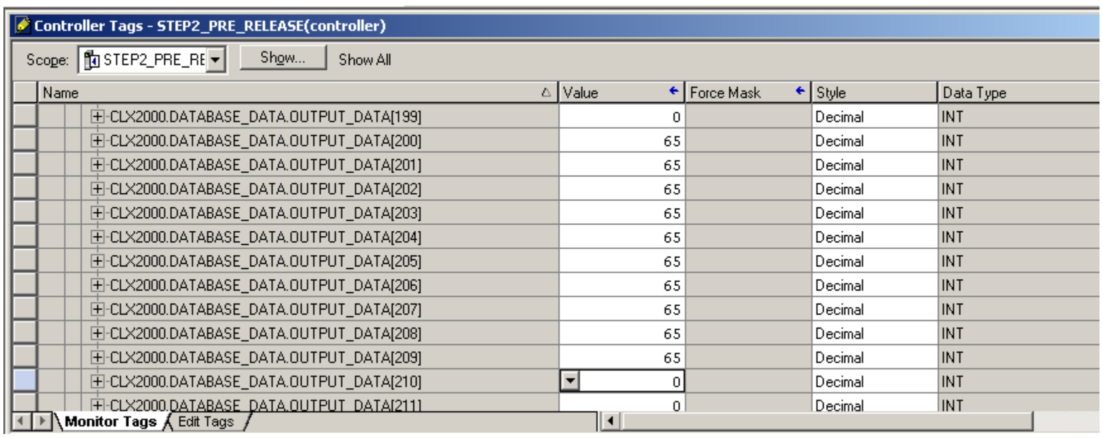
Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.INPUT_DATA[1199]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1200]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1201]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1202]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1203]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1204]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1205]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1206]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1207]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1208]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1209]	2		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1210]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[1211]	0		Decimal	INT

Monitor Tags / Edit Tags

f. Cyclic Function # 8:

i. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

ii. The cyclic function is configured to write data from database address 1800 to AW100.



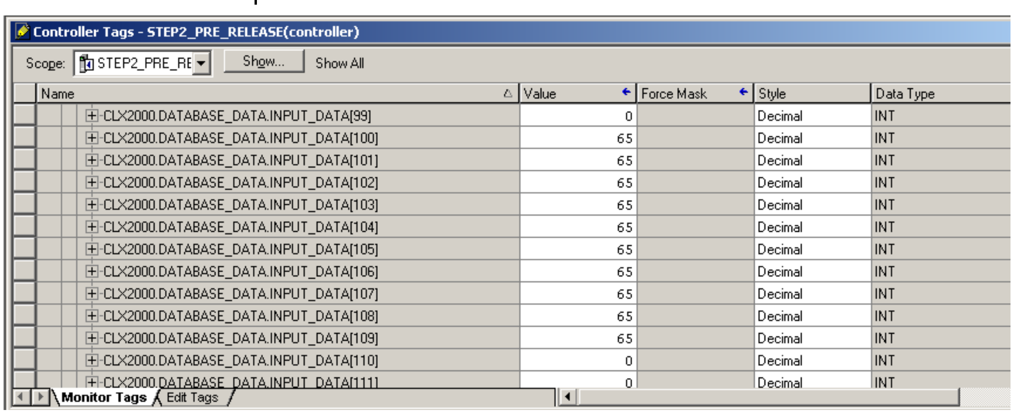
Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[199]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[200]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[201]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[202]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[203]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[204]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[205]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[206]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[207]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[208]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[209]	65		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[210]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[211]	0		Decimal	INT

Monitor Tags / Edit Tags

iii. Expand the CLX2000.DATABASE_DATA.INPUT_DATA, the value which is set in Output table 200 to AW100.



Controller Tags - STEP2_PRE_RELEASE(controller)

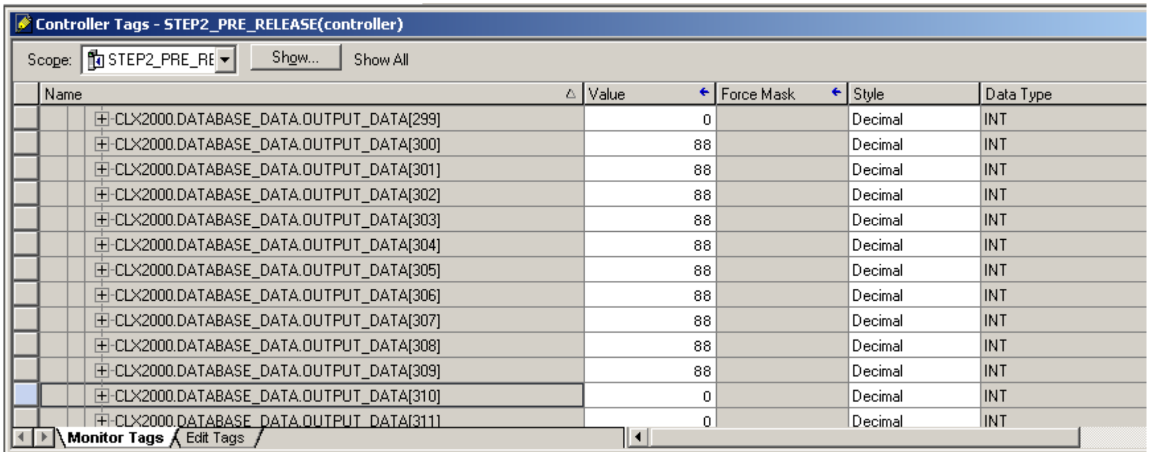
Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.INPUT_DATA[99]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[100]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[101]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[102]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[103]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[104]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[105]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[106]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[107]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[108]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[109]	65		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[110]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[111]	0		Decimal	INT

Monitor Tags / Edit Tags

g. Cyclic Function # 9:

i. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

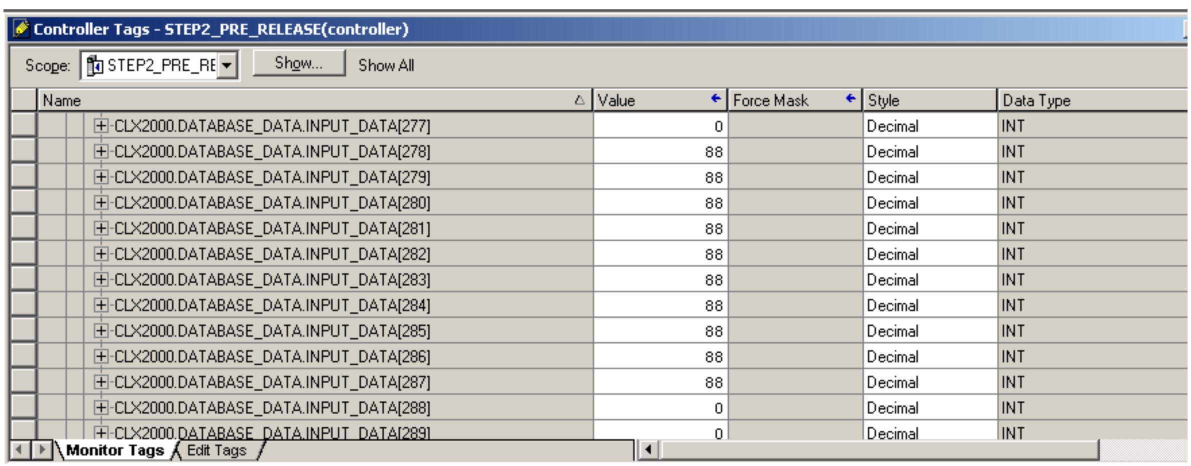

Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.OUTPUT_DATA[299]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[300]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[301]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[302]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[303]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[304]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[305]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[306]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[307]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[308]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[309]	88		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[310]	0		Decimal	INT
CLX2000.DATABASE_DATA.OUTPUT_DATA[311]	0		Decimal	INT

Monitor Tags / Edit Tags

ii. The cyclic function is configured to write data from database address 1900 to MW150 (150 – 128 = 22 + 256 -> offset 278 in Input Table).


Controller Tags - STEP2_PRE_RELEASE(controller)

Scope: STEP2_PRE_RE Show... Show All

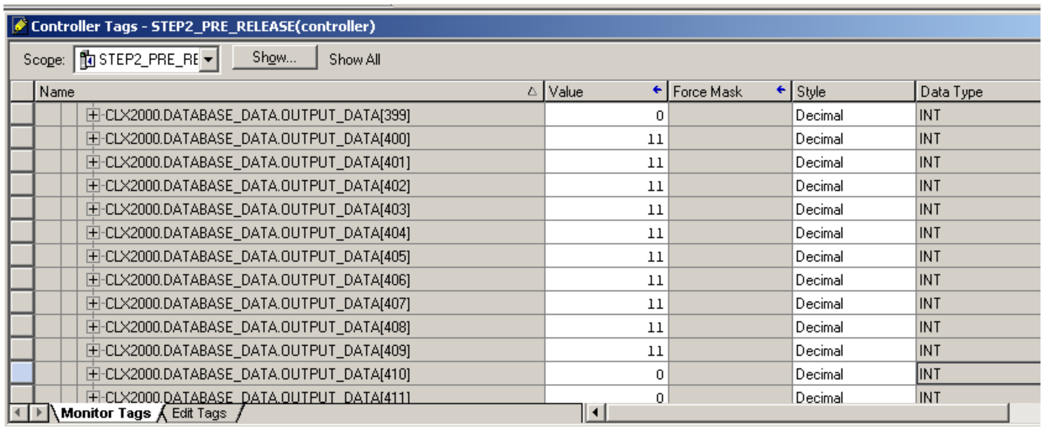
Name	Value	Force Mask	Style	Data Type
CLX2000.DATABASE_DATA.INPUT_DATA[277]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[278]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[279]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[280]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[281]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[282]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[283]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[284]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[285]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[286]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[287]	88		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[288]	0		Decimal	INT
CLX2000.DATABASE_DATA.INPUT_DATA[289]	0		Decimal	INT

Monitor Tags / Edit Tags

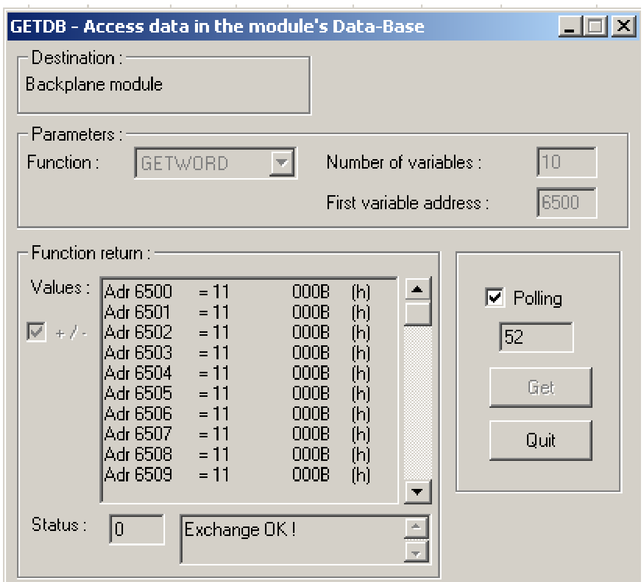
h. Cyclic Function # 10:

i. Expand the CLX2000.DATABASE_DATA.OUTPUT_DATA and fill in values as shown in the table below.

ii. The cyclic function is configured to write data from database address 2000 to 6500 (DB25.DW100). The server equipment destination address is outside of the database address mapped to CPU.



iii. From the Console application window, locate and launch GetDB. .



To summarize the addresses and values used. See the table below.

Table 18: Read Cyclic Function Test Summary

Cyclic Function #	CPU INPUT Table Address	Values	Client Equipment Database Start Address	Server Equipment Database Start Address
1	0	10-19	0	5120
5	1000	12	1000	1650
6	1100	33	1100	1906
7	1200	2	1200	50

Table 19: Write Cyclic Function Test Summary

Cyclic Function #	CPU OUTPUT Table Address	Values	Client Equipment Database Start Address	Server Equipment Database Start Address
3	0	44	0	5632
8	200	65	1800	100
9	300	88	1900	1878
10	400	70	2000	6500

Example 3: Changing the database mapping size.

This sample configuration will only change the database mapping size. This will be a continuation from Example 2.

11. In the RsLogix 5000 software, change the connection state to Offline.

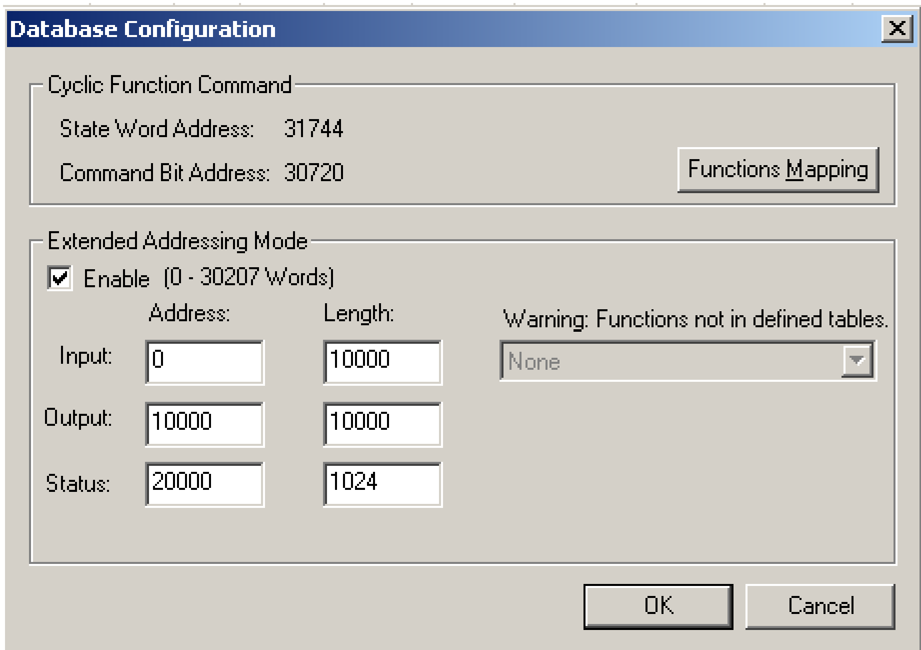
12. From the console application window, run 

and change the INPUT, OUTPUT and STATUS table addresses and size.

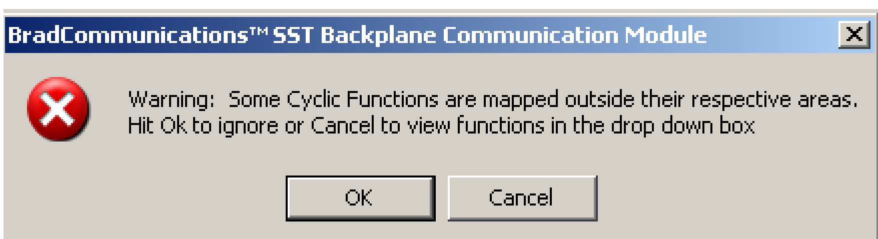
Warning

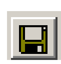
This is just a demonstration of how to change the AOI mapping address. The values that will be used here will cause the cyclic function to stop working since addresses between client and server in the same module are now overlapping

Figure 84: Database Configuration Window



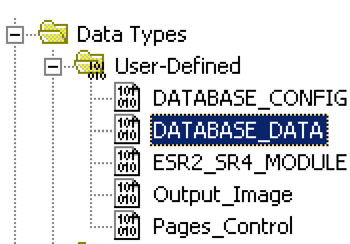
When the values are entered above, the message below will pop up, but you may continue.



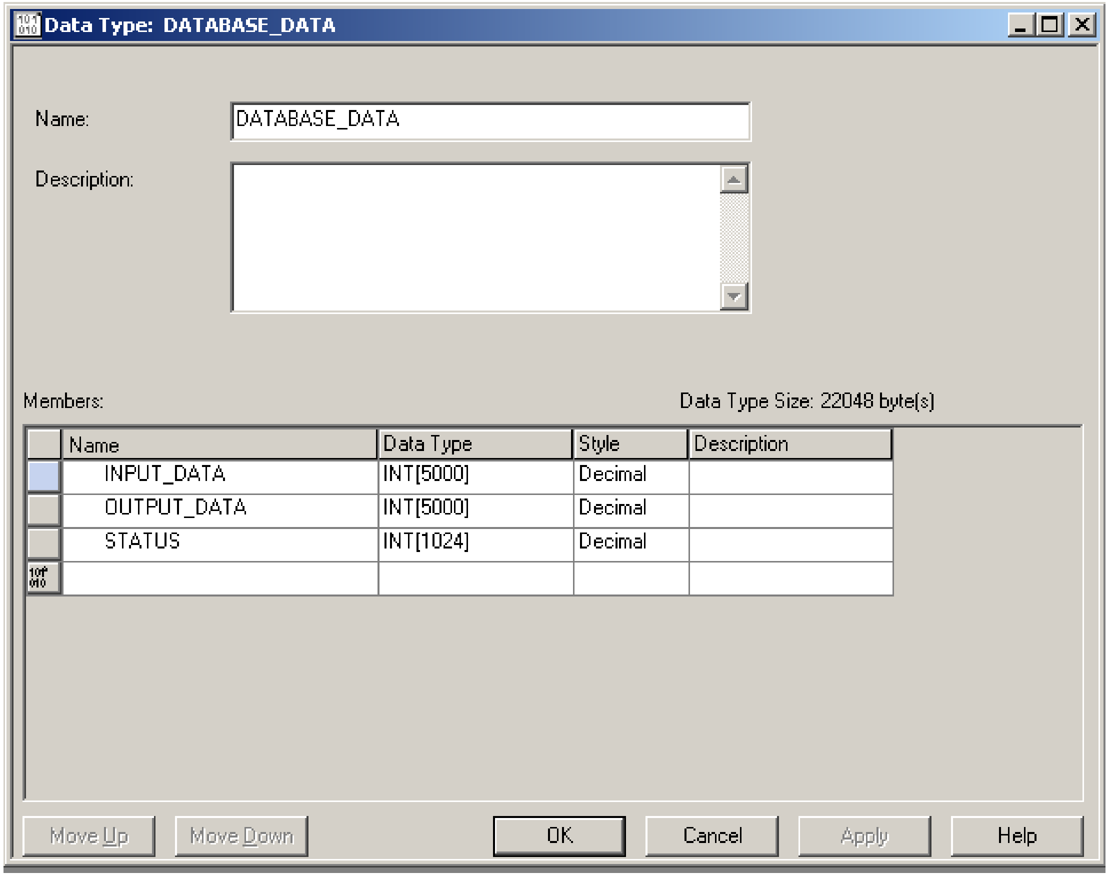
13. Save and download configuration, click  and . The configuration download takes about 2-3 minutes.

14. Check the Status Bar indicator to see if downloading of configuration is successful. Refer to [Console Status Bar Information](#) for status information on the communication to the module.

15. From the Rslogix 5000, double click DATABASE_DATA under Data Types, see below.



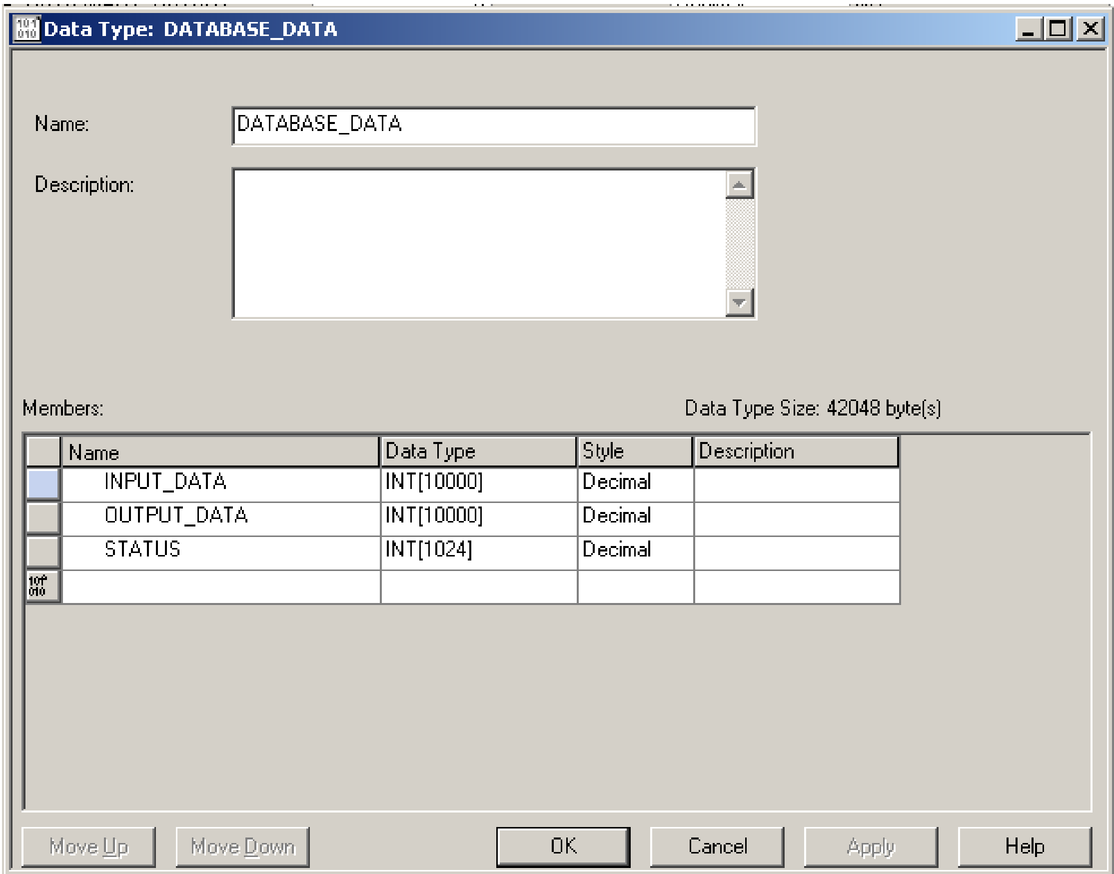
16. The window below should be displayed in your screen



The current AOI configuration allows up 5000 direct mapping of INPUT and OUTPUT table. Before changing the value read some important notes below:

- If the desired mapping size fits in this size, there is no need to change anything from the PLC backplane configuration. Only the database configuration from the console needs to be modified.
- If the desired mapping is greater than what is currently defined, modify them to the desired values.

17. Since the INPUT and OUTPUT mapping sizes are 10000 words each, the sizes here need to be modified. After the change, the display should be similar as below.



18. Click OK and check if CLX2000.DATABASE_DATA.INPUT_DATA and CLX2000.DATABASE_DATA.OUTPUT_DATA size arrays did change.

19. The new array size should now be 0-9999.

20. Save the configuration file and download it to the CLX Module.

6.6 Dynamically Triggering Cyclic Functions

6.6.1 Using Visucyc Tool

Using the new visucyc v1.3.8 or higher, dynamic management of cyclic function is made easy. Modify the configuration of the cyclic function without changing and re-downloading the configuration. See below.

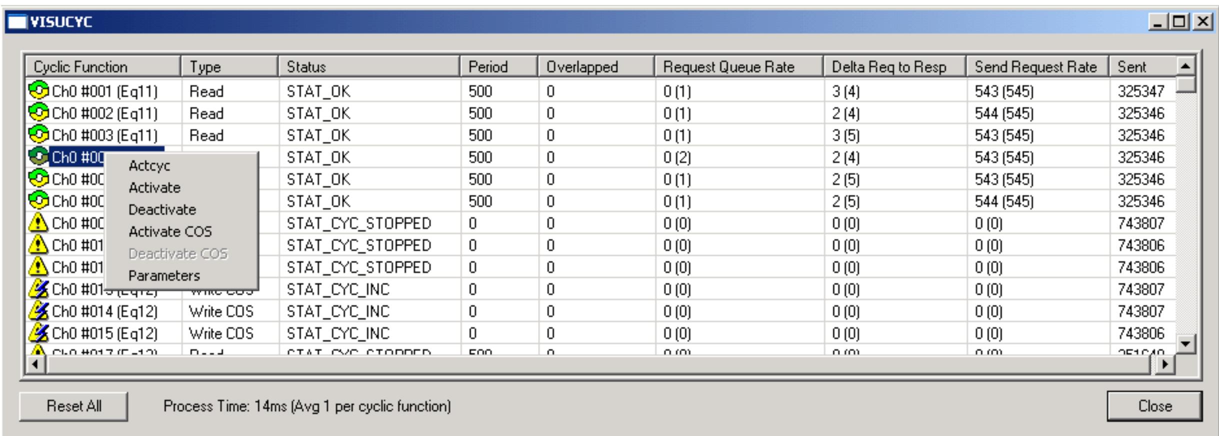


Table 20: Cyclic function commands

State Word	Action
ActCyc	One-shot triggering of the cyclic function
Activate	Periodic triggering activated
Deactivate	Periodic triggering deactivated
Activate COS	On Value Change (COS) triggering activated
Deactivate COS	On Value Change (COS) triggering deactivated

Any commands sent from the VisuCyc will dynamically modify the runtime behavior of the cyclic function. The behavior will remain active unless the module is power cycled and the cyclic function is re-initialized using the downloaded configuration (configured via Console) into the module.

Note

Dynamic triggering of cyclic using VisuCyc is only available to cyclic function #1 - #127 when database address used for Command bits and State words is 32200 and 32100 respectively. This restriction does not apply if database is in extended addressing mode.

6.6.2 Using CIP Messaging

Refer to READ_WRITE_CYCLIC_FUNCTION.ACD sample project for dynamically trigger cyclic functions using CIP messages.

6.6.3 Using Database Paging and Cyclic Function AOI

1. After importing the AOI go to Controller tags and find the CLX2000 tag. This tag is the interface to access the database on our module and controlling cyclic functions dynamically.

2. First set the state words for the cyclic functions that to change to a different state. Under the CLX2000 tag, expand CLX2000.CYCLIC_FUNCTION_STATE_WORDS to show all 4 channels. Select the channel to use. In this example, cyclic functions 1 and 8 on channel 0 are to be periodically deactivated using command 2.

   Available commands values (0 = One-Shot, Periodic Activate = 1, Periodic deactivated = 2, Change of State activated = 3, Change of State deactivated = 4).

   CH0_State_Words[1] corresponds to cyclic function 1, CH0_State_Words[255] corresponds to cyclic function 255.

Name	Value	Style
CLX2000	{...}	
CLX2000.CONFIGURATION	{...}	
CLX2000.DATABASE_DATA	{...}	
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS	{...}	
CLX2000.CHANNEL_1_CYCLIC_FUNCTION_COMMAND_BITS	{...}	
CLX2000.CHANNEL_2_CYCLIC_FUNCTION_COMMAND_BITS	{...}	
CLX2000.CHANNEL_3_CYCLIC_FUNCTION_COMMAND_BITS	{...}	
CLX2000.CYCLIC_FUNCTION_STATE_WORDS	{...}	
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words	{...}	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[0]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[1]	2	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[2]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[3]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[4]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[5]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[6]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[7]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[8]	2	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[9]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[10]	0	Decimal
CLX2000.CYCLIC_FUNCTION_STATE_WORDS.CH0_State_Words[11]	0	Decimal

3. Next go to the command bit area. Expand CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS as below to get to the bit level. In this example, cyclic functions 1 and 8 command bits need to be set.

   Cyclic_Command_bits[1] corresponds to cyclic function 1. Cyclic_Command_bits[255] corresponds to cyclic function 255.

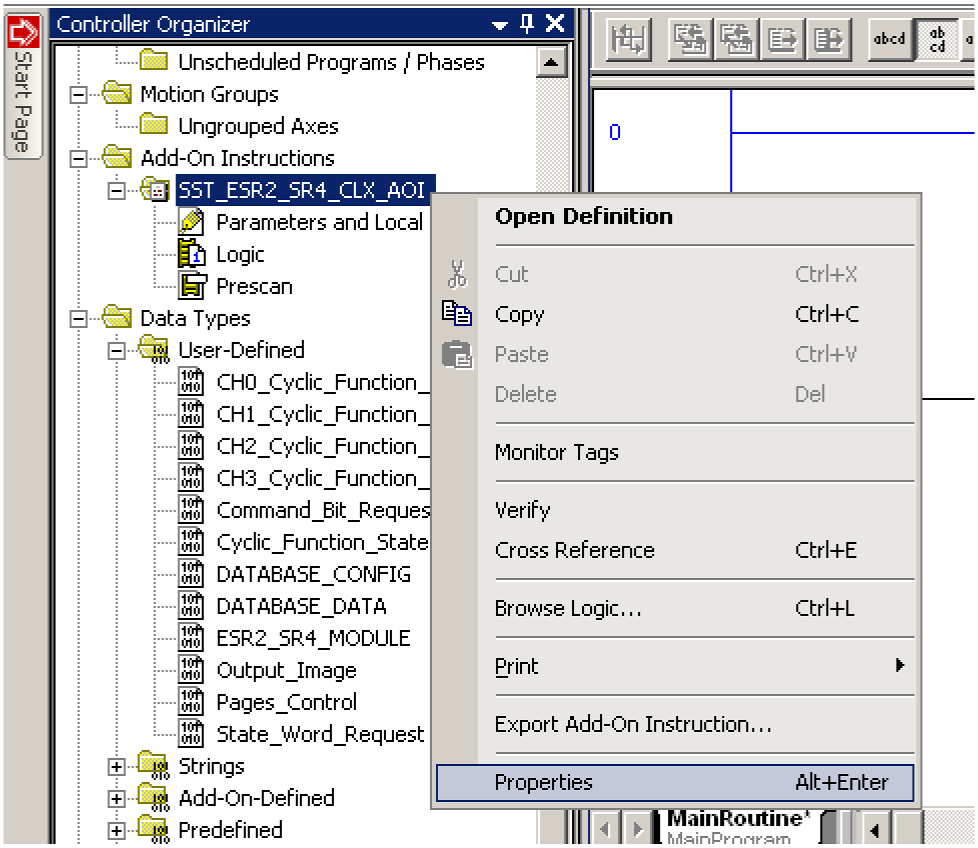
Name	Value	Style
CLX2000	{...}	
CLX2000.CONFIGURATION	{...}	
CLX2000.DATABASE_DATA	{...}	
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS	{...}	
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits	{...}	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[0]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[1]	1	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[2]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[3]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[4]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[5]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[6]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[7]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[8]	1	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[9]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[10]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[11]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[12]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[13]	0	Decimal
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS.Cyclic_Command_Bits[14]	0	Decimal

4. To change the state of cyclic functions, select the CLX2000.CHANNEL_X_TRIGGER where X stands for the channel number. In this example channel 0 is used. Set CLX2000.CHANNEL_0_TRIGGER to 1. This trigger and the Cyclic_Function_Bits that were set in step 2 will clear to zero when cyclic functions have changed state.

CLX2000	{...}
CLX2000.CONFIGURATION	{...}
CLX2000.DATABASE_DATA	{...}
CLX2000.CHANNEL_0_CYCLIC_FUNCTION_COMMAND_BITS	{...}
CLX2000.CHANNEL_1_CYCLIC_FUNCTION_COMMAND_BITS	{...}
CLX2000.CHANNEL_2_CYCLIC_FUNCTION_COMMAND_BITS	{...}
CLX2000.CHANNEL_3_CYCLIC_FUNCTION_COMMAND_BITS	{...}
CLX2000.CYCLIC_FUNCTION_STATE_WORDS	{...}
CLX2000.NUMBER_OF_BLKS	{...}
CLX2000.VALID_CONFIGURATION	0
CLX2000.CHANNEL_0_TRIGGER	1
CLX2000.CHANNEL_1_TRIGGER	0
CLX2000.CHANNEL_2_TRIGGER	0
CLX2000.CHANNEL_3_TRIGGER	0

6.7 Changing the I/O connection size

1. Highlight the Add-On-Instruction "SST_ERSR2_SR4_CLX_AOI" in the Rslogix5000 project and right-click on it and select Properties.



Note

Before changing the I/O connection size in AOI, the connection parameters in the ESR2 module (1756-MODULE) properties must be modified with desired settings.

For default address mode, the supported parameter settings are:

***Input** - Assembly Instance to 1 and Size from 2 to 250 16-bit elements*

***Output** - Assembly Instance to 2 and Size from 2 to 248 16-bit elements*

***Status Input** - Assembly Instance to 5 and Size from 50 to 250 16-bit elements*

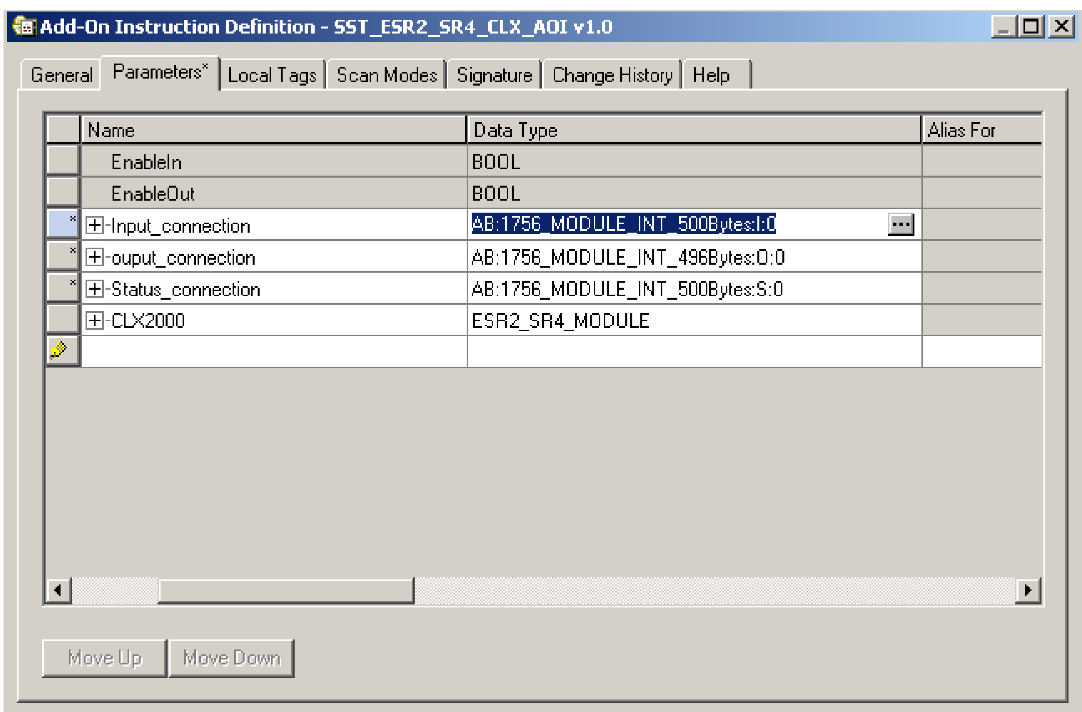
For extended address mode, the supported parameter settings are:

***Input** - Assembly Instance to 1 and Size from 14 to 250 16-bit elements*

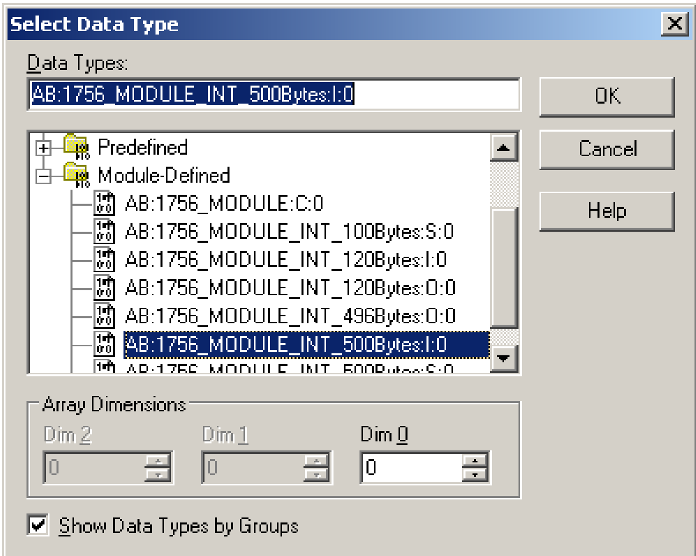
***Output** - Assembly Instance to 2 and Size from 14 to 248 16-bit elements*

***Status Input** - Assembly Instance to 5 and Size from 50 to 250 16-bit elements*

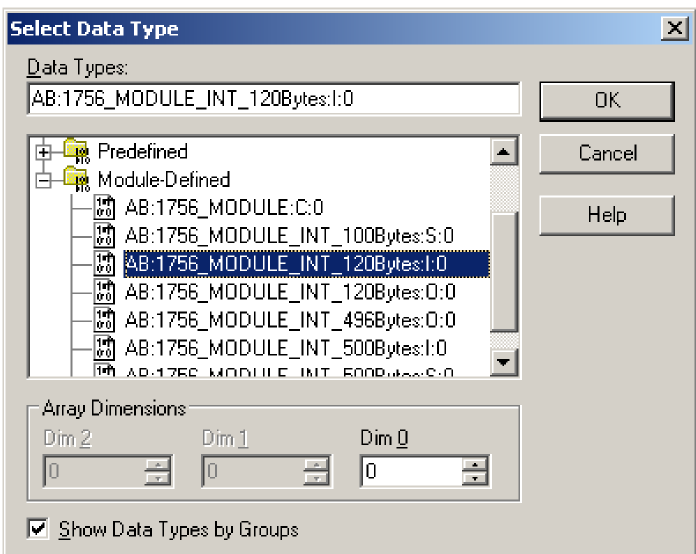
2. The input_connection, output connection and the Status_connection data types need changing to support the smaller I/O sizes.



