Allen-Bradley

DriveLogix™ 5730 Controller

for PowerFlex® 700S Drives with Phase II Control

Firmware Version

13.XX

User Manual

Rockwell Automation

Important User Information

Solid state equipment has operational characteristics differing from those of electromechanical equipment. *Safety Guidelines for the Application, Installation and Maintenance of Solid State Controls* (Publication SGI-1.1 available from your local Rockwell Automation sales office or online at **http://www.rockwellautomation.com/literature**) describes some important differences between solid state equipment and hard-wired electromechanical devices. Because of this difference, and also because of the wide variety of uses for solid state equipment, all persons responsible for applying this equipment must satisfy themselves that each intended application of this equipment is acceptable.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Throughout this manual, when necessary we use notes to make you aware of safety considerations.

WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

Important: Identifies information that is critical for successful application and understanding of the product.

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you:

- identify a hazard
- avoid the hazard
- recognize the consequences

Shock Hazard labels may be located on or inside the equipment (e.g., drive or motor) to alert people that dangerous voltage may be present.

Burn Hazard labels may be located on or inside the equipment (e.g., drive or motor) to alert people that surfaces may be at dangerous temperatures.

Overview

Who Should Use This Manual

This manual is intended for qualified personnel. You must be able to program and operate Adjustable Frequency AC Drive devices and programmable controllers.

Purpose of this Manual

This manual guides the development of projects for DriveLogix controllers. It provides procedures on how to establish communications:

- with the following networks
  - ControlNet
  - DeviceNet
  - EtherNet/IP
  - serial
- with the following devices
  - PowerFlex® 700S drive
  - controllers
  - I/O
  - workstations
  - PanelView terminals

Related Documentation

Allen-Bradley publications are available on the internet at *www.rockwellautomation.com/literature*.

These core documents address the Logix5000 family of controllers:

If you are:	Use this publication:
a new user of a Logix5000 controller This quick start provides a visual, step-by-step overview of the basic steps you need to complete to get your controller configured and running.	Logix5000 Controllers Quick Start publication 1756-QS001
an experienced user of Logix5000 controllers This system reference provides a high-level listing of configuration information, controller features, and instructions (ladder relay, function block diagram, and structured text).	Logix5000 Controllers System Reference publication 1756-QR107
any user of a Logix5000 controller This common procedures manual explains the common features and functions of all Logix5000 controllers.	Logix5000 Controllers Common Procedures publication 1756-PM001

DriveLogix-specific information is also available:

For	Read this document	Document number
Information on the DriveLogix Instruction Set	Logix5000 Controllers General Instruction Set Reference Manual	1756-RM003…
Information on function block programming Logix controllers.	Logix5000 Controllers Process Control/Drives Instruction Set Reference Manual	1756-RM006…
Execution times and memory use for instructions	Logix5000 Controllers Execution Time and Memory Use Reference Manual	1756-RM087…
Information on selecting CompactLogix and Compact I/O components and modules	Compact I/O Selection Guide	1769-SG001…
Information on installing, configuring, and using Compact Analog I/O modules	Compact I/O Analog Modules User Manual	1769-UM002…
Information on installing, configuring and using PowerFlex® 700S drives	PowerFlex 700S Phase II User Manual	20D-UM006…
Information on the DriveLogix Motion Control Instruction Set	Logix Controller Motion Instruction Set	1756-RM007…
Information on installing an ControlNet communications daughtercard (Coax)	ControlNet Communications Daughtercard Installation Instructions	1788-IN002…
Information on installing an ControlNet communications daughtercard (Fiber)	ControlNet Communications Daughtercard Installation Instructions	1788-IN005…
Information on installing an EtherNet/IP communications daughtercard	EtherNet/IP Communications Daughtercard Installation Instructions	1788-IN054…
Information on installing an DeviceNet communications daughtercard	DeviceNet Communications Daughtercard Installation Instructions	1788-IN053…
Information on installing 1769-SDN DeviceNet Scanner Module	Compact I/O 1769-SDN DeviceNet Scanner Module	1769-IN060…
Information on using 1769-SDN DeviceNet Scanner Module	Compact I/O 1769-SDN DeviceNet Scanner Module	1769-UM009…
Information on converting from Phase I PowerFlex 700S and DriveLogix5720 to Phase II PowerFlex 700S and DriveLogix5730	PowerFlex 700S Conversion Guide Phase I to Phase II Control	20D-AT001…
Information on the basic installation of PowerFlex 700S drives and DriveLogix5730 controllers	Quick Start - PowerFlex 700S High Performance AC Drive	20D-QS001…

Controller Firmware Revision

This revision on the DriveLogix 5730 User Manual corresponds to the following:

- version 13 and later controller firmware
- version 13 and later RSLogix 5000 programming software
- version 3.02 and later DriveExecutive programming software

General Precautions

Class 1 LED Product

ATTENTION: Hazard of permanent eye damage exists when using optical transmission equipment. This product emits intense light and invisible radiation. Do not look into module ports or fiber optic cable connectors.

ATTENTION: This drive contains **ESD** (Electrostatic Discharge) sensitive parts and assemblies. Static control precautions are required when installing, testing, servicing or repairing this assembly. Component damage may result if ESD control procedures are not followed. If you are not familiar with static control procedures, reference A-B publication 8000-4.5.2, "Guarding Against Electrostatic Damage" or any other applicable ESD protection handbook.

ATTENTION: Only **qualified personnel** familiar with the PowerFlex 700S Drive and associated machinery should plan or implement the installation, start-up and subsequent maintenance of the system. Failure to comply may result in personal injury and/or equipment damage.

ATTENTION: To avoid an electric shock hazard, verify that the voltage on the bus capacitors has discharged before performing any work on the drive. Measure the DC bus voltage at the +DC & –DC terminals of the Power Terminal Block. The voltage must be zero.

Notes:

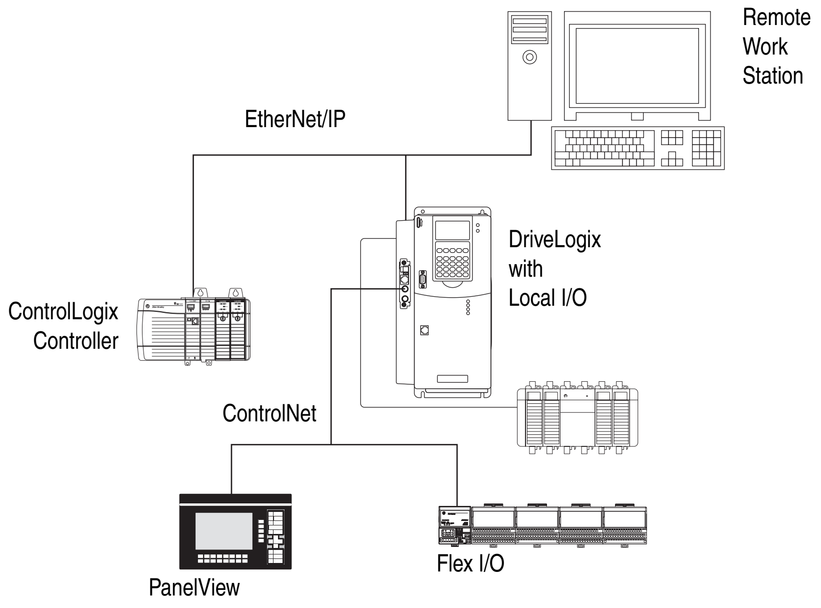
What is DriveLogix5730?

The DriveLogix controller is part of the Logix environment. The DriveLogix controller provides a distributed control system built on these components:

- The DriveLogix5730 controller has one RS-232 port. The controller supports the Logix instructions.
- RSLogix 5000 programming software that supports every Logix controller.
- Direct connection to host PowerFlex 700S drive.
- Compact I/O modules that provide a compact, DIN-rail or panel mounted I/O system.
- Embedded EtherNet/IP option provides communication over an EtherNet/IP network
- 1788 communication daughtercard that provides communication over a standards-based ControlNet, EtherNet/IP, DeviceNet or third party network.

The newer DriveLogix5730 controller offers significant performance and capacity improvements over the DriveLogix5720 controller. It offers:

- increased user memory up to 1.5 Mbytes
- CompactFlash for non-volatile memory storage
- extended I/O capacity up to 16 I/O modules
- integrated EtherNet/IP support, including control of distributed I/O
- Run/Rem/Prog switch



Loading Controller Firmware

De-energizing the Drive to Connect or Disconnect a Cable

ATTENTION: Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of loading controller firmware you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive’s input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with loading firmware.

Loading Firmware

The controller ships with working firmware. You may decide to upgrade the firmware. To load firmware, you can use:

- ControlFlash utility that ships with RSLogix 5000 programming software.
- AutoFlash that launches through RSLogix 5000 software when you download a project to a controller that does not have the current firmware.
- a 1784-CF64 CompactFlash card with valid memory already loaded.

The firmware is available with RSLogix 5000 software or you can download it from the support website:

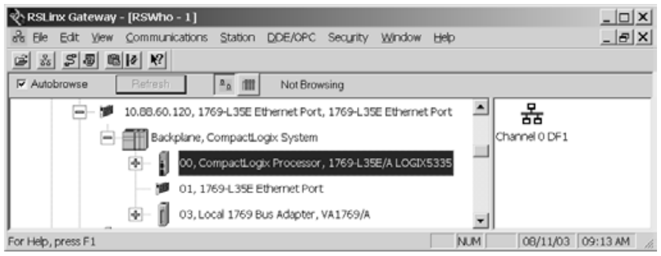
1. Go to *http://support.rockwellautomation.com/*
2. In the left column (frame), select Firmware Updates under Technical Support
3. Select the firmware revision.

   The download process will require you to enter the serial number of your RSLogix 5000 programming software.

If you load (flash) controller firmware via the ControlFlash or AutoFlash utilities, you need a serial or EtherNet/IP connection to the controller. Flashing via an EtherNet/IP connection is faster than the serial connection.

The controller’s EtherNet/IP configuration settings are maintained during a flash process.

If you load firmware via an EtherNet/IP connection, browse through the network port, across the virtual backplane, and select the appropriate controller.


RSLinx Gateway - [RSWho - 1]
File Edit View Communications Station DDE/OPC Security Window Help
Autobrowse
Refresh
Not Browsing

```
10.88.60.120, 1769-L35E Ethernet Port, 1769-L35E Ethernet Port
  Backplane, CompactLogix System
    00, CompactLogix Processor, 1769-L35E/A LOGIX5335
    01, 1769-L35E Ethernet Port
    03, Local 1769 Bus Adapter, VA1769/A
```

Channel 0 DF1
For Help, press F1
NUM
08/11/03
09:13 AM

Using ControlFlash to load firmware

You can use ControlFlash to load firmware through either an Ethernet connection (an IP address must already be assigned to the Ethernet port) or a serial connection.

1. Make sure the appropriate network connection is made before starting.
2. Start the ControlFlash utility. Click Next when the Welcome screen appears.
3. Select the catalog number of the controller and click Next.
4. Expand the network until you see the controller. If the required network is not shown, first configure a driver for the network in RSLinx software.

   If you use an Ethernet connection to load the firmware (which is much faster than the serial connection), the utility will require a valid IP address before connecting to the controller.
5. Select the controller and click OK
6. Select the revision level to which you want to update the controller and click Next.
7. To start the update of the controller, click Finish and then click Yes.
8. After the controller is updated, the status box displays Update complete. Click OK.
9. To close ControlFlash software, click Cancel and then click Yes.

Using AutoFlash to load firmware

You can use AutoFlash to load firmware through either an Ethernet connection (an IP address must already be assigned to the Ethernet port) or a serial connection.

1. Make sure the appropriate network connection is made before starting.
2. Use RSLogix 5000 programming software to download a controller project. If the processor firmware does not match that project revision, AutoFlash automatically launches.
3. Select the catalog number of the controller and click Next.
4. Expand the network until you see the controller. If the required network is not shown, first configure a driver for the network in RSLinx software.

   If you use an Ethernet connection to load the firmware (which is much faster than the serial connection), the utility will ask for a valid IP address before connecting to the controller.
5. Select the controller and click OK
6. Select the revision level to which you want to update the controller and click Next.
7. To start the update of the controller, click Finish and then click Yes.
8. After the controller is updated, the status box displays Update complete. Click OK.
9. To close AutoFlash software, click Cancel and then click Yes.

Using a CompactFlash card to load firmware

If you have an existing DriveLogix5730 controller that is already configured and has firmware loaded, you can store the current controller user program and firmware on CompactFlash and use that card to update other controllers.

1. Store the controller user program and firmware of a currently configured DriveLogix5730 controller to the CompactFlash card.

   Make sure to select Load Image On Power-up when you save to the card.
2. Remove the card and insert it into a DriveLogix5730 controller that you want to have the same firmware and controller user program.
3. When you power up the second DriveLogix5730 controller, the image stored on the CompactFlash card is loaded into the controller.

Using CompactFlash

The 1784-CF64 CompactFlash card provides nonvolatile memory storage for the DriveLogix5730 controller. The card stores the contents of the controller memory (program logic and tag values) and the controller firmware at the time that you store the project. Storing information to the CompactFlash card is like storing a snapshot of controller memory at a given time.

ATTENTION: If you configured the CompactFlash card to “restore on power up” and you make changes to a project, such as online edits or changes to tag values, you must store the project to the CompactFlash card again after you make changes. Otherwise, your changes are not saved and you will lose those changes on the next power cycle to the controller.

Tag values stored in flash are a snapshot at the time of the store. During a program restore the processor tag values will be equal to tag data stored on flash.

The locking tab on the front of the controller helps hold the CompactFlash card in its socket.

ATTENTION: Do not remove the CompactFlash card while the controller is reading from or writing to the card, as indicated by a flashing green CF LED. This could corrupt the data on the card or in the controller, as well as corrupt the latest firmware in the controller.

The CompactFlash card supports removal and insertion under power.

ATTENTION: When you insert or remove the card while backplane power is on, an electrical arc can occur. This could cause an explosion in hazardous location installations.

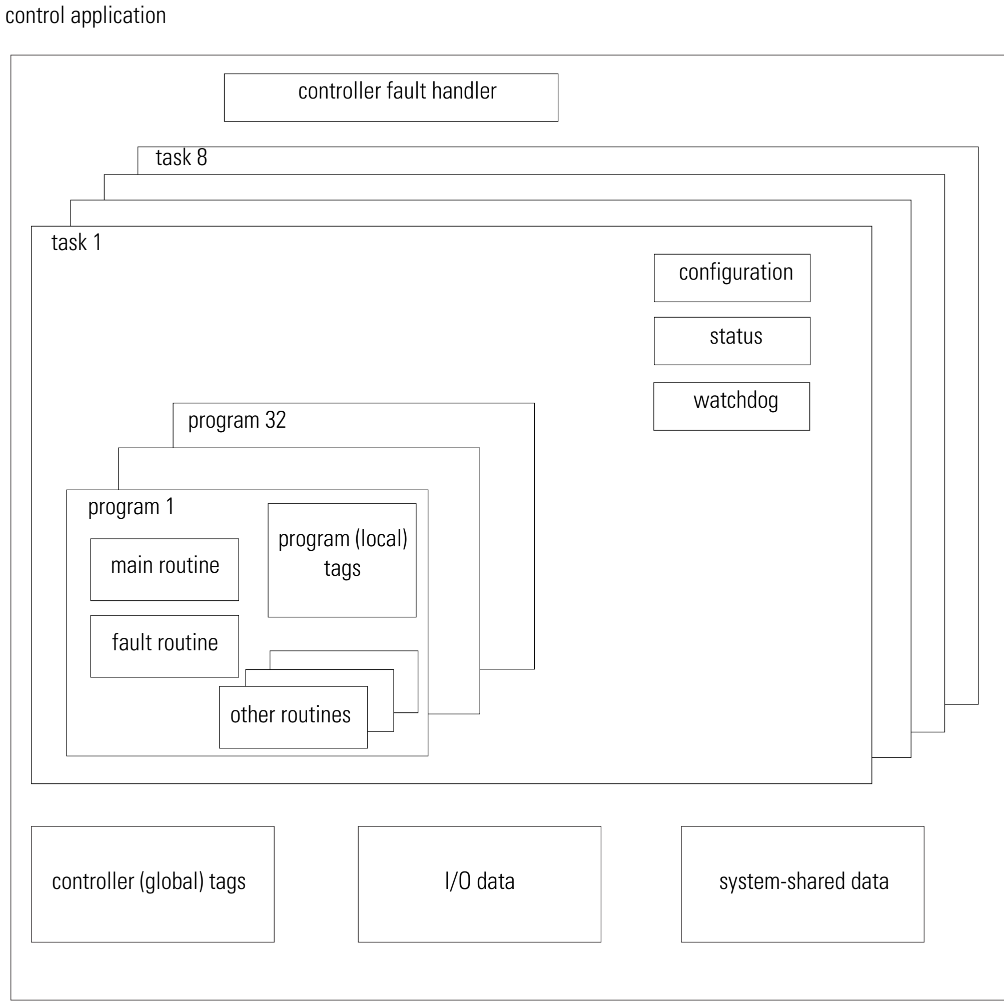
Be sure that power is removed or the area is nonhazardous before proceeding. Repeated electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance that can affect module operation.

See the *Logix5000 Controllers Common Procedures Programming Manual*, publication 1756-PM001 for steps on storing an image on the CompactFlash card.

Developing Programs

The controller operating system is a preemptive multitasking system that is IEC 1131-3 compliant. This environment provides:

- tasks to configure controller execution
- programs to group data and logic
- routines to encapsulate executable code written in a single programming language



Defining tasks

A task provides scheduling and priority information for a set of one or more programs. You can configure tasks as continuous, periodic, or event. Only one task can be continuous. The DriveLogix5730 controller supports as many as eight (8) tasks.

A task can have as many as 32 separate programs, each with its own executable routines and program-scoped tags. Once a task is triggered (activated), all the programs assigned to the task execute in the order in which they are grouped. Programs can only appear once in the Controller Organizer and cannot be shared by multiple tasks.

Specifying task priorities

Each task in the controller has a priority level. The operating system uses the priority level to determine which task to execute when multiple tasks are triggered. You can configure periodic tasks to execute from the lowest priority of 15 up to the highest priority of 1. A higher priority task will interrupt any lower priority task. The continuous task has the lowest priority and is always interrupted by a periodic task.

The DriveLogix5730 controller uses a dedicated periodic task at priority 7 to process I/O data. This periodic task executes at the RPI you configure for the CompactBus, which can be as fast as once every 1 ms. Its total execution time is as long as it takes to scan the configured I/O modules.

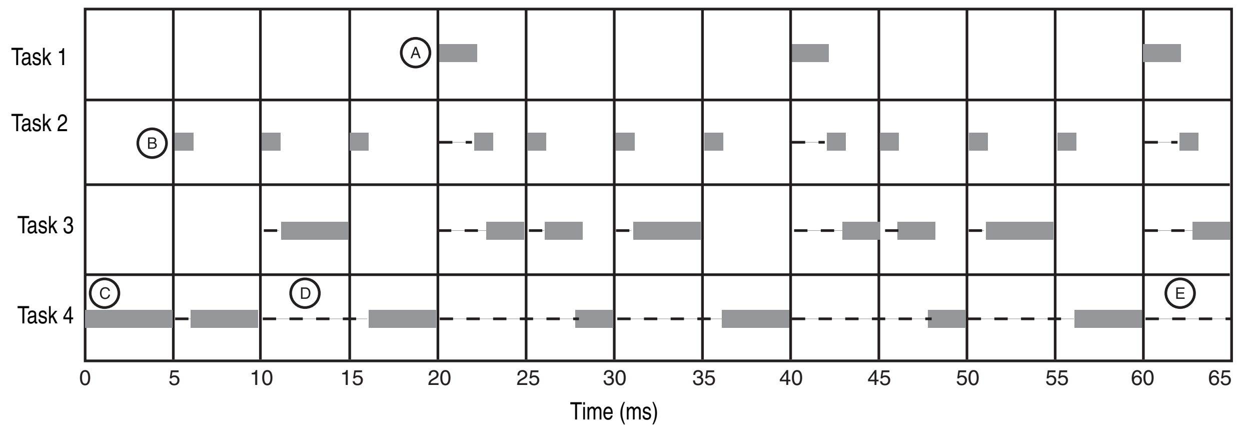
How you configure your tasks affects how the controller receives I/O data. Tasks at priorities 1 to 6 take precedence over the dedicated I/O task. Tasks in this priority range can impact I/O processing time. If you configure the I/O RPI at 1ms and you configure a task of priority 1 to 6 that requires 500 µs to execute and is scheduled to run every millisecond. This leaves the dedicated I/O task 500 µs to complete its job of scanning the configured I/O.

However, if you schedule two high priority tasks (1 to 6) to run every millisecond, and they both require 500 µs or more to execute, no CPU time would be left for the dedicated I/O task. Furthermore, if you have so much configured I/O that the execution time of the dedicated I/O task approaches 2 ms (or the combination of the high priority tasks and the dedicated I/O task approaches 2 ms) no CPU time is left for low priority tasks (8 to 15).

▶ **TIP:** For example, if your program needs to react to inputs and control outputs at a deterministic rate, configure a periodic task with a priority higher than 7 (1 through 6). This keeps the dedicated I/O task from affecting the periodic rate of your program. However, if your program contains a lot of math and data manipulation, place this logic in a task with priority lower than 7 (8 through 15), such as the continuous task, so that the dedicated I/O task is not adversely affected by your program.

The following example shows the task execution order for an application with periodic tasks and a continuous task.

Task:	Priority Level:	Task Type:	Example Execution Time:	Worst Case Completion Time:
1	5	20 ms periodic task	2 ms	2 ms
2	7	dedicated I/O task		
5 ms selected RPI	1	3 ms		
3	10	10 ms periodic task	4 ms	8 ms
4	none (lowest)	continuous task	25 ms	60 ms



Notes:

A. The highest priority task interrupts all lower priority tasks.

B. The dedicated I/O task can be interrupted by tasks with priority levels 1 to 6. The dedicated I/O task interrupts tasks with priority levels 8 to 15. This task runs at the selected RPI rate scheduled for the DriveLogix5730 system (2ms in this example).

C. The continuous task runs at the lowest priority and is interrupted by all other tasks.

D. A lower priority task can be interrupted multiple times by a higher priority task.

E. When the continuous task completes a full scan it restarts immediately, unless a higher priority task is running.

Defining programs

Each program contains program tags, a main executable routine, other routines, and an optional fault routine. Each task can schedule as many as 32 programs.

The scheduled programs within a task execute to completion from first to last. Programs that are not attached to any task show up as unscheduled programs. You must specify (schedule) a program within a task before the controller can scan the program.

Defining routines

A routine is a set of logic instructions in a single programming language, such as ladder logic. Routines provide the executable code for the project in a controller. A routine is similar to a program file or subroutine in a PLC or SLC controller.

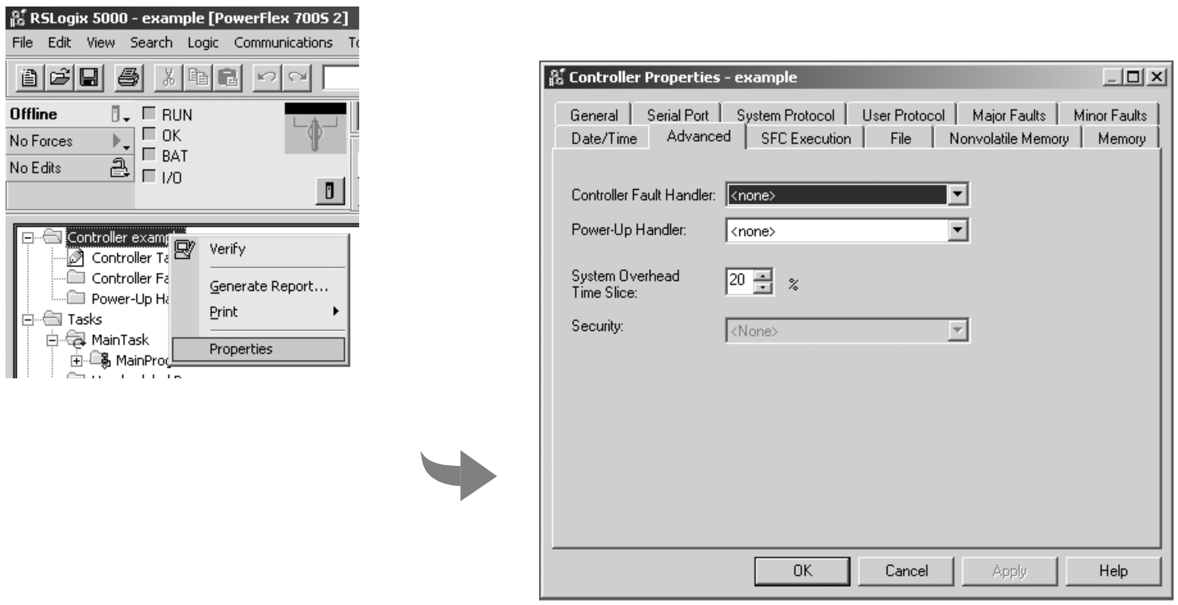
Each program has a main routine. This is the first routine to execute when the controller triggers the associated task and calls the associated program. Use logic, such as the Jump to Subroutine (JSR) instruction, to call other routines.

You can also specify an optional program fault routine. The controller executes this routine if it encounters an instruction-execution fault within any of the routines in the associated program.

Selecting a System Overhead Percentage

The Controller Properties dialog lets you specify a percentage for system overhead. This percentage specifies the percentage of controller time (excluding the time for periodic tasks) that is devoted to communication and background functions.

1. View properties for the controller and select the Advanced



Controller Fault Handler: <none>
Power-Up Handler: <none>
System Overhead Time Slice: 20 %
Security: <None>

System overhead functions include:

- communicating with programming and HMI devices (such as RSLogix 5000 software)
- responding to messages
- sending messages

The controller performs system overhead functions for up to 1 ms at a time. If the controller completes the overhead functions in less than 1 ms, it resumes the continuous task.

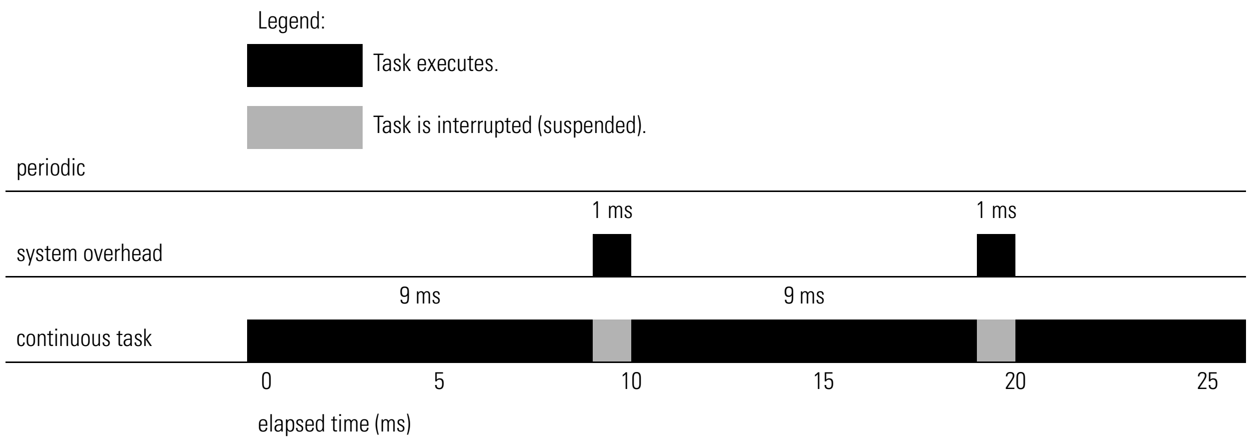
As the system overhead percentage increases, time allocated to executing the continuous task decreases. If there are no communications for the controller to manage, the controller uses the communications time to execute the continuous task. While increasing the system overhead percentage decreases execution time for the continuous task, it does increase communications performance. However, increasing the system overhead percentage also increases the amount of time it takes to execute a continuous task - increasing overall scan time.

The following table shows the ratio between the continuous task and the system overhead functions:

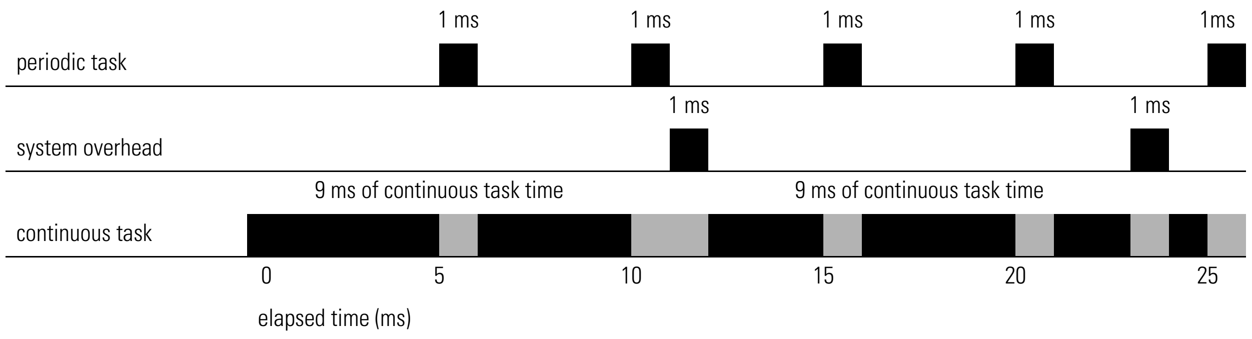
At this time slice:	The continuous tasks runs for:	And then overhead occurs for up to:
10%	9 ms	1 ms
20%	4 ms	1 ms
33%	2 ms	1 ms
50%	1 ms	1 ms

▶ **TIP:** For typical DriveLogix applications, a setting of 20-33% is recommended.

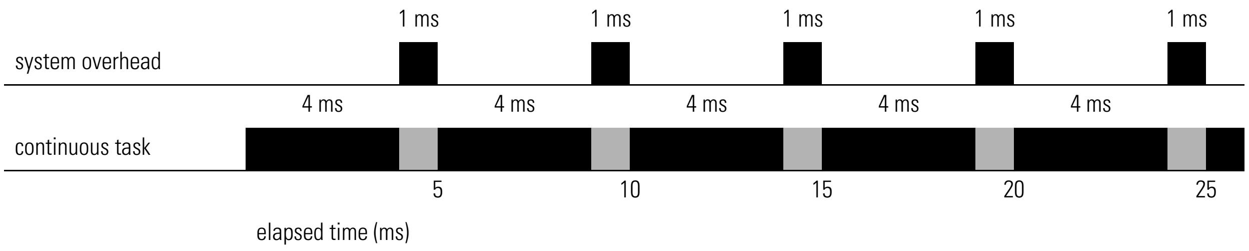
At the default time slice of 10%, system overhead interrupts the continuous task every 9 ms (of continuous task time), as illustrated below.



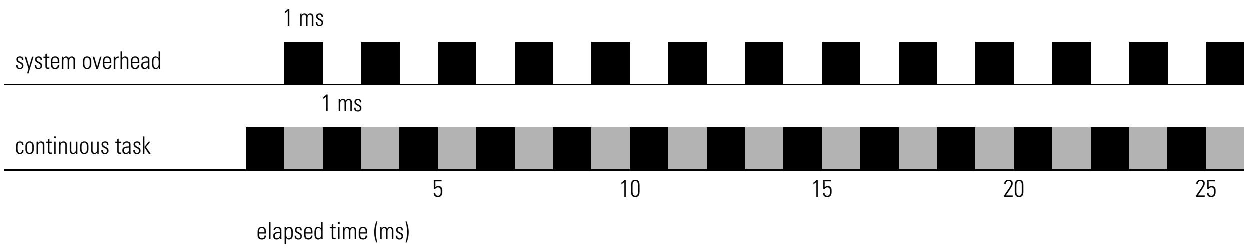
The interruption of a periodic task increases the elapsed time (clock time) between the execution of system overhead, as shown below.



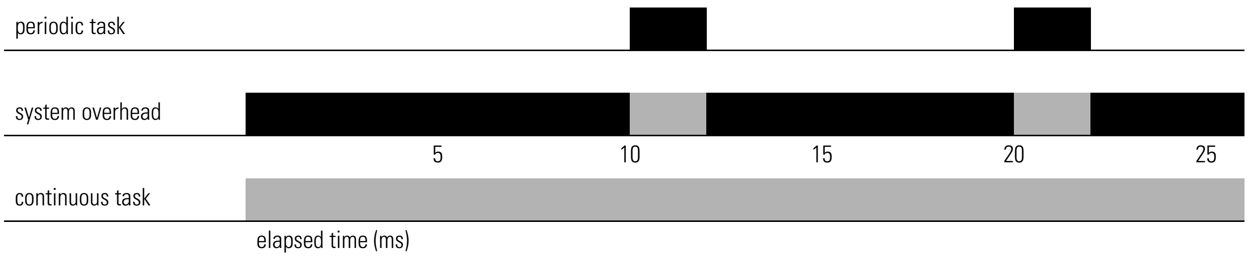
If you increase the time slice to 20%, the system overhead interrupts the continuous task every 4 ms (of continuous task time).



If you increase the time slice to 50%, the system overhead interrupts the continuous task every 1 ms (of continuous task time).



If the controller only contains a periodic task(s), the system overhead timeslice value has no effect. System overhead runs whenever a periodic task is not running.



Understanding the Virtual Backplane

The DriveLogix5730 system has a five-slot virtual backplane. The controller, drive and other components occupy different assigned slots on the backplane.

Virtual Backplane

Slot 0	Slot 1	Slot 2	Slot 3	Slot 4
DrvieLogix5730 Controller	Embedded EtherNet Option	PowerFlex 700S Drive	Compact I/O	NetLinx Daughtercard

Notes:

Placing and Configuring the Drive

De-energizing the Drive to Connect or Disconnect a Cable

ATTENTION: Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of placing and configuring the drive you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive's input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with placing and configuring the drive.

Understanding the Interface to the Drive

The DriveLogix controller supports a direct connection to the drive consisting of 21 inputs and 21 outputs. The tag names and data types associated with the inputs and outputs are determined by the communication format selection. Currently, the following five communications formats are available:

- Speed Control – for typical speed regulated applications
- Position Control – for typical positioning applications
- Motion Control - for use with Logix motion commands
- User-Defined 1 – for general use as required.
- User-Defined 2 - for general use as required.

Each communication format contains a number of pre-defined tags and user-defined tags.

The pre-defined tag names and data types correspond to the associated parameters within the drive necessary to support the selected communications format. Links must be established in the drive to support the pre-defined tags and are configured using DriveExecutive software. Linking is a mechanism within the drive that configures data flow within the drive. The links within the drive to support the pre-defined tags are protected and must be present. If the associated links are not present, or are deleted, the communication connection between the controller and drive will be broken.

The user-defined tags are made up of a fixed number of REAL (floating point) and DINT (double integer) data types. No links are required within the drive to support these tags. Therefore, links may be created and deleted as required with no affect on the communication connection between the controller and the drive. The user-defined tags may be used to address application specific data needs not covered by the pre-defined tags.

Mapping for Inputs and Outputs

For each of the 21 inputs and 21 outputs, there a dedicated parameter within the drive for a total of 42 parameters. Selecting a communication format defines the data types for each input and output. It also determines the data type for the dedicated parameter in the drive. The selection also configures parameters 601 [From DL DataType] and 625 [To DL DataType], which indicate the data types for each dedicated parameter within the drive.

Determining When the Controller Updates the Drive

The DriveLogix controller follows a producer/consumer model for the drive connection, similar to the interface to an I/O module. The drive acts as both an input module, producing data for the controller; and an output module, consuming data from the controller. Although the producer/consumer model multi casts data, all data in the drive connection is exclusive to the DriveLogix controller.

The controller updates the input and output data in the drive connection asynchronously to the logic scan, consistent with the way it handles other I/O data. All input data from the drive is read in a single block and all output data is written to the drive in a single block.

You must configure the Requested Packet Interval (RPI) rate for the drive. This setting affects how fast the controller reads and writes the data in the drive interface.

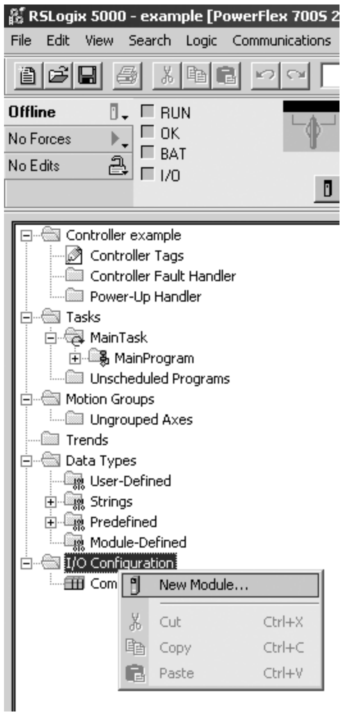
▶ **TIP:** If you want data to remain constant throughout one scan, make a copy of the data at the beginning of the scan and use the copy throughout the scan.

The Drive consumes data from the DriveLogix controller every 2 milliseconds, and produces data to the controller every 2 milliseconds. The drive updates the inputs and outputs to the controller asynchronous to both the program scan and I/O scan of the controller.

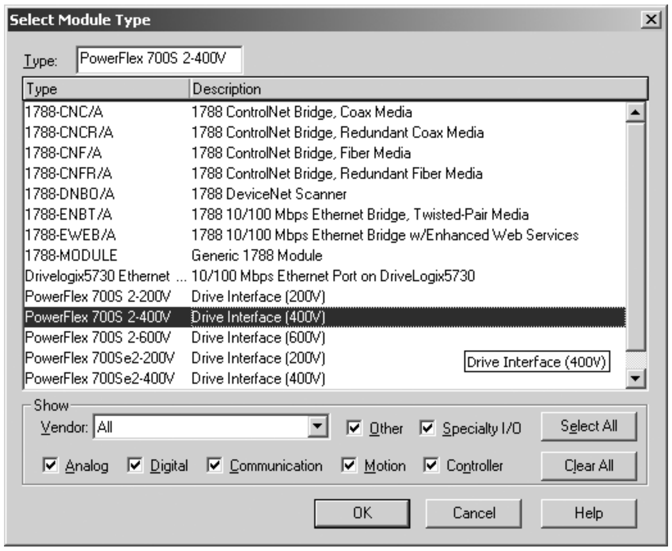
Placing and Configuring the Drive

When you create a project for the DriveLogix controller in RSLogix 5000, the Controller Organizer automatically displays the local DIN rail for Flex I/O. You must add the PowerFlex 700S drive to the configuration, in a manner similar to adding an I/O module. The Controller Organizer automatically places the drive in slot two.

1. In the Controller Organizer, select the I/O Configuration folder. Right-click the selected folder and select New Module.



2. Select the drive (PowerFlex 700S 2-400V in this example).


Select Module Type

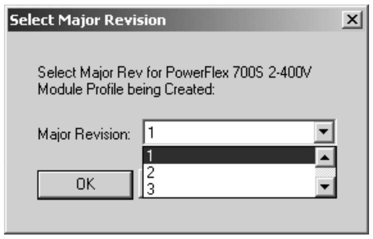
Type: PowerFlex 700S 2-400V

Type	Description
1788-CNC/A	1788 ControlNet Bridge, Coax Media
1788-CNCR/A	1788 ControlNet Bridge, Redundant Coax Media
1788-CNF/A	1788 ControlNet Bridge, Fiber Media
1788-CNFR/A	1788 ControlNet Bridge, Redundant Fiber Media
1788-DNBO/A	1788 DeviceNet Scanner
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1788-MODULE	Generic 1788 Module
Drivelogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730
PowerFlex 700S 2-200V	Drive Interface (200V)
PowerFlex 700S 2-400V	Drive Interface (400V)
PowerFlex 700S 2-600V	Drive Interface (600V)
PowerFlex 700Se2-200V	Drive Interface (200V)
PowerFlex 700Se2-400V	Drive Interface (400V)

Show
Vendor: All
[✓] Other [✓] Specialty I/O
[✓] Analog [✓] Digital [✓] Communication [✓] Motion [✓] Controller

Important: You must select the correct voltage rating for the drive, when adding the drive. You can find this on the drive data nameplate.

3. Select the Major Revision.

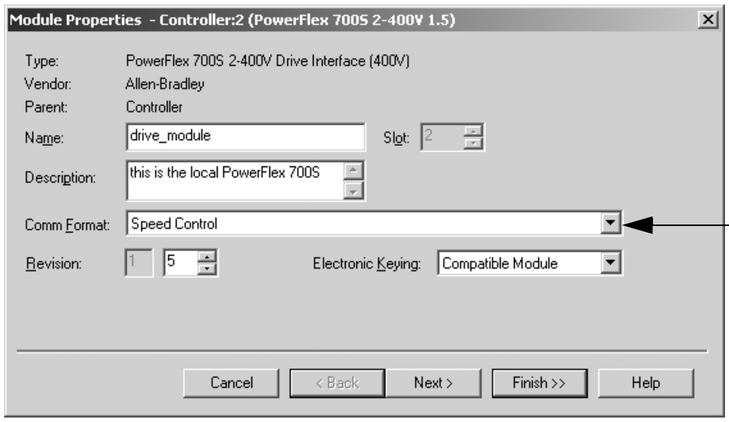


Select Major Rev for PowerFlex 700S 2-400V Module Profile being Created:

Major Revision: 1

4. Configure the drive. Use the module properties wizard to specify characteristics for the module. Click Next to continue through the wizard.

5. Name the drive and specify the Comm Format. Click finish when you are done. The completed module appears in the Controller Organizer.

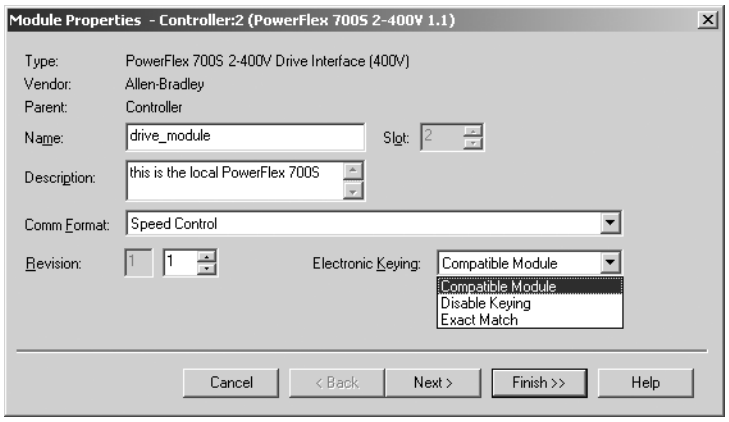

Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)
Cancel
< Back
Next >
Finish >>
Help

Type: PowerFlex 700S 2-400V Drive Interface (400V)
Vendor: Allen-Bradley
Parent: Controller
Name: drive_module
Slot: 2
Description: this is the local PowerFlex 700S
Comm Format: Speed Control
Revision: 1 5
Electronic Keying: Compatible Module

The selection you make for the Comm Format determines the communication format for the connection to the drive. This determines the tag names and data types. See page 2-5. Once you complete adding a module, you cannot change this selection.

Electronic Keying

Electronic keying has no effect on drive module. However, the default setting (Compatible Module) is recommended.

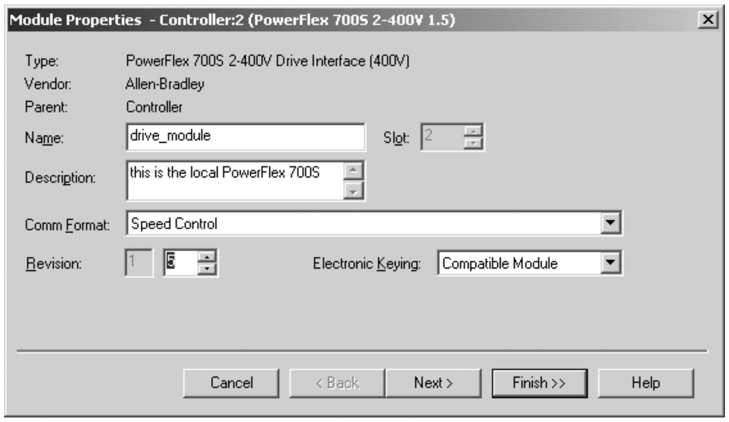


Type: PowerFlex 700S 2-400V Drive Interface (400V)
Vendor: Allen-Bradley
Parent: Controller
Name: drive_module
Slot: 2
Description: this is the local PowerFlex 700S
Comm Format: Speed Control
Revision: 1 1
Electronic Keying: Compatible Module

Selecting “Compatible Module” allows you to enter the drive firmware minor revision.

Revision

You must enter the correct drive VPL firmware revision, in order to launch DriveExecutive and create the appropriate links for the selected communication format. Determine the firmware revision by viewing parameter 314 [VPL Firmware Rev] in the drive.

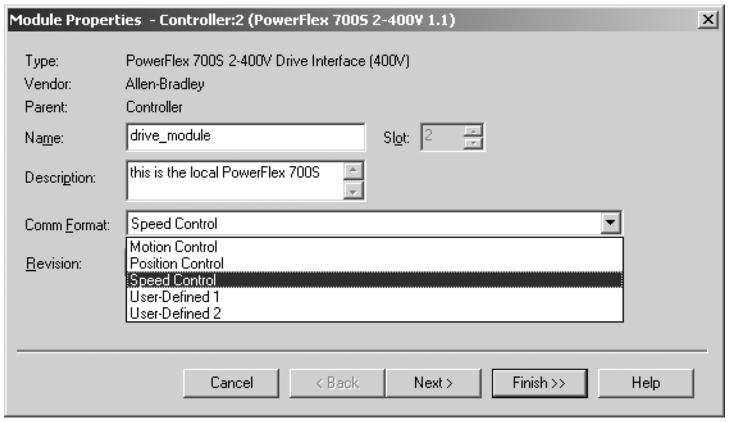

Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

Type: PowerFlex 700S 2-400V Drive Interface (400V)
Vendor: Allen-Bradley
Parent: Controller
Name: drive_module
Slot: 2
Description: this is the local PowerFlex 700S
Comm Format: Speed Control
Revision: 1 5
Electronic Keying: Compatible Module

Communication Formats

The communication format determines the data structure, tag names, and required links for communication to the drive. Each communication format has been structured to meet the requirements of a specific type of application (Speed Control, Position Control, or general purpose), and supports a different data structure. The links within the PowerFlex 700S required to support the selected format are also different. Any of the available communication formats create one direct connection to the drive.

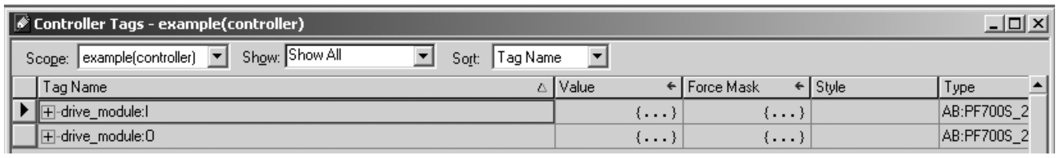
You select the communication format when you configure the drive module.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.1)

Type: PowerFlex 700S 2-400V Drive Interface (400V)
Vendor: Allen-Bradley
Parent: Controller
Name: drive_module
Slot: 2
Description: this is the local PowerFlex 700S
Comm Format: Speed Control
Revision:

The default communication format for the drive is Speed Control. The tags are created as controller-scoped tags. The following tag structure shows the

Speed Control format. The tag structure for this example's drive connection has the tag name of "drive_module".


Controller Tags - example(controller)
Scope: example(controller)
Show: Show All
Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type
drive_module:I	{...}	{...}		AB:PF700S_2
drive_module:O	{...}	{...}		AB:PF700S_2

The following tables show the tag names and their relation ship to parameters in the drive. These examples use a module name of "drive_module".

Table 2.A Mapping for Speed Control Communication Format

Controller Output Tag Element	Drive Parameter	Linked Parameter
LogicCommand	602 [FromDriveLogix00]	151 [Logic Command]
SpeedRef1	603 [FromDriveLogix01]	10 [Speed Ref 1]
TorqueRef1	604 [FromDriveLogix02]	111 [Torque Ref 1]
SpdTorqModeSelect	605 [FromDriveLogix03]	110 [Spd/Torq ModeSel]
TorqueStep	606 [FromDriveLogix04]	116 [Torque Step]
SpdRegDroop	607 [FromDriveLogix05]	86 [Spd Reg Droop]
UserDefinedRealData[0]	608 [FromDriveLogix06]	User Defined
UserDefinedRealData[1]	609 [FromDriveLogix07]	User Defined
UserDefinedRealData[2]	610 [FromDriveLogix08]	User Defined
UserDefinedRealData[3]	611 [FromDriveLogix09]	User Defined
UserDefinedRealData[4]	612 [FromDriveLogix10]	User Defined
UserDefinedRealData[5]	613 [FromDriveLogix11]	User Defined
UserDefinedRealData[6]	614 [FromDriveLogix12]	User Defined
UserDefinedRealData[7]	615 [FromDriveLogix13]	User Defined
UserDefinedRealData[8]	616 [FromDriveLogix14]	User Defined
UserDefinedRealData[9]	617 [FromDriveLogix15]	User Defined
UserDefinedIntegerData[0]	618 [FromDriveLogix16]	User Defined
UserDefinedIntegerData[1]	619 [FromDriveLogix17]	User Defined
UserDefinedIntegerData[2]	620 [FromDriveLogix18]	User Defined
UserDefinedIntegerData[3]	621 [FromDriveLogix19]	User Defined
UserDefinedIntegerData[4]	622 [FromDriveLogix20]	User Defined

Controller Input Tag Element	Drive Parameter	Linked Parameter
LogicStatus	626 [To DriveLogix00]	155 [Logic Status]
FilteredSpdFdbk	627 [To DriveLogix01]	71 [Filtered SpdFdbk]
MotorTorqueRef	628 [To DriveLogix02]	303 [Motor Torque Ref]
OutputCurrent	629 [To DriveLogix03]	308 [Output Current]
MCStatus	630 [To DriveLogix04]	555 [MC Status]
LocalIOStatus	631 [To DriveLogix05]	824 [Local I/O Status]
UserDefinedRealData[0]	632 [To DriveLogix06]	User Defined
UserDefinedRealData[1]	633 [To DriveLogix07]	User Defined
UserDefinedRealData[2]	634 [To DriveLogix08]	User Defined
UserDefinedRealData[3]	635 [To DriveLogix09]	User Defined
UserDefinedRealData[4]	636 [To DriveLogix10]	User Defined
UserDefinedRealData[5]	637 [To DriveLogix11]	User Defined
UserDefinedRealData[6]	638 [To DriveLogix12]	User Defined
UserDefinedRealData[7]	639 [To DriveLogix13]	User Defined
UserDefinedRealData[8]	640 [To DriveLogix14]	User Defined
UserDefinedIntegerData[0]	641 [To DriveLogix15]	User Defined
UserDefinedIntegerData[1]	642 [To DriveLogix16]	User Defined
UserDefinedIntegerData[2]	643 [To DriveLogix17]	User Defined
UserDefinedIntegerData[3]	644 [To DriveLogix18]	User Defined
UserDefinedIntegerData[4]	645 [To DriveLogix19]	User Defined
UserDefinedIntegerData[5]	646 [To DriveLogix20]	User Defined

Table 2.B Mapping for Position Control Communication Format

Controller Output Tag Element	Drive Parameter	Linked Parameter
LogicCommand	602 [FromDriveLogix00]	151 [Logic Command]
SpeedRef1	603 [FromDriveLogix01]	10 [Speed Ref 1]
PositionControl	604 [FromDriveLogix02]	740 [Position Control]
CoarsePositTrgt	605 [FromDriveLogix03]	748 [CoarsePosit Trgt]
PtPtPositRef	606 [FromDriveLogix04]	758 [Pt-Pt Posit Ref]
PositOffset1	607 [FromDriveLogix05]	753 [Posit Offset 1]
UserDefinedRealData[0]	608 [FromDriveLogix06]	User Defined
UserDefinedRealData[1]	609 [FromDriveLogix07]	User Defined
UserDefinedRealData[2]	610 [FromDriveLogix08]	User Defined
UserDefinedRealData[3]	611 [FromDriveLogix09]	User Defined
UserDefinedRealData[4]	612 [FromDriveLogix10]	User Defined
UserDefinedRealData[5]	613 [FromDriveLogix11]	User Defined
UserDefinedRealData[6]	614 [FromDriveLogix12]	User Defined
UserDefinedRealData[7]	615 [FromDriveLogix13]	User Defined
UserDefinedRealData[8]	616 [FromDriveLogix14]	User Defined
UserDefinedIntegerData[0]	617 [FromDriveLogix15]	User Defined
UserDefinedIntegerData[1]	618 [FromDriveLogix16]	User Defined
UserDefinedIntegerData[2]	619 [FromDriveLogix17]	User Defined
UserDefinedIntegerData[3]	620 [FromDriveLogix18]	User Defined
UserDefinedIntegerData[4]	621 [FromDriveLogix19]	User Defined
UserDefinedIntegerData[5]	622 [FromDriveLogix20]	User Defined

Controller Input Tag Element	Drive Parameter	Linked Parameter
LogicStatus	626 [To DriveLogix00]	155 [Logic Status]
FilteredSpdFdbk	627 [To DriveLogix01]	71 [Filtered SpdFdbk]
OutputCurrent	628 [To DriveLogix02]	308 [Output Current]
LocalIOStatus	629 [To DriveLogix03]	824 [Local I/O Status]
PositionStatus	630 [To DriveLogix04]	741 [Position Status]
PositionFdbk	631 [To DriveLogix05]	762 [Position Fdbk]
PositionActual	632 [To DriveLogix06]	763 [Position Actual]
PositionError	633 [To DriveLogix07]	769 [Position Error]
UserDefinedRealData[0]	634 [To DriveLogix08]	User Defined
UserDefinedRealData[1]	635 [To DriveLogix09]	User Defined
UserDefinedRealData[2]	636 [To DriveLogix10]	User Defined
UserDefinedRealData[3]	637 [To DriveLogix11]	User Defined
UserDefinedRealData[4]	638 [To DriveLogix12]	User Defined
UserDefinedRealData[5]	639 [To DriveLogix13]	User Defined
UserDefinedIntegerData[0]	640 [To DriveLogix14]	User Defined
UserDefinedIntegerData[1]	641 [To DriveLogix15]	User Defined
UserDefinedIntegerData[2]	642 [To DriveLogix16]	User Defined
UserDefinedIntegerData[3]	643 [To DriveLogix17]	User Defined
UserDefinedIntegerData[4]	644 [To DriveLogix18]	User Defined
UserDefinedIntegerData[5]	645 [To DriveLogix19]	User Defined
UserDefinedIntegerData[6]	646 [To DriveLogix20]	User Defined

Table 2.C Mapping for Motion Control Communication Format

Controller Output Tag Element	Drive Parameter	Linked Parameter
UserDefinedRealData[0]	602 [FromDriveLogix00]	User Defined
UserDefinedRealData[1]	603 [FromDriveLogix01]	User Defined
UserDefinedRealData[2]	604 [FromDriveLogix02]	User Defined
UserDefinedRealData[3]	605 [FromDriveLogix03]	User Defined
UserDefinedRealData[4]	606 [FromDriveLogix04]	User Defined
UserDefinedRealData[5]	607 [FromDriveLogix05]	User Defined
UserDefinedRealData[6]	608 [FromDriveLogix06]	User Defined
UserDefinedRealData[7]	609 [FromDriveLogix07]	User Defined
UserDefinedRealData[8]	610 [FromDriveLogix08]	User Defined
UserDefinedRealData[9]	611 [FromDriveLogix09]	User Defined
UserDefinedRealData[10]	612 [FromDriveLogix10]	User Defined
UserDefinedRealData[11]	613 [FromDriveLogix11]	User Defined
UserDefinedIntegerData[0]	614 [FromDriveLogix12]	User Defined
UserDefinedIntegerData[1]	615 [FromDriveLogix13]	User Defined
UserDefinedIntegerData[2]	616 [FromDriveLogix14]	User Defined
UserDefinedIntegerData[3]	617 [FromDriveLogix15]	User Defined
UserDefinedIntegerData[4]	618 [FromDriveLogix16]	User Defined
UserDefinedIntegerData[5]	619 [FromDriveLogix17]	User Defined
UserDefinedIntegerData[6]	620 [FromDriveLogix18]	User Defined
UserDefinedIntegerData[7]	621 [FromDriveLogix19]	User Defined
UserDefinedIntegerData[8]	622 [FromDriveLogix20]	User Defined

Controller Input Tag Element	Drive Parameter	Linked Parameter
LogicStatus	626 [To DriveLogix00]	155 [Logic Status]
UserDefinedRealData[0]	627 [To DriveLogix01]	User Defined
UserDefinedRealData[1]	628 [To DriveLogix02]	User Defined
UserDefinedRealData[2]	629 [To DriveLogix03]	User Defined
UserDefinedRealData[3]	630 [To DriveLogix04]	User Defined
UserDefinedRealData[4]	631 [To DriveLogix05]	User Defined
UserDefinedRealData[5]	632 [To DriveLogix06]	User Defined
UserDefinedRealData[6]	633 [To DriveLogix07]	User Defined
UserDefinedRealData[7]	634 [To DriveLogix08]	User Defined
UserDefinedRealData[8]	635 [To DriveLogix09]	User Defined
UserDefinedRealData[9]	636 [To DriveLogix10]	User Defined
UserDefinedRealData[10]	637 [To DriveLogix11]	User Defined
UserDefinedRealData[11]	638 [To DriveLogix12]	User Defined
UserDefinedIntegerData[0]	639 [To DriveLogix13]	User Defined
UserDefinedIntegerData[1]	640 [To DriveLogix14]	User Defined
UserDefinedIntegerData[2]	641 [To DriveLogix15]	User Defined
UserDefinedIntegerData[3]	642 [To DriveLogix16]	User Defined
UserDefinedIntegerData[4]	643 [To DriveLogix17]	User Defined
UserDefinedIntegerData[5]	644 [To DriveLogix18]	User Defined
UserDefinedIntegerData[6]	645 [To DriveLogix19]	User Defined
UserDefinedIntegerData[7]	646 [To DriveLogix20]	User Defined

Table 2.D Mapping for User-Defined 1 Communication Format

Controller Output Tag Element	Drive Parameter	Linked Parameter
LogicCommand	602 [FromDriveLogix00]	151 [Logic Command]
UserDefinedRealData[0]	603 [FromDriveLogix01]	User Defined
UserDefinedRealData[1]	604 [FromDriveLogix02]	User Defined
UserDefinedRealData[2]	605 [FromDriveLogix03]	User Defined
UserDefinedRealData[3]	606 [FromDriveLogix04]	User Defined
UserDefinedRealData[4]	607 [FromDriveLogix05]	User Defined
UserDefinedRealData[5]	608 [FromDriveLogix06]	User Defined
UserDefinedRealData[6]	609 [FromDriveLogix07]	User Defined
UserDefinedRealData[7]	610 [FromDriveLogix08]	User Defined
UserDefinedRealData[8]	611 [FromDriveLogix09]	User Defined
UserDefinedRealData[9]	612 [FromDriveLogix10]	User Defined
UserDefinedRealData[10]	613 [FromDriveLogix11]	User Defined
UserDefinedRealData[11]	614 [FromDriveLogix12]	User Defined
UserDefinedRealData[12]	615 [FromDriveLogix13]	User Defined
UserDefinedRealData[13]	616 [FromDriveLogix14]	User Defined
UserDefinedIntegerData[0]	617 [FromDriveLogix15]	User Defined
UserDefinedIntegerData[1]	618 [FromDriveLogix16]	User Defined
UserDefinedIntegerData[2]	619 [FromDriveLogix17]	User Defined
UserDefinedIntegerData[3]	620 [FromDriveLogix18]	User Defined
UserDefinedIntegerData[4]	621 [FromDriveLogix19]	User Defined
UserDefinedIntegerData[5]	622 [FromDriveLogix20]	User Defined

Controller Input Tag Element	Drive Parameter	Linked Parameter
LogicStatus	626 [To DriveLogix00]	155 [Logic Status]
UserDefinedRealData[0]	627 [To DriveLogix01]	User Defined
UserDefinedRealData[1]	628 [To DriveLogix02]	User Defined
UserDefinedRealData[2]	629 [To DriveLogix03]	User Defined
UserDefinedRealData[3]	630 [To DriveLogix04]	User Defined
UserDefinedRealData[4]	631 [To DriveLogix05]	User Defined
UserDefinedRealData[5]	632 [To DriveLogix06]	User Defined
UserDefinedRealData[6]	633 [To DriveLogix07]	User Defined
UserDefinedRealData[7]	634 [To DriveLogix08]	User Defined
UserDefinedRealData[8]	635 [To DriveLogix09]	User Defined
UserDefinedRealData[9]	636 [To DriveLogix10]	User Defined
UserDefinedRealData[10]	637 [To DriveLogix11]	User Defined
UserDefinedRealData[11]	638 [To DriveLogix12]	User Defined
UserDefinedIntegerData[0]	639 [To DriveLogix13]	User Defined
UserDefinedIntegerData[1]	640 [To DriveLogix14]	User Defined
UserDefinedIntegerData[2]	641 [To DriveLogix15]	User Defined
UserDefinedIntegerData[3]	642 [To DriveLogix16]	User Defined
UserDefinedIntegerData[4]	643 [To DriveLogix17]	User Defined
UserDefinedIntegerData[5]	644 [To DriveLogix18]	User Defined
UserDefinedIntegerData[6]	645 [To DriveLogix19]	User Defined
UserDefinedIntegerData[7]	646 [To DriveLogix20]	User Defined

Table 2.E Mapping for User-Defined 2 Communication Format

Controller Output Tag Element	Drive Parameter	Linked Parameter
LogicCommand	602 [FromDriveLogix00]	151 [Logic Command]
UserDefinedRealData[0]	603 [FromDriveLogix01]	User Defined
UserDefinedRealData[1]	604 [FromDriveLogix02]	User Defined
UserDefinedRealData[2]	605 [FromDriveLogix03]	User Defined
UserDefinedRealData[3]	606 [FromDriveLogix04]	User Defined
UserDefinedRealData[4]	607 [FromDriveLogix05]	User Defined
UserDefinedRealData[5]	608 [FromDriveLogix06]	User Defined
UserDefinedRealData[6]	609 [FromDriveLogix07]	User Defined
UserDefinedRealData[7]	610 [FromDriveLogix08]	User Defined
UserDefinedRealData[8]	611 [FromDriveLogix09]	User Defined
UserDefinedRealData[9]	612 [FromDriveLogix10]	User Defined
UserDefinedRealData[10]	613 [FromDriveLogix11]	User Defined
UserDefinedIntegerData[0]	614 [FromDriveLogix12]	User Defined
UserDefinedIntegerData[1]	615 [FromDriveLogix13]	User Defined
UserDefinedIntegerData[2]	616 [FromDriveLogix14]	User Defined
UserDefinedIntegerData[3]	617 [FromDriveLogix15]	User Defined
UserDefinedIntegerData[4]	618 [FromDriveLogix16]	User Defined
UserDefinedIntegerData[5]	619 [FromDriveLogix17]	User Defined
UserDefinedIntegerData[6]	620 [FromDriveLogix18]	User Defined
UserDefinedIntegerData[7]	621 [FromDriveLogix19]	User Defined
UserDefinedIntegerData[8]	622 [FromDriveLogix20]	User Defined

Controller Input Tag Element	Drive Parameter	Linked Parameter
LogicStatus	626 [To DriveLogix00]	155 [Logic Status]
UserDefinedRealData[0]	627 [To DriveLogix01]	User Defined
UserDefinedRealData[1]	628 [To DriveLogix02]	User Defined
UserDefinedRealData[2]	629 [To DriveLogix03]	User Defined
UserDefinedRealData[3]	630 [To DriveLogix04]	User Defined
UserDefinedRealData[4]	631 [To DriveLogix05]	User Defined
UserDefinedRealData[5]	632 [To DriveLogix06]	User Defined
UserDefinedRealData[6]	633 [To DriveLogix07]	User Defined
UserDefinedRealData[7]	634 [To DriveLogix08]	User Defined
UserDefinedRealData[8]	635 [To DriveLogix09]	User Defined
UserDefinedRealData[9]	636 [To DriveLogix10]	User Defined
UserDefinedIntegerData[0]	637 [To DriveLogix11]	User Defined
UserDefinedIntegerData[1]	638 [To DriveLogix12]	User Defined
UserDefinedIntegerData[2]	639 [To DriveLogix13]	User Defined
UserDefinedIntegerData[3]	640 [To DriveLogix14]	User Defined
UserDefinedIntegerData[4]	641 [To DriveLogix15]	User Defined
UserDefinedIntegerData[5]	642 [To DriveLogix16]	User Defined
UserDefinedIntegerData[6]	643 [To DriveLogix17]	User Defined
UserDefinedIntegerData[7]	644 [To DriveLogix18]	User Defined
UserDefinedIntegerData[8]	645 [To DriveLogix19]	User Defined
UserDefinedIntegerData[9]	646 [To DriveLogix20]	User Defined

For each of the communication formats, drive_module:O.LogicCommand and drive_module:I.LogicStatus are provided as DINT data types. In addition to these tags, the control bits for each are also available as Boolean values with tag names that correspond to the control bits in the drive. This gives you the option of programming the Logic Command and Status words at the Boolean level or as an integer value.

Not all 32-bits within parameter 151 [Logic Command], are directly visible in the PowerFlex 700S. To view all 32-bits, refer to parameter 152 [Applied LogicCmd].

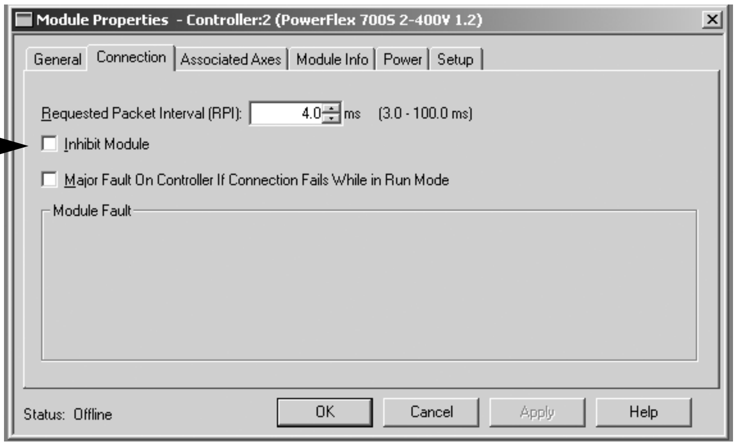
Inhibiting the Drive Connection

RSLogix 5000 programming software allows you to inhibit the controller's connection to the drive, in the same way you inhibit its connection to an I/O module. Inhibiting the drive module shuts down the connection from the controller to the drive. When you create the module you can choose to inhibit it. After you have created the module you can inhibit or un-inhibit it by manipulating its properties window.

ATTENTION: Inhibiting a drive module breaks the controller's connection to the drive. In this situation, the controller can neither, start/stop the drive nor read the status of the drive. The drive can continue to operate based on its parameter settings and inputs. To avoid potential personal injury and damage to machinery, make sure this does not create unsafe operation.

On the Connection tab during creation or in the Properties window

Check the inhibit box to inhibit the connection to the drive

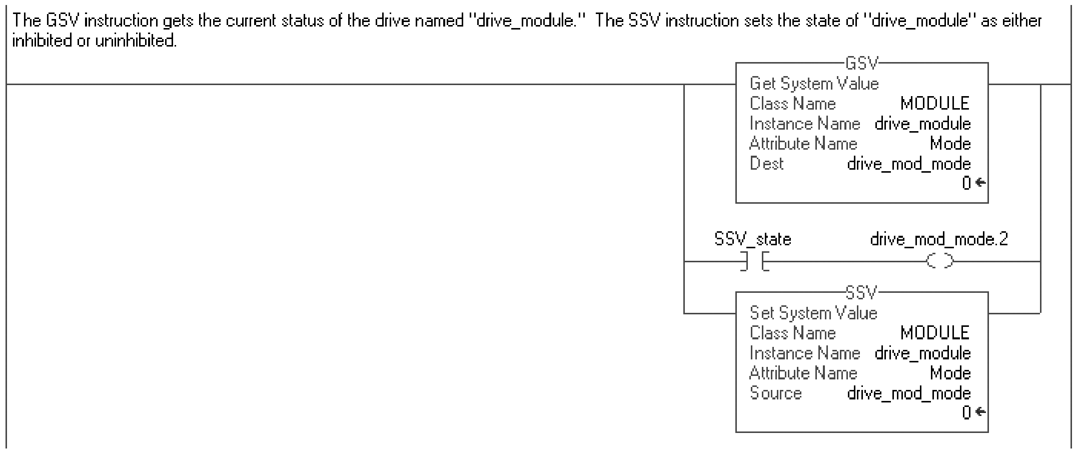


Requested Packet Interval (RPI): 4.0 ms (3.0 - 100.0 ms)
☐ Inhibit Module
☐ Major Fault On Controller If Connection Fails While in Run Mode
Status: Offline

When you inhibit the drive module, the Controller Organizer displays a yellow attention symbol ⚠ over the module.

If you are:	Inhibit the drive module to:
offline	put a place holder for the drive module to indicate that configuration is not yet complete. The inhibit status is stored in the project. When you download the project, the module is still inhibited.
online	stop communication to the drive. If you inhibit the drive while you are connected to the module, the connection to the module is closed. By default, the PowerFlex 700S drive will fault. The data inputs to the drive will either hold last state, or reset to zero data based on the setting of parameter 385 [Lgx Comm Loss Data]. If you inhibit the drive but a connection to the module was not established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is inhibited and not faulted. If you uninhibit the drive (clear the check box), and no fault condition occurs, a connection is made to the drive.

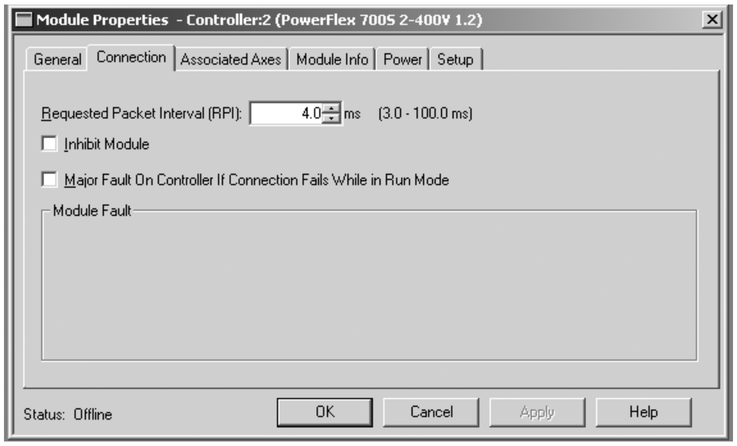
To inhibit a module from logic, you must first read the Mode attribute for the module using a GSV instruction. Set bit 2 to the inhibit status (1 to inhibit or 0 to uninhibit). Use a SSV instruction to write the Mode attribute back to the module. For example:


The GSV instruction gets the current status of the drive named "drive_module." The SSV instruction sets the state of "drive_module" as either inhibited or uninhibited.

Using DriveExecutive Lite

In order to launch DriveExecutive Lite from within RSLogix 5000, the drives power rating must be selected. The drive firmware revision must be applied prior to selecting the power rating.

1. If not already done, enter the drive firmware revision. Click the Finish button to apply the revision data.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.2)

Requested Packet Interval (RPI): 4.0 ms (3.0 - 100.0 ms)

☐ Inhibit Module

☐ Major Fault On Controller If Connection Fails While in Run Mode

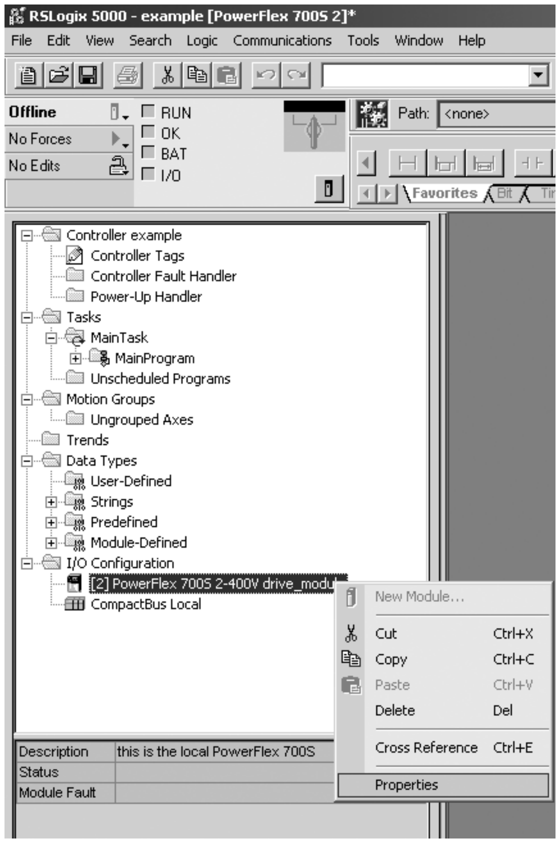
Module Fault

Status: Offline

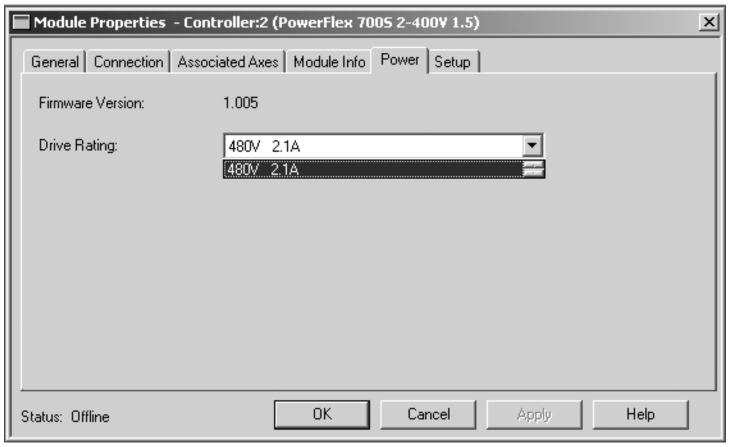
2. In the Controller Organizer, select the PowerFlex 700S drive. Right-click the drive module and select Properties.

3. Select the Power tab.

4. Select the correct Drive Rating. This data can be found on the PowerFlex 700S data nameplate.



Description: this is the local PowerFlex 700S
Status:
Module Fault:



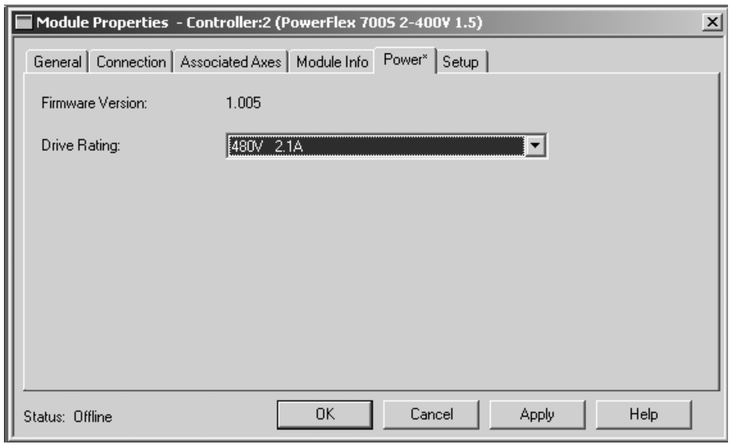
Firmware Version: 1.005

Drive Rating: 480V 2.1A

Status: Offline

TIP: If your drive's power rating does not appear as a selection, you do not have the DriveExecutive Lite database file for your drive. To create a database file, connect to the drive with DriveExecutive Lite. This will automatically create the database. You can also download the database file from ***http://www.ab.com/drives/data.html***

5. Once the power rating is selected, apply your changes by selecting the Apply button.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)
General | Connection | Associated Axes | Module Info | Power* | Setup
OK | Cancel | Apply | Help

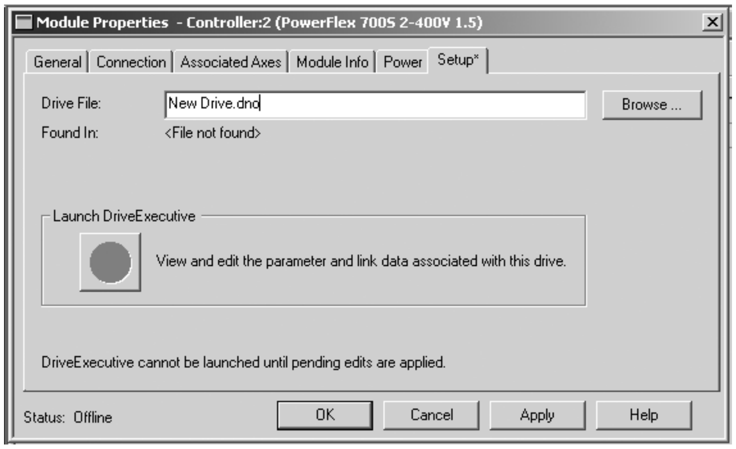
Firmware Version: 1.005

Drive Rating: 480V 2.1A

Status: Offline

6. Select the Setup tab.

7. Enter the file name for your DriveExecutive Lite parameter file, then click the Apply button.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)
General | Connection | Associated Axes | Module Info | Power | Setup*
Browse ...
Launch DriveExecutive
OK | Cancel | Apply | Help

Drive File: New Drive.dno

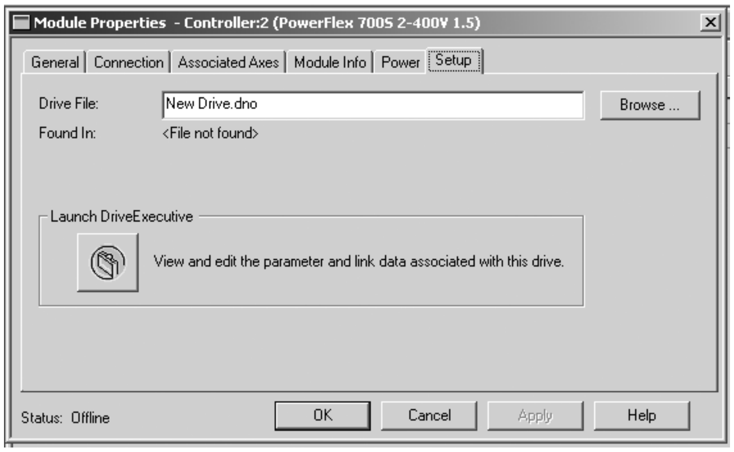
Found In: <File not found>

View and edit the parameter and link data associated with this drive.

DriveExecutive cannot be launched until pending edits are applied.

Status: Offline

8. Click the DriveExecutive button to launch DriveExecutive Lite.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)
General | Connection | Associated Axes | Module Info | Power | Setup
Browse ...
Launch DriveExecutive
OK | Cancel | Apply | Help

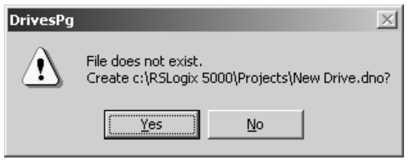
Drive File: New Drive.dno

Found In: <File not found>

View and edit the parameter and link data associated with this drive.

Status: Offline

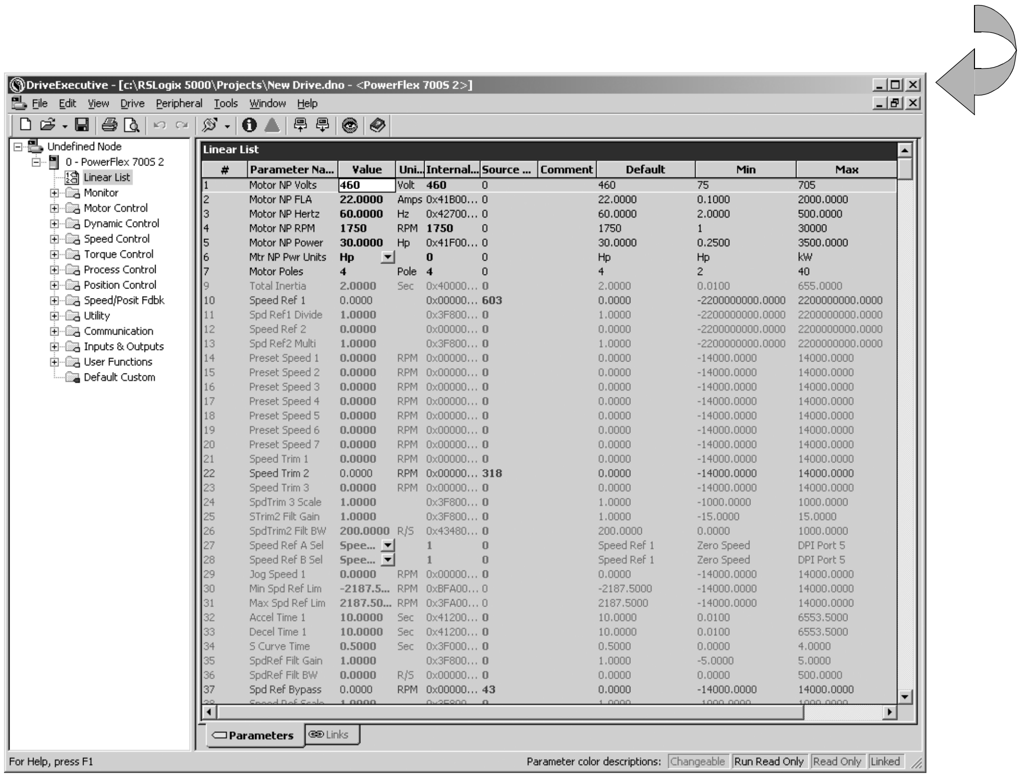
9. When asked to create a new DriveExecutive Lite file, select yes.



DrivesPg

File does not exist.
Create c:\RSLogix 5000\Projects\New Drive.dno?

DriveExecutive will then launch and open the newly created file



DriveExecutive - [c:\RSLogix 5000\Projects\New Drive.dno - <PowerFlex 700S 2>]

Linear List

#	Parameter Na...	Value	Uni...	Internal...	Source ...	Comment	Default	Min	Max
1	Motor NP Volts	460	Volt	460	0		460	75	705
2	Motor NP FLA	22.0000	Amps	0x41B00...	0		22.0000	0.1000	2000.0000
3	Motor NP Hertz	60.0000	Hz	0x42700...	0		60.0000	2.0000	500.0000
4	Motor NP RPM	1750	RPM	1750	0		1750	1	30000
5	Motor NP Power	30.0000	Hp	0x41F00...	0		30.0000	0.2500	3500.0000
6	Mtr NP Pwr Units	Hp		0	0		Hp	Hp	kW
7	Motor Poles	4	Pole	4	0		4	2	40
9	Total Inertia	2.0000	Sec	0x40000...	0		2.0000	0.0100	655.0000
10	Speed Ref 1	0.0000		0x00000...	603		0.0000	-2200000000.0000	2200000000.0000
11	Spd Ref1 Divide	1.0000		0x3F800...	0		1.0000	-2200000000.0000	2200000000.0000
12	Speed Ref 2	0.0000		0x00000...	0		0.0000	-2200000000.0000	2200000000.0000
13	Spd Ref2 Multi	1.0000		0x3F800...	0		1.0000	-2200000000.0000	2200000000.0000
14	Preset Speed 1	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
15	Preset Speed 2	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
16	Preset Speed 3	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
17	Preset Speed 4	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
18	Preset Speed 5	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
19	Preset Speed 6	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
20	Preset Speed 7	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
21	Speed Trim 1	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
22	Speed Trim 2	0.0000	RPM	0x00000...	318		0.0000	-14000.0000	14000.0000
23	Speed Trim 3	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
24	SpdTrim 3 Scale	1.0000		0x3F800...	0		1.0000	-1000.0000	1000.0000
25	STrim2 Filt Gain	1.0000		0x3F800...	0		1.0000	-15.0000	15.0000
26	SpdTrim2 Filt BW	200.0000	R/S	0x43480...	0		200.0000	0.0000	1000.0000
27	Speed Ref A Sel	Spee...		1	0		Speed Ref 1	Zero Speed	DPI Port 5
28	Speed Ref B Sel	Spee...		1	0		Speed Ref 1	Zero Speed	DPI Port 5
29	Jog Speed 1	0.0000	RPM	0x00000...	0		0.0000	-14000.0000	14000.0000
30	Min Spd Ref Lim	-2187.5...	RPM	0xBFA00...	0		-2187.5000	-14000.0000	14000.0000
31	Max Spd Ref Lim	2187.50...	RPM	0x3FA00...	0		2187.5000	-14000.0000	14000.0000
32	Accel Time 1	10.0000	Sec	0x41200...	0		10.0000	0.0100	6553.5000
33	Decel Time 1	10.0000	Sec	0x41200...	0		10.0000	0.0100	6553.5000
34	S Curve Time	0.5000	Sec	0x3F000...	0		0.5000	0.0000	4.0000
35	SpdRef Filt Gain	1.0000		0x3F800...	0		1.0000	-5.0000	5.0000
36	SpdRef Filt BW	0.0000	R/S	0x00000...	0		0.0000	0.0000	500.0000
37	Spd Ref Bypass	0.0000	RPM	0x00000...	43		0.0000	-14000.0000	14000.0000

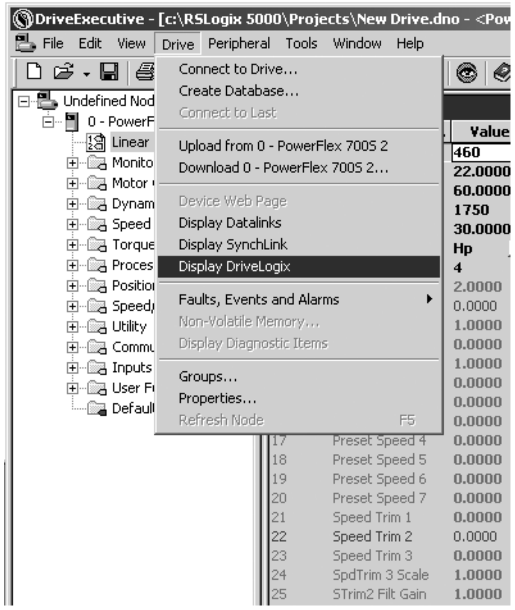
For Help, press F1

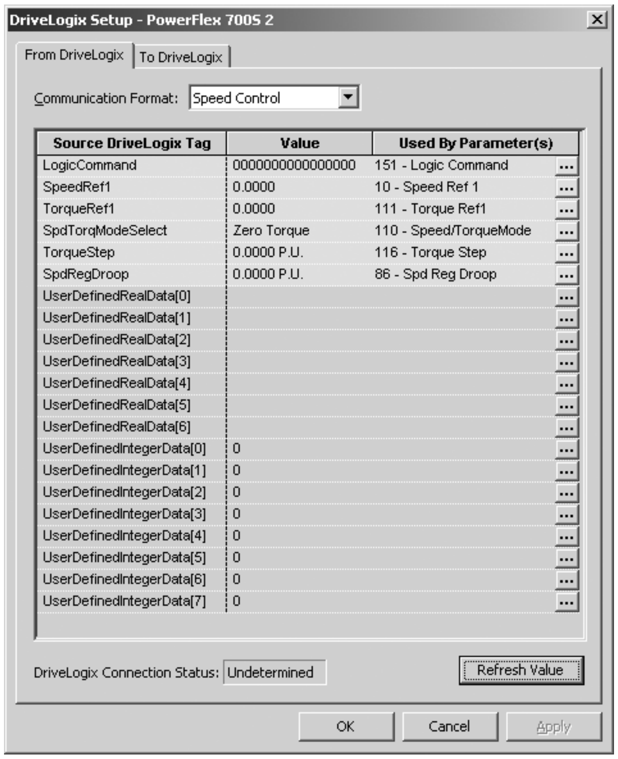
Parameter color descriptions: Changeable | Run Read Only | Read Only | Linked

Viewing the Communication Interface to the Controller

DriveExecutive Lite has a setup screen that details the communication interface between the controller and drive. From this screen, the relationship between drive parameters and controller tags is presented for the selected communication format. You can create additional links within the drive for use with the user-defined tags in the controller.

1. To view the setup screen select Peer Communication from the Drive drop-down menu. Then select the From Controller tab.





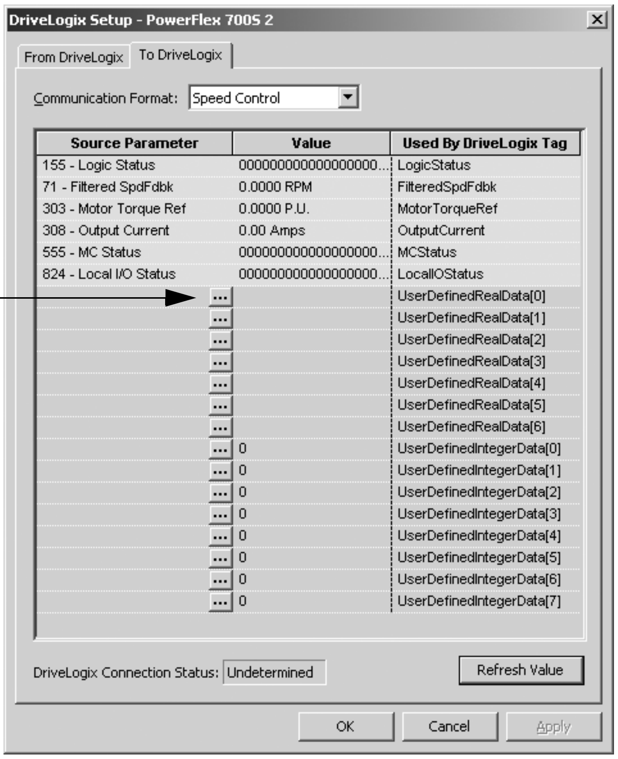
Communication Format: Speed Control

Source DriveLogix Tag	Value	Used By Parameter(s)
LogicCommand	0000000000000000	151 - Logic Command
SpeedRef1	0.0000	10 - Speed Ref 1
TorqueRef1	0.0000	111 - Torque Ref1
SpdTorqModeSelect	Zero Torque	110 - Speed/TorqueMode
TorqueStep	0.0000 P.U.	116 - Torque Step
SpdRegDroop	0.0000 P.U.	86 - Spd Reg Droop
UserDefinedRealData[0]		
UserDefinedRealData[1]		
UserDefinedRealData[2]		
UserDefinedRealData[3]		
UserDefinedRealData[4]		
UserDefinedRealData[5]		
UserDefinedRealData[6]		
UserDefinedIntegerData[0]	0	
UserDefinedIntegerData[1]	0	
UserDefinedIntegerData[2]	0	
UserDefinedIntegerData[3]	0	
UserDefinedIntegerData[4]	0	
UserDefinedIntegerData[5]	0	
UserDefinedIntegerData[6]	0	
UserDefinedIntegerData[7]	0	

DriveLogix Connection Status: Undetermined

2. To send additional data from the drive to the controller, go to the To DriveLogix tab.

Click the button in front of the UserDefinedRealData[0] tag

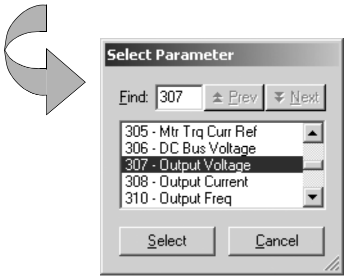


DriveLogix Setup - PowerFlex 700S 2

Communication Format: Speed Control

Source Parameter	Value	Used By DriveLogix Tag
155 - Logic Status	000000000000000000...	LogicStatus
71 - Filtered SpdFdbk	0.0000 RPM	FilteredSpdFdbk
303 - Motor Torque Ref	0.0000 P.U.	MotorTorqueRef
308 - Output Current	0.00 Amps	OutputCurrent
555 - MC Status	000000000000000000...	MCStatus
824 - Local I/O Status	000000000000000000...	LocalIOStatus
...		UserDefinedRealData[0]
...		UserDefinedRealData[1]
...		UserDefinedRealData[2]
...		UserDefinedRealData[3]
...		UserDefinedRealData[4]
...		UserDefinedRealData[5]
...		UserDefinedRealData[6]
...	0	UserDefinedIntegerData[0]
...	0	UserDefinedIntegerData[1]
...	0	UserDefinedIntegerData[2]
...	0	UserDefinedIntegerData[3]
...	0	UserDefinedIntegerData[4]
...	0	UserDefinedIntegerData[5]
...	0	UserDefinedIntegerData[6]
...	0	UserDefinedIntegerData[7]

DriveLogix Connection Status: Undetermined



Select Parameter

Find: 307

305 - Mtr Trq Curr Ref
306 - DC Bus Voltage
307 - Output Voltage
308 - Output Current
310 - Output Freq

3. Select the desired source (parameter 307 [Output Voltage] in this example) in the resulting window.

Configuring the Drive's Response to a Connection Failure or Controller Mode Change

The drive contains several parameters that allow you to configure the drive's response to communication loss to the controller. From the drive's perspective, a communication loss can come in the following two forms.

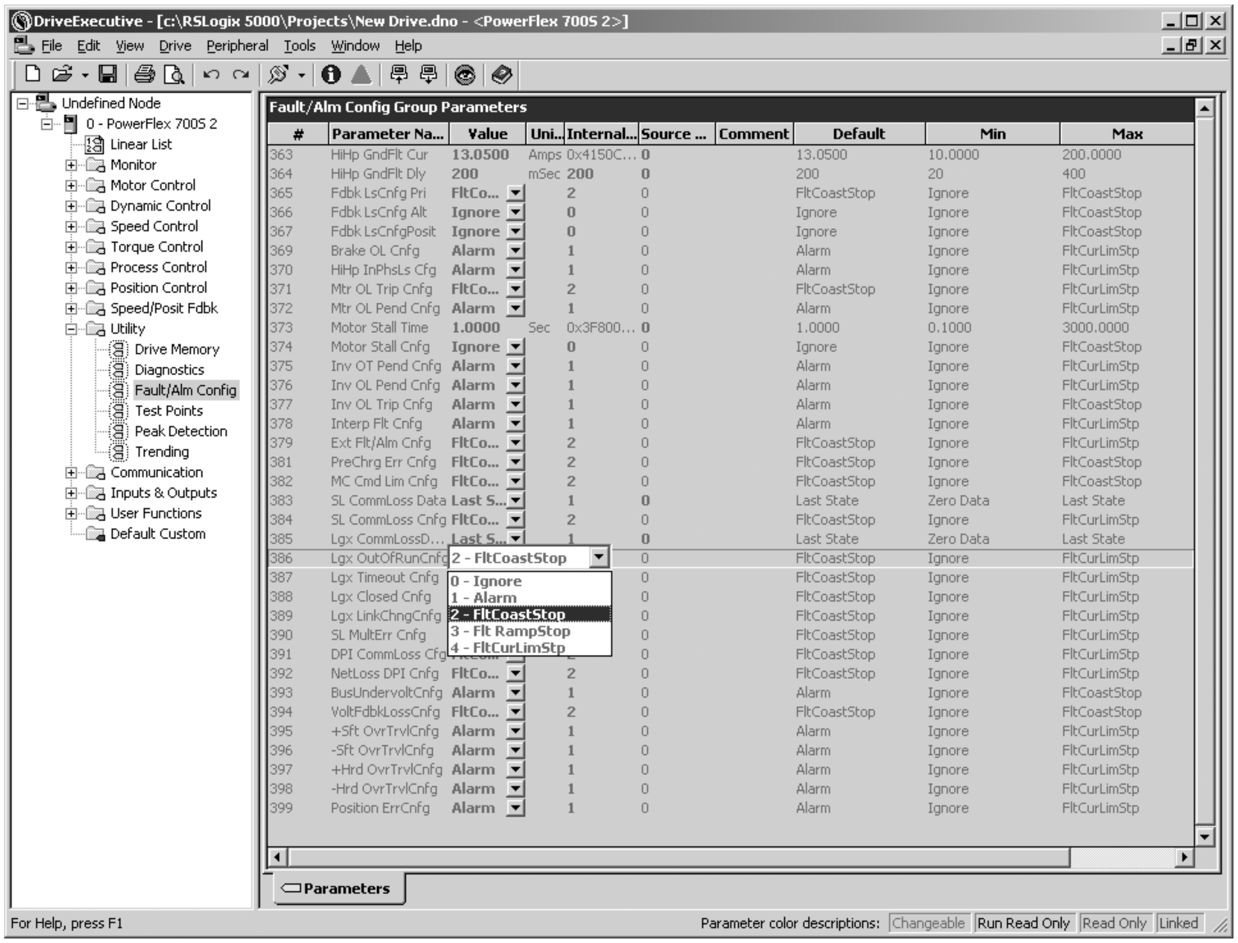
- The controller closes the connection (for example, the connection is inhibited).
- A general failure occurs causing the connection to time out.

Parameter 386 [Lgx OutofRunCnfg] configures the drive's response to the controller is removed from the Run Mode. Parameter 387 [Lgx Timeout

Cnfg] configures the drive’s response to a general connection failure as detected by the drive. Parameter 388 [Lgx Closed Cnfg] configures the drive’s response to the controller closing the connection. All of these parameters configure the drive’s response to these exception events in the following ways: ignore, alarm, fault and coast to stop, fault and ramp to stop, fault and stop in current limit.

Parameter 385 [Lgx CommLossData] determines what the drive does with data from the controller when communication is lost. It determines if the drive resets the data to zero or holds the data in its last state.

Configure these parameters, using DriveExecutive Lite. Locate them in the Fault/Alm Config group of the Utility file.



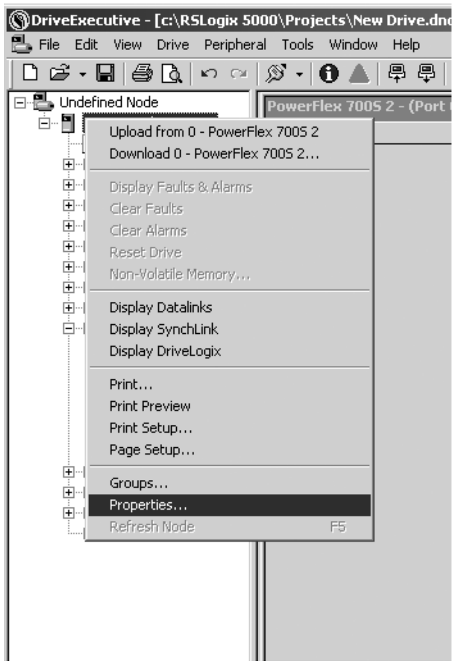
Fault/Alm Config Group Parameters

#	Parameter Na...	Value	Uni...	Internal...	Source ...	Comment	Default	Min	Max
363	HiHp GndFlt Cur	13.0500	Amps	0x4150C...	0		13.0500	10.0000	200.0000
364	HiHp GndFlt Dly	200	mSec	200	0		200	20	400
365	Fdbk LsCnfg Pri	FltCo...		2	0		FltCoastStop	Ignore	FltCoastStop
366	Fdbk LsCnfg Alt	Ignore		0	0		Ignore	Ignore	FltCoastStop
367	Fdbk LsCnfgPosit	Ignore		0	0		Ignore	Ignore	FltCoastStop
369	Brake OL Cnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
370	HiHp InPhsLs Cfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
371	Mtr OL Trip Cnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCurLimStp
372	Mtr OL Pend Cnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
373	Motor Stall Time	1.0000	Sec	0x3F800...	0		1.0000	0.1000	3000.0000
374	Motor Stall Cnfg	Ignore		0	0		Ignore	Ignore	FltCoastStop
375	Inv OT Pend Cnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
376	Inv OL Pend Cnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
377	Inv OL Trip Cnfg	Alarm		1	0		Alarm	Ignore	FltCoastStop
378	Interp Flt Cnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
379	Ext Flt/Alm Cnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCurLimStp
381	PreChrg Err Cnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCoastStop
382	MC Cmd Lim Cnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCoastStop
383	SL CommLoss Data	Last S...		1	0		Last State	Zero Data	Last State
384	SL CommLoss Cnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCurLimStp
385	Lgx CommLossD...	Last S...		1	0		Last State	Zero Data	Last State
386	Lgx OutOfRunCnfg	2 - FltCoastStop			0		FltCoastStop	Ignore	FltCurLimStp
387	Lgx Timeout Cnfg				0		FltCoastStop	Ignore	FltCurLimStp
388	Lgx Closed Cnfg				0		FltCoastStop	Ignore	FltCurLimStp
389	Lgx LinkChngCnfg				0		FltCoastStop	Ignore	FltCurLimStp
390	SL MultErr Cnfg				0		FltCoastStop	Ignore	FltCurLimStp
391	DPI CommLoss Cfg				0		FltCoastStop	Ignore	FltCurLimStp
392	NetLoss DPI Cnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCurLimStp
393	BusUndervoltCnfg	Alarm		1	0		Alarm	Ignore	FltCoastStop
394	VoltFdbkLossCnfg	FltCo...		2	0		FltCoastStop	Ignore	FltCoastStop
395	+Sft OvrTrvlCnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
396	-Sft OvrTrvlCnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
397	+Hrd OvrTrvlCnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
398	-Hrd OvrTrvlCnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp
399	Position ErrCnfg	Alarm		1	0		Alarm	Ignore	FltCurLimStp

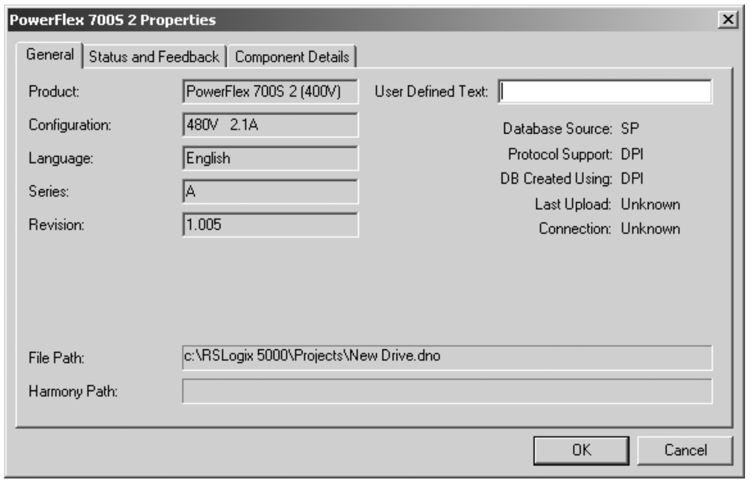
Using Existing DriveExecutive Lite Files

Before using an existing DriveExecutive Lite file, verify the firmware revision, communication format, and power rating in the drive file match the data entered in drive module properties in your DriveLogix application.

1. Select Properties from the Drive menu.



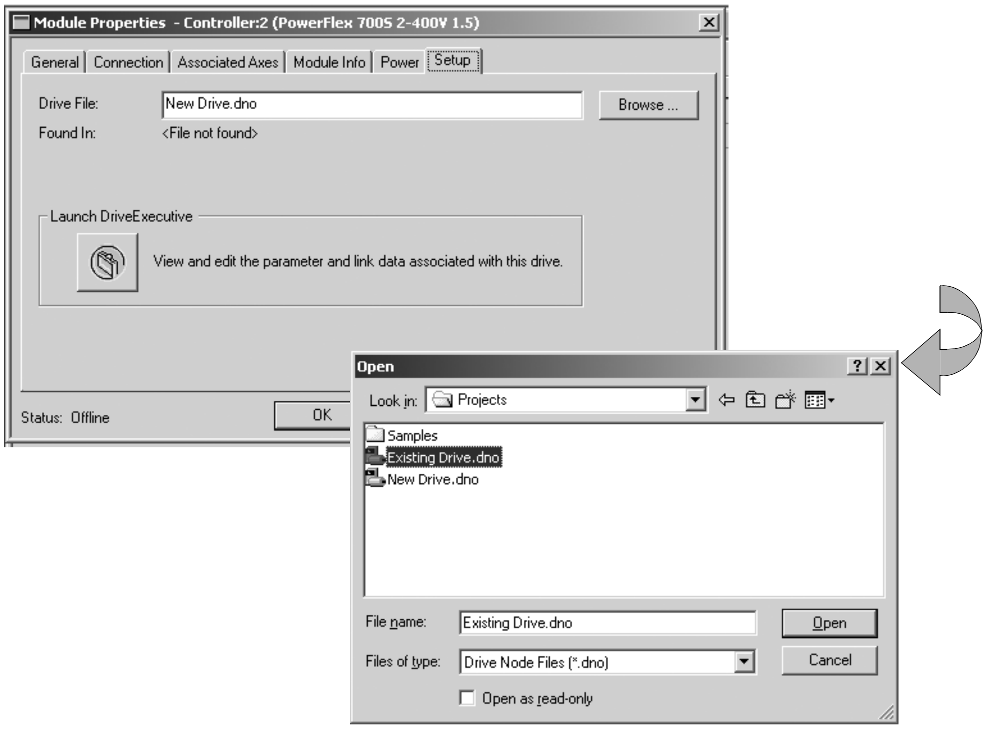
2. View the revision and ratings on the General tab of the Properties window.



Product:	PowerFlex 700S 2 (400V)
Configuration:	480V 2.1A
Language:	English
Series:	A
Revision:	1.005
User Defined Text:	
Database Source:	SP
Protocol Support:	DPI
DB Created Using:	DPI
Last Upload:	Unknown
Connection:	Unknown
File Path:	c:\RSLogix 5000\Projects\New Drive.dno
Harmony Path:	

3. Refer to Viewing the Communication Interface to the Controller on page 2-16, to view the communication format.

4. In RSLogix 5000, go to the Setup tab o the Properties window. Click the Browse button. Select the existing DriveExecutive file (Existing Drive.dno in this example). Click the open button.



Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

General | Connection | Associated Axes | Module Info | Power | Setup

Drive File: New Drive.dno

Found In: <File not found>

Launch DriveExecutive

View and edit the parameter and link data associated with this drive.

Status: Offline

Open

Look in: Projects

Samples

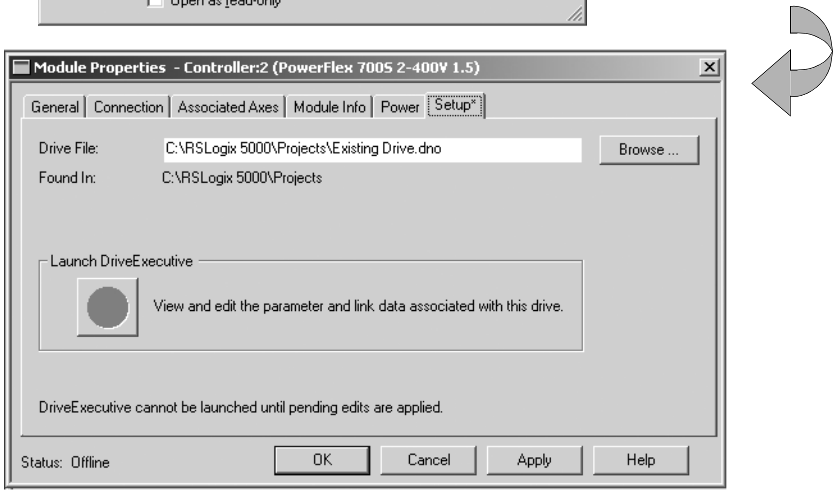
Existing Drive.dno

New Drive.dno

File name: Existing Drive.dno

Files of type: Drive Node Files (*.dno)

Open as read-only



Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

General | Connection | Associated Axes | Module Info | Power | Setup*

Drive File: C:\RSLogix 5000\Projects\Existing Drive.dno

Found In: C:\RSLogix 5000\Projects

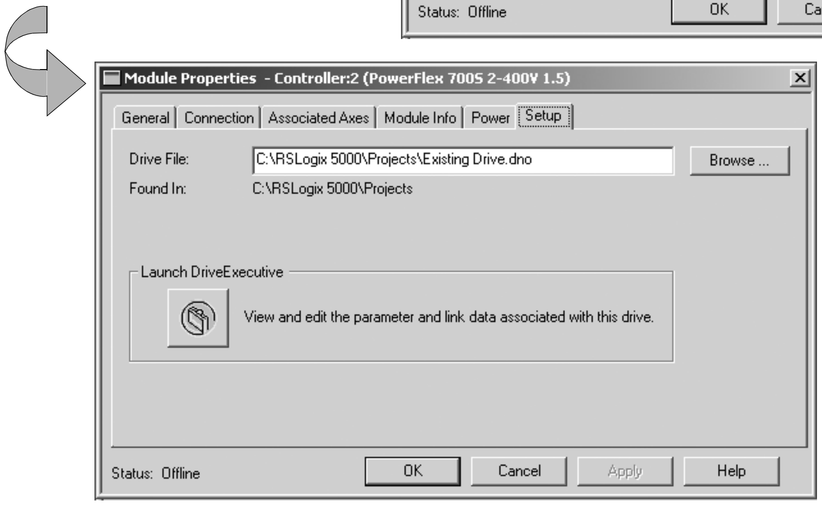
Launch DriveExecutive

View and edit the parameter and link data associated with this drive.

DriveExecutive cannot be launched until pending edits are applied.

Status: Offline

5. Click the Apply button and then launch DriveExecutive.



Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

General | Connection | Associated Axes | Module Info | Power | Setup

Drive File: C:\RSLogix 5000\Projects\Existing Drive.dno

Found In: C:\RSLogix 5000\Projects

Launch DriveExecutive

View and edit the parameter and link data associated with this drive.

Status: Offline

Accessing Drive Data

Drive data is displayed as structures of multiple tags. The names and data structures are based on the selected communication format. The programming software automatically creates the necessary structures and tags when you configure the drive module. Each tag name follows this format:

ModuleName:Type.MemberName.SubMemberName.Bit

where:

This address variable:	Is:
ModuleName	Identifies the module name entered during the drive module configuration

Type	Type of data
	I = input
	O = output

MemberName	Specific data from the drive; depends on the selected communication format. For tags associated with pre-defined data links, this name will be the same as the corresponding parameter name in the drive.
SubMemberName	Specific data related to a MemberName
Bit (optional)	Specific bit of a DINT data value

Refer to Communication Formats on page 2-5 for sample tag names.

Monitoring Drive Data

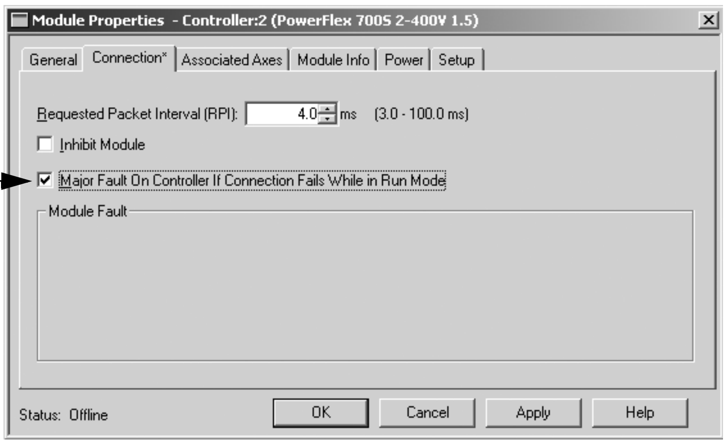
The DriveLogix controller offers different levels at which you can monitor the drive module. You can:

- configure the drive module so that the controller faults if the drive loses its connection to the controller.
- use the programming software to display fault data
- program logic to monitor fault data so you can take appropriate action

Configuring the Controller's Response to a Connection Failure

You can configure the drive module to generate a major fault in the controller if the drive loses its connection to the controller.

Check this box to configure the drive module to generate a major fault if it loses its connection to the controller



Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

Requested Packet Interval (RPI): 4.0 ms (3.0 - 100.0 ms)

☐ Inhibit Module

☑ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

If you do not configure the major fault to occur, you should monitor the drive module status. If the drive loses its connection to the controller:

- Outputs remain in their last
- Inputs remain in their last state
- By default the drive will fault

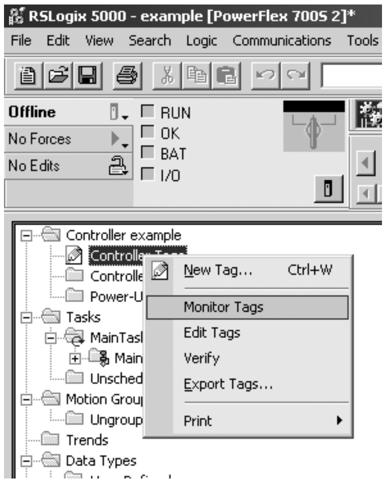
ATTENTION: If a drive loses its connection to the controller, the controller and other I/O modules continue to operate based on old data from the drive, and the drive can continue to operate based on old data from the controller. To avoid potential personal injury and damage to machinery, make sure this does not create unsafe operation.

Configure the drive to generate a controller major fault when the drive loses its connection to the controller. Or, monitor the status of the drive module.

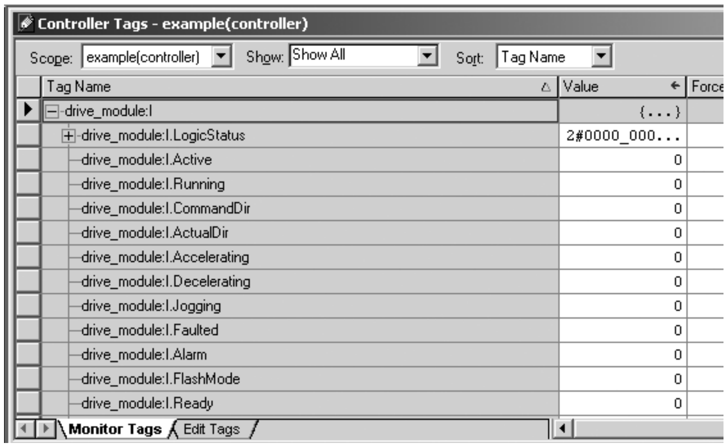
Monitoring the drive module

Each communication format provides a drive status word that will indicate when a drive fault or alarm occurs. To view this data through the programming software:

1. In the Controller Organizer, select Controller Tags. Right-click on the selected icon and select Monitor Tags.



2. Expand the data as necessary.


Controller Tags - example(controller)

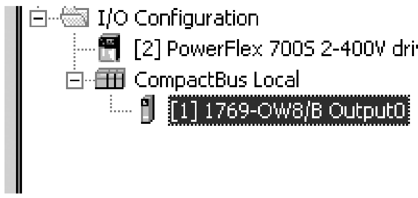
Scope: example(controller) Show: Show All Sort: Tag Name

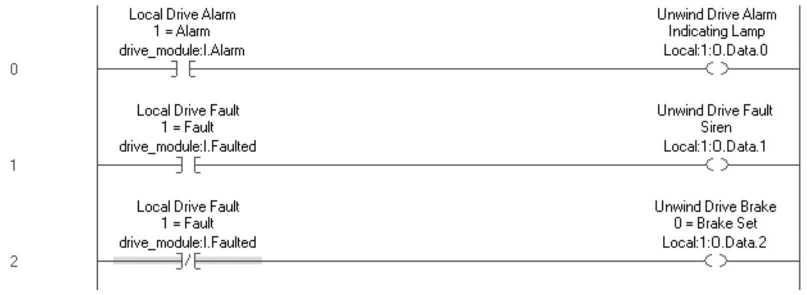
Tag Name	Value	Force
drive_module:I	{...}	
drive_module:I.LogicStatus	2#0000_000...	
drive_module:I.Active	0	
drive_module:I.Running	0	
drive_module:I.CommandDir	0	
drive_module:I.ActualDir	0	
drive_module:I.Accelerating	0	
drive_module:I.Decelerating	0	
drive_module:I.Jogging	0	
drive_module:I.Faulted	0	
drive_module:I.Alarm	0	
drive_module:I.FlashMode	0	
drive_module:I.Ready	0	

You can write logic to monitor these bits and take appropriate action if a fault or alarm occurs. For example, you may want a drive alarm to turn on a warning lamp and a drive fault to sound an alarm and set the motor brake.

Example: Energizing Alarm Lamp, Siren and Brake in Response to Fault and Alarm Status Bits

Given this configuration, the following logic checks the fault and alarm drive status bits.





Recommended Programming Techniques

Naming Tags

Use a convention when naming tags and consistently follow it. The following convention is used internally at Allen-Bradley:

<prefix>_<function>_<suffix>

Prefix - identifies the machine section in multi section programs

Prefix examples:

Sct1 = Section 1

Fan2 = Fan 2

RewA = Rewind A

Function - describes the function or signal

Function examples:

SpdRef = Speed Reference

FaultState = Status of a fault

Suffix - identifies the units of the signal or control status

Suffix examples:

Rpm = Rotations per Minute

Ok = status OK

Off = contact Off

Full tag examples:

Sct2_SpdRef_Fpm = Speed Reference, in feet per minute, on Section 2

Fan5_FaultState_OK = Status of fault, where 1 = OK, on Fan 5

▶ **TIP:** Add descriptions for each tag when creating the tag

Use Aliasing for all Communication Format Connections Between DriveLogix and the PowerFlex 700S

Using aliases for the tags in the Communication Format (i.e. Speed Control, Motion Control, Position Control, User-Defined 1 and User-Defined 2) has the following benefits:

- Improves program portability over processors and through upgrades to DriveLogix, PowerFlex 700S, and RSLogix 5000 firmware.
- Allows real names to be applied to the User Defined tags of the static assembly.
- Allows new functions of DriveLogix and the PowerFlex 700S to be clearly named in the program even if RSLogix 5000 has not been updated.
- Allows long tag names in RSLogix 5000 to be shortened for easier program viewing.
- Allows tags to be named using the above naming convention to identify machine section association.

Apply aliases to all external connections including the PowerFlex 700S Communication Format and I/O. All defined bits should be included.

Use "Periodic Tasks" to optimize processor utilization.

Name periodic tasks to identify the update time
(e.g. Periodic_020ms_P9 = 20ms task with priority 9).

Set the periodic task time appropriate programming requirements.

Important: Remember that tasks with faster the task times consume more processor bandwidth than those with slower task times.

Set the priority of each task to coincide with the task speed. Set faster tasks to higher priority.

Important: Remember lower priority numbers define higher priorities (e.g. a task with a priority number of 9 has a higher priority than one with a priority of 10).

Important: Do not set the priority number lower than 8 (the recommended priority range is 8-15). This will provide I/O scanning with optimal updating. Flex I/O and Compact I/O are coded as priority 7 for DriveLogix.

Placing and Configuring Local I/O

De-energizing the Drive to Connect or Disconnect a Cable

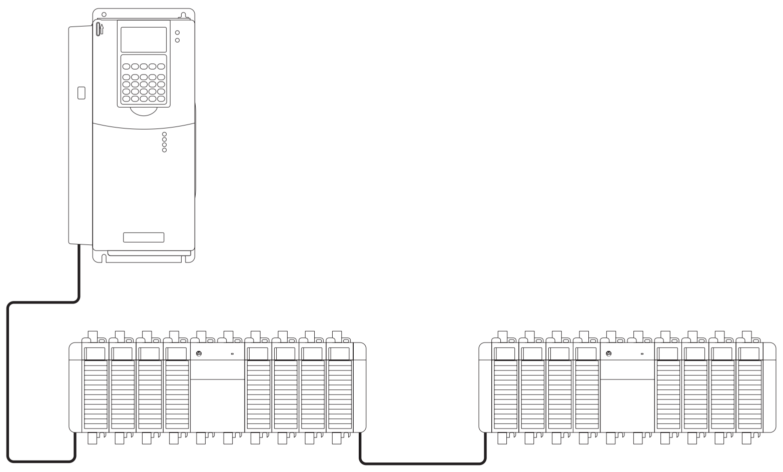
ATTENTION: Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of placing and configuring I/O you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive's input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with placing and configuring the I/O.

Understanding How the DriveLogix5730 Supports I/O

The DriveLogix5730 controller supports up to 2 banks of modules with a maximum of 8 modules per bank. You must distribute the modules within each bank so that no more than 4 modules are on either side of the power supply.



Placing Local I/O Modules

Refer to the *Compact I/O Selection Guide*, publication 1769-SG001, for information about selecting Compact I/O modules. Use the 20D-DL2-CR3 or 20D-DL2-CL3 cables to connect a DriveLogix5730 controller to a bank of Compact I/O. Use 1769-CRR1/-CRR3 or 1769-CRL1/-CRL3 expansion cable to connect banks of I/O modules. You can split a bank right after the power supply or after any I/O module. Each bank must contain one power supply. An end cap/terminator must be used on the last I/O bank opposite of the expansion cable.

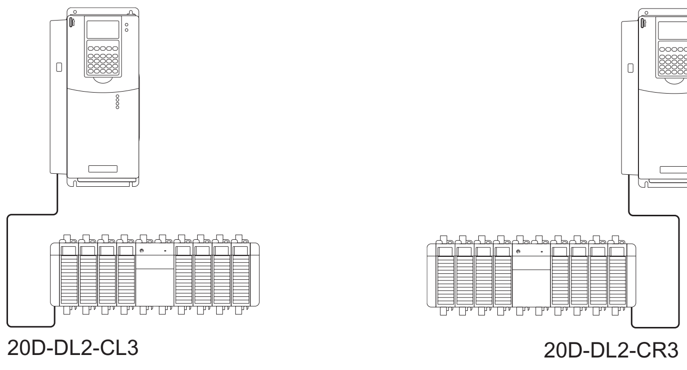
Only the local DriveLogix5730 controller can scan the local Compact I/O.

ATTENTION: The Compact I/O system does not support Removal and Insertion Under Power (RIUP).

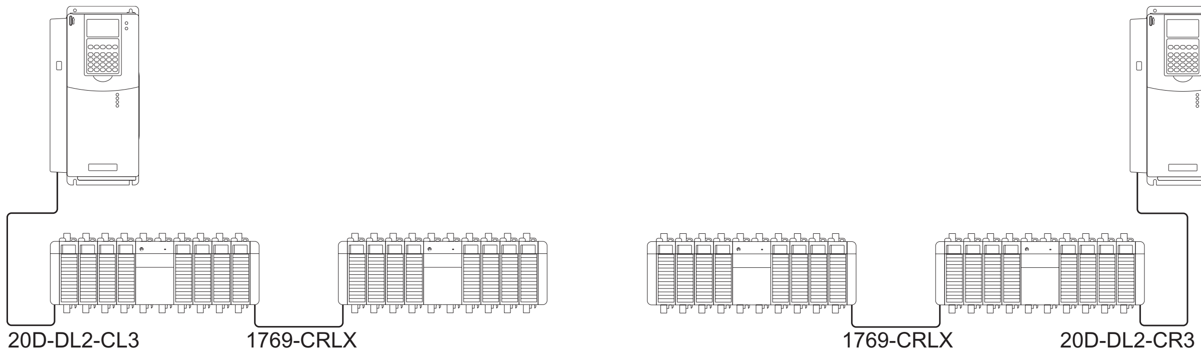
While the DriveLogix / Compact I/O is under power:

- any break in the connection between the power supply and the controller (i.e. removing the power supply, controller, or an I/O module) may subject the logic circuitry to transient conditions above the normal design thresholds and may result in damage to system components or unexpected behavior.
- removing an end cap or an I/O module faults the controller and may also result in damage to system components.

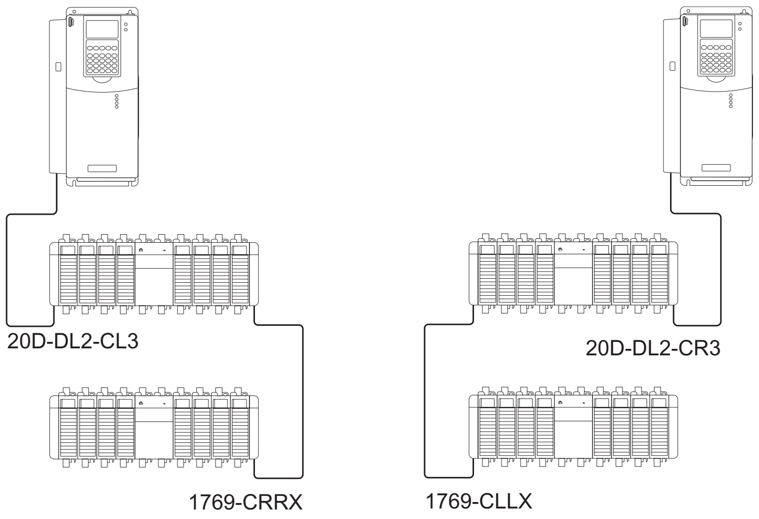
A 20D-DL2-CL3 cable connects the controller to the left side of an I/O bank. A 20D-DL2-CR3 cable connects the controller to the right side of an I/O bank.



A 1769-CRL1/-CRL3 connects the left side of one bank of Compact I/O to the right side of another. This facilitates a horizontal I/O orientation.



A 1769-CLL1/-CLL3 connects the left side of one bank of Compact I/O to the left side of another. A 1769-CRR1/-CRR3 connects the right side of one bank of Compact I/O to the right side of another. This facilitates a vertical I/O orientation.



Validating I/O Layout

To validate your planned I/O layout, consider these requirements:

- Each module in a Compact I/O system uses a set amount of backplane memory, in addition to the data that the module stores or transfers. As you add modules, the minimum backplane RPI increases.
- The I/O modules must be distributed such that the current consumed from the left or right side of the power supply never exceeds 2.0A at 5V dc and 1.0A at 24V dc.

Estimating RPI

As you install modules, the minimum backplane RPI increases. The RPI (request packet interval) defines the frequency at which the controller sends and receives all I/O data on the backplane. There is one RPI for the entire 1769 backplane. Consider these guidelines when installing modules:

Type of Module:	Considerations:
digital and analog (any mix)	• 1-4 modules can be scanned in 1.0 ms • 5-16 modules can be scanned in 1.5 ms • some input modules have a fixed 8.0 ms filter, so selecting a faster RPI has no affect
specialty	• “full-sized” 1769-SDN modules add 1.5 ms per module • 1769-HSC modules add 0.5 ms per module

You can always select an RPI that is slower than listed above. These considerations show how fast modules can be scanned - not how fast an application can use the data. The RPI is asynchronous to the program scan. Other factors, such as program execution duration, affect I/O throughput.

Determining When the Controller Updates I/O

The controller continually scans the control logic. One scan is the time it takes the controller to execute the logic once. Input data transfers to the controller and output data transfers to output modules are asynchronous to the logic scan.

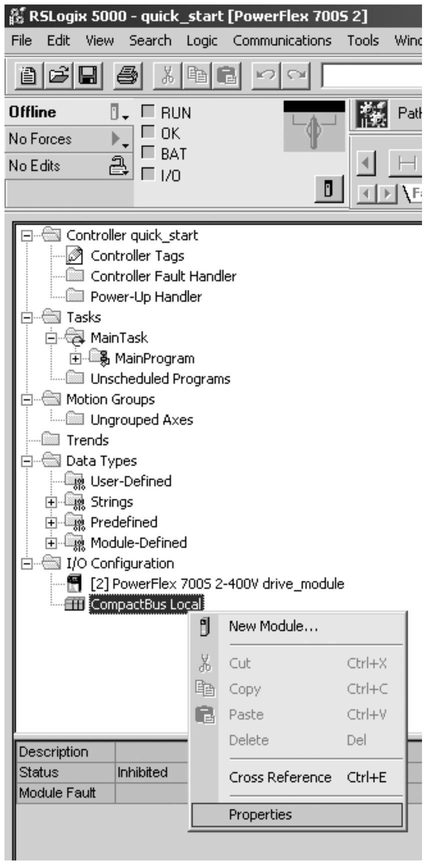
► **TIP:** If you need to ensure that the I/O values being used during logic execution are from one moment in time (such as at the beginning of a ladder program), use the Synchronous Copy instruction (CPS) to buffer I/O data.