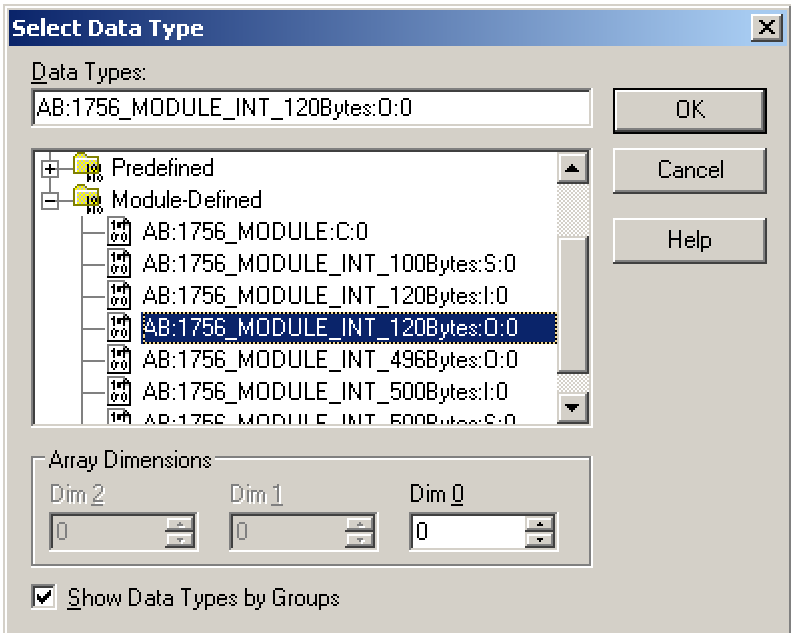
3. The select Data type dialog is displayed.



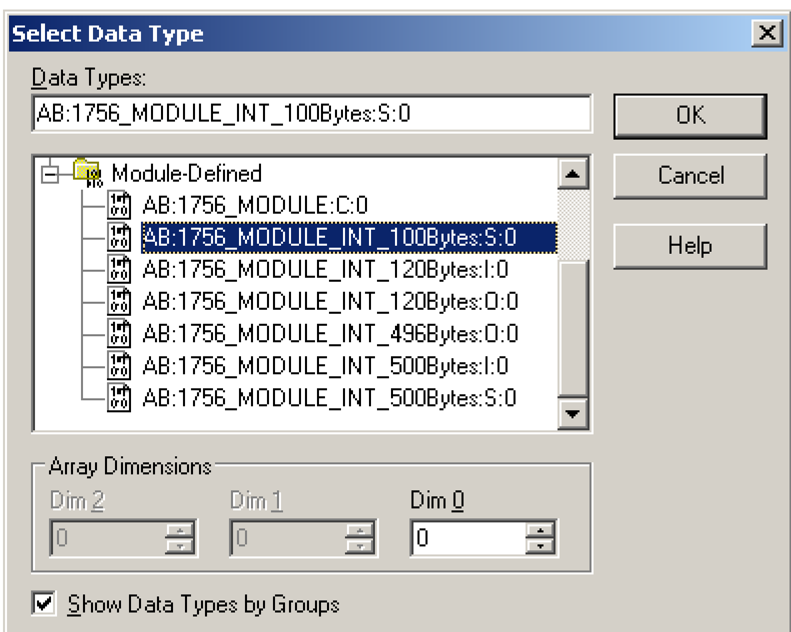
4. Select the new Module-Defined Data type. In this example 120 Bytes (60 words) is the new size for the Input connection. Click OK.



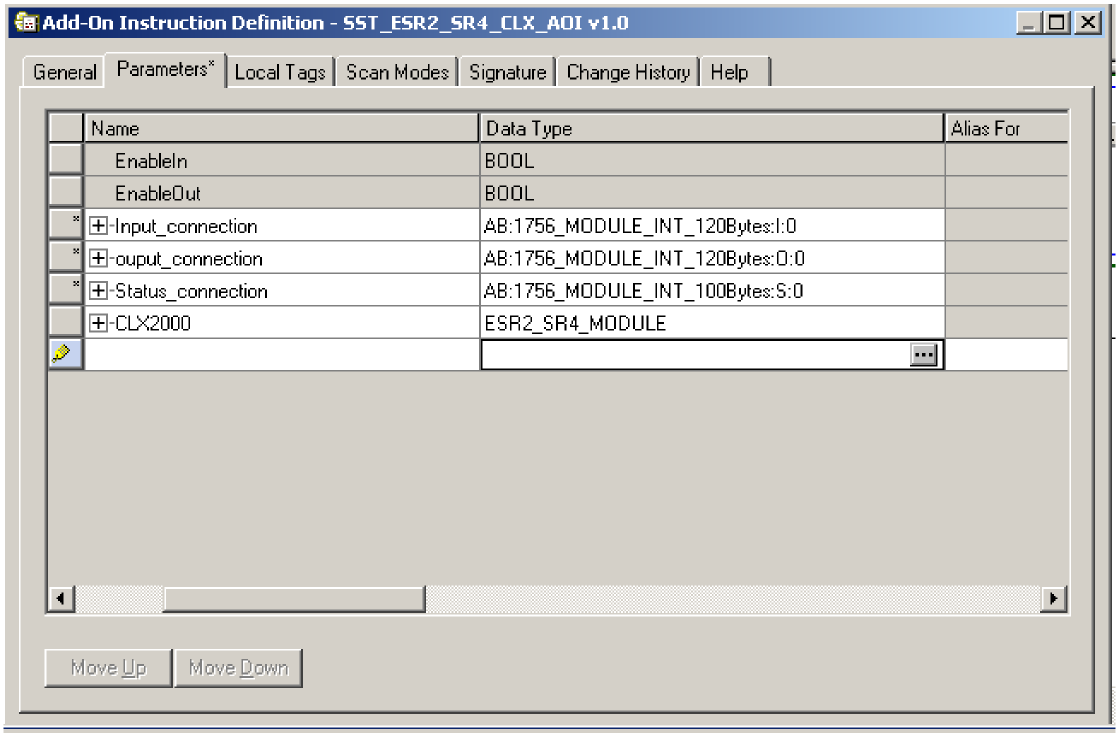
5. Now change the Output_Connection data type to 120Bytes:O:0. And click OK.



6. Now change the Status_Connection data type to 100Bytes:S:0



7. The following connections data types are now changed as below.



6.8 Configure the Module in Listen-Only Mode

Listen-only connection refers to an input-only connection that is made to an SST-ESR2/SR4-CLX-RLL module that is already configured with outputs from the main controller. A listen-only connection allows inputs on an SST-ESR2/SSR4-CLX-RLL module to be monitored from another CLX Processor. Up to 15 listen-only connections plus 1 output connection are supported on SST-ESR2/SR4-CLX-RLL.

Note

Listen-only support is available in backplane firmware ESRCLX.SS4 Version 1.7 and higher.

Up to the maximum 250 words in the CLX input table and 250 words in the Status table can be monitored from another CLX processor that has a listen-only connection to SST-ESR2/SR4-CLX-RLL module.

When setting up a listen-only connection in RSLogix 5000 makes sure to use the same data format that is configured on Main CLX controller with the output connection. Supported data formats for listen-only connections are:

- Input Data INT - With Status
- Input Data DINT - With Status
- Input Data SINT - With Status

With a listen-only connection, controlling an SST-ESR2/SR4-CLX-RLL module is not possible. A listen-only connection could be used to trigger another CLX controller to take control of SST-ESR2/SR4-CLX-RLL module when the main controller with an output connection has gone down.

When the main output connection is lost to the SST-ESR2/SR4-CLX-RLL while it has a listen-only connection, the SST-ESR2/SR4-CLX-RLL will remain in RUN mode with outputs held in their last state. This allows another CLX controller to take control over the SST-ESR2/SR4-CLX -RLL module and resume updating output data on a Modbus network.

Note

*If the listen-only connection is closed before the output connection is reestablished, the SST-ESR2/SR4-CLX-RLL will enter the configured output state set by the user. (all zeros and Cyclic Functions Stop, Hold Last State and Cyclic Functions Run (**default**) or Hold Last State and Cyclic Functions Stop)*

The following considerations should be taken into account when setting up a listen-only connection:

1. Input Data being monitored from a listen only connection should be analyzed to determine if any outputs should be changed prior to re-establishing a new output connection.

2. Output data on the CLX processor that is taking over the I/O connection would have to be updated prior to the connection being opened to the SST-ESR2/SR4-CLX-RLL.

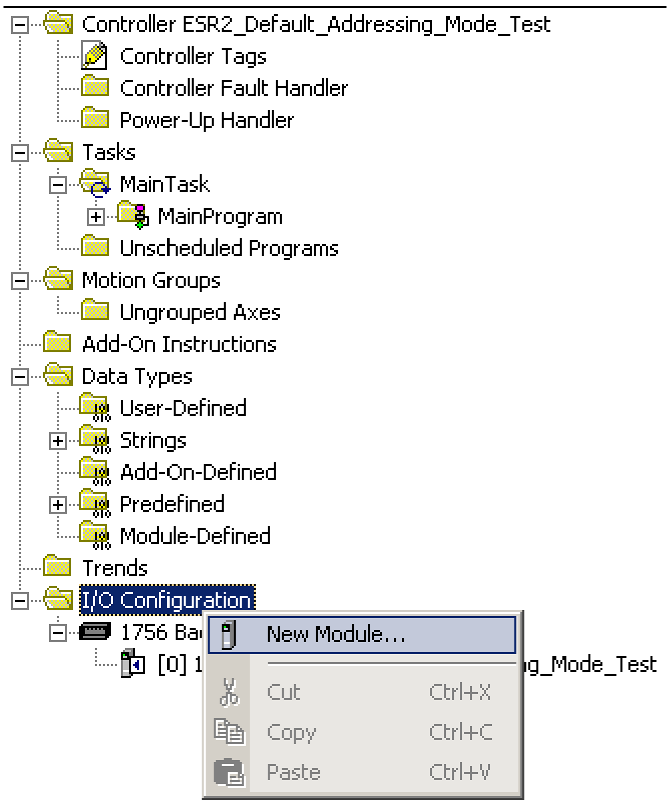
6.9 Configure the Module in Remote Rack via ControlNet Bridge

The SST-ESR2-CLX and SST-SR4-CLX modules can operate in a remote rack via ControlNet Bridge. When scheduled ControlNet connections and very fast RPIs are required, the input and out data sizes for our module may have to be reduced. The factors that affect this data size reductions are the amount of other I/O in your system, number of ControlNet bridges, and the RPIs (Requested Packet Interval) configured for all modules.

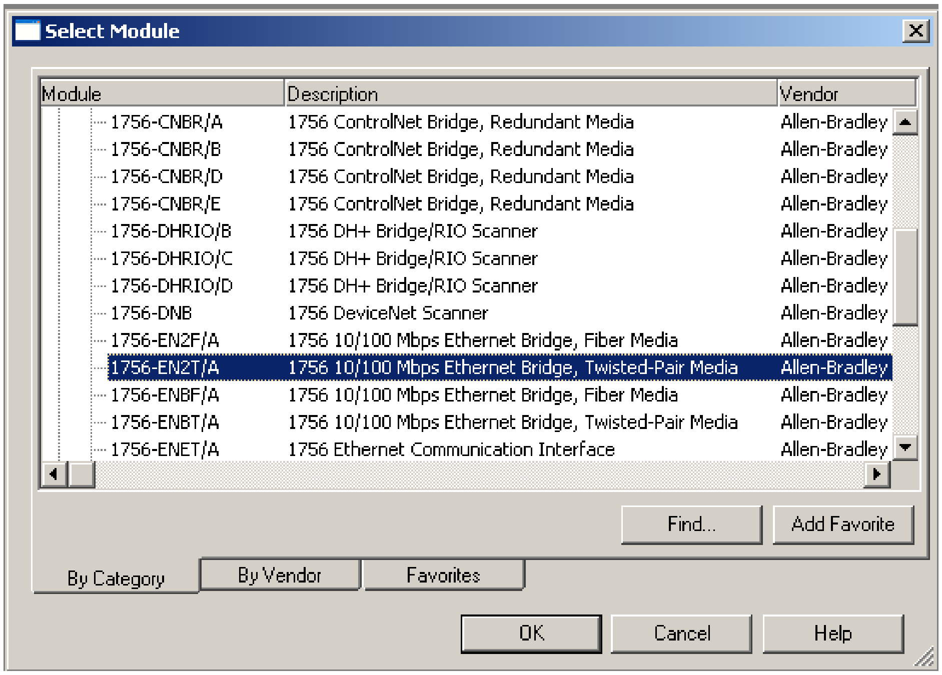
To use the communication module in remote rack configuration using Generic Profile follow the steps below:

1. Complete procedures 1-5 in Using 1756 Generic Profile section.

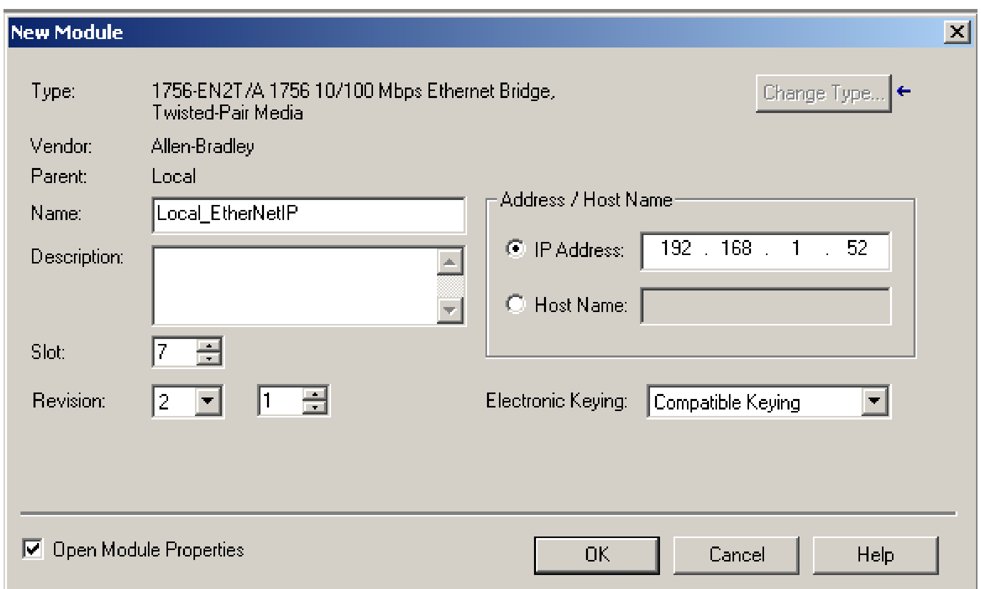
2. In the project tree, right-click on I/O Configuration and select New Module from the shortcut menu. See below for illustration.



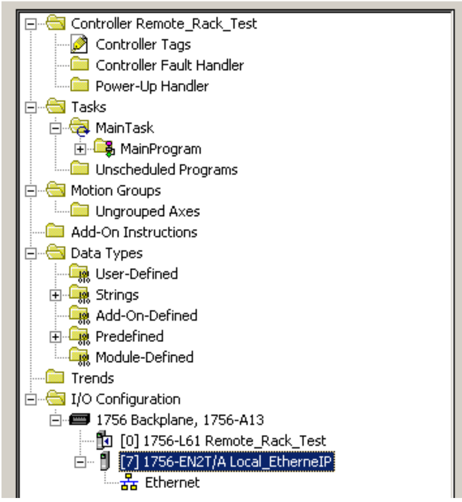
3. Expand “Communication” and select the Ethernet Module you will use for local connection.



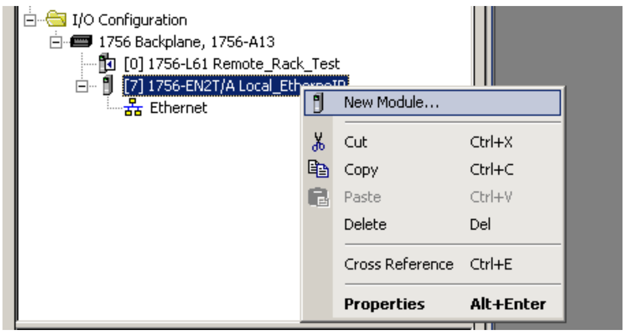
4. Then click OK, then select the revision applicable in your Ethernet/IP module, click OK, and in the next window, fill in applicable information from your setup, and then hit OK.



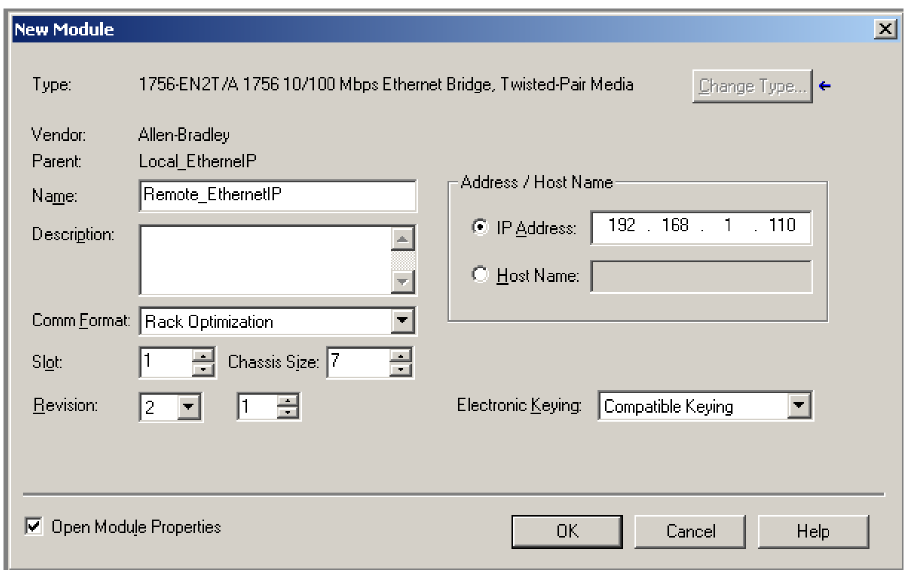
5. The similar display as shown below.



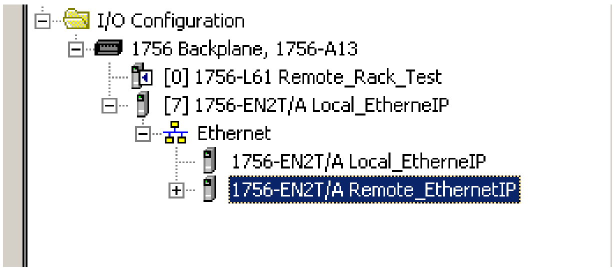
6. Right click on the newly added local Ethernet Module and select "New Module"



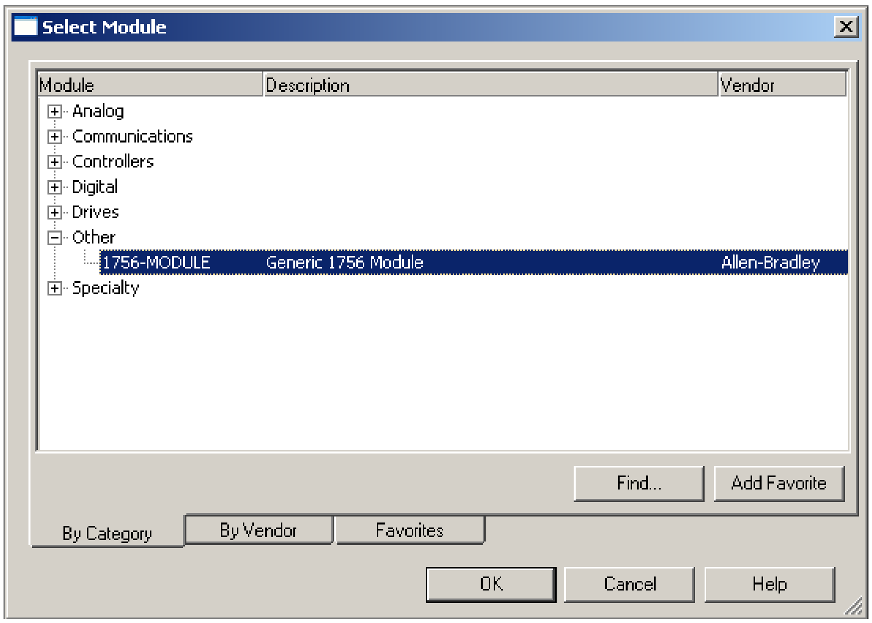
7. Expand "Communications" and select the applicable Ethernet/IP module that will be used for remote configuration



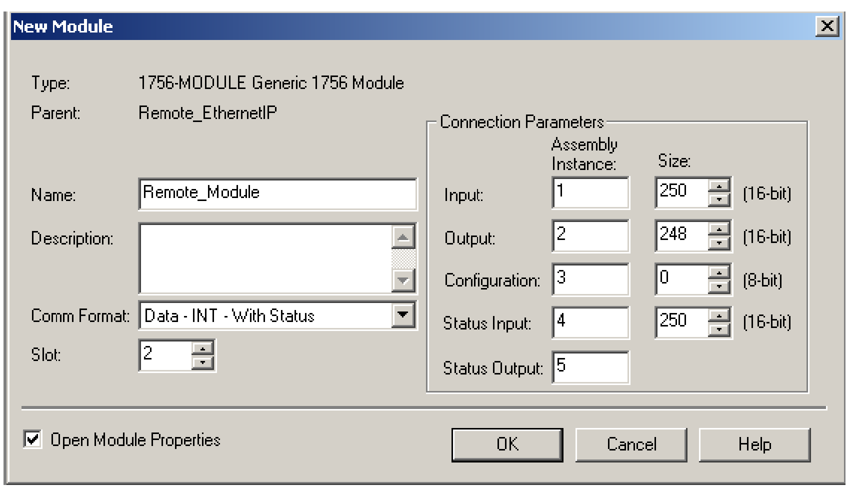
8. Click OK, the remote Ethernet/IP module is added



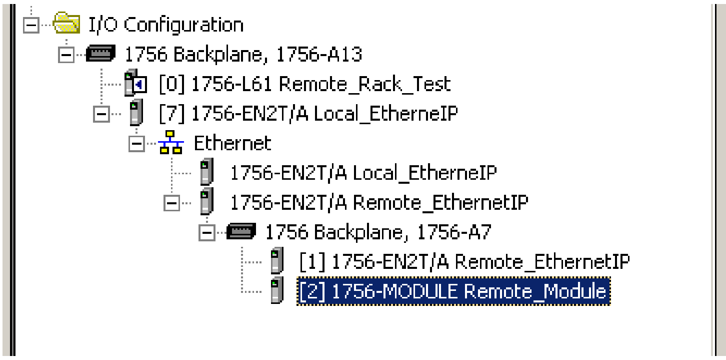
9. Right click on the newly added remote Ethernet Module and select “New Module”, expand “Other” and select “Generic 1756 Module”, click OK.



10. Fill in applicable parameters and click OK.



11. The module is now configured in remote rack configuration.

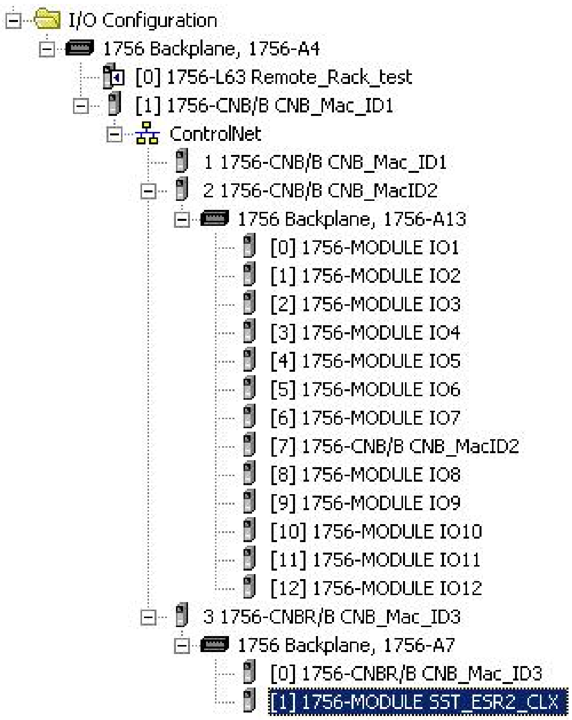


In the following subsection, remote rack tests were executed to determine the number of modules possible in a scheduled connection remote rack environment. The following hardware and software were used to produce the test datato show what data sizes would be required for our modules when used in a remote rack via ControlNet bridge.

6.9.1 Performance results of using module in Remote Rack via ControlNet Bridge

The following I/O configuration was used in the performance testing of the SST-ESR2-CLX in a remote rack via ControlNet Bridge.

Figure 85: Remote Rack via Control Bridge Configuration Test Environment Example(Other I/O included was 24 words Input, 24 words output in remote rack at Node #2)



User Scenario (using system test environment as in Figure 85)

1. Determine how many of our modules is required in the remote system and what RPI must be met.
2. Use Table 21 and Table 22 to determine what I/O sizes you require.
3. Add the modules as required to the remote rack using 1756-Module Generic Profile.

Hardware used:

Backplane Module	SST-ESR2-CLX, SST-SR4-CLX
Physical I/O	Mix of 12 I/O modules each with 2 words I/O totalling 24 words Input /24 words Output. All of this I/O was included in scheduling ControlNet connections.
Rockwell Hardware used	1756-L63 Processor version 20 2 x 1756-CNB/B version 2.30 1 x 1756-CNBR/B version 2.30 1756-A4/B Rev B01 (4-Slot Local Rack) 1756-A13/B Rev A01 (13-Slot Remote Rack #1) 12 slots filled with I/O each with 2 Words Input/ 2 Words Output. 1 slot used for 1756-CNB/B 1756-A7/A (7-Slot Remote Rack #2 for adding SST-ESR2-CLX modules) 1756-ENBT (for configuration and monitoring only)

Software used:

Rockwell Software	RSLogix 5000 Version 20 RSNetwox for ControlNet v10.01 (CPR 9 SR3) RSlinx Gateway Version 2.57

6.9.2 Performance results of using module in remote rack via ControlNet Bridge

The following tables show what I/O sizes for our modules are possible at a specified RPI value range (5ms – 24ms) when used in a remote rack with a ControlNet Bridge. As the number of our modules increase, the input and output sizes decrease due to the amount of scheduled ControlNet traffic. This table shows for example that if 2 SST-ESR2-CLX modules with maximum I/O sizes (250 Words Output and 248 Words Input with no status) are required in a remote system then an RPI of 12ms or higher would be required.

The first table (Table 21) is for connections requiring no status and the second table (Table 22) is for connections requiring status. This data is based on the configuration (See Above) and is to be used as a guideline to give you an estimate of what I/O sizes you can expect at certain RPI range when using the modules in a remote rack via ControlNet Bridge.

Table 21: Successful Scheduled ControlNet connections with no status with RPIs 5ms, 6ms, 12ms, 24ms

RPI (ms) set on all modules (SST, CNB, I/O)	# of SST Modules	SST Input Size (Words)	SST Output size (Words)	Status Size (Words)	Total other I/O in remote racks	ControlNet NUT (ms)
5	1	130	130	0	24words I/O	2
	2	60	60	0	24words I/O	2
	3	40	40	0	24words I/O	2
	4	20	20	0	24words I/O	2
6	1	250	248	0	24words I/O	3
	2	160	160	0	24words I/O	3
	3	100	100	0	24words I/O	3
	4	80	80	0	24words I/O	3
12	1	250	248	0	24words I/O	3
	2	250	248	0	24words I/O	3
	3	250	248	0	24words I/O	3
	4	200	200	0	24words I/O	3
24	1	250	248	0	24words I/O	3
	2	250	248	0	24words I/O	3
	3	250	248	0	24words I/O	3
	4	250	248	0	24words I/O	3

Summary

- The lowest possible RPI that can be configured for SST-ESR2-CLX module with maximum I/O sizes with no status is 6ms in the test environment (See Figure 85).
- More than 2 modules with maximum I/O sizes with no status requires an RPI of 12ms or higher.
- More than 3 modules with maximum I/O sizes with no status requires and RPI of 24ms or higher.

Table 22: Successful Scheduled ControlNet connections with status with RPIs 6ms, 12ms, 24ms

RPI (ms) set on all modules (SST, CNB, I/O)	# of SST Modules	SST Input Size (Words)	SST Output size (Words)	Status Size (Words)	Total other I/O in remote racks	ControlNet NUT (ms)
6	1	230	230	250	24words I/O	3
12	1	250	248	250	24words I/O	3
	2	230	230	250	24words I/O	3
	3	70	70	250	24words I/O	3
24	1	250	248	250	24words I/O	3
	2	250	248	250	24words I/O	3
	3	250	248	250	24words I/O	3
	4	230	230	250	24words I/O	3

Summary

- The lowest possible RPI that can be configured for SST-ESR2-CLX module with maximum I/O sizes with status is 12ms in the test environment (See Figure 85).
- More than 2 modules with maximum I/O sizes with status requires an RPI of 24ms or higher.

Note

If connections with no status are used, and cyclic update of cyclic function status is required, cyclic function status can be mapped to database area 0 – 249 using Expert mode in the SST backplane Communication Module Console. To enable Expert Mode in console, go to File Menu, select Preferences and then Expert Mode.

6.10 Optimizing Module's Performance when Using Extended Database Mapping

- Cyclic Function data should not span into multiple page. (Input Page size is equivalent to the Input connection size (In Words) - 4 size configured in the RSLogix Software, Output size is equivalent to the Output connection size (in Words) – 4 in the RSLogix Software).

  For example:

  A write cyclic function is configured at Database address offset 700 with a length of 100. The I/O connection size used is 250 words Input and 248 Words Output. The Input page size would be 246 words and the output page size would be 244 words.

  Database settings used are the defaults (Input Address 0 / Input Length 512, Output Address 512 / Output Length 512, Status Address 1024 / Status length 255). Page 1, 2 and 3 of the output area would start at 512, 756 and 1000 consecutively.

  The first 56 words of data would be in the first page and the remaining data would in the subsequent page. To improve the performance, changing the Database address to start at 756 for the write cyclic function would keep all the data in the same page.

- Using a polling interval higher than the total process time (Delta Req to Resp) for all cyclic functions.

7

Communication Module General References

Chapter Sections:

- Brad Communication ™ SST Backplane Communication Module Software (BCMS)
- Database Mapping
- Cyclic Function
- Initialization Principle and Methods
- Modifying the TCP Interface Configuration of SST-ESR2-CLX-RLL Product Variant

7.1 Brad Communication ™ SST Backplane Communication Module Software (BCMS)

BCMS main components are the Configuration Manager and the Console application. Section 3.2 briefly described the functionality provided by Configuration Manager and Console application. This section will cover in detail the different commands, functionality and links that BCMS offers and can be utilized to configure and initialize our communication module.

7.1.1 Configuration Manager

The Configuration Manager includes the following commands:

Table 23: Configuration Manager Commands

Group	Name	Function
Management		
	Duplicate	Creates a new configuration based on the currently selected one.
	Rename	Changes the name of the selected configuration.
	Delete	Deletes the selected configuration.
	Properties	Modifies the settings of the selected configuration.
	Get	Gets the IP address, gateway address, and mask from the Module. For RLL and Ethernet/IP configuration only. For TCP, use the ClxIPConfiguration tool. Applies to SST-ESR2-CLX only
	Set	Sets the IP address, gateway address, and mask on the Module. For RLL and Ethernet/IP configuration only. For TCP, use ClxIPConfiguration tool. Applies to SST-ESR2-CLX only
	Default (Factory Setting)	Set the Ethernet settings (IP address, Mask and Gateway are set back to factory settings, 192.168.1.12, 255.255.255.0 and 0.0.0.0 respectively). Applies to SST-ESR2-CLX only

Local Configuration		

	Create empty configuration	Creates a configuration from scratch. The following must be filled in with the applicable values:
		• Configuration name
		• Configuration description
		• Type – communication module variant
		  o SST-ESR2-CLX-RLL
		  o SST-SR4-CLX-RLL
		• Connection Type
		  o Files Only
		  o RSLinx
		  o TCP/IP
		  o Ethernet/IP
		• RSLinx Driver Name – required when using Rslinx connection type
		• Extended Path - required when using module in Remote Rack configuration
		• IP address - Enter the IP address of the Ethernet/IP ControlLogix module such as a 1756-ENBT
		• Slot Number - the slot number of module in either local or remote rack.
		• Virtual Backplane Slot Number - Only required if connecting to Rockwell ControlLogix Ethernet/IP module via USB
		• Local Network Interface Card IP address - required if using Ethernet/IP Driver as the connection type. Select the IP address of Network card in PC to use to make a connection

	Open selected configuration	Opens the selected configuration
Download to Backplane		
	Configuration	Downloads the selected configuration to the Backplane
	Firmware	Launches the Flash Updater application to be able to update protocol and backplane firmware on module. The flash updater will use the communication settings from the project that was used to configure your module.
Upload from Backplane		
	Create new configuration	Creates a new configuration on the hard drive from the configuration contained in the Backplane.
	Replace selected configuration	Replaces the selected configuration with that contained in the Backplane.

When the "**Open Selected Configuration**" option is selected and the **Open** button is clicked, the Backplane Console is started with the selected configuration. This means that:

- All modifications made in the Console will be applied to this configuration
- This configuration will be used to initialize the Backplane Module
- The cyclic functions will be those declared in this configuration

7.1.2 Console

This section discusses in details the different Console Work Environment areas and their associated functions.