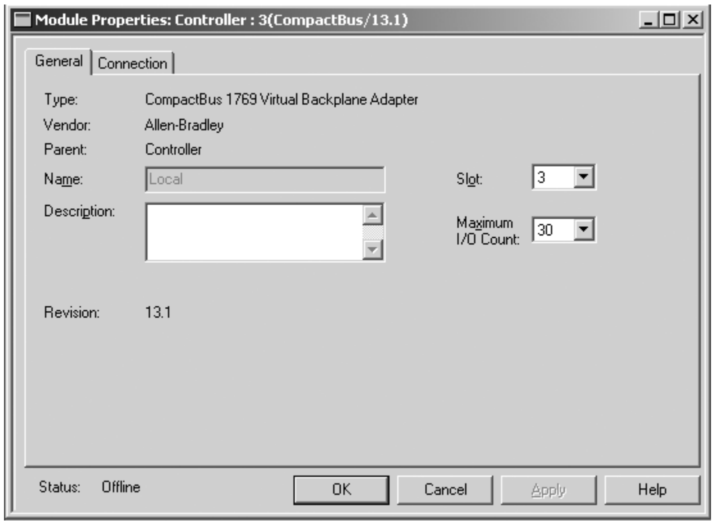
Refer to the *Logix5000 Controllers Common Procedures Programming Manual*, publication number 1756-PM001 for examples of I/O buffering or to the *Logix5000 Controllers General Instruction Set Reference Manual*, publication number 1756-RM003 for information on the CPS instruction.

Configuring the CompactBus

When you create a DriveLogix5730 project, the programming software automatically creates the local CompactBus. You must configure the CompactBus.

1. In the Controller Organizer, select CompactBus Local icon. Right click and select Properties.



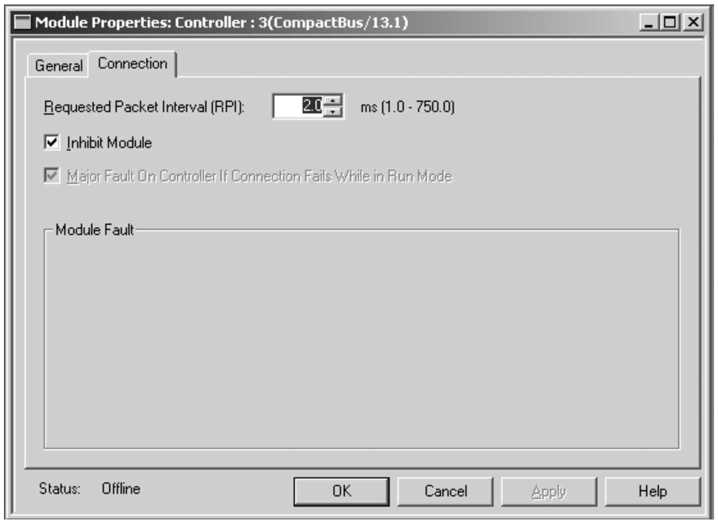


Type: CompactBus 1769 Virtual Backplane Adapter
Vendor: Allen-Bradley
Parent: Controller
Name: Local
Slot: 3
Description:
Maximum I/O Count: 30
Revision: 13.1
Status: Offline

On the General tab, specify the size of the chassis. Enter the number of modules you plan to install. Include the DriveLogix5730 controller and drive in this total, along with a maximum of 16 I/O modules, not including the power supply.

The Comm Format for the CompactBus is automatically set to Rack Optimized and cannot be changed.

Using the Connection tab, you can specify the RPI for the systems and choose to inhibit or uninhibit the CompactBus.


Module Properties: Controller : 3(CompactBus/13.1)

Requested Packet Interval (RPI): 2.0 ms (1.0 - 750.0)

☑ Inhibit Module

☑ Major Fault On Controller If Connection Fails While in Run Mode

Status: Offline

The RPI you specify here is the RPI for every 1769 module on this controller’s local CompactBus. Specify an RPI from 1-750ms for the system. You do not specify individual RPI values for each module.

By inhibiting and uninhibiting the CompactBus, you can write new configuration data to the entire system at once.

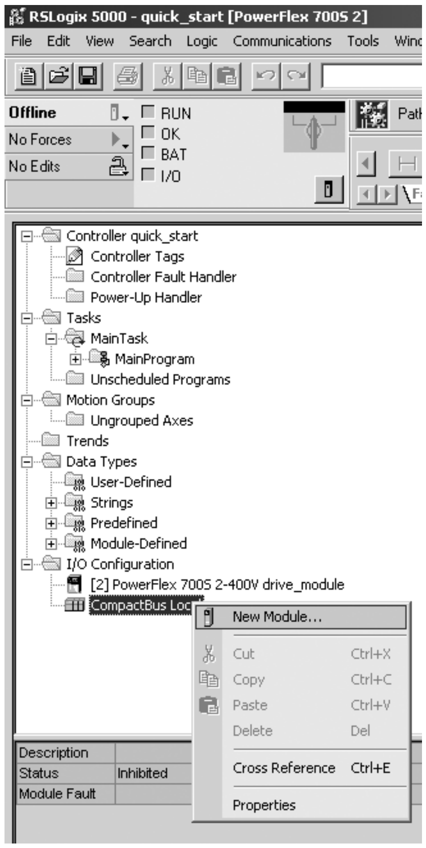
The controller’s response to a CompactBus connection failure is fixed to always fault the controller. It is not configurable.

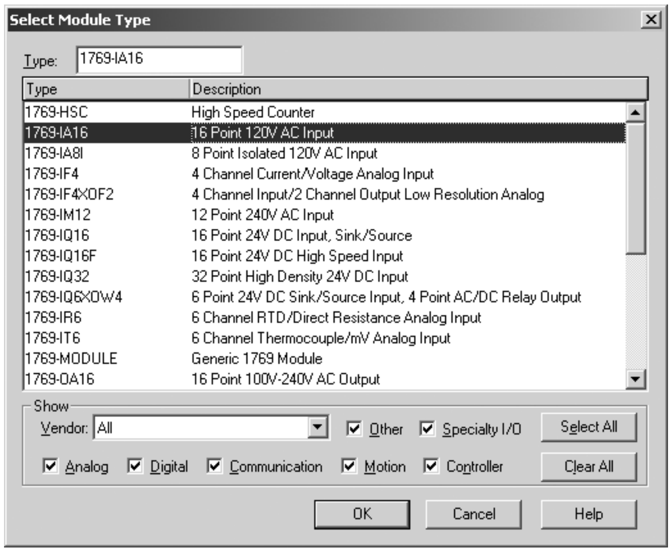
Configuring Local I/O Modules

Use your programming software to configure the I/O modules for the controller.

1. In the Controller Organizer, select CompactBus Local icon. Right click and select New Module.

2. Select the new module (1769-IA16 in this example).

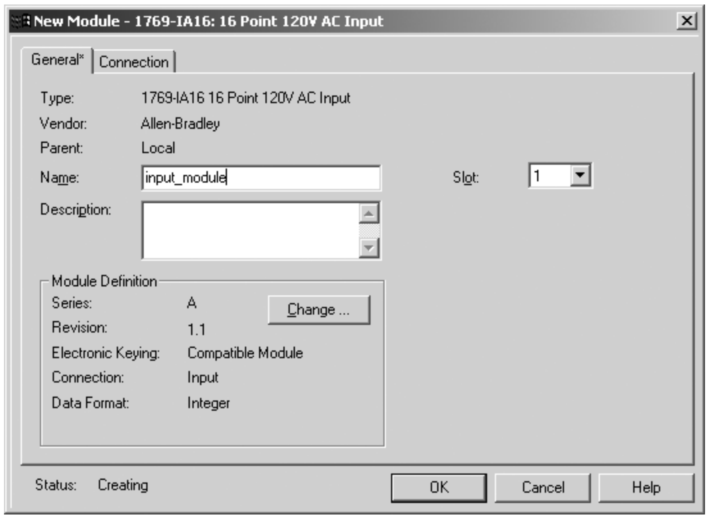



Select Module Type

Type: 1769-IA16

Type	Description
1769-HSC	High Speed Counter
1769-IA16	16 Point 120V AC Input
1769-IA8I	8 Point Isolated 120V AC Input
1769-IF4	4 Channel Current/Voltage Analog Input
1769-IF4XOF2	4 Channel Input/2 Channel Output Low Resolution Analog
1769-IM12	12 Point 240V AC Input
1769-IQ16	16 Point 24V DC Input, Sink/Source
1769-IQ16F	16 Point 24V DC High Speed Input
1769-IQ32	32 Point High Density 24V DC Input
1769-IQ6XOW4	6 Point 24V DC Sink/Source Input, 4 Point AC/DC Relay Output
1769-IR6	6 Channel RTD/Direct Resistance Analog Input
1769-IT6	6 Channel Thermocouple/mV Analog Input
1769-MODULE	Generic 1769 Module
1769-OA16	16 Point 100V-240V AC Output

Show
Vendor: All ✓ Other ✓ Specialty I/O
✓ Analog ✓ Digital ✓ Communication ✓ Motion ✓ Controller


New Module - 1769-IA16: 16 Point 120V AC Input

Type: 1769-IA16 16 Point 120V AC Input
Vendor: Allen-Bradley
Parent: Local
Name: input_module
Slot: 1
Description:

Module Definition
Series: A
Revision: 1.1
Electronic Keying: Compatible Module
Connection: Input
Data Format: Integer

Status: Creating

3. Configure the module, using the module wizard to specify characteristics. Click Next to advance the wizard. Click Finish when you are done.

   The completed module will appear in the Controller Organizer.

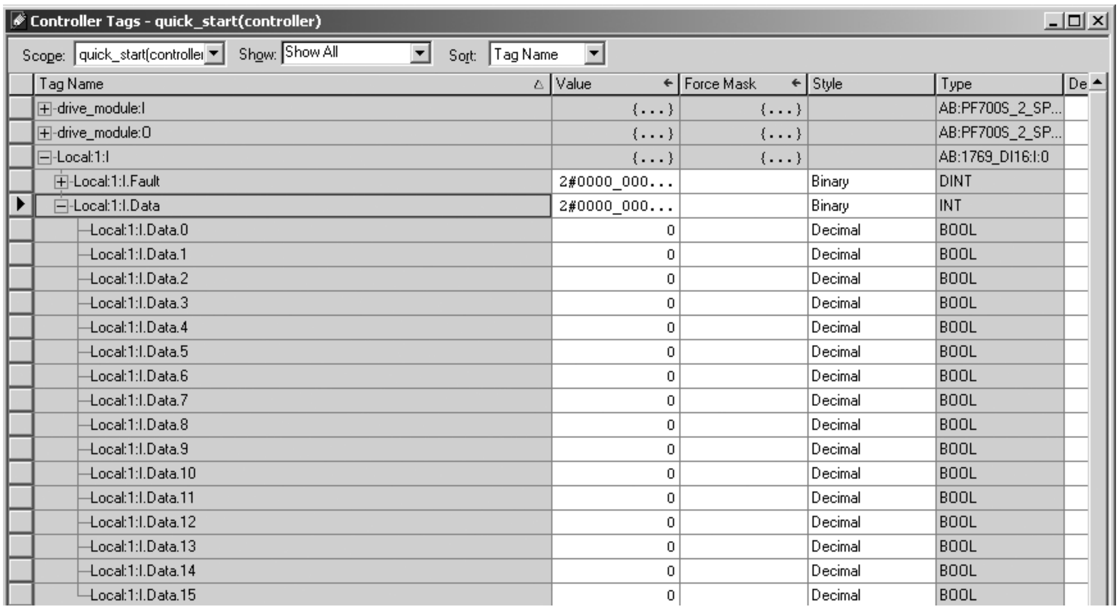
Communication Formats

The communication format determines the data structure the I/O module uses. Each format supports a different data structure. Presently, the DriveLogix5730 controller supports two data formats:

- Input Data – INT (for 1769 input modules)
- Data – INT (for 1769 output modules)

TIP: The DriveLogix5730 controller must own its local I/O modules. No other Logix-based controller can own the local Compact I/O.

The communication format determines the tag structure that is created for the module. Assume that a 1769-IA16 Input module is in slot 1. The software creates the appropriate tags using the slot number to differentiate the tags for this example module from any other module.


Controller Tags - quick_start(controller)
Scope: quick_start(controller
Show: Show All
Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type	De
drive_module:I	{...}	{...}		AB:PF700S_2_SP...	
drive_module:O	{...}	{...}		AB:PF700S_2_SP...	
Local:1:I	{...}	{...}		AB:1769_DI16:I:0	
Local:1:I.Fault	2#0000_000...		Binary	DINT	
Local:1:I.Data	2#0000_000...		Binary	INT	
Local:1:I.Data.0	0		Decimal	BOOL	
Local:1:I.Data.1	0		Decimal	BOOL	
Local:1:I.Data.2	0		Decimal	BOOL	
Local:1:I.Data.3	0		Decimal	BOOL	
Local:1:I.Data.4	0		Decimal	BOOL	
Local:1:I.Data.5	0		Decimal	BOOL	
Local:1:I.Data.6	0		Decimal	BOOL	
Local:1:I.Data.7	0		Decimal	BOOL	
Local:1:I.Data.8	0		Decimal	BOOL	
Local:1:I.Data.9	0		Decimal	BOOL	
Local:1:I.Data.10	0		Decimal	BOOL	
Local:1:I.Data.11	0		Decimal	BOOL	
Local:1:I.Data.12	0		Decimal	BOOL	
Local:1:I.Data.13	0		Decimal	BOOL	
Local:1:I.Data.14	0		Decimal	BOOL	
Local:1:I.Data.15	0		Decimal	BOOL	

Hold Last State and User-Defined Safe State Not Supported

ATTENTION: Risk of injury or equipment damage exists. When 1769 Compact I/O modules are used as local I/O modules in a DriveLogix5730 system, the local I/O modules do not support the Hold Last State or User-Defined Safe State features, even though you can configure these options in the programming software. Precautions should be taken to ensure that these settings do not create a hazard of injury or equipment damage.

If a local I/O module fails such that its communication to the controller is lost, or if any module is disconnected from the system bus while under power, the controller will go into the fault mode. All outputs turn off when the system bus or any module faults.

RSLogix 5000 software creates tags for modules when you add them to the I/O configuration. The 1769 module tags define configuration (C) data type members which may include attributes for alternate outputs. DriveLogix5730 does not enable local modules to use the alternate outputs. Do not configure the attributes listed below:

For digital output modules:	For analog output modules:
• ProgToFaultEn • ProgMode • ProgValue • FaultMode • FaultValue	• CHxProgToFaultEn • CHxProgMode • CHxFaultMode • where CHx = the channel number

Any 1769 Compact I/O modules used as remote I/O modules in a DriveLogix5730 system do support the Hold Last State and User-Defined Safe State features.

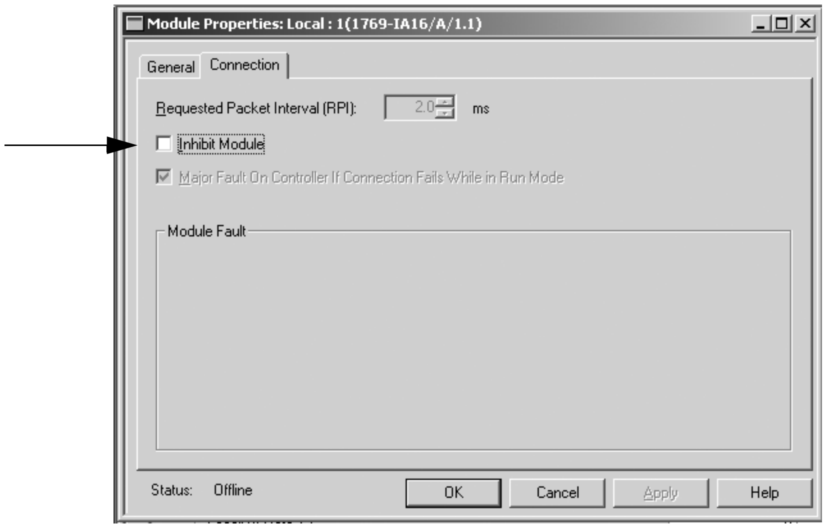
Inhibiting I/O Module Operation

In some situations, such as when initially commissioning a system, it is useful to disable portions of a control system and enable them as you wire up the control system. The controller lets you inhibit individual modules or groups of modules, which prevents the controller from trying to communicate with these modules. Inhibiting a module shuts down the connection from the controller to that module.

When you create an I/O module, it defaults to being not inhibited. You can change an individual module's properties to inhibit a module.

ATTENTION: Inhibiting a module causes the connection to the module to be broken and prevents communication of I/O data. The controller and other I/O modules continue to operate based on old data from that module. To avoid potential injury and damage to machinery, make sure this does not create unsafe operation.

On the Connection tab of the Module Properties dialog, you can select to inhibit that specific module.


Module Properties: Local : 1(1769-IA16/A/1.1)
General
Connection
Requested Packet Interval (RPI): 2.0 ms
Inhibit Module
Major Fault On Controller If Connection Fails While in Run Mode
Module Fault
Status: Offline
OK
Cancel
Apply
Help

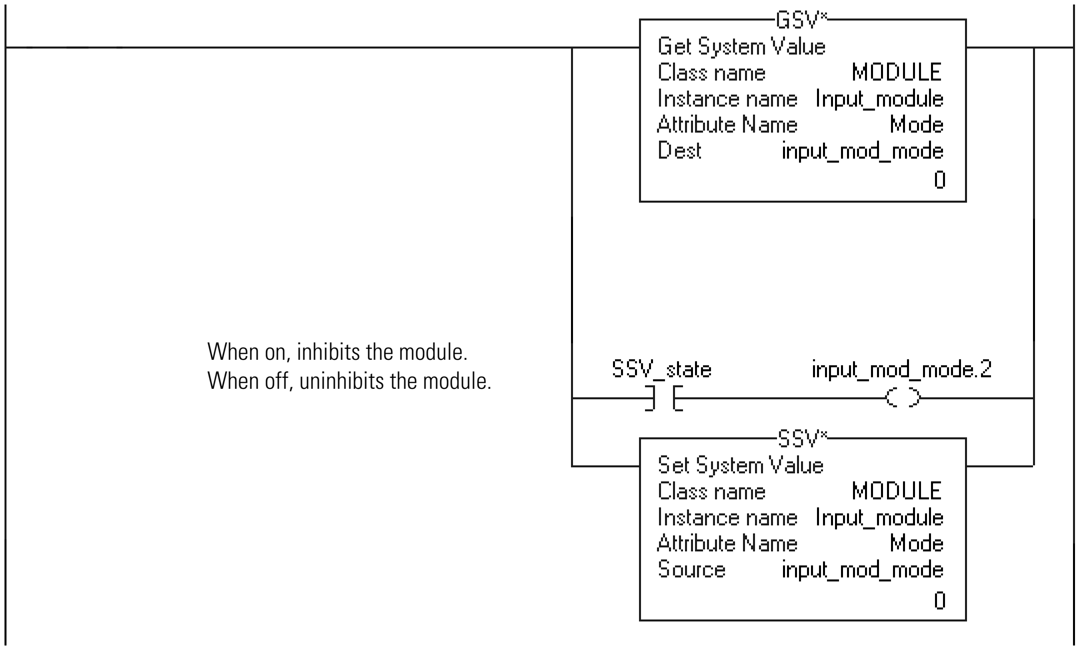
TIP: To easily inhibit all local I/O modules, you can inhibit the CompactBus, which in turn inhibits all the modules on that bus. See Configuring the CompactBus on page 3-5.

When you select to inhibit a module, the controller organizer displays a yellow circle symbol over the module.

If you are:	Inhibit a module to:
offline	put a place holder for a module you are configuring. The inhibit status is stored in the project. When you download the project, the module is still inhibited.
online	stop communication to a module. If you inhibit a module while you are connected to the module, the connection to the module is closed. The module's outputs turn off. If you inhibit a module but a connection to the module was not established (perhaps due to an error condition or fault), the module is inhibited. The module status information changes to indicate that the module is inhibited and not faulted. If you uninhibit a module (clear the check box), and no fault condition occurs, a connection is made to the module and the module is dynamically reconfigured with the configuration you created for that module. If you uninhibit the module and a fault condition occurs, a connection is not made to the module. The module status information changes to indicate the fault condition.

To inhibit a module from logic, you must first read the Mode attribute for the module using a GSV instruction. Set bit 2 to the inhibit status (1 to inhibit or 0 to uninhibit). Use a SSV instruction to write the Mode attribute back to the module. For example:

The GSV instruction gets the current status of the module named “input_module.” The SSV instruction sets the state of “input_module” as either inhibited or uninhibited.



Sending Module Configuration Information

The controller sends module configuration information once module connections are established.

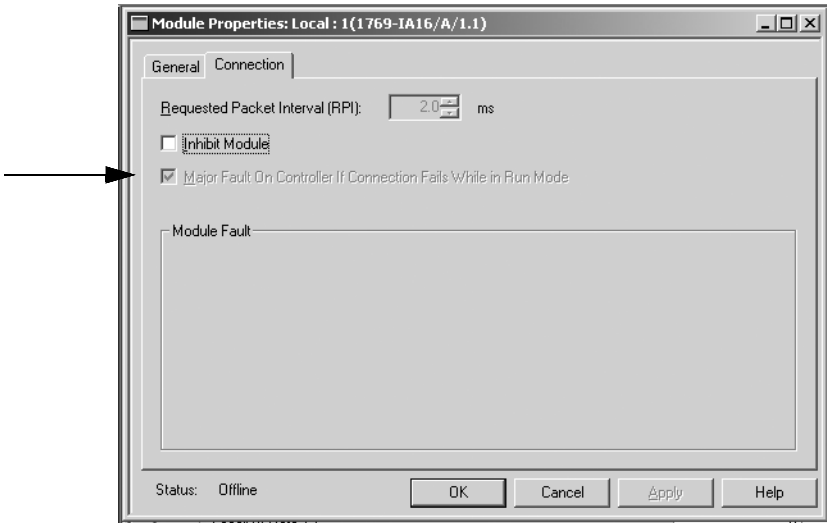
ATTENTION: If you make a configuration change to any module in the system do one of the following to resend module configuration data:

- cycle power to the controller
- inhibit and then uninhibit the bus
- inhibit and then uninhibit the individual module
- send a MSG instruction of type Module Reconfigure (for information on configuring a MSG to send configuration data, see the Logix5000 Controllers General Instructions Reference Manual, publication 1756-RM003)

Configuring the Controller's Response to a Connection Failure

In a DriveLogix5730 system, the controller's response to a CompactBus connection failure is fixed to always fault the controller. The CompactBus setting supersedes the individual module's setting.

Important: The controller's response to a connection failure of any I/O module is fixed to always fault the controller.


Module Properties: Local : 1(1769-IA16/A/1.1)

General | Connection

Requested Packet Interval (RPI): 2.0 ms

☐ Inhibit Module

☑ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

The I/O modules respond to a connection failure by turning off output.

Accessing I/O Data

The programming software displays I/O data as structures of multiple tags that depend on the specific features of the I/O module. The names of the data structures are based on the location of the I/O module. The programming software automatically creates the necessary structures and tags when you configure the module. Each tag name follows this format:

Location:SlotNumber:Type.MemberName.SubMemberName.Bit

where:

This address variable:	Is:
Location	Identifies network location LOCAL = local chassis
SlotNumber	Slot number of I/O module in its chassis

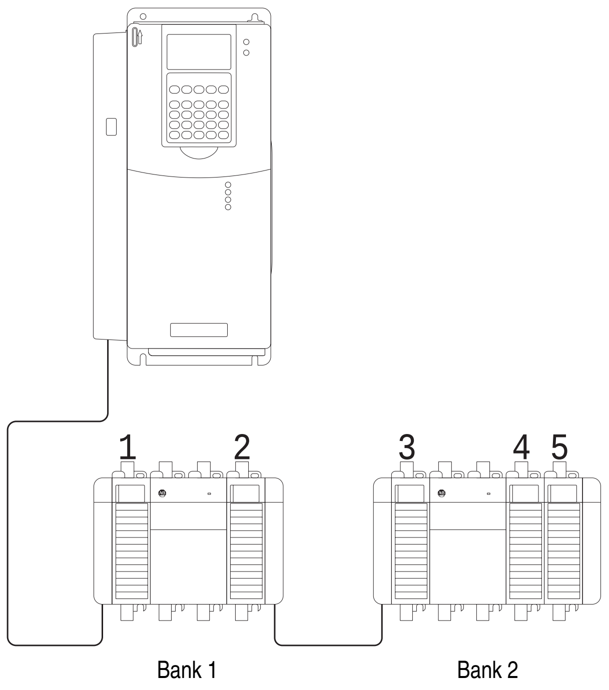
Type	Type of data
	I = input
	O = output
	C = configuration

This address variable:	Is:
MemberName	Specific data from the I/O module; depends on the type of data the module can store For example, Data and Fault are possible fields of data for an I/O module. Data is the common name for values that are sent to or received from I/O points.
SubMemberName	Specific data related to a MemberName.
Bit (optional)	Specific point on the I/O module; depends on the size of the I/O module (0-31 for a 32-point module)

The following examples show addresses for data in a DriveLogix5730 system.

Example:

I/O module on the local CompactBus utilizing two banks



Sample tag names for this example:

Location:	Example Tag Name:
input module in slot 1, LOCAL Bank 1	Local:1:C Local:1:I
output module in slot 2, LOCAL Bank 1	Local:2:C Local:2:I Local:2:O
analog input module in slot 3, LOCAL Bank 2	Local:3:C Local:3:I
analog output module in slot 4, LOCAL Bank 2	Local:4:C Local:4:I Local:4:O
analog input module in slot 5, LOCAL Bank 2	Local:5:C Local:5:I

Using aliases to simplify tag names

An alias lets you create a tag that represents another tag. This is useful for defining descriptive tag names for I/O values. For example:

Example:		Description:
I/O structure	Local:1:I:Data[0].0 Local:1:I:Fault.0	The aliases describe the specific I/O points.

alias	light_on = Local:1:I:Data[0].0
	module_failed = Local:1:I:Fault.0	

Direct Connections for I/O Modules

Each local I/O module uses a direct connection to the DriveLogix5730 controller. A direct connection is a real-time, data transfer link between the controller and an I/O module. The controller maintains and monitors the connection between the controller and the I/O module. Any break in the connection, such as a module fault, causes the controller to set fault status bits in the input data area associated with the module.

ATTENTION: The DriveLogix5730 controller does not support Removal and Insertion Under Power (RIUP). While the DriveLogix5730 system is under power:

- any break in the connection between the power supply and the controller (i.e. removing the power supply, controller, or an I/O module) may subject the logic circuitry to transient conditions above the normal design thresholds and may result in damage to system components or unexpected behavior.
- removing an end cap or an I/O module faults the controller and may also result in damage to system components.

Monitoring I/O Modules

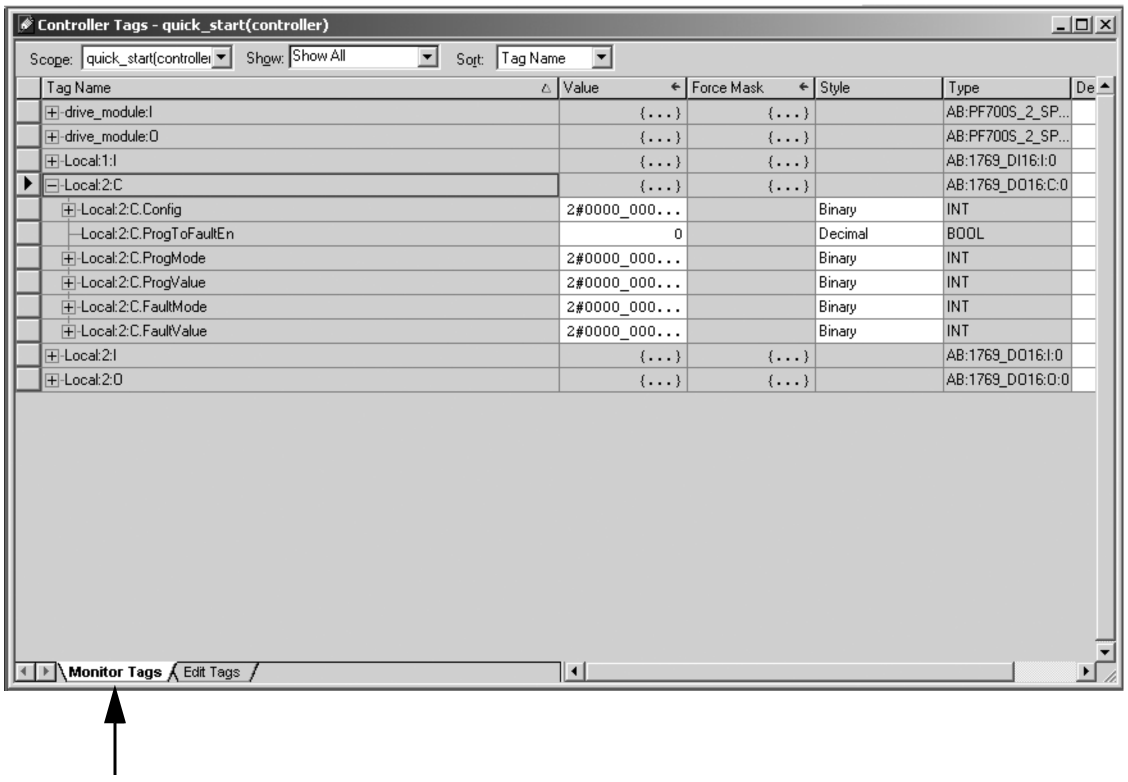
The DriveLogix5730 controller offers different levels at which you can monitor I/O modules. You can:

- use the programming software to display fault data (See Displaying Fault Data on page 3-15)
- program logic to monitor fault data so you can take appropriate action (Refer to *Logix5000 Controllers Common Procedures Programming Manual*, publication number 1756-PM001, for examples.)

Displaying Fault Data

Fault data for certain types of module faults can be viewed through the programming software.

To view this data, select Controller Tags in the Controller Organizer. Right-click to select Monitor Tags.

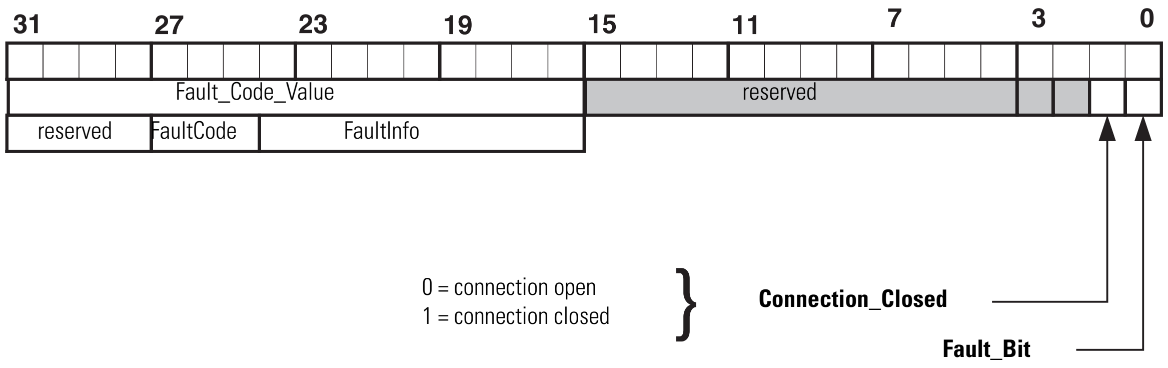

Controller Tags - quick_start(controller)

Scope: quick_start(controlle Show: Show All Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type	De
+-drive_module:I	{...}	{...}		AB:PF700S_2_SP...	
+-drive_module:O	{...}	{...}		AB:PF700S_2_SP...	
+-Local:1:I	{...}	{...}		AB:1769_DI16:I:0	
−-Local:2:C	{...}	{...}		AB:1769_DO16:C:0	
+-Local:2:C.Config	2#0000_000...		Binary	INT	
-Local:2:C.ProgToFaultEn	0		Decimal	BOOL	
+-Local:2:C.ProgMode	2#0000_000...		Binary	INT	
+-Local:2:C.ProgValue	2#0000_000...		Binary	INT	
+-Local:2:C.FaultMode	2#0000_000...		Binary	INT	
+-Local:2:C.FaultValue	2#0000_000...		Binary	INT	
+-Local:2:I	{...}	{...}		AB:1769_DO16:I:0	
+-Local:2:O	{...}	{...}		AB:1769_DO16:O:0	

The display for the fault data defaults to decimal. Change it to Hex to read the fault code.

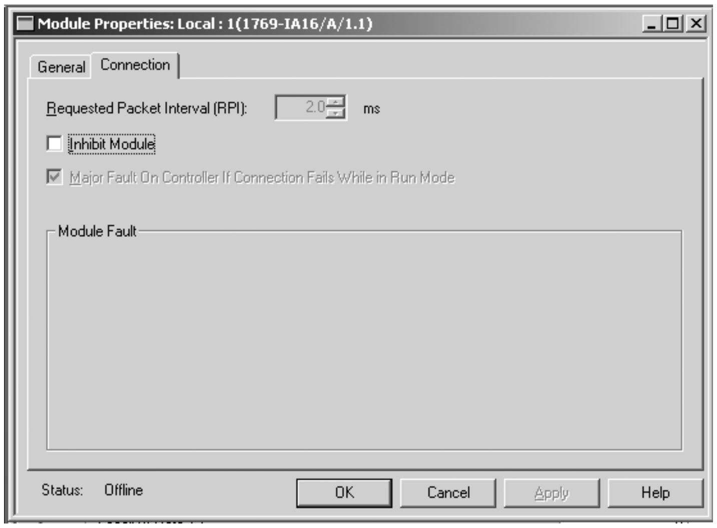
If the module faults, but the connection to the controller remains open, the controller tags database displays the fault value 16#0E01_0001. The fault word uses this format:



Where:

Bit	Description
Fault_Bit	This bit indicates that at least one bit in the fault word is set (1). If all the bits in the fault word are cleared (0), this bit is cleared (0).
Connection_Closed	This bit indicates whether the connection to the module is open (0) or closed (1). If the connection is closed (1), the Fault_Bit it set (1).

You can also view module fault data on the Connection tab of the Module Properties screen.


Module Properties: Local : 1(1769-IA16/A/1.1)

Requested Packet Interval (RPI): 2.0 ms

[ ] Inhibit Module

[✓] Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Status: Offline

See your 1769 module's user documentation for a description of module faults. To recover from module faults, correct the module fault condition and send new data to the module by downloading the user program with configuration data, inhibiting and then uninhibiting the module, or cycling power.

End-cap Detection and Module Faults

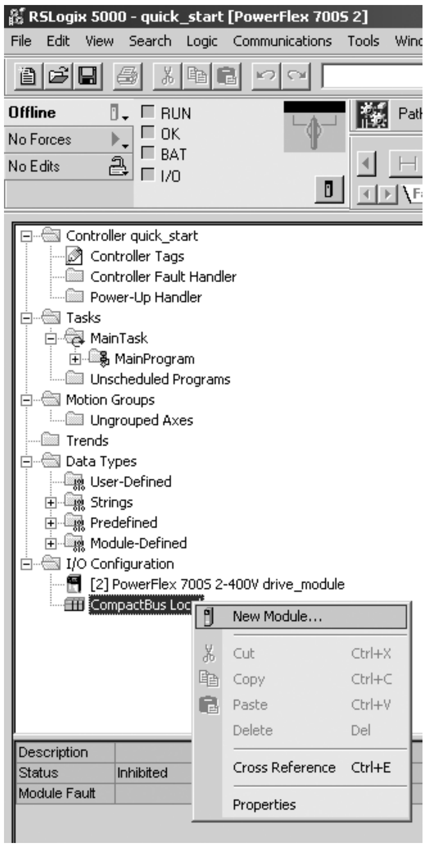
If a module that is not adjacent to an end cap experiences a fault and the connection to the controller is not broken, only the module enters the fault state.

If a module that is adjacent to an end cap experiences a fault, both the module and the controller transition to the fault state.

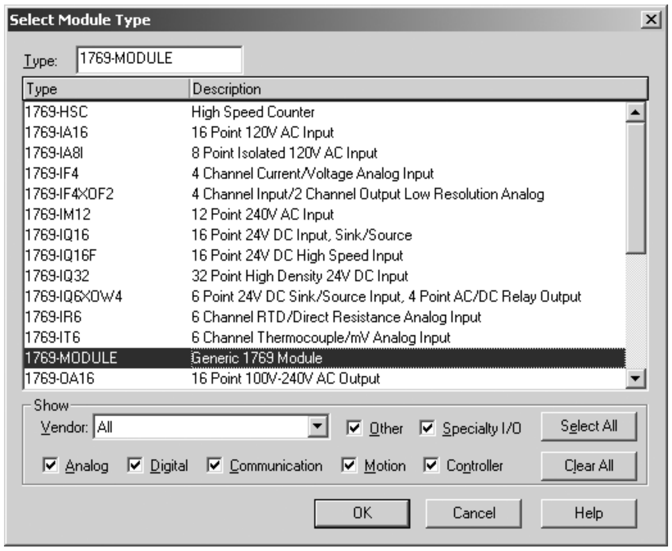
Configuring I/O Modules Using the Generic 1769-MODULE

Use the Generic 1769 Module only when a 1769 I/O module does not appear in the list of modules to add to the Controller Organizer. To configure a 1769 I/O module for a DriveLogix5730 controller using the generic 1769-MODULE:

1. In the Controller Organizer, select CompactBus Local icon. Right click and select New Module.



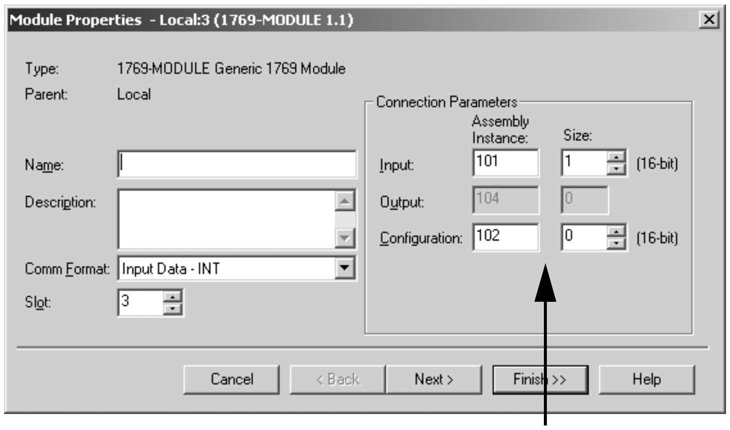
2. Select the 1769-MODULE (Generic 1769 Module).



Type	Description
1769-HSC	High Speed Counter
1769-IA16	16 Point 120V AC Input
1769-IA8I	8 Point Isolated 120V AC Input
1769-IF4	4 Channel Current/Voltage Analog Input
1769-IF4XOF2	4 Channel Input/2 Channel Output Low Resolution Analog
1769-IM12	12 Point 240V AC Input
1769-IQ16	16 Point 24V DC Input, Sink/Source
1769-IQ16F	16 Point 24V DC High Speed Input
1769-IQ32	32 Point High Density 24V DC Input
1769-IQ6XOW4	6 Point 24V DC Sink/Source Input, 4 Point AC/DC Relay Output
1769-IR6	6 Channel RTD/Direct Resistance Analog Input
1769-IT6	6 Channel Thermocouple/mV Analog Input
1769-MODULE	Generic 1769 Module
1769-OA16	16 Point 100V-240V AC Output

3.Configure the module, using the module wizard to specify characteristics. Click Next to advance the wizard. Click Finish when you are done.

The completed module will appear in the Controller Organizer.


Module Properties - Local:3 (1769-MODULE 1.1)

Type: 1769-MODULE Generic 1769 Module
Parent: Local
Name:
Description:
Comm Format: Input Data - INT
Slot: 3

Connection Parameters

	Assembly Instance:	Size:	
Input:	101	1	(16-bit)
Output:	104	0	
Configuration:	102	0	(16-bit)

The generic module requires you to specify more parameters of the module.

Important: The values you enter for these parameters are device specific. See the documentation for the device to determine which values to enter.

On the generic module screen, you define the parameters of the module.

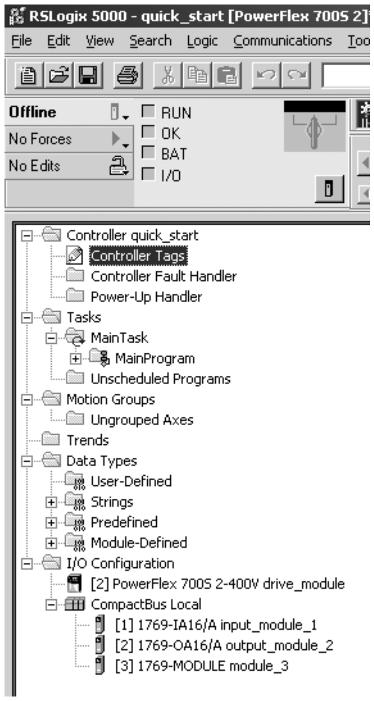
In this field:	Specify:
Name	name of the module
Description	(optional) provide more details about the module
Comm Format	communication format 1769 analog output modules, digital output modules, analog combination modules, and digital combination modules, use Data – INT. 1769 analog input modules and digital input modules use Input Data – INT.
Slot	slot placement of the module on the CompactBus
Connection Parameters Input Output Configuration	connection information unique to the module The documentation for module should list the assembly instance and size numbers for the input, output, and configuration parameters.

Entering the Configuration Information for the Module

Once you configure a module using the generic 1769-MODULE, you must enter the configuration information for the module into the tag database. The configuration information is downloaded to the module at program download, power up, and whenever a module is inhibited and then uninhibited.

1. In the Controller Organizer, double-click on Controller Tags.

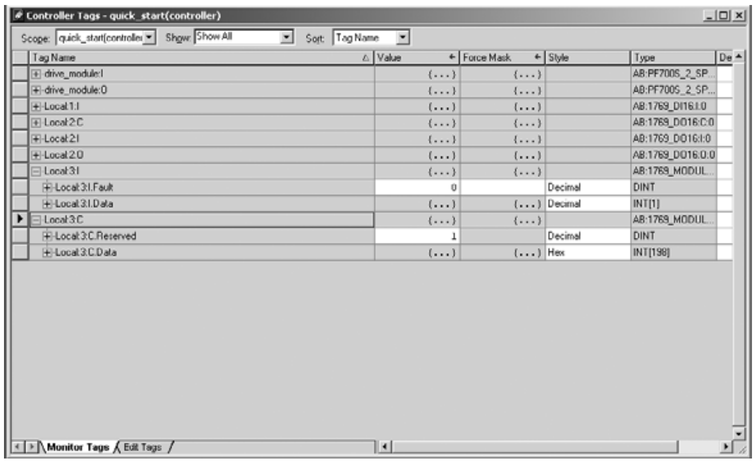
2. Edit the tags for the module so that the tags contain the appropriate configuration



```
RSLogix 5000 - quick_start [PowerFlex 700S 2]
File  Edit  View  Search  Logic  Communications  Too

Offline     RUN
No Forces   OK
No Edits    BAT
            I/O

Controller quick_start
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
Tasks
    MainTask
        MainProgram
    Unscheduled Programs
Motion Groups
    Ungrouped Axes
Trends
Data Types
    User-Defined
    Strings
    Predefined
    Module-Defined
I/O Configuration
    [2] PowerFlex 700S 2-400V drive_module
    CompactBus Local
        [1] 1769-IA16/A input_module_1
        [2] 1769-OA16/A output_module_2
        [3] 1769-MODULE module_3
```



Controller Tags - quick_start(controller)

Scope: quick_start(controlle Show: Show All Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type	De
drive_module:I	{...}	{...}		AB:PF700S_2_SP...	
drive_module:O	{...}	{...}		AB:PF700S_2_SP...	
Local:1:I	{...}	{...}		AB:1769_DI16:I:0	
Local:2:C	{...}	{...}		AB:1769_DO16:C:0	
Local:2:I	{...}	{...}		AB:1769_DO16:I:0	
Local:2:O	{...}	{...}		AB:1769_DO16:O:0	
Local:3:I	{...}	{...}		AB:1769_MODUL...	
Local:3:I.Fault	0		Decimal	DINT	
Local:3:I.Data	{...}	{...}	Decimal	INT[1]	
Local:3:C	{...}	{...}		AB:1769_MODUL...	
Local:3:C.Reserved	1		Decimal	DINT	
Local:3:C.Data	{...}	{...}	Hex	INT[198]	

Monitor Tags / Edit Tags

The generic module was added to slot 3, so you want to enter configuration data into the Local:3:C tags.

RSLogix 5000 programming software automatically create tags for configured I/O modules. All local I/O addresses are preceded by the word Local. These addresses have the following format:

- Input Data: Local:s:I
- Output Data: Local:s:O
- Configuration Data: Local:s:C

Where *s* is the slot number assigned the I/O module.

Open the configuration tag for that module by clicking on the plus sign to the left of its configuration tag in the tag database. The configuration information depends on the module. See the documentation on the I/O module for the appropriate configuration information.

Notes:

Configuring DriveLogix Motion

De-energizing the Drive to Connect or Disconnect a Cable

ATTENTION: Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of configuring DriveLogix motion you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.

2. Verify that there is no voltage at the drive’s input power terminals.

3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

4. Connect or disconnect the programming or network cable.

5. Turn power back on and proceed with configuring DriveLogix motion.

About this Chapter

This chapter introduces DriveLogix5730 motion. The steps in this chapter provide the minimum settings required to begin testing DriveLogix motion.

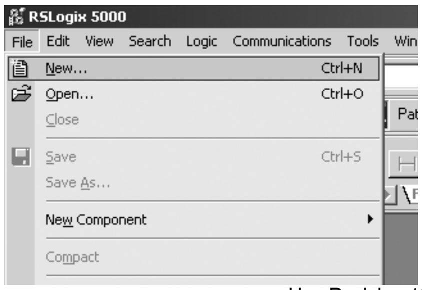
System Requirements

- PowerFlex 700S Phase II Drive with firmware revision 1.XX or higher
- DriveLogix5730 Controller with firmware revision 13.XX or higher
- DriveExecutive programming software version 3.02 or higher
- RSLogix 5000 programming software version 13 or higher

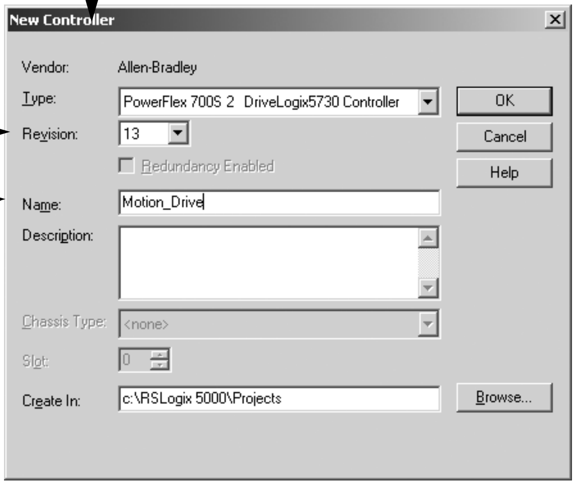
Programming the Controller

In RSLogix 5000, create a new project.

1. From the File Menu, select New.



2. Define the project.



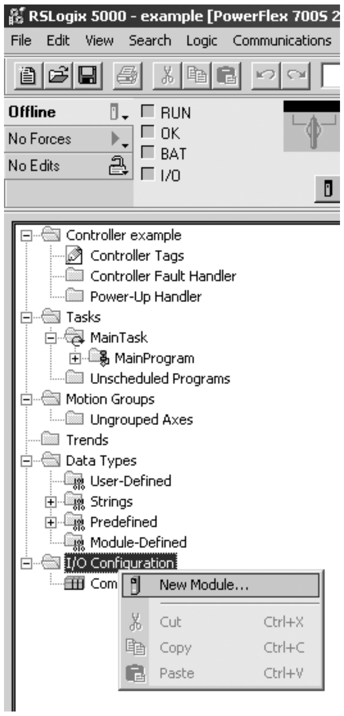
Vendor: Allen-Bradley
Type: PowerFlex 700S 2 DriveLogix5730 Controller
Revision: 13
Name: Motion_Drive
Description:
Create In: c:\RSLogix 5000\Projects

Use Revision 13 or

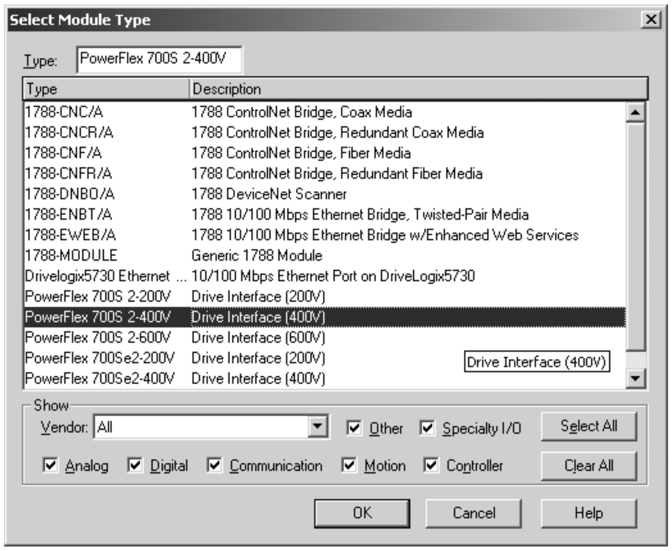
You must enter a Name.

Click OK.

3. In the Controller Organizer, select the I/O Configuration folder.
Right-click the selected folder and select New Module.



4. Select the drive (PowerFlex 700S-400V in this example).



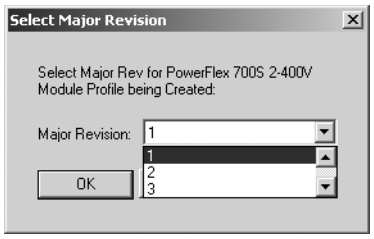
Type: PowerFlex 700S 2-400V

Type	Description
1788-CNC/A	1788 ControlNet Bridge, Coax Media
1788-CNCR/A	1788 ControlNet Bridge, Redundant Coax Media
1788-CNF/A	1788 ControlNet Bridge, Fiber Media
1788-CNFR/A	1788 ControlNet Bridge, Redundant Fiber Media
1788-DNBO/A	1788 DeviceNet Scanner
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1788-MODULE	Generic 1788 Module
Drivelogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730
PowerFlex 700S 2-200V	Drive Interface (200V)
PowerFlex 700S 2-400V	Drive Interface (400V)
PowerFlex 700S 2-600V	Drive Interface (600V)
PowerFlex 700Se2-200V	Drive Interface (200V)
PowerFlex 700Se2-400V	Drive Interface (400V)

Vendor: All
☑ Other ☑ Specialty I/O
☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller

Click OK.

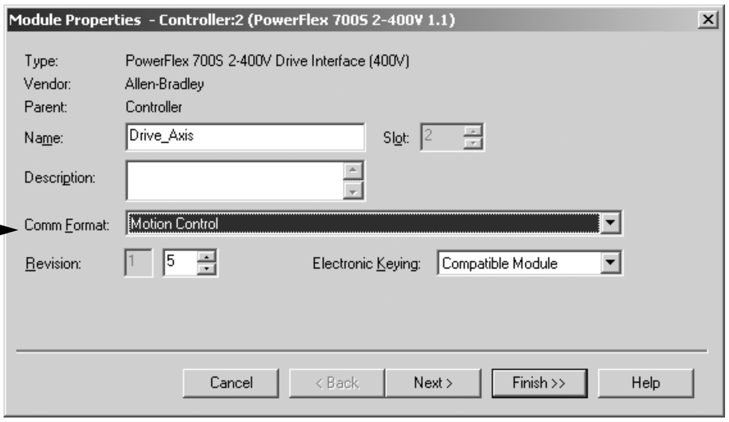
5. Select the Major Revision.



Select Major Rev for PowerFlex 700S 2-400V Module Profile being Created:
Major Revision: 1

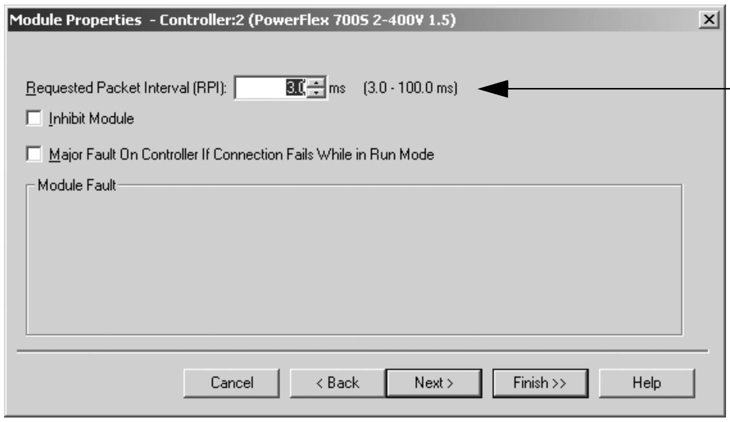
4. Configure the drive. Use the module properties wizard to specify characteristics for the module. Click Next to continue through the wizard.

Select the Motion Control Communication Format.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.1)
Cancel
< Back
Next >
Finish >>
Help

Type: PowerFlex 700S 2-400V Drive Interface (400V)
Vendor: Allen-Bradley
Parent: Controller
Name: Drive_Axis
Slot: 2
Description:
Comm Format: Motion Control
Revision: 1 5
Electronic Keying: Compatible Module

Click Next.



Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

Requested Packet Interval (RPI): 3.0 ms (3.0 - 100.0 ms)

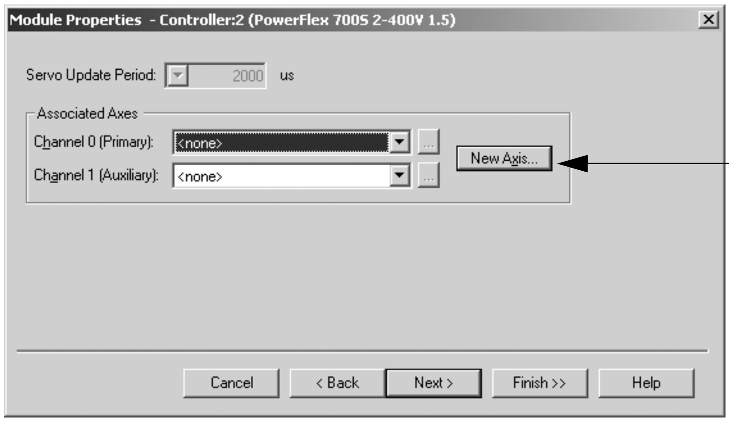
☐ Inhibit Module

☐ Major Fault On Controller If Connection Fails While in Run Mode

Module Fault

Change the RPI to 3.0 ms.

Click Next.



Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

Servo Update Period: 2000 us

Associated Axes

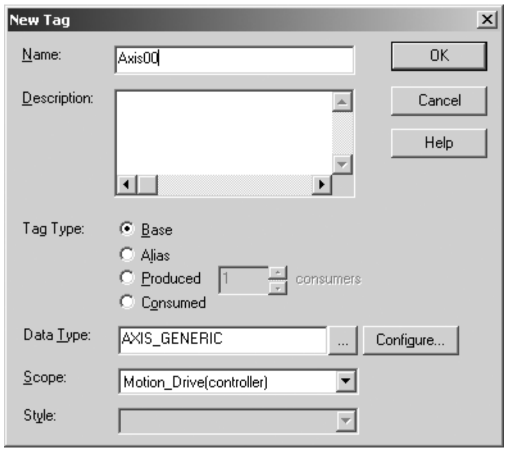
Channel 0 (Primary): <none>

Channel 1 (Auxiliary): <none>

Click New Axis, to assign a channel to an axis.

5. Create new tag structures for Axis00 and Axis01.

Type Axis00 in the Name field and click OK. Repeat for Axis01.



New Tag

Name: Axis00

Description:

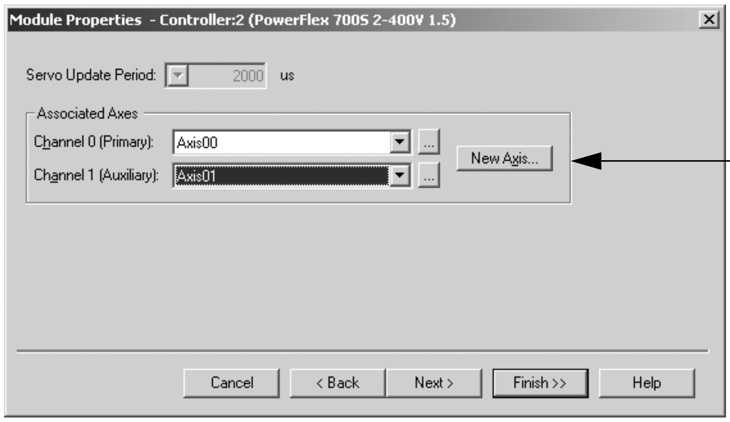
Tag Type: [✓ Base] Alias Produced 1 consumers Consumed

Data Type: AXIS_GENERIC

Scope: Motion_Drive(controller)

Style:

6. Return to configuring the new drive.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

Servo Update Period: 2000 us

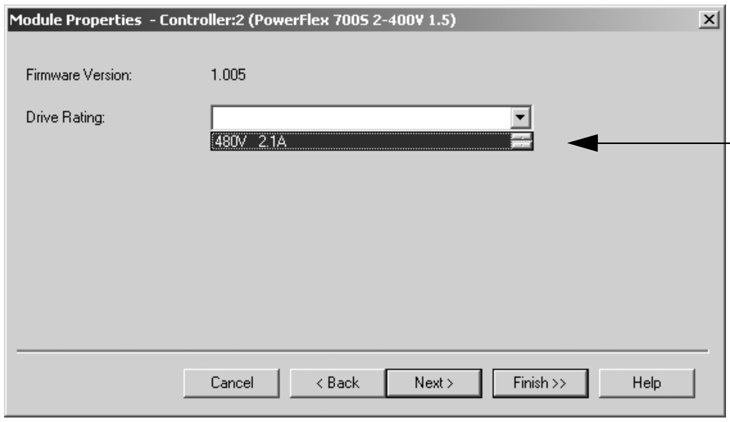
Associated Axes

Channel 0 (Primary): Axis00

Channel 1 (Auxiliary): Axis01

Assign the new axes to the channels.

Click Next.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

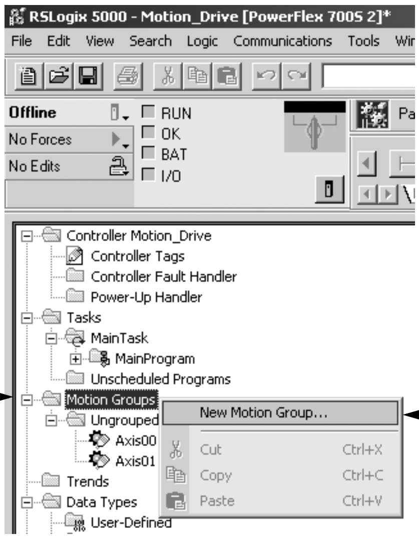
Firmware Version: 1.005

Drive Rating: 480V 2.1A

Select the proper Drive Rating.

Click Finish.

7. Create a new Motion Group.

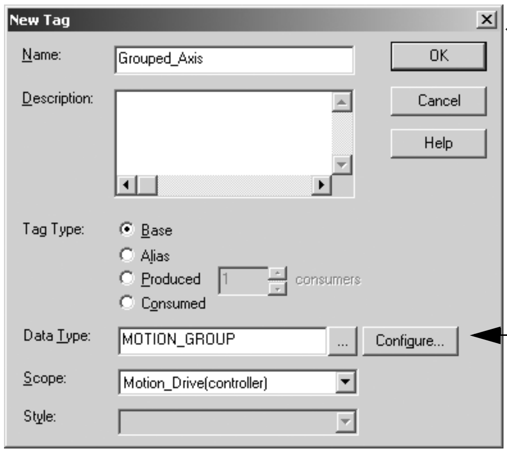


Right click on Motion Groups

Select New Motion Group.

7. Create a new Motion Group continued…

Enter a Name for the Motion Group.



Name: Grouped_Axis

Description:

Tag Type: [✓ Base] Alias Produced 1 consumers Consumed

Data Type: MOTION_GROUP

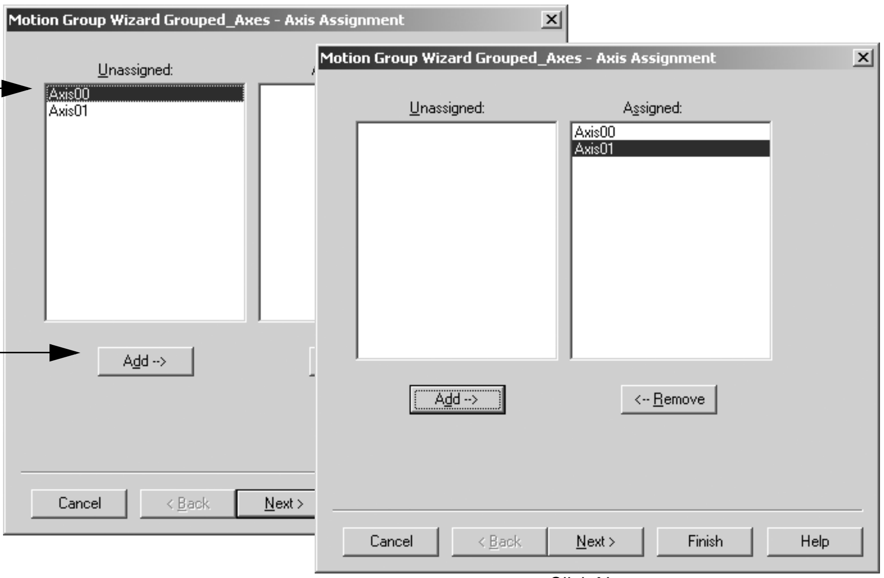
Scope: Motion_Drive(controller)

Style:

Click Configure.

Move each axis to the Assigned side by clicking on the axis and then clicking Add.

Move both axes.



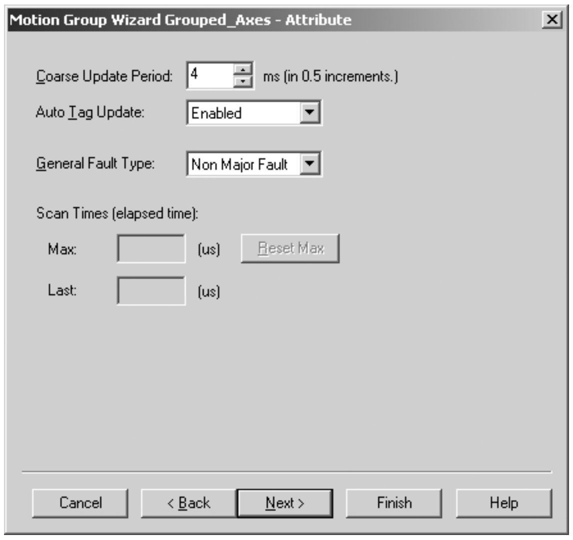
Unassigned: Axis00, Axis01

Motion Group Wizard Grouped_Axes - Axis Assignment

Unassigned:

Assigned: Axis00, Axis01

Click Next.



Coarse Update Period: 4 ms (in 0.5 increments.)

Auto Tag Update: Enabled

General Fault Type: Non Major Fault

Scan Times (elapsed time):

Max: (us)

Last: (us)

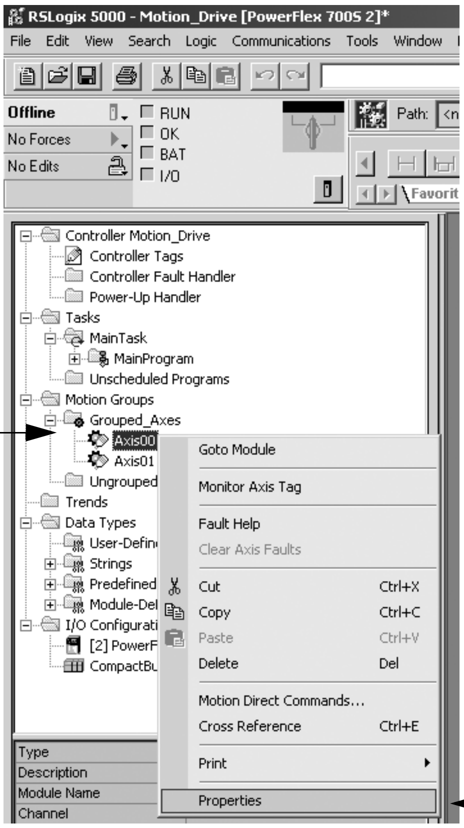
Set the Coarse Update Period to 4 ms, this is the minimum time usable with DriveLogix motion.

Set the Auto Tag Update to Enabled.

Set the General Fault Type to Non Major Fault.

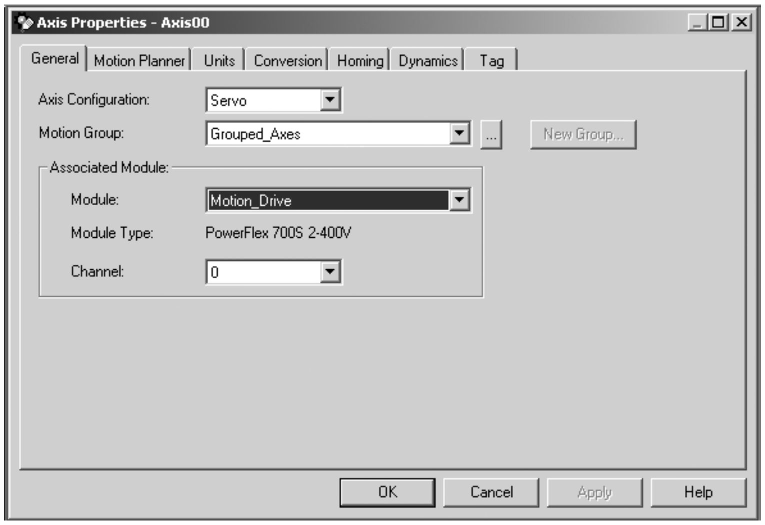
Click Finish.

8. Configure Axis00.



Right click on the axis you want to configure (Axis00 in this example).

Select Properties.



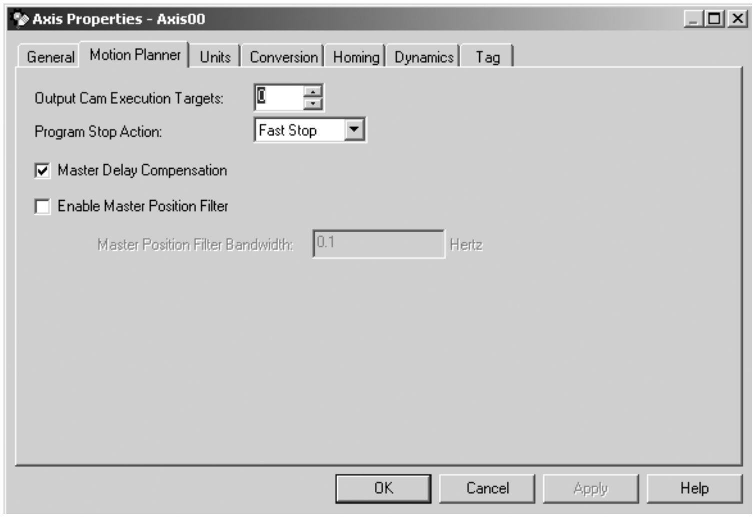
Axis Configuration: Servo
Motion Group: Grouped_Axes
Associated Module:
Module: Motion_Drive
Module Type: PowerFlex 700S 2-400V
Channel: 0

On the General tab:

Associate the module with the drive. Match the Module name with the drive's name in the I/O Configuration.

Set the Channel to the channel being used for the encoder. Channel 0 must be selected for the Servo axis and Channel 1 can be used only for feedback.

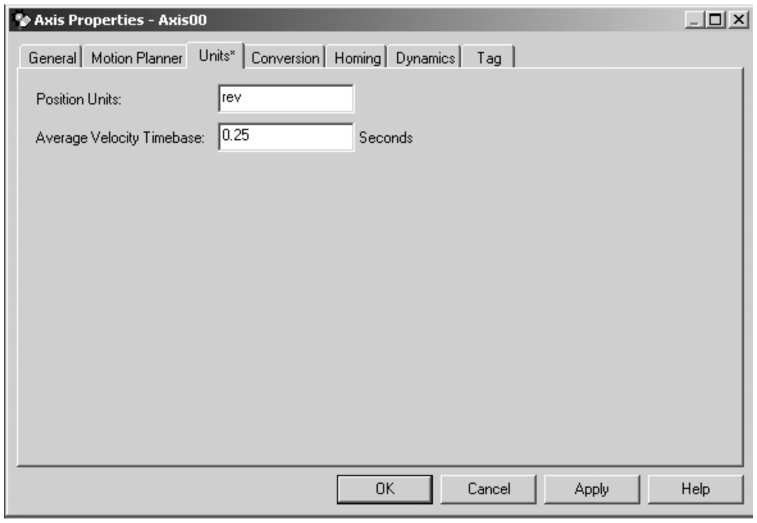
8. Configure Axis00 continued...



Output Cam Execution Targets: 0
Program Stop Action: Fast Stop
[✓] Master Delay Compensation
[ ] Enable Master Position Filter
Master Position Filter Bandwidth: 0.1 Hertz

On the Motion Planner tab:

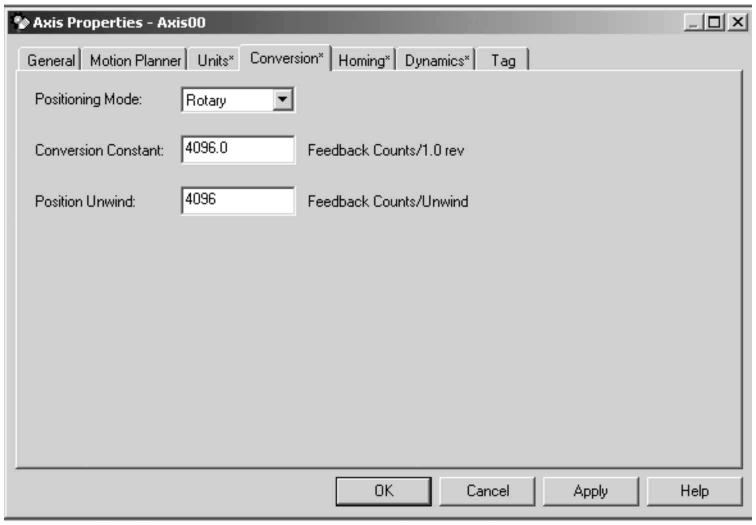
Determine how many Output Cam execution nodes (instances) are created for a specific axis The value specified for Execution Target in the MAOC instruction references a specific instance in which a value of zero selects the first instance.



Position Units: rev
Average Velocity Timebase: 0.25 Seconds

On the Units tab:

Define the Position Units. In this example the Position Units are revs. Position Units can be almost anything (e.g. degrees, radians, pallets, widgets etc. ...)
Define Average Velocity Timebase is the sample rate that is used for the Average Velocity tag in the controller tags.



Positioning Mode: Rotary
Conversion Constant: 4096.0 Feedback Counts/1.0 rev
Position Unwind: 4096 Feedback Counts/Unwind

On the Conversion tab:

Setup Positioning Mode for Linear or Rotary.

The Conversion Constant is the number of feedback counts per Position Unit.

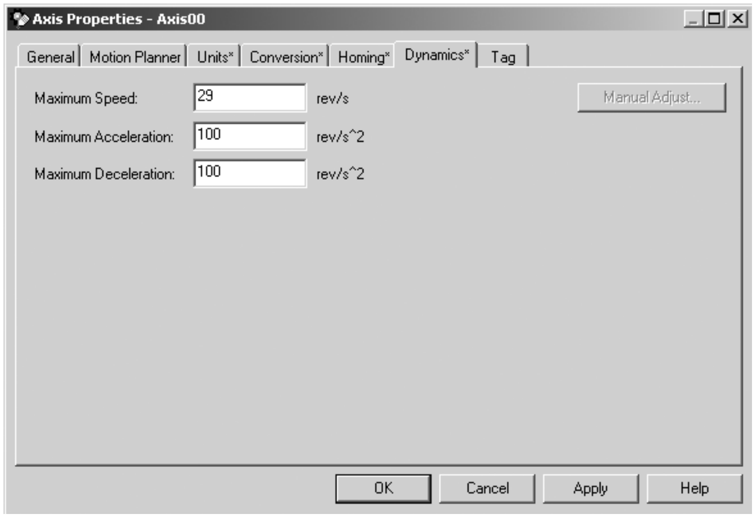
8. Configure Axis00 continued...

On the Dynamics tab:

Define the limits for speed, acceleration and deceleration.

Important: Do not exceed the system dynamics.

Click Apply and OK when you have completely defined the axis.

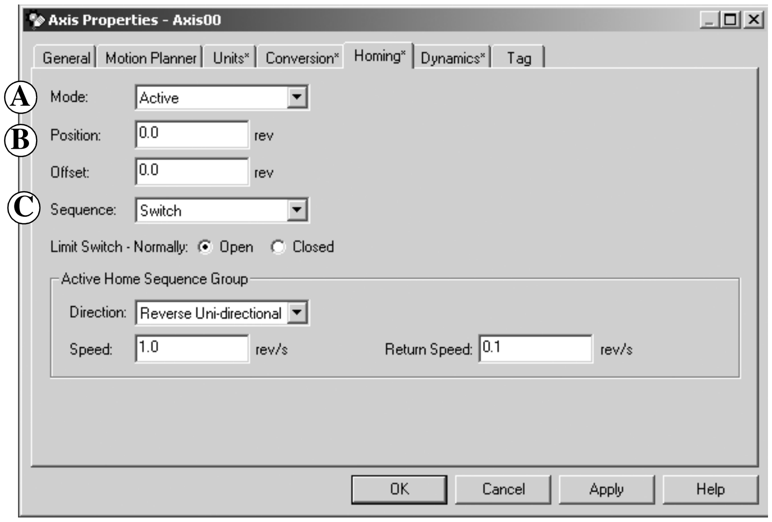


Maximum Speed: 29 rev/s
Maximum Acceleration: 100 rev/s^2
Maximum Deceleration: 100 rev/s^2

On the Homing tab:

.Setup the Homing Mode, Position, Offset and Sequence.

Refer to the tables at the right.



Mode: Active
Position: 0.0 rev
Offset: 0.0 rev
Sequence: Switch
Limit Switch - Normally: [✓ Open] Closed
Active Home Sequence Group
Direction: Reverse Uni-directional
Speed: 1.0 rev/s Return Speed: 0.1 rev/s

A	Homing Mode	Active - the desired homing sequence is selected by specifying whether a home limit switch and/or the encoder marker are used for this axis. Active homing sequences always use the trapezoidal velocity profile. Passive - homing redefines the absolute position of the axis on the occurrence of a home switch or encoder marker event. Passive homing is most commonly used to calibrate uncontrolled axes, although it can also be used with controlled axes to create a custom homing sequence. Passive homing, for a given home sequence, works similar to the corresponding active homing sequence, except that no motion is commanded; the controller just waits for the switch and marker events to occur.
B	Homing Position	Type the desired absolute position, in position units, for the axis after the specified homing sequence has been completed. In most cases, this position will be set to zero, although any value within the software travel limits can be used. After the homing sequence is complete, the axis is left in this position. If the Positioning Mode (set in the Conversion tab) of the axis is Linear, then the home position should be within the travel limits, if enabled. If the Positioning Mode is Rotary, then the home position should be less than the unwind distance in position units.
C	Offset	Type the desired offset (if any) in position units the axis is to move, upon completion of the homing sequence, to reach the home position. In most cases, this value will be zero.
D	Homing Sequence	Select the event that will cause the Home Position to be set: Sequence Type — Description Immediate — Sets the Home Position to the present actual position, without motion. Switch — Sets the Home Position when axis motion encounters a home limit switch. Marker — Sets the Home Position when axis encounters an encoder marker. Switch-Marker — Sets the Home Position when axis first encounters a home limit switch, then encounters an encoder marker.

Limit Switch	If a limit switch is used, indicate the normal state of that switch (i.e., before being engaged by the axis during the homing sequence): Normally Open Normally Closed	
Direction	For active homing sequences, except for the Immediate Sequence type, select the desired homing direction:	
	Forward Uni-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
	Forward Bi-directional	The axis jogs in the positive axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).
	Reverse Uni-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then continues in the same direction until axis motion stops (after decelerating or moving the Offset distance).
	Reverse Bi-directional	The axis jogs in the negative axial direction until a homing event (switch or marker) is encountered, then reverses direction until motion stops (after decelerating or moving the Offset distance).

9. Configure Axis01 by repeating the tasks in Step 8.

Important: Only Channel 0 will function for a Servo axis.
Channel 1 may be used for a Feedback Only axis.

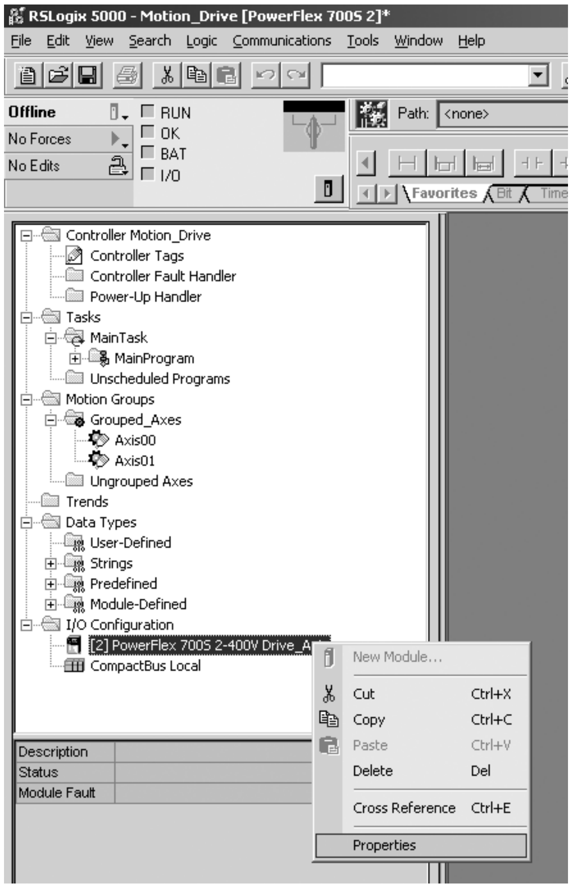
10. Save the RSLogix 5000 project and download it to the controller.

ATTENTION: Running the system without proper tuning can cause unstable and unpredictable operation. To avoid potential personal injury and damage to machinery, determine the proper values for system dynamics and tune the system before beginning operation.

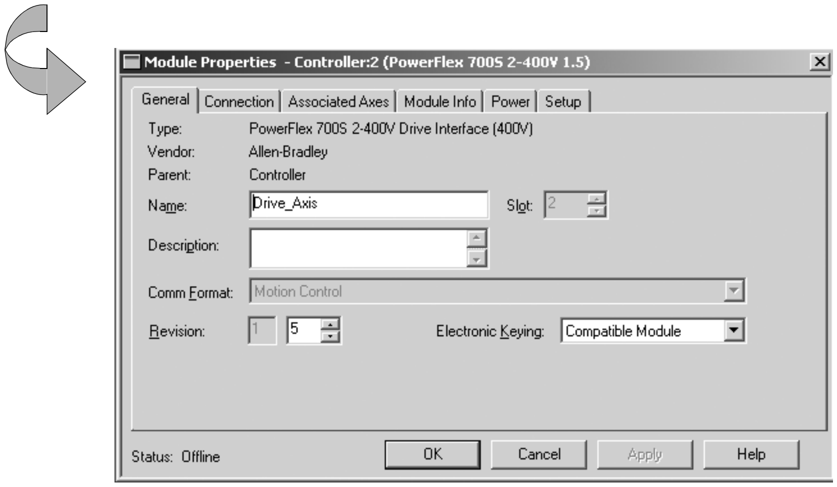
Launching DriveExecutive from RSLogix

Next open the drive properties window and launch DriveExecutive drive programming software.

1. In the Controller Organizer, select the PowerFlex 700S drive. Right-click the drive module and select Properties



That will launch the Module Properties Window for the drive.


Module Properties - Controller:2 (PowerFlex 700S 2-400V 1.5)

General | Connection | Associated Axes | Module Info | Power | Setup

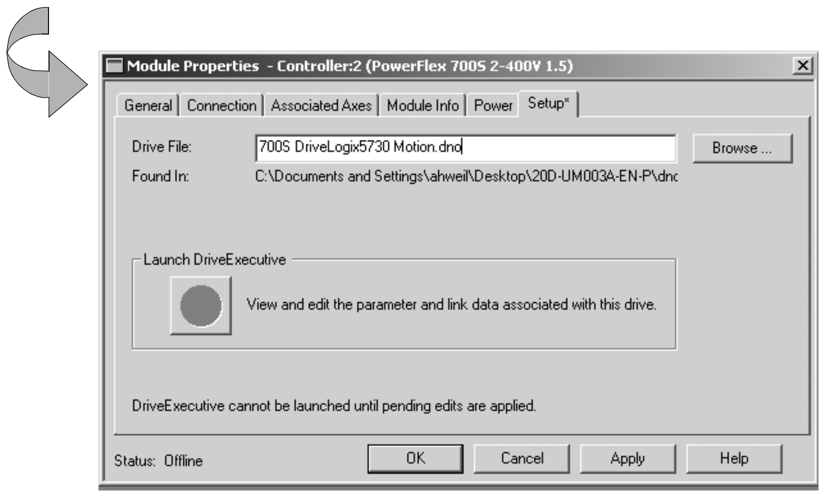
Type: PowerFlex 700S 2-400V Drive Interface (400V)
Vendor: Allen-Bradley
Parent: Controller
Name: Drive_Axis
Slot: 2
Description:
Comm Format: Motion Control
Revision: 1 5
Electronic Keying: Compatible Module

Status: Offline

Click on the Setup tab.

2. Type the desired DriveExecutive filename or browse for an existing one.

Then click Apply.



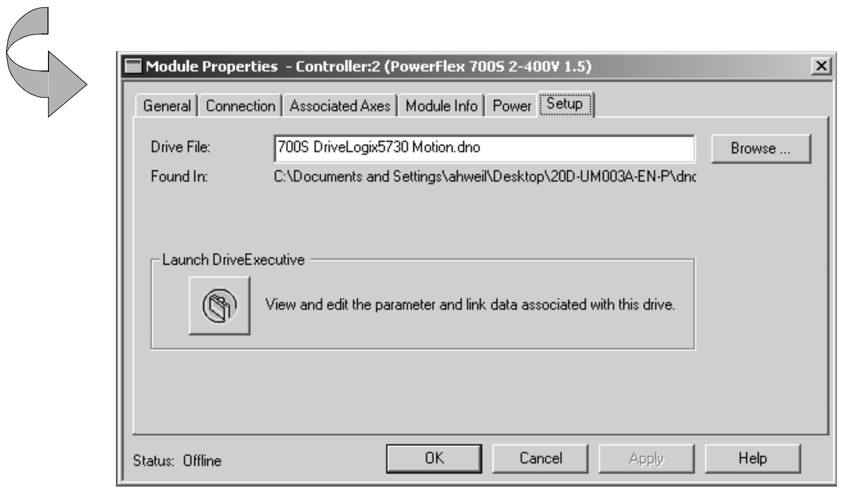
Drive File: 700S DriveLogix5730 Motion.dno

Found In: C:\Documents and Settings\ahweil\Desktop\20D-UM003A-EN-P\dnc

View and edit the parameter and link data associated with this drive.

DriveExecutive cannot be launched until pending edits are applied.

Status: Offline



Drive File: 700S DriveLogix5730 Motion.dno

Found In: C:\Documents and Settings\ahweil\Desktop\20D-UM003A-EN-P\dnc

View and edit the parameter and link data associated with this drive.

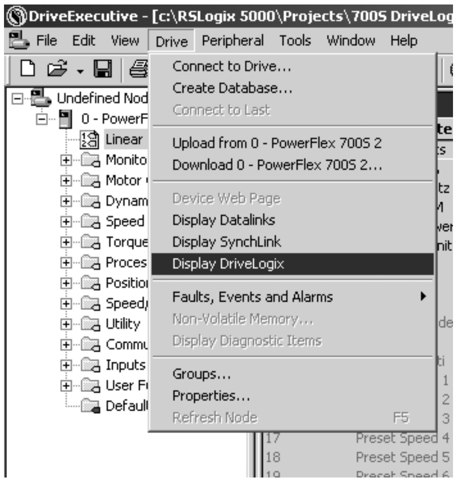
Status: Offline

3. Click the Launch DriveExecutive button to launch DriveExecutive Lite.

Configuring the Drive with DriveExecutive Software

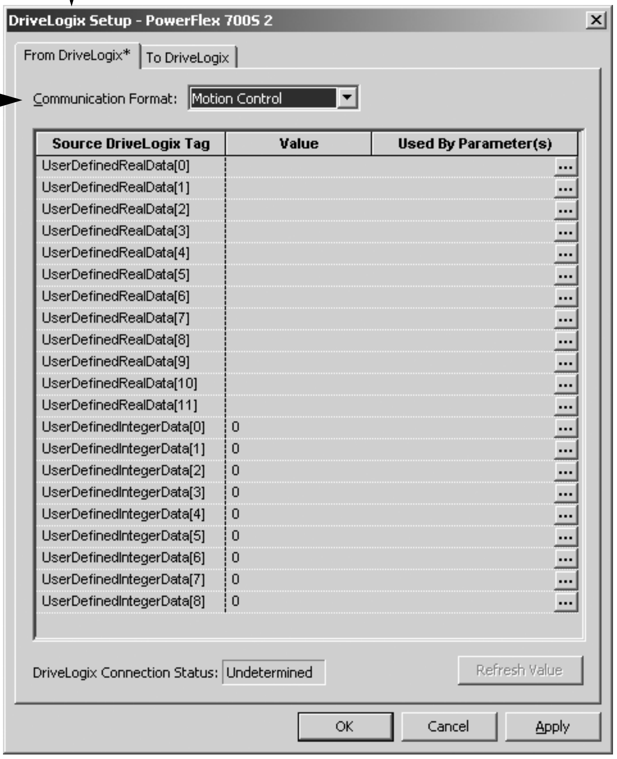
In DriveExecutive software, connect to the drive and access the Display DriveLogix dialog as shown below:

1. From the Drive Menu, select Display DriveLogix.



2. Click on the From DriveLogix tab.

3. Verify the Communication Format is set to Motion Control.



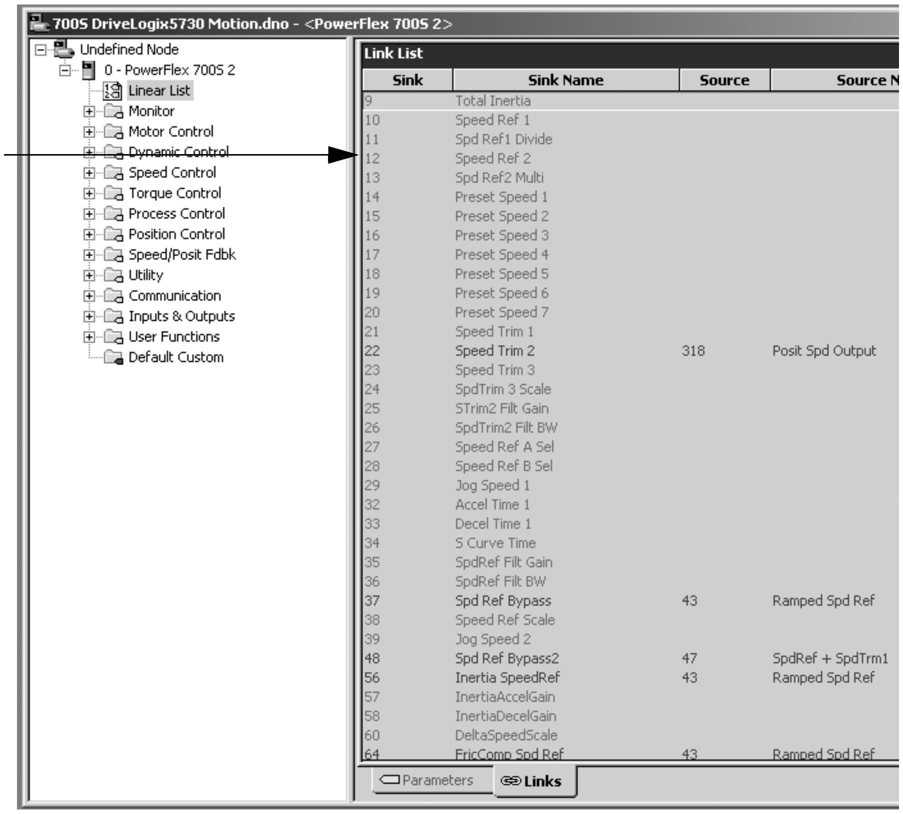
Communication Format: Motion Control

Source DriveLogix Tag	Value	Used By Parameter(s)
UserDefinedRealData[0]		...
UserDefinedRealData[1]		...
UserDefinedRealData[2]		...
UserDefinedRealData[3]		...
UserDefinedRealData[4]		...
UserDefinedRealData[5]		...
UserDefinedRealData[6]		...
UserDefinedRealData[7]		...
UserDefinedRealData[8]		...
UserDefinedRealData[9]		...
UserDefinedRealData[10]		...
UserDefinedRealData[11]		...
UserDefinedIntegerData[0]	0	...
UserDefinedIntegerData[1]	0	...
UserDefinedIntegerData[2]	0	...
UserDefinedIntegerData[3]	0	...
UserDefinedIntegerData[4]	0	...
UserDefinedIntegerData[5]	0	...
UserDefinedIntegerData[6]	0	...
UserDefinedIntegerData[7]	0	...
UserDefinedIntegerData[8]	0	...

DriveLogix Connection Status: Undetermined

Click OK to apply and close the DriveLogix Setup Dialog.

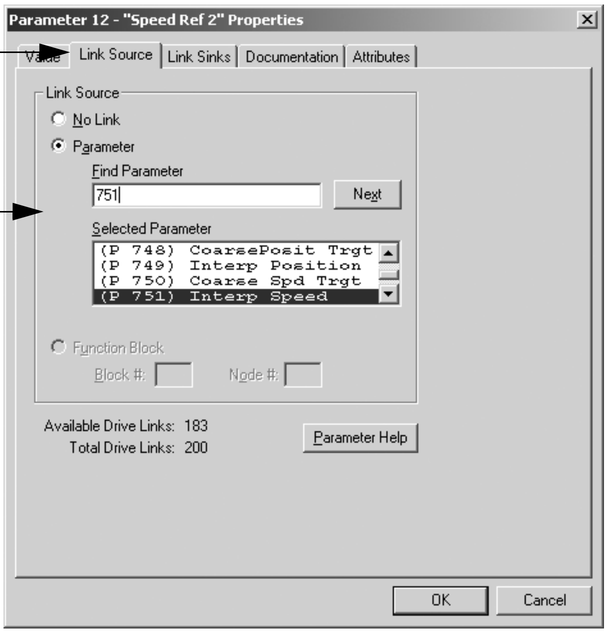
Next link the appropriate parameters to the words being produced and consumed by the controller.



7005 DriveLogix5730 Motion.dno - <PowerFlex 700S 2>

Sink	Sink Name	Source	Source N
9	Total Inertia		
10	Speed Ref 1		
11	Spd Ref1 Divide		
12	Speed Ref 2		
13	Spd Ref2 Multi		
14	Preset Speed 1		
15	Preset Speed 2		
16	Preset Speed 3		
17	Preset Speed 4		
18	Preset Speed 5		
19	Preset Speed 6		
20	Preset Speed 7		
21	Speed Trim 1		
22	Speed Trim 2	318	Posit Spd Output
23	Speed Trim 3		
24	SpdTrim 3 Scale		
25	STrim2 Filt Gain		
26	SpdTrim2 Filt BW		
27	Speed Ref A Sel		
28	Speed Ref B Sel		
29	Jog Speed 1		
32	Accel Time 1		
33	Decel Time 1		
34	S Curve Time		
35	SpdRef Filt Gain		
36	SpdRef Filt BW		
37	Spd Ref Bypass	43	Ramped Spd Ref
38	Speed Ref Scale		
39	Jog Speed 2		
48	Spd Ref Bypass2	47	SpdRef + SpdTrm1
56	Inertia SpeedRef	43	Ramped Spd Ref
57	InertiaAccelGain		
58	InertiaDecelGain		
60	DeltaSpeedScale		
64	FricComp Spd Ref	43	Ramped Spd Ref

4. On the Links view, double-click on the desired Sink parameter.



Parameter 12 - "Speed Ref 2" Properties

Link Source: ( ) No Link [✓ Parameter]

Find Parameter: 751

Selected Parameter:
```
(P 748) CoarsePosit Trgt
(P 749) Interp Position
(P 750) Coarse Spd Trgt
(P 751) Interp Speed
```

Available Drive Links: 183
Total Drive Links: 200

5. Click on the Link Source tab.

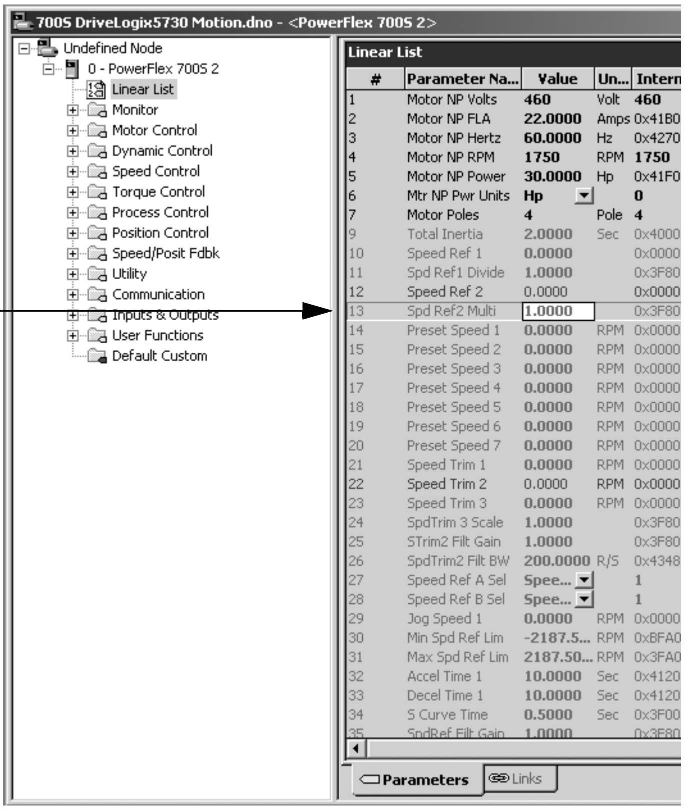
6. Type or select the desired Source parameter.

Create the links in the table below:

Destination (Sink) Parameter	Source Parameter	Description
12 [Speed Ref 2]	751 [Interp Speed Ref]	Speed reference from the Coarse to Fine Interpolator, to Speed Ref 2
22 [Speed Trim 2]	318[Posit Spd Output]	This is a default link. It connects the position regulator in the drive to the velocity loop.
23 [Speed Trim 3]	55[Spd Comp]	This setup is for speed compensation to reduce position error.
626 [Integer Out00]	155[Logic Status]	The status of the drive is sent to the DriveLogix.
693[Intrep SyncInput]	700 [Motn Posit Sync]	The drive receives the synchronization pulse from the DriveLogix. This keeps the interpolators synchronized.
748 [CoarsePosit Trgt]	698 [Motn Posit Cmmd]	The position interpolator of the drive receives the coarse position target from the DriveLogix Motion Interpolator
750 [Coarse Spd Trgt]	699 [Motn Speed Cmmd]	The Speed loop of the drive receives the coarse velocity target from the DriveLogix Motion Interpolator

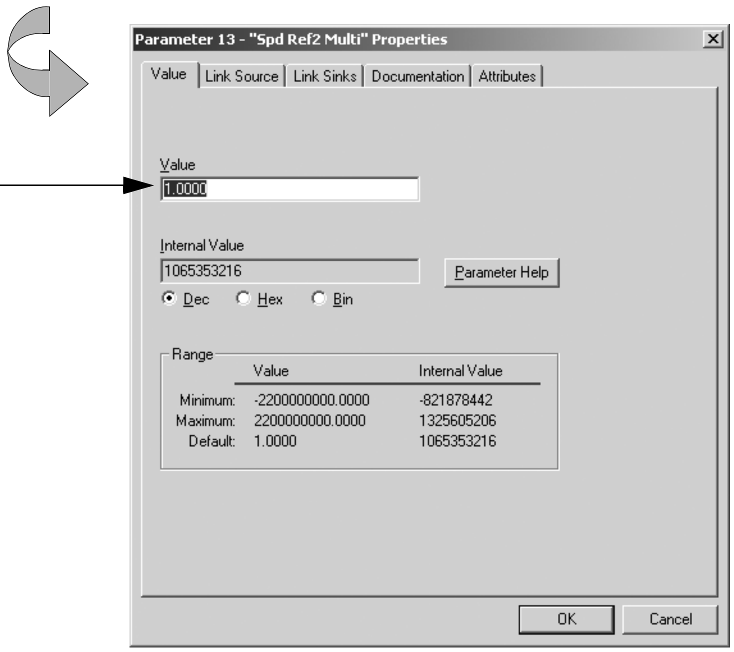
Next, make the appropriate parameter settings.

7. On the Parameters view, double-click on the desired parameter.



#	Parameter Na...	Value	Un...	Intern
1	Motor NP Volts	460	Volt	460
2	Motor NP FLA	22.0000	Amps	0x41B0
3	Motor NP Hertz	60.0000	Hz	0x4270
4	Motor NP RPM	1750	RPM	1750
5	Motor NP Power	30.0000	Hp	0x41F0
6	Mtr NP Pwr Units	Hp		0
7	Motor Poles	4	Pole	4
9	Total Inertia	2.0000	Sec	0x4000
10	Speed Ref 1	0.0000		0x0000
11	Spd Ref1 Divide	1.0000		0x3F80
12	Speed Ref 2	0.0000		0x0000
13	Spd Ref2 Multi	1.0000		0x3F80
14	Preset Speed 1	0.0000	RPM	0x0000
15	Preset Speed 2	0.0000	RPM	0x0000
16	Preset Speed 3	0.0000	RPM	0x0000
17	Preset Speed 4	0.0000	RPM	0x0000
18	Preset Speed 5	0.0000	RPM	0x0000
19	Preset Speed 6	0.0000	RPM	0x0000
20	Preset Speed 7	0.0000	RPM	0x0000
21	Speed Trim 1	0.0000	RPM	0x0000
22	Speed Trim 2	0.0000	RPM	0x0000
23	Speed Trim 3	0.0000	RPM	0x0000
24	SpdTrim 3 Scale	1.0000		0x3F80
25	STrim2 Filt Gain	1.0000		0x3F80
26	SpdTrim2 Filt BW	200.0000	R/S	0x4348
27	Speed Ref A Sel	Spee...		1
28	Speed Ref B Sel	Spee...		1
29	Jog Speed 1	0.0000	RPM	0x0000
30	Min Spd Ref Lim	-2187.5...	RPM	0xBFA0
31	Max Spd Ref Lim	2187.50...	RPM	0x3FA0
32	Accel Time 1	10.0000	Sec	0x4120
33	Decel Time 1	10.0000	Sec	0x4120
34	S Curve Time	0.5000	Sec	0x3F00
35	SpdRef Filt Gain	1.0000		0x3F80

8. Type the desired value.



Value: 1.0000

Internal Value: 1065353216

Range

	Value	Internal Value
Minimum:	-2200000000.0000	-821878442
Maximum:	2200000000.0000	1325605206
Default:	1.0000	1065353216

Make the parameter settings shown in the table below:

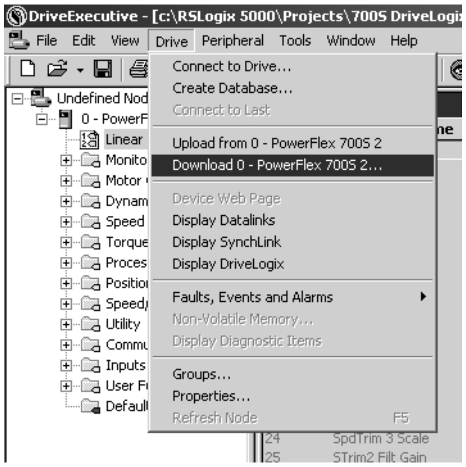
Parameter	Value	Description
13 [Spd Ref2 Multi]	1	Speed Ref 2 scale value
24 [Spd Trim 3 Scale]	0.003	Speed Trim 3 scale value
27 [Speed Ref Sel]	2	Select Speed Ref 2 as the speed reference

151 [Logic Command]	Bit0=1
	Bit10=1
	Bit13 = 1
		Bit 0 bypasses the speed ramp in the drive
		Bit 10 enables Inertia Comp
		Bit 13 enables the position loop within the drive.

222 [Motor Fdbk Sel]	0 - 6	Select the Motor Speed Feedback being used. This is also the feedback for the Servo Axis in DriveLogix Motion Group
600 [Lgx Comm Format]	19	This selects the format of the commands coming from DriveLogix
740 [Position Control]	Bit 1 = 1 Bit 6 = 0 Bit 2 = 1 Bit 8 = 0	This sets the position regulator to work with the DriveLogix configuration and enables the Integral part of the regulator
742 [Posit Ref Sel]	0	This configures the drive position loop to receive position commands from DriveLogix via the Interpolator
686 [Motn Config]	Bits 0 - 2	Bit 0 = Change the polarity of the feedback only feedback axis Bit 1 = Enable Software OverTravel Limits Bit 2 = Enable Hardware Overtravel Limits
701 [FdbkAxis FdbkSel]	1 – 10	This selects the feedback channel used for the “Feedback Only axis” in the Motion Group in DriveLogix. Encoders, Optional Feedback, and SynchLink selections. See User’s manual for details.
904 [SL Node Cnfg]	Bit 0 =1	This sets up the SynchLink as the time keeper. This is used to synchronize the Drive and DriveLogix.
146 [FW Task Time Sel]	0	Firmware Task Time Selection.
147 [FW Functions Enable]	Bit 16 =1	Position Control Enabled.

Downloading the Settings and Links to the Drive

1. From the Drive Menu, select Display Download.



2. Using the resulting windows and RSLinx navigate to the drive and download the settings and links.

Additional Testing and Programming is Necessary

The steps in this chapter provide the minimum settings required to begin testing DriveLogix motion.

ATTENTION: Running the system without proper tuning can cause unstable and unpredictable operation. To avoid potential personal injury and damage to machinery, determine the proper values for system dynamics and tune the system before beginning operation.

You must perform testing to determine the actual system dynamics and tune the drive.

In addition, you must create ladder logic that uses the Logix Motion Instructions. Refer to Supported Motion Commands on page 4-18 and publication 1756-RM007D, *Reference Manual - Logix Controller Motion Instruction Set*.

ATTENTION: There is no default Position Error Fault logic in this system. To avoid potential personal injury and damage to machinery, detect Position Error faults, by using parameter links and ladder logic.

Supported Motion Commands

The following Logix Motion Instructions are supported by the DriveLogix controller:

Motion State

- MSO (Motion Servo On)
- MSF (Motion Servo Off)
- MASD (Motion Axis Shutdown)
- MASR (Motion Axis Shutdown Reset)
- MAFR (Motion Axis Fault Reset)

Motion Move

- MAJ (Motion Axis Jog)
- MAM (Motion Axis Move)
- MAS (Motion Axis Stop)
- MAH (Motion Axis Home)
- MAG (Motion Axis Gearing)
- MCD (Motion Change Dynamics)
- MRP (Motion Redefine Position)
- MCCP (Motion Calculate Position Profile)
- MAPC (Motion Axis Position Cam)
- MATC (Motion Axis Time Cam)

Motion Event

- MAW (Motion Arm Watch)
- MDW (Motion Disarm Watch)
- MAR (Motion Arm Registration)
- MDR (Motion Disarm Registration)
- MAOC (Motion Arm Output Cam)
- MDOC (Motion Disarm Output Cam)

Motion Group

- MGS (Motion Group Stop)
- MGSD (Motion Group Shutdown)
- MGSR (Motion Group Shutdown Reset)
- MGSP (Motion Group Strobe Position)

Communicating with Devices on a Serial Link

De-energizing the Drive to Connect or Disconnect a Cable

⚠ **ATTENTION:** Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of configuring serial communication you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive's input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with configuring serial communication.

Configuring Your System for a Serial Link

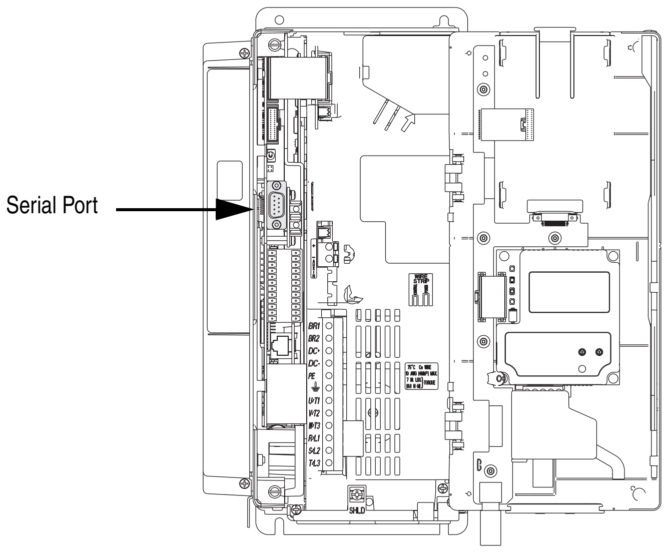
For the DriveLogix controller to operate on a serial network, you need:

- a workstation with a serial port
- RSLinx software to configure the serial communication driver
- RSLogix 5000 programming software to configure the serial port of the controller

Important: Limit the length of serial (RS-232) cables to 15.2m (50 ft.).

Configuring the Hardware

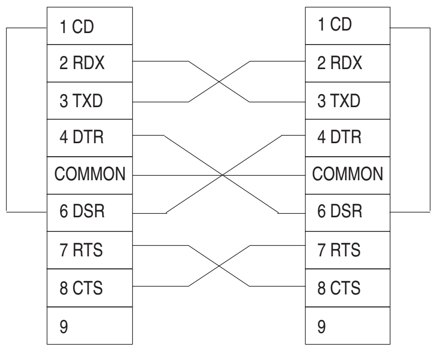
The RS-232 port is an isolated serial port built-in to the front of the Main Control Board. Refer to Opening Door Over Power Structure and Main Control Board on page B-2.



To connect to the serial port:

1. Select the appropriate cable.

The 1756-CP3 cable attaches the controller directly to the controller.



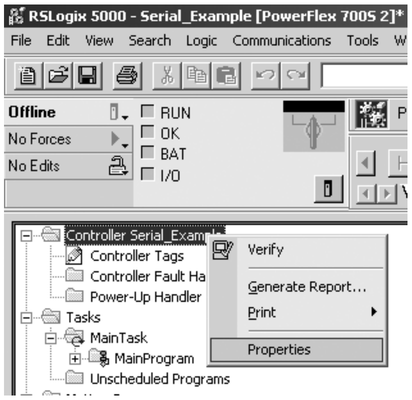
If you make your own cable, it must be shielded and the shields must be tied to the metal shell (that surrounds the pins) on both ends of the cable.

You can also use a 1747-CP3 cable (from the SLC product family). This cable has a taller right-angle connector housing than the 1756-CP3 cable.

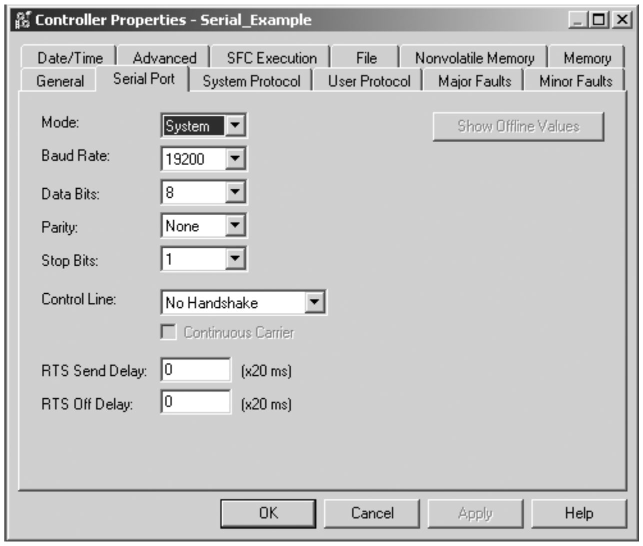
2. Connect the cable to the serial port on the controller.

Configuring the Serial Port of the Controller

1. In the Controller Organizer, select the Controller folder. Right-click the selected folder and select Properties.



2. On the Serial tab, specify serial port characteristics.

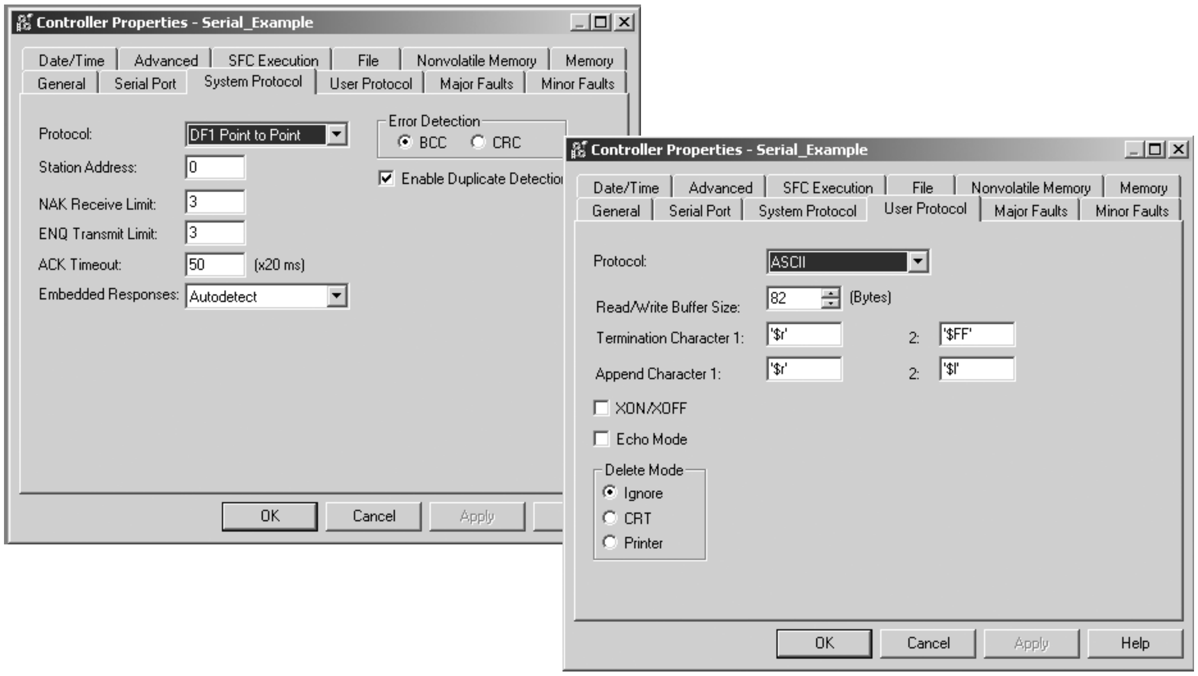


Mode: System
Baud Rate: 19200
Data Bits: 8
Parity: None
Stop Bits: 1
Control Line: No Handshake
RTS Send Delay: 0 (x20 ms)
RTS Off Delay: 0 (x20 ms)

Characteristic:	Description (default is shown in bold):
Mode	Select **System** (for DF1 communication) or User mode (for ASCII communication).
Baud rate	Specifies the communication rate for the serial port. Select a baud rate that all devices in your system support. Select 110, 300 600, 1200, 2400, 4800, 9600, **19200**, or 38400 Kbps.
Parity	Specifies the parity setting for the serial port. Parity provides additional message-packet error detection. Select **None** or Even.
Data bits	Specifies the number of bits per message packet. Select **8**.
Stop bits	Specifies the number of stop bits to the device with which the controller is communicating. Select **1** or 2.
Control line	Specifies the mode in which the serial driver operates. Select **No Handshake**, Full-Duplex, Half-Duplex with Continuous Carrier, or Half-Duplex without Continuous Carrier. If you are not using a modem, select No Handshake If both modems in a point-to-point link are full-duplex, select Full-Duplex for both controllers.

3. On the System Protocol tab, select the appropriate DF1 communication mode for point-to-point or master/slave communications.

   Or use the User Protocol tab to specify ASCII protocol to communicate to an ASCII device.



Protocol: DF1 Point to Point
Station Address: 0
NAK Receive Limit: 3
ENQ Transmit Limit: 3
ACK Timeout: 50 (x20 ms)
Embedded Responses: Autodetect
Error Detection: [✓ BCC] CRC
[✓] Enable Duplicate Detection

Protocol: ASCII
Read/Write Buffer Size: 82 (Bytes)
Termination Character 1: '$r' 2: '$FF'
Append Character 1: '$r' 2: '$l'
[ ] XON/XOFF
[ ] Echo Mode
Delete Mode: [✓ Ignore] CRT Printer

Use this mode:	For:	See page:
DF1 point-to-point	communication between the controller and one other DF1-protocol-compatible device. This is the default system mode. This mode is typically used to program the controller through its serial port.	5-6
DF1 master mode	control of polling and message transmission between the master and slave nodes. The master/slave network includes one controller configured as the master node and as many as 254 slave nodes. Link slave nodes using modems or line drivers. A master/slave network can have node numbers from 0-254. Each node must have a unique node address. Also, at least 2 nodes must exist to define your link as a network (1 master and 1 slave station are the two nodes).	5-9
DF1 slave mode	using a controller as a slave station in a master/slave serial communication network. When there are multiple slave stations on the network, link slave stations using modems or line drivers. When you have a single slave station on the network, you do not need a modem to connect the slave station to the master; you can configure the control parameters for no handshaking. You can connect 2-255 nodes to a single link. In DF1 slave mode, a controller uses DF1 half-duplex protocol. One node is designated as the master and it controls who has access to the link. All the other nodes are slave stations and must wait for permission from the master before transmitting.	5-8
User mode	communicating with ASCII devices This requires your program logic to use the ASCII instructions to read and write data from and to an ASCII device.	5-12

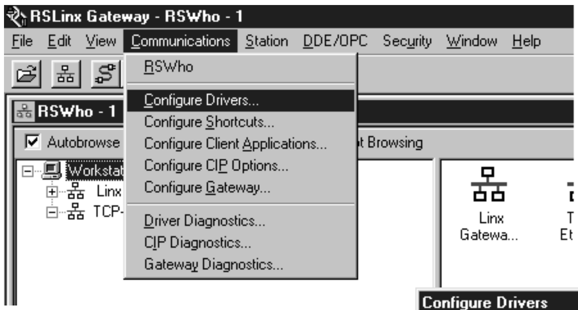
Important: Half-Duplex settings do not work as with other Logix controllers. RTS and CTS are not functional.

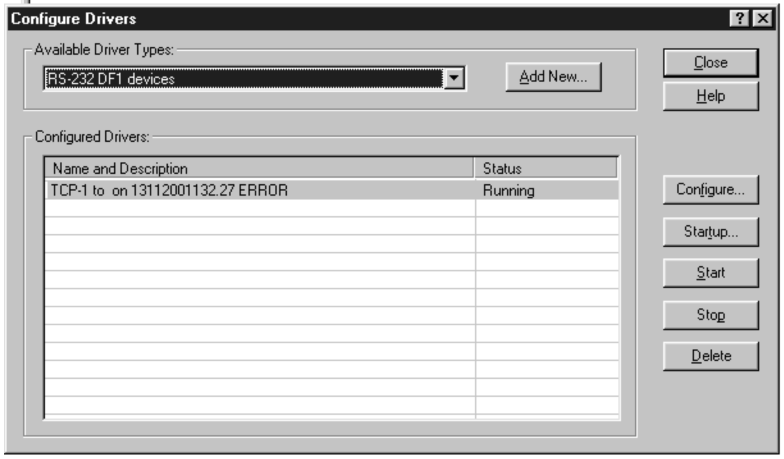
Configuring the Communication Driver

Use RSLinx software to configure the serial communication driver. Select the “DF1” driver.

1. In the Communications menu, select the Configure Driver.

   From the Available Driver Type list select the DF1 Driver, then click Configure.



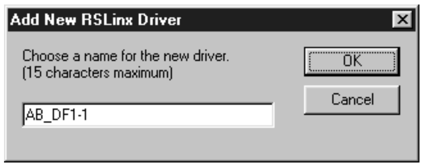


Available Driver Types: RS-232 DF1 devices

Configured Drivers:

Name and Description	Status
TCP-1 to on 13112001132.27 ERROR	Running

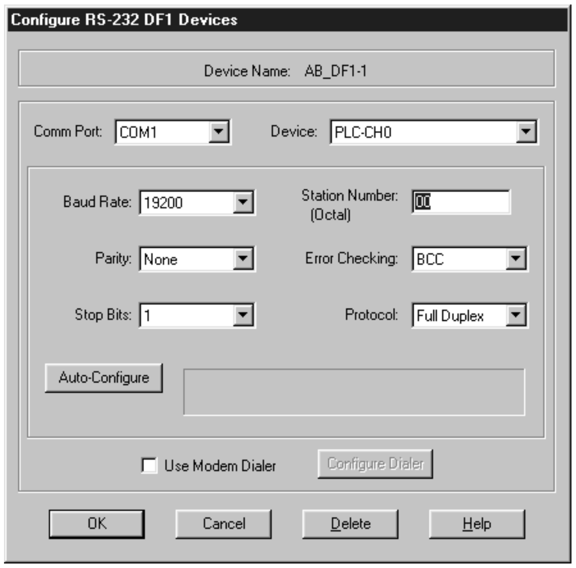
2. Specify a name for the driver and click OK.



Choose a name for the new driver.
(15 characters maximum)

AB_DF1-1

3. Specify the appropriate communication settings.



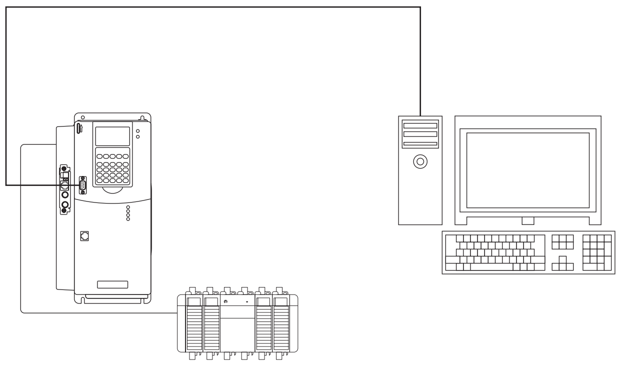
Device Name: AB_DF1-1

Comm Port:	COM1	Device:	PLC-CH0
Baud Rate:	19200	Station Number: (Octal)	00
Parity:	None	Error Checking:	BCC
Stop Bits:	1	Protocol:	Full Duplex

☐ Use Modem Dialer

Example 1: Workstation Directly Connected to a DriveLogix Controller

In the following example, a workstation directly connects to a DriveLogi5730 controller over a serial link. This is useful for downloading a controller project directly to the controller.



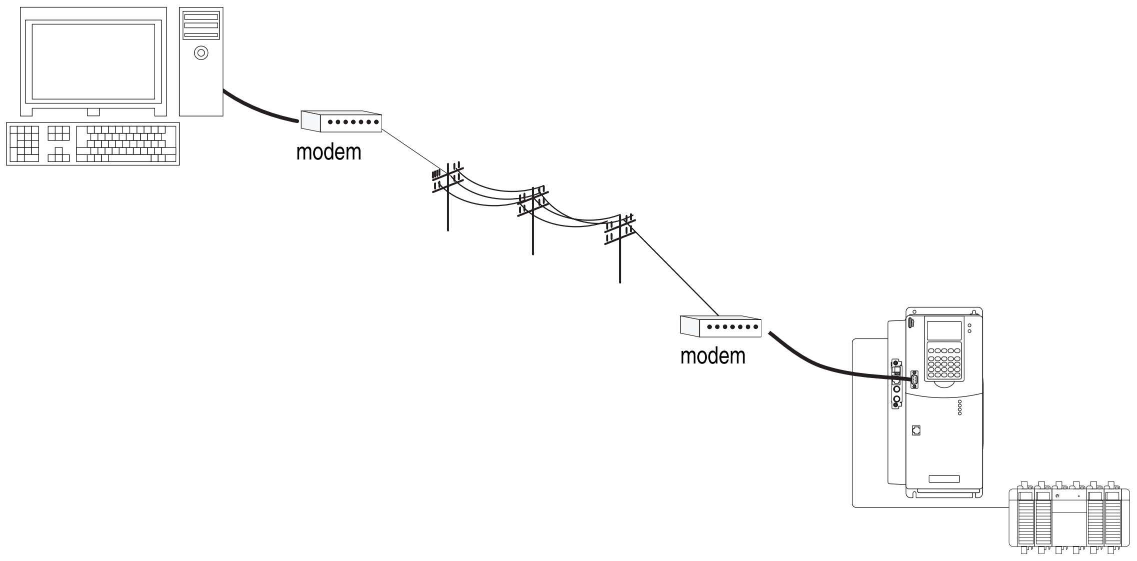
Use RSLogix 5000 programming software to configure the controller's serial port for the DF1 point-to-point (full-duplex) protocol. This type of protocol supports simultaneous transmission between two devices in both directions. The DF1 point-to-point protocol controls message flow, detects and signals errors, and retries if errors are detected.

Configuring a DF1 Point-to-Point Station

This field:	Description:
Station address	The station address for the serial port on the DF1 point-to-point network. Enter a valid DF1 address (0-254). Address 255 is reserved for broadcast messages. The default is 0.
NAK receive limit	Specifies the number of NAKs the controller can receive in response to a message transmission. Enter a value 0-127. The default is 3.
ENQ transmit limit	Specifies the number of inquiries (ENQs) you want the controller to send after an ACK timeout. Enter a value 0-127. The default is 3.
ACK timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0-32767. Limits are defined in 20ms intervals. The default is 50 (1000ms).
Embedded response	Specifies how to enable embedded responses. Select Autodetect (enabled only after receiving one embedded response) or Enabled. The default is Autodetect.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

Example 2: Workstation Remotely Connected to a DriveLogix Controller

In the following example, a workstation remotely connects to a DriveLogix controller over s serial link. A modem is connected to the controller to provide remote access.



If you use a modem to remotely connect the controller to one workstation, use RSLogix 5000 programming software to configure the serial port of the controller for the DF1 point-to-point (full-duplex) protocol, as in the previous example. If the controller is part of a master/slave serial network, configure the serial port of the controller for either the DF1 master or DF1 slave protocol (both half-duplex).

Master/Slave Communication Methods

A master station can communicate with a slave station in two ways:

Name:	This method:	Benefits:
standard communication mode	Initiates polling packets to slave stations according to their position in the polling array(s). Polling packets are formed based on the contents of the normal poll array and the priority poll array.	This communication method is most often used for point-to-multipoint configurations. This method provides these capabilities: • slave stations can send messages to the master station (polled report-by-exception) • slave stations can send messages to each other via the master • master maintains an active station array The poll array resides in a user-designated data file. You can configure the master: • to send messages during its turn in the poll array *or* • for between-station polls (master transmits any message that it needs to send before polling the next slave station) In either case, configure the master to receive multiple messages or a single message per scan from each slave station.
message-based communication mode	initiates communication to slave stations using only user-programmed message (MSG) instructions. Each request for data from a slave station must be programmed via a MSG instruction. The master polls the slave station for a reply to the message after waiting a user-configured period of time. The waiting period gives the slave station time to formulate a reply and prepare the reply for transmission. After all of the messages in the master's message-out queue are transmitted, the slave-to-slave queue is checked for messages to send.	If your application uses satellite transmission or public switched-telephone-network transmission, consider choosing message-based communication. Communication to a slave station can be initiated on an as-needed basis. Also choose this method if you need to communicate with non-intelligent remote terminal units (RTUs).

Configuring a DF1 Slave Station

This field:	Description:
Station address	The station address for the serial port on the DF1 slave. Enter a valid DF1 address (0-254). Address 255 is reserved for broadcast messages. The default is 0.
Transmit retries	The number of times the remote station retries a message after the first attempt before the station declares the message undeliverable. Enter a value 0-127. The default is 3.
Slave poll timeout	Specifies the amount of time the slave station waits to be polled by a master before indicating a fault. Enter a value 0-32767. Limits are defined in 20ms intervals. The default is 3000 (60,000ms).
EOT suppression	Select whether or not to suppress sending EOT packets in response to a poll. The default is not to suppress sending EOT packets.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

Configuring a DF1 Master Station

This field:	Description:
Station address	The station address for the serial port on the DF1 master. Enter a valid DF1 address (0-254). Address 255 is reserved for broadcast messages. The default is 0.
Transmit retries	Specifies the number of times a message is retried after the first attempt before being declared undeliverable. Enter a value 0-127. The default is 3.
ACK timeout	Specifies the amount of time you want the controller to wait for an acknowledgment to its message transmission. Enter a value 0-32767. Limits are defined in 20ms intervals. The default is 50 (1000ms).
Reply message wait	**Message-based polling mode only** Specifies the amount of time the master station waits after receiving an ACK to a master-initiated message before polling the slave station for a reply. Enter a value 0-65535. Limits are defined in 20ms intervals. The default is 5 (100ms).
Polling mode	Select one of these: • Message Based (slave cannot initiate messages) • Message Based (slave can initiate messages) - default • Standard (multiple message transfer per node scan) • Standard (single message transfer per node scan)
Master transmit	**Standard polling modes only** Select when the master station sends messages: • between station polls (default) • in polling sequence
Normal poll node tag	**Standard polling modes only** An integer tag array that contains the station addresses of the slave stations. Create a single-dimension array of data type INT that is large enough to hold all the normal station addresses. The minimum size is three elements. This tag must be controller-scoped. The format is: *list[0]* contains total number of stations to poll *list[1]* contains address of station currently being polled *list[2]* contains address of first slave station to poll *list[3]* contains address of second slave station to poll *list[n]* contains address of last slave station to poll
Normal poll group size	**Standard polling modes only** The number of stations the master station polls after polling all the stations in the priority poll array. Enter 0 (default) to poll the entire array.
Priority poll node tag	**Standard polling modes only** An integer tag array that contains the station addresses of the slave stations you need to poll more frequently. Create a single-dimension array of data type INT that is large enough to hold all the priority station addresses. The minimum size is three elements. This tag must be controller-scoped. The format is: *list[0]* contains total number of stations to be polled *list[1]* contains address of station currently being polled *list[2]* contains address of first slave station to poll *list[3]* contains address of second slave station to poll *list[n]* contains address of last slave station to poll
Active station tag	**Standard polling modes only** An array that stores a flag for each of the active stations on the DF1 link. Both the normal poll array and the priority poll array can have active and inactive stations. A station becomes inactive when it does not respond to the master's poll. Create a single-dimension array of data type SINT that has 32 elements (256 bits). This tag must be controller-scoped.
Error detection	Select BCC or CRC error detection. Configure both stations to use the same type of error checking. **BCC:** the controller sends and accepts messages that end with a BCC byte for error checking. BCC is quicker and easier to implement in a computer driver. This is the default. **CRC:** the controller sends and accepts messages with a 2-byte CRC for error checking. CRC is a more complete method.
Enable duplicate detection	Select whether or not the controller should detect duplicate messages. The default is duplicate detection enabled.

If you choose one of the standard polling modes

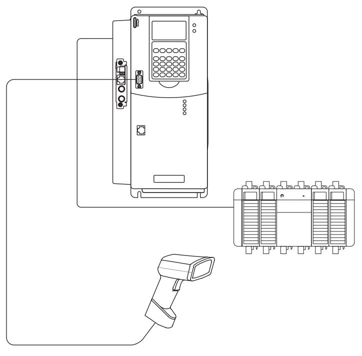
The master station polls the slave stations in this order:

1. all stations that are active in the priority poll array
2. one station that is inactive in the priority poll array
3. the specified number (normal poll group size) of active stations in the normal poll array
4. one inactive station, after all the active stations in the normal poll array have been polled

Use the programming software to change the display style of the active station array to binary so you can view which stations are active.

Example 3: DriveLogix Controller to a Bar Code Reader

In the following example, a workstation connects to a bar code reader. A bar code reader is an ASCII device, so you configure the serial port differently than in the previous examples. Configure the serial port for user mode, rather than a DF1 mode.