Figure 86: Work Environment Areas

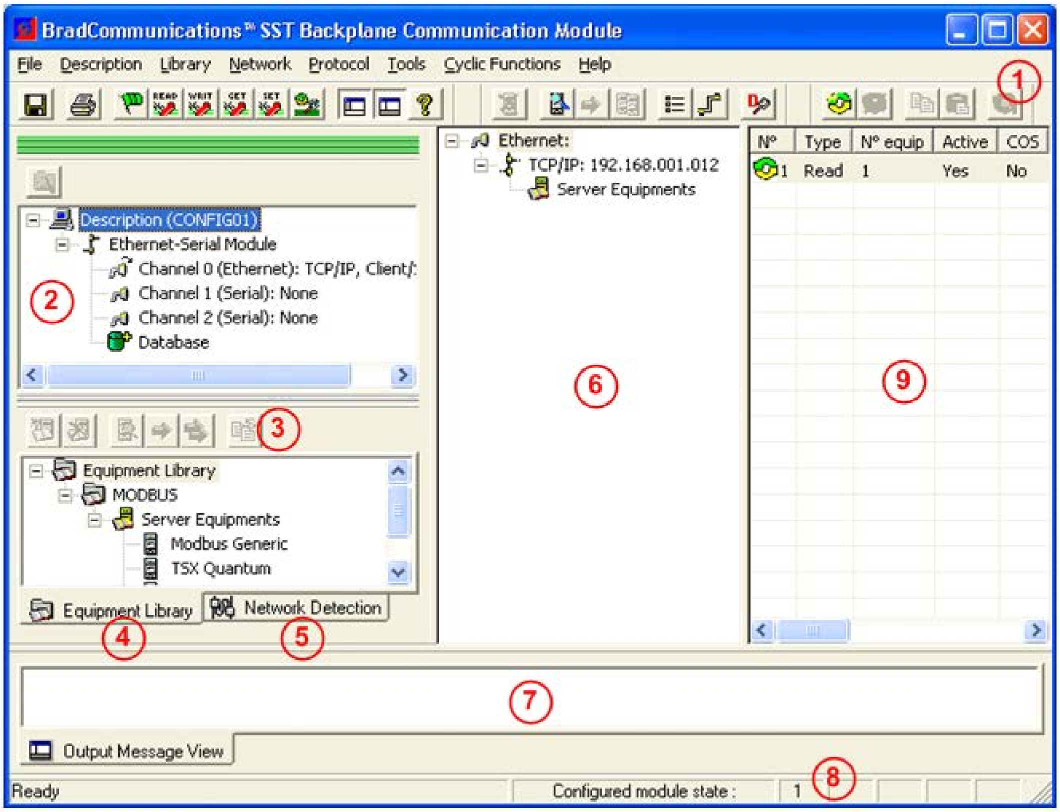


Table 24: Work Environment Area Description

Area	Area Name	Function
1	Menu Bar	Contains all commands linked to areas 2, 4, 6, 7 and 8. Refer to Figure 86, Work Environment Areas.
2	Configuration Description	Used to choose the gate channel to be configured, and in particular, the protocol used.
3	Resource Area	Denotes the area containing the **Device Library** and **Network Detection** tabs. These tabs are resource areas for devices that can be included within the configuration.
4	Equipment Library Tab	Contains a list of devices that can be included within the configuration (Protocol view).
5	Network Detection Tab	Used to detect the configuration that exists on the network. The detected devices can subsequently be included within the configuration (Protocol view).
6	Protocol View Area	Contains the configuration currently being built. This configuration is then used to set up network parameters and declare devices.
7	Message Window Area	Contains the notification messages or error messages returned by the Console.

8	Status Indicator	Indicates whether the Backplane is initialized and whether the configuration present in the Backplane is identical to the current configuration.
9	Cyclic Function Area	List all the cyclic functions configured for the channel.

7.1.2.1 The Menu Bar

The Menu Bar provides access to the entire set of commands that can be used within the Console. It is divided into submenus that correspond to each work environment area.
Each submenu is documented within the specific chapter on the area that it corresponds to.
Only the **File** and **Tools** submenu commands are not linked to a particular area; they are linked instead to general commands.

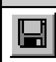
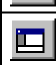
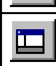
Table 25: Menu Bar Areas

Area Name	Sub-menu Name
Configuration Description	Description
Equipment Library Tab	Library
Network Detection Tab	Network
Protocol View Area	Protocol
Cyclic Function Area	Cyclic functions

General File Menu Commands

The following table lists the File Menu commands, the toolbar icons, the shortcuts and the corresponding actions.

Table 26: File Menu Description

Name in the Menu	Icon	Key-board	Action
Save		Ctrl + S	Used to save the entire configuration at any time
List and Print...			Used to display and print your configuration
Backplane Initialization			Initializes the Backplane
Preferences / Work Environment			Displays or hides the Resource Area
Preference / Message Window			Displays or hides the Message Window Area
Preference / Expert Mode			Switches the Console into Expert Configuration Mode, to access commands that are not available in Normal mode
Exit			Closes the Console

General Tools Menu Commands

The following table lists the Tools Menu commands. This menu is used to run test utilities that are external to the Console. These tools can also be run independently of the Console, using the various available shortcuts.

Table 27: General Tools Menu Commands

Name in the Menu	Action
“Read in device” test tool	Runs the **readwait** utility, which carries out data read operations on a remote device
“Write in device” test tool	Runs the **writewait** utility, which carries out data write operations on a remote device
“Read in database” test tool	Runs the **getdb** utility, which carries out data read operations on a database
“Write in database” test tool	Runs the **setdb** utility, which carries out data write operations on a database
“Cyclic functions”’monitoring test tool	Runs the **visucyc** utility, to dynamically monitor the cyclic function in the Backplane

7.1.2.2 Configuration Description Area

This area is used to select protocols for each of the three module channels and navigate between the different Protocol views, which give access to the definition of devices and cyclic functions. It also allows accessing the database configuration.

Figure 87: Configuration Description Area

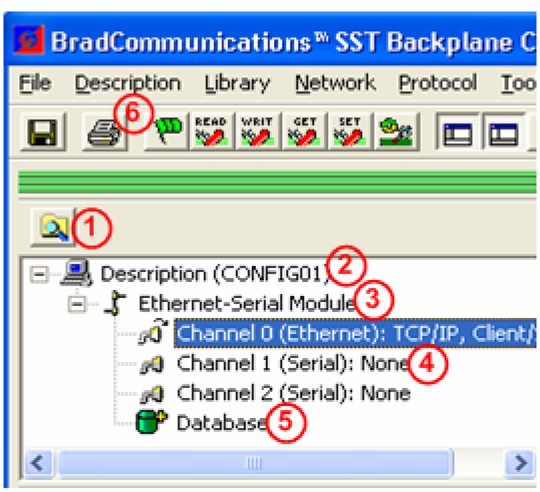
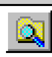


Table 28: Configuration Description Area Description

Reference	Area Name	Comments
1	Configuration Description Toolbar	Includes the most frequently used commands of the Configuration Description area
2	Description Node	Contains the name of the active configuration. This node contains your entire configuration.
3	Backplane Node	Represents the Backplane module variant
4	Protocol Nodes	Represent the protocol allocated to the Backplane communication channel. Select this node to display the associated protocol view. To change a protocol, double-click it.
5	Database Node	Use to configure the database addressing mode
6	Description Menu	Contains all commands available for the Configuration Description area

The following table lists the commands that apply to the Configuration Description area, menus, toolbar icons, shortcuts and corresponding actions. There is also a pop-up menu that can be accessed by right-clicking the relevant node.

Table 29: Configuration Description Commands

Name in the Menu	Icon	Keyboard	Action
Properties		Space	Used to display and modify the properties of the selected node.

7.1.2.3 Protocol View Area

This is the central element of the network configuration. To display the channel configuration for a given card, select the **Card** node. This allows the following:

- The **Device Library** tab to be updated with a list of available devices
- The **Network Detection** tab to be enabled
- The **Protocol View** area to be enabled. This contains the configuration previously saved onto hard disk.

Note

If several cards have been configured, you will need to switch the Protocol view. Only one Protocol view can be displayed at once.
When switching views, the Console will prompt user to save any changes that have been made to the current one.

The Protocol View Area is used for:

- Setting network parameters
- Declaring devices and setting their parameters

Figure 88: Protocol View Area

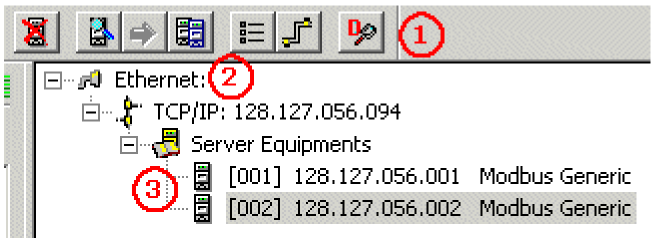
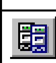
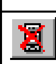


Table 30: Protocol Area Description

Area	Area Name	Comments
1	Protocol View Toolbar	Groups the most frequently used commands from the Protocol view.
2	Root of the tree structure	Denotes the communication channel. This is used to access global information parameters for the network
3	Device Type Node	Denotes a device declared in your configuration

The following table lists the Protocol View area commands, menus, toolbar icons, shortcuts and corresponding actions. There is also a pop-up menu that can be accessed by right-clicking the relevant node.

Table 31: Protocol Area Commands

Name in the Menu	Icon	Keyboard	Action
Insert			Inserts a new device in the configuration. If required, a dialog box will be displayed so that you can select the appropriate device type. This list depends on the node that is currently selected in the Protocol view. Note: This command is not available for all protocols. However, in all cases, a device can also inserted using Drag & Drop from the Device Library. To do this, select the appropriate device type and move it to the Protocol view. The device will automatically position itself. It doesn't matter which node is currently selected.
Duplicate			Duplicates the selected device, along with its properties and items
Delete		Del	Deletes the selected device
Properties		SPACE	Used to display and change the properties for the selected node

Diagnostic…			Launches the diagnostic tool used to check that the configuration is working correctly.
Options…			Used to modify protocol-specific and non-protocol-specific options, such as display characteristics

7.1.2.4 Equipment Library Tab

General Overview

During the network configuration phase, all devices to be used on the network will need to be declared in the Protocol view. The **Device Library** tab constitutes one of the Console resources, and allows you to include a new device in the Protocol view.

Figure 89: Device Library Tab

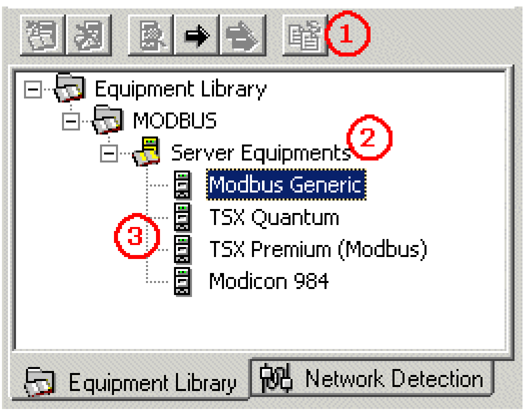


Table 32: Device Library Tab Description

Ref.	Area Name	Comments
1	Device Library Toolbar	Groups the most frequently used commands for managing the Device Library.
2	Sort by Category Node(s)	Intermediate nodes that are used to sort the devices. The type, name and number of intermediate nodes depend on the sort criteria used.
3	Device Node(s)	Terminal node in the tree structure corresponding to a device that can be included in the Protocol view.

Devices can be inserted via Drag & Drop. To do this, select the appropriate device type and move it to the Protocol view. The device will automatically position itself, regardless of which node is currently selected in the Protocol view.

Note

*It is also possible to insert a device directly from the Protocol view by selecting the parent node and using the **Insert** command. In this case, a dialog box will display so that you can select the appropriate device type for the selected node.*

Equipment Library Management

The *Device Library* contains a list of devices that can be included within the configuration.
The **Add** and **Delete** commands are not available.

The following table lists the Device Library area commands, menus, toolbar icons, shortcuts and corresponding actions. There is also a pop-up menu that can be accessed by right-clicking the relevant node.

Table 33: Device Library Actions

Name in the Menu	Icon	Keyboard	Actions
Add		INS	Not available in this version.
Delete		DEL	Not available in this version.
Sort			Used to sort the devices, according to the sort criteria.
Insert in the configuration			Used to insert the device into the configuration. The device can also be inserted using Drag & Drop from the Library to the Protocol view. The device will automatically position itself, regardless of which node is selected in the Protocol view.
Properties		SPACE	Used to display the properties for the selected device. The properties can also be displayed by double-clicking the device.

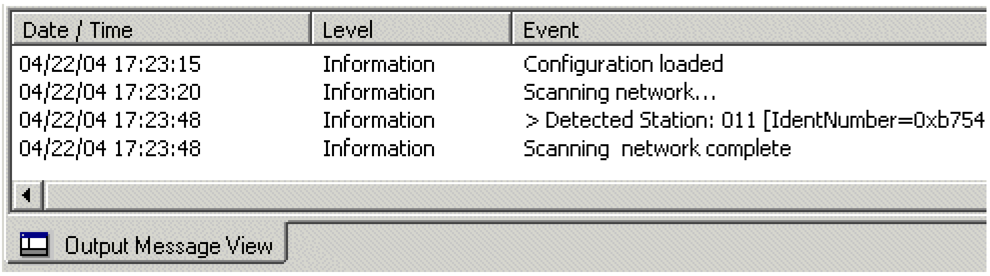
Note

If encountering any problems while using the Device Library, remember to look for messages in the Message window.

7.1.2.5 Message Window Area

The Message Window area allows the Console to display error or notification messages related to the network detection.

Figure 90: Output Message View



This area comprises three columns:

- Date / Time: Date and time of the message
- Level: Message status level (Information, Warning, Error)
- Event: Text of the message

The following commands are available via the File > Message View menu:

Table 34: File Message View Commands

Name in the Menu	Actions
Copy	Copies the selected text onto the clipboard
Clean	Deletes all messages in the area

These messages are logged in the Console.log file, located in the current configuration directory (which can be found via the **File > Configuration Manager** command). The maximum file size can be configured via **File > Preferences**.

Note

- *The message window has a limited size, so the oldest messages may be deleted*
- *This area can be hidden via **File > Preferences**. The messages are still logged in the log file, even if the display area is invisible.*

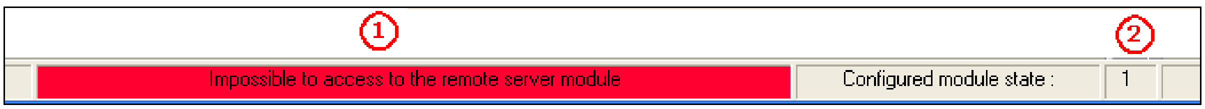
7.1.2.6 Status Indicators

The status indicators display the Backplane's initialization status, and how it agrees with the configuration open in the Console. These indicators are inactive by default but can be activated upon request. A hint is displayed when the mouse cursor is positioned over the status indicator.

Activating Status Indicators

To view the Backplane's status, right-click Box 2 and select the "**Online**" command. Box 2 is empty before the first connection.

Figure 91: Status Indicator



Box 1 contains:

- A message explaining the status of the connection with the Backplane
- A background color, which represents the status

The following table summarizes the various possible contexts for Box1.

Table 35: Card Status Indicator States – Box 1

Background Color	Status
Gray	The Console is not connected to the card
Red	The remote server Module cannot be accessed
Yellow	Communication problem with the remote server
Green	The connection with the remote server is OK

Box 2 contains:

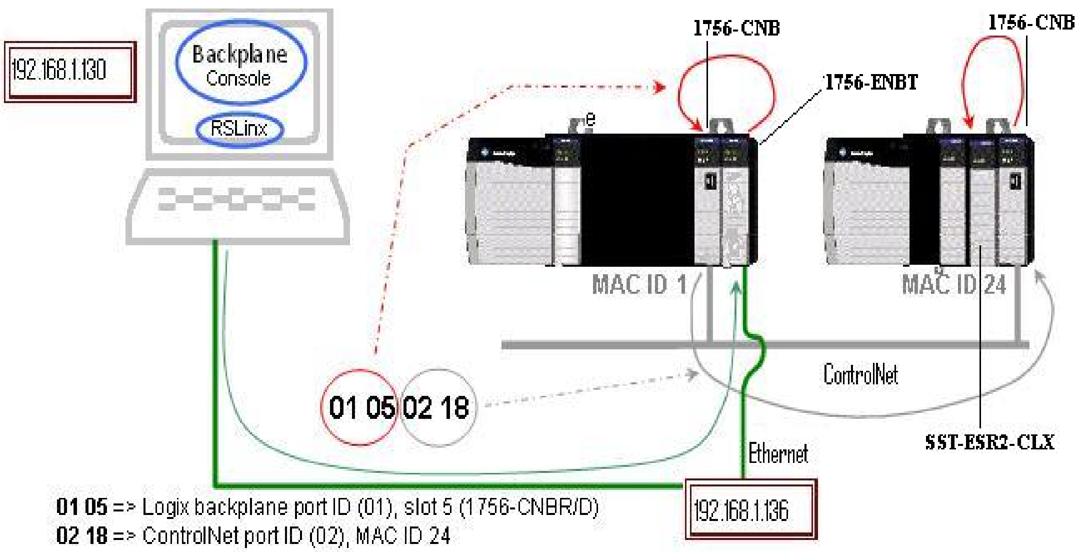
- The Backplane number (always 1)
- A background color, symbolizing the Backplane's status

The following table summarizes the various possible contexts for Box 2.

Table 36: Card Status Indicator States – Box 2

Background Color	Status
Gray	No communication with the Backplane
Red	The Backplane is not initialized
Dark green with white characters	The Backplane was initialized, but with a configuration different from the currently open one (e.g., earlier version). This state enables the current Backplane configuration to be diagnosed. It is recommended that you reinitialize the Backplane before carrying out these actions.
Light green with black characters	The Backplane was initialized with the configuration open in the Console. This state enables the configuration to be diagnosed.

Figure 92: Accessing module remotely using ControlNet Bridge



Here is an example of using an extended path with RSlinx Driver with 1756-CNB modules to connect to a remote rack where the SST-ESR2-CLX module resides.

Extended Path = 01 05 02 18 (values entered are in hex)
IP Address: 192.168.1.136 (1756-ENBT)
01 05 = Logix backplane port ID (01), slot 5 (1756-CNB)
02 18 = ControlNet Port ID (02), MAC ID 24

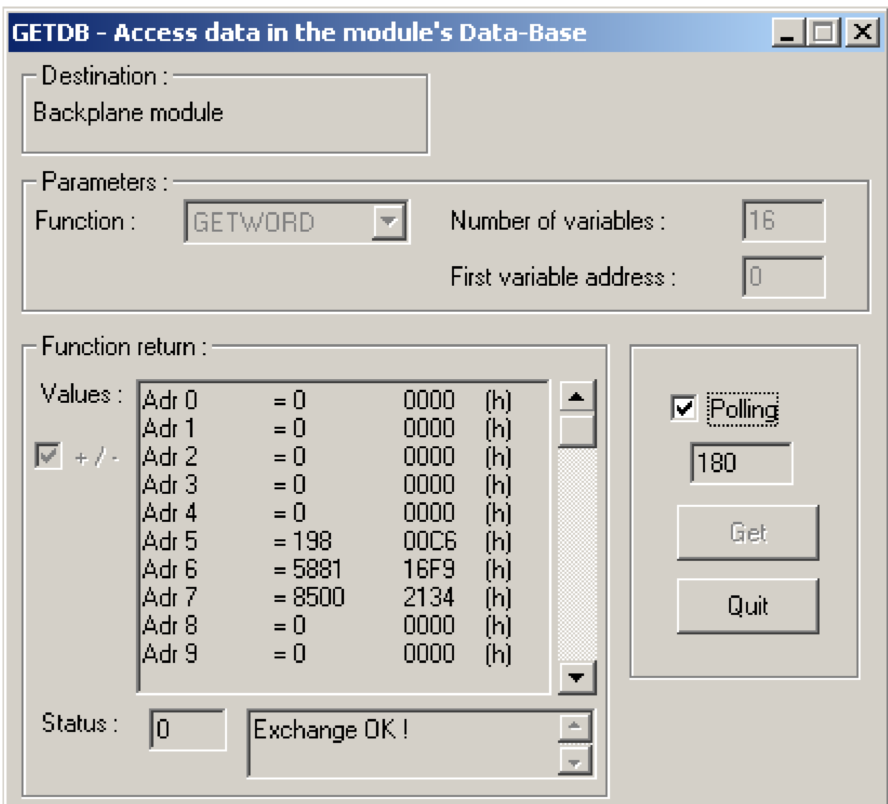
7.1.3 Diagnostic Tools

The Backplane Software Solution includes utilities that can be used remotely through an Ethernet connection or via an RSLinx driver. These utilities can facilitate the production of diagnostics for the Backplane.

7.1.3.1 GetDB

This tool is useful in reading the values / variables directly from the database.

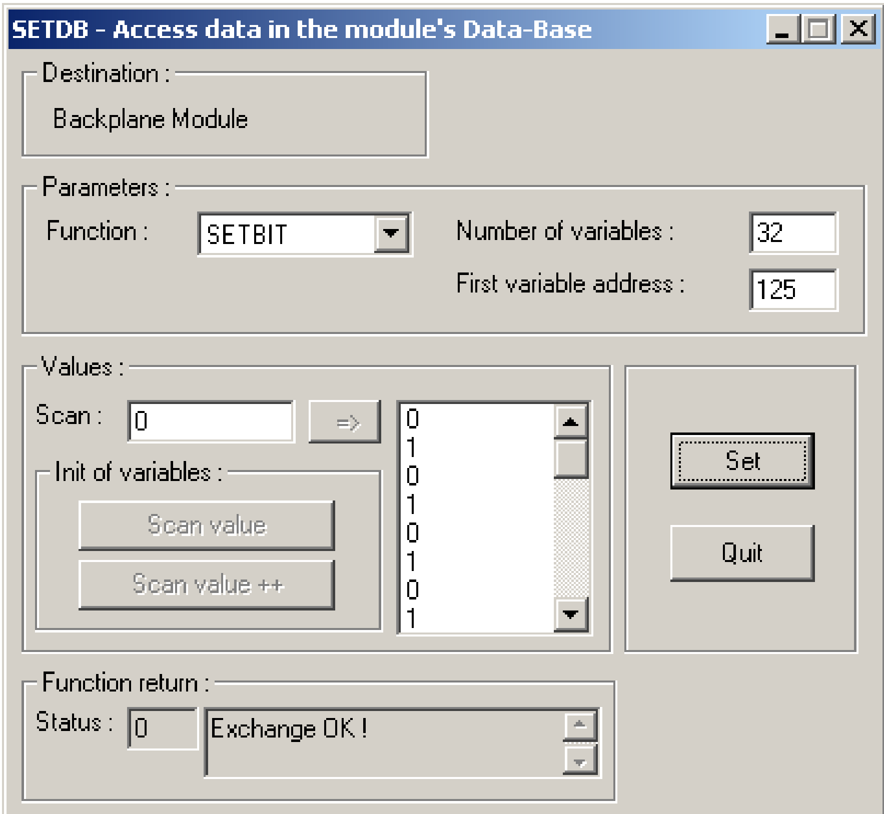
Figure 93: GETDB Dialog



7.1.3.2 SetDB

This tool is useful in writing data directly into the database

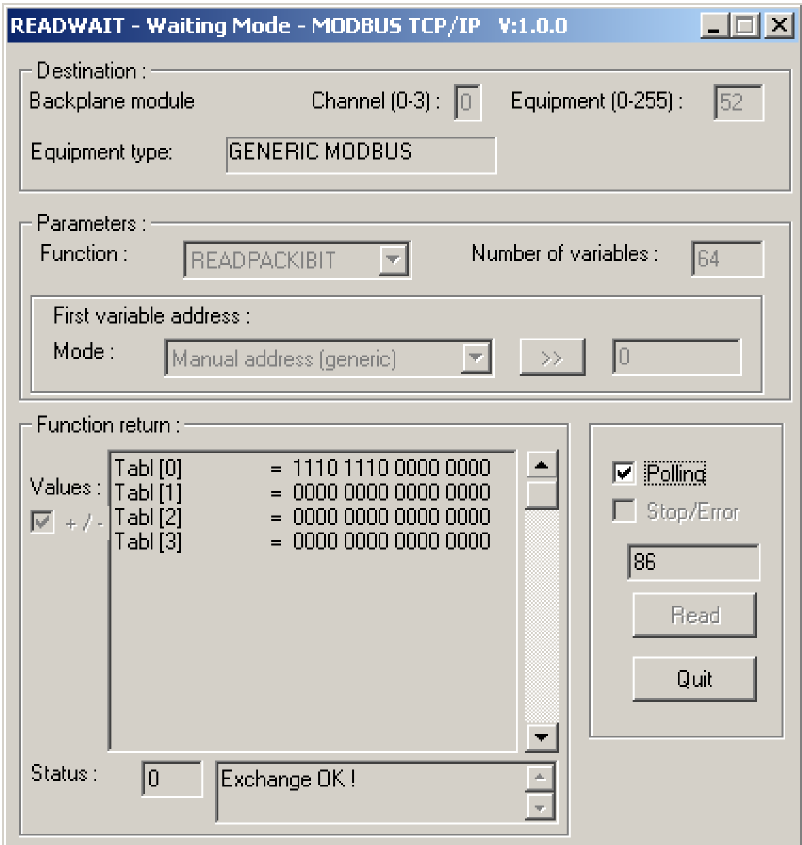
Figure 94: SETDB Dialog



7.1.3.3 ReadWait

This tool reads variables from a remote device

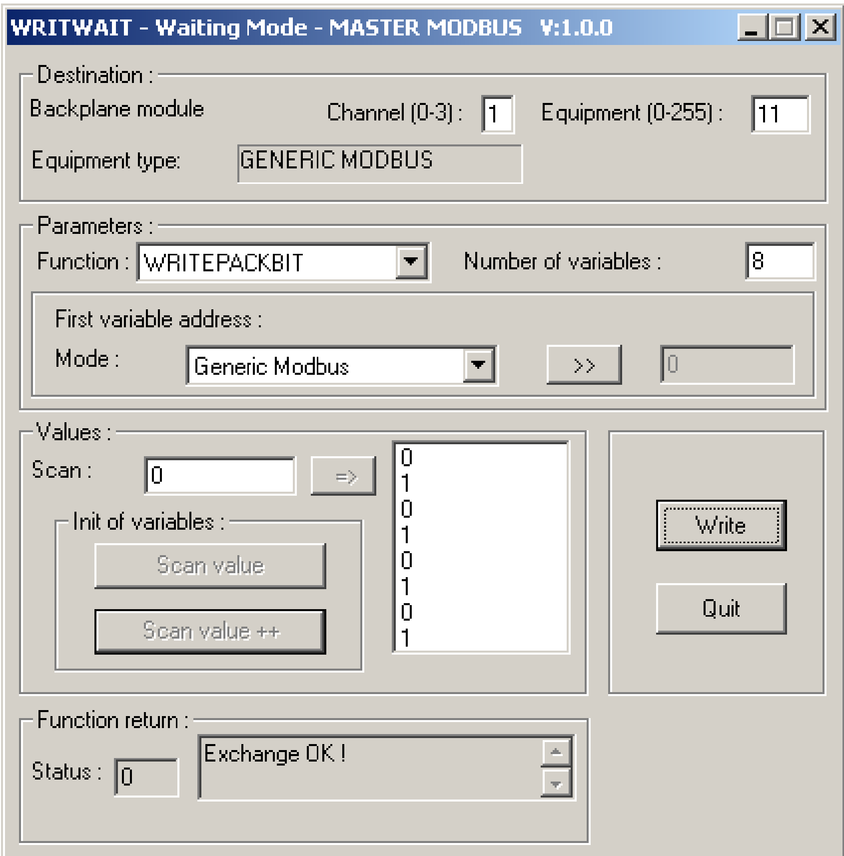
Figure 95: READWAIT Dialog



7.1.3.4 WriteWait

This tool writes variables to a remote device

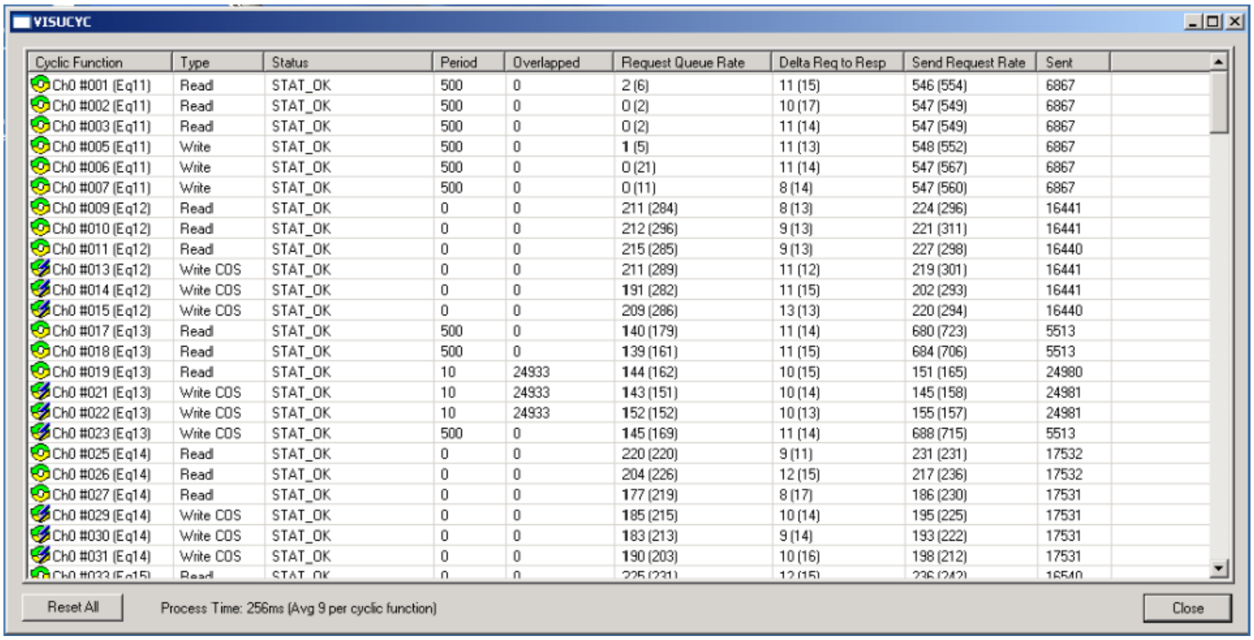
Figure 96: WRITWAIT Dialog



7.1.3.5 Visucyc

This tool monitors cyclic functions

Figure 97: VISUCYC



Cyclic Function	Type	Status	Period	Overlapped	Request Queue Rate	Delta Req to Resp	Send Request Rate	Sent
Ch0 #001 (Eq11)	Read	STAT_OK	500	0	2 (6)	11 (15)	546 (554)	6867
Ch0 #002 (Eq11)	Read	STAT_OK	500	0	0 (2)	10 (17)	547 (549)	6867
Ch0 #003 (Eq11)	Read	STAT_OK	500	0	0 (2)	11 (14)	547 (549)	6867
Ch0 #005 (Eq11)	Write	STAT_OK	500	0	1 (5)	11 (13)	548 (552)	6867
Ch0 #006 (Eq11)	Write	STAT_OK	500	0	0 (21)	11 (14)	547 (567)	6867
Ch0 #007 (Eq11)	Write	STAT_OK	500	0	0 (11)	8 (14)	547 (560)	6867
Ch0 #009 (Eq12)	Read	STAT_OK	0	0	211 (284)	8 (13)	224 (296)	16441
Ch0 #010 (Eq12)	Read	STAT_OK	0	0	212 (296)	9 (13)	221 (311)	16441
Ch0 #011 (Eq12)	Read	STAT_OK	0	0	215 (285)	9 (13)	227 (298)	16440
Ch0 #013 (Eq12)	Write COS	STAT_OK	0	0	211 (289)	11 (12)	219 (301)	16441
Ch0 #014 (Eq12)	Write COS	STAT_OK	0	0	191 (282)	11 (15)	202 (293)	16441
Ch0 #015 (Eq12)	Write COS	STAT_OK	0	0	209 (286)	13 (13)	220 (294)	16440
Ch0 #017 (Eq13)	Read	STAT_OK	500	0	140 (179)	11 (14)	680 (723)	5513
Ch0 #018 (Eq13)	Read	STAT_OK	500	0	139 (161)	11 (15)	684 (706)	5513
Ch0 #019 (Eq13)	Read	STAT_OK	10	24933	144 (162)	10 (15)	151 (165)	24980
Ch0 #021 (Eq13)	Write COS	STAT_OK	10	24933	143 (151)	10 (14)	145 (158)	24981
Ch0 #022 (Eq13)	Write COS	STAT_OK	10	24933	152 (152)	10 (13)	155 (157)	24981
Ch0 #023 (Eq13)	Write COS	STAT_OK	500	0	145 (169)	11 (14)	688 (715)	5513
Ch0 #025 (Eq14)	Read	STAT_OK	0	0	220 (220)	9 (11)	231 (231)	17532
Ch0 #026 (Eq14)	Read	STAT_OK	0	0	204 (226)	12 (15)	217 (236)	17532
Ch0 #027 (Eq14)	Read	STAT_OK	0	0	177 (219)	8 (17)	186 (230)	17531
Ch0 #029 (Eq14)	Write COS	STAT_OK	0	0	185 (215)	10 (14)	195 (225)	17531
Ch0 #030 (Eq14)	Write COS	STAT_OK	0	0	183 (213)	9 (14)	193 (222)	17531
Ch0 #031 (Eq14)	Write COS	STAT_OK	0	0	190 (203)	10 (16)	198 (212)	17531
Ch0 #033 (Eq15)	Read	STAT_OK	0	0	225 (231)	12 (15)	236 (242)	16540

Each column displays a different timing and configuration parameter of the cyclic functions:

__Cyclic Function Column__: Shows the Channel number, Cyclic Function number and Equipment number. The cyclic function icon also displays the configuration of the cyclic function e.g. Activated, Deactivated, Activated Change of State, and Deactivated Change of State.

__Type Column__: Shows whether the cyclic is Read or Write cyclic function. It also shows if write cyclic function is active on Change of State or Deactivated on Change of State.

__Status__: Shows the current status of cyclic function

__Period__: Shows the periodic polling interval set for the cyclic functions

__Overlapped__: Shows the number of times the periodic polling interval expires and the previous request is still on the queue waiting to be processed and sent.

__Request Queue Rate__: Shows the timing where the request has been placed into the queue waiting to be sent out to the network

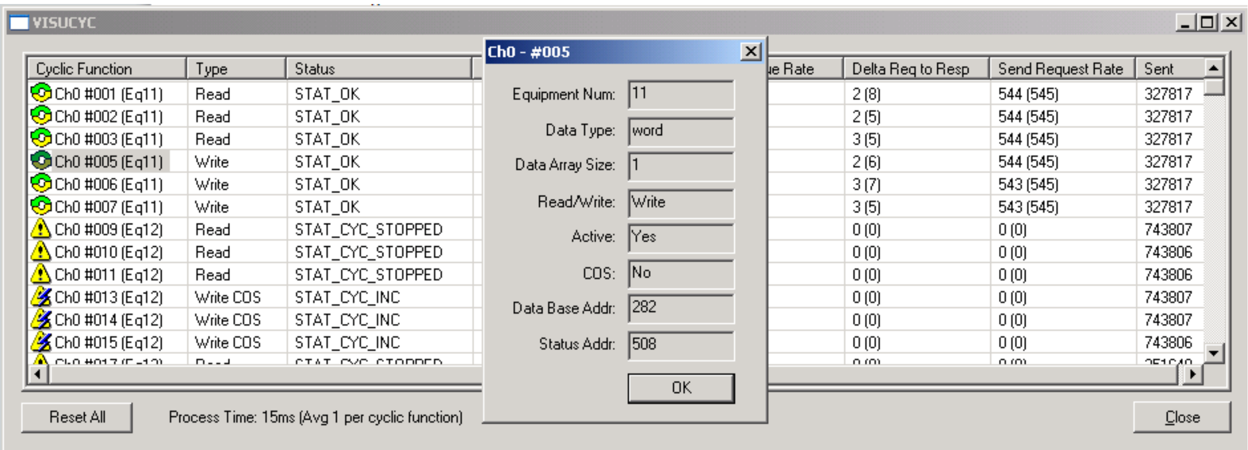
__Delta Req to Resp__: Shows the actual turnaround time between message request and message reply from the cyclic function manager task.

__Sent Request Rate__: Shows the sent request rate of a particular message/cyclic function which includes poll time and queue time rate.

<u>Sent Column:</u> Shows the number of times a message request was sent to the server equipment.

The detail of each cyclic function will be displayed by double clicking the desired cyclic function from the *<u>Cyclic Function Column</u>*

Figure 98: VISUCYC with detail of Cyclic Function displayed



7.1.3.6 Apsym

This application is used to view the different task version running the protocol processor.

Shortcut name: Protocol Information View.

7.1.3.7 Network Diagnostic Tool

For some protocols, a network diagnostic tool is supplied with the Backplane Solution**.** This tool offers valuable assistance in validating the correct operation of your Backplane Module.