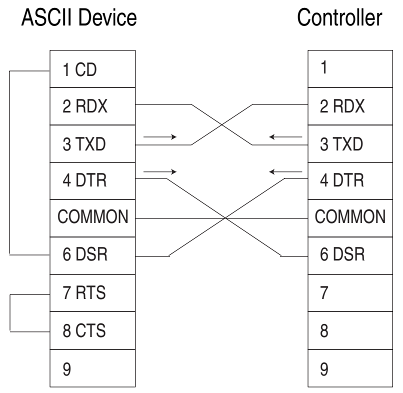
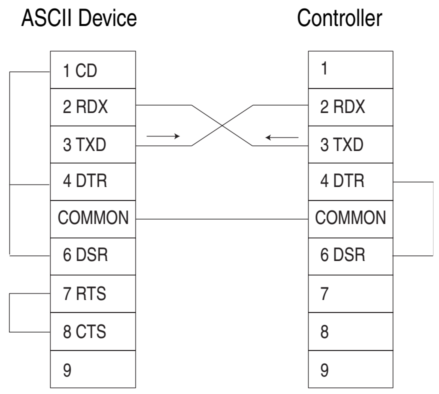


Connecting the ASCII Device to the Controller

To connect the ASCII device to the serial port of the controller:

1. For the serial port of the ASCII device, determine which pins send signals and which pins receive signals.

2. Connect the sending pins to the corresponding receiving pins and attach jumpers:

If the communications:	Then wire the connectors as follows:
handshake	 ASCII Device Controller 1 CD 2 RDX 3 TXD 4 DTR COMMON 6 DSR 7 RTS 8 CTS 9 1 2 RDX 3 TXD 4 DTR COMMON 6 DSR 7 8 9
do *not* handshake	 ASCII Device Controller 1 CD 2 RDX 3 TXD 4 DTR COMMON 6 DSR 7 RTS 8 CTS 9 1 2 RDX 3 TXD 4 DTR COMMON 6 DSR 7 8 9

3. Attach the cable shield to both connectors and tie the cable to both connectors.

4. Connect the cable to the controller and the ASCII device.

The following table lists the default serial port configuration settings for the ASCII protocol. You specify these settings on the User Protocol tab under Controller Properties.

Configuring User Mode

This field:	Description:
Buffer size	Specify the maximum size (in bytes) of the data array you plan to send and receive. The default is 82 bytes.
Termination characters	Specify the characters you will use to designate the end of a line. The default characters are '$r' and '$FF'.
Append characters	Specify the characters you will append to the end of a line. The default characters are '$r' and '$l'.
XON/XOFF	Select whether or not to regulate the flow of incoming data. The default is disabled.
Echo mode	Select whether or not to echo data back to the device from which it was sent. The default is disabled.
Delete mode	Select Ignore, CTR, or Printer for the delete mode. The default is Ignore.

Programming ASCII instructions

The controller supports ASCII instructions to communicate with ASCII devices. Your RSLogix 5000 programming software CDROM includes programming examples using ASCII instructions.

For information about using these examples, see the Logix5000 Controllers Reference Manual, publication 1756-RM001.

Communicating with Devices on an EtherNet/IP Link

De-energizing the Drive to Connect or Disconnect a Cable

⚠ **ATTENTION:** Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of configuring EtherNet/IP communication you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive's input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with configuring EtherNet/IP communication.

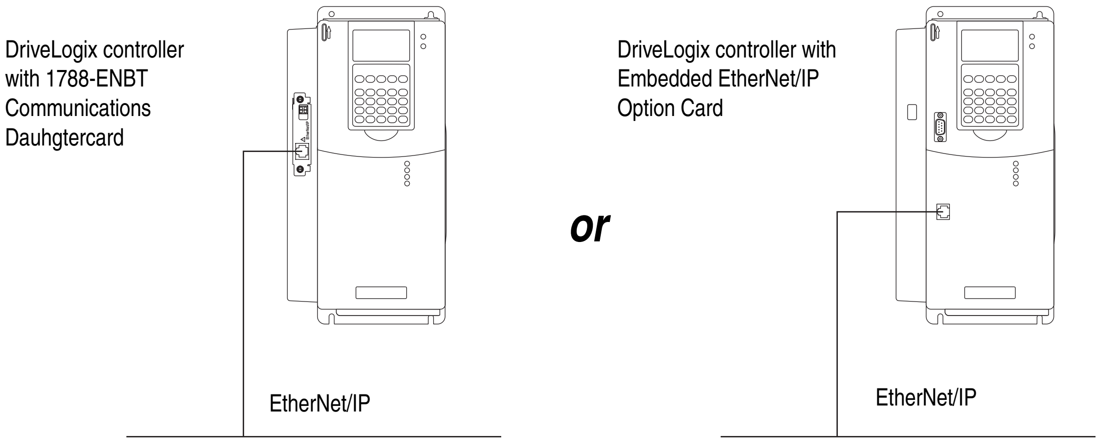
Communicating Through the Embedded EtherNet/IP Option

For the DriveLogix controller to operate on an Ethernet network, you need:

- a workstation with an appropriate EtherNet/IP communication daughtercard
- RSLinx software to configure the EtherNet/IP communication driver
- RSLogix 5000 programming software (Version 13 or later) to configure the embedded EtherNet/IP option or NetLinx EtherNet/IP communication daughtercard as part of the DriveLogix system
- an embedded EtherNet/IP option card installed on the DriveLogix controller (refer to Installing the Embedded EtherNet/IP Option Board on page B-7)

  or

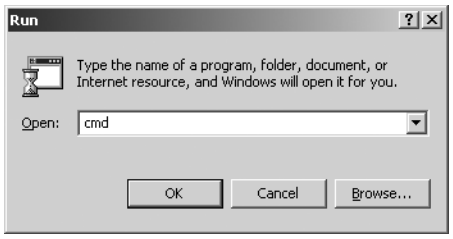
- a 1788-ENBT EtherNet/IP communication daughtercard installed on the DriveLogix controller (refer to Installing the Communications Daughtercard on page B-9)


DriveLogix controller with 1788-ENBT Communications Dauhgtercard
EtherNet/IP
or
DriveLogix controller with Embedded EtherNet/IP Option Card
EtherNet/IP

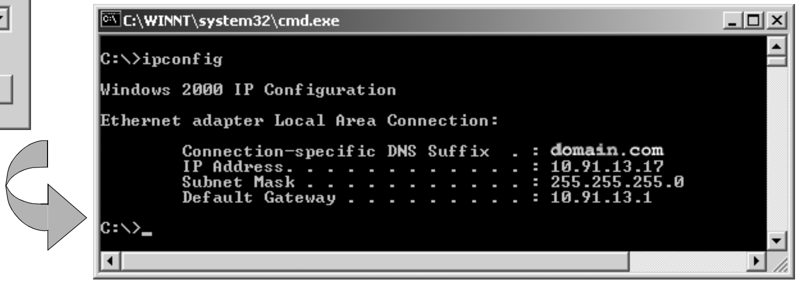
Determining Network Settings

Before configuring the system you must determine several network settings.

1. In the Run dialog box of Windows, type “cmd”.


Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open: cmd
OK
Cancel
Browse...

2. At the Command Prompt, type “ipconfig”, and note the values for the network settings.



```
C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter Local Area Connection:

        Connection-specific DNS Suffix  . : domain.com
        IP Address. . . . . . . . . . . . : 10.91.13.17
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 10.91.13.1

C:\>_
```

Assigning Network Parameters with BOOTP/DHCP

The EtherNet/IP Option board ships with BOOTP enabled. You must assign an IP address to the Ethernet port in order for the controller to communicate over an EtherNet/IP network.

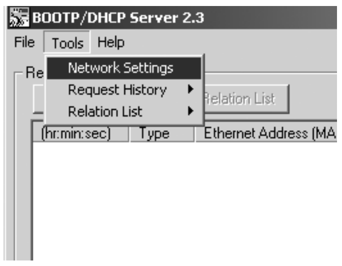
The BOOTP/DHCP utility is a stand alone program that is located in the:

- BOOTP-DHCP Server folder in the Rockwell Software program folder on the Start menu (the utility is automatically installed when you install RSLinx software)
- Tools directory on the RSLogix 5000 installation CD.

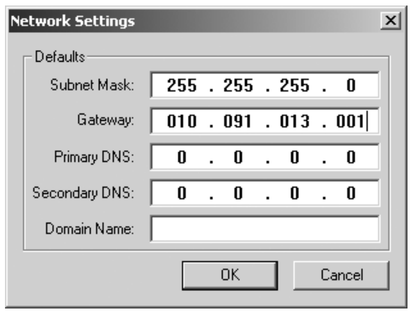
The computer running the BOOTP/DHCP utility and the DriveLogix5730 controller must be on the same EtherNet/IP network.

To use the BOOTP/DHCP utility:

1. Start the BOOTP/DHCP software.
2. From the Tools menu, select Network Settings.



Note: If you have not run the BOOTP/DHCP program before, it will automatically launch the Network Settings window.

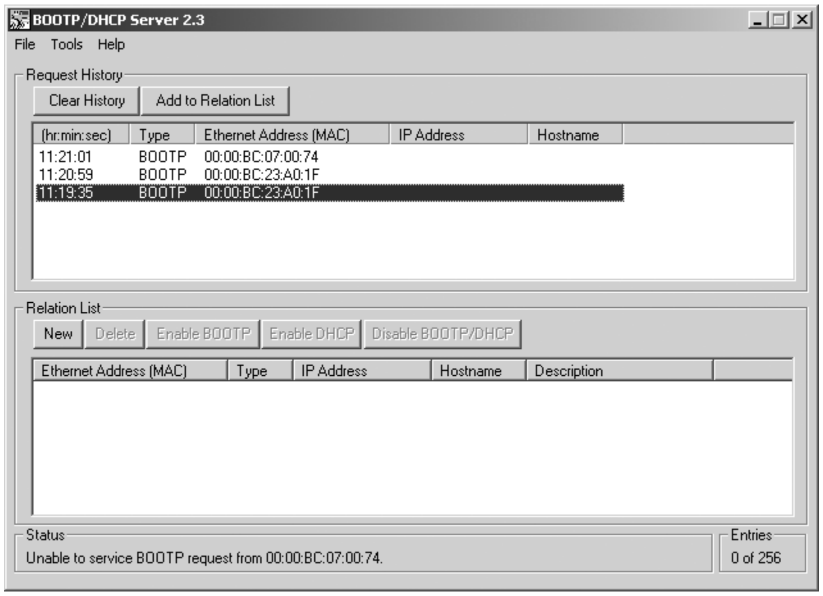


Subnet Mask: 255 . 255 . 255 . 0
Gateway: 010 . 091 . 013 . 001
Primary DNS: 0 . 0 . 0 . 0
Secondary DNS: 0 . 0 . 0 . 0
Domain Name:

3. Enter the required data.

Click OK.

4. In the Request History panel you see the hardware addresses of devices issuing BOOTP requests. Double-click on the hardware address of the device you want to configure.



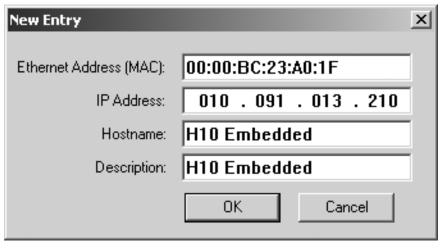
(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
11:21:01	BOOTP	00:00:BC:07:00:74		
11:20:59	BOOTP	00:00:BC:23:A0:1F		
11:19:35	BOOTP	00:00:BC:23:A0:1F		

Ethernet Address (MAC)	Type	IP Address	Hostname	Description

Status
Unable to service BOOTP request from 00:00:BC:07:00:74.

Entries
0 of 256

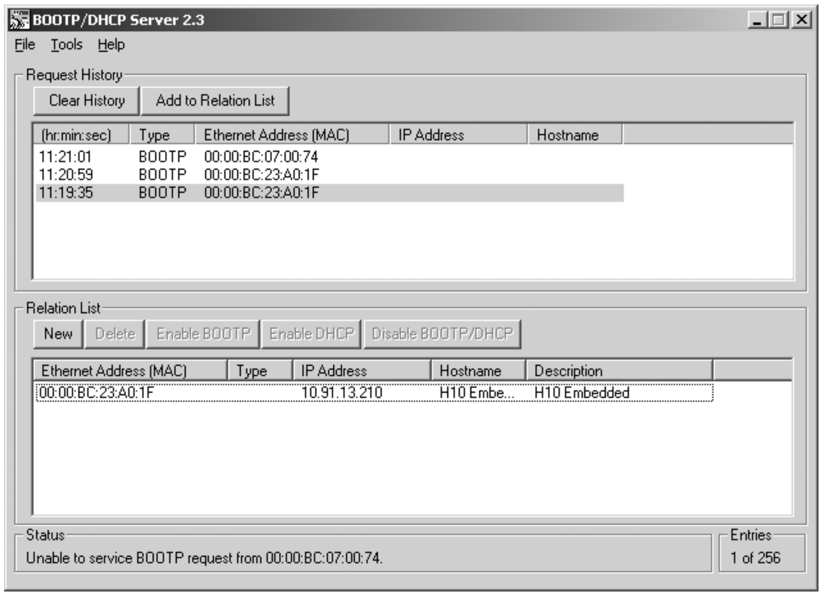
5. The New Entry window appears with the device's Ethernet Address (MAC). Enter the Ethernet address, IP address, subnet mask, and gateway.



Ethernet Address (MAC):	00:00:BC:23:A0:1F
IP Address:	010 . 091 . 013 . 210
Hostname:	H10 Embedded
Description:	H10 Embedded

Click OK.

The new entry then appears in the Relation List.



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
11:21:01	BOOTP	00:00:BC:07:00:74		
11:20:59	BOOTP	00:00:BC:23:A0:1F		
11:19:35	BOOTP	00:00:BC:23:A0:1F		

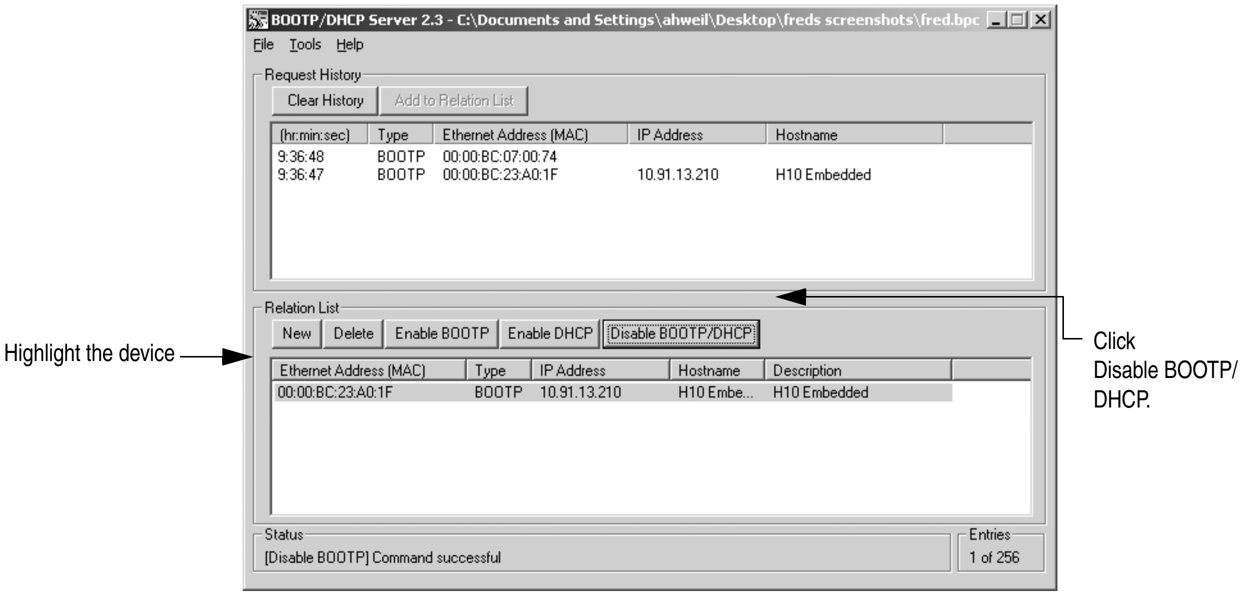
Ethernet Address (MAC)	Type	IP Address	Hostname	Description
00:00:BC:23:A0:1F		10.91.13.210	H10 Embe...	H10 Embedded

Status
Unable to service BOOTP request from 00:00:BC:07:00:74.

Entries
1 of 256

6. To permanently assign this configuration to the device, highlight the device and click on the Disable BOOTP/DHCP button. When power is recycled, the device uses the configuration you assigned and not issue a BOOTP request.

Important: If you do not select the Disable BOOTP/DHCP button, on a power cycles and drive resets, the controller clears the current IP
configuration and will again begin sending BOOTP requests.



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
9:36:48	BOOTP	00:00:BC:07:00:74		
9:36:47	BOOTP	00:00:BC:23:A0:1F	10.91.13.210	H10 Embedded

Ethernet Address (MAC)	Type	IP Address	Hostname	Description
00:00:BC:23:A0:1F	BOOTP	10.91.13.210	H10 Embe...	H10 Embedded

Status: [Disable BOOTP] Command successful

Entries: 1 of 256

Other methods to assign network parameters include:

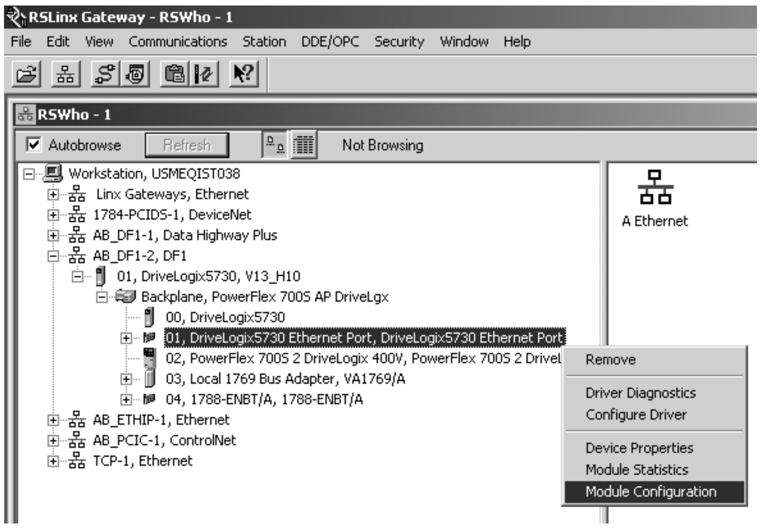
If you are working in these conditions:	Use this method for assigning network parameters:	See page:
• a BOOTP server is not available • the EtherNet/IP module is connected to another NetLinx network	RSLinx software	6-6
• the RSLogix 5000 project is online with the controller that communicates to or through the EtherNet/IP module	RSLogix 5000 software	6-7

If you use the Rockwell Software BOOTP or DHCP server in an up-linked subnet where an enterprise DHCP server exists, a module may get an address from the enterprise server before the Rockwell Automation utility even sees the module. You might have to disconnect from the uplink to set the address and have the module remember its static address before reconnecting to the uplink.

Using RSLinx Software to Set the IP Address Via the Controller Serial Port

You need RSLinx software version 2.41 or higher.

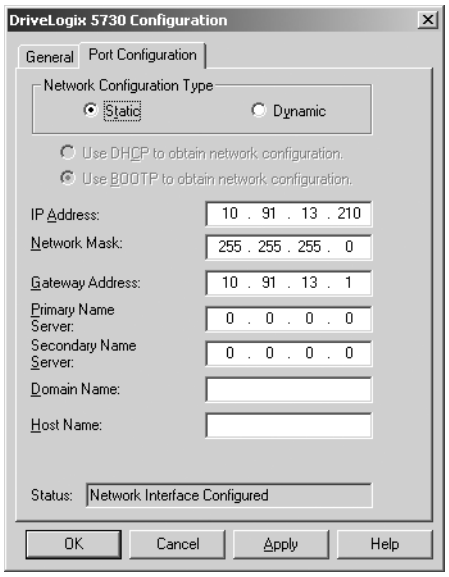
1. Make sure the controller that uses the IP address is installed and running.
2. Make a serial connection to the controller via the CH0 serial connector.
   You might also need to use RSLinx software to create a DF1 driver for the workstation. See Chapter 5 for more information.
3. Start RSLinx. Open the RSWho window. Navigate in RSWho to the Ethernet network.
4. Right-click on the Ethernet port (not the controller) and select Module Configuration



Workstation, USMEQIST038
Linx Gateways, Ethernet
1784-PCIDS-1, DeviceNet
AB_DF1-1, Data Highway Plus
AB_DF1-2, DF1
01, DriveLogix5730, V13_H10
Backplane, PowerFlex 700S AP DriveLgx
00, DriveLogix5730
01, DriveLogix5730 Ethernet Port, DriveLogix5730 Ethernet Port
02, PowerFlex 700S 2 DriveLogix 400V, PowerFlex 700S 2 DriveL
03, Local 1769 Bus Adapter, VA1769/A
04, 1788-ENBT/A, 1788-ENBT/A
AB_ETHIP-1, Ethernet
AB_PCIC-1, ControlNet
TCP-1, Ethernet

5. Select the Port Configuration tab, choose Static Network Configuration type, and enter the IP address, network (subnet) mask, and gateway address (if needed).

   You must leave the Static radio button selected to permanently assign this configuration to the port. If you select Dynamic, the controller clears its IP configuration and sends BOOTP requests on every power cycle and drive reset.



Network Configuration Type: [✓ Static] Dynamic

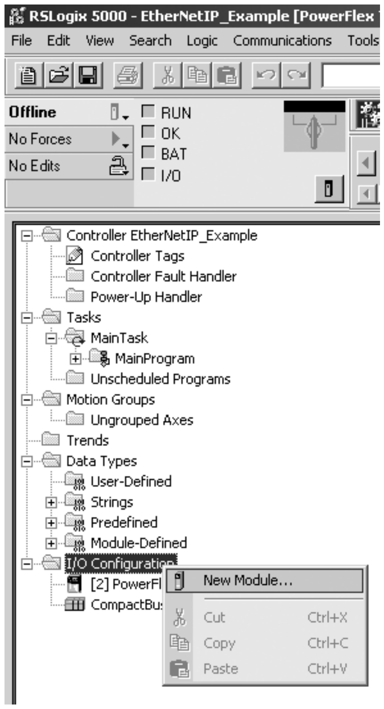
Use DHCP to obtain network configuration.
[✓ Use BOOTP to obtain network configuration.]

IP Address:	10 . 91 . 13 . 210
Network Mask:	255 . 255 . 255 . 0
Gateway Address:	10 . 91 . 13 . 1
Primary Name Server:	0 . 0 . 0 . 0
Secondary Name Server:	0 . 0 . 0 . 0
Domain Name:	
Host Name:	

Status: Network Interface Configured

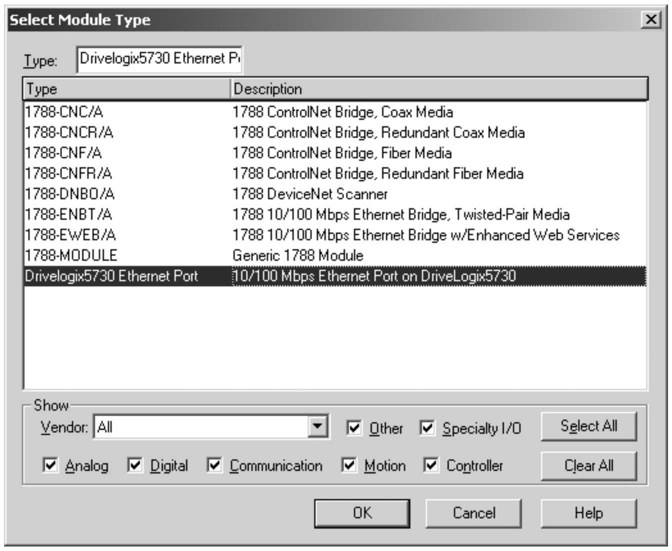
Using RSLogix 5000 Software to Set the IP Address

1. Create a new offline project, with the drive and any other necesary componenents.
2. In the Controller Organizer, select the I/O Configuration folder. Right-click the selected folder and select New Module.



Important: You must create an offline project and download it to the controller before going online. If the controller is already programmed, skip these first three steps.

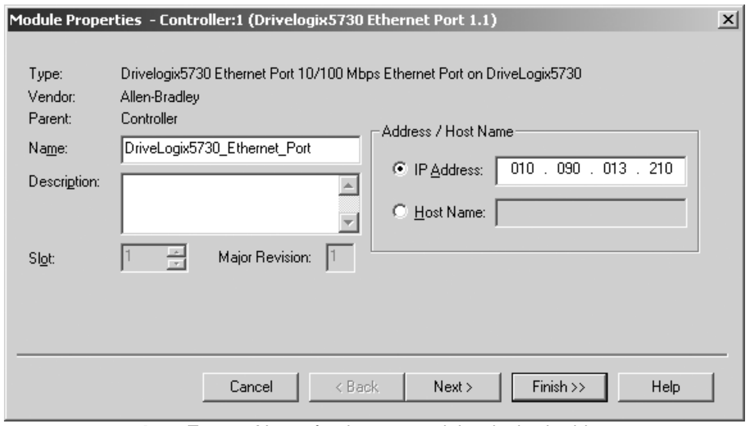
3. Select a DriveLogix5730 Ethernet Port from the list of possible communication devices. Click OK.


Select Module Type

Type: Drivelogix5730 Ethernet P

Type	Description
1788-CNC/A	1788 ControlNet Bridge, Coax Media
1788-CNCR/A	1788 ControlNet Bridge, Redundant Coax Media
1788-CNF/A	1788 ControlNet Bridge, Fiber Media
1788-CNFR/A	1788 ControlNet Bridge, Redundant Fiber Media
1788-DNBO/A	1788 DeviceNet Scanner
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1788-MODULE	Generic 1788 Module
Drivelogix5730 Ethernet Port	10/100 Mbps Ethernet Port on DriveLogix5730

Show: Vendor: All; [✓ Other] [✓ Specialty I/O] [✓ Analog] [✓ Digital] [✓ Communication] [✓ Motion] [✓ Controller]


Module Properties - Controller:1 (Drivelogix5730 Ethernet Port 1.1)

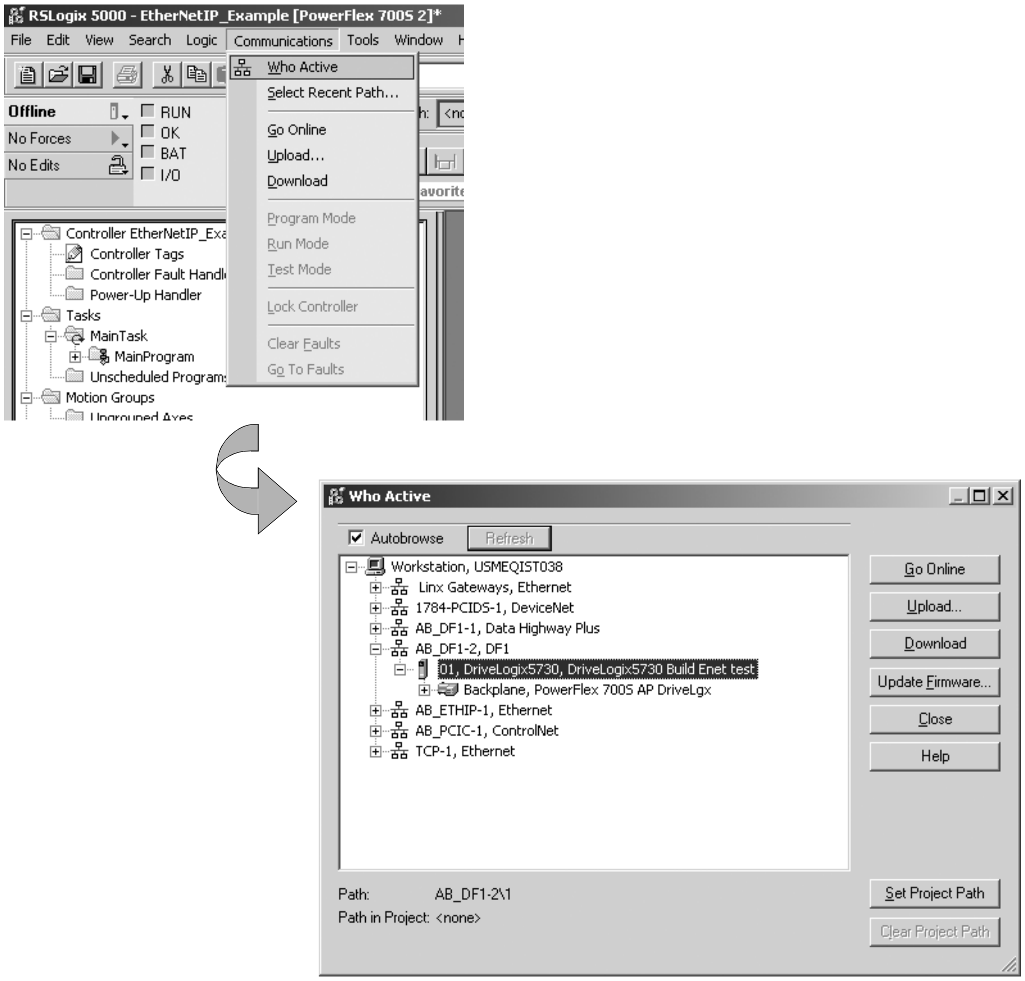
Type: Drivelogix5730 Ethernet Port 10/100 Mbps Ethernet Port on DriveLogix5730
Vendor: Allen-Bradley
Parent: Controller
Name: DriveLogix5730_Ethernet_Port
Description:
Slot: 1
Major Revision: 1

Address / Host Name
[✓ IP Address:] 010 . 090 . 013 . 210
Host Name:

4. Enter a Name for the port and the desired address. Click Finish.

4. To go online with the controller in RSLogix 5000 programming software, select Who Active from the Communications menu. Then highlight the controller and click the Download button.

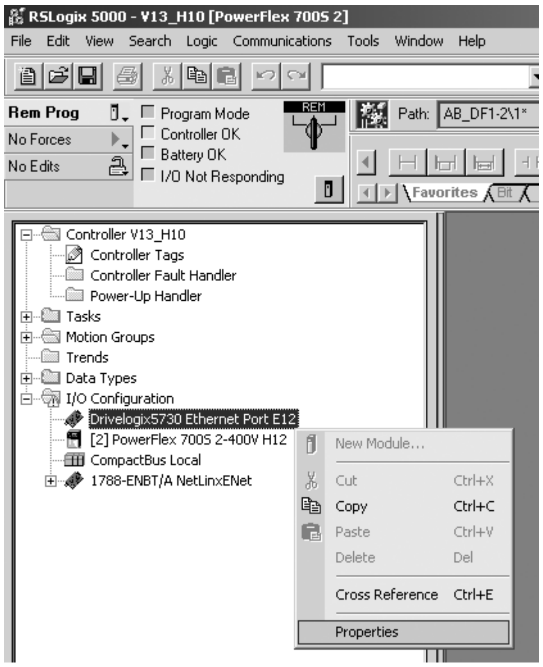
   If the controller is already programmed click Go Online instead.



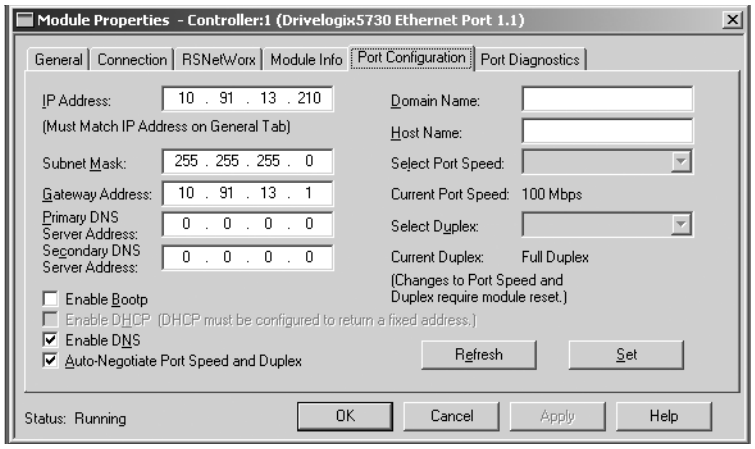
```
Workstation, USMEQIST038
    Linx Gateways, Ethernet
    1784-PCIDS-1, DeviceNet
    AB_DF1-1, Data Highway Plus
    AB_DF1-2, DF1
        01, DriveLogix5730, DriveLogix5730 Build Enet test
            Backplane, PowerFlex 700S AP DriveLgx
    AB_ETHIP-1, Ethernet
    AB_PCIC-1, ControlNet
    TCP-1, Ethernet
```

Path: AB_DF1-2\1
Path in Project: <none>

5. In the Controller Organizer, right click on the icon for the port and select Properties.



6. Select the Port Configuration tab and enter the proper data.
Uncheck the Enable Bootp checkbox.
Click Set, Apply and OK to make changes.



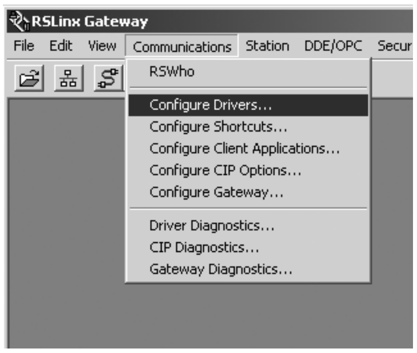
IP Address: 10 . 91 . 13 . 210
(Must Match IP Address on General Tab)
Subnet Mask: 255 . 255 . 255 . 0
Gateway Address: 10 . 91 . 13 . 1
Primary DNS Server Address: 0 . 0 . 0 . 0
Secondary DNS Server Address: 0 . 0 . 0 . 0
Domain Name:
Host Name:
Select Port Speed:
Current Port Speed: 100 Mbps
Select Duplex:
Current Duplex: Full Duplex
(Changes to Port Speed and Duplex require module reset.)
☐ Enable Bootp
☐ Enable DHCP (DHCP must be configured to return a fixed address.)
☑ Enable DNS
☑ Auto-Negotiate Port Speed and Duplex
Status: Running

Configuring the EtherNet/IP Communications Driver

Important: EtherNet/IP only works within the local subnet and with EtherNet/IP devices (not Ethernet control products).

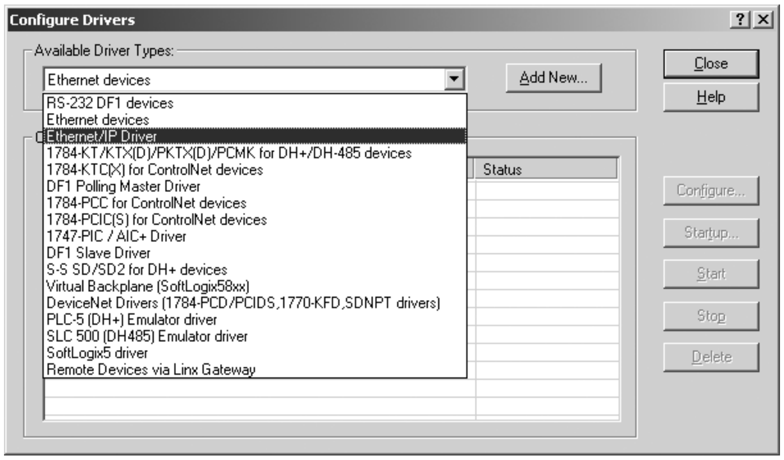
To configure the AB_ETH Ethernet communication driver perform the following steps in **RSLinx**:

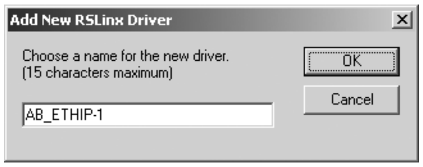
1. From the Communications menu, select Configure Drivers.



2. Select EtherNet/IP Driver from the list of Available Driver Types.

   Then Click Add New.


Configure Drivers
Available Driver Types:
Ethernet devices
Add New...
Close
Help
RS-232 DF1 devices
Ethernet devices
Ethernet/IP Driver
1784-KT/KTX(D)/PKTX(D)/PCMK for DH+/DH-485 devices
1784-KTC(X) for ControlNet devices
DF1 Polling Master Driver
1784-PCC for ControlNet devices
1784-PCIC(S) for ControlNet devices
1747-PIC / AIC+ Driver
DF1 Slave Driver
S-S SD/SD2 for DH+ devices
Virtual Backplane (SoftLogix58xx)
DeviceNet Drivers (1784-PCD/PCIDS,1770-KFD,SDNPT drivers)
PLC-5 (DH+) Emulator driver
SLC 500 (DH485) Emulator driver
SoftLogix5 driver
Remote Devices via Linx Gateway
Status
Configure...
Startup...
Start
Stop
Delete

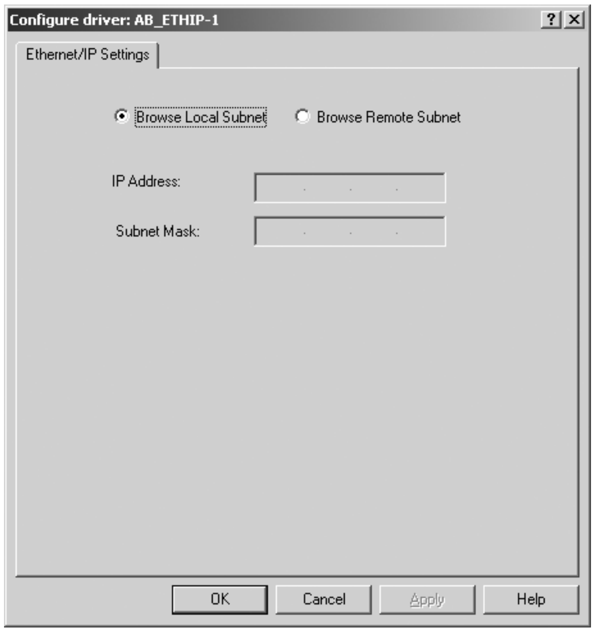


3. Selecte the default driver name (i.e. AB-ETHIP-1) or type in your own name.

   Then Click OK.

After you create the driver, configure it to correspond to the Ethernet port on the Embedded EtherNet/IP option board.

4. Select where the EtherNet/IP devices reside. The software locates valid IP addresses.

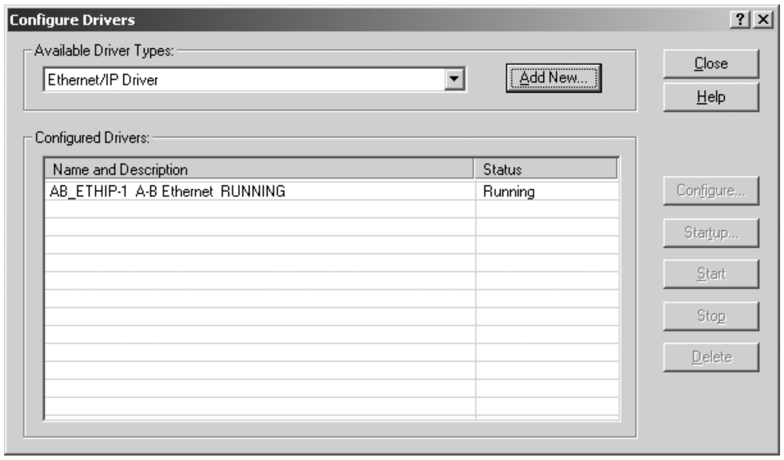


[✓ Browse Local Subnet] Browse Remote Subnet

IP Address:

Subnet Mask:

Click OK.



Available Driver Types: Ethernet/IP Driver

Configured Drivers:

Name and Description	Status
AB_ETHIP-1 A-B Ethernet RUNNING	Running

5. The driver is now available and you can select the EtherNet/IP port in RSLogix 5000.

Controller Connections Over EtherNet/IP

A Logix system uses a connection to establish a communication link between two devices. Connections can be:

- controller to distributed I/O or remote communication modules
- produced and consumed tags
- messages

You indirectly determine the number of connections the controller uses by configuring the controller to communicate with other devices in the system.

Connections are allocations of resources that provide more reliable communications between devices than unconnected messages.

All EtherNet/IP connections are unscheduled. An unscheduled connection is a message transfer between controllers that is triggered by the requested packet interval (RPI) or the program (such as a MSG instruction). Unscheduled messaging lets you send and receive data when needed.

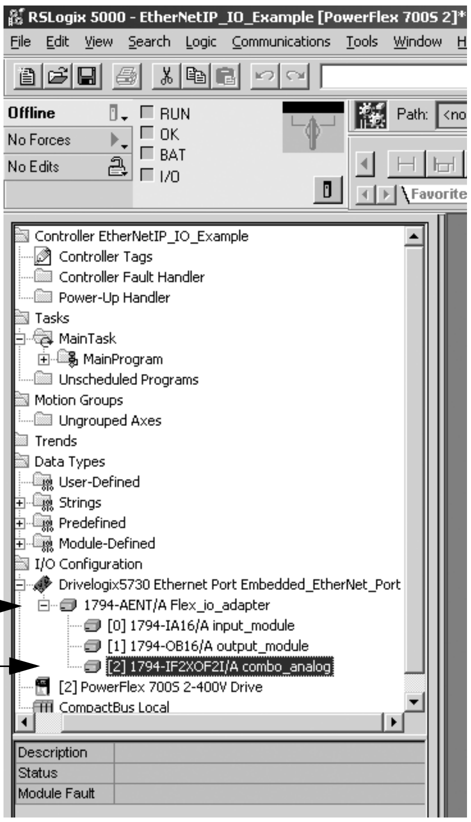
The DriveLogix5730 controller supports 32 CIP connections over an EtherNet/IP network.

Configuring Distributed I/O

The DriveLogix5730 controller supports distributed I/O over a EtherNet/IP link. Configuring I/O in a remote chassis is similar to configuring local I/O. You create the remote communication module and distributed I/O modules on the local Ethernet port.

To communicate with distributed I/O modules, you add a remote adapter and I/O modules to the I/O Configuration folder of the controller.

To Add Distributed I/O Build the I/O Configuration in this Order


RSLogix 5000 - EtherNetIP_IO_Example [PowerFlex 700S 2]*
File Edit View Search Logic Communications Tools Window
Offline
No Forces
No Edits
RUN
OK
BAT
I/O
Path: <no
Favorite
Controller EtherNetIP_IO_Example
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled Programs
Motion Groups
Ungrouped Axes
Trends
Data Types
User-Defined
Strings
Predefined
Module-Defined
I/O Configuration
Drivelogix5730 Ethernet Port Embedded_EtherNet_Port
1794-AENT/A Flex_io_adapter
[0] 1794-IA16/A input_module
[1] 1794-OB16/A output_module
[2] 1794-IF2XOF2I/A combo_analog
[2] PowerFlex 700S 2-400V Drive
CompactBus Local
Description
Status
Module Fault

1. Add the remote adapter to the embedded EtherNet/IP port of the controller.

2. Add the I/O modules to the remote adapter.

Accessing Distributed I/O

I/O information is presented as a structure of multiple fields, which depend on the specific features of the I/O module. The name of the structure is based on the location of the I/O module in the system. Each I/O tag is automatically created when you configure the I/O module through the programming software. Each tag name follows this format:

Location:SlotNumber:Type.MemberName.SubMemberName.Bit

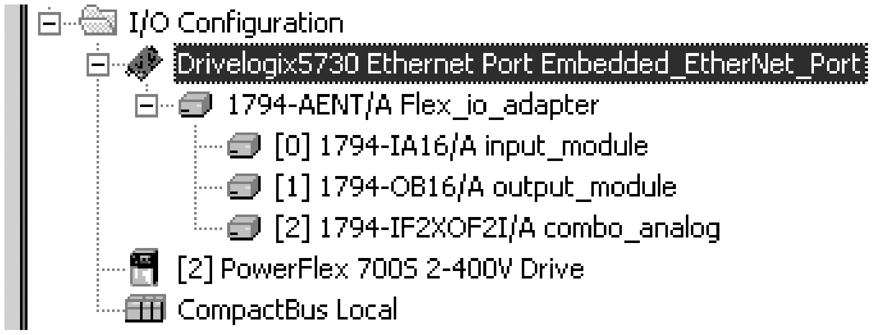
where:

This address variable:	Is:
Location	Identifies network location LOCAL = local DIN rail or chassis ADAPTER_NAME = identifies remote adapter or bridge
SlotNumber	Slot number of I/O module in its chassis

Type	Type of data
	I = input
	O = output
	C = configuration
	S = status

MemberName	Specific data from the I/O module; depends on the type of data the module can store For example, Data and Fault are possible fields of data for an I/O module. Data is the common name for values the are sent to or received from I/O points.
SubMemberName	Specific data related to a MemberName.
Bit (optional)	Specific point on the I/O module; depends on the size of the I/O module (0-31 for a 32-point module)

Example



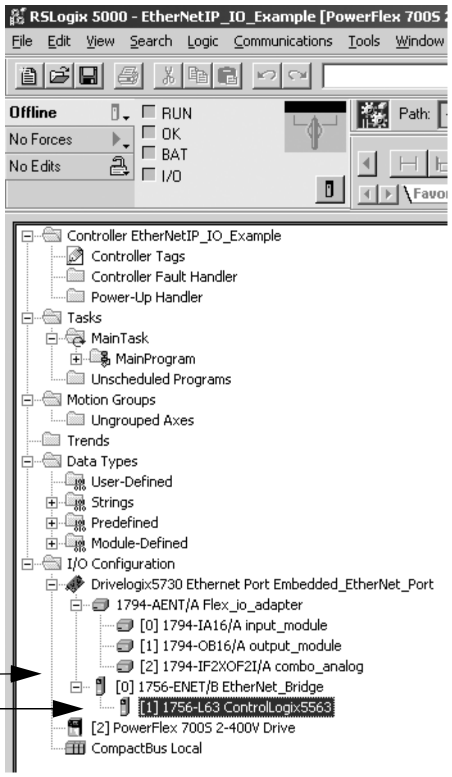
```
I/O Configuration
  Drivelogix5730 Ethernet Port Embedded_EtherNet_Port
    1794-AENT/A Flex_io_adapter
      [0] 1794-IA16/A input_module
      [1] 1794-OB16/A output_module
      [2] 1794-IF2XOF2I/A combo_analog
    [2] PowerFlex 700S 2-400V Drive
    CompactBus Local
```

Device:	**Example Tag Names (automatically created by the software):**
remote adapter “FLEX_io_adapter”	FLEX_io_adapter:I FLEX_io_adapter:I.SlotStatusBits FLEX_io_adapter:I.Data FLEX_io_adapter:O FLEX_io_adapter:O.Data
remote “input_module” in slot 0 rack-optimized connection	FLEX_io_adapter:0:C FLEX_io_adapter:0:C.Config FLEX_io_adapter:0:C.DelayTime_0 FLEX_io_adapter:0:C.DelayTime_1 FLEX_io_adapter:0:C.DelayTime_2 FLEX_io_adapter:0:C.DelayTime_3 FLEX_io_adapter:0:C.DelayTime_4 FLEX_io_adapter:0:C.DelayTime_5 FLEX_io_adapter:0:I
remote “output_module” in slot 1 rack-optimized connection	FLEX_io_adapter:1:C FLEX_io_adapter:1:C.SSData FLEX_io_adapter:1:O FLEX_io_adapter:1:O.Data
remote “combo_analog” in slot 2 direct connection	FLEX_io_adapter:2:C FLEX_io_adapter:2:C.InputFIlter FLEX_io_adapter:2:C.InputConfiguration FLEX_io_adapter:2:C.OutputConfiguration FLEX_io_adapter:2:C.RTSInterval FLEX_io_adapter:2:C.SSCh0OuputData FLEX_io_adapter:2:C.SSCH1OutputData FLEX_io_adapter:2:I

Adding a Remote Controller

If you want to add the controller as a remote consumed controller to the I/O configuration, you first add the EtherNet/IP port and then the controller.

To Add a Remote Controller Build the I/O Configuration in this Order


RSLogix 5000 - EtherNetIP_IO_Example [PowerFlex 700S
File Edit View Search Logic Communications Tools Window
Offline
No Forces
No Edits
RUN
OK
BAT
I/O
Path:
Controller EtherNetIP_IO_Example
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled Programs
Motion Groups
Ungrouped Axes
Trends
Data Types
User-Defined
Strings
Predefined
Module-Defined
I/O Configuration
Drivelogix5730 Ethernet Port Embedded_EtherNet_Port
1794-AENT/A Flex_io_adapter
[0] 1794-IA16/A input_module
[1] 1794-OB16/A output_module
[2] 1794-IF2XOF2I/A combo_analog
[0] 1756-ENET/B EtherNet_Bridge
[1] 1756-L63 ControlLogix5563
[2] PowerFlex 700S 2-400V Drive
CompactBus Local

1. Add the remote EtherNet/IP port or bridge to the local EtherNet/IP port.

2. Add the remote controller to the remote EtherNet/IP port.

Producing and Consuming Data

The DriveLogix5730 controller supports the ability to produce (broadcast) and consume (receive) system-shared tags over an EtherNet/IP link. Produced and consumed data is accessible by multiple controllers over an Ethernet network. The controller sends or receives data at a predetermined RPI rate. This is the recommended method of communication between Logix controllers.

Produced and consumed tags must be controller-scoped tags of DINT or REAL data type, or in an array or structure.

Tag type:	Description:	Specify:
produced	These are tags that the controller produced for other controllers to consume.	Enabled for producing How many consumers allowed
consumed	These are tags whose values are produced by another controller.	Controller name that owns the tag that the local controller wants to consume Tag name or instance that the controller wants to consume Data type of the tag to consume Update interval of how often the local controller consumes the tag

The producer and consumer must be configured correctly for the specified data to be shared. A produced tag in the producer must be specified exactly the same as a consumed tag in the consumer.

If any produced/consumed tag between a producer and consumer is not specified correctly, none of the produced/consumed tags for that producer and consumer will be transferred. For example, if a DriveLogix5730 controller is consuming three tags that another controller consumes but the first tag is specified incorrectly, none of the tags are transferred to the consuming DriveLogix5730 controller.

However, one consumer failing to access shared data does not affect other consumers accessing the same data. For example, if the producing DriveLogix5730 controller from the previous example also produced tags for other consuming controllers but did so correctly, those tags are still transferred to the additional consuming controllers.

Maximum Number of Produced and Consumed Tags

The maximum number of produced/consumed tags that you can configure depends on the connection limits of the Ethernet port on the controller. You can have a maximum of 32 connections through the Ethernet port.

Each produced tag uses one connection for the tag and the first configured consumer of the tag. Each consumer thereafter uses an additional connection.

If you have a lot of data to produce or consume, organize that data into an array. An array is treated as one tag, so it uses only one connection. An array can be as large as 488 bytes, which is also the limit of a produced or consumed tag.

Size Limit of a Produced or Consumed Tag

A produced or consumed tag can be as large as 488 bytes, but it must also fit within the bandwidth of the EtherNet/IP network.

Producing a tag

Produced data must be of DINT or REAL data type or a structure. You can use a user-defined structure to group BOOL, SINT, and INT data to be produced. To create a produced tag:

1. You must be programming offline.
2. In the controller organizer, double-click the Controller Tags folder and then click the Edit Tags tab.
3. Select the tag that you want to produce, or enter a new tag, and display the Tag Properties dialog box.
4. Make sure the tag is controller scope.
5. Select the “Produce this tag” check box. Specify how many controllers can consume the tag.

You can produce a base, alias, or consumed tag.

The consumed tag in a receiving controller must have the same data type as the produced tag in the originating controller. The controller performs type checking to ensure proper data is being received.

Produced tags require connections. The number of connections depends on how many controllers are consuming the tags. The controller requires one connection for the produced tag and the first consumer. Then, the controller requires an additional connection for each subsequent consumer.

Consuming a Tag

A consumed tag represents data that is produced (broadcast) by one controller and received and stored by the consuming controller. To create a consumed tag:

1. You must be programming offline.
2. In the controller organizer, double-click the Controller Tags folder and then click the Edit Tags tab.
3. Select the tag that you want to consume, or enter a new tag, and display the Tag Properties dialog box.
4. Specify:

In this field:	Type or select:
Tag Type	Select Consumed.
Controller	Select the name of the other controller. You must have already created the controller in the controller organizer for the controller name to be available.
Remote Tag Name Remote Instance	Type a name for the tag in the other controller you want to consume. **Important:** The name must match the name in the remote controller exactly, or the connection faults.
RPI (requested packet interval)	Type the amount of time in msec between updates of the data from the remote controller. The local controller will receive data at least this fast. Virtual-backplane controllers, such as DriveLogix5730, CompactLogix and FlexLogix controllers, only produce data at RPIs in powers of two milliseconds (such as 2, 4, 8, 16, 32, 64, etc.), or when triggered by an IOT instruction.
Display Style	If you are creating a consumed tag that refers to a tag whose data type is BOOL, SINT, INT, DINT, or REAL, you can select a display style. This display style defines how the tag value will be displayed in the data monitor and ladder editor. The display style does not have to match the display style of the tag in the remote controller.

All consumed tags are automatically controller-scope.

The produced tag in the originating DriveLogix5730 controller must have the same data type as the consumed tag in the consuming controller. The DriveLogix5730 controller performs type checking to make sure proper data is being received.

Important: If a consumed-tag connection fails, none of the tags are transferred from the producing controller to the consuming controller.

Sending Messages

The DriveLogix5730 controller can send MSG instructions to other controllers and devices over an EtherNet/IP link. Each MSG instruction requires you to specify a target and an address within the target.

MSG instructions are unscheduled. The type of MSG determines whether or not it requires a connection. If the MSG instruction requires a connection, it opens the needed connection when it is executed. You can configure the

MSG instruction to keep the connection open (cache) or to close it after sending the message.

This type of MSG:	Using this communication method:	Uses a connection:	Which you can cache:
CIP data table read or write	CIP	X	X
PLC-2, PLC-3, PLC-5, or SLC (all types)	CIP	X	X
	CIP with Source ID	X	X
	DH+	X	
CIP generic	CIP	X(1)	X
block-transfer read or write	na	X	X

(1) You can connect CIP generic messages, but for most applications, we recommend you leave CIP generic messages unconnected.

The update time of local I/O modules may increase when the controller is bridging messages.

Important: Bridging over the DriveLogix5730 controller should be targeted toward applications that are not real time dependent, such as RSLogix 5000 program downloads and ControlFlash updates.

Communicating with Another Logix-based Controller

All Logix-based controllers can use MSG instructions to communicate with each other. The following examples show how to use tags in MSG instructions between Logix-based controllers.

Type of MSG Instruction:	Example Source and Destination:	
Logix-based controller writes to Logix-based controller (CIP Data Table Write)	source tag	*array_1*
	destination tag	*array_2*
Logix-based controller reads from Logix-based controller (CIP Data Table Read)	source tag	*array_1*
	destination tag	*array_2*

The source and destination tags:

- must be controller-scoped tags.
- can be of any data type, except for AXIS, MESSAGE, or MOTION_GROUP.

Communicating with other controllers over EtherNet/IP

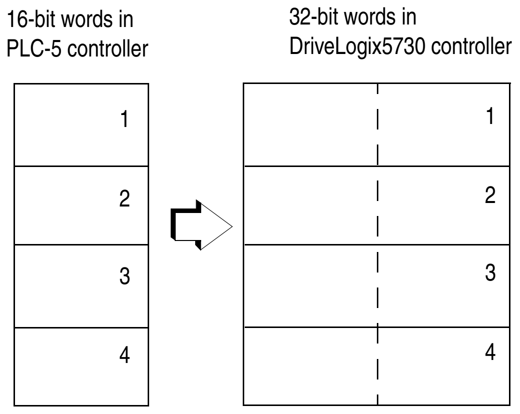
The DriveLogix5730 controller also uses MSG instructions to communicate with PLC and SLC controllers. The MSG instructions differ depending on which controller initiates the instruction.

For MSG instructions originating from a DriveLogix5730 controller to a PLC or SLC controller:

Type of MSG Instruction:	Supported Source File Types:	Supported Destination File Types:
DriveLogix5730 writes to PLC-5 or SLC	In the DriveLogix5730 controller, specify the source data type based on the destination device: PLC-5: SINT, INT, DINT, or REAL SLC: INT, REAL Example source element: *array_1*	Specify the destination file type based on the destination device: PLC-5 typed write: S, B, N, or F PLC-5 word-range write: S, B, N, F, I, O, A, or D SLC: B, N or F Example destination tag: *N7:10*
DriveLogix5730 writes to PLC-2	In the DriveLogix5730 controller, select one of these data types: SINT, INT, DINT, or REAL Example source element: *array_1*	Use the PLC-2 compatibility file. Example destination tag: *010*
DriveLogix5730 reads from PLC-5 or SLC	Specify the destination file type based on the destination device: PLC-5 typed read: S, B, N, or F PLC-5 word-range read: S, B, N, F, I, O, A, or D SLC: B, N or F Example source element: *N7:10*	In the DriveLogix5730 controller, specify the destination data type based on the destination device: PLC-5: SINT, INT, DINT, or REAL SLC: INT, REAL Example destination tag: *array_1*
DriveLogix5730 reads from PLC-2	Use the PLC-2 compatibility file. Example source element: *010*	In the DriveLogix5730 controller, select one of these data types: SINT, INT, DINT, or REAL Example destination tag: *array_1*

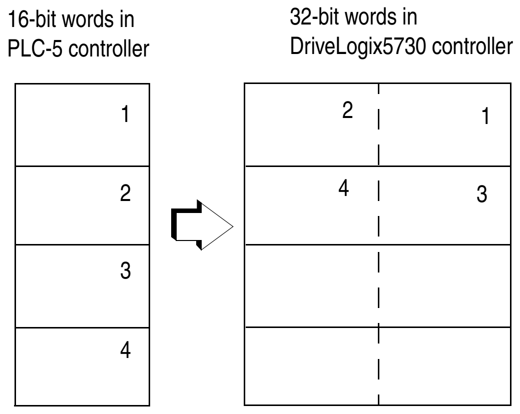
The DriveLogix5730 controller can send typed or word-range commands to PLC-5 controllers. These commands read and write data differently. The following diagrams show how the typed and word-range commands differ.

Typed read command


16-bit words in PLC-5 controller
32-bit words in DriveLogix5730 controller
1
2
3
4
1
2
3
4

The typed commands maintain data structure and value.

Word-range read command


16-bit words in PLC-5 controller
32-bit words in DriveLogix5730 controller
1
2
3
4
2
1
4
3

The word-range commands fill the destination tag contiguously. Data structure and value change depending on the destination data type.

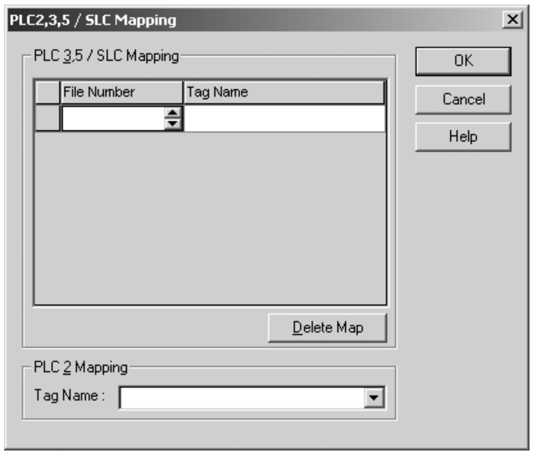
The DriveLogix5730 controller can process messages initiated from PLC or SLC controllers. These messages use data table addresses. In order for these controllers to access tags within the DriveLogix5730 controller, you map tags to data table addresses.

Mapping Addresses

The programming software includes a PLC/SLC mapping tool which allows you to make an existing controller array tag in the local controller available to PLC-2, PLC-3, PLC-5, or SLC controllers.

To map addresses:

1. From the Logic menu, select Map PLC/SLC Messages.



2. Specify this information:

For:	In this field:	Specify:	For example:
PLC-3, PLC-5, and SLC controllers	File Number	Type the file number of the data table in the PLC/SLC controller.	*10*
	Tag Name	Type the array tag name the local controller uses to refer to the PLC/SLC data table address. The tag must be an integer array (SINT, INT, or DINT) that is large enough for the message data.	*array_1*
PLC-2 controllers	Tag Name	Type the tag name to be the PLC-2 compatibility file.	*200*

TIP: You can map as many tags as you want to a PLC-3, PLC-5, or SLC controller. You can map only one tag to a PLC-2 controller.

The following table shows example source and destination tags and elements for different controller combinations.

Type of MSG Instruction:	Example Source and Destination:	
PLC-5 writes to DriveLogix5730 SLC writes to DriveLogix5730 SLC 5/05 SLC 5/04 OS402 and above SLC 5/03 OS303 and above	source element	*N7:10*
	destination tag	*"array_1"*
	The PLC-5, PLC-3, and SLC controllers support logical ASCII addressing so you do not have to map a compatibility file for MSG instructions initiated by a PLC-5, PLC-3, or SLC controller. Place the DriveLogix5730 tag name in double quotes ("). You could optionally map a compatibility file. For example, if you enter *10* for the compatibility file, you enter *N10:0* for the destination tag.	
PLC-2 writes to DriveLogix5730	source element	*010*
	destination tag	*200*
	The destination tag is the three-digit PLC-2 address you specified for PLC-2 mapping.	
PLC-5 reads from DriveLogix5730 SLC reads from DriveLogix5730 SLC 5/05 SLC 5/04 OS402 and above SLC 5/03 OS303 and above	source tag	*"array_1"*
	destination element	*N7:10*
	The PLC-5, PLC-3, and SLC controllers support logical ASCII addressing so you do not have to map a compatibility file for MSG instructions initiated by a PLC-5, PLC-3, or SLC controller. Place the DriveLogix5730 tag name in double quotes ("). You could optionally map a compatibility file. For example, if you enter *10* for the compatibility file, you enter *N10:0* for the source tag.	
PLC-2 reads from DriveLogix5730	source tag	*200*
	destination element	*010*
	The source tag is the three-digit PLC-2 address you specified for PLC-2 mapping.	

When the DriveLogix5730 controller initiates messages to PLC or SLC controllers, you do not have to map compatibility files. You enter the data table address of the target device just as you would a tag name.

SLC 5/05 controllers, SLC 5/04 controllers (OS402 and above), and SLC 5/03 controllers (OS303 and above) support logical ASCII addressing and support PLC/SLC mapping (see the examples above). For all other SLC or MicroLogix1000 controllers, you must map a PLC-2 compatibility file (see the PLC-2 examples above).

Using a MSG Instruction to Send an Email

The controller is an email client that uses a mail relay server to send email. The DriveLogix5730 controller can execute a generic CIP message that sends an email message to an SMTP mail relay server using the standard SMTP protocol.

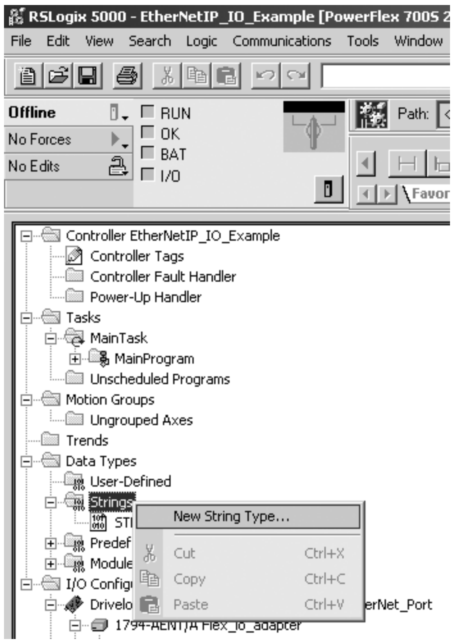
Some mail relay servers require a domain name be provided during the initial handshake of the SMTP session. For these mail relay servers, make sure you specify a domain name when you configure the network settings. See 6-3 for information on configuring the network settings of the controller and specifying a domain name.

Important: Be careful to write the ladder logic to ensure the MSG instructions are not continuously triggered to send email messages.