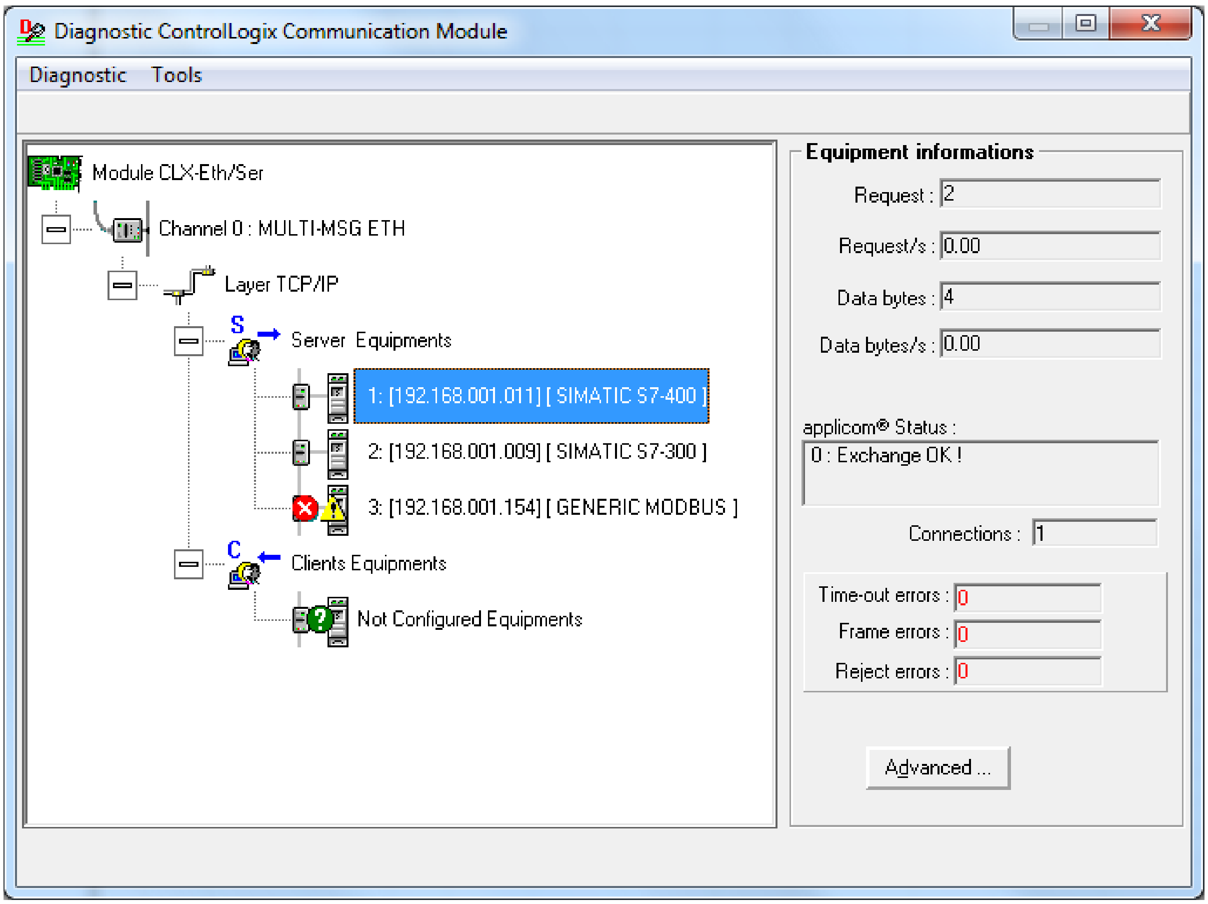
The diagnostic tool is accessed from the Console's Menu bar and offers the following features:

- Depending on the protocol and transport type, it's a specialized tool for analyzing the network. It gives the real-time traffic information (e.g., occupation rate, number of transmitted requests, bitrates), and has two consultation modes: Normal and Expert (File/Preferences/Expert Mode).
- The display shows a tree of communication objects (boards, channels, device), and animated diagnostic screens can be displayed for each object type. Dynamic symbols supply information on the current status of these objects (e.g., connected, active, in error, not configured).

Note

- *Depending on the configuration communication channel, the network diagnostic tool may not be available.*
- *If the Module is in Run mode, the diagnostic application may fail due to insufficient time to send commands to the Module through the RLL. The lower the RPI, the higher the chance of this failure mode. Timeout for mail messages is calculated by RPI * timeout multiplier.*

Figure 99: Communication Objects Tree



7.2 Database Mapping

The module can be configured in either of the two supported database configuration. The size and some control commands available for use is different in the two supported configuration. To understand the difference between the two database configuration and choose the right database addressing mode for the setup.

7.2.1 Default Address Mapping

The previous revision of the firmware (< v2.9.6) can only support the default address mapping in the database. This means that the first 700 words of the database are directly mapped to the ControlLogix IO table. The Input table is mapped to 2-249 words of the database, the Output table is mapped to 250-498 and the Status table is mapped to 500-699.

Any access to database address above 700 will require the use of CIP messaging. Our installation package provides sample ladder logic code to read and write data to the database using CIP messages. Refer to Start-Up Code Example section to check the ladder logic sample code.

When module is configured in Default database, there are two databases available for use to the network, the 32000 Words database and the 32000 Bits database. Bits database is used for bits data type when running Modbus protocol.

If the previously created configuration using firmware version < 2.10.2, the command and state words used for dynamic management of cyclic functions are located in 32200 (Bits database) and 32100 (Words database) respectively.

If the created new configuration using firmware version 2.10.2 or higher, the command and state words used for dynamic management of cyclic functions are located in 30720 (Words database) and 31744 (Words database) respectively.

The database has some fixed mapping when in configured in Siemens Server configuration. Below is an illustration how PLC variables is mapped in our module when running as server equipment.

INPUT TABLE (2-249)

S7/S5 PLC Variable mapping
AW0-AW246 / AB0-AB246 (PUT/GET)

OUTPUT TABLE (250-497)

S7/S5 PLC Variable mapping
EW0-EW246 / EB0-EB246 (GET)

STATUS TABLE (500-699)

Database Address (768-1023)

S7/S5 PLC Variable mapping
MW0-AW246 / MB0-MB246 (GET/PUT)

Database Address (1024-30207)

S7/S5 PLC Variable mapping

DB4.DBW0-DB116.DBW255 (PUT/GET)

To understand the Siemens PLC Variable addressing, refer to our 717-0055___Industrial Ethernet Protocol Reference Guide.

7.2.2 Extended Address Mapping

This new feature is added into the product feature set in firmware revision 2.10. Extended addressing mode provides direct access to the modules 30208 Words database. It provides the flexibility to map the entire 30208 Words database or select a location applicable to the network and map it to the ControlLogix.

The mapping should be done in two locations.

- Console Database Configuration (Refer to Changing to Extended Addressing Mode)
- And in the AOI CLX2000 configuration. (Refer to sample configuration that uses Extended addressing mode for)

AOI ladder logic code is used to directly access the mapped database to the ControlLogix. For the AOI to work properly, it verifies if the mapping in both location matches. Mismatch mapping will result to AOI unable to perform proper data exchanges.

The extended addressing mode also allows dynamic management of cyclic function using the mapped Command and State word in the database. Database address 30720-31743 is reserved for cyclic function command words. Database address 31744 – 32767 is reserved for cyclic function state words.

The database has some fixed mapping when in configured in Siemens Server configuration. Below is an illustration how PLC variables is mapped in our module when running as server equipment.

Unmapped Database Address
(0000-4864)

DB1.DWx – DB19.DWx

INPUT Table (5120-9999)

AW0-AW255 / AB0-AB255 (GET/PUT)
MW128-MW255 / MB128-MB255 (GET/PUT)
DB20.DWx – DB38.DWx (GET/PUT)

OUPUT Table (10000-19999)

S7/S5 PLC Variable mapping
EW0-EW255 / EB0-EB255 (GET)
MW0-AW127 / MB0-MB127 (GET)
DB39.DWx / DB78.DWx (GET)

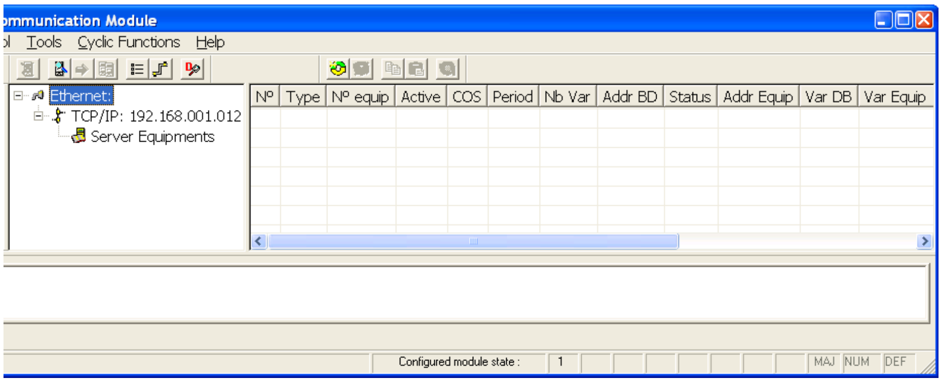
Unmapped Database Address
(20000- 30207)

S7/S5 PLC Variable mapping
DB79.DBW0-DB116.DBW255

To understand the Siemens PLC Variable addressing, refer to our 717-0055___Industrial Ethernet Protocol Reference Guide.

7.3 Cyclic Function

Figure 100: Main Cyclic Function Configuration Screen



7.3.1 Cyclic Function User Interface Commands

The following commands can be used for configuring cyclic functions.

Table 37: User Interface Commands

Menu	Icon	Actions
Cyclic Function/Add		Inserts a cyclic function into the selected device
Cyclic Function/Delete		Deletes the selected cyclic function
Cyclic Function/Copy		Copies the selected cyclic function
Cyclic Function/Paste		Adds a copy of the cyclic function to the selected device
Cyclic Function/Properties		Displays the Properties dialog box for the selected cyclic function

Copy/Paste

To copy or paste a function line:

1. Select the line.
2. From the Cyclic Functions menu, choose the **Copy** command (or click the corresponding button).
3. From the Cyclic Functions menu, choose the **Paste** command (or click the corresponding button). The Configuration dialog box is displayed, enabling you to make any necessary changes.
4. Click the **OK** button to finish copying the cyclic function.

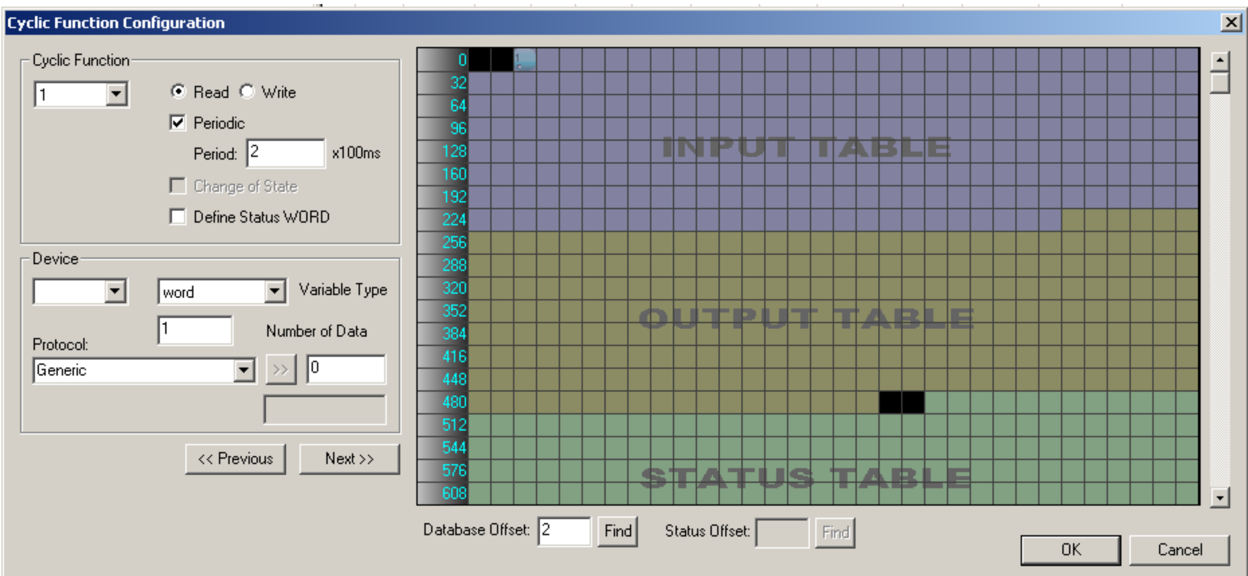
7.3.2 Cyclic Function Configuration Parameters

To access a cyclic function's configuration parameters, select the appropriate line and do one of the following:

- Choose the command from the Cyclic Functions menu
- Click on the corresponding button in the Menu Bar

The Cyclic Function Properties Screen is displayed.

Figure 101: Cyclic Function Properties Screen



This screen is used to enter information for a cyclic function.

Table 38: Cyclic Function Parameters

Parameter	Actions
Function Number	Used to choose an available cyclic function number from a list. The number acts only as an identifier. The number of cyclic functions available for each communication channel is 255.
Function Type	**Read** Variables are read from the device and stored in the Backplane's database. **Write** Variables are read from the Backplane's database and sent to the device. This field is mandatory.
Periodic	This mode allows the cyclic function to be triggered periodically. To enable this behavior, set the «Activation Mode / Periodic» field to «yes». **yes**: The cyclic function will be active after module initialization. **no**: The cyclic function won't be active after module initialization. **Note:** It's possible to modify the activation mode of the cyclic functions of each channel during runtime. For details, refer to Dynamic Cyclic Function Management
Period	Period is used in both of the following cases: **Function with periodic activation mode** Regardless of the Activation mode/OnvalueChange field's value, the Period field defines the triggering period for the cyclic function, as a multiple of 100ms. **Write function with On Value Change activation and not periodic (field activation mode: Periodic set to no)** The period is used for communication error management. If a value changes in the database area covered by the cyclic function, and the triggered cyclic function returns a communication error status, the cyclic function will be automatically activated at the configured period until the communication can be restored and the value(s) can be sent to the remote device. Be careful to not set this period too short, otherwise the network may overload. The period is a multiple of 100ms. Example: For a period of 1 second, enter a value of 10 (10 * 100ms = 1 second). The maximum period is 65535 * 100ms (about 1h 50mn). **Notes:** This periodicity is only observed if: • The overall execution time for all active functions is less than the period defined • The network characteristics allow it (Ex : PLC cyclic time, network load…) If the time is greater, the cyclic functions will be chained without an intervening pause. If the period is set to 0, the cyclic function is retriggered as fast as possible

Parameter	Actions
Change of State(COS)	This mode is only possible for a write cyclic function. It allows a cyclic function to be triggered if one or several values are modified in the cyclic function's database area. To enable this behavior, the «Activation Mode / On Value Change (COS)» field must be set to "yes". **yes**: The cyclic function will be triggered each time a value changes in the cyclic function's database area. **no**: The cyclic function won't be influenced by the value of the database area. **Notes:** It is possible to modify the activation mode of the 128 first cyclic functions of each channel afterwards, during runtime. For details, refer to For details, refer to Dynamic Cyclic Function Management. This mode can be combined with the Periodic Activation mode. For details, refer to Cyclic Function Triggering Mode.
Device Number	Refers to the devices configured in the Console. The value range is from 0 to 127. This field is mandatory.
Device Variable Type	Allows the type of variable in the remote device to be selected
First Data Address	Address of the first variable in the device. The address can be entered manually (when "generic" is selected), and in a few cases, it must be calculated manually, as well. The calculation rules are in the protocol manual. It is, however, possible to select a manufacturer address descriptor that corresponds to the device. In this case, a manufacturer mnemonic may apply..
Number of Data in Device	Refers to variables defined in Device variable types. The number of data items in the device and database can be different if the variable type defined in Device variable type differs from that defined in Database variable type. The maximum number of variables accessed depends on the protocol and the connected device.
Database Offset	The address of the first variable in the module's database. The address must range from 0 to 30207. Based on the Database Configuration, assign the proper offsets for the Input, Output and Status
Status Offset	Used to define a storage address for the data exchange status word in the Backplane's database. The status of this word ensures that the data exchange is carried out correctly (it is updated by the cyclic function when the data exchange status changes). The storage address must be between 0 and 30207. To activate this option, check the "Define Status WORD" box.

7.3.3 Cyclic Function Triggering Mode

There are two different automatic triggering modes:

- Periodic
- On Value Change (also called Change Of State, or COS). This mode is only possible for write cyclic functions.

In addition, a cyclic function (read or write) can be triggered on external command via a **State word** and a **Command flag**. For details, refer to Section 7.3.7, Dynamic Cyclic Function Management.

The following diagrams show the different ways of triggering cyclic functions after the Backplane initializes.

Figure 102: Periodic Mode Only (Read or Write Cyclic Function)

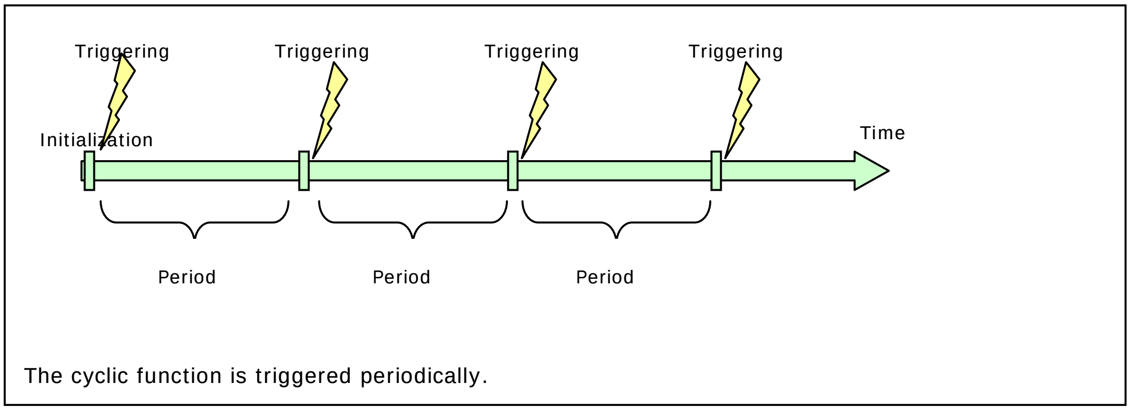


Figure 103: On Value Change Mode Only (Write Cyclic Function)

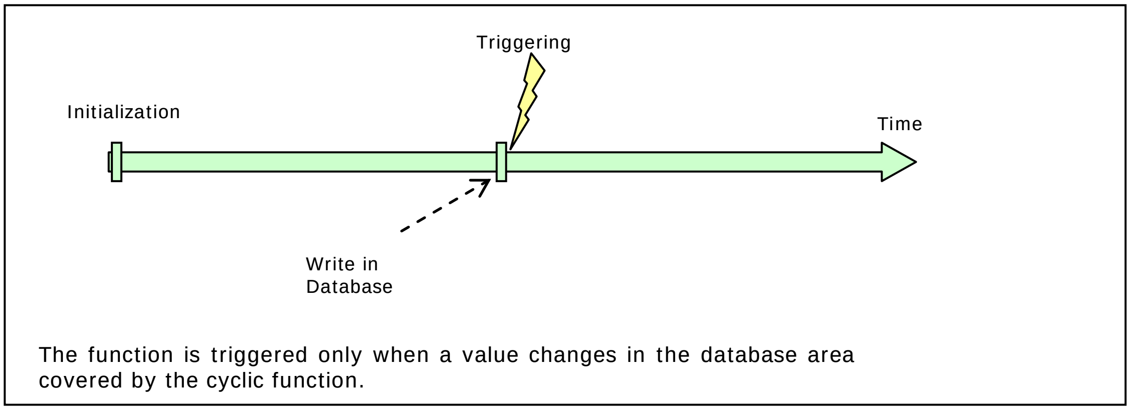
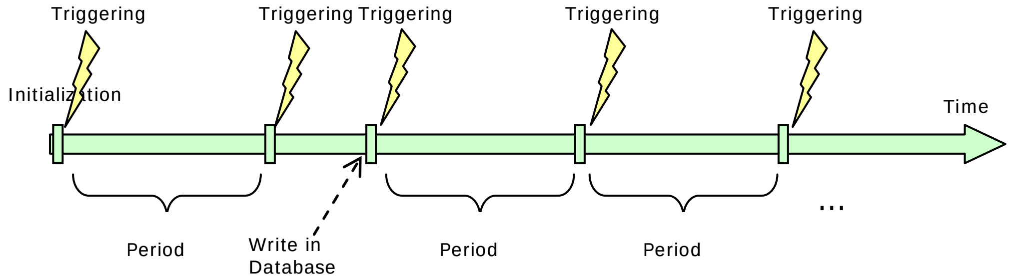
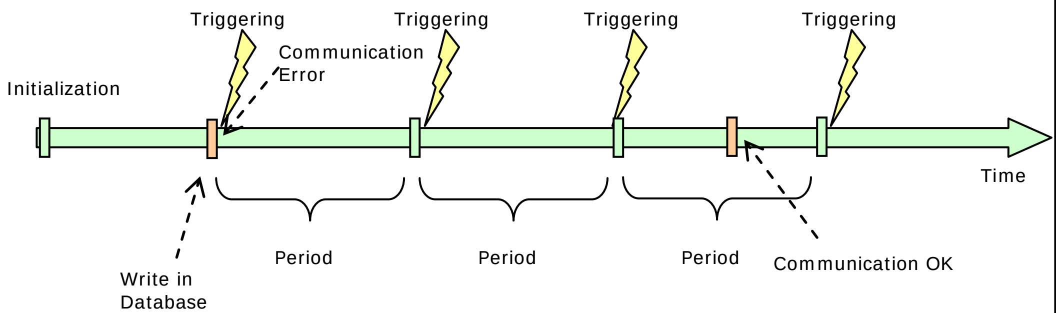


Figure 104: Periodic and On Value Change Modes (Write Cyclic Function)



The function is triggered both periodically and after each write operation (with a different value) in the cyclic function's database area. After the trigger occurs, the periodic cycle is reset.

Figure 105: On Value Change Mode with Communication Errors (Write Cyclic Function)



When a COS cyclic function returns a communication error, the function retriggers periodically, until the error disappears. Afterwards, the normal COS mode is reactivated.

The Configuration Console displays a list of all configured cyclic functions and their parameters.

Figure 106: Configured Cyclic Functions

N°	Type	N° equip	Active	COS	Period	Nb Var	Addr DB	Status	Addr Equip	Var DB	Var Equip	Equipment	Syntax
1	Read	1	Yes	No	2	8	2	500	0	word	input byte	Siemens Simatic S7	IB0
2	Write	1	Yes	No	2	10	250	Undefined	0	word	output word	Siemens Simatic S7	AW0
3	Write	1	Yes	Yes	2	6	260	501	655360	word	word	Siemens Simatic S7	DB10.DBW0
4	Write	2	No	No	2	10	266	Undefined	1	word	word	Generic Modbus	W1
5	Write	2	No	Yes	2	20	276	Undefined	100	word	word	Generic Modbus	W100

The displayed icons depend on the function's triggering mode:

: Read or Write function with periodic triggering mode

: Write function with both periodic and COS triggering mode

: Write function without triggering mode

: Write function COS triggering mode

7.3.4 Cyclic Function Status

The following table describes the status related to the triggering mode. For more information on the communication status, refer to the relevant protocol manual.

Table 39: Cyclic Function Status

Status	Description
-6	The cyclic function has no triggering mode configured
0	The cyclic function is configured in periodic triggering mode and/or COS triggering mode and the first data exchange has been performed without communication.
35	The cyclic function is configured in periodic triggering mode and/or COS triggering mode and the first data exchange has not completed yet. This is a general status that appears mainly after the Backplane initializes.
255< status < 100	The cyclic function is configured in periodic triggering mode and/or COS triggering mode and a communication error occurred during the last exchange. For more details on the communication status, refer to the relevant protocol manual.
256 (0x0100)	The cyclic function has no triggering mode configured (inactive), but has been triggered via “one shot” successfully (no communication error), using the Status word and the Command flag. For details, refer to Dynamic Cyclic Function Management. The MSB indicates that the status has been returned after a “one-shot” trigger; therefore, it could be obsolete. The communication status resides in the LSB (which, in this case, is equal to 0).
Others status > 256 (0x01xx)	The cyclic function has no triggering mode configured (inactive) but has been triggered via “one shot” unsuccessfully (communication error), using the Status word and the Command flag. For details, refer to Dynamic Cyclic Function Management. The status MSB indicates that the status has been returned after a “one shot” trigger; therefore, it could be obsolete. The communication status is contained in the LSB. For more information, refer to the relevant protocol manual.

Note

For more cyclic status information, refer to section 8.1 General Status Messages

7.3.5 Cyclic Function Communication Status Management

A frequent use scenario for the module is to map together:

- A read cyclic function retrieving data from Device X into a database area
- A write cyclic function sending this data from the database area to Device Y

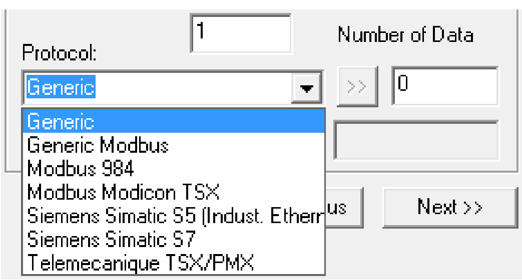
By default, if communication with Device X fails, Device Y does not detect the error.

This problem can be solved by configuring the status address of the read cyclic function, so that the status is stored contiguously with the retrieved data. This enables the write cyclic function to easily integrate the communication status and data for Device Y.

7.3.6 Cyclic Function Device Addressing Mode

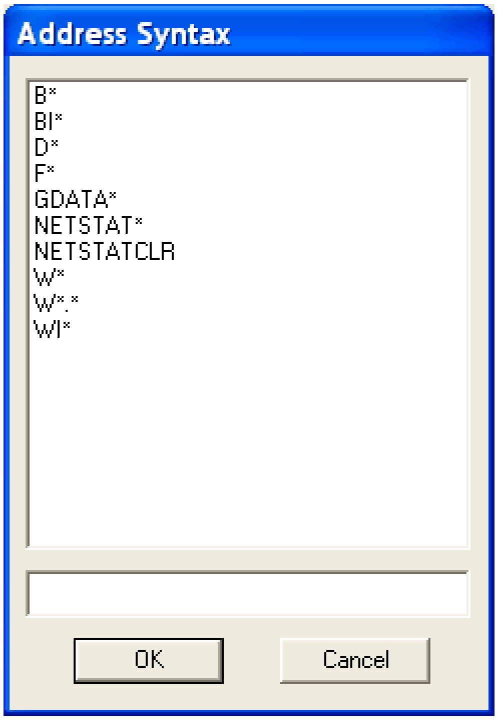
In the Configuration dialog box, click on the **Protocol** drop-down menu, and select one of the following options:

Figure 107: Protocol Drop-Down Menu, Generic Selected



- **Generic** mode: calculate the physical address manually and enter it in the rightmost field (the **>>** button is not available).
- Other modes: Each of these corresponds to a specific protocol device type. When select one of these modes, the >> button becomes active. Click on it to configure the address of the first variable in the device. The following dialog box is displayed:

Figure 108: Address Syntax Dialog



Select the addressing syntax for using in the configuration.

When confirm the dialog box, the **First device data address** field automatically populates with the corresponding address. The relevant syntax is displayed on the right side of the dialog box.

Example

For a MODBUS device, the W0 syntax will initialize the **First device data address** field.

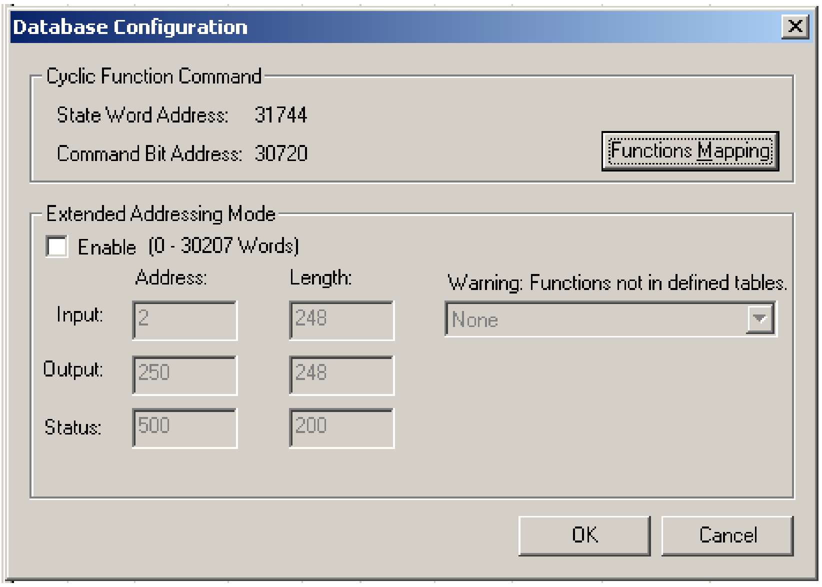
7.3.7 Dynamic Cyclic Function Management

Set the triggering mode of each channel's 255 cyclic functions to be dynamically modified during runtime. An area of 256 command flags (bits) and an area of 256 command words for each channel are reserved in the Backplane's database for this purpose.

The command flag area starts at address 30720 of the database, and the command word area starts at address 31744 of the database word area.

Click the **Functions Mapping** button in the Database Configuration dialog box, to generate a printable file that contain each existing cyclic function in the Configuration Console, the addresses of the corresponding command word and the command flag.

Figure 109: Database Configuration Screen



Note

Configuring the cyclic functions before setting the areas' addresses is recommended.

By setting a cyclic function's **command flag** bit to 1, the triggering mode will be modified, depending on the **state word** value:

Table 40: State Word Values

State Word	Action
0	One-shot triggering of the cyclic function
1	Periodic triggering activated
2	Periodic triggering deactivated
3	On Value Change (COS) triggering activated
4	On Value Change (COS) triggering deactivated

Table 41: Command Flag Values

Command Flag	Action
Set to 0	Nothing happens
Set to 1	Action corresponding to the state word is executed
Return to 0	Execution confirmation. The flag is automatically reset once the action has been taken into account. In the case of «one shot» activation, the flag is reset only after the cyclic function has been executed. It therefore allows synchronization to take the retrieved data into account.

Note

Modifying the triggering mode to activate the periodic or COS mode will immediately cause the cyclic function to trigger for the first time. The following diagram explains this behavior.

Figure 110: Activation of Periodic Triggering Mode

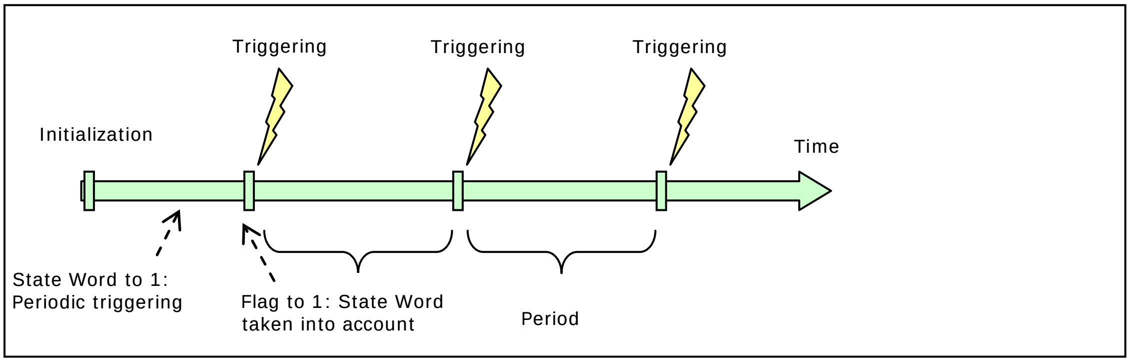


Figure 111: Activation of On Value Change (COS) Triggering Mode

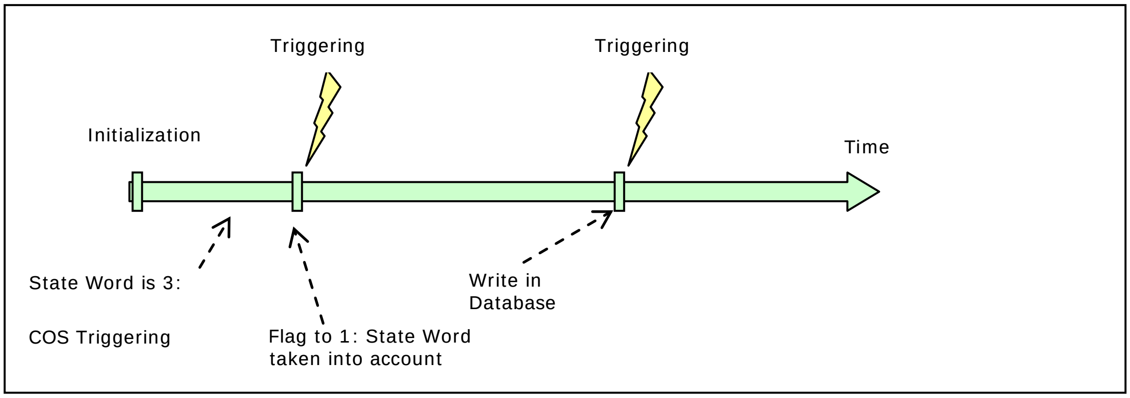
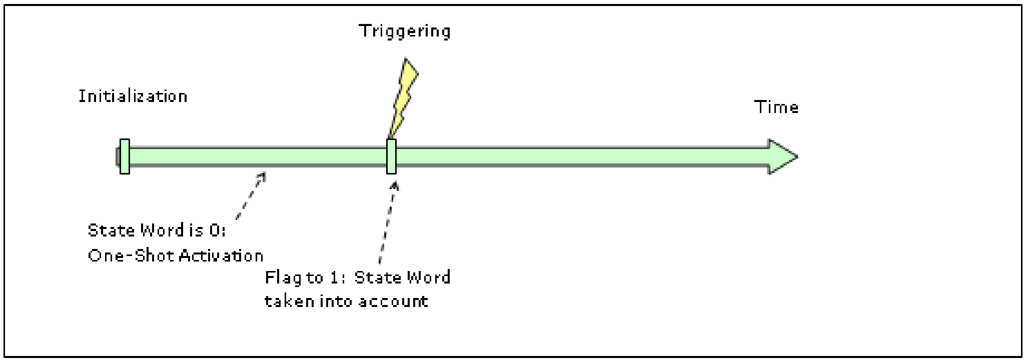


Figure 112: One-Shot Activation



7.4 Initialization Principle and Methods

The **Backplane Solution** offers two types of initialization, each meeting particular conditions of use:

- **Remote** initialization. This can be run at any time (via an Ethernet connection or through RSLinx) on user request from the configuration Console. The configuration will be downloaded to the Backplane and will also be used for subsequent automatic initializations.

- **Automatic** initialization. This is carried out automatically whenever the Backplane is rebooted. The last remotely downloaded configuration is used.

Note

When the Backplane is first turned on, no configuration is present, so the automatic initialization process is not activated. The Backplane waits for its first remote initialization, a state which can be seen by the Comm LED (permanently red). The Module does not allow a connection from a CLX CPU, so if a connection is attempted, the display indicates that the connection is blocked.

7.4.1 Remote Initialization

Remote initialization can be carried out at any time upon user request, via a special utility (Pcinit.exe). Its role is to transfer a saved configuration to the Backplane.

The utility can be run from the configuration Console by clicking on the icon: **.**

Following remote initialization:

- The downloaded configuration is saved within the Backplane and is used by the automatic initialization process
- The configuration is implemented. Communication on the Ethernet and Serial networks starts.