Creating String Type for Email String Data Type

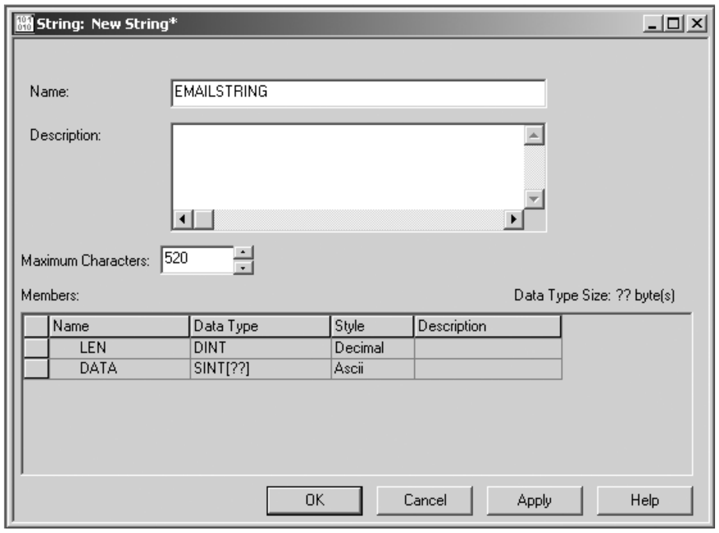
The tags for the email text and transmission status can contain as many as 474 characters. For these tags, you must create a user-defined STRING data type that is larger than the default. For example, create a STRING data type named EMAILSTRING.

1. While offline, in the Controller Organizer, right click on the folder for Strings and select New String Type.



2. Enter the Name (e.g. EMAILSTRING) and the maximum number of characters (e.g. 520).

   Click Apply, then OK.



Name: EMAILSTRING

Maximum Characters: 520

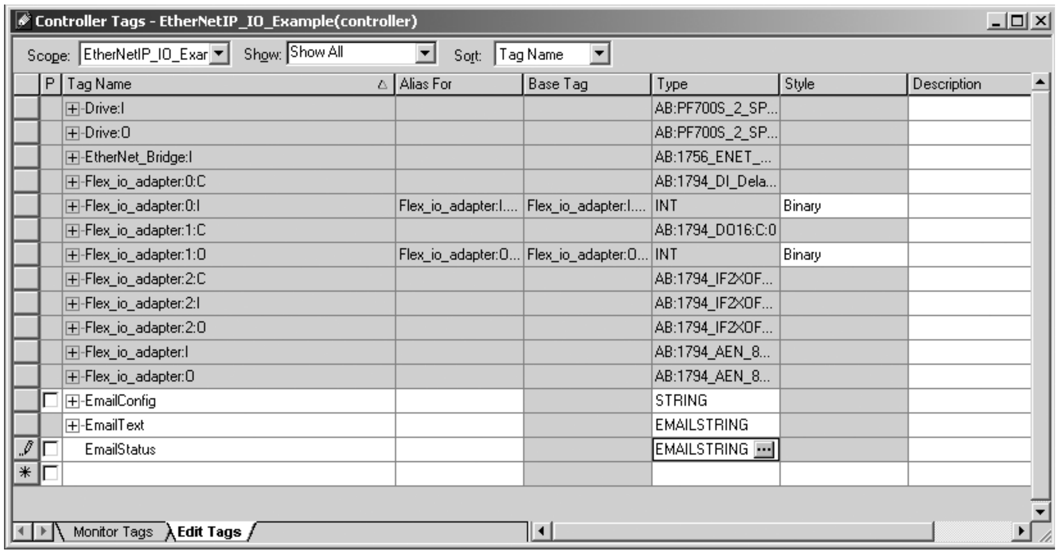
Members:

Name	Data Type	Style	Description
LEN	DINT	Decimal	
DATA	SINT[??]	Ascii	

Creating String Tags

On the Edit tab of the Controller Tags window, create three tags:

- one to identify the mail server (use STRING data type for this tag)
- one for the email text (use EMAILSTRING data type for this tag)
- one for the email transmission status (use EMAILSTRING data type for this tag)



Controller Tags - EtherNetIP_IO_Example(controller)

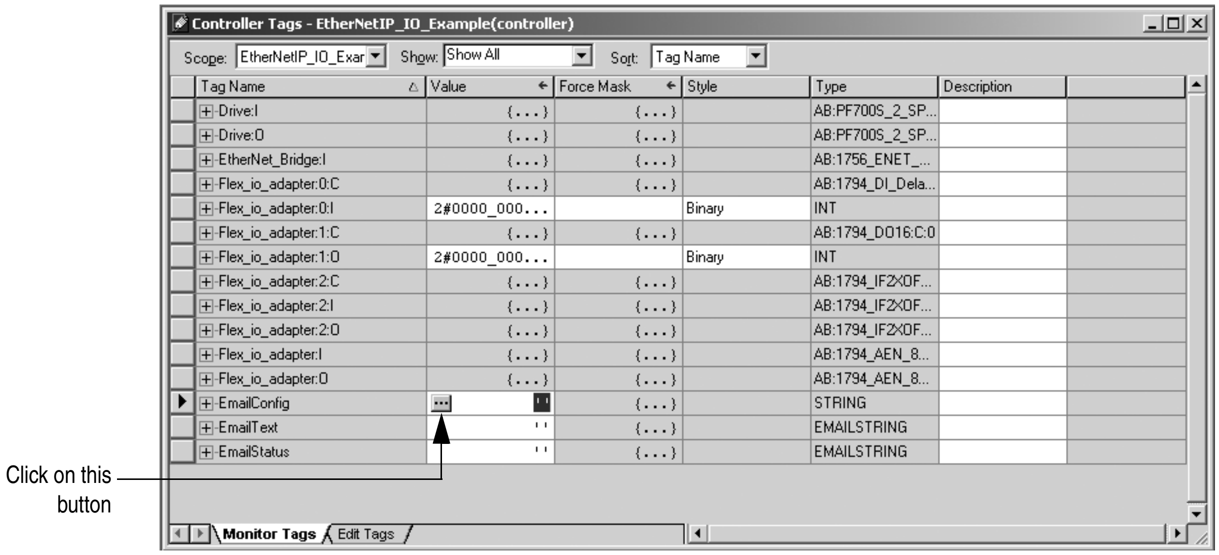
Scope: EtherNetIP_IO_Exar Show: Show All Sort: Tag Name

P	Tag Name	Alias For	Base Tag	Type	Style	Description
	⊞-Drive:I			AB:PF700S_2_SP...		
	⊞-Drive:O			AB:PF700S_2_SP...		
	⊞-EtherNet_Bridge:I			AB:1756_ENET_...		
	⊞-Flex_io_adapter:0:C			AB:1794_DI_Dela...		
	⊞-Flex_io_adapter:0:I	Flex_io_adapter:I....	Flex_io_adapter:I....	INT	Binary	
	⊞-Flex_io_adapter:1:C			AB:1794_DO16:C:0		
	⊞-Flex_io_adapter:1:O	Flex_io_adapter:O...	Flex_io_adapter:O...	INT	Binary	
	⊞-Flex_io_adapter:2:C			AB:1794_IF2XOF...		
	⊞-Flex_io_adapter:2:I			AB:1794_IF2XOF...		
	⊞-Flex_io_adapter:2:O			AB:1794_IF2XOF...		
	⊞-Flex_io_adapter:I			AB:1794_AEN_8...		
	⊞-Flex_io_adapter:O			AB:1794_AEN_8...		
☐	⊞-EmailConfig			STRING		
	⊞-EmailText			EMAILSTRING		
☐	EmailStatus			EMAILSTRING ...		
☐						

Entering String Data

On the Monitor Tags tab of the Controller Tags window, click the button in the Value column to launch the String Browser for each tag.

1. On the Monitor Tags tab of the Controller Tags window, click the button in the Value column to launch the String Browser.

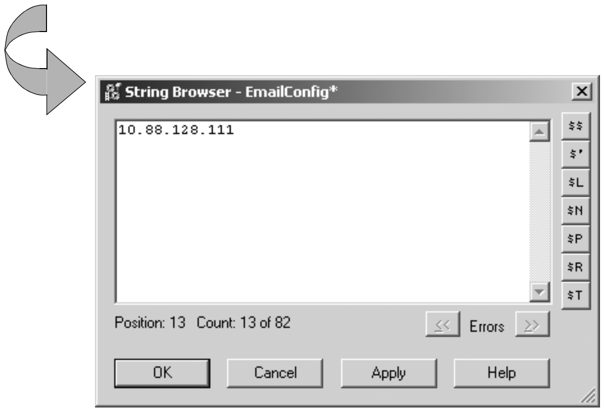


Controller Tags - EtherNetIP_IO_Example(controller)

Scope: EtherNetIP_IO_Exar Show: Show All Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type	Description
⊞-Drive:I	{...}	{...}		AB:PF700S_2_SP...	
⊞-Drive:O	{...}	{...}		AB:PF700S_2_SP...	
⊞-EtherNet_Bridge:I	{...}	{...}		AB:1756_ENET_...	
⊞-Flex_io_adapter:0:C	{...}	{...}		AB:1794_DI_Dela...	
⊞-Flex_io_adapter:0:I	2#0000_000...		Binary	INT	
⊞-Flex_io_adapter:1:C	{...}	{...}		AB:1794_DO16:C:0	
⊞-Flex_io_adapter:1:O	2#0000_000...		Binary	INT	
⊞-Flex_io_adapter:2:C	{...}	{...}		AB:1794_IF2XOF...	
⊞-Flex_io_adapter:2:I	{...}	{...}		AB:1794_IF2XOF...	
⊞-Flex_io_adapter:2:O	{...}	{...}		AB:1794_IF2XOF...	
⊞-Flex_io_adapter:I	{...}	{...}		AB:1794_AEN_8...	
⊞-Flex_io_adapter:O	{...}	{...}		AB:1794_AEN_8...	
⊞-EmailConfig	... ''	{...}		STRING	
⊞-EmailText	''	{...}		EMAILSTRING	
⊞-EmailStatus	''	{...}		EMAILSTRING	

2. In the String Browser for EmailConfig, enter the IP address for your facility's SMTP server.
Click Apply.
Click OK.

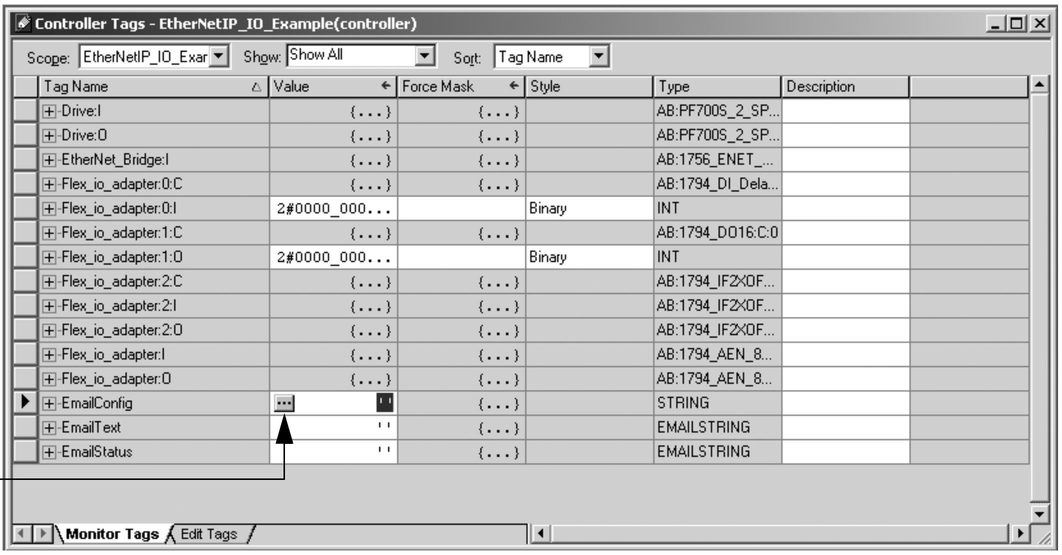


```
10.88.128.111
```

Position: 13 Count: 13 of 82

Important: Obtain this address from your company's systems administrators.

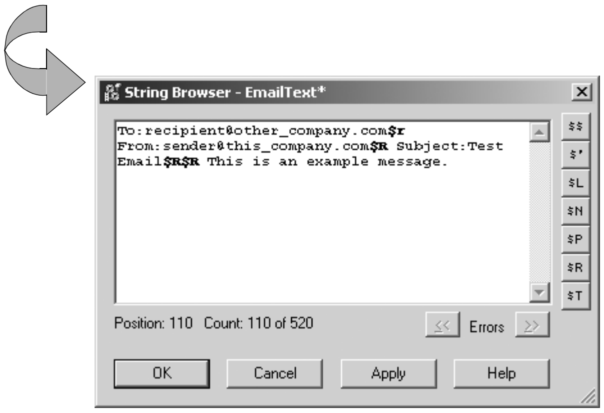
3. Repeat this process for EmailText.



Scope: EtherNetIP_IO_Exar Show: Show All Sort: Tag Name

Tag Name	Value	Force Mask	Style	Type	Description
+-Drive:I	{...}	{...}		AB:PF700S_2_SP...	
+-Drive:O	{...}	{...}		AB:PF700S_2_SP...	
+-EtherNet_Bridge:I	{...}	{...}		AB:1756_ENET_...	
+-Flex_io_adapter:0:C	{...}	{...}		AB:1794_DI_Dela...	
+-Flex_io_adapter:0:I	2#0000_000...		Binary	INT	
+-Flex_io_adapter:1:C	{...}	{...}		AB:1794_DO16:C:0	
+-Flex_io_adapter:1:O	2#0000_000...		Binary	INT	
+-Flex_io_adapter:2:C	{...}	{...}		AB:1794_IF2XOF...	
+-Flex_io_adapter:2:I	{...}	{...}		AB:1794_IF2XOF...	
+-Flex_io_adapter:2:O	{...}	{...}		AB:1794_IF2XOF...	
+-Flex_io_adapter:I	{...}	{...}		AB:1794_AEN_8...	
+-Flex_io_adapter:O	{...}	{...}		AB:1794_AEN_8...	
+-EmailConfig	... ''	{...}		STRING	
+-EmailText	''	{...}		EMAILSTRING	
+-EmailStatus	''	{...}		EMAILSTRING	

Click on this button



```
To:recipient@other_company.com$r
From:sender@this_company.com$R Subject:Test
Email$R$R This is an example message.
```

Position: 110 Count: 110 of 520

To include “To:”, “From:”, and “Subject:” fields in the email, use <CR><LF> symbols to separate each of these fields. The “To:” and “From”” fields are required; the “Subject:” field is optional. Use a second set of <CR><LF> symbols after the last one of these fields you enter. For example:

To: *email address* of recipient $r$l

From: *email address* of sender$r$l

Subject: *subject of message* $r$l$r$l

body of email message

Use the “From” address to specify where the mail relay server can send an undeliverable email message.

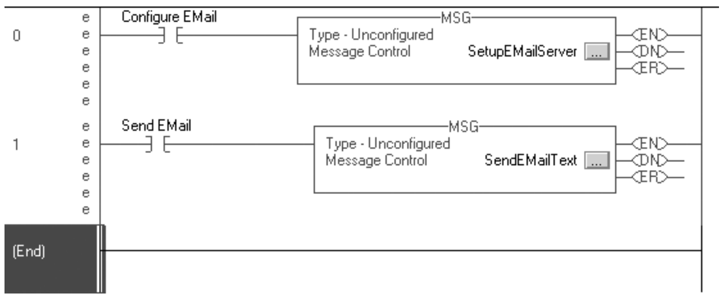
The maximum length of an email message is 474 characters. An additional 4-byte string-length value is added to the tag. As a result, the maximum source length is 478 characters.

Entering Ladder Logic

You need two MSG instructions. One MSG instruction configures the mail server. This only needs to be executed once. The second MSG instruction triggers the email transmission. This can be executed as often as needed.

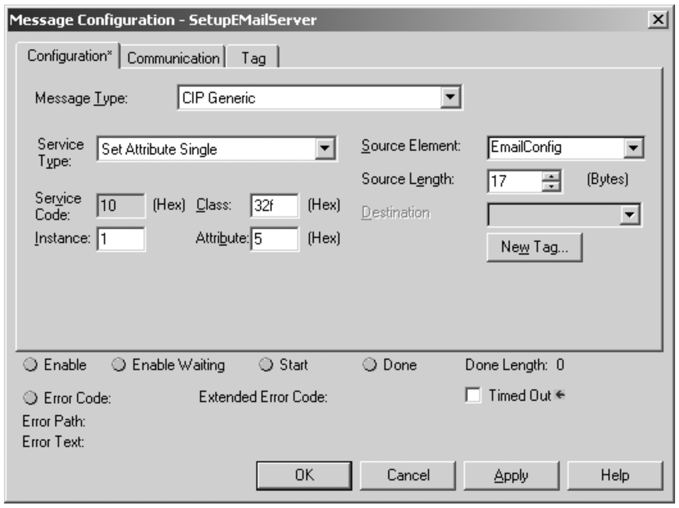
Important: Be careful to write the ladder logic to ensure the MSG instructions are not continuously triggered to send email messages.

This is an example of the basic MSG logic. You can build more complex logic that automatically manages the Source Lengthes within the MSG blocks. Refer to Using Ladder Logic to Dynamically Configure Messages on page 6-30.



Configuring the MSG Instruction that Identifies the Mail Relay Server

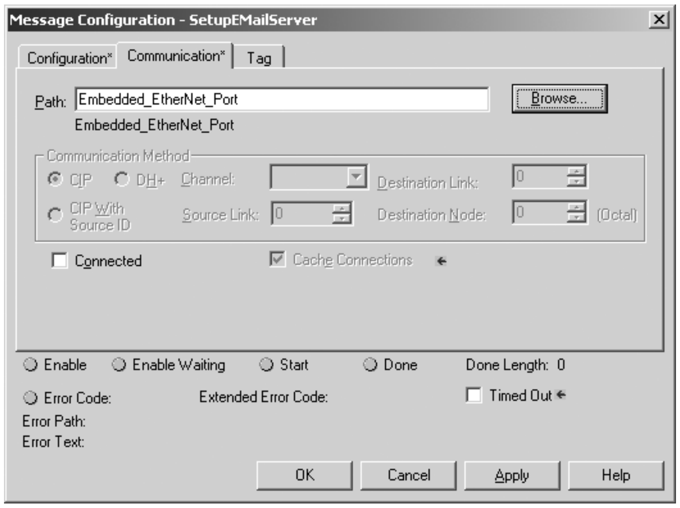
1. On the Configuration tab of the Message Configuration, fill out the following fields:



Message Type: CIP Generic
Service Type: Set Attribute Single
Source Element: EmailConfig
Source Length: 17 (Bytes)
Service Code: 10 (Hex) Class: 32f (Hex)
Destination
Instance: 1 Attribute: 5 (Hex)
New Tag...
Done Length: 0
Error Code: Extended Error Code:
Timed Out
Error Path:
Error Text:

In this field:	Enter:
Service Type	Set Attribute Single
Instance	1
Class	32f
Attribute	5
Source Element	the STRING tag that contains the IP address or host name of the mail relay server In this example, enter EmailConfig
Source Length	the number of characters in the IP address or host name of the mail server plus 4 In this example, enter 17 (13 characters in the IP address 10.88.128.111 + 4)

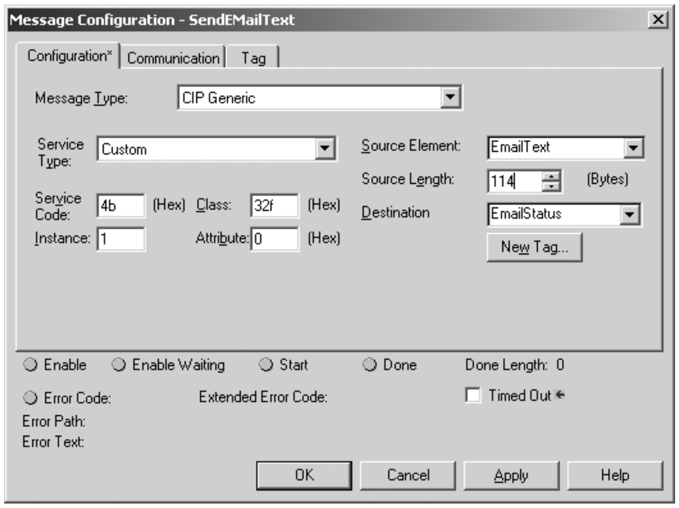
2. On the Communication tab of the Message Configuration, set the Path (e.g. browse for the Embedded EtherNet Port).
   Click Apply.
   Click OK.



Path: Embedded_EtherNet_Port
Embedded_EtherNet_Port
Communication Method
CIP DH+ Channel: Destination Link: 0
CIP With Source ID Source Link: 0 Destination Node: 0 (Octal)
Connected Cache Connections
Done Length: 0
Error Code: Extended Error Code:
Timed Out
Error Path:
Error Text:

Configuring the MSG Instruction that Transmits the Email

1. On the Configuration tab of the Message Configuration, fill out the following fields:


Message Configuration - SendEMailText

Configuration* | Communication | Tag

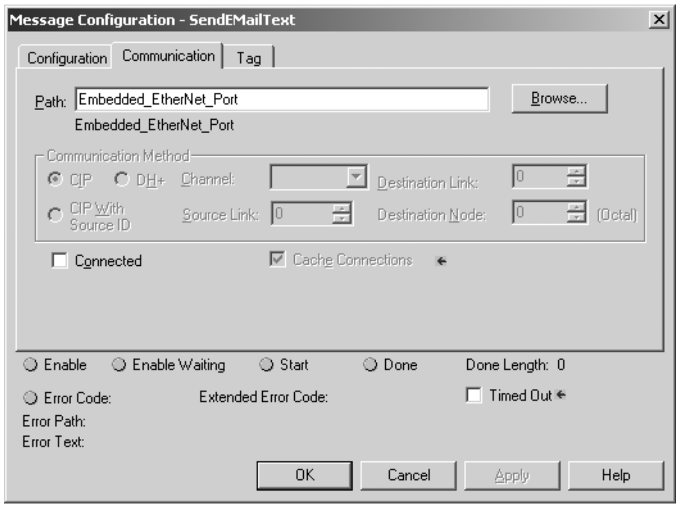
Message Type: CIP Generic

Service Type: Custom
Service Code: 4b (Hex) Class: 32f (Hex)
Instance: 1 Attribute: 0 (Hex)

Source Element: EmailText
Source Length: 114 (Bytes)
Destination: EmailStatus

In this field:	Enter:
Service Type	Custom
Service Code	4b
Instance	1
Class	32f
Attribute	0
Source Element	the tag that contains the email text This tag is of the STRING data type you created to contain the email text. In this example, EmailText.
Source Length	the number of characters in the email text plus 4 In this example, enter 114 (110 characters in the email + 4)
Destination	a tag to contain the status of the email transmission This tag is also of the STRING data type you created to contain the email text. In this example, EmailStatus.

2. On the Communication tab of the Message Configuration, set the Path in the same manner as in the MSG that identifies the mail relay server.
Click Apply.
Click OK.


Message Configuration - SendEMailText

Path: Embedded_EtherNet_Port
Embedded_EtherNet_Port

Communication Method: CIP (selected), DH+, CIP With Source ID
Channel:
Destination Link: 0
Source Link: 0
Destination Node: 0 (Octal)

Connected: unchecked
Cache Connections: checked

Enable, Enable Waiting, Start, Done
Done Length: 0
Error Code:
Extended Error Code:
Timed Out: unchecked
Error Path:
Error Text:

Understanding Email Status Codes

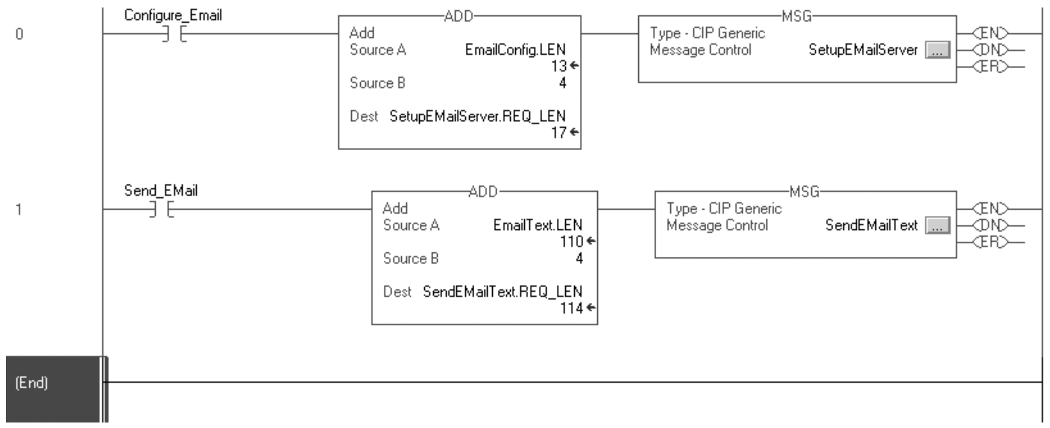
Examine the destination element of the email MSG to see whether the email was successfully delivered to the mail relay server. This indicates that the mail relay server placed the email message in a queue for delivery. It does not mean the intended recipient successfully received the email message. Possible codes that could be in this destination element are:

Error Code (hex):	Extended-Error Code (hex):	Description:
0x00	none	Delivery successful to the mail relay server.
0x02	none	Resource unavailable. The email object was unable to obtain memory resources to initiate the SMTP session.
0x08	none	Unsupported Service Request. Make sure the service code is 0x4B and the Class is 0x32F.
0x11	none	Reply data too large. The Destination string must reserve space for the SMTP server reply message. The maximum reply can be 470 bytes.
0x13	none	Configuration data size too short. The Source Length is less than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x15	none	Configuration data size too large. The Source Length is greater than the Source Element string size plus the 4-byte length. The Source Length must equal the Source Element string size + 4.
0x19	none	Data write failure. An error occurred when attempting to write the SMTP server address (attribute 4) to non-volatile memory.

Error Code (hex):	Extended-Error Code (hex):	Description:
0xFF	0x0100	Error returned by email server; check the Destination string for reason. The email message was not queued for delivery.
	0x0101	SMTP mail server not configured. Attribute 5 was not set with a SMTP server address.
	0x0102	"To:" address not specified. Attribute 1 was not set with a "To:" address AND there is not a "To:" field header in the email body.
	0x0103	"From:" address not specified. Attribute 2 was not set with a "From:" address AND there is not a "From:" field header in the email body.
	0x0104	Unable to connect to SMTP mail server set in Attribute 5. If the mail server address is a hostname, make sure that the device supports DNS, and that a Name Server is configured. If the hostname is not fully qualified, i.e., "mailhost" and not "mailhost.xx.yy.com" then the domain must be configured as "xx.yy.com". Try "ping <*mail server address*>" to insure the mail server is reachable from your network. Also try "telnet <*mail server address*> 25" which attempts to initiate a SMTP session with the mail server via telnet over port 25. (If you connect then enter "QUIT").
	0x0105	Communication error with SMTP mail server. An error occurred after the initial connection with the SMTP mail server. See the ASCII text following the error code for more details as to the type of error.
	0x0106	SMTP mail server host name DNS query did not complete. A previous send service request with a host name as the SMTP mail server address did not yet complete. Note that a timeout for a DNS lookup with an invalid host name can take up to 3 minutes. Long timeouts can also occur if a domain name or name server is not configured correctly.

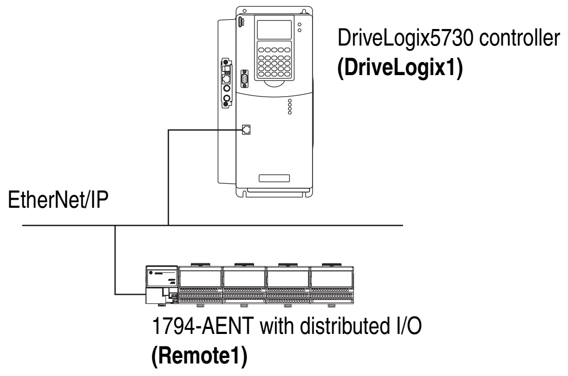
Using Ladder Logic to Dynamically Configure Messages

Use ADD instructions to calculate the MSG Source Length.



Example 1: DriveLogix5730 Controller and Distributed I/O

In the following example, one DriveLogix5730 controller controls distributed I/O through a 1794-AENT module.



Controlling Distributed I/O

This example has DriveLogix1 controlling the I/O connected to the remote 1794-AENT module. The data the DriveLogix5730 controller receives from the distributed I/O modules depends on how you configure the I/O modules. You can configure each module as a direct connection or as rack optimized. One chassis can have a combination of some modules configured as a direct connection and others as rack optimized.

All analog modules require direct connections. Diagnostic modules support rack-optimized connections, but require direct connections to take full advantage of their diagnostic features.

Total Connections Required by DriveLogix1

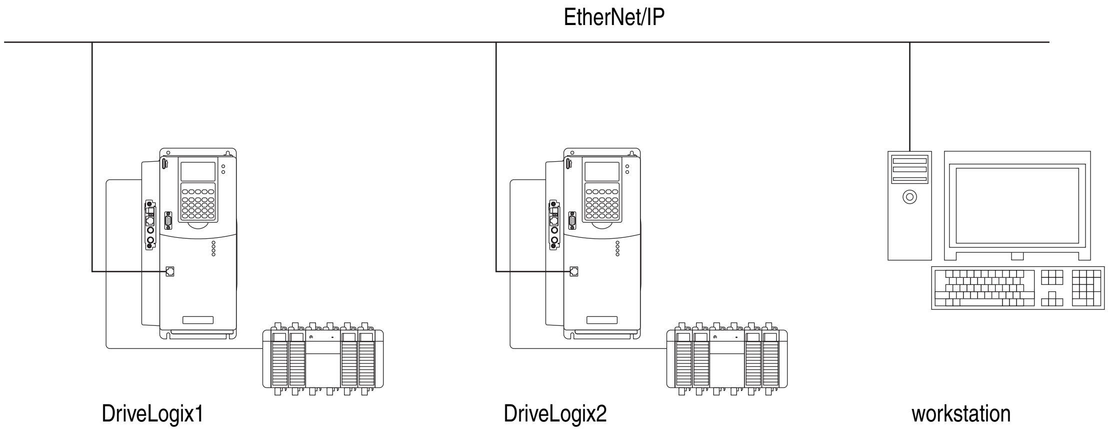
The following table calculates the connections used in this example.

Connection:	Amount:
DriveLogix1 to 4 distributed I/O modules (through 1794-AENT) all I/O modules configured as direct connection no connection to the 1794-AENT	4
total connections used:	4

If you configured the distributed I/O modules as rack-optimized, you would only need a rack-optimized connection to the 1794-AENT, reducing the above example by 3 connections.

Example 2: Controller to Controller

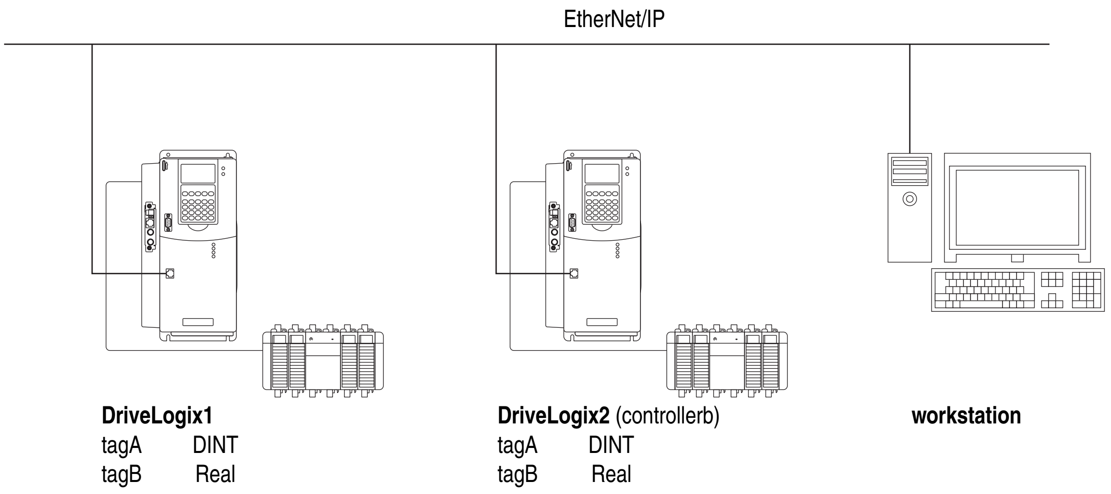
In the following example, one EtherNet/IP DriveLogix5730 controller communicates with another EtherNet/IP DriveLogix5730 controller over EtherNet/IP. Each controller has its own local I/O.



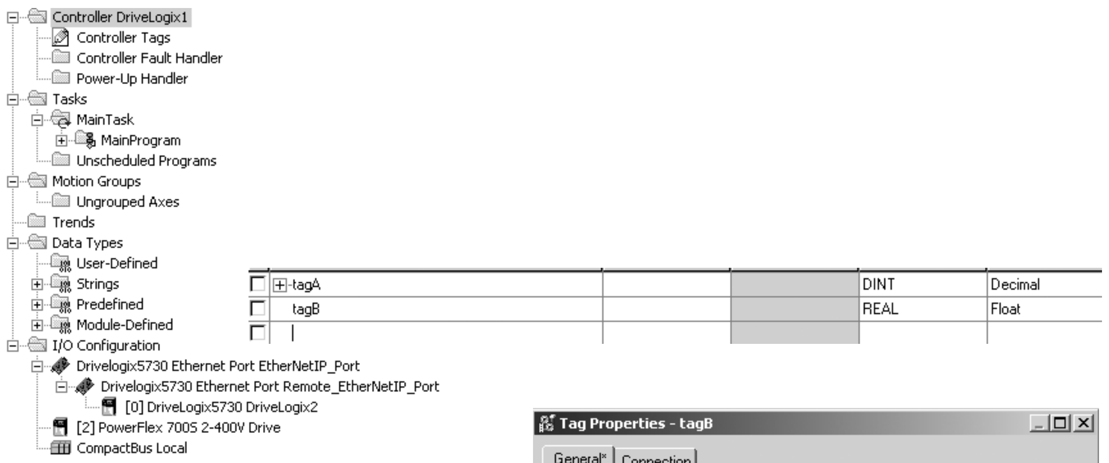
Producing and Consuming Tags

Produced data must be of DINT or REAL data type or an array or structure. You can use a user-defined structure to group BOOL, SINT, and INT data to be produced. You can produce a base, alias, or consumed tag.

The consumed tag must have the same data type as the produced tag in the originating controller. The controller performs type checking to ensure proper data is being received.



This example shows DriveLogix1 producing tagA and consuming tagB:



Controller DriveLogix1
- Controller Tags
- Controller Fault Handler
- Power-Up Handler

Tasks
- MainTask
  - MainProgram
- Unscheduled Programs

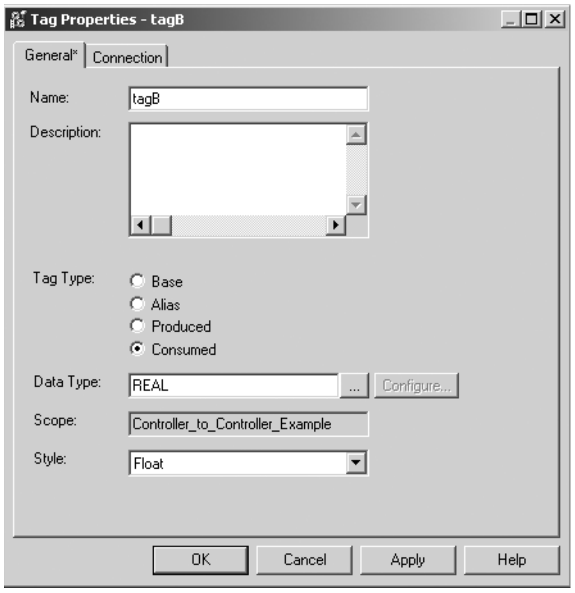
Motion Groups
- Ungrouped Axes

Trends

Data Types
- User-Defined
- Strings
- Predefined
- Module-Defined

I/O Configuration
- Drivelogix5730 Ethernet Port EtherNetIP_Port
  - Drivelogix5730 Ethernet Port Remote_EtherNetIP_Port
    - [0] DriveLogix5730 DriveLogix2
- [2] PowerFlex 700S 2-400V Drive
- CompactBus Local

tagA			DINT	Decimal
tagB			REAL	Float



Tag Properties - tagB

General* | Connection

Name: tagB

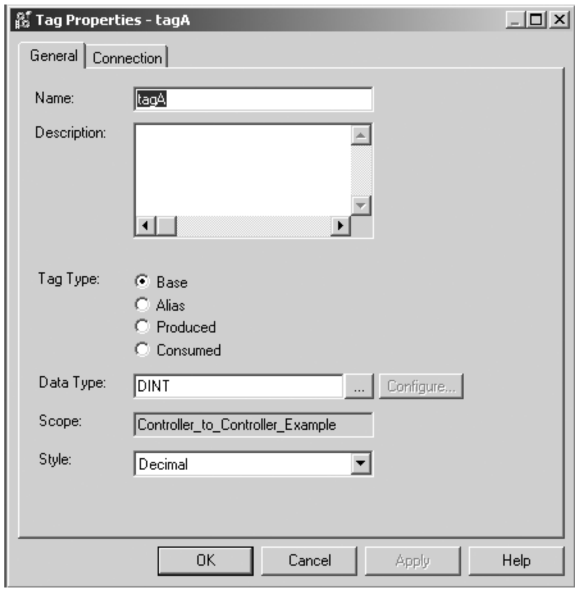
Description:

Tag Type: ( ) Base ( ) Alias ( ) Produced [✓ Consumed]

Data Type: REAL

Scope: Controller_to_Controller_Example

Style: Float



Tag Properties - tagA

General | Connection

Name: tagA

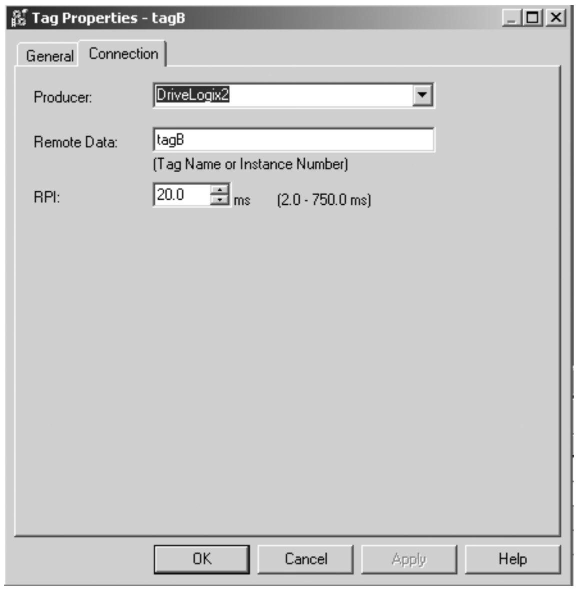
Description:

Tag Type: [✓ Base] ( ) Alias ( ) Produced ( ) Consumed

Data Type: DINT

Scope: Controller_to_Controller_Example

Style: Decimal



Tag Properties - tagB

General | Connection

Producer: DriveLogix2

Remote Data: tagB

(Tag Name or Instance Number)

RPI: 20.0 ms (2.0 - 750.0 ms)

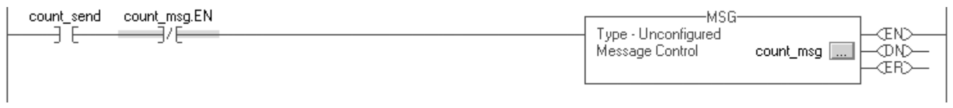
Each produced tags requires one connection for the producing controller and an additional connection for each consuming controller. Each consumed tag requires one connection.

Sending a MSG instruction

To send a MSG from DriveLogix1 to DriveLogix2:

1. For DriveLogix1, create a controller-scoped tag and select the MESSAGE data type.
2. Enter a MSG instruction.

In this example logic, a message is sent when a specific condition is met. When count_send is set, the ladder logic sends the count_msg MSG.



3. Configure the MSG instruction. On the Configuration tab:

For this item:	Specify:
Message Type	CIP Data Table Read or CIP Data Table Write
Source Tag	Tag containing the data to be transferred
Number of Elements	Number of array elements to transfer
Destination Tag	Tag to which the data will be transferred

4. On the Communication tab, specify the communication path.

   Use the Browse button to select the device that will receive the MSG instruction. The communication path in this example is:

For this item:	Specify:
Communication Path	1,1,2,xxx.xxx.xxx.xxx,1,0 where: 1 is the virtual backplane of DriveLogix1 1 is the slot of the Ethernet port in the controller (note, the 1,1 displays as LocalENB) 2 is the EtherNet/IP network 100.100.115.11 is the IP address of DriveLogix2 1 is the virtual backplane of DriveLogix2 0 is the controller slot of DriveLogix2

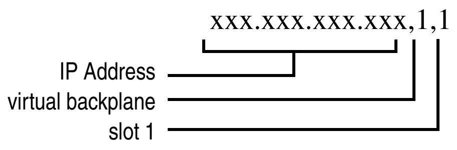
Total Connections Required by DriveLogix1

The following table calculates the connections used in this example.

Connection:	Amount:
connected, cached MSG from DriveLogix1 to DriveLogix2	1
produced tagA produced from DriveLogix1 to DriveLogix2 other consumer (2 are configured)	 1 1
consumed tagB	1
total connections used:	4

Receiving a MSG Instruction

When other devices send messages to the DriveLogix5730 controller, the path for the message must identify the controller. Configure a CIP-type message in the originating device. Specify the path the DriveLogix5730 controller as:



Notes:

Communicating with Devices on an ControlNet Link

De-energizing the Drive to Connect or Disconnect a Cable

⚠ **ATTENTION:** Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of configuring ControlNet communication you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive's input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with configuring ControlNet communication.

Configuring Your System for a ControlNet Link

For the DriveLogix5730 controller to operate on a ControlNet network, you need:

- a workstation with an appropriate ControlNet communication daughtercard
- a 1788-CNx communication daughtercard installed in the DriveLogix communication slot
- RSLinx software to configure the ControlNet communication driver
- RSLogix 5000 programming software to configure the 1788-CNx communication daughtercard as part of the DriveLogix system
- RSNetWorx software to schedule the DriveLogix system on the network

Configuring the Hardware

Before you can connect the DriveLogix system to the ControlNet network, you must configure the 1788-CNx communication daughtercard and make sure it's properly installed in the DriveLogix controller. Refer to Installing the Communications Daughtercard on page B-9.

You'll need to configure the communication daughtercard slot number to 1 in the RSLogix 5000 programming software. The DriveLogix controller uses slot 0 for the Communication Format with the local drive.

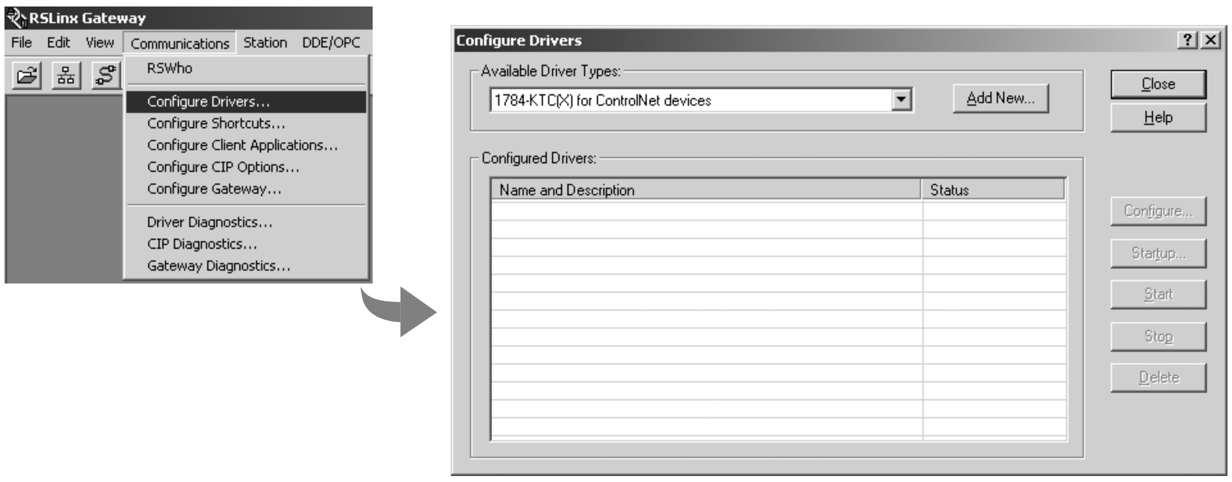
For more information about configuring a 1788-CNx communication daughtercard, see:

For this card:	See this document:
1788-CNC, -CNCR	1788-IN002
1788-CNF, -CNFR	1788-IN005

Configuring the Communication Driver

Use RSLinx software to configure the ControlNet communication driver. Select the appropriate communication driver for the communication daughtercard in your workstation.

1. In RSLinx software, select Configure Driver. Select the appropriate driver.



The installation instructions for the communications daughtercard should identify which communication driver to install.

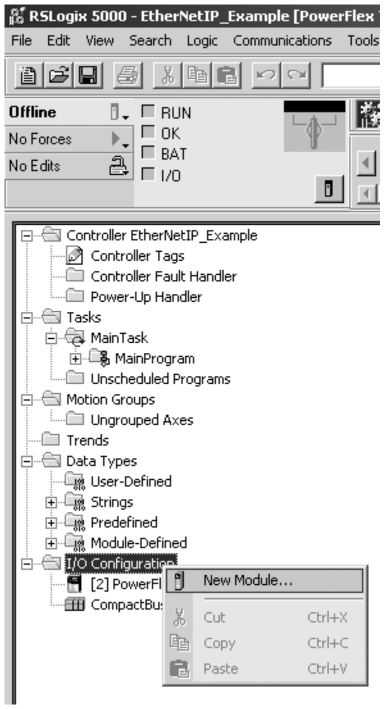
2. Specify the appropriate settings. For example.

If you are using this device:	Specify this information:
1784-KTCx card	memory address, which must match the switch setting on the card I/O base address, which must match the switch setting on the card ControlNet node address
1784-PCC card	ControlNet node address (MAC ID)
1784-PCIC card	ControlNet node address (MAC ID)

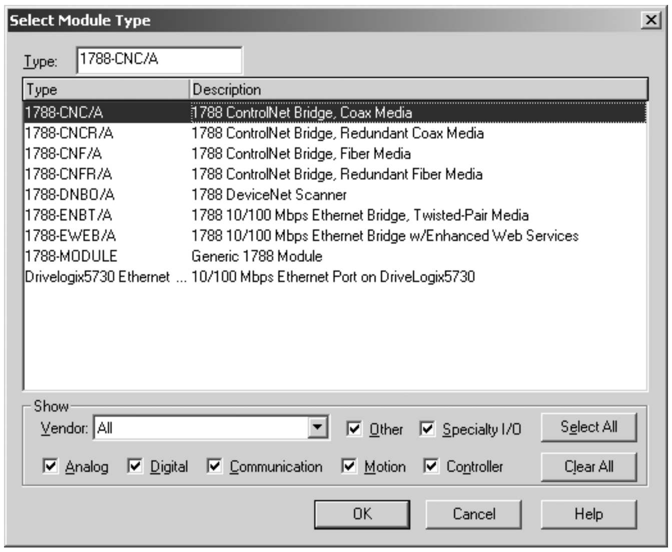
Configuring Your Daughtercard as Part of the System

Use RSLogix 5000 programming software to map the 1788-CNx communication daughtercard as part of the DriveLogix5730 system. In the Controller Organizer, add the communication daughtercard to the I/O Configuration folder.

1. In the Controller Organizer, select the I/O Configuration folder. Right-click the selected folder and select New Module.



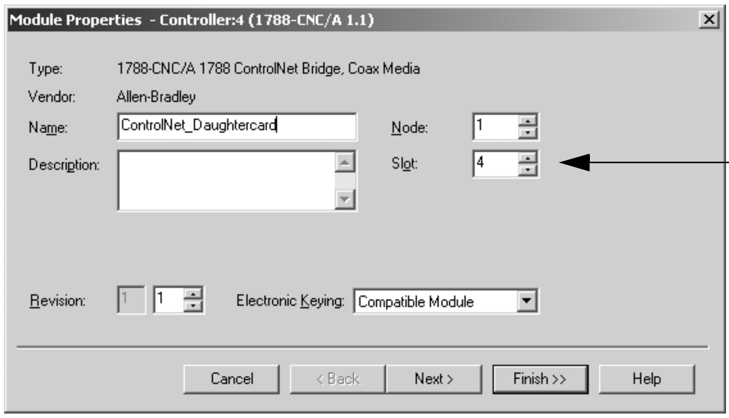
2. Select the proper ControlNet daughtercard from the list of possible communication devices. Click OK.



Type: 1788-CNC/A

Type	Description
1788-CNC/A	1788 ControlNet Bridge, Coax Media
1788-CNCR/A	1788 ControlNet Bridge, Redundant Coax Media
1788-CNF/A	1788 ControlNet Bridge, Fiber Media
1788-CNFR/A	1788 ControlNet Bridge, Redundant Fiber Media
1788-DNBO/A	1788 DeviceNet Scanner
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1788-MODULE	Generic 1788 Module
Drivelogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730

Show
Vendor: All
[✓] Other [✓] Specialty I/O
[✓] Analog [✓] Digital [✓] Communication [✓] Motion [✓] Controller



Type: 1788-CNC/A 1788 ControlNet Bridge, Coax Media
Vendor: Allen-Bradley
Name: ControlNet_Daughtercard
Node: 1
Description:
Slot: 4
Revision: 1 1
Electronic Keying: Compatible Module

Do not change the slot number. This must be slot 4.

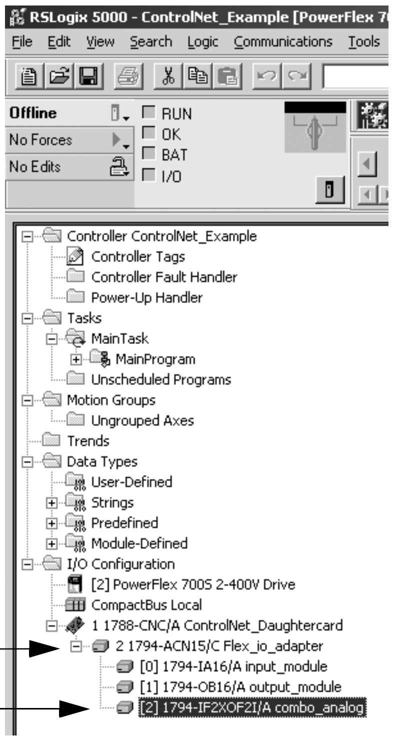
3. Enter a Name for the daughtercard and the desired. Click Finish.

Configuring Distributed I/O

The DriveLogix5730 controller supports distributed I/O over a ControlNet link. Configuring I/O in a remote chassis is similar to configuring local I/O. You create the remote communication module and distributed I/O modules on the local ControlNet daughtercard.

To communicate with distributed I/O modules, you add a remote adapter and I/O modules to the I/O Configuration folder of the controller.

To Add Distributed I/O Build the I/O Configuration in this Order


RSLogix 5000 - ControlNet_Example [PowerFlex 7
File Edit View Search Logic Communications Tools
Offline
No Forces
No Edits
RUN
OK
BAT
I/O

Controller ControlNet_Example
Controller Tags
Controller Fault Handler
Power-Up Handler
Tasks
MainTask
MainProgram
Unscheduled Programs
Motion Groups
Ungrouped Axes
Trends
Data Types
User-Defined
Strings
Predefined
Module-Defined
I/O Configuration
[2] PowerFlex 700S 2-400V Drive
CompactBus Local
1 1788-CNC/A ControlNet_Daughtercard
2 1794-ACN15/C Flex_io_adapter
[0] 1794-IA16/A input_module
[1] 1794-OB16/A output_module
[2] 1794-IF2XOF2I/A combo_analog

1. Add the remote adapter to the ControlNet daughtercard of the controller.

2. Add the I/O modules to the remote adapter.

Accessing Distributed I/O

I/O information is presented as a structure of multiple fields, which depend on the specific features of the I/O module. The name of the structure is based on the location of the I/O module in the system. Each I/O tag is automatically created when you configure the I/O module through the programming software. Each tag name follows this format:

Location:SlotNumber:Type.MemberName.SubMemberName.Bit

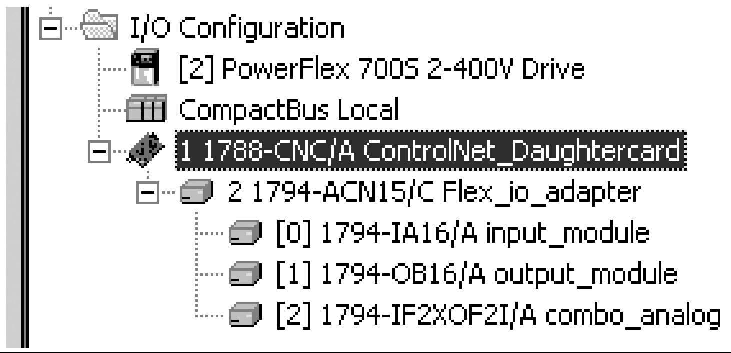
where:

This address variable:	**Is:**
Location	Identifies network location LOCAL = local DIN rail or chassis ADAPTER_NAME = identifies remote adapter or bridge
SlotNumber	Slot number of I/O module in its chassis

Type	Type of data
	I = input
	O = output
	C = configuration
	S = status

MemberName	Specific data from the I/O module; depends on the type of data the module can store For example, Data and Fault are possible fields of data for an I/O module. Data is the common name for values the are sent to or received from I/O points.
SubMemberName	Specific data related to a MemberName.
Bit (optional)	Specific point on the I/O module; depends on the size of the I/O module (0-31 for a 32-point module)

Example

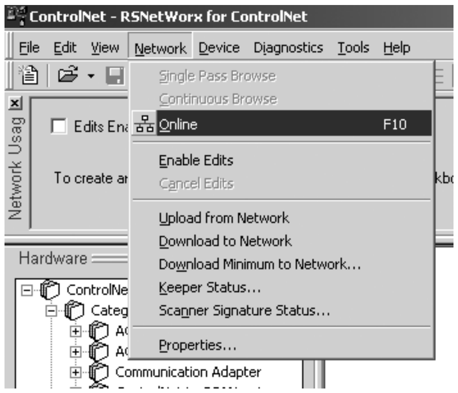

I/O Configuration
[2] PowerFlex 700S 2-400V Drive
CompactBus Local
1 1788-CNC/A ControlNet_Daughtercard
2 1794-ACN15/C Flex_io_adapter
[0] 1794-IA16/A input_module
[1] 1794-OB16/A output_module
[2] 1794-IF2XOF2I/A combo_analog

Device:	Example Tag Names (automatically created by the software):
remote adapter “FLEX_io_adapter”	FLEX_io_adapter:I FLEX_io_adapter:I.SlotStatusBits FLEX_io_adapter:I.Data FLEX_io_adapter:O FLEX_io_adapter:O.Data
remote “input_module” in slot 0 rack-optimized connection	FLEX_io_adapter:0:C FLEX_io_adapter:0:C.Config FLEX_io_adapter:0:C.DelayTime_0 FLEX_io_adapter:0:C.DelayTime_1 FLEX_io_adapter:0:C.DelayTime_2 FLEX_io_adapter:0:C.DelayTime_3 FLEX_io_adapter:0:C.DelayTime_4 FLEX_io_adapter:0:C.DelayTime_5 FLEX_io_adapter:0:I
remote “output_module” in slot 1 rack-optimized connection	FLEX_io_adapter:1:C FLEX_io_adapter:1:C.SSData FLEX_io_adapter:1:O FLEX_io_adapter:1:O.Data
remote “combo_analog” in slot 2 direct connection	FLEX_io_adapter:2:C FLEX_io_adapter:2:C.InputFIlter FLEX_io_adapter:2:C.InputConfiguration FLEX_io_adapter:2:C.OutputConfiguration FLEX_io_adapter:2:C.RTSInterval FLEX_io_adapter:2:C.SSCh0OuputData FLEX_io_adapter:2:C.SSCH1OutputData FLEX_io_adapter:2:I

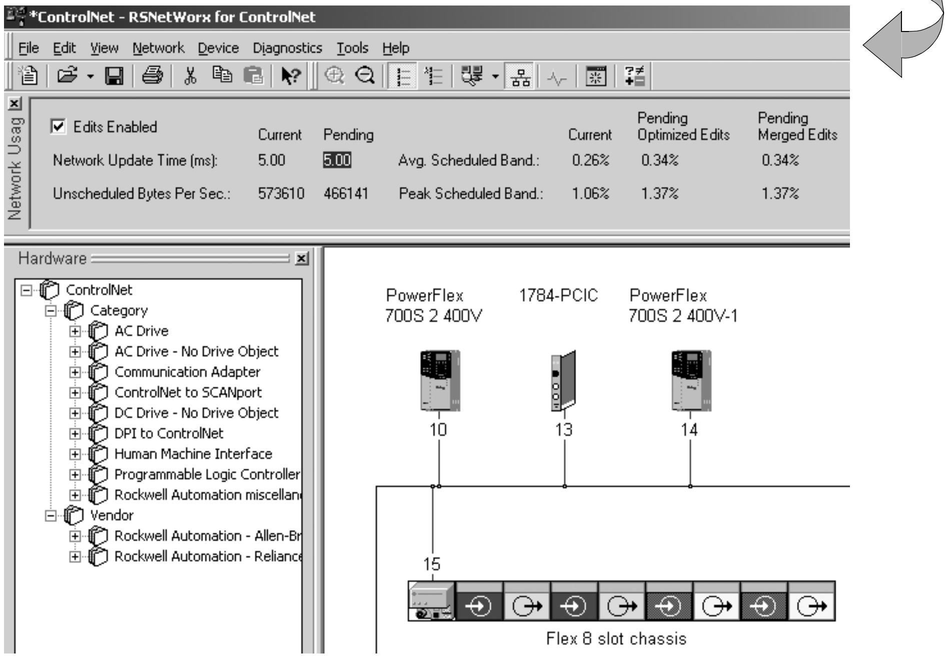
Scheduling the ControlNet Network

Use RSNetWorx software to schedule the ControlNet network. The controller project must already be downloaded from RSLogix 5000 programming software to the controller and the controller must be in Program or Remote Program mode.

1. In the Network menu, select Online.

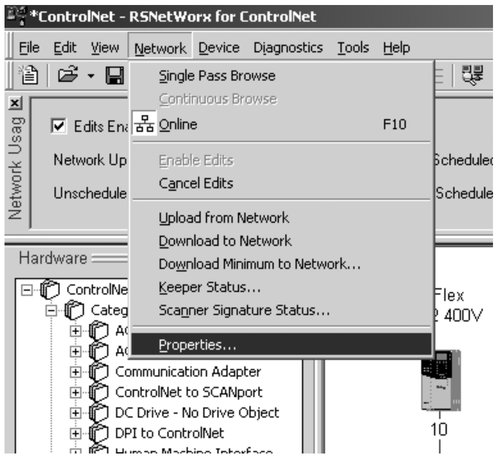

ControlNet - RSNetWorx for ControlNet
File Edit View Network Device Diagnostics Tools Help
Single Pass Browse
Continuous Browse
Online F10
Enable Edits
Cancel Edits
Upload from Network
Download to Network
Download Minimum to Network...
Keeper Status...
Scanner Signature Status...
Properties...
Hardware

2. Survey the network.
Enable Edits

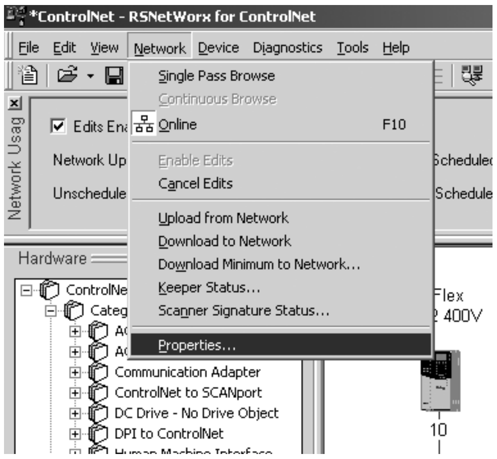

*ControlNet - RSNetWorx for ControlNet
File Edit View Network Device Diagnostics Tools Help
Edits Enabled
Hardware
ControlNet
Category
AC Drive
AC Drive - No Drive Object
Communication Adapter
ControlNet to SCANport
DC Drive - No Drive Object
DPI to ControlNet
Human Machine Interface
Programmable Logic Controller
Rockwell Automation miscellan
Vendor
Rockwell Automation - Allen-Br
Rockwell Automation - Reliance
PowerFlex 700S 2 400V
1784-PCIC
PowerFlex 700S 2 400V-1
10
13
14
15
Flex 8 slot chassis

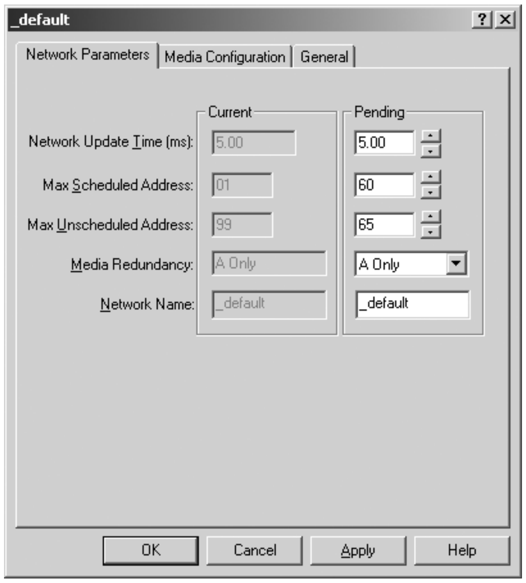
	Current	Pending		Current	Pending Optimized Edits	Pending Merged Edits
Network Update Time (ms):	5.00	5.00	Avg. Scheduled Band.:	0.26%	0.34%	0.34%
Unscheduled Bytes Per Sec.:	573610	466141	Peak Scheduled Band.:	1.06%	1.37%	1.37%

3. In the Network menu, select Properties.



3. In the Network menu, select Properties.

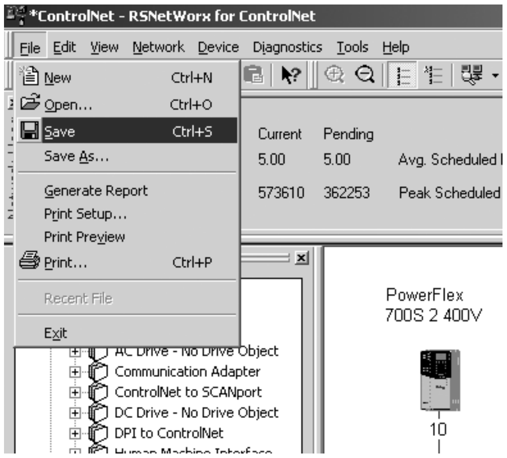




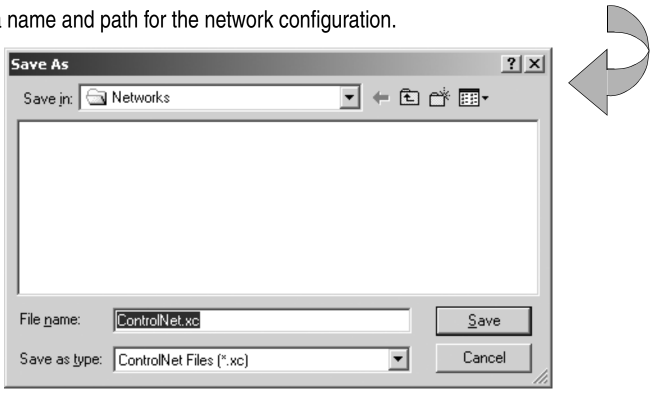
	Current	Pending
Network Update Time (ms):	5.00	5.00
Max Scheduled Address:	01	60
Max Unscheduled Address:	99	65
Media Redundancy:	A Only	A Only
Network Name:	_default	_default

4. Edit the parameters as desired.
Click Apply.
Click OK.

5. In the File menu, select Save.

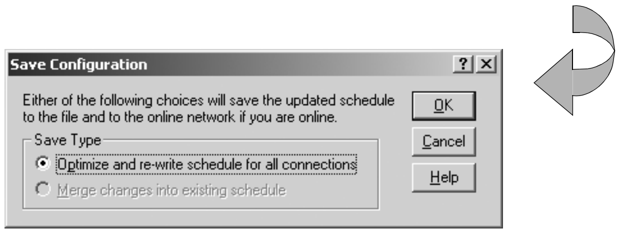


6. Enter a name and path for the network configuration.



File name: ControlNet.xc

Save as type: ControlNet Files (*.xc)



Either of the following choices will save the updated schedule to the file and to the online network if you are online.