Depending on the level of detail requested (**Detailed actions** check box), the utility's graphic interface can take the following forms:

Figure 113: Without Details

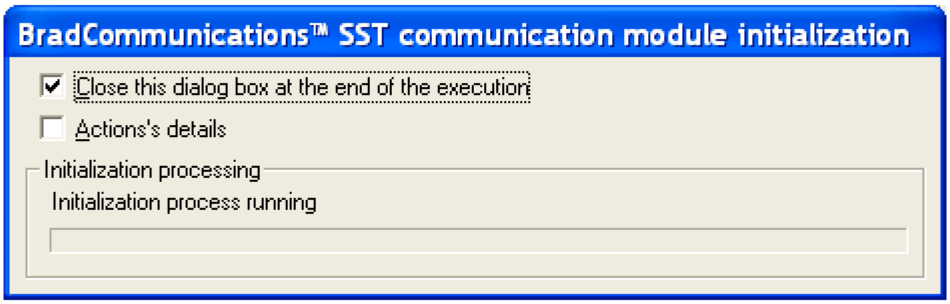
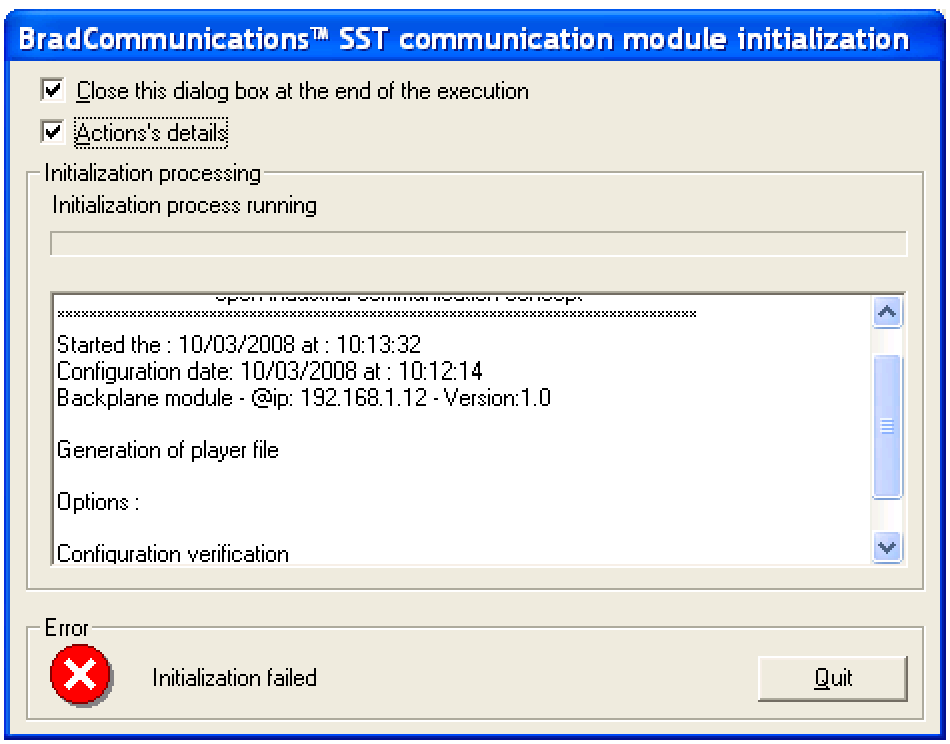
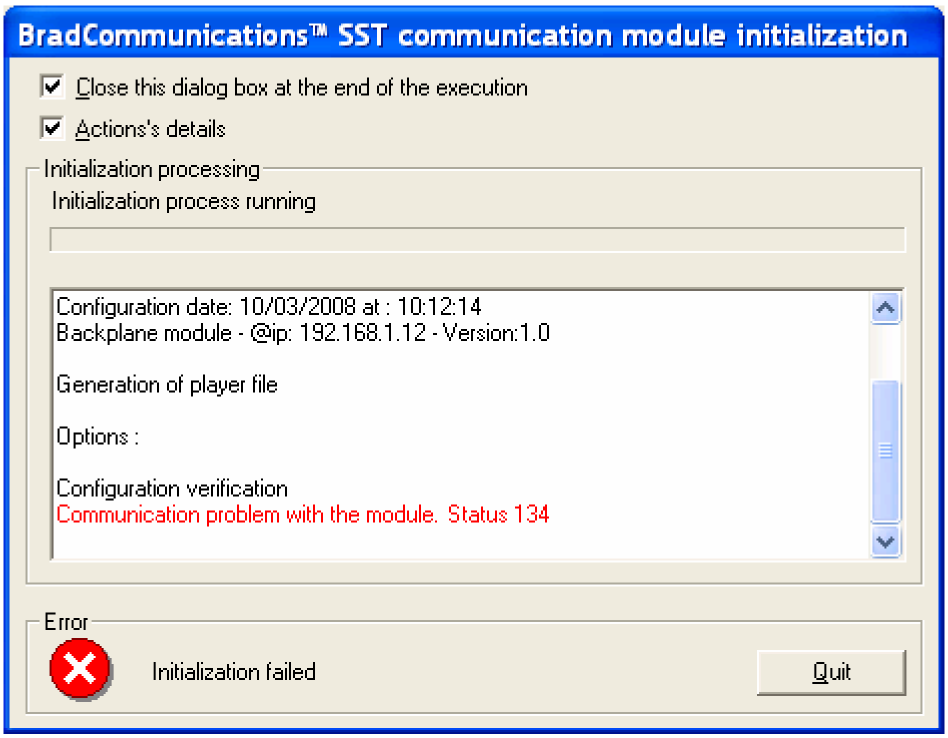


Figure 114: With Details



If a problem occurs, the interface automatically switches over to **Detailed actions** mode and displays an error message.

Figure 115: With Error



It is recommended to solve the problem before attempting any further initialization. For more information, refer to section 8.1 General Status Messages.

Note

- *The remote initialization process is the only one that can change the Backplane configuration. It should therefore be used during the development phase of your architecture.*
- *All messages produced during remote initialization are stored in the **pcinit.log** text file, located directly in the subdirectory containing the active configuration (config/"Configuration name"/pcinit.log)*
- *To start the remote initialization process, all other software tools that communicate with the Backplane must be closed.*
- *The configuration files are found in: C:\Documents and Settings\All Users\Application Data\BradCommunications\ SST Backplane Communication Module\config\....*
- *If the Module is in Run mode, the Pcint may fail due to insufficient time to send commands to the Module through the RLL. The lower the RPI, the higher the chance of this failure mode.*

7.4.2 Automatic Initialization

Automatic initialization is carried out every time the Backplane is turned on, providing that a remote initialization has previously been carried out.

Following an automatic initialization, the Backplane is in an identical state to that when the last remote initialization was carried out.

7.4.3 Serial Port Wiring

The Module contains a standard RJ45 connector with a DB9 adaptor, which can be connected to the configured Serial network. There is no built-in termination, so it needs to be provided. For more details, refer to the Protocol manual.

Figure 116: RJ45 and DB9 Pin Numbering

SST-SR4-CLX-RLL	SST-ESR2-CLX-RLL
4 x RJ45 to DB9 Male	2 x RJ45 to DB9 Male

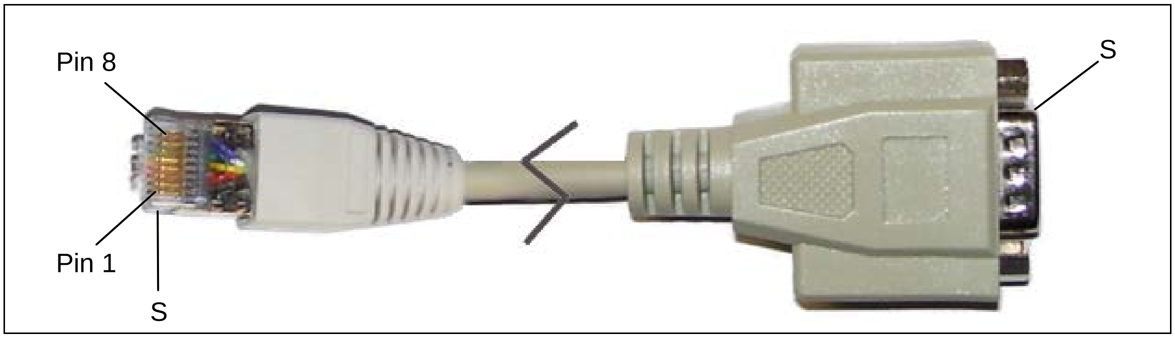


Table 42: Wiring Instructions

RJ45	DB9	Function		
Pin #		**232**	**485**	**422**
1	1	N/C	TXRD +	TXD+
2	2	RXD	N/C	RXD +
3	3	TXD	N/C	N/C
4	4	N/C	N/C	N/C
5	5	GND	GND	GND
6	6	N/C	N/C	RXD-
7	7	N/C	N/C	N/C
8	8	N/C	TXRD -	TXD -
-	9	N/C	N/C	N/C
S	S	SHIELD		

7.4.4 LED and Display States

The following tables show how the LEDs indicate the Module's operation.

Table 43: COMM LED Behavior

LED State	Meaning
Solid Green	Communication OK on all ports.
Flashing Red	Errors exist on one or more channels.
Solid Red	Network errors; no communication. Check your cabling termination and configuration.

Table 44: SYS LED Behavior

LED State	Meaning
Solid Green	Connection open; scanning all devices in Run mode.
Solid Red	Connection lost; waiting to re-establish.
Amber	Waiting for connection to open for the first time.

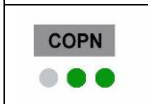
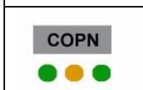
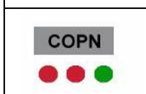
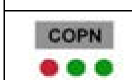
Table 45: OK LED Behavior

LED State	Meaning
Solid Green	Initialization is complete and Module is ready.
Flashing Red	Fatal error. Call Technical Support with the information on the display.
Solid Red	Watchdog timeout error.

7.4.5 LED and Display Combinations

From left to right, the LEDs are COMM, SYS and OK. While a connection is open, COPN shows on the LCD display.

Table 46: LED and Display Combinations

COPN State	Meaning
 COMM, SYS and OK LEDs are green.	All channels are communicating. Connection is open.
 SYS and OK LEDs are green; COMM is off.	Module is OK. Connection is open and there are no active cyclic functions.
 COMM and OK LEDs are green; SYS LED is amber.	Module is OK. Connection has never been made by the CLX CPU.
 COMM and SYS LEDs are red; OK LED is green.	Module is OK. Connection has been lost and no communication on any channels. The display shows an error or connection closed “CCSD”.
 SYS and OK LEDs are green; COMM LED is red.	Module is OK. Connection is open but configuration on Module is bad or in error.
 COMM and SYS LED don’t care; OK LED is red.	Module is indicating a watchdog, or if flashing, a fatal error.

7.4.6 Display Tables

The following tables show how the display indicates the Module's operation.

Table 47: Display Messages

Display Message	Meaning/Description
COPN: [Config Name]	Connection open for I/O data transfer. The [ ] represent the name of the configuration.
OK: [config Name]	Module waiting for first connection.
CONNBLKD	Connection blocked. An error occurred while forwarding the open request.
DUPLCONNDTCT	Duplicate connection detected.
RSRCERR1INCM	Resource issue while forwarding the open request.
NVLDFWD OPEN	Invalid open command forwarded.
NVLDSRVCCODE	Unsupported service request.
NVLDXMITSIZE	Invalid connection size; does not match existing connection.
NVLDOUT CNPT	Bad Output connection point.
NVLDIN CNPT	Bad Input connection point.
NVLDCFG CNPT	Bad Configuration connection point.
NVLDRCV BUFF	Invalid receive buffer.
BUS_INIT	Initializing the bus between PPC and NIOS.
AB_INIT	Initializing the CLX Stack.
CCSD: [Config Name]	Connection closed.
RESET	Resetting the Module.
FE:……..	Fatal error; all tasks have been killed, except for the display task.
WDOG	Watchdog has been kicked. Module has gone into reset.
Backing up config	Backing up configuration on card to USB Stick
Updating	Transferring configuration from USB stick to Card

Note

When the Module is working correctly, the default LCD display is COPN, followed by the configuration name.

7.5 Modifying the TCP Interface Configuration of SST-ESR2-CLX-RLL Product Variant

Using the default configuration of the TCP interface should allow supported protocols (currently Modbus TCP and Siemens S5/S7) to exchange message request/reply between client and server connection. However, the TCP interface can be modified if any of the listed requirements is applicable in the network.

- The network configuration requires a specific timeout for a message request
- The network configuration requires a specific retry interval in between message request that has been timing out.
- A specific number of retries is needed for a message that has been timing out.

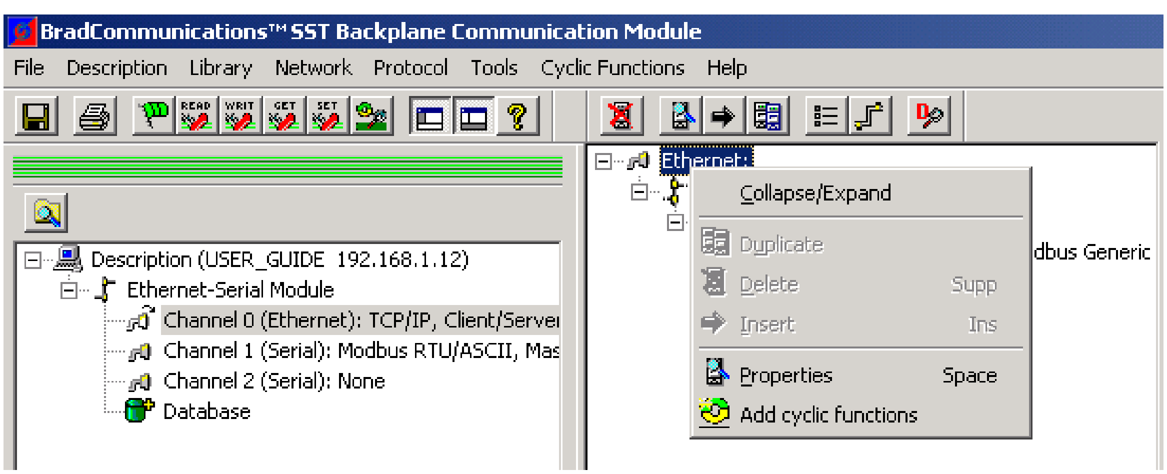
Note

This retry count applies to number of times the client will attempt to send the message that has timed out. Keep alive packet will be observed from the network when the client continue to send this retry.

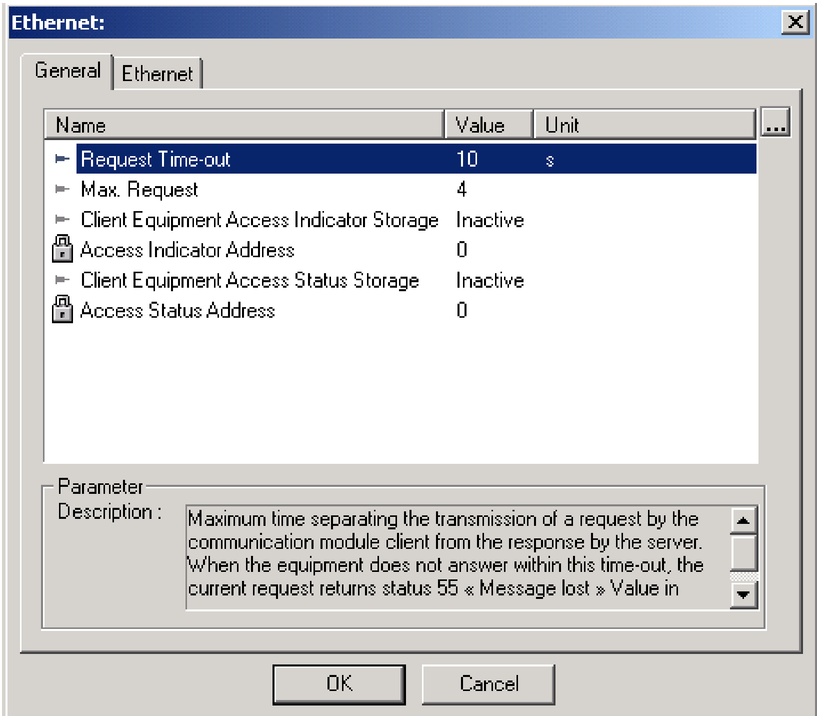
- When configured as server equipment, to monitor the client access count and the client access status. ensure that the database address is using is not being used by a cyclic function.
- Increase the maximum request that is being processed or sent.

To change the configuration, follow the steps below:

1. Right click on the “Ethernet” and click on Properties.

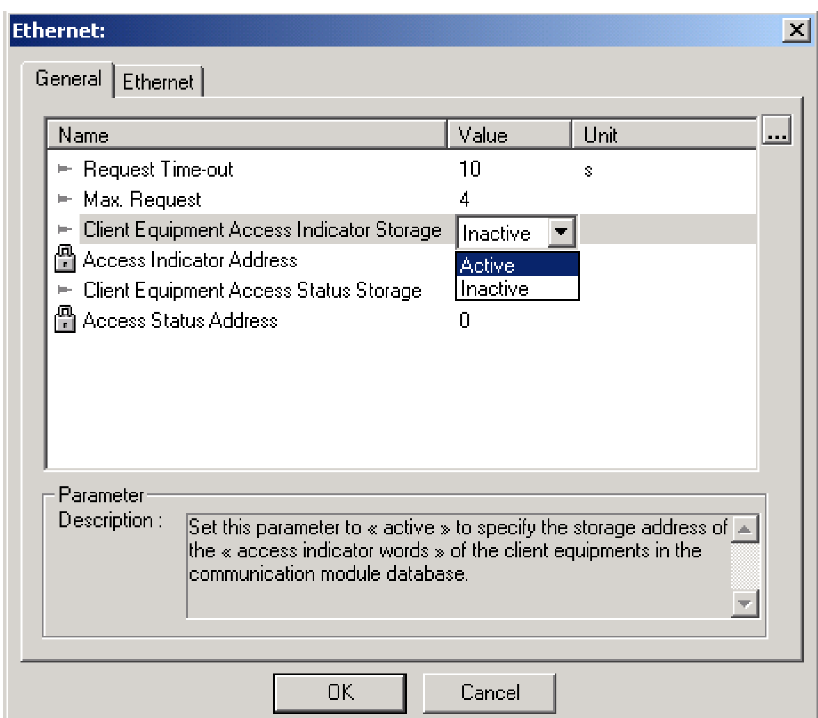


2. The Ethernet Properties will be shown similar to the diagram below.



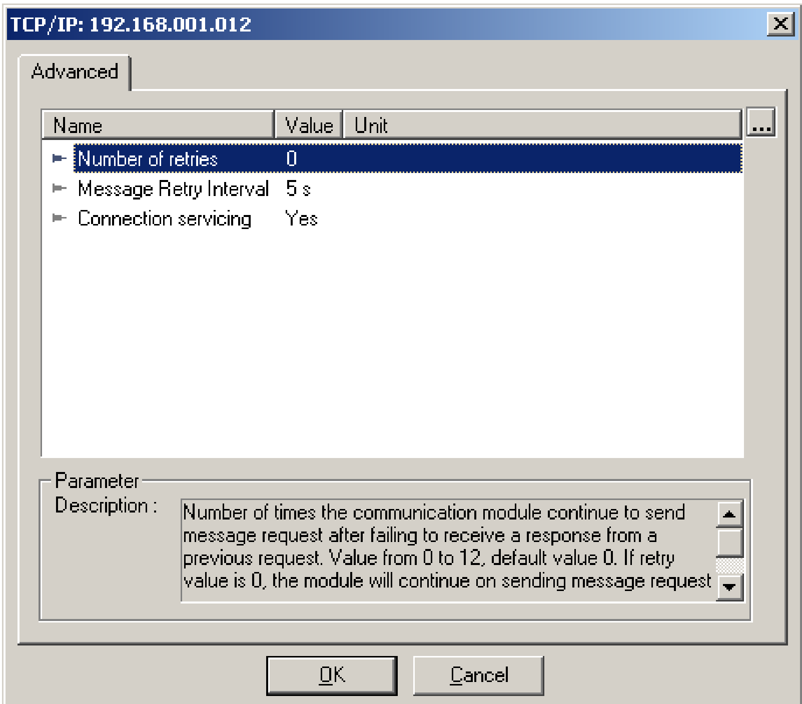
3. Modify the parameter that applies to the requirement.

4. To monitor the client access count and client access status, the parameter needs to be active first by selecting Active in the drop down menu as shown below.



5. Assign a location in the database where to store the status of the client accesses.

6. To change the number of retry and retry interval, right click on the TCP/IP and click Properties as shown below.



8

Communication Module Status Information

Chapter Sections:

- General Status Messages
- Communication Module's I/O and Status Messages
- Module Status Register
- Console Status Bar Information
- User Status Table
- Diagnostic Counters, Status and Information

8.1 General Status Messages

The following table lists the Backplane's general status messages.

Table 48: General Status Messages

Value	Status	Status Displayed in Visucyc
0	No anomaly detected. The function is executing correctly.	STAT_OK
1	Unknown function. The requested function is not supported.	STAT_ERR_FONC
2	Bad address. The address of the solicited variable is bad.	STAT_ERR_ADDR
3	Bad data. Check the address of the read or write request sent to the station.	STAT_BAD_FRAME
4	Inaccessible data.	STAT_LOCK_DATA
32	Bad parameter passed to the function. Bad number of variables (check # of bytes in your CIP message buffer). Request incomplete.	STAT_PAR
33	Message sent timed out	STAT_TIMEOUT
45	Communication task not loaded. Initialize the Backplane before use via the initialization utility, which can be accessed from the configuration Console.	STAT_NO_SOFT
46	Board number not configured, or the master/client functionality is targeting a channel configured as slave/server or vice versa.	STAT_CONF
49	Response timed out while a mail message was being sent to the Protocol CPU.	STAT_TIMEOUT_WAIT
47	Backplane absent.	STAT_NO_BOARD
51	Backplane access problem. Check the IP address or that the Backplane has finished starting up.	STAT_SYS
55	Too much time between requests and responses. Status 33 will occur first followed by 55 when no communication occurs at all	STAT_BADINTPOL
60	Mail message not returned by the Protocol CPU.	STAT_NO_RETMAIL
66	Not enough interface memory for the Backplane Solution.	STAT_RESSOURCE
135	There is no Connection to ControlLogix CPU. Cyclic function is configured to access ControLogix I/O tables.	STAT_CYC_STOP_CLX
-6	Cyclic function was stopped or deactivated. This happens when server equipment has not replied any request sent by the client after a few retries or cyclic function has been dynamically deactivated. See Section 6.6 for details.	STAT_CYC_STOPPED
35	Cyclic function configured in COS to a disconnected server equipment.	STAT_CYC_INC
36	Cyclic function is associated with an inactive device.	STAT_ESCL_INCONNU

8.2 Communication Module's I/O and Status Messages

The I/O interface consists of the following areas.

Note

When no connection to the Module exists, the Controller overwrites the first two words of the CLX status file and CLX input file with a status of 'connection lost' (-1) These words can still be configured for cyclic function status words.

When a connection is lost, all data in the status and input files is invalid. Before using this data, check your ladder code for the connection lost status.

Table 49: ControlLogix Controller Interface Image

Area Description	Offset	Area Layout	Description	Size
Input Data Area	0	Module Status	In both word locations, if this value is 0, the CLX CPU is connected to the module, and if it's -1, the connection has been lost. For more information, refer to DataExchangeGood and DataExchangeStatus, in the Status area below.	2 x 16-bits
	2	Input Data Area	Data in this table is read form word database offset 2 – 249. For AOP, data is read into AOP tags DataIn[0] to DataIn[247].	248 x 16-bits
Output Area	0	Output Data Area	Data in this table is written to word database offset 250 – 498. For AOP, Output data area starts at DataOut[0] and ends at DataOut[247]	248 x 16-bits
Status Area	0	Configurable Status Area	Data in this area is read from word database offset 500 – 699. This is filled by any cyclic function configured to write a Status value. For AOP, this area consists of AOP tags ConfigurableStatus[0] to ConfigurableStatus[199]	200 x 16-bits
	200	TransactionCount	This counter is incremented every time the I/O and status tables are updated.	1 x 16-bits
	201	LastScanTime	This value represents the time, in milliseconds, requested to update the I/O table. This value should equal the RPI set for the Module.	1 x 16-bits

Area Description	Offset	Area Layout	Description	Size
	202	LastTransferTime	This value represents the time, in milliseconds, to update the I/O tables. If this value approaches the value of the Lastscan time, you may want to increase the RPI time. If the transfer time exceeds the RPI, the connection to the Module will be lost.	1 x 16-bits
	203	DataExchangeGood	If this value is not zero, an error occurred during the I/O and Status Table updates and the connection to the CLX CPU was closed. Once the error condition is cleared, the connection can be re-established. For the values, refer to Table 49, Good Data Exchange Values.	1 x 16-bits
	204	DataExchangeStatus	Last error returned by a failure while updating the I/O and status table. 0 = Good , Non Zero = Data Not available (reinitialize module)	1 x 16-bits
	205	DiagnosticMsgSent	Counts the number of diagnostic messages sent to the Module. For example, this counter will increment during a firmware update.	1 x 16-bits
	206	LadderMsgSent	Counts the number of CIP messages sent to class 0x0101. E.g., GetWord or Setword commands to the database.	1 x 16-bits
	207	RLLMsgSent	Counts the number messages sent to the Module from the Configuration tool.	1 x 16-bits
	208h	CLX_StackMajorVer	Major version of the Backplane firmware.	1 X 8 bits
	208l	CLX_StackMinorVer	Minor version of the Backplane firmware	1 x 8 bits
	219h	ProductMajorVer	Major version of the product when it was shipped.	1 x 8 bits
	219l	ProductMinorVer	Minor version of the product when it was shipped.	1 x 8 bits
	210	Protocol_Status	Overall status of the communication processor. For more information, refer to Table 51, Protocol State Definitions.	1 x 16-bits
	211	Channel0_Status	Overall Status of Channel 0. For more information, refer to Table 53, Channelx_State Values.	1 x 16-bits
	212	Channel1_Status	Overall Status of Channel 1. For more information, refer to Table 53, Channelx_State Values.	1 x 16-bits
	213	Channel2_Status	Overall Status of Channel 2. For more information, refer to Table 53, Channelx_State Values.	1 x 16-bits
	214	Channel3_Status	Overall Status of Channel 3. For more information, refer to Table 53, Channelx_State Values.	1 x 16-bits

Area Description	Offset	Area Layout	Description	Size
	215	Reserved	Reserved for future use.	2 x 16-bits
	217	IPAddress	Character array of the Module's IP address (e.g., 010.010.200.100).	16 x 8 bits
	225	Reserved	Reserved for future use.	16 x 8 bits
	333	Reserved	Reserved for future use.	9 x 16-bits
	242L	Reserved	Reserved for future use.	1 x 8-bits
	242H	ReconfigStatus	Status of Reconfig message that is sent when applying configuration change while a connection is open to module in RSLogix5000. Bit 0 – 3 Reply message Counter. Verify counter has changed to confirm reply has been returned. Then check Bit 4 to see if applying configuration data update was successful. Bit 4 indicates whether configuration data was accepted (0) or rejected due to error (1).	1 x 8 Bits
	242L	Reserved	Reserved for future use.	7 x 16-bits
	243	CyclicStatusOffset	Cyclic function status offset	1x16-bits
Config Table	0	Configuration Name	This field can be set to the configuration name saved in the Console config tool. The Module will not allow a connection if the configuration name does not match. All characters that are not part of the config name shall be null (0). If this field is blank, all characters are null. The Module will ignore the config name created from the Console and allow any configuration to connect to it.	30 x 8 bits
	30	Reserved		10 x 8 bits
	40	Clx_Zero_Output_Prog	This bit can be set to decide what happens to the output table when in Program/Test mode. For values and a description, refer to Table 50, Program Mode Output Table State.	1 x 8 bits

Table 50: Good Data Exchange Values (Local:Slot:S.Data[203], Local:Slot:S.DataExchangeStatus)

Bit	Description
0	If this bit is set, an error occurred during an input table update. For a description of the error, refer to DataExchangeStatus.
1	If this bit is set, an error occurred during an output table update. For a description of the error, refer to DataExchangeStatus.
2	If this bit is set, an error occurred during a status table update. For a description of the error, refer to DataExchangeStatus.
3 – 15	Reserved

Table 51: Program Mode Output Table State (Local:Slot:C.Data[40], Local:Slot:C.CLX_Zero_Output_PROG)

Value	Description
0	When in Program mode, leaves outputs in their last state. During this state, cyclic functions continue to operate.
1	Zeroes outputs when in Program mode. During this state, cyclic functions continue to operate.
2	When in Program mode, leaves outputs in their last state. Cyclic functions related to the output table are stopped.
3-255	Reserved for future use.

Table 52: Protocol State Definitions (Local:Slot:S.Data[210],Local:Slot:S.Protocol_Status)

Value	State	Description / Status
0	Unknown protocol state	Module is not configured or is in a unknown state.
1	No configuration	Module does not have a configuration.
2	Configuration OK, not connected	Module has a valid configuration, but a connection has not been established with the CPU.
3	Configuration OK	Module is running with a known good configuration.
-1	Configuration error	Module has loaded a bad configuration. Rerun pcinit.
-2	Watchdog error	Watchdog failure detected. The CLX Module will go into fatal error state and stop all communication with the network.
-3	Heartbeat failure	Heartbeat failure within the protocol. The CPU and CLX Module will go into a fatal error state and stop all communication with network.

Table 53: ReconfigStatus (Local:Slot:S.Data[242].8-12, AOP Tag Local:Slot:S.ReconfigStatus.0-4)

Generic Bit	AOP Bit	Meaning/Description
8 -11	0 – 3	Reply message counter. Verify counter has changed to confirm reply has been returned. Then check bit 4 (Tag Error) to see if applying configuration data update was successful. This counter rolls over to 0 after It reaches 15
12	4	Bit is 1 means configuration rejected (Bad configuration data) Bit is 0 means configuration data was accepted

8.3 Channelx_State

Channelx_State indicates the state of each communication channel on the Module. This information is used to update the CLX Status table.

Table 54: Channelx_State Values (Local:Slot:S.Channel[0 – 3]_Status)

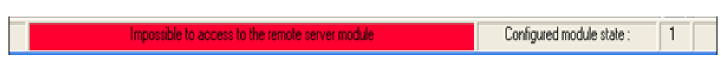
Value	State	Description
0x82	Communication good	Indicates that the last status value for all cyclic functions configured for this channel was good.
0x81	Communication bad	Indicates that an error occurred on one or more cyclic functions configured for this channel since the last update.
0	Not active	No cyclic functions detected on this channel.

Caution

Ladder code for the Module Status Registers should be monitored for error states. If an error state occurs, the processor has lost the connection to the Module or to a channel and should assume a safe state.

8.4 Console Status Bar Information

Table 55: Console and Module communication status information

Definition	Configuration	Display
Connection OK	Active configuration matches with the module's configuration	
Connection OK	Active configuration **does not** match with the module's configuration	
Connection NOT OK	Unverified configuration since no connection available to the module	

8.5 User Status Table

In default Address Mode:

This table (Local:Slots:S.Data[0] to [199] and AOP tag ConfigurableStatus[200]) consists of 200 status words, read from offset 500-699 of the database. The individual words are configured in the cyclic function, created in the Console.

In Extended Address Mode:

The AOI tag CLX2000.DATABASE_DATA.STATUS consists of all defined status words (maximum 1024 status words). The individual words are configured in the cyclic function, created in the Console.

For more information, refer to Section 7.3.2, Configuring Cyclic Functions. For possible status values, refer to Section 7.3.4, Cyclic Function Status. Always check these values to determine if the I/O data is valid.

8.6 Diagnostic Counters, Status and Information

The Module maintains a variety of diagnostic counters to indicate:

- General statistics on messages sent and received
- The product and Backplane firmware version
- The communication processor's status
- The channel's status

All these information are located in the last 50 words of Local:Slot:S.Data[] .For AOP, these start at tag Transcount and end at tag ReconfigStatus.

Note

The Status Input size in ESR2 module properties can be set from 50 to 250.

If the Status size is 250, the last 50 word would be in Local:Slot:S.Data[200] to Local:Slot:S.[249]

If the Status size is 50, it means there is no defined user status. The 50 words would be in Local:Slot:S.Data[0] to Local:Slot:S.[49]

If the Status size is 150, the last 50 word would be in Local:Slot:S.Data[100] to Local:Slot:S.[149]

9

Upgrading the Module Firmware

Chapter Sections:

- Using Console Application
- Using Firmware Updater
- Using USB Port
- Checking the Protocol Firmware Version with APSYM

There are three ways to upgrade the module's firmware; which are by Console, Firmware Updater and USB port.

Note
It is important to check the matrix below to see what is supported by the firmware version running in your module before starting to update your module.

Update Tool	Firmware Version		Applicable Section
	2.3.0 and below	2.7.0 and higher	
Firmware Updater	√	√	Using Firmware Updater
Console		√	Using Console Application
USB port		√	Using USB port

Note

Rockwell RSLinx is no longer required to make a connection to SST-SR4-CLX with SST Backplane Communication Products install v1.8 or later. This install includes its own Ethernet/IP driver which can be used for connection to an Ethernet/IP ControlLogix module to access the backplane where the module resides.

9.1 Using Console Application

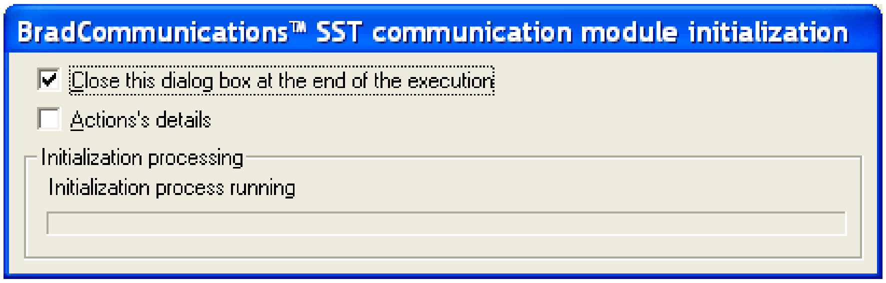
If the module currently has the firmware version 2.7.0 or higher, use the Console application to update the module's firmware by following the steps below:

1. Copy and unzip Clx2000_firmware.zip file to the current configuration folder. Make sure there are no similar files in the folder with extension ".UPD".

2. See Module's Installation Directory Location to locate the firmware as well as the configuration folder.

3. Locate and run PcInit  download the firmware Module. During download, the initialization software will copy the files to flash and change the name of the file in the configuration directory to *.UPD, indicating that the transfer was successful.

4. During download, the window below will show up and close automatically when it is complete.



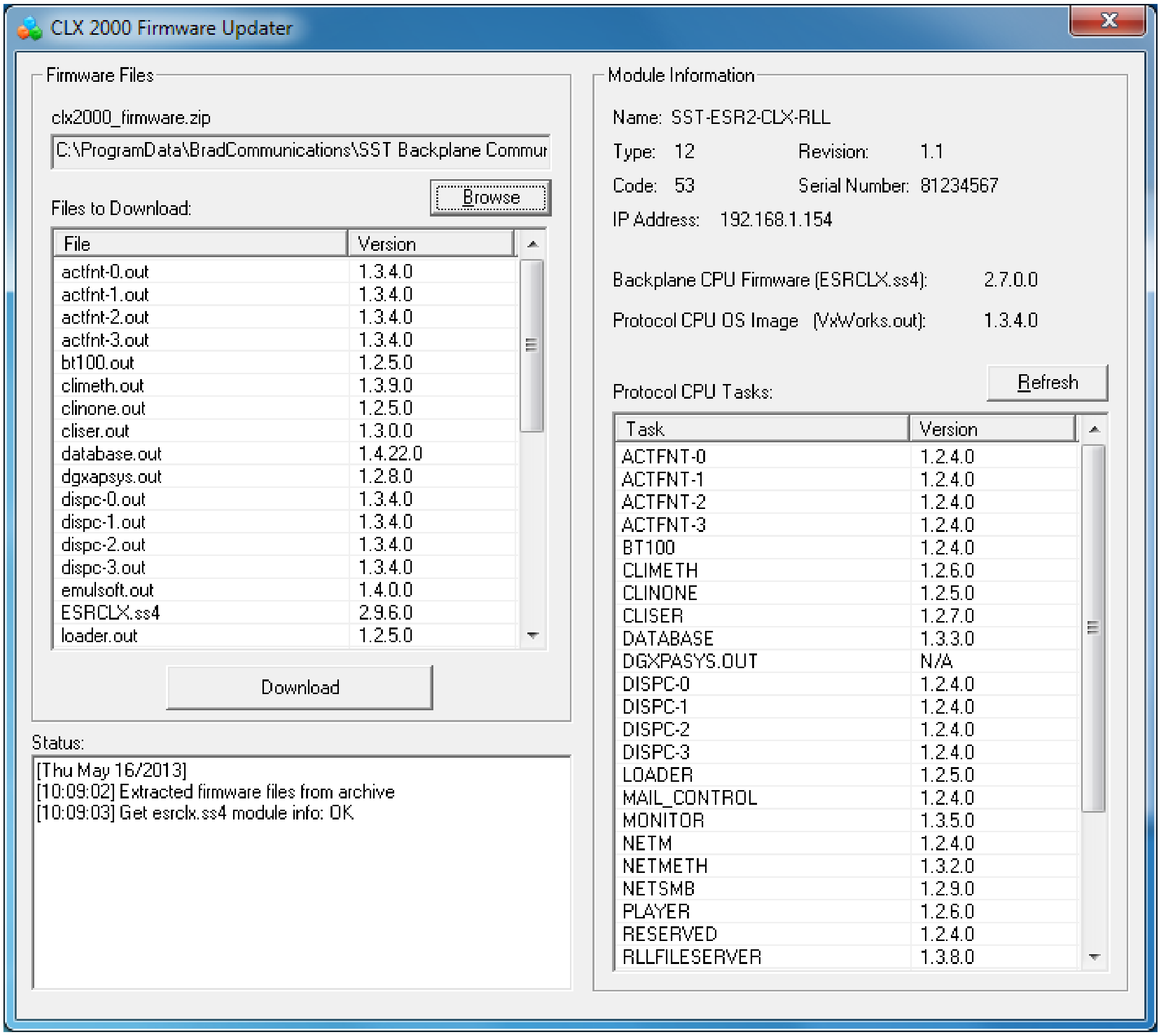
5. The download will take approximately 5mins.

9.2 Using Firmware Updater

To upgrade the Protocol CPU's firmware using the Flash Updater.

1. Run the SST Backplane Communication Products install v1.9 or later.
2. When Install is complete, run the console using the shortcut *All programs\BradCommunications\SST Backplane Communication module\console*.
3. Select or create an empty project (no cyclic function configuration) and go to properties.
4. Set the connection type to RSLinx or Ethernet/IP.
5. Open the configuration and make sure the connection to module is successful. This is indicated by seeing a GREEN status at the bottom of console.
6. Close console.
7. If there is an active backplane connection, make sure to inhibit the module first before starting the upgrade.
8. Reopen console.
9. When Configuration Manager Dialog box is displayed, select the firmware button located at the bottom of dialog under Download to backplane. This will launch the Flash Updater. The Flash Updater will appear similar to below.

Figure 117: Flash Updater Display



10. The left side pane will display the version of the files to be downloaded into the module, while the right side pane will display the version of the files currently running in the module.

Note

If the module firmware version is 2.5 or below, there won't be any display from the right side pane of the flash updater.

11. Select the Browse button and browse for clx2000_firmware.zip. This will be located in LocalAppDataFolder\BradCommunications\SST Backplane Communication Module\Firmware Update
12. After the zip file has been opened a list of task files will be displayed in left hand pane with versions beside it.
13. Select Download button.
14. The download operation will begin and as each task is updated on module a message will be displayed under status (left-hand window).
15. Wait until a prompt message box is displayed asking to Reset the module for the new firmware to take effect.
16. Select Yes.
17. The module should reboot and initialize successfully with the OK LED going solid GREEN.
18. Verify you see the following messages in flash Updater displayed under status.
    - Module Reset: Success
    - Module Reboot: Success
19. Close the Flash updater.

If a failure is encountered during the firmware update, please follow the recovery procedure below:

1. Do not power cycle the module.

2. Locate the clx2000_firmware.zip as indicated in Module's Installation Directory Location.

3. Copy and unzip the clx2000_firmware.zip to the selected configuration folder in Step # 3 above located in Module's Installation Directory Location.

4. Open Console application

5. Locate and click , download will run for approximate 5 minutes.

9.3 Using USB port

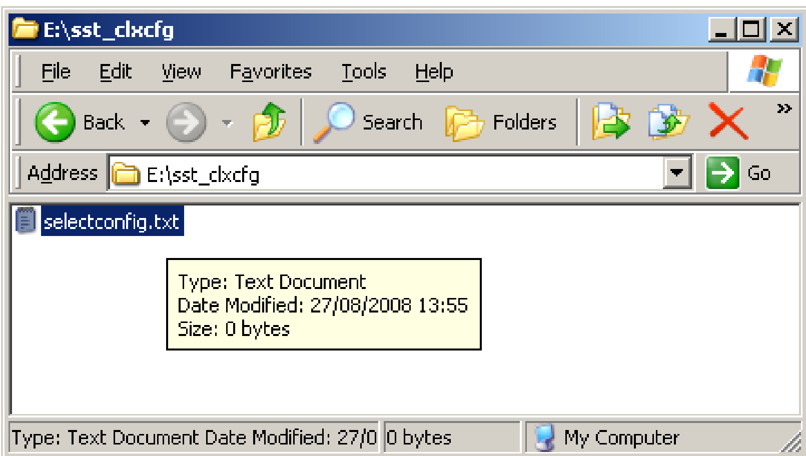
To upgrade the Protocol CPU's firmware using the USB port.

1. On the USB stick, do the following:

    - Check if the "sst_clxcfg" directory exists. If not, create it and ensure that it doesn't contain a folder named "**oldconfig**".

    - Inside the directory, check if selectconfig.txt is available. If not, create it using Notepad.

Figure 118: SelectConfig.txt, Inside Sst_clxcfg Folder



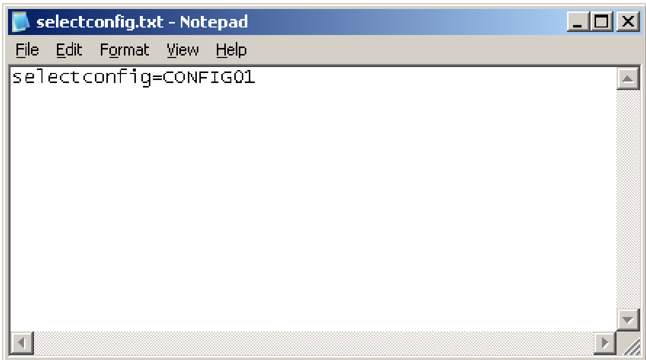
2. Select the configuration folder to use for download from Module's Installation Directory Location. Once selected copy and unzip Clx2000_firmware.zip to the selected configuration folder.
3. In selectconfig.txt, insert "selectconfig=configuration name", where "configuration name" stands for the configuration folder name that is selected.

Note

The selected configuration will overwrite the current configuration running in the module. If the configuration does not need to be changed, choose the configuration folder that matches to the configuration running in the module.

4. There should be NO escape character after the folder name (e.g., "\n" for new line, or "\r" for carriage return).
5. Close the notepad.

Figure 119: SelectConfig.txt, with CONFIG01 as Selected Configuration



6. Copy the configuration into the sst_clxcfg.

Figure 120: Configuration Directory

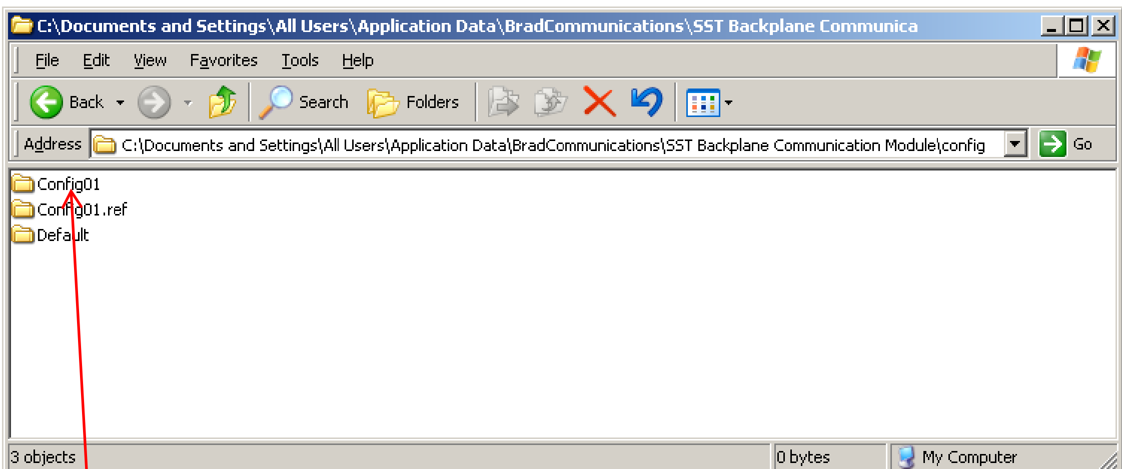
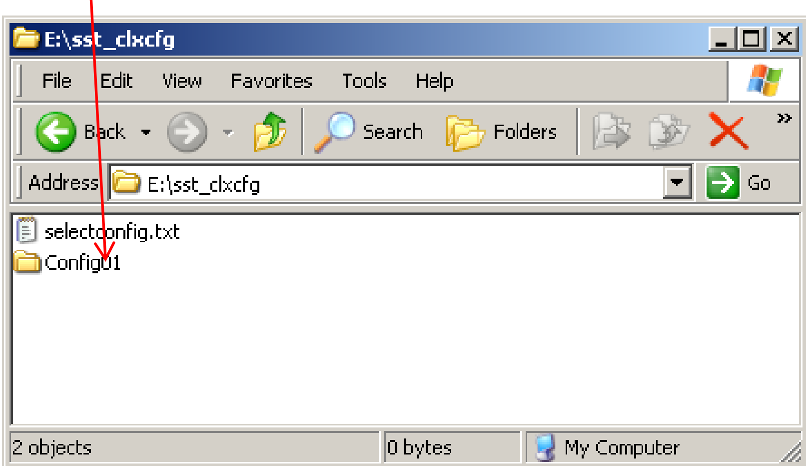


Figure 121: Selected Configuration, Copied into SST_clxcfg



7. Ensure that the configuration to be downloaded matches the module variant.

8. Close backplane connection to module if connection exists. This can be done by inhibiting the connection in RSlogix 5000.

9. Plug the USB stick into the configuration port. A message “Reset the card to start update ” will appear on LCD display. Reset the card. The message “reading files…” will appear and then transition to “backup in progress…”. The existing configuration will be copied into a directory named “**oldconfig**” inside sst_clxcfgbefore the upgrade starts.

10. Ensure that “backup in progress…” and “update in progress…” are displayed during the backup and update, respectively. If the module is connected to a controller while the update is executing, the messages will only be displayed as the process begins. The display will change to CONNBLK until the update completes. When the update is complete the message “Update complete, remove USB stick then reset the card” is displayed. Remove the USB stick and reset the card.

11. All files with extension .out and .ss4 will be replaced with .UPD.

9.4 Checking the Protocol Firmware Version with APSYM

If the module has protocol firmware version1.2.3.0 or later, you can view the versions of all protocol firmware running on card:

1. Ensure that the connection (Ethernet/IP or RSLINX or TPC/IP) to the card is already defined through the Console configuration that it is being used with the card.

2. Go to Start Menu > BradCommunication > SST Backplane Communications Module > Protocol Information Viewer. Two dialog boxes are displayed: the VxWorks Image Info dialog, indicating the version of VxWorks that’s running on the card, and the ApSymDeb dialog, indicating the protocol firmware and its version.

Figure 122: VxWorks Image Info Dialog

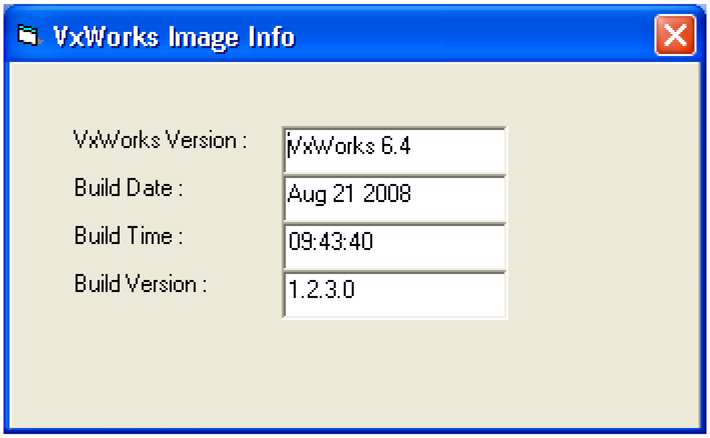
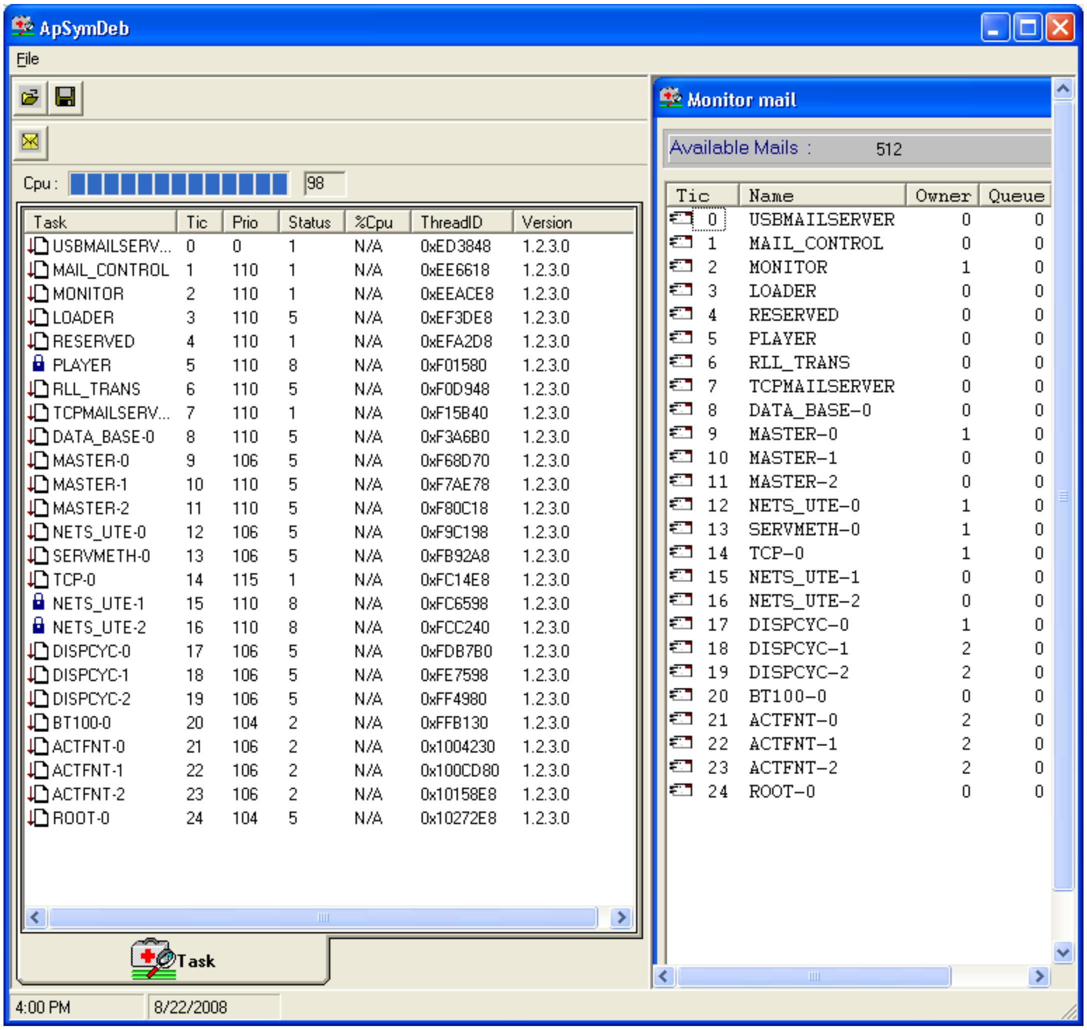


Figure 123: ApSymDeb Dialog



3. After finish viewing the protocol firmware, select File Menu > Quit

10

Using the Configuration Tool via RSLinx or Ethernet/IP

Chapter Sections:

- Configuration Tool Overview
- Functions
- Configuring via RSLinx Connection
- Configuring via Ethernet/IP

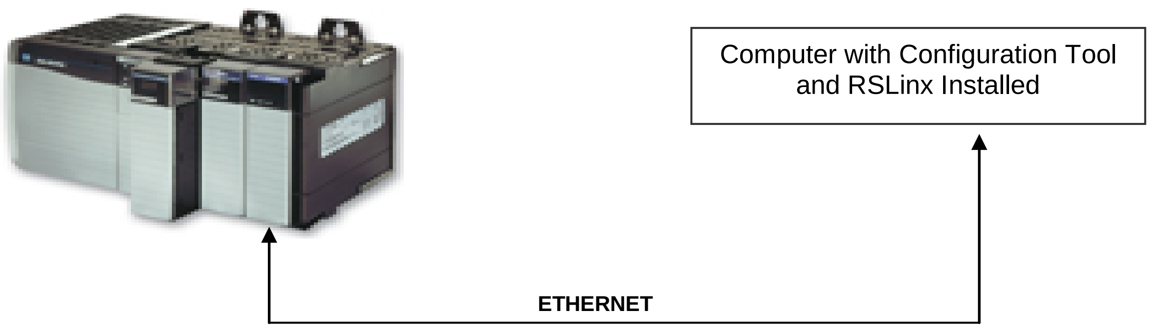
10.1 Configuration Tool Overview

Using Ethernet, the SST Console Configuration Tool can exchange data with a communication Module. This communication uses RSLinx driver version 2.2 OEM/ Professional or higher.

Note

For details on configuring the driver, refer to Section B.1, Installing and Configuring the RSLinx Driver.

Figure 124: ESR2/SR4, Ethernet and Configuration Tool Relationship



10.2 Functions

The following Configuration Tool functions are available through Ethernet:

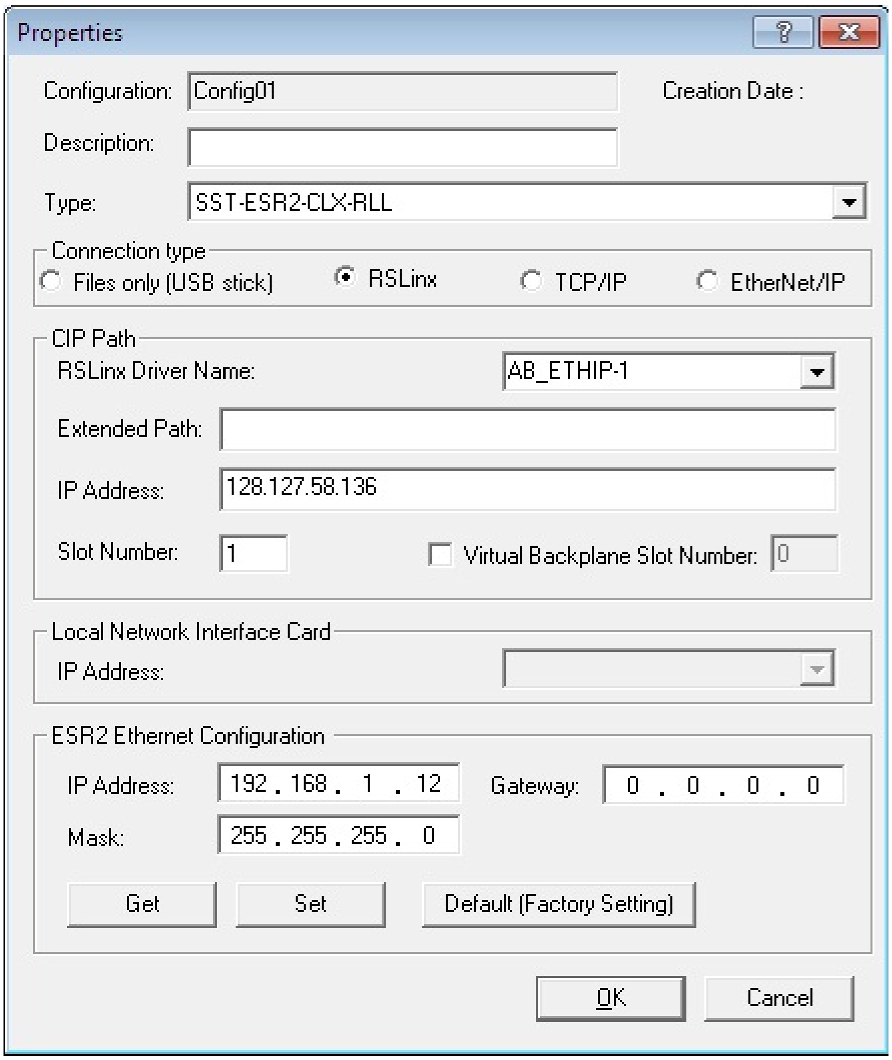
- Configuration loading
- Diagnostics

10.3 Configuring the RSLinx Connection

To use the Configuration Tool with an RSLinx connection, follow these steps:

1. Launch the Console.
2. Select Properties. The Properties screen is displayed.

Figure 125: Properties Screen



3. Select “RSLinx”.
4. Select the RSLinx driver name.

5. If the Module is in a remote rack, enter the Extended path. Here is an example:

   The remote Ethernet Module is in slot 6 and its address in the remote rack is 130.151.132.1. The SST-PFB-CLX Module is in slot 8. So the string would be:

   01 06 12 0D 31 33 30 2e 31 35 31 2e 31 33 32 2e 31 00

   01h = Communication with the Backplane

   06h = Slot of Remote Ethernet Module (in local rack)

   12h = Port Address

   0Dh= Length of Byte (Number of ASCII bytes in IP Address)

   31 33 30 2e 31 35 31 2e 31 33 32 2e 31 00 = IP Address in ASCII. If the number of bytes in the IP address in ASCII is odd, pad another byte (00h) on the end of the path, like in this example.

6. Enter the IP address of the 1756-ENBT when using the Ethernet/IP driver for local racks.

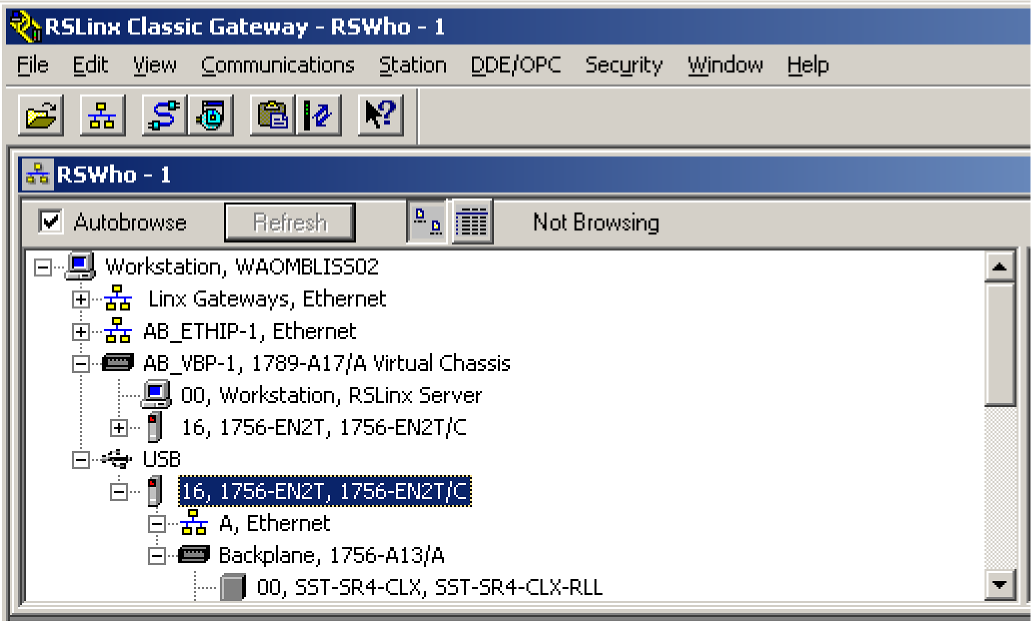
   Note

   Remote Devices via Linx Gateway driver is not supported with the RSLinx Classic Profession/Gateway v2.54 or higher.

   The IP Address field can be left blank when using the 1756-ENET Ethernet TCP/IP Communications ControlLogix module or when RSLinx path contains an IP address.

7. Enter the slot number where the SST-ESR2-CLX or SST-SR4-CLX module is located in rack.

8. Enter the virtual backplane slot number if using a Virtual Backplane driver (i.e. AB_VBP-1). This can be found in RSLinx in RSWho under USB. The slot number will appear beside the Communication module. For example below, the slot number would be 16.



9. Click OK to save the configuration.

10. Connect to the Module by clicking the Open button under local configuration and selecting Yes to the message prompt that follows.

11. If connection was successful, a GREEN bar will be displayed at bottom of console.

10.4 Configuring the Ethernet/IP Connection

To use the Configuration Tool with an Ethernet/IP connection, follow these steps:

1. Launch the Console.
2. Select Properties. The Properties screen is displayed.

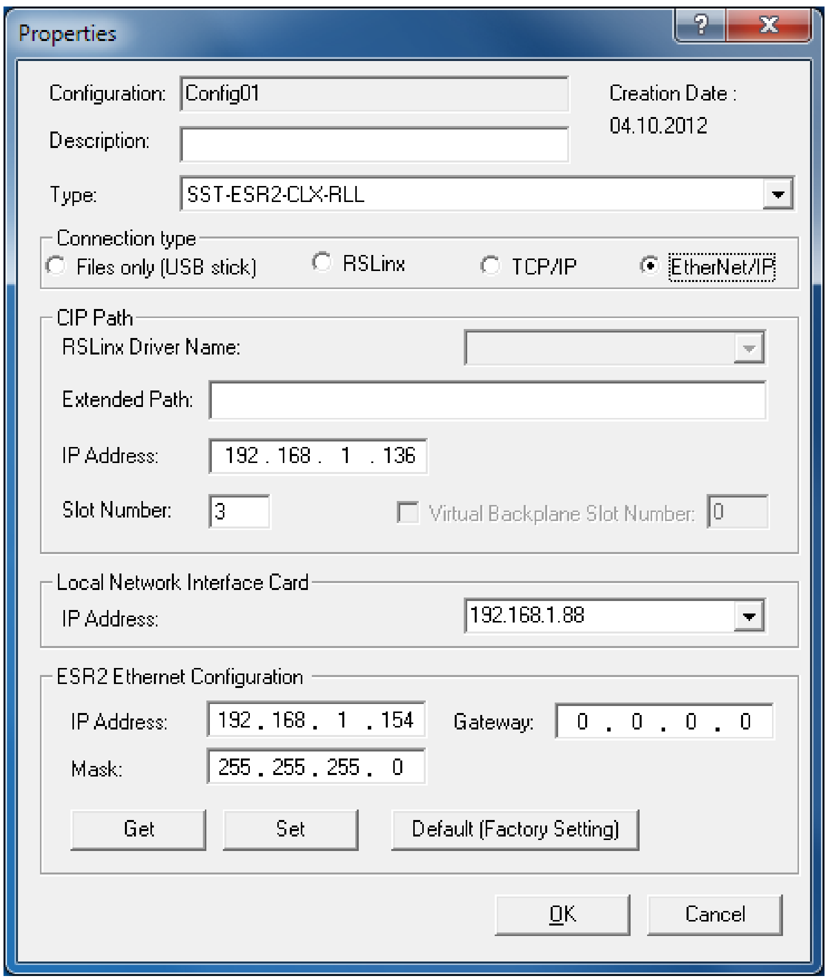


Figure 126: Properties

3. Select Ethernet/IP.

4. If the Module is in a remote rack, enter the Extended path. Here is an example:

The remote Ethernet Module is in slot 6 and its address in the remote rack is 130.151.132.1. The SST-PFB-CLX Module is in slot 8. So the string would be:

01 06 12 0D 31 33 30 2e 31 35 31 2e 31 33 32 2e 31 00

- 01h = Communication with the Backplane
- 06h = Slot of Remote Ethernet Module (in local rack)
- 12h = Port Address
- 0Dh= Length of Byte (Number of ASCII bytes in IP Address)
- 31 33 30 2e 31 35 31 2e 31 33 32 2e 31 00 = IP Address in ASCII. If the number of bytes in the IP address in ASCII is odd, pad another byte (00h) on the end of the path, like in this example.

5. Enter the IP address of the 1756-ENBT when using the Ethernet/IP driver for local racks.
6. Enter the slot number where the SST-ESR2-CLX or SST-SR4-CLX module is located in rack.
7. Select the local network interface card IP address.
8. Click OK to save the configuration
9. If connection was successful, a GREEN bar will be displayed at bottom of console.

11

Console Tool Tips

Chapter sections:

- Cyclic Function Configuration Graphical Interface
- Navigating to cyclic functions
- Duplicating devices that have the same profiles
- Overlapping of cyclic function data

11.1 Cyclic Function Configuration Graphical Interface

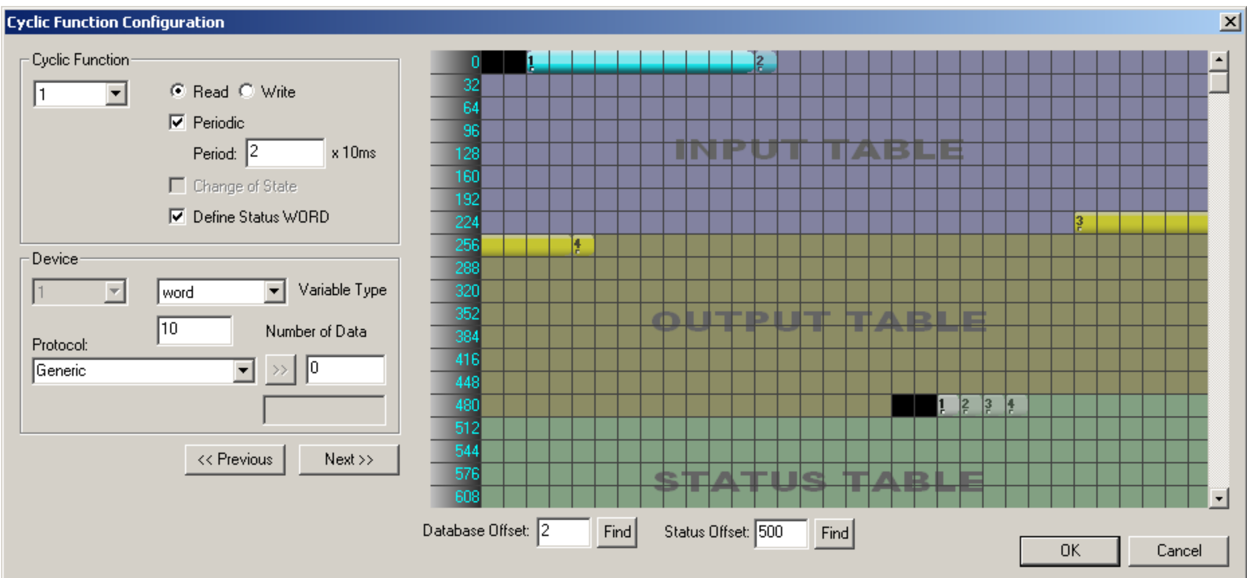
The cyclic function configuration interface displays the database area on the module into the following colour schemes:

Read cyclic functions – BLUE

Write cyclic functions – YELLOW

Status words for cyclic functions – GRAY

Navigation to cyclic functions is simple using the user-friendly graphical interface. Cyclic functions can be easily dragged and dropped to new locations as required.



Note

The first two BLACK locations of the Input table and Status table are reserved and cannot be used. The black mapping area locations after Status table are available but will only be accessible via CIP message. It's recommended to map all cyclic function data in the colored regions.

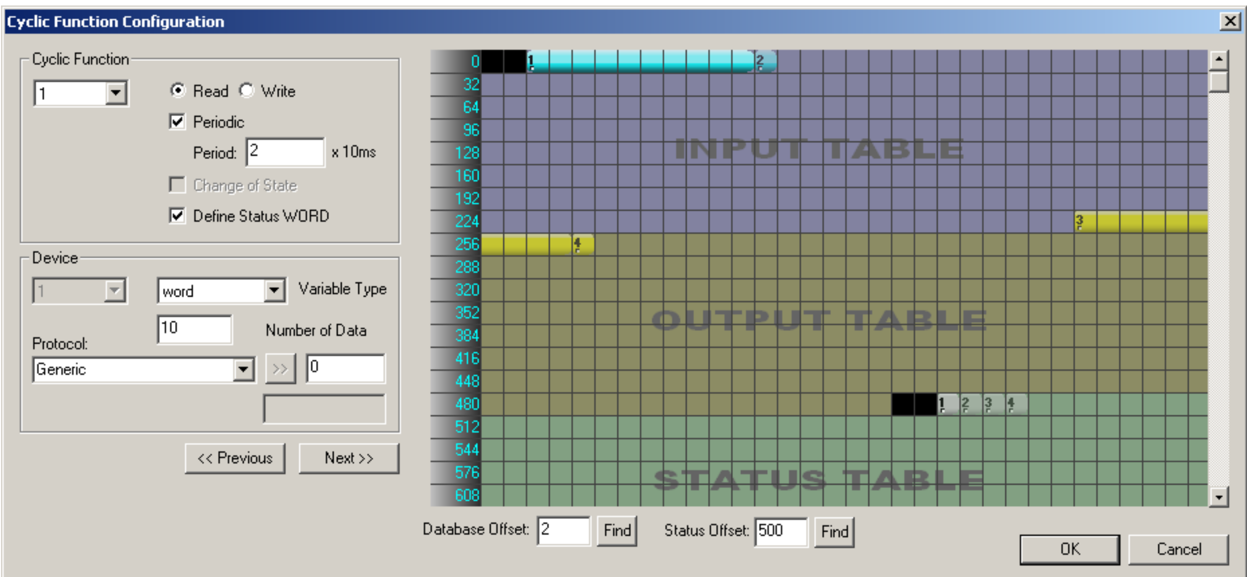
Make sure to set the Input, Output and Status to their required sizes before creating cyclic functions to avoid running out space and having to rearrange cyclic functions.

11.2 Navigating to cyclic functions

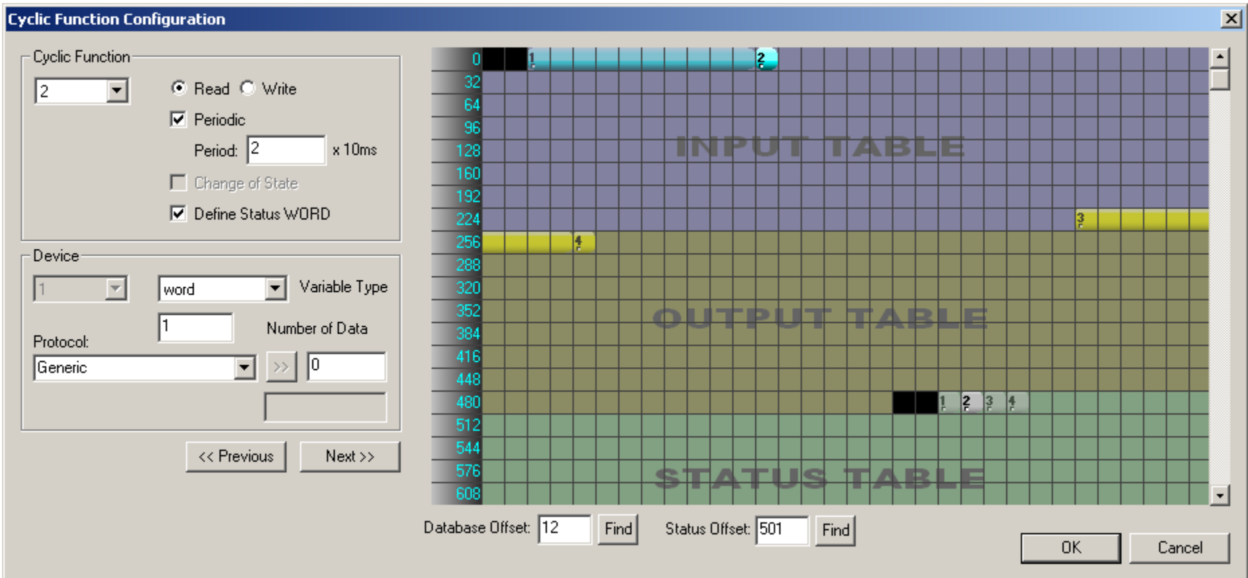
Once the graphical interface for the cyclic function configuration is opened, all existing cyclic functions can be modified by using the method below.