Save Type

(●) Optimize and re-write schedule for all connections

( ) Merge changes into existing schedule

7. Click OK to optimize and reschedule the connections.

Sending Messages

The DriveLogix5730 controller can send MSG instructions to other controllers over a ControlNet link. Each MSG instruction requires you to specify a target and an address within the target. The number of messages that a device can support depends on the type of message and the type of device:

This device:	Support this many unconnected messages:	Support this many connected messages:
1756-CNB or 1756-CNBR module (for a Logix5550 controller)	20	64
1788-CNx daughtercard (for a DriveLogix controller)	5	32 with a maximum of 9 scheduled
ControlNet PLC-5 controller	32	128

MSG instructions are unscheduled. The type of MSG determines whether or not it requires a connection. If the MSG instruction requires a connection, it opens the needed connection when it is executed. You can configure the

MSG instruction to keep the connection open (cache) or to close it after sending the message.

This type of message:	And this communication method:	Uses a connection:
CIP data table read or write		X
PLC2, PLC3, PLC5, or SLC (all types)	CIP	
	CIP with Source ID	
	DH+	X
CIP generic	CIP	Optional(1)
block-transfer read or write		X

(1) You can connect CIP generic messages, but for most applications, we recommend you leave CIP generic messages unconnected.

Connected messages are unscheduled connections on ControlNet.

If a MSG instruction uses a connection, you have the option to leave the connection open (cache) or close the connection when the message is done transmitting.

If you:	Then:
Cache the connection	The connection stays open after the MSG instruction is done. This optimizes execution time. Opening a connection each time the message executes increases execution time.
Do not cache the connection	The connection closes after the MSG instruction is done. This frees up that connection for other uses.

<u>The following is from the 5720 manual and is a place holder. We need to put real information here!</u>
The controller has the following limits on the number of connections that you can cache:

If you have this software and firmware revision:	Then you can cache:
11.x or earlier	• block transfer messages for up to 16 connections • other types of messages for up to 16 connections
12.x or later	up to 32 connections

Communicating with Another Logix-Based Controller

All Logix-based controllers can use MSG instructions to communicate with each other. The following examples show how to use tags in MSG instructions between Logix-based controllers.

Type of MSG Instruction:	Example Source and Destination:	
Logix-based controller writes to Logix-based controller (CIP Data Table Write)	source tag destination tag	*array_1* *array_2*
Logix-based controller reads from Logix-based controller (CIP Data Table Read)	source tag destination tag	*array_1* *array_2*

The source and destination tags:

- must be controller-scoped tags.
- can be of any data type, except for AXIS, MESSAGE, or MOTION_GROUP.

Communicating with Other Controllers Over ControlNet

The DriveLogix5730 controller also uses MSG instructions to communicate with PLC and SLC controllers. The MSG instructions differ depending on which controller initiates the instruction.

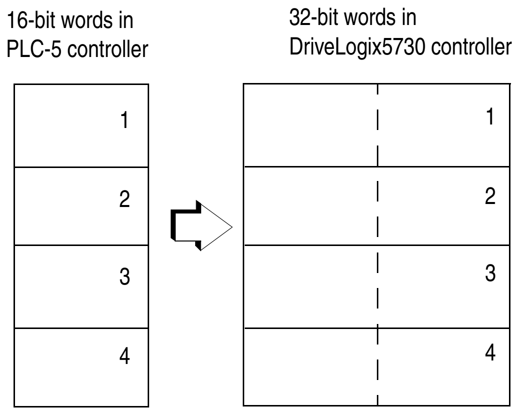
For MSG instructions originating from a DriveLogix5730 controller to a PLC or SLC controller:

Type of MSG Instruction:	Supported Source File Types:	Supported Destination File Types:
DriveLogix writes to PLC-5 or SLC	In the DriveLogix controller, specify the source data type based on the destination device: PLC-5: SINT, INT, DINT, or REAL SLC: INT Example source element: *array_1*	Specify the destination file type based on the destination device: PLC-5 typed write: S, B, N, or F PLC-5 word-range write: S, B, N, F, I, O, A, or D SLC: B or N Example destination tag: *N7:10*

Type of MSG Instruction:	Supported Source File Types:	Supported Destination File Types:
DriveLogix writes to PLC-2	In the DriveLogix controller, select one of these data types: SINT, INT, DINT, or REAL Example source element: *array_1*	Use the PLC-2 compatibility file. Example destination tag: *010*
DriveLogix reads from PLC-5 or SLC	Specify the destination file type based on the destination device: PLC-5 typed read: S, B, N, or F PLC-5 word-range read: S, B, N, F, I, O, A, or D SLC: B or N Example source element: *N7:10*	In the DriveLogix controller, specify the destination data type based on the destination device: PLC-5: SINT, INT, DINT, or REAL SLC: INT Example destination tag: *array_1*
DriveLogix reads from PLC-2	Use the PLC-2 compatibility file. Example source element: *010*	In the DriveLogix controller, select one of these data types: SINT, INT, DINT, or REAL Example destination tag: *array_1*

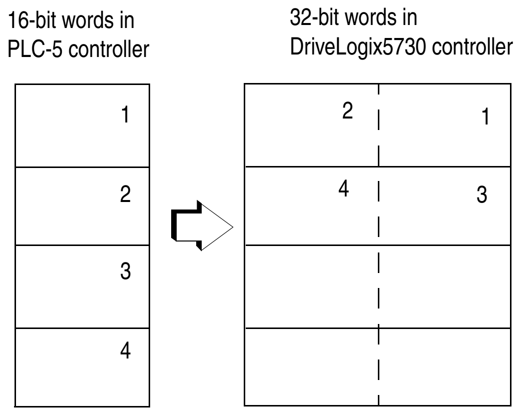
The DriveLogix5730 controller can send typed or word-range commands to PLC-5 controllers. These commands read and write data differently. The following diagrams show how the typed and word-range commands differ.

Typed read command


16-bit words in PLC-5 controller
32-bit words in DriveLogix5730 controller
1
2
3
4
1
2
3
4

The typed commands maintain data structure and value.

Word-range read command


16-bit words in PLC-5 controller
32-bit words in DriveLogix5730 controller
1
2
3
4
2
1
4
3

The word-range commands fill the destination tag contiguously. Data structure and value change depending on the destination data type.

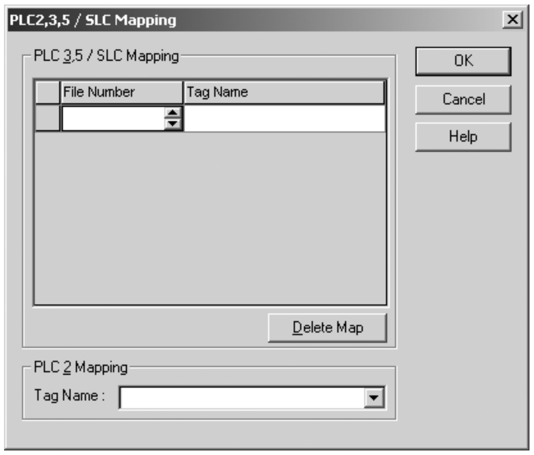
The DriveLogix5730 controller can process messages initiated from PLC or SLC controllers. These messages use data table addresses. In order for these controllers to access tags within the DriveLogix5730 controller, you map tags to data table addresses.

Mapping Addresses

The programming software includes a PLC/SLC mapping tool which allows you to make an existing controller array tag in the local controller available to PLC-2, PLC-3, PLC-5, or SLC controllers.

To map addresses:

1. From the Logic menu, select Map PLC/SLC Messages.



2. Specify this information:

For:	In this field:	Specify:	For example:
PLC-3, PLC-5, and SLC controllers	File Number	Type the file number of the data table in the PLC/SLC controller.	*10*
	Tag Name	Type the array tag name the local controller uses to refer to the PLC/SLC data table address. The tag must be an integer array (SINT, INT, or DINT) that is large enough for the message data.	*array_1*
PLC-2 controllers	Tag Name	Type the tag name to be the PLC-2 compatibility file.	*200*

▶ **TIP:** You can map as many tags as you want to a PLC-3, PLC-5, or SLC controller. You can map only one tag to a PLC-2 controller.

The following table shows example source and destination tags and elements for different controller combinations.

Type of MSG Instruction:	Example Source and Destination:	
PLC-5 writes to DriveLogix5730 SLC writes to DriveLogix5730 SLC 5/05 SLC 5/04 OS402 and above SLC 5/03 OS303 and above	source element	*N7:10*
	destination tag	*"array_1"*
	The PLC-5, PLC-3, and SLC controllers support logical ASCII addressing so you do not have to map a compatibility file for MSG instructions initiated by a PLC-5, PLC-3, or SLC controller. Place the DriveLogix5730 tag name in double quotes ("). You could optionally map a compatibility file. For example, if you enter *10* for the compatibility file, you enter *N10:0* for the destination tag.	
PLC-2 writes to DriveLogix5730	source element	*010*
	destination tag	*200*
	The destination tag is the three-digit PLC-2 address you specified for PLC-2 mapping.	

Type of MSG Instruction:	Example Source and Destination:	
PLC-5 reads from DriveLogix5730 SLC reads from DriveLogix5730 SLC 5/05 SLC 5/04 OS402 and above SLC 5/03 OS303 and above	source tag	*“array_1”*
	destination element	*N7:10*
	The PLC-5, PLC-3, and SLC controllers support logical ASCII addressing so you do not have to map a compatibility file for MSG instructions initiated by a PLC-5, PLC-3, or SLC controller. Place the DriveLogix5730 tag name in double quotes (“). You could optionally map a compatibility file. For example, if you enter *10* for the compatibility file, you enter *N10:0* for the source tag.	
PLC-2 reads from DriveLogix5730	source tag	*200*
	destination element	*010*
	The source tag is the three-digit PLC-2 address you specified for PLC-2 mapping.	

When the DriveLogix5730 controller initiates messages to PLC or SLC controllers, you do not have to map compatibility files. You enter the data table address of the target device just as you would a tag name.

SLC 5/05 controllers, SLC 5/04 controllers (OS402 and above), and SLC 5/03 controllers (OS303 and above) support logical ASCII addressing and support PLC/SLC mapping (see the examples above). For all other SLC or MicroLogix1000 controllers, you must map a PLC-2 compatibility file (see the PLC-2 examples above).

Producing and Consuming Data

The DriveLogix5730 controller supports the ability to produce (broadcast) and consume (receive) system-shared tags over a ControlNet link. Produced and consumed data is accessible by multiple controllers over a ControlNet network. Produced and consumed data are scheduled connections because the controller sends or receives data at a predetermined RPI rate.

Produced and consumed tags must be controller-scoped tags of DINT or REAL data type, or in an array or structure.

Tag type:	Description:	Specify:
produced	These are tags that the controller produced for other controllers to consume.	Enabled for producing How many consumers allowed
consumed	These are tags whose values are produced by another controller.	Controller name that owns the tag that the local controller wants to consume Tag name or instance that the controller wants to consume Data type of the tag to consume Update interval of how often the local controller consumes the tag

The producer and consumer must be configured correctly for the specified data to be shared. A produced tag in the producer must be specified exactly the same as a consumed tag in the consumer.

If any produced/consumed tag between a producer and consumer is not specified correctly, none of the produced/consumed tags for that producer and consumer will be transferred. However, other consumers can still access their shared tags, as long as their tags are specified correctly. One consumer failing to access shared data does not affect other consumers accessing the same data.

Maximum Number of Produced and Consumed Tags

The maximum number of produced/consumed tags that you can configure depends on the connection limits of the communication device that transfers the produced/consumed data.

Each produced tag uses one connection for the tag and the first configured consumer of the tag. Each consumer thereafter uses an additional connection.

Size Limit of a Produced or Consumed Tag

A produced or consumed tag can be as large as 488 bytes, but it must also fit within the bandwidth of the ControlNet network:

- As the number of connections over a ControlNet network increases, several connections, including produced or consumed tags, may need to share a network update.
- Since a ControlNet network can only pass 500 bytes in one update, the data of each connection must be less than 488 bytes to fit into the update.

If a produced or consumed tag is too large for your ControlNet network, make one or more of the following adjustments:

- Reduce the Network Update Time (NUT). At a faster NUT, less connections have to share an update slot.
- Increase the Requested Packet Interval (RPI) of all connections. At a higher RPI, connections can take turns sending data during an update slot.
- For a ControlNet bridge module (CNB) in a remote chassis, select the most efficient communication format for that chassis:

Are most of the modules in the chassis non-diagnostic, digital I/O modules?	Then select this communication format for the remote communication module:
yes	rack optimization
no	none

The Rack Optimization format uses an additional 8 bytes for each slot in its chassis. Analog modules or modules that are sending or getting diagnostic, fuse, or timestamp data require direct connections and cannot take advantage of the rack optimized form. Selecting “None” frees up the 8 bytes per slot for other uses, such as produced or consumed tags.

- Separate the tag into two or more smaller tags:
  - Group the data according to similar update rates. For example, you could create one tag for data that is critical and another tag for data that is not as critical.
  - Assign a different RPI to each tag.
- Create logic to transfer the data in smaller sections (packets).

Producing a Tag

Produced data must be of DINT or REAL data type or an array or structure. You can use a user-defined structure to group BOOL, SINT, and INT data to be produced. To create a produced tag:

1. You must be programming offline.
2. In the controller organizer, double-click the Controller Tags folder and then click the Edit Tags tab.
3. Select the tag that you want to produce, or enter a new tag, and display the Tag Properties dialog box.
4. Make sure the tag is controller scope.
5. Select the “Produce this tag” check box. Specify how many controllers can consume the tag.

You can produce a base, alias, or consumed tag.

The consumed tag in a receiving controller must have the same data type as the produced tag in the originating controller. The controller performs type checking to ensure proper data is being received.

Produced tags require connections. The number of connections depends on how many controllers are consuming the tags. The controller requires one connection for the produced tag and the first consumer. Then, the controller requires an additional connection for each subsequent consumer.

Consuming a Tag

A consumed tag represents data that is produced (broadcast) by one controller and received and stored by the consuming controller. To create a consumed tag:

1. You must be programming offline.
2. In the controller organizer, double-click the Controller Tags folder and then click the Edit Tags tab.
3. Select the tag that you want to consume, or enter a new tag, and display the Tag Properties dialog box.
4. Specify:

In this field:	Type or select:
Tag Type	Select Consumed.
Controller	Select the name of the other controller. You must have already created the controller in the controller organizer for the controller name to be available.
Remote Tag Name Remote Instance	Type a name for the tag in the other controller you want to consume. **Important:** The name must match the name in the remote controller exactly, or the connection faults. If the remote controller is a ControlNet PLC-5, this field is Remote Instance. Select the instance number (1-128) of the data on the remote controller.
RPI (requested packet interval)	Type the amount of time in msec between updates of the data from the remote controller. The local controller will receive data at least this fast.
Display Style	If you are creating a consumed tag that refers to a tag whose data type is BOOL, SINT, INT, DINT, or REAL, you can select a display style. This display style defines how the tag value will be displayed in the data monitor and ladder editor. The display style does not have to match the display style of the tag in the remote controller.

All consumed tags are automatically controller-scope.

To consume data from a remote controller, use RSNetWorx software to schedule the connection over the ControlNet network.

The produced tag in the originating DriveLogix controller must have the same data type as the consumed tag in the other DriveLogix controller. The DriveLogix controller performs type checking to ensure proper data is being received.

Important: If a consumed-tag connection fails, all of the other tags being consumed from that remote controller stop receiving data.

Guidelines for Configuring Connections

The 1788-CNx communication daughtercard supports 9 scheduled connections. How you configure these connections determines how many devices the daughtercard can support.

The NUT and RPI also play a part in determining how many connections a 1788-CNx can support in a given application, assuming the RPIs will be the same for all connections. You must also make sure that you do not exceed the maximum number of bytes per NUT.

- With the NUT = 5ms, the limit is 3 connections
- With the NUT = 10ms, the limit is 7connections
- With the NUT > 20ms, the limit is 9 connections

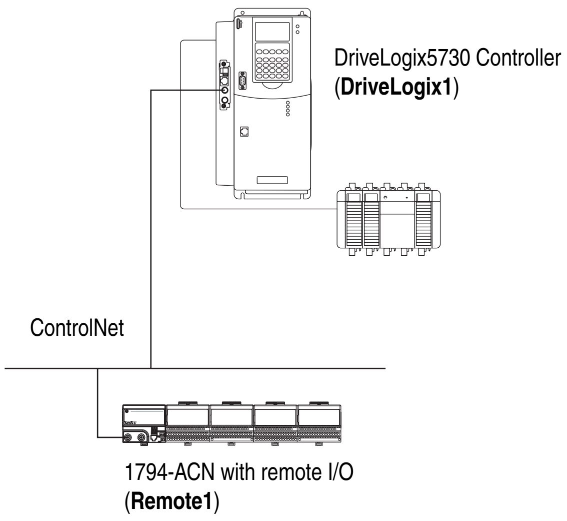
Determining the API

The API (actual packets per interval) is related to the RPI for the connection and the NUT of the network. Use this table to select the API to enter in the above worksheet:

If:	Enter this value for the API:
RPI ≥ NUT and RPI < 2∗NUT	NUT
RPI ≥ 2∗NUT and RPI < 4∗NUT	2∗NUT
RPI ≥ 4∗NUT and RPI < 8∗NUT	4∗NUT
RPI ≥ 8∗NUT and RPI < 16∗NUT	8∗NUT
RPI ≥ 16∗NUT and RPI < 32∗NUT	16∗NUT
RPI ≥ 32∗NUT and RPI < 64∗NUT	32∗NUT
RPI ≥ 64∗NUT and RPI < 128∗NUT	64∗NUT
RPI ≥ 128∗NUT	128∗NUT

Example 1: DriveLogix Controller and Remote I/O

In the following example, one DriveLogix5730 controller controls remote I/O through a 1794-ACN15 module.



Example 1: Controlling Remote Devices

This example has DriveLogix1 controlling the I/O connected to the remote 1794-ACN15 module. The data the DriveLogix controller receives from the remote I/O modules depends on how you configure the remote I/O modules. You can configure each module as a direct connection or as rack optimized. One chassis can have a combination of some modules configured as a direct connection and others as rack optimized.

Example 1: Total Connections Required by DriveLogix1

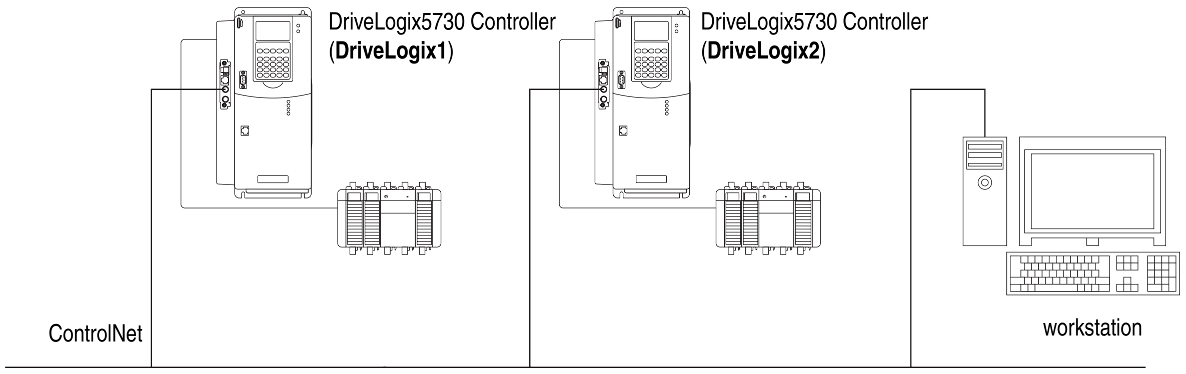
The following table calculates the connections used in this example.

Connection:	Amount:
DriveLogix1 controller to 3 local I/O modules rack-optimized connection for the DIN rail direct connection for each I/O module	 1 3
DriveLogix1 controller to remote 1794-ACNR15	1
DriveLogix1 to 4 remote I/O modules (through 1794-ACNR15) all I/O modules configured as direct connection no connection to the 1794-ACNR15	4
total connections used:	9

If you configured the remote I/O modules as rack-optimized, you would only need a rack-optimized connection to the 1794-ACNR15, reducing the above example by 3 connections.

Example 2: DriveLogix5730 Controller to DriveLogix5730 Controller

In the following example, one DriveLogix5730 controller communicates with another DriveLogix5730 controller over ControlNet. Each DriveLogix controller has its own local I/O.



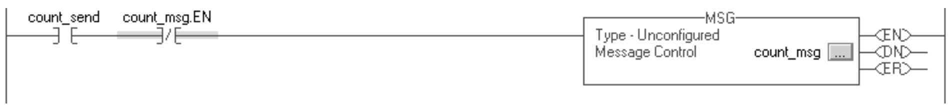
Example 2: Sending a MSG Instruction

To send a MSG from DriveLogix1 to DriveLogix2:

1. For DriveLogix1, create a controller-scoped tag and select the MESSAGE data type.

2. Enter a MSG instruction.

   In this example logic, a message is sent when a specific condition is met. When count_send is set, the ladder logic sends the count_msg MSG.



3. Configure the MSG instruction. On the Configuration tab:

For this item:	Specify:
Message Type	CIP Data Table Read or CIP Data Table Write
Source Tag	Tag containing the data to be transferred
Number of Elements	Number of array elements to transfer
Destination Tag	Tag to which the data will be transferred

4. On the Communication tab, specify the communication path.

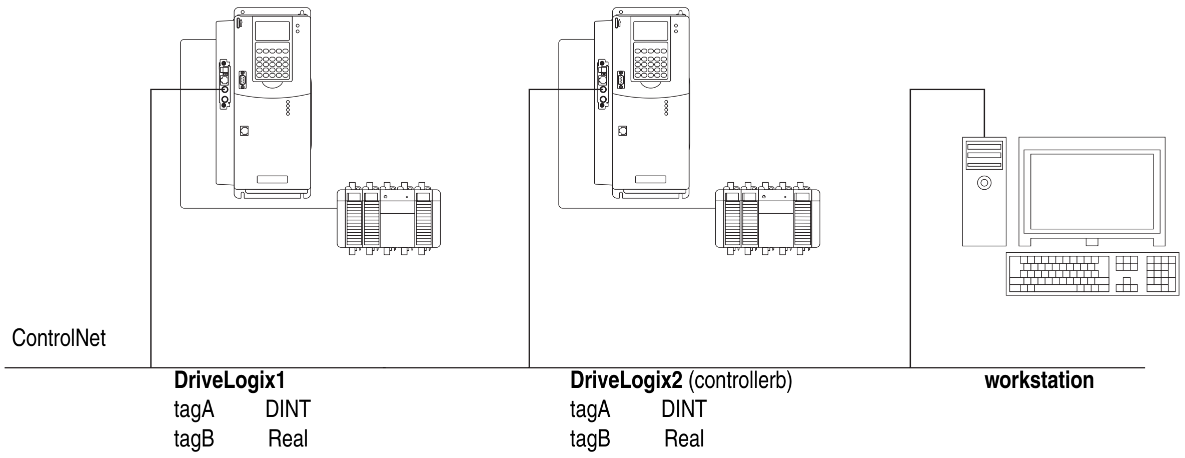
A communication path requires pairs of numbers. The first number in the pair identifies the port from which the message exits. The second number in the pair designates the node address of the next device.

For this item:	Specify:
Communication Path	1,4,2,27,1,0 where: 1 is the virtual backplane of DriveLogix1 4 is 1788-CNC daughtercard in slot 1 2 is the ControlNet port 27 is the ControlNet node of DriveLogix2 1 is the virtual backplane of DriveLogix2 0 is the controller slot of DriveLogix2

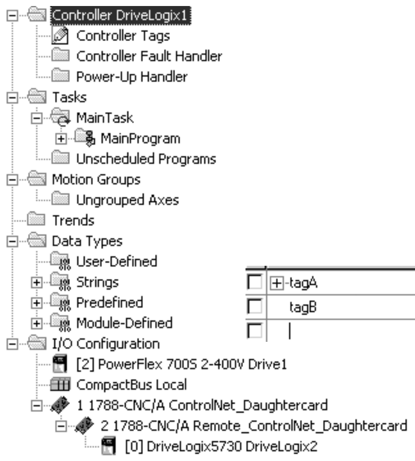
Example 2: Producing and Consuming Tags

Produced data must be of DINT or REAL data type or an array or structure. You can use a user-defined structure to group BOOL, SINT, and INT data to be produced. You can produce a base, alias, or consumed tag.

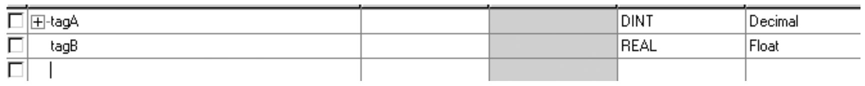
The consumed tag must have the same data type as the produced tag in the originating controller. The controller performs type checking to ensure proper data is being received.



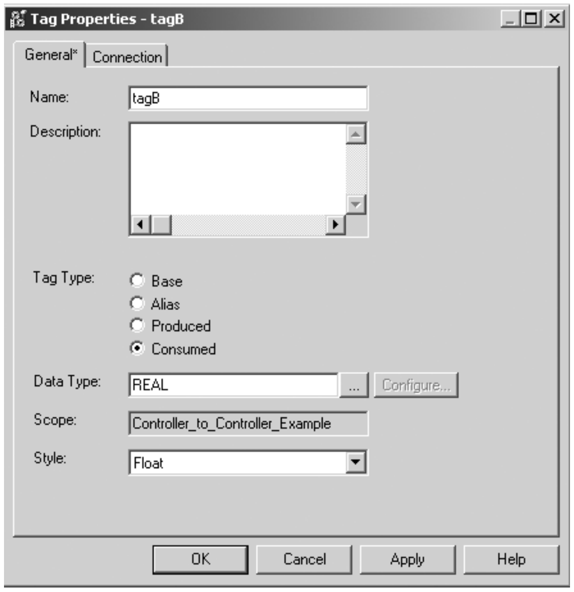
This example shows DriveLogix1 producing tagA and consuming tagB:



```
Controller DriveLogix1
    Controller Tags
    Controller Fault Handler
    Power-Up Handler
Tasks
    MainTask
        MainProgram
    Unscheduled Programs
Motion Groups
    Ungrouped Axes
Trends
Data Types
    User-Defined
    Strings
    Predefined
    Module-Defined
I/O Configuration
    [2] PowerFlex 700S 2-400V Drive1
    CompactBus Local
    1 1788-CNC/A ControlNet_Daughtercard
        2 1788-CNC/A Remote_ControlNet_Daughtercard
            [0] DriveLogix5730 DriveLogix2
```



tagA				DINT	Decimal
tagB				REAL	Float



Tag Properties - tagB

Name: tagB

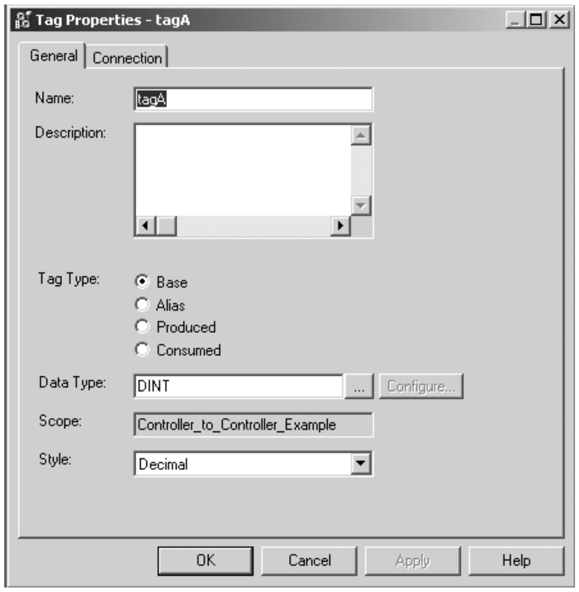
Description:

Tag Type: Base / Alias / Produced / [✓ Consumed]

Data Type: REAL

Scope: Controller_to_Controller_Example

Style: Float



Tag Properties - tagA

Name: tagA

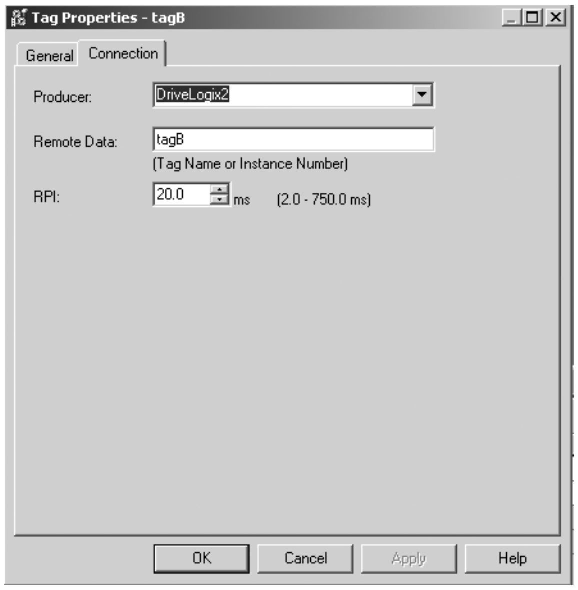
Description:

Tag Type: [✓ Base] / Alias / Produced / Consumed

Data Type: DINT

Scope: Controller_to_Controller_Example

Style: Decimal



Tag Properties - tagB

Producer: DriveLogix2

Remote Data: tagB
(Tag Name or Instance Number)

RPI: 20.0 ms (2.0 - 750.0 ms)

Each produced tags requires one connection for the producing controller and an additional connection for each consuming controller. Each consumed tag requires one connection.

Example 2: Total connections required by DriveLogix1

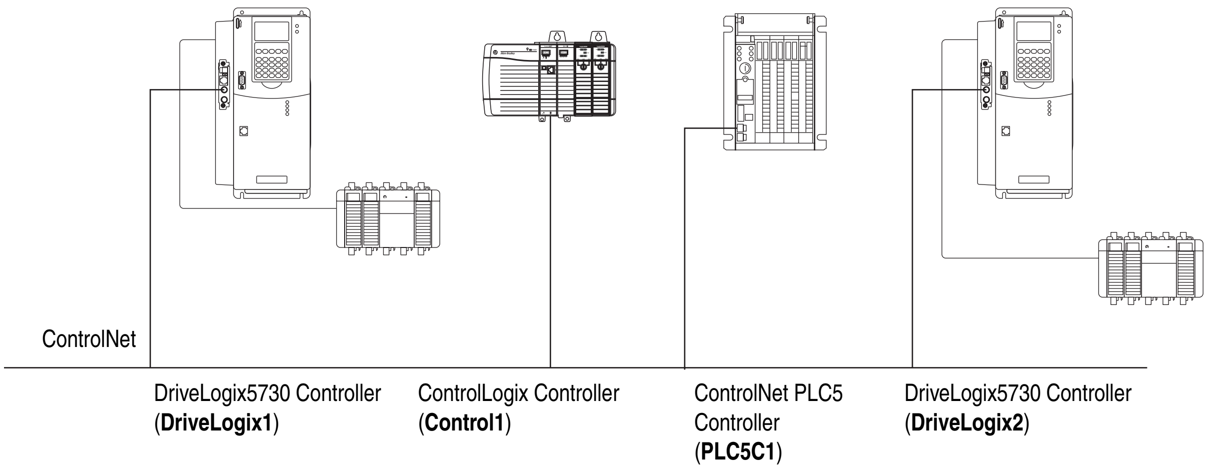
The following table calculates the connections used in this example.

Connection:	Amount:
DriveLogix1 controller to 3 local I/O modules rack-optimized connection for the DIN rail direct connection for each I/O module	 1 3
DriveLogix1 controller to local 1788-CNC	0
DriveLogix1 controller to remote 1788-CNC	0
connected, cached MSG from DriveLogix1 to DriveLogix2	1
produced tagA produced from DriveLogix1 to DriveLogix2 other consumer (2 are configured)	 1 1
produced tag**A** produced from DriveLogix1 to DriveLogix2 other consumer (2 are configured)	 1 1
consumed TagB	1
total connections used:	**10**

If you configured the local I/O modules as rack-optimized, you would only need the DIN-rail connection to the I/O modules, reducing the above example by 3 connections.

Example 3: DriveLogix Controller to Other Devices

In the following example, one DriveLogix controller communicates with a Logix5550 controller and a ControlNet PLC-5 controller over ControlNet.


ControlNet
DriveLogix5730 Controller (**DriveLogix1**)
ControlLogix Controller (**Control1**)
ControlNet PLC5 Controller (**PLC5C1**)
DriveLogix5730 Controller (**DriveLogix2**)

Example 3: Sending MSG Instructions

You configure a MSG instruction to a Logix5550 controller the same as you do for a DriveLogix controller. All Logix-based controllers follow the same MSG configuration requirements. See Example 2 above.

Configuring a MSG instruction for a PLC-5 controller depends on the originating controller.

For MSG instructions originating from the DriveLogix controller to the ControlNet PLC-5 controller:

Type of Logix MSG instruction:	Source:	Destination:
Typed Read	any integer element (such as B3:0, T4:0.ACC, C5:0.ACC, N7:0, etc.)	SINT, INT, or DINT tag
	any floating point element (such as F8:0, PD10:0.SP, etc.)	REAL tag
Typed Write	SINT or INT tag	any integer element (such as B3:0, T4:0.ACC, C5:0.ACC, N7:0, etc.)
	REAL tag	any floating point element (such as F8:0, PD10:0.SP, etc.)
Word Range Read	any data type (such as B3:0, T4:0, C5:0, R6:0, N7:0, F8:0, etc.)	SINT, INT, DINT, or REAL
Word Range Write	SINT, INT, DINT, or REAL	any data type (such as B3:0, T4:0, C5:0, R6:0, N7:0, F8:0, etc.)

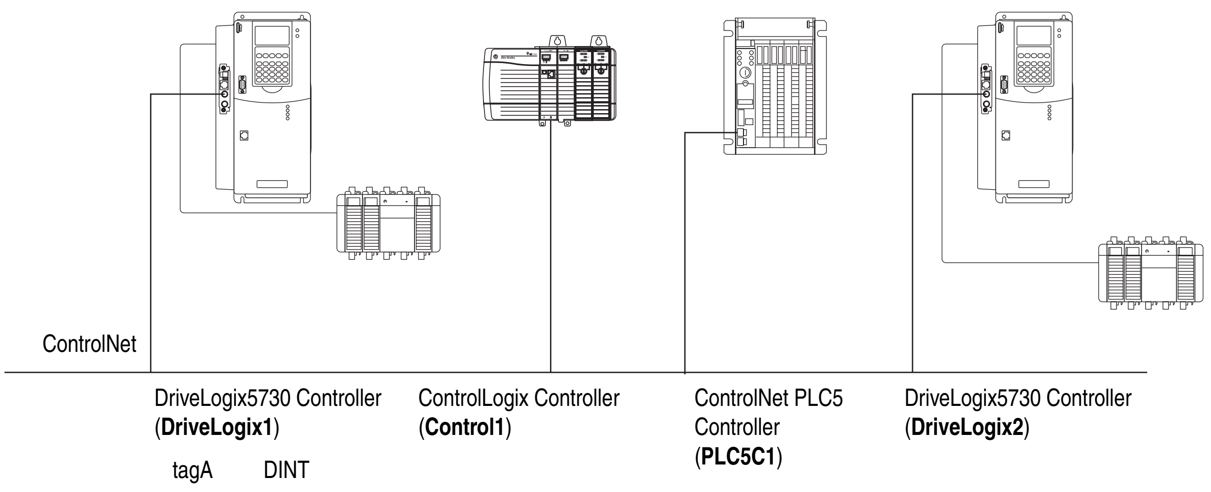
The PLC-5 controller supports logical ASCII addressing so you do not have to map a compatibility file for MSG instructions initiated by a PLC-5 controller. Place the DriveLogix tag name in double quotes (“).

Type of MSG Instruction:	Example Source and Destination:	
PLC-5 writes to DriveLogix	source element	*N7:10*
	destination tag	*“array_1”*
PLC-5 reads from DriveLogix	source tag	*“array_1”*
	destination element	*N7:10*

Example 3: Producing and Consuming Tags

You can produce and consume tags with any Logix controller the same as you do with a DriveLogix controller. All Logix controllers follow the same requirements for producing and consuming tags. See Example 2 above.

Producing and consuming tags with a ControlNet PLC-5 controller depends on the type of data.


ControlNet
DriveLogix5730 Controller (**DriveLogix1**)
tagA DINT
ControlLogix Controller (**Control1**)
ControlNet PLC5 Controller (**PLC5C1**)
DriveLogix5730 Controller (**DriveLogix2**)

Producing a Tag to a ControlNet PLC-5 controller

To produce a tag that a ControlNet PLC-5 controller can consume:

1. Determine the type of data to produce?

If:	And you are producing:	Then:
INT	na	Create a user-defined data type that contains an array of INTs with an even number of elements, such as INT[2]. When you produce INTs, you must produce two or more. Create a produced tag and select the user-defined data type you created.
DINT or REAL	Only one DINT or REAL value	Create a produced tag and select the DINT or REAL data type, as appropriate.
	More than one DINT or REAL	Create a user-defined data type that contains an array of DINTs or REALs, as appropriate. Create a produced tag and select the user-defined data type you created.

2. In RSNetWorx software, open the ControlNet configuration for the target ControlNet PLC-5 controller, insert a Receive Scheduled Message and enter the following Message size:

If the produced tag contains:	Then, for the Message size, enter:
INTs	The number of integers in the produced tag
DINTs	Two times the number of DINTs or REALs in the produced tag. For example, if the produced tag contains 10 DINTs, enter 20 for the Message size.
REALs	

3. In the RSNetWorx software, reschedule (save) the network.

The ControlNet PLC-5 controller does not perform type checking. Make sure the PLC-5 data type can correctly receive the DriveLogix produced tag to ensure proper data is being received.

When a ControlNet PLC-5 controller consumes a tag that is produced by a Logix5000 controller, it stores the data in consecutive 16-bit integers.

The ControlNet PLC-5 controller stores floating-point data, which requires 32-bits regardless of the type of controller, as follows:

– The first integer contains the upper (left-most) bits of the value.

– The second integer contains the lower (right-most) bits of the value.

To re-construct the floating point data within the ControlNet PLC-5 controller, first reverse the order of the integers and then copy them to a floating-point file.

Consuming a Tag from a ControlNet PLC-5 Controller

To consume a tag from a ControlNet PLC-5 controller,:

1. In RSNetWorx software, open the ControlNet configuration of the ControlNet PLC-5 controller, insert a Send Scheduled Message.
2. In RSLogix 5000 software, add the ControlNet PLC-5 controller to the Controller Organizer.
3. Create a user-defined data type that contains these members:

Data type:	**Description:**
DINT	Status
INT[x], where “x” is the output size of the data from the ControlNet PLC-5 controller. (If you are consuming only one INT, no dimension is required.)	Data produced by a ControlNet PLC-5 controller

4. Create a consumed tag with the following properties:

For this tag property:	**Type or select:**
Tag Type	Consumed
Controller	The ControlNet PLC-5 that is producing the data
Remote Instance	The message number from the ControlNet configuration of the ControlNet PLC-5 controller
RPI	A power of two times the NUT of the ControlNet network. For example, if the NUT is 5ms, select an RPI of 5, 10, 20, 40, etc.
Data Type	The user-defined data type that you created.

5. In the RSNetWorx for ControlNet software, reschedule (save) the network.

Example 3: Total Connections Required by DriveLogix1

The following table calculates the connections used in this example.

Connection:	Amount:
DriveLogix1 controller to 3 local I/O modules rack-optimized connection for the DIN rail direct connection for each I/O module	 1 3
DriveLogix1 controller to local 1788-CNC	0
DriveLogix1 controller to remote 1756-CNB	1
DriveLogix1 controller to remote ControlNet PLC-5	1
connected, cached MSG from DriveLogix1 to Control1	1
connected, cached MSG from DriveLogix1 to PLC5C1	1
Produced TagA produced from DriveLogix1 to DriveLogix2 consumed by PLC5C1	 1 1
Consumed TagB from DriveLogix2	1
Consumed INT from PLC5C1	1
total connections used:	**12**

If you configured the local I/O modules as rack-optimized, you would only need the DIN-rail connection to the I/O modules, reducing the above example by 3 connections.

You can configure the 1756-CNB module to use no connection. This is useful if you configure all direct connections to their associated I/O modules and do not need a rack-optimized connection.

Communicating with Devices on an DeviceNet Link

De-energizing the Drive to Connect or Disconnect a Cable

⚠ **ATTENTION:** Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of configuring DeviceNet communication you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

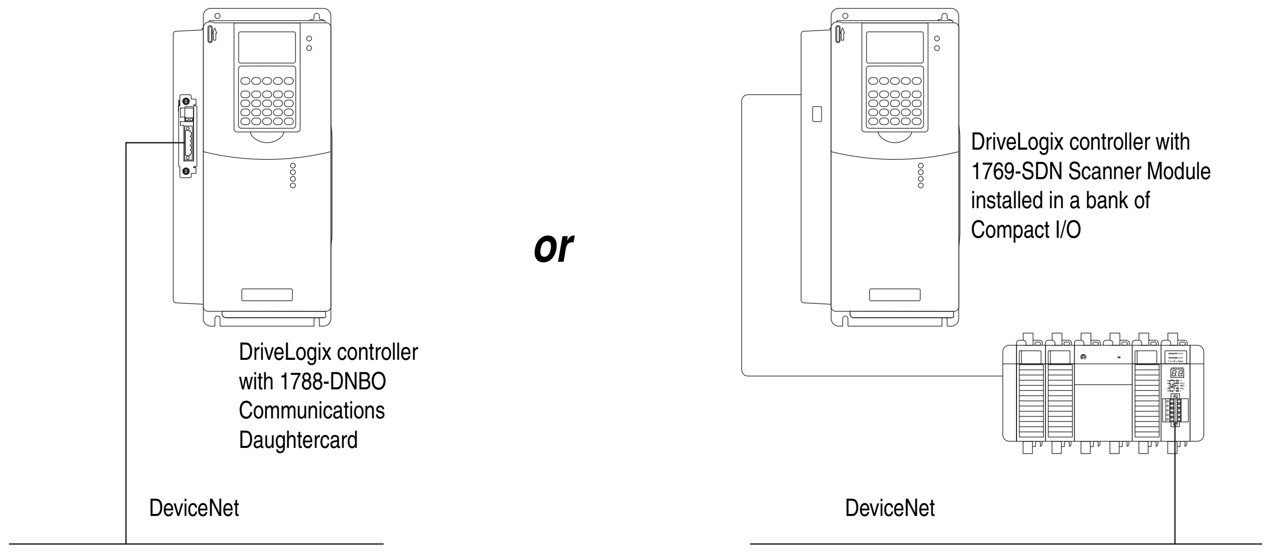
1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive’s input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with configuring DeviceNet communication.

For the DriveLogix controller to operate on a DeviceNet link, you need:

- a workstation with an appropriate DeviceNet interface card
- RSLinx software to configure the DeviceNet communication driver
- RSLogix 5000 programming software (Version 13 or later) to configure the DeviceNet communication daughtercard or local 1769-SDN DeviceNet scanner module as part of the DriveLogix system
- a NetLinx DeviceNet communication daughtercard installed on the DriveLogix controller (refer to Installing the Communications Daughtercard on page B-9)

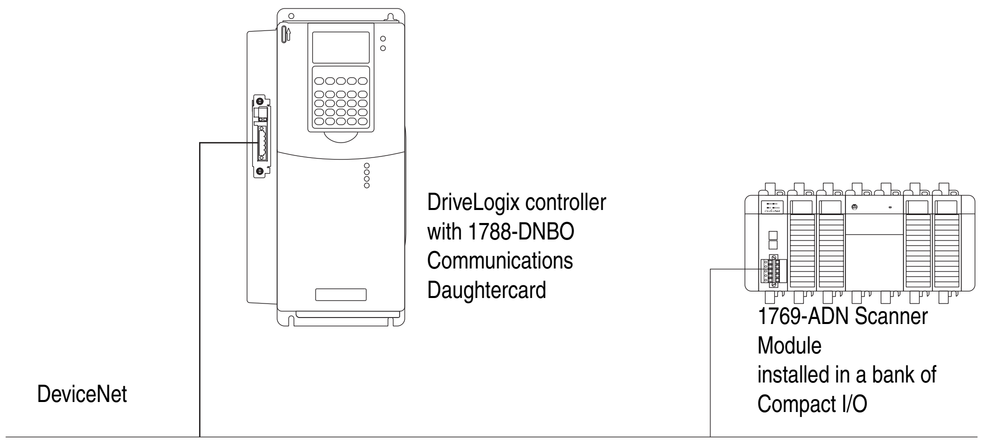
  or

- 1769-SDN DeviceNet scanner module installed on the a bank of local Compact I/O controlled by the DriveLogix controller


DriveLogix controller with 1788-DNBO Communications Daughtercard
DeviceNet
or
DriveLogix controller with 1769-SDN Scanner Module installed in a bank of Compact I/O
DeviceNet

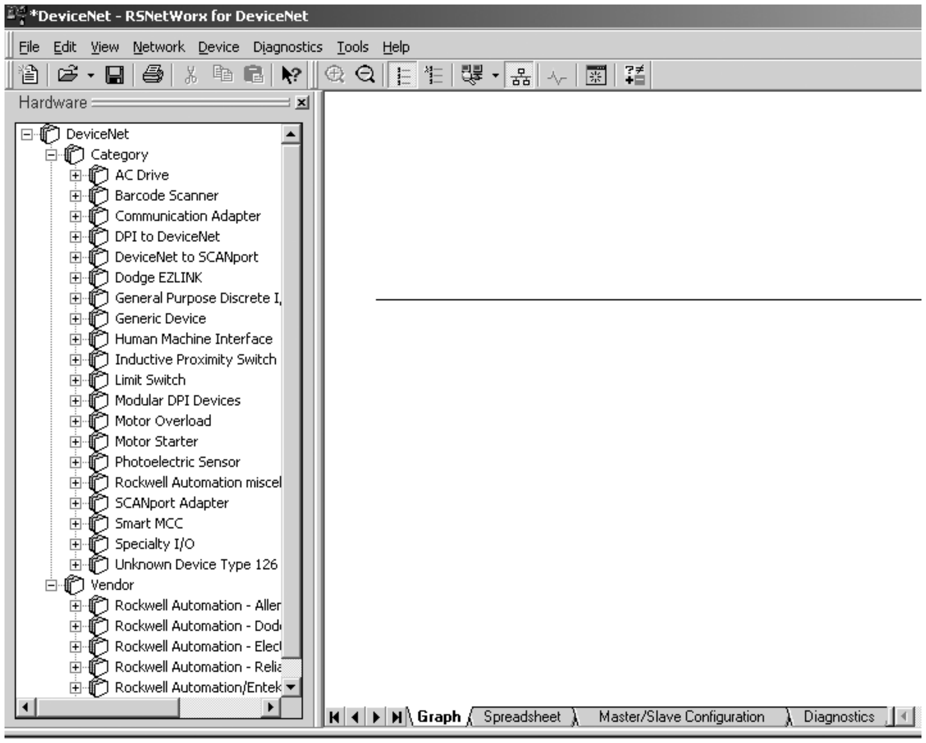
Example: Controlling I/O Over DeviceNet

This example uses a NetLinx DeviceNet communication daughtercard installed on the DriveLogix controller to control a bank of I/O attached to a 1769-ADN adapter module.


DriveLogix controller with 1788-DNBO Communications Daughtercard
1769-ADN Scanner Module installed in a bank of Compact I/O
DeviceNet

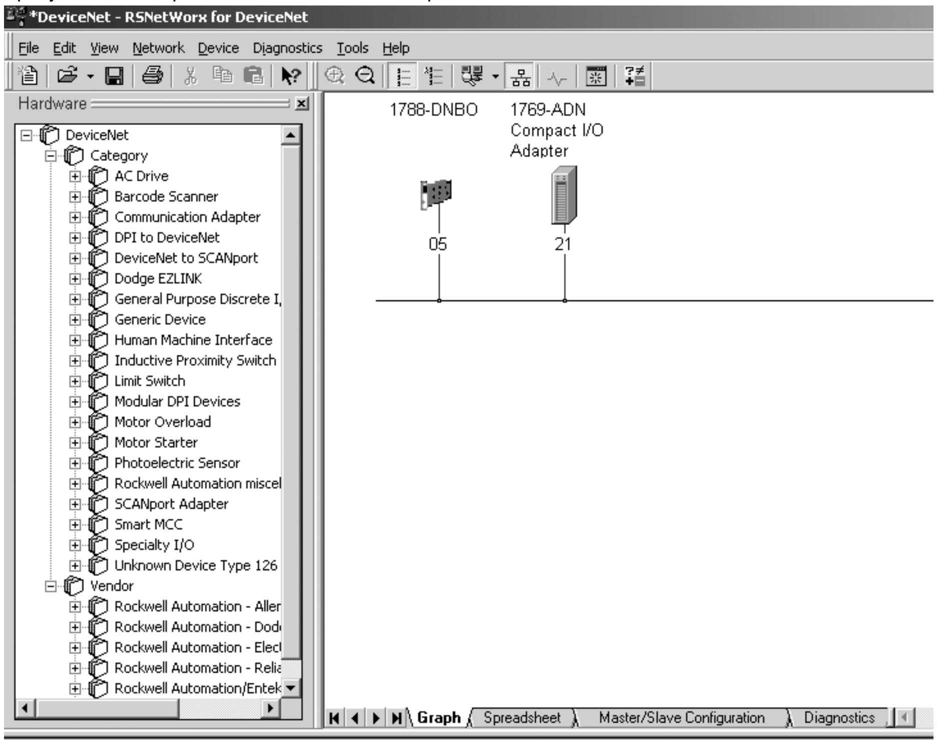
Configuring the 1769-ADN Adapter

1. In RSNetWorx for DeviceNet, select Online from the Network menu.
Choose the appropriate driver depending on whether the computer is directly connected to DeviceNet or you are accessing the network through a controller's backplane and using a different port.

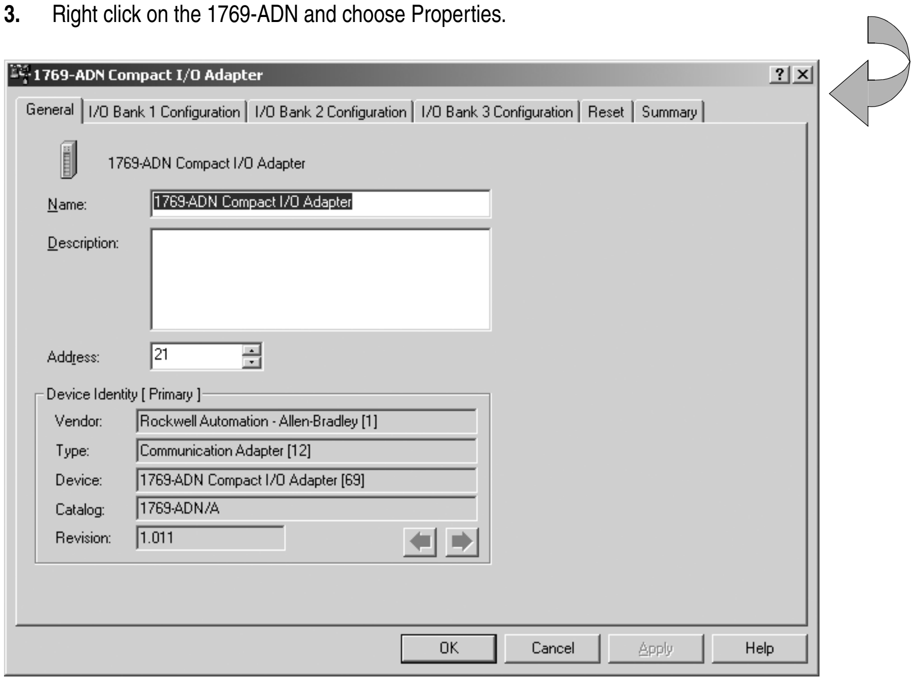


continued

2. The software then prompts you to either upload or download. Choose upload. RSNetWorx will then browse the network and obtain their settings.


*DeviceNet - RSNetWorx for DeviceNet
File Edit View Network Device Diagnostics Tools Help
Hardware
DeviceNet
Category
AC Drive
Barcode Scanner
Communication Adapter
DPI to DeviceNet
DeviceNet to SCANport
Dodge EZLINK
General Purpose Discrete I,
Generic Device
Human Machine Interface
Inductive Proximity Switch
Limit Switch
Modular DPI Devices
Motor Overload
Motor Starter
Photoelectric Sensor
Rockwell Automation miscel
SCANport Adapter
Smart MCC
Specialty I/O
Unknown Device Type 126
Vendor
Rockwell Automation - Aller
Rockwell Automation - Dod
Rockwell Automation - Elec
Rockwell Automation - Relia
Rockwell Automation/Entek
1788-DNBO
1769-ADN Compact I/O Adapter
05
21
Graph
Spreadsheet
Master/Slave Configuration
Diagnostics

3. Right click on the 1769-ADN and choose Properties.



Name: 1769-ADN Compact I/O Adapter
Description:
Address: 21
Device Identity [Primary]
Vendor: Rockwell Automation - Allen-Bradley [1]
Type: Communication Adapter [12]
Device: 1769-ADN Compact I/O Adapter [69]
Catalog: 1769-ADN/A
Revision: 1.011

4. Click on the I/O Bank 1 Configuration tab, then choose upload when prompted. The actual 1769-ADN I/O layout appears. From this screen you can configure the I/O modules in the 1769-ADN system by simply clicking on the slot number box associated with each I/O module.

5. When the I/O modules are configured, click on the Summary tab. Note the number of bytes of input and output data. This will be used later when adding the adapter to the 1769-SDN's scanlist.

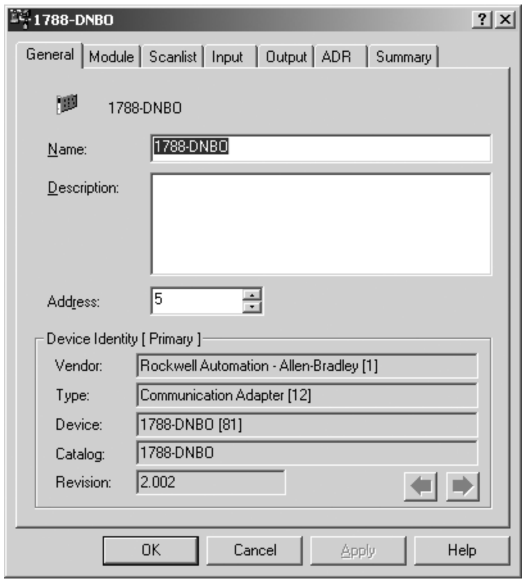
6. Click Apply, then OK to save the configuration and download it to the adapter.

TIP: Configuration changes made to the adapter or any of its I/O modules with RSNetWorx will not be saved or downloaded to the adapter once the adapter is configured in a scanner's scanlist.

To make configuration changes, the controller must be placed into the Program mode and the adapter must be temporarily removed from the scanner's scanlist.

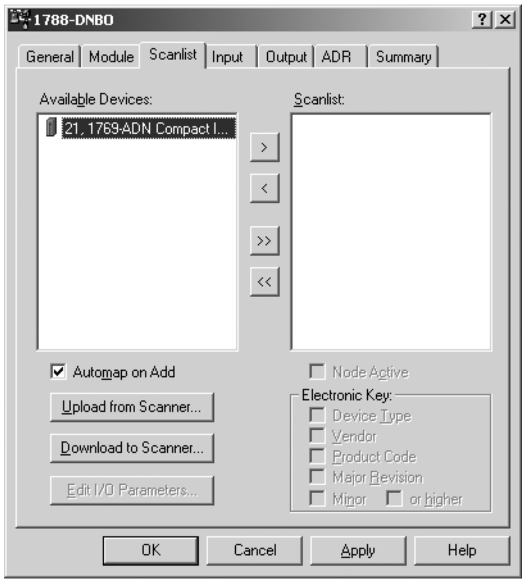
Setting Up the 1788-DNBO DeviceNet Scanner Scanlist

1. Right click on the 1788-DNBO and choose Properties.



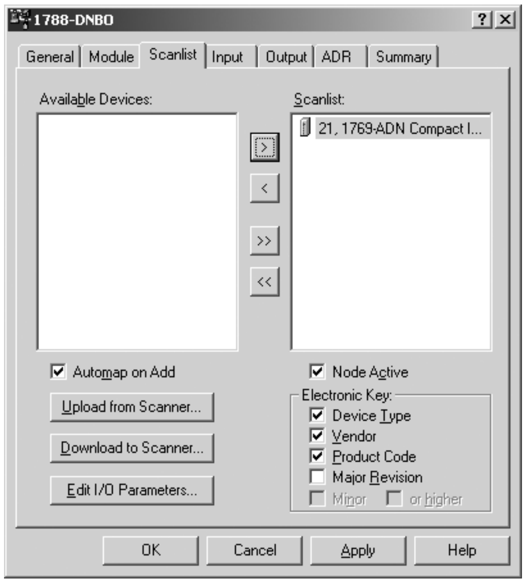
1788-DNBO
Name: 1788-DNBO
Description:
Address: 5
Device Identity [Primary]
Vendor: Rockwell Automation - Allen-Bradley [1]
Type: Communication Adapter [12]
Device: 1788-DNBO [81]
Catalog: 1788-DNBO
Revision: 2.002

2. Click the Scanlist tab, then click Upload when prompted. The area on the left is called "Available Devices" and the area on the right is called "Scanlist". The 1769-ADN adapter should be on the left.



Available Devices:
21, 1769-ADN Compact I...
Scanlist:
☑ Automap on Add
☐ Node Active
Electronic Key:
☐ Device Type
☐ Vendor
☐ Product Code
☐ Major Revision
☐ Minor ☐ or higher

3. Click on the adapter, then click on the single arrow pointing to the right. This moves the adapter from Available Devices to the scanner's scanlist.


1788-DNBO

General | Module | Scanlist | Input | Output | ADR | Summary

Available Devices:

Scanlist:
21, 1769-ADN Compact I...

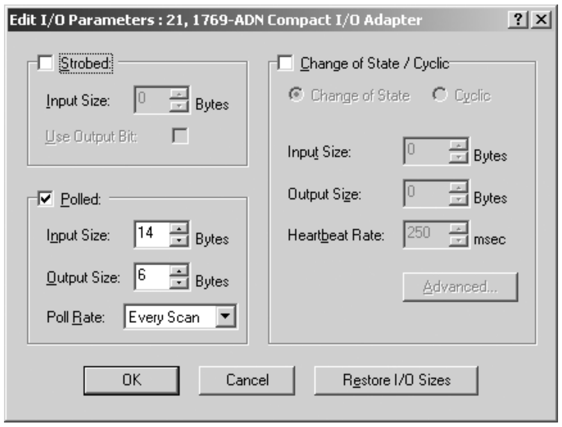
☑ Automap on Add

Upload from Scanner... | Download to Scanner... | Edit I/O Parameters...

☑ Node Active

Electronic Key:
☑ Device Type
☑ Vendor
☑ Product Code
☐ Major Revision
☐ Minor ☐ or higher

OK | Cancel | Apply | Help


Edit I/O Parameters : 21, 1769-ADN Compact I/O Adapter

☐ Strobed:
Input Size: 0 Bytes
Use Output Bit: ☐

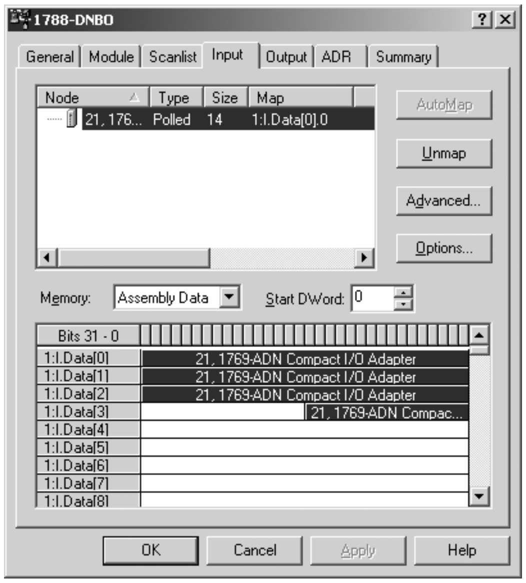
☑ Polled:
Input Size: 14 Bytes
Output Size: 6 Bytes
Poll Rate: Every Scan

☐ Change of State / Cyclic
◉ Change of State ○ Cyclic
Input Size: 0 Bytes
Output Size: 0 Bytes
Heartbeat Rate: 250 msec
Advanced...

OK | Cancel | Restore I/O Sizes

4. Verify that the Input Size and Output Size are correct. The Output and Input sizes correspond to the total number of output and input bytes noted from the adapter's summary page. In this example, the scanner transmits 6 bytes to the adapter (output data) and receives 28 bytes from the adapter (input data).

Click OK when finished with this screen.


1788-DNBO

General | Module | Scanlist | Input | Output | ADR | Summary

Node	Type	Size	Map
21, 176...	Polled	14	1:I.Data[0].0

AutoMap | Unmap | Advanced... | Options...

Memory: Assembly Data   Start DWord: 0

	Bits 31 - 0
1:I.Data[0]	21, 1769-ADN Compact I/O Adapter
1:I.Data[1]	21, 1769-ADN Compact I/O Adapter
1:I.Data[2]	21, 1769-ADN Compact I/O Adapter
1:I.Data[3]	21, 1769-ADN Compac...
1:I.Data[4]	
1:I.Data[5]	
1:I.Data[6]	
1:I.Data[7]	
1:I.Data[8]	

OK | Cancel | Apply | Help

5. Click on the Input tab.

Then click Apply and OK.

Download when prompted.

Mapping starts at word 0 for both the input and the output data image. The input status and output configuration words are no longer included with the I/O data scanlist. Use the status and configuration tags created in RSLogix 5000 software to read status or set configuration bits.

▶ **TIP:** The input and output data being exchanged by the scanner and adapter is packed data. This means that there is no special structure to it that makes it obvious which I/O module it is associated with.

To establish which data is from which module, you must list the number of input and output words each module has. Then, based on its position in the I/O bank, you can determine where any module's data is in the controller's I/O tags.