To switch to another cyclic function in the Cyclic Function Configuration window, hold the CTRL Key down while clicking on the cyclic function number using left mouse button. Once the cyclic function starts flashing then it can be dragged and dropped into another location in the database.

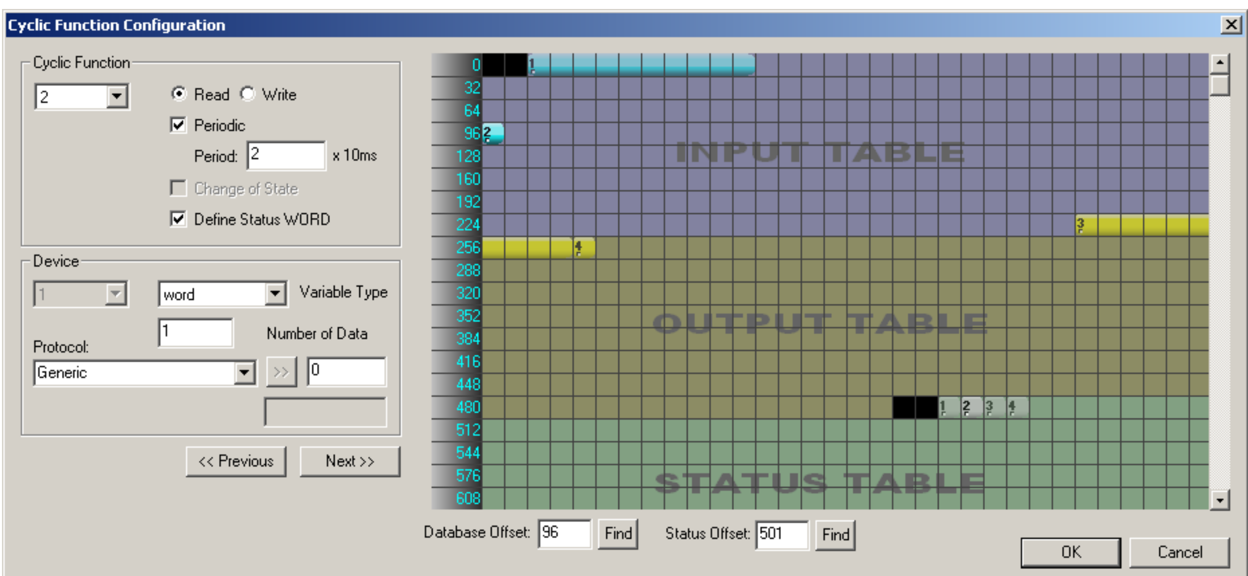
In the example below, the cyclic function configuration dialog has been opened for cyclic function 1 for channel 0. The number of circles below the cyclic function number indicates the channel. One circle indicates the first channel. Two circles indicate the 2nd channel and so on.



Now to activate cyclic function 2, hold down the CTRL key and click on the cyclic function number to activate. Cyclic function 2 will begin flashing.



Drag cyclic function 2 to new database location 96. Continue modifying existing cyclic functions as needed and click OK button to close the Cyclic Function Configuration Dialog.



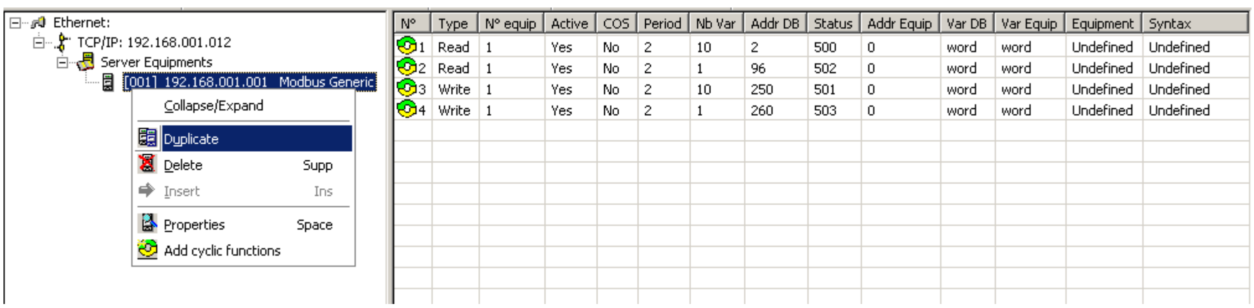
Note

It’s recommended that read cyclic functions stay in the Input Table area and write cyclic functions stay in the Output Table area. Status words for cyclic functions should also remain in the Status Table area since the paging AOI copies all status into the status tag array.

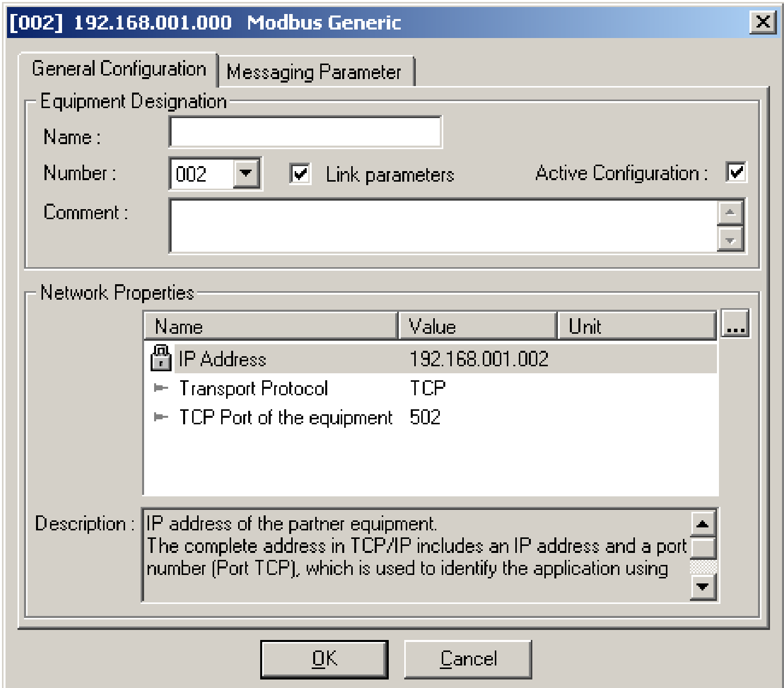
11.3 Duplicating devices that have same profiles

A device along with its cyclic functions can be duplicated in the console. There are two ways to duplicate the device. First is by clicking on the  icon in the toolbar. The second is by right-clicking on the device and selecting Duplicate from the mouse menu.

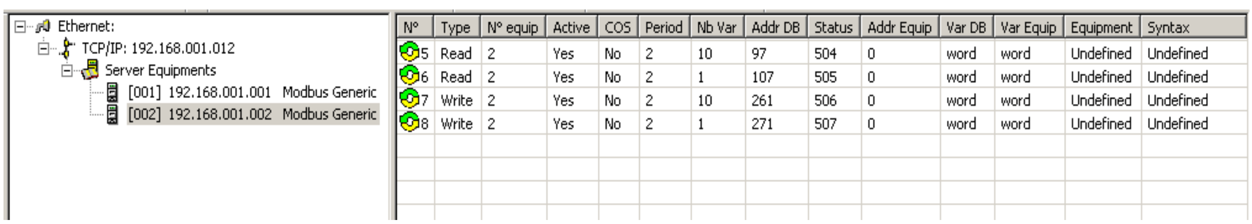
In the example below, device 1 is duplicated by selecting Duplicate from the Right-click mouse menu.



After selecting Duplicate from Mouse menu the following dialog will appear asking to set the device properties as below. Set the properties of device and select Ok button.



In the example below, device 2 has been duplicated with the same number of cyclic functions and similar cyclic function properties. The database addresses have been automatically adjusted. The generated database addresses will be consecutive addresses.

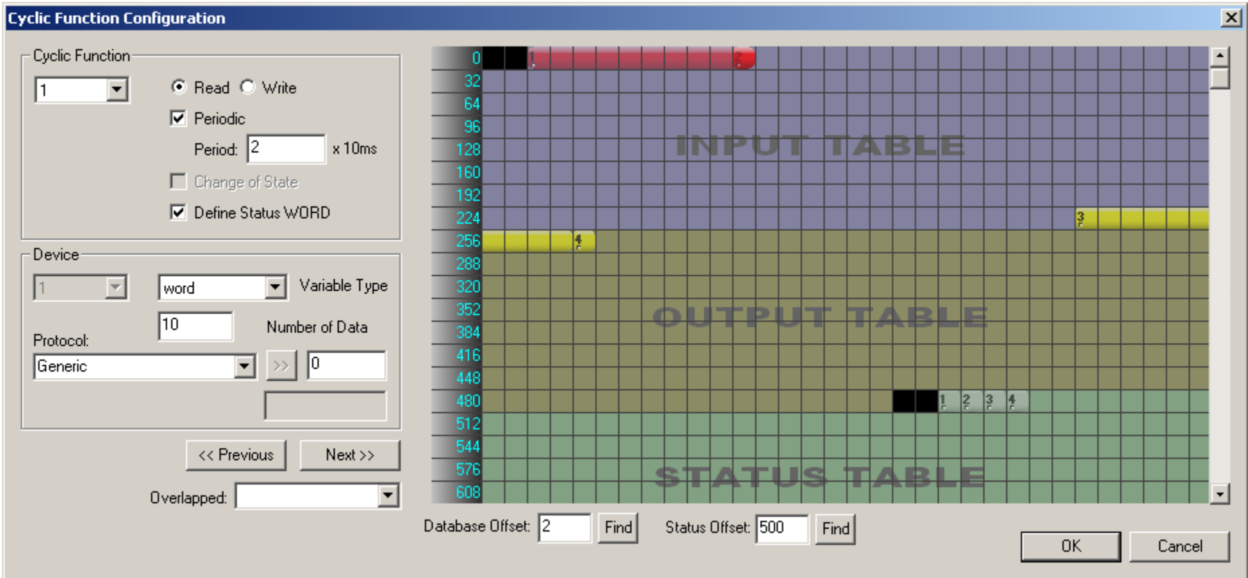


N°	Type	N° equip	Active	COS	Period	Nb Var	Addr DB	Status	Addr Equip	Var DB	Var Equip	Equipment	Syntax
5	Read	2	Yes	No	2	10	97	504	0	word	word	Undefined	Undefined
6	Read	2	Yes	No	2	1	107	505	0	word	word	Undefined	Undefined
7	Write	2	Yes	No	2	10	261	506	0	word	word	Undefined	Undefined
8	Write	2	Yes	No	2	1	271	507	0	word	word	Undefined	Undefined

11.4 Overlapping of cyclic function data

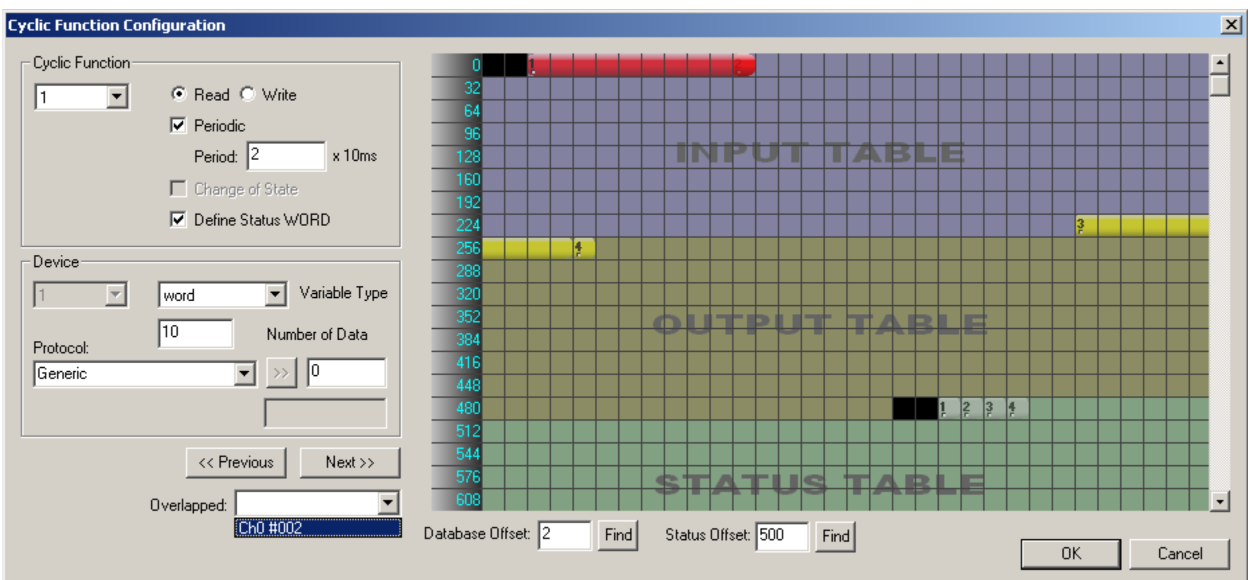
Overlapping of cyclic function data can result when duplicating devices in large configurations. Here is an example of two cyclic functions overlapping.

In the example below, cyclic functions 1 and 2 are overlapping. Overlapping cyclic functions appear in red. Once cyclic functions are no longer overlapping the cyclic function will appear using its normal color.

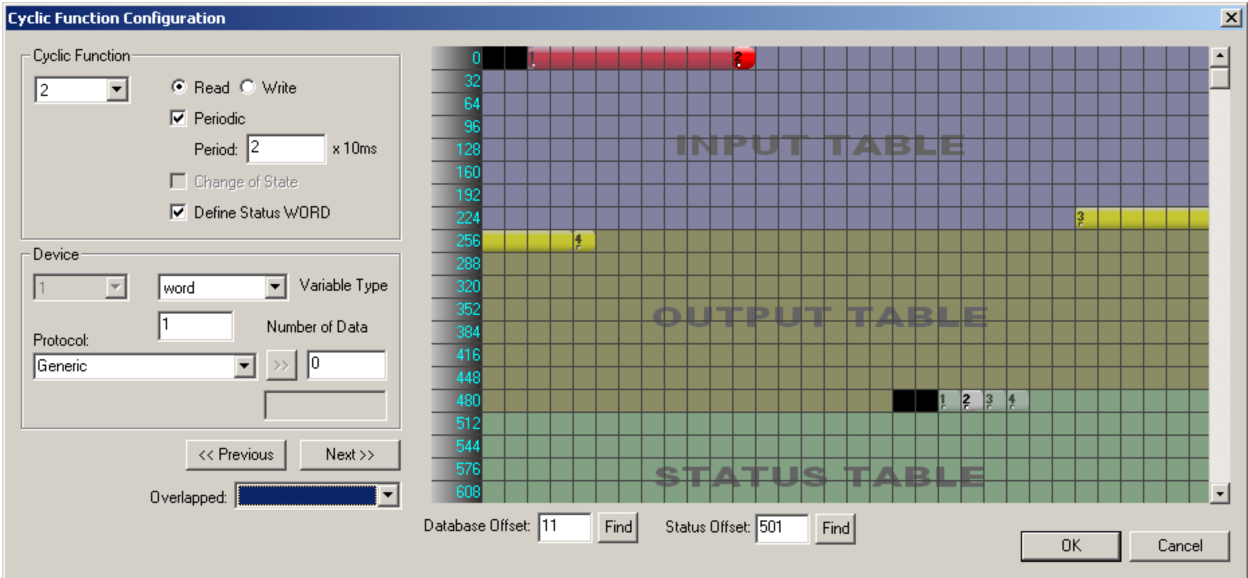


To correct the overlapping cyclic functions in the example above follow the steps below:

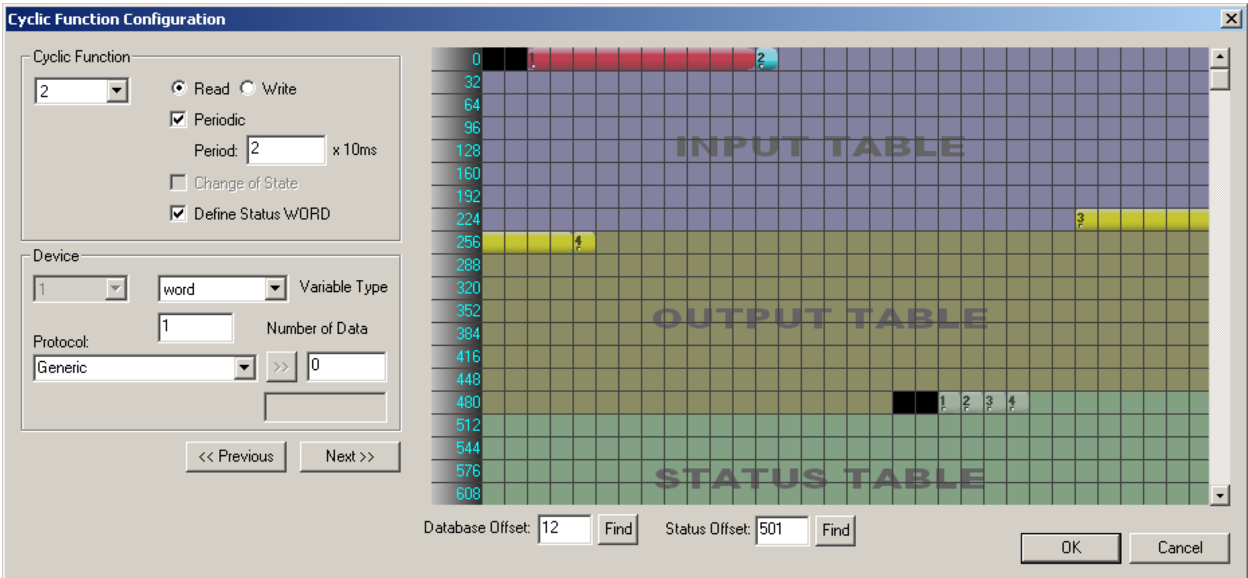
1. Expand the **Overlapped** drop-down box (bottom-left side) to select the cyclic function to move so no more overlap occurs. In the example below, CH 0 #002 is selected.



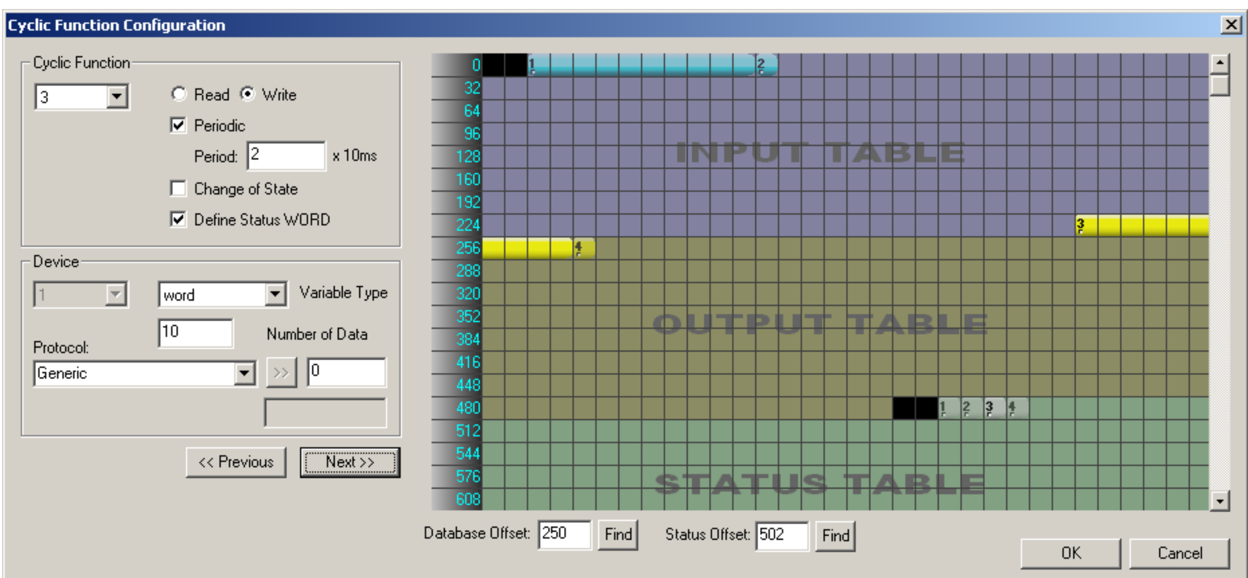
2. Now Cyclic function #2 is active and can be dragged to a new location so no more overlap occurs.


Cyclic Function Configuration
Cyclic Function
2
Read
Write
Periodic
Period: 2 x 10ms
Change of State
Define Status WORD
Device
1
word
Variable Type
1
Number of Data
Protocol:
Generic
>>
0
<< Previous
Next >>
Overlapped:
0
32
64
96
128
160
192
224
256
288
320
352
384
416
448
480
512
544
576
608
INPUT TABLE
OUTPUT TABLE
STATUS TABLE
Database Offset: 11
Find
Status Offset: 501
Find
OK
Cancel

3. Click on Cyclic function 2 and drag it to a new location so that it no longer overlaps cyclic function 1.



4. Click on **Next >>** button to go to the next cyclic function. Cyclic function 1 and 2 are no longer displayed in RED and appear normal to indicate there is no overlapping..



A

Application Examples

Appendix Sections:

- Addressing
- Start-Up Code Example
- Creating a Generic CIP Message for Accessing Extended Database Areas
- Data Type Structures
- CIP Messaging Sample Database Code

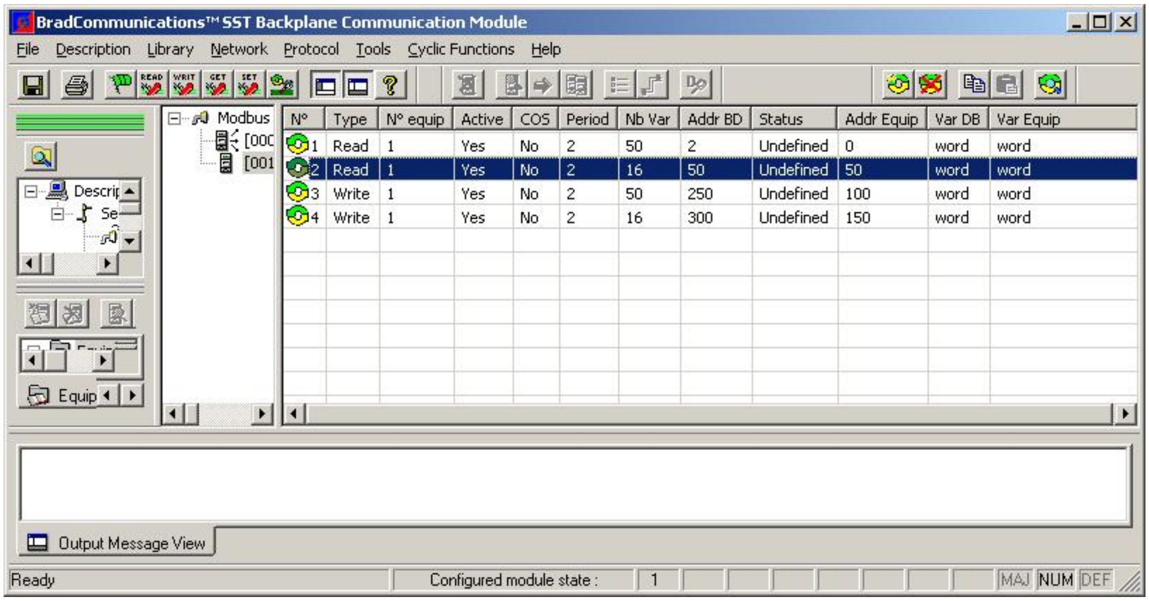
A.1 Addressing

A.1.1 I/O Data Packing Example

Cyclic function Number 1, below, is reading data from offset 0 of the equipment and writing it to the second offset of the input table. The second cyclic function is reading bits from offset 50 of the equipment and writing them to word offset 50 of the input table. Using this data packing method, it is able to combine different cyclic functions from the same or different equipment.
This enables the I/O tables to be optimized.

The same principle can be applied to the database, allowing for Generic CIP message optimization. For more details, refer to Section A.3, Creating a Generic CIP Message for Accessing Extended Database Areas.

Figure 126: I/O Data Packing Example



A.2 Start-Up Code Example

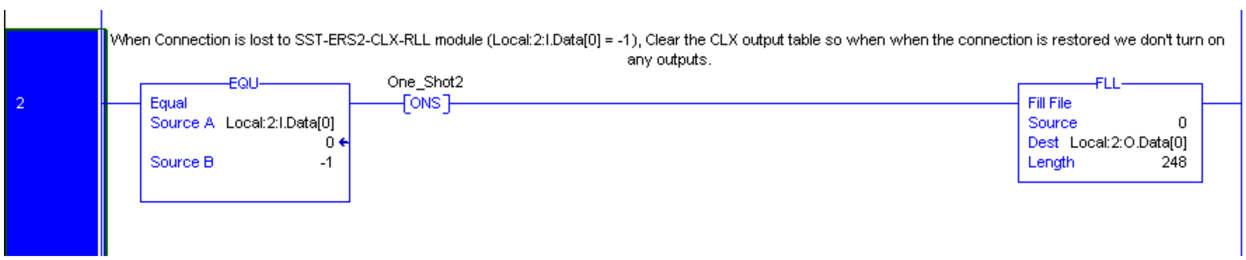
The following example shows how to interface with the Module.

- Read the Input table, Word offset 0 (Local:Slot: Input[0], Local:Slot:I.ModuleStatus), to see if the card's I/O tables are being updated. If the value is -1, the connection to CLX processor has been lost and the I/O tables are no longer being updated. If the value is 0, there is a connection to processor. Continue processing your logic.

Note

It is recommended that your Control Logic check Local:Slot: Input[0] or Local:Slot:I.ModuleStatus for a value of 0 before processing the I/O tables.

Figure 127:   Start-Up Code CLX Example



A.3 Creating a Generic CIP Message for Accessing Extended Database Areas

The communication Module allows the PLC program to read or write database areas not accessible by the input, output and status tables. This data can be read at any time by the PLC program. For a list of the available commands, refer to Table 55, Service Codes. Also, unconnected CIP messages are supported. An example of using unconnected CIP Messages would be when configure the module as a server and the Word database offsets 250 – 497sshould not be over written by CLX controller output table.

Note

Unconnected CIP Messages are supported in backplane firmware ESRCLX.SS4 v1.7 and higher.

Note

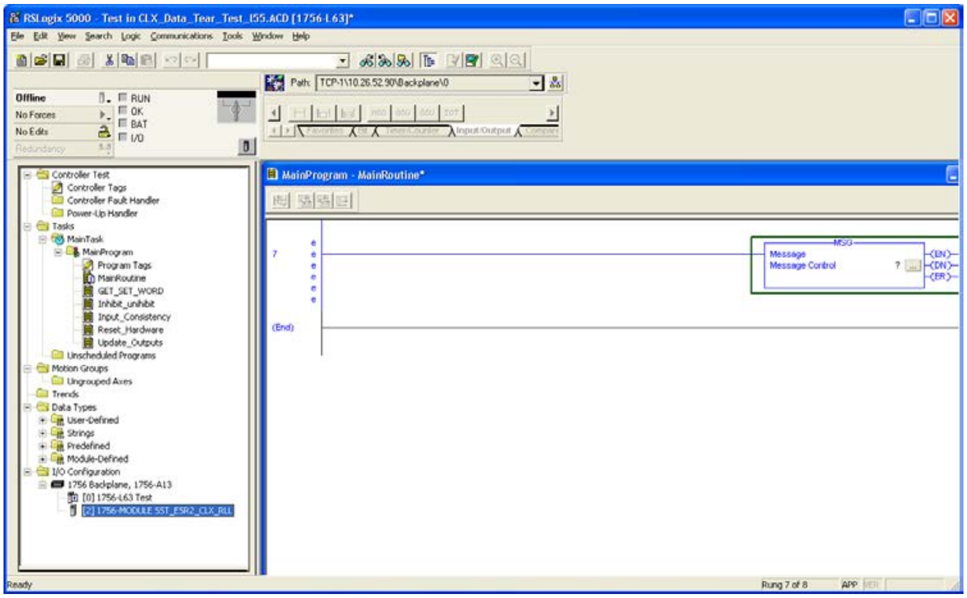
There is limit to the number of MSG (cache connection) instructions that can leave opened (cached) in RSLogix 5000. For more information, check the RSLogix online help.

If there are more than 16 MSG instructions are used in the ladder logic, it is recommend to disable the connected messaging option.

To create a generic CIP message, follow these steps:

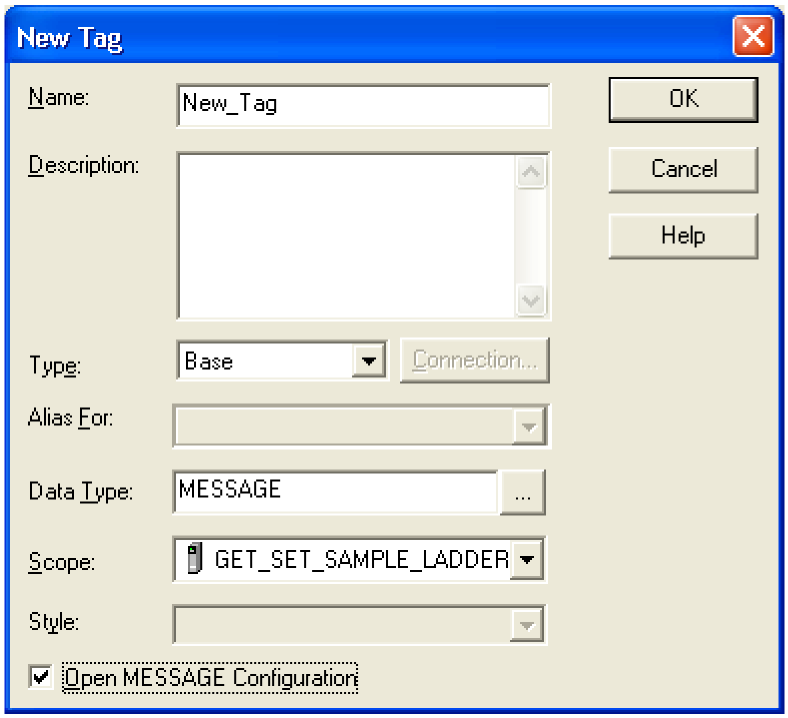
1. Run the RSLogix 5000 software.
2. From the Input/Output tab, select the MSG instruction and place it into a rung of ladder code.

Figure 128: MSG Instruction in Ladder Code



3. If there is no tag created for the message, right-click on “Message Control” inside the MSG instruction and select the New Tag. The New Tag dialog box displays.

Figure 129: New Tag Dialog



4. Enter a unique tag name.

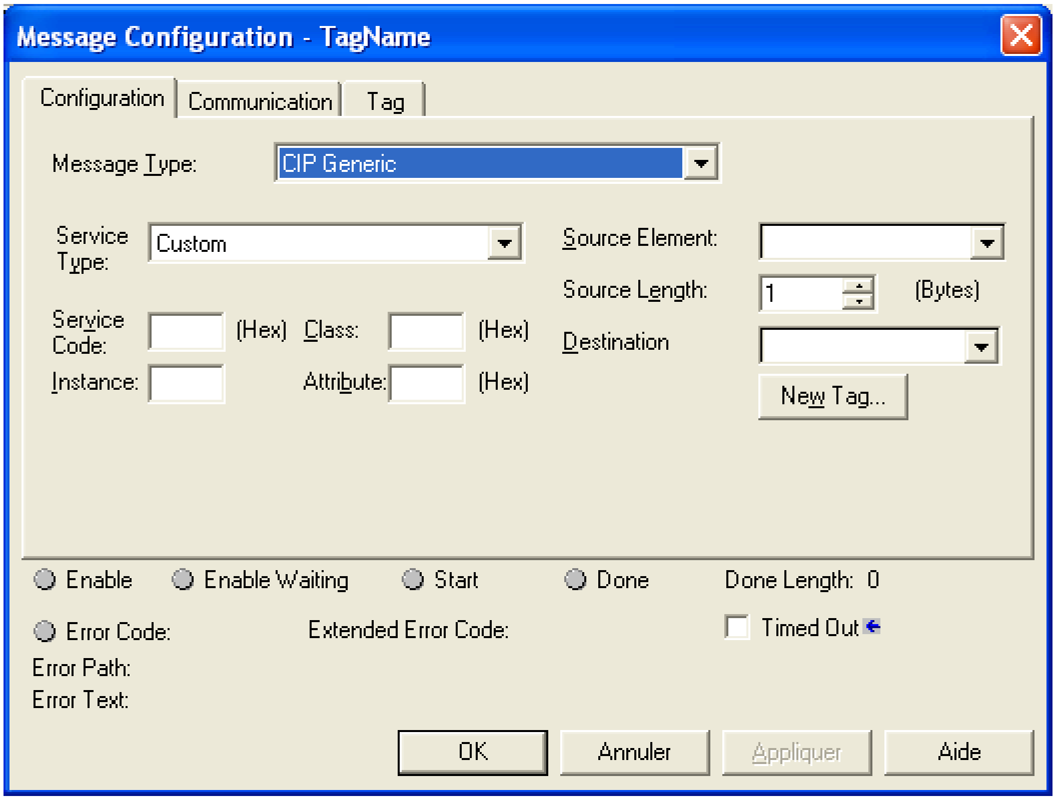
5. Check “Open MESSAGE Configuration” and click OK. The Message Configuration dialog displays.

Note

This step may differ between versions of RSLogix 5000. If the option does not exist, click OK and then right-click on ... *of the Message instruction. Select Edit Instruction.*

.

Figure 130: Message Configuration Dialog



6. Under the Configuration tab, enter the following information:

Message Type: CIP Generic

Service Code:	The desired command. For a list of commands, refer to Table 55, Service Codes.
Class:	Set to 101h
Instance:	Set to 0
Attribute:	Set to 0
Source Element:	Select the variable source you created to hold the command request
Source Length:	Select the number of bytes to transfer from the variable source
Destination:	Select the destination variable you created to hold the command reply/response

7. Under the Communication tab, set the desired Module path.

8. Click OK to return to the Tag Properties dialog box.

9. Click OK again to close the dialog box.

10. If disabling the cache connection, go to Controller tags > Message Tag and set the Message tag name: EN_CC to 0.

Table 56: Service Codes

Service Code	Function Name	Read/Write	Description
0x0-0x4c	Reserved. Do not use.		
0x4b	GetWord	Read	Reads data from the word database.
0x4c	SetWord	Write	Writes data to the word database.
0x4d	GetByte	Read	Reads data from the byte database.
0x4e	SetByte	Write	Writes data to the byte database.
0x4f	GetPackedBit	Read	Reads data from the packed bit database.
0x50	SetPackedBit	Write	Writes data to the packed bit database.
0x51- 0x7f	Reserved for future Set/Get or Read/Write functionality.		
0x80+	Reserved for future Cyclic functionality.		

A.4 Data Type Structures

The following tables describe the data type structures used to send the request through the CIP Generic message block. For examples, refer to Section A.4, Sample Database Code.

Table 57: RSLogix Data Type GetWord Command

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of words to get
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Set to zero during command

Table 58: RSLogix Data Type GetWord Reply

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equimpment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of words read
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Returned status. For details, refer to the Status table.
DATA	INT[231] (16-bit)	Decimal	Data buffer, in words

Table 59: RSLogix Data Type SetWord Command

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equimpment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of words to be set
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Set to zero during command
DATA	INT[231] (16-bit)	Decimal	Data buffer, in words

Table 60: RSLogix Data Type SetWord Reply

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equimpment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of words read
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Returned status

Table 61: RSLogix Data Type GetByte Command

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bytes to get
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Set to zero during command

Table 62: RSLogix Data Type GetByte Reply

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bytes read
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Returned status. For details, refer to the Status table.
DATA	SINT[462] (8 bit)	Decimal	Data buffer, in bytes

Table 63: RSLogix Data Type SetByte Command

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bytes to be set
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Set to zero during command
DATA	SINT[462] (8 bit)	Decimal	Data buffer, in bytes

Table 64: RSLogix Data Type SetByte Reply

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bytes read
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Returned status

Table 65: RSLogix Data Type GetBit Command

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bits to get
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Set to zero during command

Table 66: RSLogix Data Type GetWord Reply

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of Bits Read
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Returned status. For details, refer to the Status table.
DATA	SINT[462] (8 bit)	Decimal	Data buffer, in bytes. The bits are packed into bytes and any extra bits returned are zero.

Table 67: RSLogix Data Type SetBit Command

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bits to be set
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Set to zero during command
DATA	SINT[462] (8 bit)	Decimal	Data buffer, in bytes. The bits are packed into bytes and any extra bits shall be zero.

Table 68: RSLogix Data Type SetBit Reply

Name	DataType	Radix	Description
Board Number	INT (16-bit)	Decimal	Always 1
Channel Number	INT (16-bit)	Decimal	Always 0
Equipment Number	INT (16-bit)	Decimal	Always 0
Size	INT (16-bit)	Decimal	Number of bits read
Offset	INT (16-bit)	Decimal	Data offset
Pad	INT (16-bit)	-	This value is added by RSLinx automatically (due to 32-bit alignment)
Status	DINT (32-bit)	Decimal	Returned status

These data types may be entered in the CLX Program’s user-defined data types to make programming easier. The user-defined Data Types can be found in the sample ladder programs included on the CD.

In the following example, user-defined SET_WORD_COMMAND and SET_WORD_REPLY types are used to create the SET_WORD_REQUEST and SET_WORD_REPLY tags.

Figure 131: Example of SETWORD Command

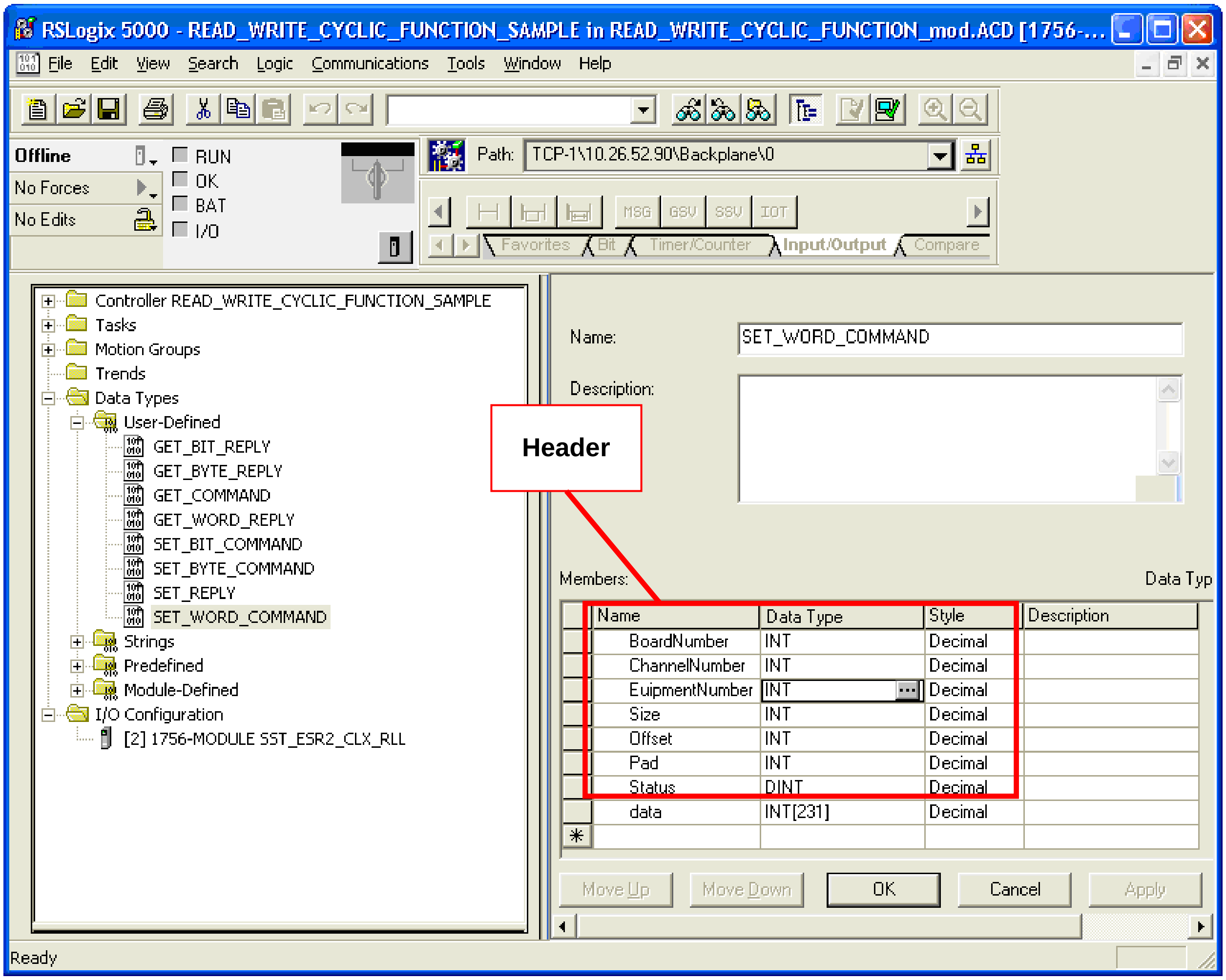


Figure 132: SET_WORD_REQUEST Tag (Assigned to Source Element for Message)

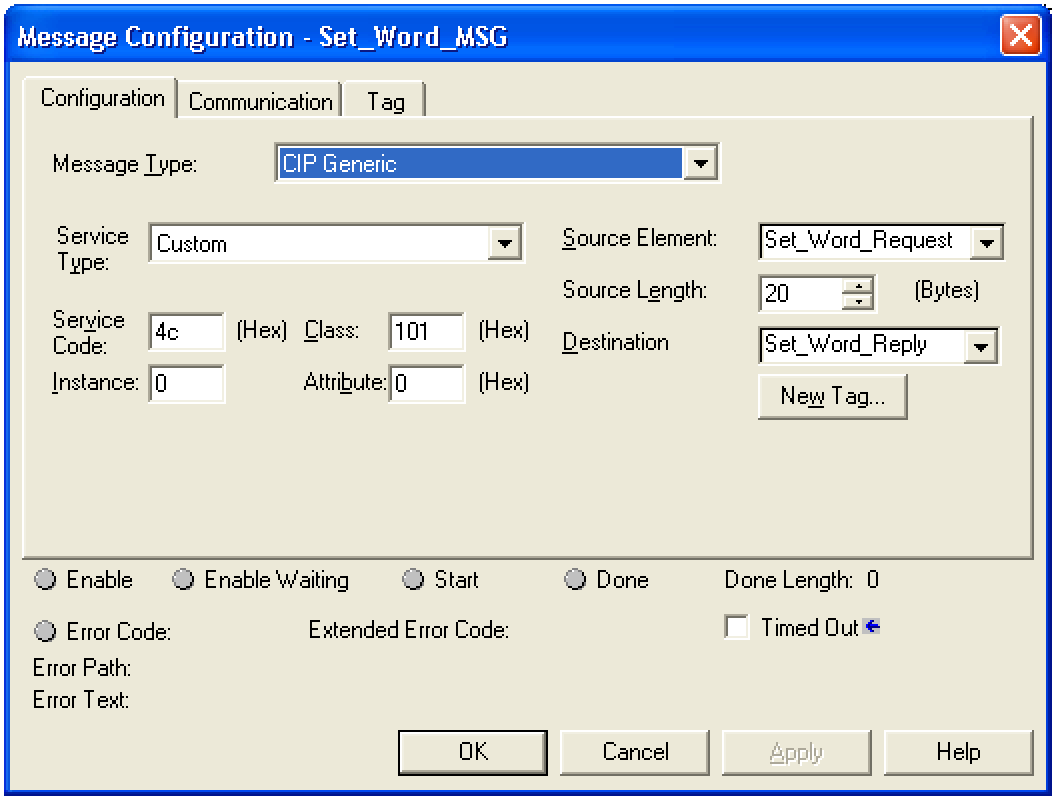
P	Tag Name	Alias For	Type	Style
☑	⊟-Set_Word_Request		SET_WORD_COMMAND	
	⊞-Set_Word_Request.BoardNumber		INT	Decimal
	⊞-Set_Word_Request.ChannelNumber		INT	Decimal
	⊞-Set_Word_Request.EuipmentNumber		INT	Decimal
	⊞-Set_Word_Request.Size		INT	Decimal
	⊞-Set_Word_Request.Offset		INT	Decimal
	⊞-Set_Word_Request.Pad		INT	Decimal
	⊞-Set_Word_Request.Status		DINT	Decimal
	⊞-Set_Word_Request.data		INT[231]	Decimal

Figure 133: SET_WORD_REPLY Tag (assigned to Destination for Message)

P	Tag Name	Alias For	Type	Style
☐	⊟-Set_Word_Reply		SET_REPLY	
	⊞-Set_Word_Reply.BoardNumber		INT	Decimal
	⊞-Set_Word_Reply.ChannelNumber		INT	Decimal
	⊞-Set_Word_Reply.EquipmentNumber		INT	Decimal
	⊞-Set_Word_Reply.Size		INT	Decimal
	⊞-Set_Word_Reply.Offset		INT	Decimal
	⊞-Set_Word_Reply.Pad		INT	Decimal
	⊞-Set_Word_Reply.Status		DINT	Decimal

In the following example, a CIP word message is set up to send 2 words of data. To calculate the source length in the message, add the Header Length (16 bytes) and 2 words (4 bytes). For details on the Header Length, refer to Table 34.

Figure 134: Set Word Dialog



Note

If the module is inhibited and there are outstanding CIP messages have been requested, they may enter an error state. The error bit will need to be reset once the program returns to Run mode.

A.5 CIP Messaging Sample Database Code

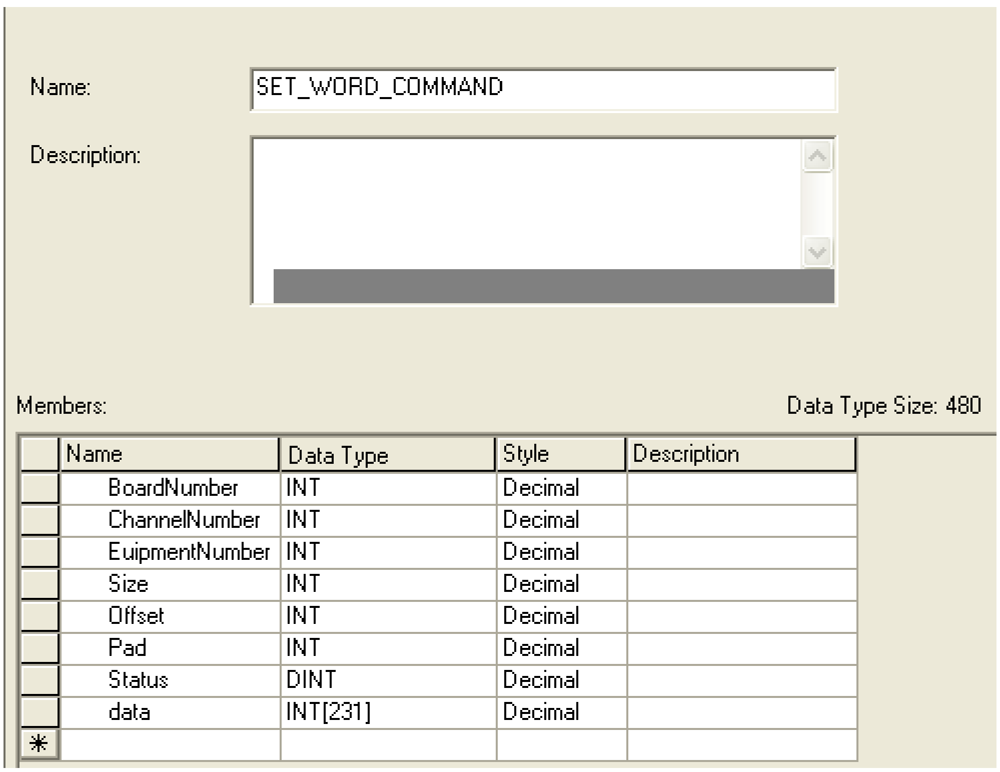
A.5.1 Database Example

This example shows how to send SET/GET CIP messages to the ERS2-CLX-RLL Module.
Send CIP messages as either connected or unconnected.

When use the connected type, ensure that the Cache Connections is selected for the message, and set the Message Request source length accordingly (16 byte Header + number of bytes of data).

All Get/Set Message Requests/Responses start with a 16-byte Header, as shown below.

Figure 135: Database Example



All Sample Ladder code includes user-defined data types that you can use to create the Requests and Reply tags for your CIP Messages.

Figure 136: User-Defined Data Types

- Data Types
  - User-Defined
    - GET_BIT_REPLY
    - GET_BYTE_REPLY
    - GET_COMMAND
    - GET_WORD_REPLY
    - SET_BIT_COMMAND
    - SET_BYTE_COMMAND
    - SET_REPLY
    - SET_WORD_COMMAND

Figure 137: Controller Tags

+ Get_Bit_Reply	{...}	{...}		GET_BIT_REPLY
+ Get_Byte_Reply	{...}	{...}		GET_BYTE_REPLY
+ Get_Bit_Request	{...}	{...}		GET_COMMAND
+ Get_Byte_Request	{...}	{...}		GET_COMMAND
+ Get_Word_Request	{...}	{...}		GET_COMMAND
+ Get_Word_Reply	{...}	{...}		GET_WORD_REPLY
+ Set_Bit_Request	{...}	{...}		SET_BIT_COMMAND
+ Set_Byte_Request	{...}	{...}		SET_BYTE_COMMAND
+ Set_Bit_Reply	{...}	{...}		SET_REPLY
+ Set_Byte_Reply	{...}	{...}		SET_REPLY
+ Set_Word_Reply	{...}	{...}		SET_REPLY
+ Set_Word_Request	{...}	{...}		SET_WORD_COMMAND

The Get command can be used for any Get DataBase Request Tag (bit, byte, word), as it's always 16 bytes long. The total length for any CIP message request or response cannot exceed 480 bytes (16 bytes standard header + 462 bytes available for data + two remaining bytes, which are used by the Controller).

With a single CIP message, it is able to read or write a maximum of 462 bytes.

If a CIP message transfer is interrupted by a inhibit or program download, it may initially get an error of 0x1f (error processing connection-related service). To fix the problem, clear the CIP message's ER bit and resend the message. For details, refer to Rung 2 of the example ladder code, below.

The default unconnected timeout is 30 Seconds. If a CIP message times out on controller, it will transition to PROG mode until the timeout has expired. In this case, the Message Unconnected Timeout should be reduced

Figure 138: Get_Bit CIP Message Example

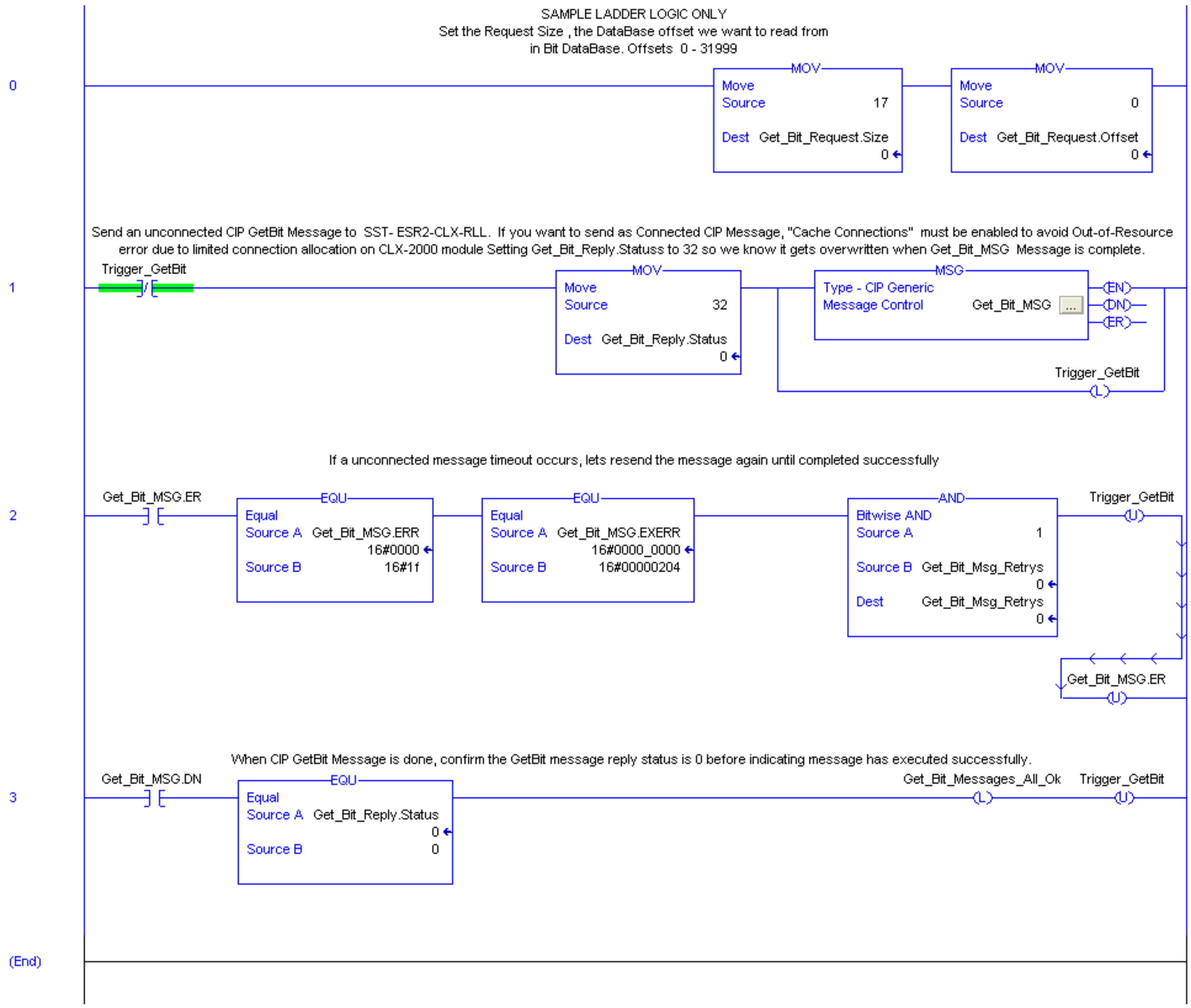
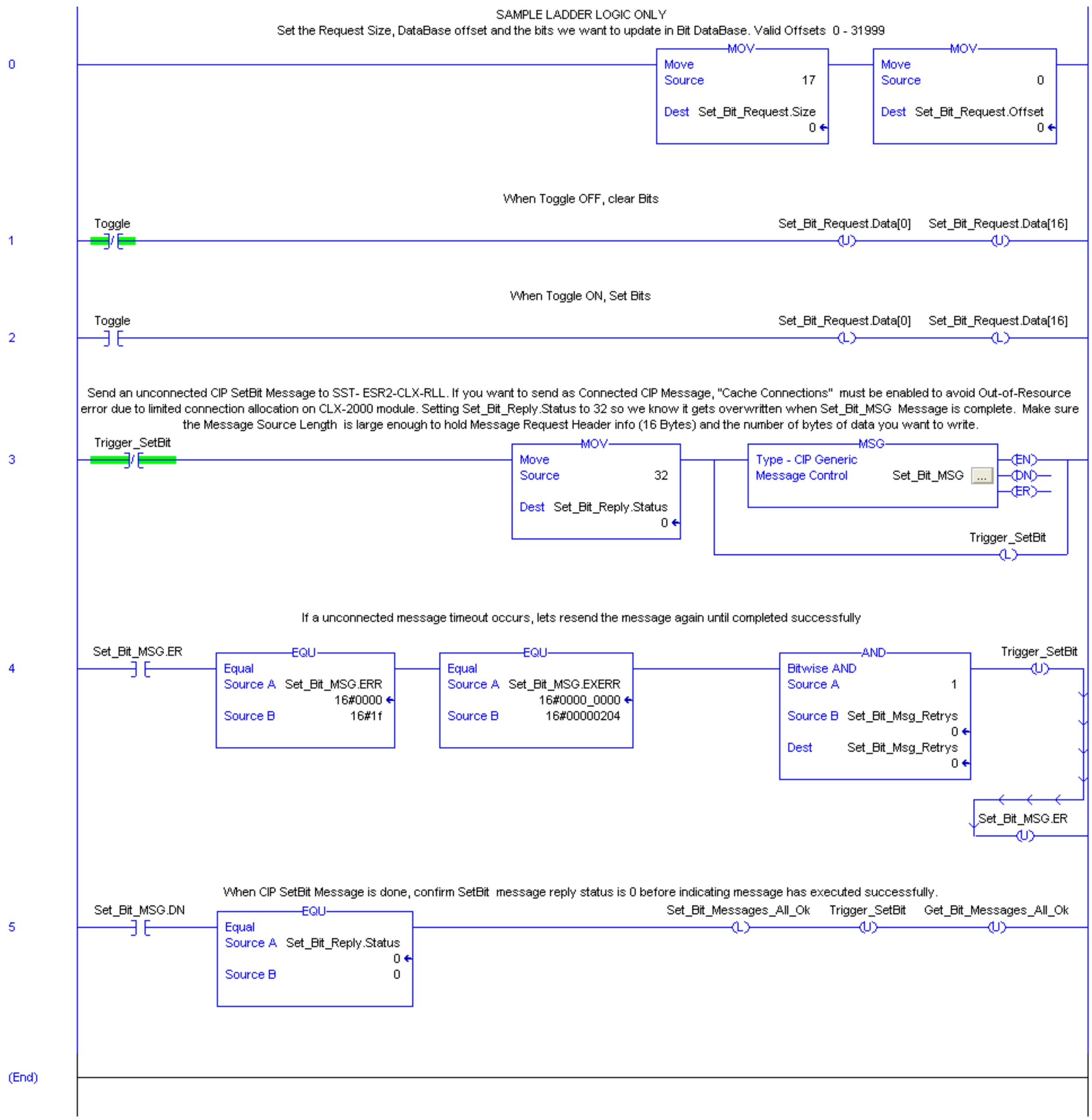


Figure 139: Set_Bit CIP Message Example



For other CIP Message type examples (Get/Set BYTE and WORD), refer to the RSLogix5000 V13.0 Sample ladder program, "GET_SET_SAMPLE_LADDER.ACD", which is included on the ControlLogix Communication Module CD.

B

Setting up the RSLinx Driver

Appendix Sections:

- Installing and Configuring the RSLinx Driver

Note

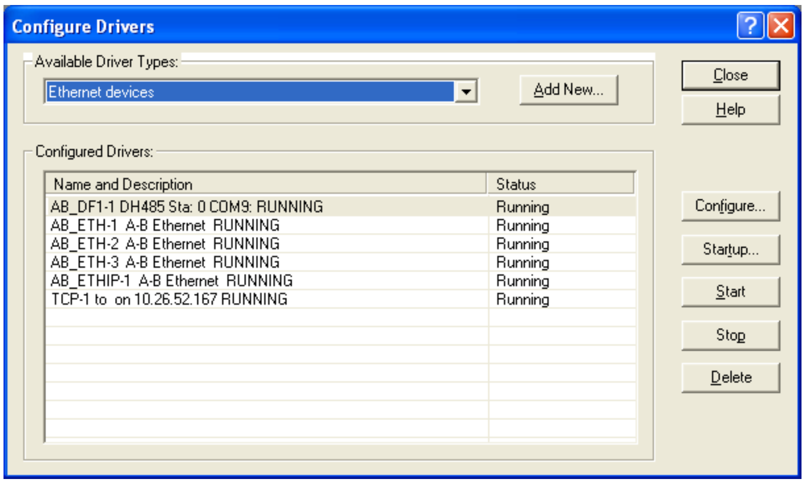
This Appendix assumes using RSLinx OEM/Professional software version 2.5.

B.1 Installing and Configuring the RSLinx Driver

1. Start the RSLinx software.

2. Under the Configuration menu, select Configure Drivers. The Configure Drivers dialog is displayed.

Figure 140: Configure Drivers Dialog



Note

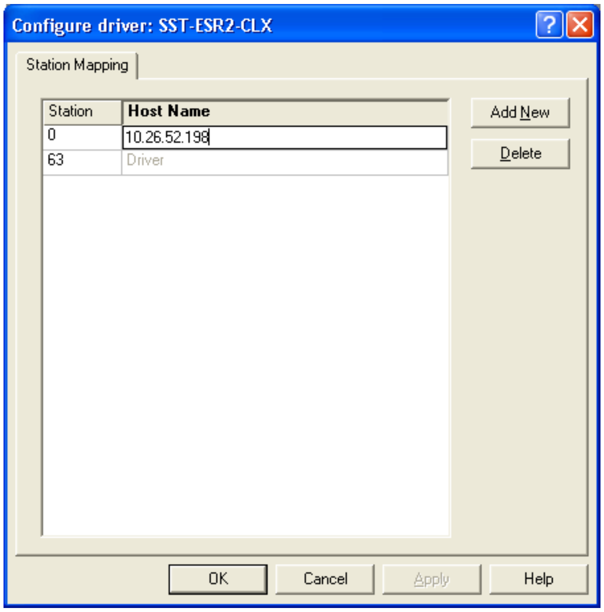
The Remote Devices via Linx Gateway driver is not supported with the RSLinx Classic Profession/Gateway v2.54 or higher. Valid drivers are Ethernet devices or Ethernet/IP Driver.

3. From the Available Driver Types list, select **Ethernet devices**, and click Add New.

4. Enter the name of the new driver. For example, “SST-ESR2-CLX”.

5. Enter the IP address of the ControlLogix Backplane you want to communicate with, and click OK.

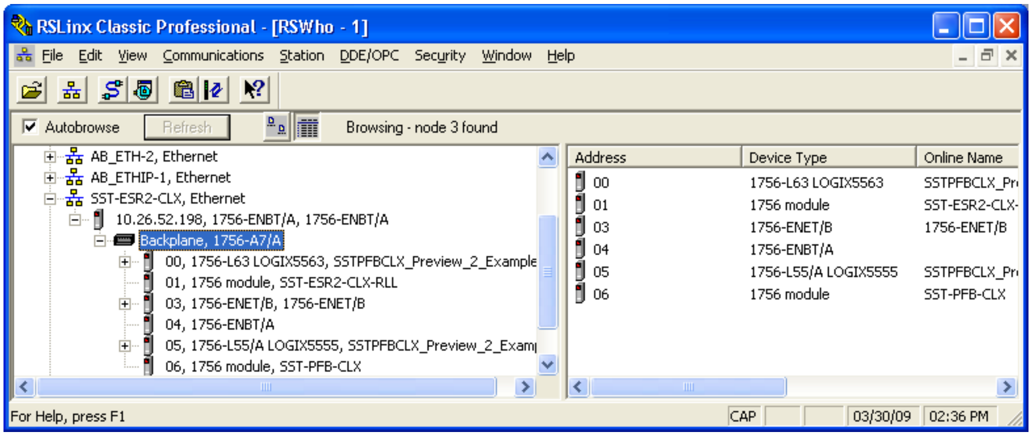
Figure 141: Configure Ethernet Devices Dialog



6. To verify that all drivers are able to communicate with the Backplane, follow these steps:

- Under the Communication menu, select RSWho. The RSWho window is displayed.

Figure 142: RSWho Window



- Click on the driver and check if the Backplane unit is viewable. the Module installed on the Backplane should be shown, along with a CPU and an Ethernet coupler (1756-ENET).

C

Error Recovery

Appendix Sections:

- Error Recovery

C.1 Error Recovery

If a bad configuration has been downloaded to the Module and reinitializing doesn't work, use either the USB port or hardware pin configuration to download a good configuration.

Error Recovery Using the USB Port

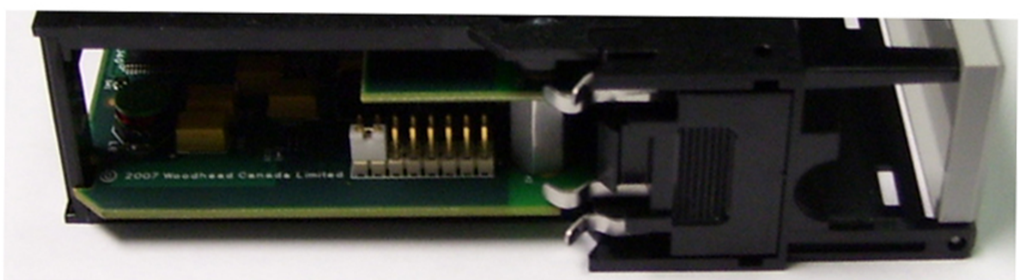
For instructions, refer to Downloading Configuration Using USB Port.

Error Recovery Using the Hardware Pin Configuration

To download a good configuration using the hardware pin configuration, follow these steps:

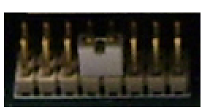
1. Remove the Module from the rack.
2. Find J4 and remove it.
3. Position as follows:

Figure 143: Default Position, Pins 14 and 16



4. Stop the configuration from running by shorting pins 8 and 10 together.

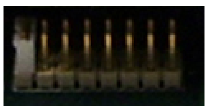
Figure 144: Pins 8 and 10



5. Re-insert the Module and power up.

6. Reinitialize the Module.
7. Remove the Module and switch the jumper back to the default position.
8. Re-insert the Module. The new configuration should take effect.
9. Erase the configuration and all tasks by shorting pins 15 and 16 together.

Figure 145: Pins 15 and 16



10. Re-insert the Module and power up.
11. Remove the Module and switch the jumper back to the default position.
12. Re-insert the Module.
13. Copy the .out files from your task directory to the configuration directory.
14. Reinitialize the Module. The Module should be back to its factory configuration.

D
Technical Specifications

Appendix Sections:

- Technical Specifications

D.1 Technical Specifications

Table 69: Technical Specifications for the SST-ESR2-CLX-RLL Module

Part Number	SST-ESR2-CLX-RLL
Function	CLX-5000 Module for Ethernet and Serial networks.
Description	• Free scale PPC processor • 16 MB of local FPGA RAM • 124 MB of onboard PPC memory • 8 MB of sectored flash memory for storing program and configuration data • 2 MB of sectored serial flash memory for FPGA program and configuration data
CLX Interface	248 words of Output data and 248 words of Input data.
	250 words of Status data
Database	Direct access to 30720 words of modules database from backplane thru AOI
Environmental	Storage temperature: -40°C to 85°C
	Operating temperature: 0°C to 60°C
	Operating RH level: 5% to 95%, non-condensing
	Pollution Degree 1 - no pollution or only non-conductive or non-corrosive pollution
Backplane Current Consumption	850 mA @ 5 VDC 1.75 mA@24 VDC
MTBF value (Mean Time Between Failure)	56 Years (Standard used to attain value: Telecordia SR332, Issue 1, Specification for Reliability Prediction)

Table 70: Technical Specifications for the SST-SR4-CLX-RLL Module

Part Number	SST-SR4-CLX-RLL
Function	CLX-5000 Module for Ethernet and Serial networks.
Description	• Freescale PPC processor • 16 MB of local FPGA RAM • 124 MB of onboard PPC memory • 8 MB of sectored flash memory for storing program and configuration data. • 2 MB of sectored serial flash memory for FPGA program and configuration data.
CLX Interface	248 words of Output data and 248 words of Input data.
	250 words of Status
Database	Direct access to 30720 words of modules database from backplane thru AOI
Environmental	Storage temperature: -40°C to 85°C
	Operating temperature: 0°C to 60°C
	Operating RH level: 5% to 95%, non-condensing
	Pollution Degree 1 - no pollution or only non-conductive or non-corrosive pollution
Backplane Current Consumption	1005 mA @ 5 VDC; 1.75 mA @ 24 VDC
MTBF value (Mean Time Between Failure)	44 Years (Standard used to attain value: Telecordia SR332, Issue 1, Specification for Reliability Prediction)

E

Approvals

Appendix Sections:

- CE/EU Compliance
- RoHS Compliance
- FM Compliance for Class 1 Div 2 Groups A-D
- FCC Compliance
- Rep. of Korea Compliance
- Marin Certification (RINA)

E.1 CE/EU Compliance

Marking of this equipment with the CE symbol indicates compliance with European Council Directive 89/336/EEC - The EMC Directive as amended by 92/31/EEC and 93/68/EEC.

Warning

This is a Class A product. In a domestic environment, this product may cause radio interference, in which case you may be required to take adequate measures.

Caution

This equipment is neither designed for, nor intended for operation in installations where it is subject to hazardous voltages and hazardous currents.

Note

To maintain compliance with the limits and requirements of the EMC Directive, it is required to use quality interfacing cables and connectors when connecting to this device. Refer to the cable specifications in the Hardware Guide for the selection of cable types.

Note

The Backplane voltage supply for this equipment must be delivered as Separated Extra Low Voltage (SELV).

E.2 RoHS Compliance

SST-ESR2-CLX-RLL and SST-SR4-CLX-RLL are RoHS II compliant and fulfill the definition and restrictions defined under Directive 2011/65/EU of the European Parliament and of the Council of June 8, 2011 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (EEE).

E.3 FM Compliance for Class 1 Div 2 Groups A-D

Product comply with FM Approvals requirements for Class I, Division 2, Groups A, B, C , D Hazardous locations, based on the following standards: C22.2 No. 142: 1987 (CSA) , C22.2 No. 213:1987 (CSA) , Class 3600:1998 (FM Approvals), Class 3611:2004 (FM Approvals) and Class 3810: 2005 (FM Approvals).

E.4 FCC Compliance

Product comply with FCC Part 15, Subpart B, Class A – Unintentional Radiators.

E.5 Rep. of Korea Compliance

Product comply with the RRA Public Notifications 2012-13 (2012.6.28) and 2012-14 (2012.6.28).

E.6 Marine Certification (RINA)

Product comply with RINA Rules for the Classification of Ships – Part C, Machinery, Systems and Fire Protection.

F

Warranty and Support

Appendix Sections:

- Warranty
- Reference Documents
- Technical Support
- Getting Help
- Feedback

F.1 Warranty

For warranty information, refer to
http://www.molex.com/images/woodhead/woodhead_limited_warranty.pdf

F.2 Reference Documents

For...	Read this Document...	Document Number
A-B power supply specifications	CLX 5000 Modular Style Installation & Operation Manual	Allen-Bradley Publication 1747-6.2
RSLogix information	ControlLogix 5000 Controllers General Instruction Set Reference Manual	AB Publication number 1756-6.4.1 and 1756-RM003A-US-P
CLX Chassis installation	ControlLogix Chassis Installation Instructions	AB Publication number 1756-5.80
Honeywell PlantScape information	PlantScape Process System and Controller Product Overview	PS03-140 Release 400

F.3 Technical Support

Please ensure that you have the following information readily available before calling for Technical Support:

- Card type and serial number
- Computer's make, model, CPU speed and hardware configuration (other cards installed)
- Operating system type and version
- Details of the problem you are experiencing: firmware module type and version, target network and circumstances that may have caused the problem

F.4 Getting Help

Technical support is available during regular business hours by telephone, fax or email. The Molex web site contents also useful information available by clicking the link below:

Molex Support and Download

- Downloads center
- Support Request Form
- Knowledge Base
- Worldwide technical support contacts

F.5 Feedback

We would gladly welcome any suggestions or clarifications to improve this manual. Please send us your comments using our help link above.