Transferring Data

There are 28 bytes of input data and 6 bytes of output data for this example. The I/O modules in the adapter's system are:

Module	Input	Output
ADN Status Information (added by the 1769-ADN)	1 DINT word	0 words
1769-IA16	1/2 DINT word	0 words
1769-OB16	1/2 DINT word	1/2 DINT word
1769-IF4	3 DINT words	0 words
1769-OF2	2 DINT words	1 DINT word
Total Words	7 DINT words	1 1/2 DINT words
Total Bytes	28 bytes	6 bytes

The total is 7 DINT words or 28 input bytes. The first DINT word is adapter status, leaving 6 DINT words (24 bytes) for data. The input data maps to the controller's input data tag at the following word locations:

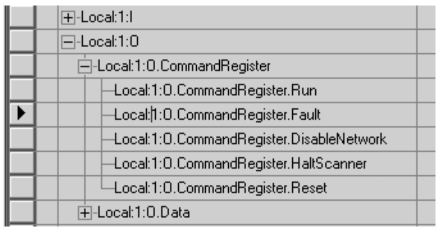
Location	Description
Word 0	1769-ADN status information
Word 1	1769-IA16 module's input word
Word 1	1769-OB16 module's input data (output data echo)
Words 2-4	1769-IF4 module's input data
Words 5-6	1769-OF2 module's input data

The output data can be determined in a similar manner. This data begins with word 0 of the output tag in the controller as follows:

Location	Description
Word 0	1769-OB16 module's output word
Words 0-1	1769-OF2 module's output words

Module Command Array

The module command array is the primary control interface between your control program and the module. In RSLogix 5000 software, the CommandRegister tag structure is as follows:

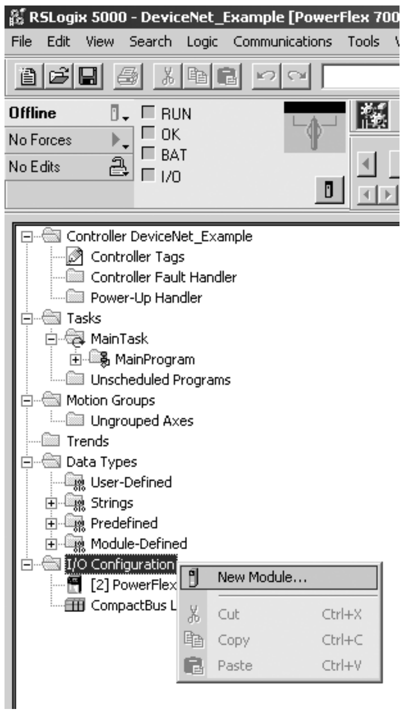


```
+-Local:1:I
-Local:1:O
  -Local:1:O.CommandRegister
    Local:1:O.CommandRegister.Run
    Local:1:O.CommandRegister.Fault
    Local:1:O.CommandRegister.DisableNetwork
    Local:1:O.CommandRegister.HaltScanner
    Local:1:O.CommandRegister.Reset
  +-Local:1:O.Data
```

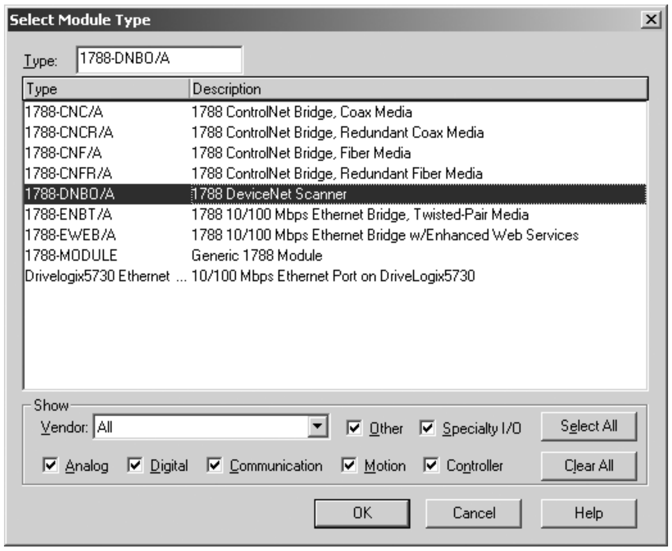
Output Word	Bit	Description	Behavior
0	0	Run	This bit controls when the module scans its mapped slave devices. When set (1), the scanner will process I/O data as defined by its scanlist. To actually scan the network the Fault and Disable Network command bits must be clear (0).
	1	Fault	When set, the scanner's I/O mode will be Halt; messaging will still operate. The fault bit is primarily used to artificially set the slave devices into a fault state due to some event or condition within the control program.
	2	Disable Network	When set, the scanner is functionally removed from the network.
	3	HaltScanner	When set, the scanner stops scanning its mapped slave devices.
	4	Reset	Restarts access to the DeviceNet network.

Creating a Project for the DriveLogix5730 Controller

1. In the Controller Organizer, select the I/O Configuration folder. Right-click the selected folder and select New Module.



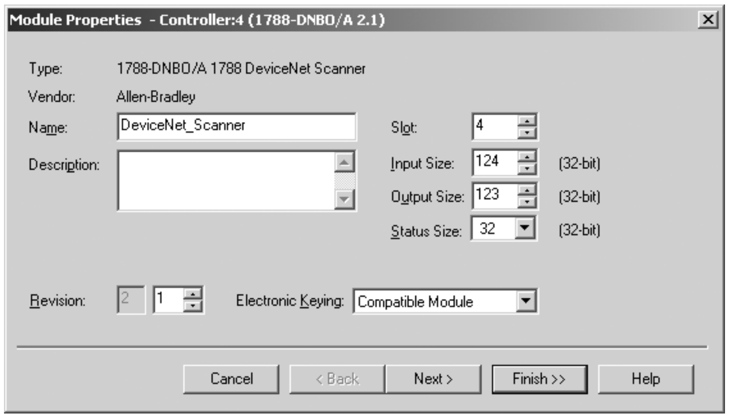
2. Select the 1788-DNBO DeviceNet Scanner from the list of possible communication devices.
Click OK.


Select Module Type

Type: 1788-DNBO/A

Type	Description
1788-CNC/A	1788 ControlNet Bridge, Coax Media
1788-CNCR/A	1788 ControlNet Bridge, Redundant Coax Media
1788-CNF/A	1788 ControlNet Bridge, Fiber Media
1788-CNFR/A	1788 ControlNet Bridge, Redundant Fiber Media
1788-DNBO/A	1788 DeviceNet Scanner
1788-ENBT/A	1788 10/100 Mbps Ethernet Bridge, Twisted-Pair Media
1788-EWEB/A	1788 10/100 Mbps Ethernet Bridge w/Enhanced Web Services
1788-MODULE	Generic 1788 Module
Drivelogix5730 Ethernet ...	10/100 Mbps Ethernet Port on DriveLogix5730

Show
Vendor: All ☑ Other ☑ Specialty I/O
☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller

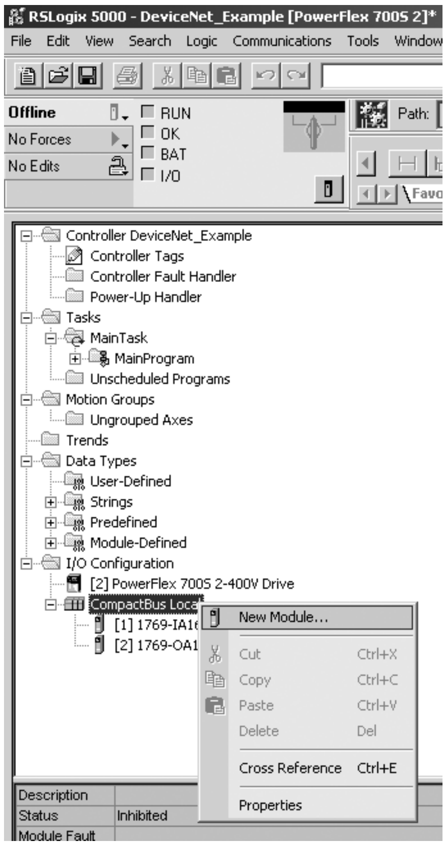

Module Properties - Controller:4 (1788-DNBO/A 2.1)

Type: 1788-DNBO/A 1788 DeviceNet Scanner
Vendor: Allen-Bradley
Name: DeviceNet_Scanner
Description:
Slot: 4
Input Size: 124 (32-bit)
Output Size: 123 (32-bit)
Status Size: 32 (32-bit)
Revision: 2 1
Electronic Keying: Compatible Module

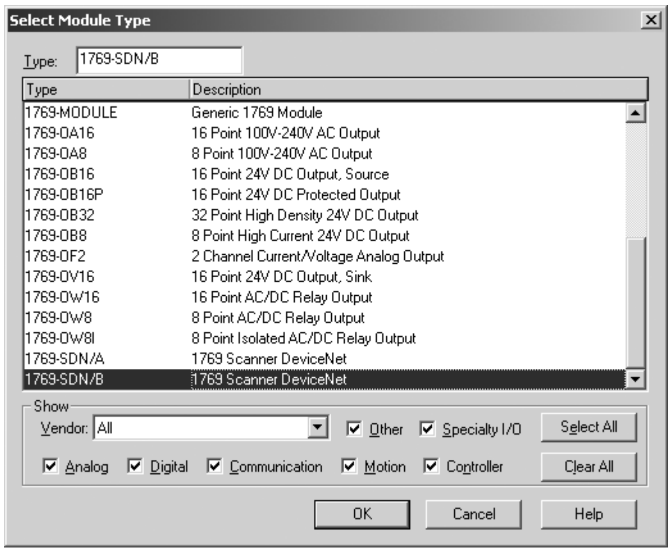
3. Enter a Name for the port and make sure it is assigned to slot 4.
Click Finish.

If you are using a 1769-SDN Scanner Module, add it to the local CompactBus:

1. In the Controller Organizer, select the I/O Configuration folder. Right-click the selected Local CompactBus and select New Module.



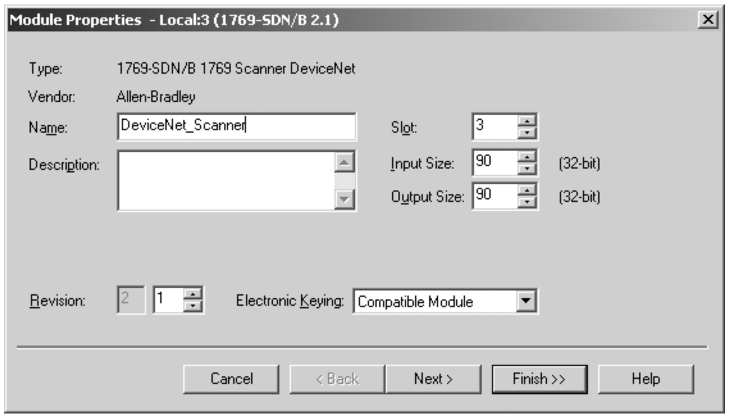
2. Select the 1769-SDN DeviceNet Scanner from the list of possible communication devices.
Click OK.


Select Module Type

Type: 1769-SDN/B

Type	Description
1769-MODULE	Generic 1769 Module
1769-OA16	16 Point 100V-240V AC Output
1769-OA8	8 Point 100V-240V AC Output
1769-OB16	16 Point 24V DC Output, Source
1769-OB16P	16 Point 24V DC Protected Output
1769-OB32	32 Point High Density 24V DC Output
1769-OB8	8 Point High Current 24V DC Output
1769-OF2	2 Channel Current/Voltage Analog Output
1769-OV16	16 Point 24V DC Output, Sink
1769-OW16	16 Point AC/DC Relay Output
1769-OW8	8 Point AC/DC Relay Output
1769-OW8I	8 Point Isolated AC/DC Relay Output
1769-SDN/A	1769 Scanner DeviceNet
1769-SDN/B	1769 Scanner DeviceNet

Show
Vendor: All
☑ Other ☑ Specialty I/O
☑ Analog ☑ Digital ☑ Communication ☑ Motion ☑ Controller


Module Properties - Local:3 (1769-SDN/B 2.1)

Type: 1769-SDN/B 1769 Scanner DeviceNet
Vendor: Allen-Bradley
Name: DeviceNet_Scanner
Description:
Slot: 3
Input Size: 90 (32-bit)
Output Size: 90 (32-bit)
Revision: 2 1
Electronic Keying: Compatible Module

3. Enter a Name for the port and make sure it is assigned to appropriate I/O slot.
Click Finish.

All tags for I/O modules are automatically created when the profiles for these modules are configured. Double click on Controller Tags in the controller organizer to view these tags. Each I/O module slot has Input, Output and Configuration tags created, if they apply. These tags are structured as:

Tag	Definition
Local:s:I	s is the slot number I represents Input Data
Local:s:O	O represents Output Data
Local:s:C	C represents Configuration Data

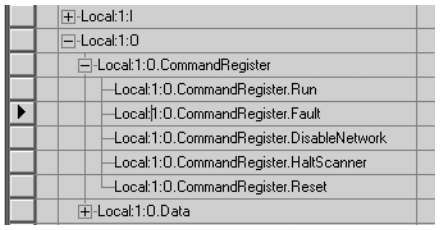
If the 1769-SDN is in slot 1, the input addresses for the scanner are:

Tag	Definition
Local:1:I.Data[0]	1769-ADN Status Information
Local:1:I.Data[1]	Input Data from 1769-IA16
Local:1:I.Data[1]	Input (output echo) Data from 1769-OB16
Local:1:I.Data[2] through Local:3:I.Data[4]	Input Data from 1769-IF4
Local:1:I.Data[5] through Local:3:I.Data[6]	Input Data from 1769-OF2

This output addresses for the scanner are:

Tag	Definition
Local:1:O.Data[0]	Output data for 1769-OB16
Local:1:O.Data[0] through Local:3:O.Data[1]	Output data for 1769-OF2

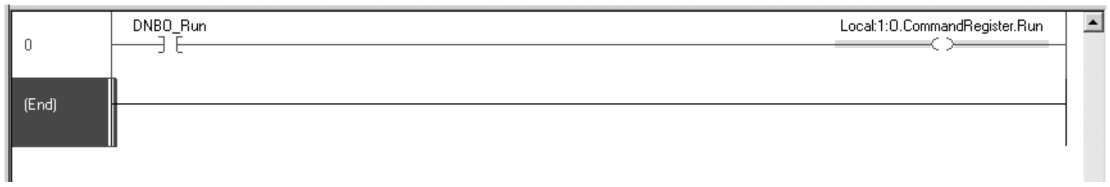
The controller uses the CommandRegister (Local:1:O.CommandRegister) to send commands to the scanner



```
+-Local:1:I
−-Local:1:O
  −-Local:1:O.CommandRegister
     ├Local:1:O.CommandRegister.Run
     ├Local:1:O.CommandRegister.Fault
     ├Local:1:O.CommandRegister.DisableNetwork
     ├Local:1:O.CommandRegister.HaltScanner
     └Local:1:O.CommandRegister.Reset
  +-Local:1:O.Data
```

Entering Program Logic

The program for this example consists of a single rung that is used to place the scanner into the RUN mode.



Communicating with Devices on a DH485 Link

De-energizing the Drive to Connect or Disconnect a Cable

ATTENTION: Severe injury or death can result from electrical shock or burn. Verify that the voltage on the bus capacitors has discharged before connecting to the communication ports. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

During the process of configuring DH485 communication you will need to connect or disconnect a programming or network cable at the controller. You should do this only if the drive is de-energized.

1. Turn off and lock out input power. Wait five minutes.
2. Verify that there is no voltage at the drive's input power terminals.
3. Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.
4. Connect or disconnect the programming or network cable.
5. Turn power back on and proceed with configuring DH485 communication.

Understanding How the DriveLogix5730 Controller Supports DH485 Communications

When using a DriveLogix5730 controller it is recommended that you use NetLinx networks (EtherNet/IP, ControlNet, or DeviceNet) because excessive traffic on a DH-485 network may make it impractical to connect to a DriveLogix5730 controller with RSLogix 5000 programming software. DriveLogix5730 processors fully support the DH-485 protocol, but using the recommended NetLinx networks is more practical.

The DH-485 protocol uses RS-485 half-duplex as its physical interface. (RS-485 is a definition of electrical characteristics; it is not a protocol.) You can configure the RS-232 port of the DriveLogix5730 controller to act as a DH-485 interface. By using a 1761-NET-AIC and the appropriate RS232 cable (1756-CP3 or 1747-CP3), a DriveLogix5730 controller can send and receive data on a DH-485 network.

Important: A DH-485 network consists of multiple cable segments. Limit the total length of all the segments to 1219m (4000 ft.).

Configuring Your System for a DH-485 Link

For the DriveLogix5730 controller to operate on a DH-485 network, you need:

a 1761-NET-AIC interface converter for each DriveLogix5730 controller you want to put on the DH-485 network.

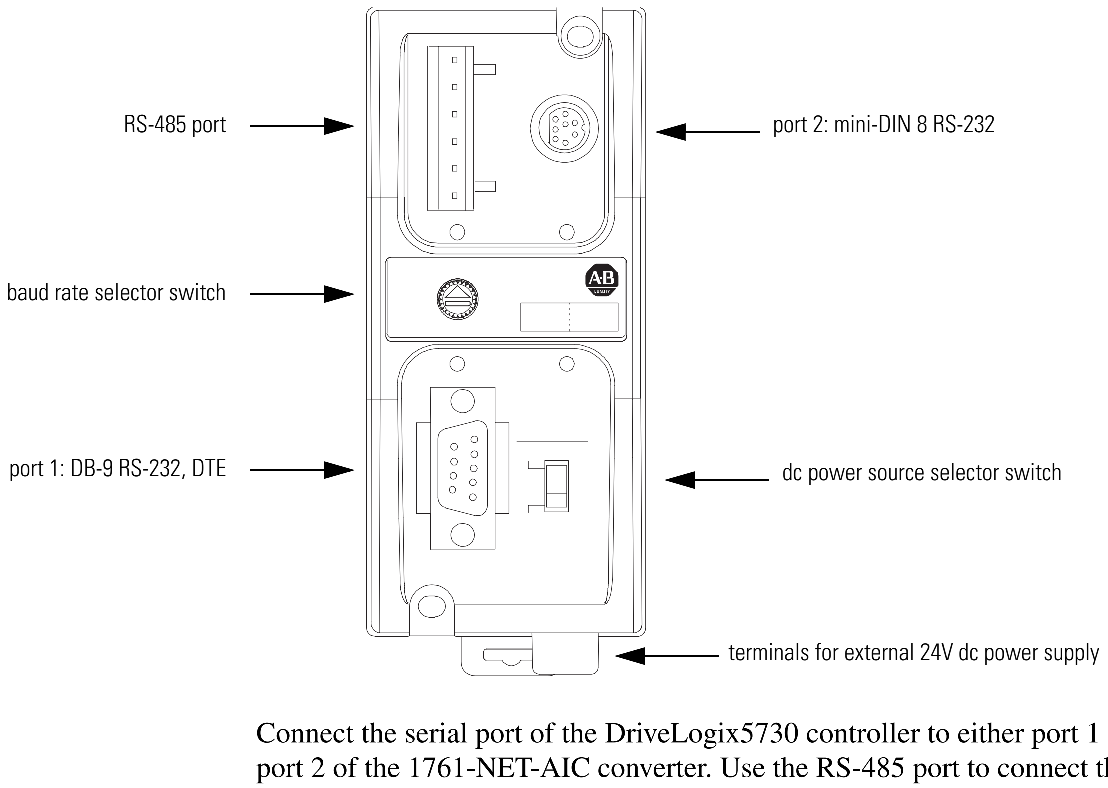
You can have two controllers per one 1761-NET-AIC converter, but you need a different cable for each controller. Connect one controller to port 1 (9-pin connector) and one controller to port 2 (mini-DIN connector).

RSLogix 5000 programming software to configure the serial port of the controller for DH-485 communications.

When attempting to go online or upload/download a program using the Communications/Who Active window in RSLogix 5000 software, disable the Autobrowse feature to minimize traffic from RSLogix 5000 software on the DH-485 network.

Step 1: Configure the Hardware

The RS-232 port is built-in to the front of the DriveLogix5730 controller. The 1769-L31 controller has two serial ports. Connect the serial port to an RS-232-to-RS-485 interface converter. One possible converter is the 1761-NET-AIC interface converter.


RS-485 port
port 2: mini-DIN 8 RS-232
baud rate selector switch
port 1: DB-9 RS-232, DTE
dc power source selector switch
terminals for external 24V dc power supply

Connect the serial port of the DriveLogix5730 controller to either port 1 or port 2 of the 1761-NET-AIC converter. Use the RS-485 port to connect the converter to the DH-485 network.

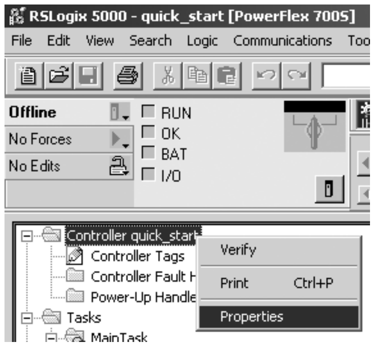
The cable you use to connect the controller depends on the port you use on the 1761-NET-AIC converter.

If you connect to this port:	Use this cable:
port 1 DB-9 RS-232, DTE connection	1747-CP3 or 1761-CBL-AC00
port 2 mini-DIN 8 RS-232 connection	1761-CBL-AP00 or 1761-CBL-PM02

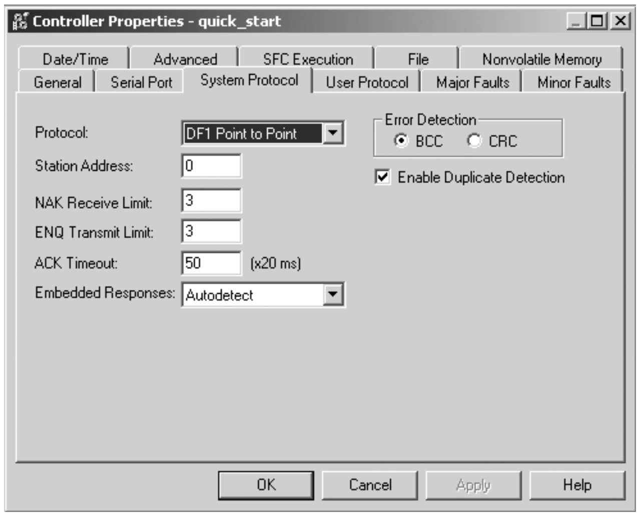
Step 2: Configure the DH-485 port of the controller

1. In RSLogix 5000 programming software, select the Controller folder. Right-click to select Properties.

2. On the System Protocol tab, specify the appropriate serial communication configuration.



ahw0600.tif

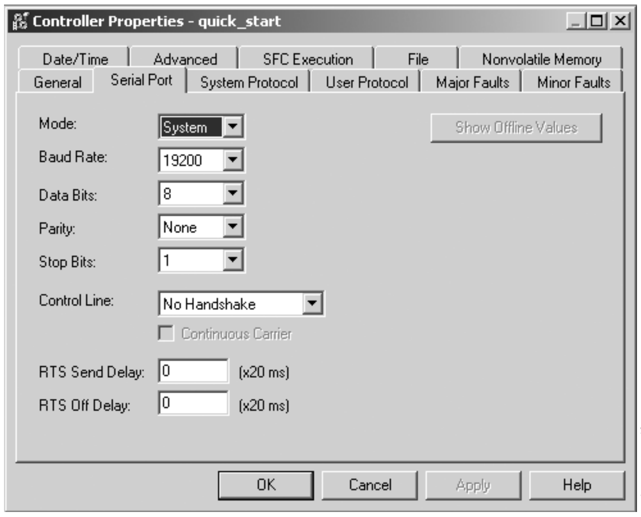


Protocol: DF1 Point to Point
Station Address: 0
NAK Receive Limit: 3
ENQ Transmit Limit: 3
ACK Timeout: 50 (x20 ms)
Embedded Responses: Autodetect
Error Detection: [✓ BCC] CRC
[✓] Enable Duplicate Detection

ahw0601.tif

3. On the Serial Port tab, specify the appropriate communication settings.

The grayed out settings are selections that do not apply to a DH-485 network.



Mode: System
Baud Rate: 19200
Data Bits: 8
Parity: None
Stop Bits: 1
Control Line: No Handshake
[ ] Continuous Carrier
RTS Send Delay: 0 (x20 ms)
RTS Off Delay: 0 (x20 ms)

ahw0602.tif

Specify these characteristics on the Serial Port tab (default values are shown in bold):

Characteristic:	Description (default is shown in bold):
Baud Rate	Specifies the communication rate for the DH-485 port. All devices on the same DH-485 network must be configured for the same baud rate. Select 9600 or **19200** Kbps.
Node Address	Specifies the node address of the DriveLogix controller on the DH-485 network. Select a number **1**-31 decimal, inclusive. To optimize network performance, assign node addresses in sequential order. Initiators, such as personal computers, should be assigned the lowest address numbers to minimize the time required to initialize the network.
Token Hold Factor	Number of transmissions (plus retries) that a node holding a token can send onto the data link each time that it receives the token. Enter a value between 1-4. The default is 1.
Maximum Node Address	Specifies the maximum node address of all the devices on the DH-485 network. Select a number 1-**31** decimal, inclusive. To optimize network performance, make sure: the maximum node address is the highest node number being used on the network that all the devices on the same DH-485 network have the same selection for the maximum node address.

Planning a DH-485 Network

The DH-485 network offers:

- interconnection of 32 devices
- multi-master capability
- token passing access control
- the ability to add or remove nodes without disrupting the network
- maximum network length of 1219 m (4000 ft.)

The DH-485 protocol supports two classes of devices: initiators and responders. All initiators on the network get a chance to initiate message transfers. The DH-485 protocol uses a token-pass algorithm to determine which initiator has the right to transmit

DH-485 Token Rotation

A node holding the token can send any valid packet onto the network. Each node gets only one transmission (plus two retries) each time it receives the token. After a node sends one message packet, it attempts to give the token to its successor by sending a "token pass" packet to its successor.

If no network activity occurs, the initiator sends the token pass packet again. After two retries (a total of three tries) the initiator attempts to find a new successor.

Important: The maximum address that the initiator searches for before starting again with zero is the value in the configurable parameter "maximum node address." The default value for this parameter is 31 for all initiators and responders.

The allowable range of the node address of an initiator is 0 to 31. The allowable address range for all responders is 1 to 31. There must be at least one initiator on the network.

Network Initialization

The network requires at least one initiator to initialize it. Network initialization begins when an initiator on the network detects a period of inactivity that exceeds the time of a link dead timeout. When the link dead timeout is exceeded, usually the initiator with the lowest address claims the token. When an initiator has the token it will begin to build the network.

Building a network begins when the initiator that claimed the token tries to pass the token to the successor node. If the attempt to pass the token fails, or if the initiator has no established successor (for example, when it powers up), it begins a linear search for a successor starting with the node above it in the addressing.

When the initiator finds another active initiator, it passes the token to that node, which repeats the process until the token is passed all the way around the network to the first node. At this point, the network is in a state of normal operation.

Number of Nodes and Node Addresses

The number of nodes on the network directly affects the data transfer time between nodes. Unnecessary nodes (such as a second programming terminal that is not being used) slow the data transfer rate. The maximum number of nodes on the network is 32.

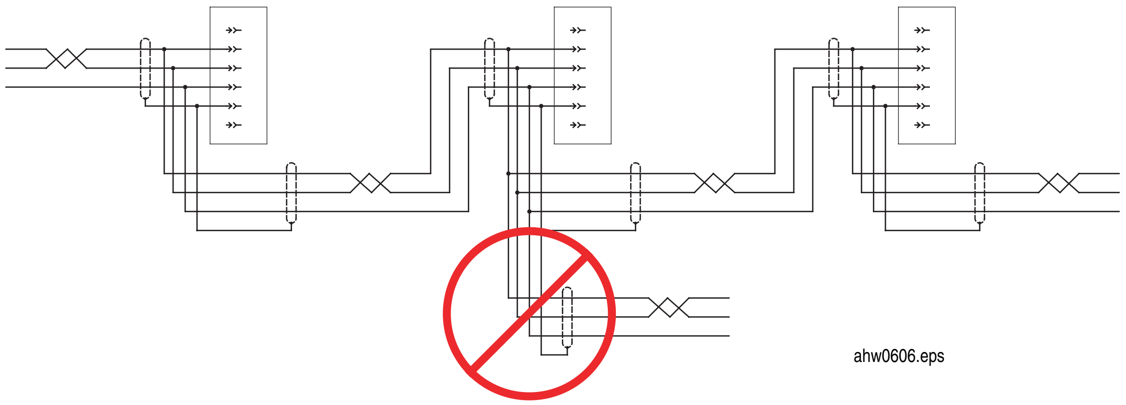
If the node addresses for controllers are assigned in sequence, starting at node 1 (with node 0 left for a programming terminal), it is as efficient to leave the maximum node address at 31 as it is to decrease it to the highest node address on the network. Then, adding devices to the network at a later time will not require modifying the maximum node address in every device on the network. The maximum node address should be the same for all devices on a DH-485 network for optimal operation.

The best network performance occurs when node addresses start at 0 and are assigned in sequential order. The controller defaults to node address 1 (controllers cannot be node 0). Initiators, such as personal computers, should be assigned the lowest numbered addresses to minimize the time required to initialize the network.

Installing a DH-485 Network

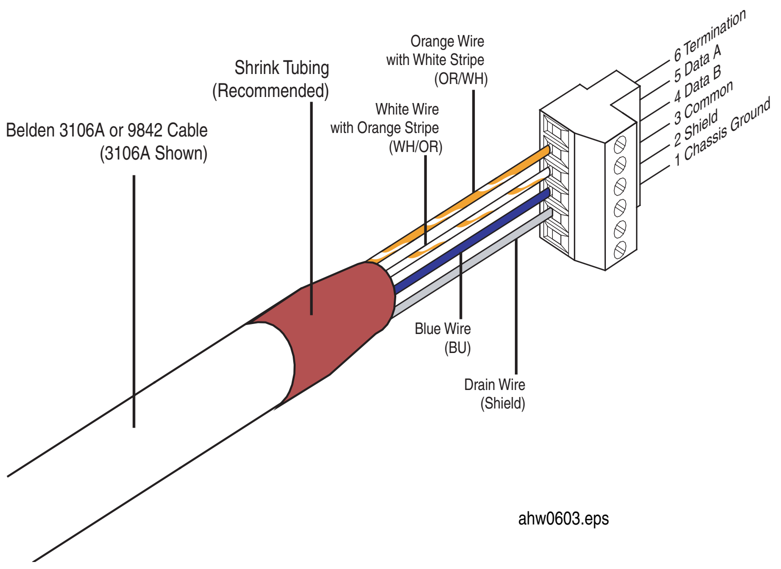
A DH-485 network consists of a number of cable segments daisy-chained together. The total length of the cable segments cannot exceed 1219 m (4000 ft).

Important: Use shielded, twisted-pair cable - either Belden 3106A or Belden 9842. A daisy-chained network is recommended. Star connections are not recommended



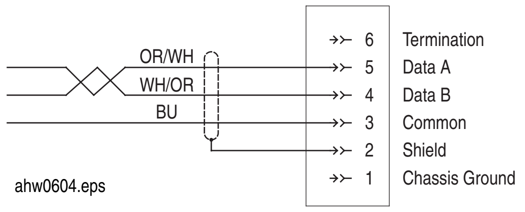
When cutting cable segments, make them long enough to route them from one link coupler to the next with sufficient slack to prevent strain on the connector. Allow enough extra cable to prevent chafing and kinking in the cable.

Figure 9.1 Single Cable Connection


Belden 3106A or 9842 Cable (3106A Shown)
Shrink Tubing (Recommended)
Orange Wire with White Stripe (OR/WH)
White Wire with Orange Stripe (WH/OR)
Blue Wire (BU)
Drain Wire (Shield)
6 Termination
5 Data A
4 Data B
3 Common
2 Shield
1 Chassis Ground
ahw0603.eps

Connections Using Belden 3106 Cable

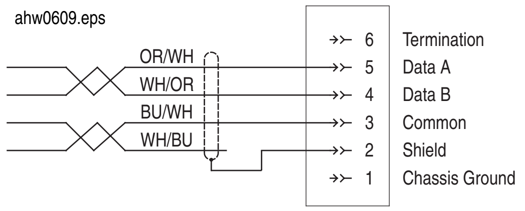
The table and schematic diagram below shows wire/terminal connections for Belden 3106A cable.



For this Wire/Pair	Connect this Wire	To this Terminal
white/orange	orange with white stripe	5 - Data A)
	white with orange stripe	4 - (Data B)
blue	blue	3 - (Common)
shield/drain	non-jacketed	2 - Shield

Connections Using Belden 9842 Cable

The table and schematic diagram below shows wire/terminal connections for Belden 9842 cable.



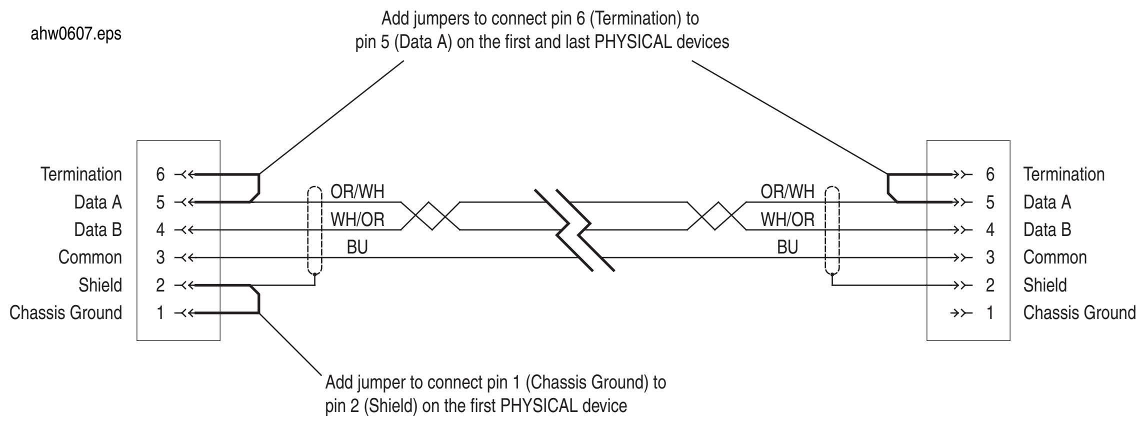
For this Wire/Pair	Connect this Wire	To this Terminal
white/orange	orange with white stripe	5 - (Data A)
	white with orange stripe	4 - (Data B)
blue/white	white with blue stripe	cut back - no connection[1]
	blue with white stripe	3 - (Common)
shield/drain	non-jacketed	2 - Shield

[1] To prevent confusion when installing the communication cable, cut back the white with blue stripe wire immediately after the insulation jacket is removed. This wire is not used by DH-485.

Grounding and Terminating a DH-485 Network

You must terminate the network at the first and last PHYSICAL devices, by connecting pin 6 (Termination) to pin 5 (Data A).

You must ground the network at the first PHYSICAL device by connecting pin 1 (Chassis Ground) to pin 2 (Shield).


ahw0607.eps
Add jumpers to connect pin 6 (Termination) to pin 5 (Data A) on the first and last PHYSICAL devices
Termination 6
Data A 5
Data B 4
Common 3
Shield 2
Chassis Ground 1
OR/WH
WH/OR
BU
OR/WH
WH/OR
BU
6 Termination
5 Data A
4 Data B
3 Common
2 Shield
1 Chassis Ground
Add jumper to connect pin 1 (Chassis Ground) to pin 2 (Shield) on the first PHYSICAL device

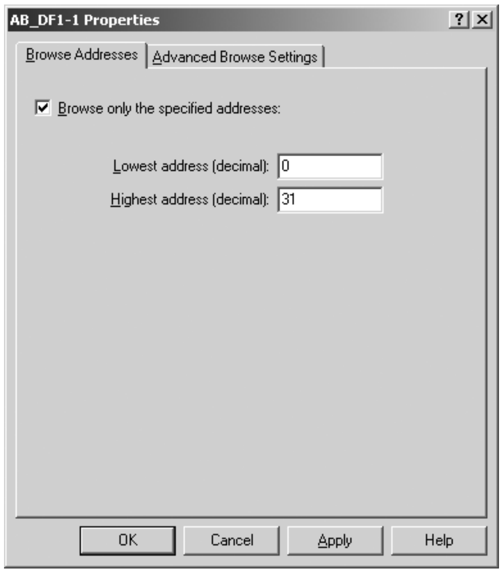
Important: A device's physical location may be independent of its node address. Make sure you ground and terminate the proper PHYSICAL locations.

Browsing a DH-485 Network Remotely

To improve performance when browsing a DH-485 network, configure the DH-485 network properties in RSLinx software to display only those nodes that actually exist on the network.

1. In RSLinx software, right-click on the DH-485 network you plan to browse and select Properties.

2. On the Browse Addresses tab, specify the lowest and highest addresses that exist on the DH-485 network.



☑ Browse only the specified addresses:

Lowest address (decimal): 0

Highest address (decimal): 31

ahw0954.tif

If you do not specify a specific range of addresses on the DH-485 network, the RSWho function in RSLinx software attempts to locate a device at every node address. Trying to locate devices that do not exist adds considerable time to displaying the RSWho window for the network.

Notes:

DriveLogix5730 Controller Specifications

DriveLogix5730 Controller Specifications

Category	Specification
serial port (CH0 -RS-232)	RS-232, fully isolated DF1, DH-485, ASCII 38.4 Kbits/s maximum
optional embedded EtherNet/IP	RJ-45 or 100BaseT 10/100 Mbytes/sec
connectivity options (these options require the Logix Expansion Board and Expanded Cassette)	• NetLinx communication daughtercards (ControlNet, EtherNet/IP, DeviceNet, Third Party) • Compact I/O connection • CompactFlash (memory card)
user memory	1.5 Mbyte
nonvolatile memory	1784-CF64 CompactFlash
maximum number of I/O banks	2
maximum number of I/O modules	16 (8 modules per bank) or (4 max on either side of the power supply)
battery	1769-BA
serial cable	1756-CP3 directly to controller 1747-CP3 directly to controller
Compact I/O Cable	20D-DL2-CR3 20D-DL2-CL3

DriveLogix5730 Environmental Specifications

Category	Specification
operating temperature	
storage temperature	
relative humidity	
vibration	
shock	
emissions	
Electrical/EMC	CISPA11: Group 1, Class A
ESD Immunity (IEC61000-4-2)	The unit has passed testing at the following levels:
Radiated RF Immunity (IEC61000-4-3)	4 kV contact discharges, 8 kV air discharges
Surge Transient Immunity (IEC61000-4-5)	10V/M with 1kHz sine-wave 80%AM from 30MHz to 2000MHz 10V/m with 200Hz 50% Pulse 100%AM at 900MHz
Conducted RF Immunity (IEC61000-4-6)	$\pm$1 kV line-line (DM) and $\pm$2 kV line-earth (CM) on signal ports

DriveLogix5730 Certifications

Certification	Description
c-UL-us	UL Listed for Class I, Division 2 Group A,B,C,D Hazardous Locations, certified for U.S. and Canada
CE[1]	European Union 89/336/EEC EMC Directive, compliant with: • EN 50082-2; Industrial Immunity • EN 61326; Meas./Control/Lab., Industrial Requirements • EN 61000-6-2; Industrial Immunity • EN 61000-6-4; Industrial Emissions
C-Tick[1] Pending at time of printing	Australian Radiocommunications Act, compliant with: • AS/NZS CISPR 11; Industrial Emissions
EtherNet/IP	ODVA conformance tested to EtherNet/IP specifications
c-UL-us	UL Listed for Class I, Division 2 Group A,B,C,D Hazardous Locations, certified for U.S. and Canada

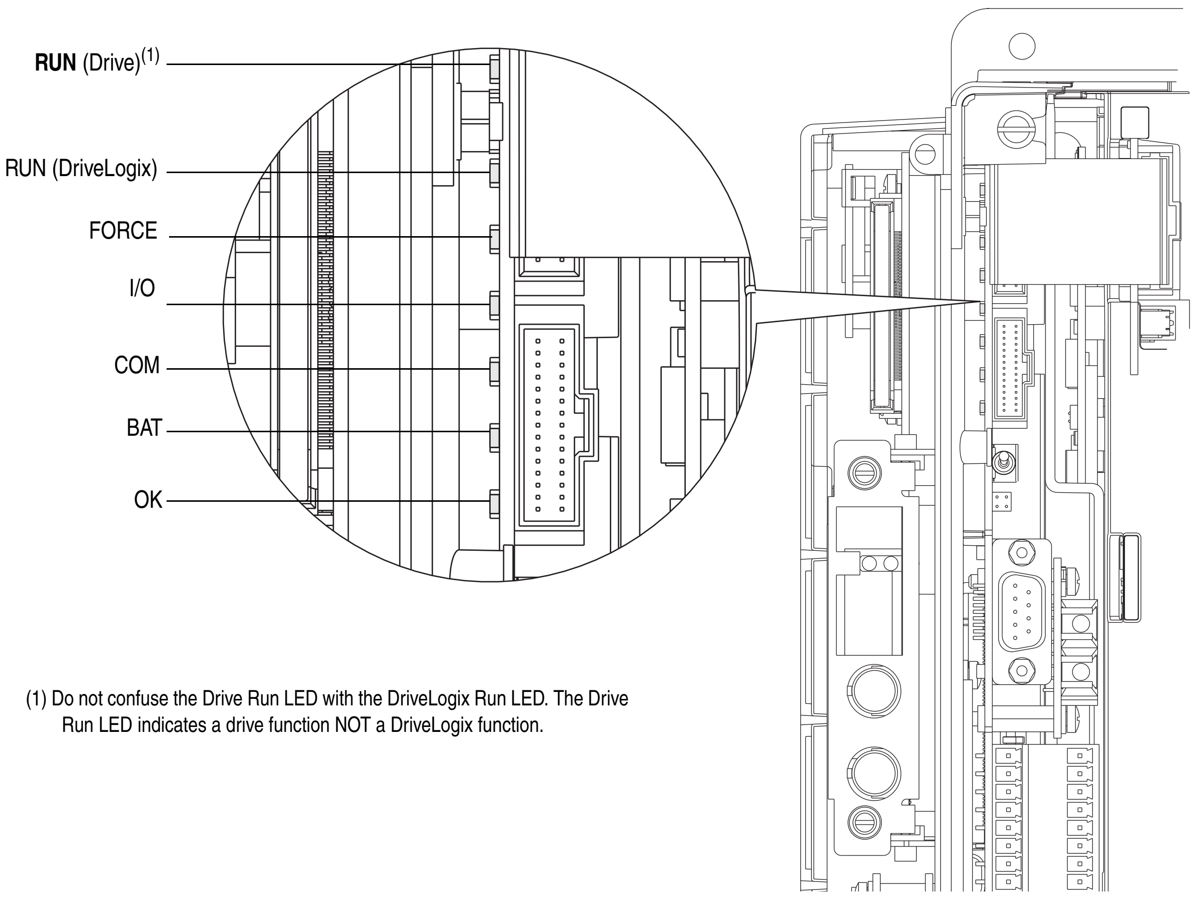
[1] See the Product Certification link at www.ab.com for Declarations of Conformity, Certificates, and other certification details.

Real-Time Clock Accuracy

Ambient °C	Accuracy
0° C	+54 to -56 seconds/month
+25° C	+9 to -124 seconds/month
+40° C	-84 to -234 seconds/month
+55° C	-228 to -394 seconds/month
+60° C	-287 to -459 seconds/month

Controller LEDs

ATTENTION: The controller LEDs are only operational when the drive is energized, and only visible with the drive door open. Servicing energized equipment can be hazardous. Severe injury or death can result from electrical shock, burn or unintended actuation of controlled equipment. Follow Safety related practices of NFPA 70E, *ELECTRICAL SAFETY FOR EMPLOYEE WORKPLACES*. DO NOT work alone on energized equipment!



(1) Do not confuse the Drive Run LED with the DriveLogix Run LED. The Drive Run LED indicates a drive function NOT a DriveLogix function.

Indicator:	**Color:**	**Description:**
RUN	off	The controller is in Program or Test mode.
	solid green	The controller is in Run mode.
FORCE	off	No tags contain I/O force values. I/O forces are inactive (disabled).
	solid amber	I/O forces are active (enabled). I/O force values may or may not exist.
	flashing amber	One or more input or output addresses have been forced to an On or Off state, but the forces have not been enabled.
BAT	off	The battery supports memory.
	solid red	Either the battery is: not installed. 95% discharged and should be replaced.
I/O	off	Either: There are *no* devices in the I/O configuration of the controller. The controller does *not* contain a project (controller memory is empty).
	solid green	The controller is communicating with all the devices in its I/O configuration.
	flashing green	One or more devices in the I/O configuration of the controller are *not* responding.
	flashing red	The controller is not communicating to any devices. The controller is faulted.
COM	off	No RS-232 activity.
	flashing green	RS-232 activity.
OK	off	No power is applied.
	flashing red	If the controller is:Then: a new controllerthe controller requires a firmware update not a new controllerA major fault occurred. To clear the fault, either: - Turn the keyswitch from PROG to RUN to PROG - Go online with RSLogix 5000 software
	solid red	The controller detected a non-recoverable fault, so it cleared the project from memory. To recover: Cycle power to the chassis. Download the project. Change to Run mode. If the OK LED remains solid red, contact your Rockwell Automation representative or local distributor.
	solid green	Controller is OK.
	flashing green	The controller is storing or loading a project to or from nonvolatile memory.

CompactFlash card LED

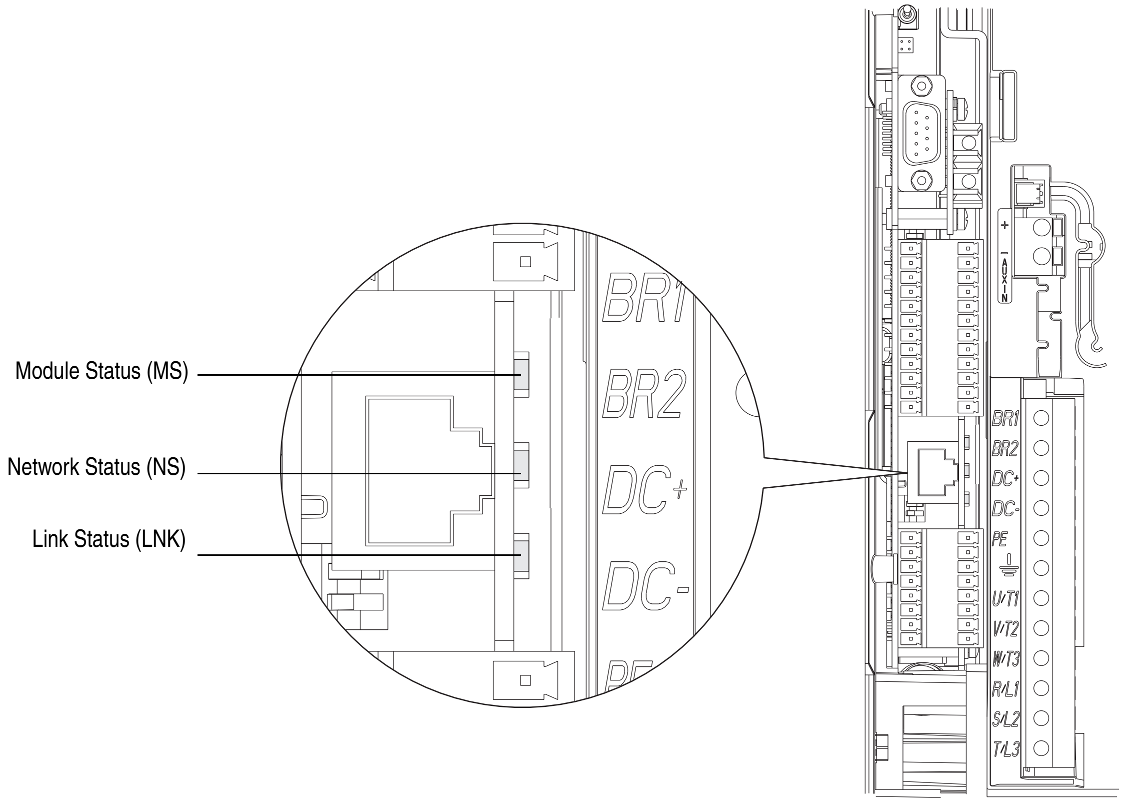
ATTENTION: The CompactFlash LED is only operational when the drive is energized, and only visible with the drive door open. Servicing energized equipment can be hazardous. Severe injury or death can result from electrical shock, burn or unintended actuation of controlled equipment. Follow Safety related practices of NFPA 70E, *ELECTRICAL SAFETY FOR EMPLOYEE WORKPLACES*. DO NOT work alone on energized equipment!

ATTENTION: Do not remove the CompactFlash card while the controller is reading from or writing to the card, as indicated by a flashing green CF LED. This could corrupt the data on the card or in the controller, as well as corrupt the latest firmware in the controller.

Indicator:	Color:	Description:
CF	off	No activity.
	flashing green	The controller is reading from or writing to the CompactFlash card.
	flashing red	CompactFlash card does not have a valid file system.

Embedded EtherNet/IP Option LEDs

ATTENTION: The Embedded EtherNet/IP Option LEDs are only operational when the drive is energized, and only visible with the drive door open. Servicing energized equipment can be hazardous. Severe injury or death can result from electrical shock, burn or unintended actuation of controlled equipment. Follow Safety related practices of NFPA 70E, *ELECTRICAL SAFETY FOR EMPLOYEE WORKPLACES*. DO NOT work alone on energized equipment!



Module Status (MS) indicator

Condition:	Status:	Indicates:	Recommended Action:
off	no power	The controller does not have power.	Check the controller power supply.
flashing green	standby	The port does not have an IP address and is operating in BOOTP mode.	Verify that the BOOTP server is running.
solid green	OK	The port is operating correctly.	Normal operation. No action required.

Condition:	Status:	Indicates:	Recommended Action:
solid red	held in reset	The controller is holding the port in reset or the controller is faulted.	Clear the controller fault. Replace the controller.
	self-test	The port is performing its power-up self-test.	Normal operation during power-up. No action required.
	major fault	An unrecoverable fault has occurred.	Cycle power to the controller. Replace the controller.
flashing red	updating firmware	The port firmware is being updated.	Normal operation during firmware update. No action required.

Network Status (NS) indicator

Condition:	Status:	Indicates:	Recommended Action:
off	not initialized	The port does not have an IP address and is operating in BOOTP mode.	Verify that the BOOTP server is running.
flashing green	no CIP connections established	The port has an IP address, but no CIP connections are established.	Normal operation if no connections are configured. No action required. If connections are configured, check connection originator for connection error code.
solid green	CIP connections established	The port has an IP address and CIP connections (Class 1 or Class 3) are established.	Normal operation. No action required.
solid red	duplicate IP address	The port has detected that the assigned IP address is already in use.	Verify that all IP addresses are unique.
flashing red/green	self-test	The port is performing its power-up self-test.	Normal operation during powerup.

Link Status (LNK) indicator

Condition:	Status:	Indicates:	Recommended Action:
off	no link	The port is not connected to a powered Ethernet device. The port cannot communicate on Ethernet.	Verify that all Ethernet cables are connected. Verify that Ethernet switch is powered.
flashing green	self-test	The port is performing its power-up self-test.	Normal operation during powerup.
	data transmission and reception	The port is communicating on Ethernet.	Normal operation. No action required.
solid green	link OK	The port is connected to a powered Ethernet device. The port can communicate on Ethernet.	Normal operation. No action required.

Battery Life

Time ON/OFF	at 25° C (77° F)	at 40° C (104° F)	at 60° C (140° F)
Always OFF	14 months	12 months	9 months
ON 8 hours per day 5 days per week	18 months	15 months	12 months
ON 16 hours per day 5 days per week	26 months	22 months	16 months
Always ON	There is almost no drain on the battery when the controller is always ON.		

Battery duration after the LED turns on

The battery indicator (BAT) warns you when the battery is low. These durations are the amounts of time the battery will retain controller memory from the time the controller is powered down after the LED first turns on.

Temperature	Duration
60° C	8 days
25° C	25 days

Access Procedures

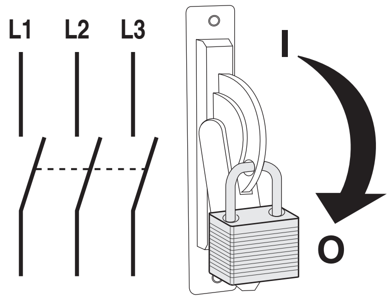
Removing Power from Drive and Compact I/O

ATTENTION: To avoid an electric shock hazard, verify that the voltage on the bus capacitors has discharged before performing any work on the drive. Measure the DC bus voltage at the +DC & –DC terminals of the Power Terminal Block. The voltage must be zero.

Remove power before making or breaking cable connections. When you remove or insert a cable connector with power applied, an electrical arc may occur. An electrical arc can cause personal injury or property damage by:

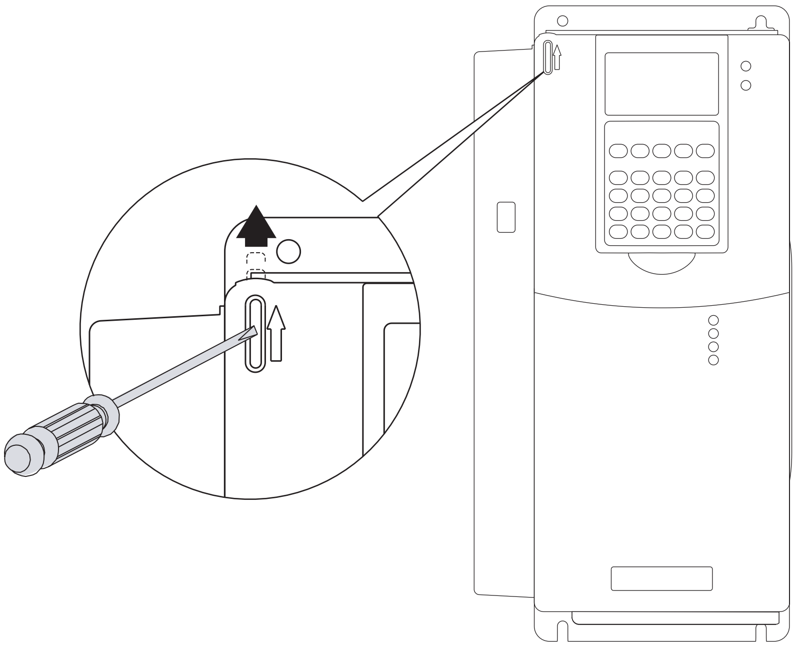
- sending an erroneous signal to your system's field devices, causing unintended machine motion
- causing an explosion in a hazardous environment

Electrical arcing causes excessive wear to contacts on both the module and its mating connector. Worn contacts may create electrical resistance.



Task	Description
A	Turn off and lock out input power. Wait five minutes.
B	Verify that there is no voltage at the drive's input power terminals.
C	Measure the DC bus voltage at the +DC & -DC terminals on the Power Terminal Block. The voltage must be zero.

Opening Door Over Power Structure and Main Control Board

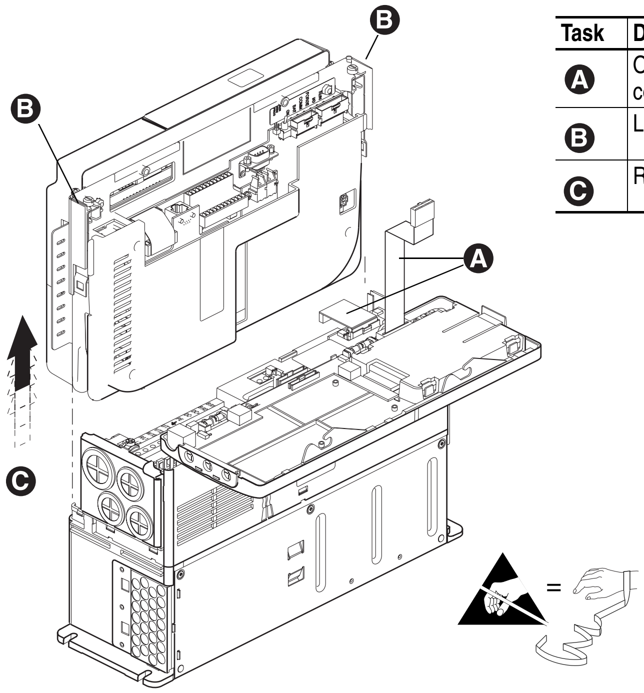


Frames 1-4
Locate the slot in the upper left corner. Slide the locking tab up and swing the cover open. Special hinges allow cover to move away from drive and lay on top of adjacent drive (if present).

Frame 5
Slide the locking tab up, loosen the right-hand cover screw and remove.

Frame 6
Loosen 2 screws at bottom of drive cover. Carefully slide bottom cover down & out. Loosen the 2 screws at top of cover and remove.

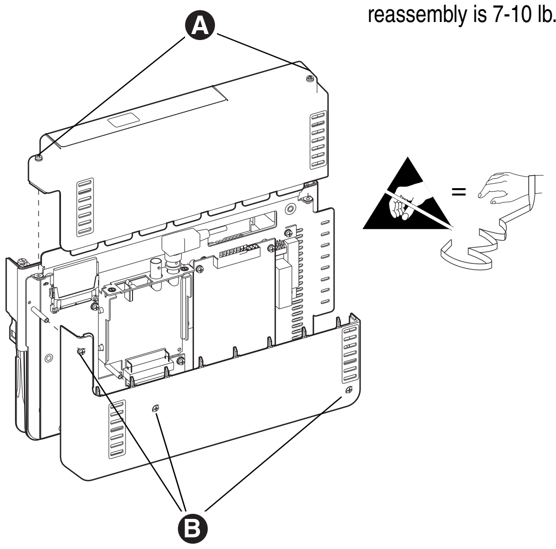
Removing the Control Cassette from Frame 1-6 Size Drives



Task	Description
A	Open the door of the power structure and disconnect the cables that connect to the main board
B	Loosen screws on face of cassette
C	Remove the cassette

Removing the Outside Covers

Task	Description
A	Loosen screws on face of front cover and remove the cover
B	Loosen screws on side of rear cover and remove the cover

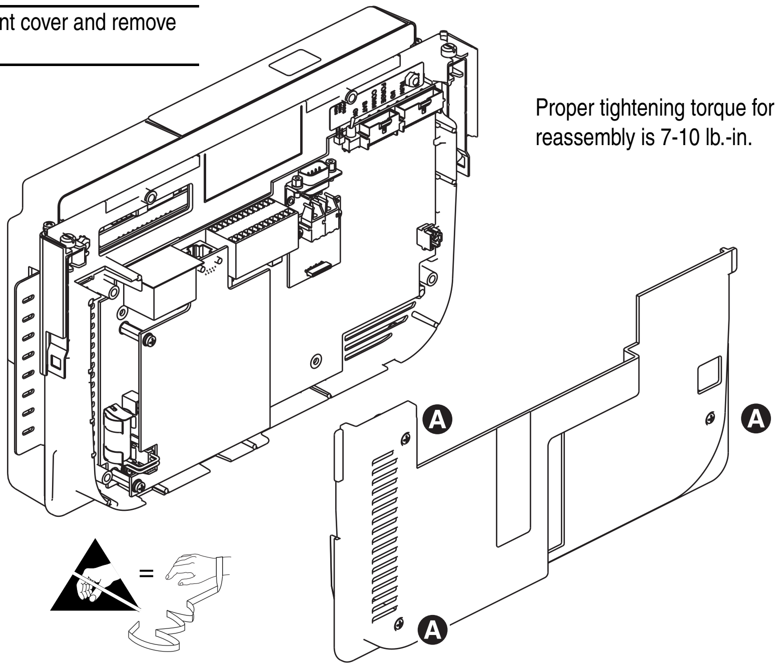


Proper tightening torque for reassembly is 7-10 lb.-in.

Removing the Inside Cover

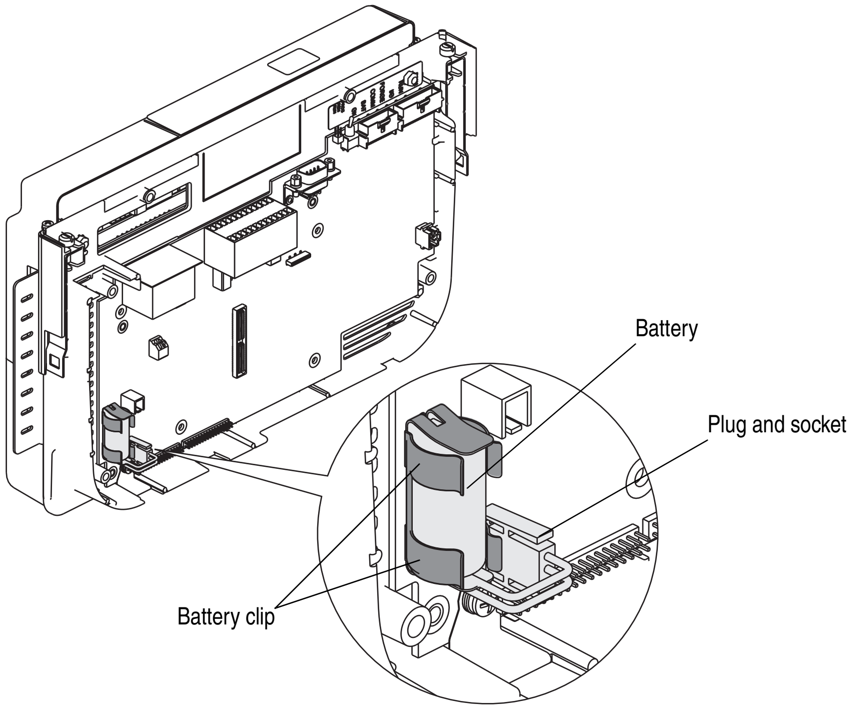
Task	Description
A	Loosen screws on face of front cover and remove the cover

Proper tightening torque for reassembly is 7-10 lb.-in.



Connecting the Battery

Allen-Bradley ships the DriveLogix controller with the battery installed, but disconnected. You must connect the battery while installing the drive.



Storing the Battery

ATTENTION: Store batteries in a cool, dry environment. We recommend 25° C with 40% to 60% relative humidity. You may store batteries for up to 30 days between -45° to 85° C, such as during transportation. To avoid possible leakage, *do not* store batteries above 60° C for more than 30 days.

Replacing the Battery

ATTENTION: Servicing energized equipment can be hazardous. Severe injury or death can result from electrical shock, burn or unintended actuation of controlled equipment. Follow Safety related practices of NFPA 70E, *ELECTRICAL SAFETY FOR EMPLOYEE WORKPLACES*. DO NOT work alone on energized equipment!

ATTENTION: The controller uses a lithium battery, which contains potentially dangerous chemicals. Before handling or disposing a battery, review *Guidelines for Handling Lithium Batteries*, publication AG-5.4.

1. Open the drive door and connect to the appropriate communication port on the controller while the drive and the DriveLogix5730 controller are energized.
2. Upload the controller's memory and program to a computer with RSLogix 5000 programming software.
3. Remove power from the drive and the DriveLogix5730 controller. Verify that the voltage on the bus capacitors has discharged before performing any work on the drive. Measure the DC bus voltage at the +DC & –DC terminals of the Power Terminal Block. The voltage must be zero.
4. Remove control cassette.
5. Does the existing battery show signs of leakage or damage?

If:	Then:
Yes	Before handling the battery, review *Guidelines for Handling Lithium Batteries*, publication AG-5.4.
No	Go to the next step.

6. Remove the old battery, by unplugging the battery and opening the clip which secures the battery.
7. Install a new 1769-BA battery, by plugging the battery lead into the socket.

ATTENTION: Only install a 1769-BA battery. If you install a different battery, you may damage the controller.

8. Attach the battery label. Write on the battery label the date you install the battery.
9. Secure the new battery in the battery clip.
10. Re-install the control cassette.
11. Energize the drive and the DriveLogix5730 controller.

12. On the front of the controller, is the BATTERY LED off?

If:	Then:
Yes	Go to the next step.
No	A. Check that the battery is correctly connected to the controller. B. If the BATTERY LED remains on, install another 1769-BA battery. C. If the BATTERY LED remains on after you complete Step B., contact your Rockwell Automation representative or local distributor.

13. Download the controller's memory and program from the computer with RSLogix 5000 programming software.

14. Remove power from the drive and the DriveLogix5730 controller. Verify that the voltage on the bus capacitors has discharged before performing any work on the drive. Measure the DC bus voltage at the +DC & –DC terminals of the Power Terminal Block. The voltage must be zero.

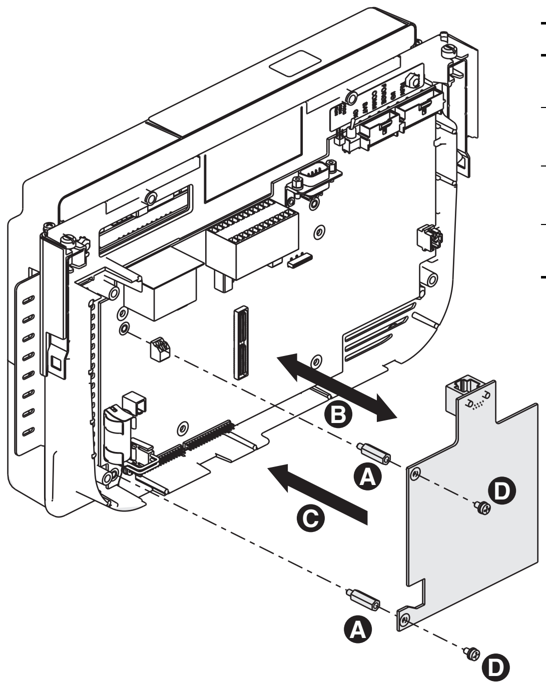
15. Disconnect from the communication port.

16. Re-install the cover(s).

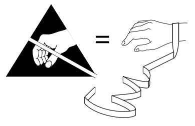
17. Dispose the old battery according to state and local regulations.

ATTENTION: Do not incinerate or dispose lithium batteries in general trash collection. They may explode or rupture violently. Follow state and local regulations for disposal of these materials. You are legally responsible for hazards created while your battery is being disposed.

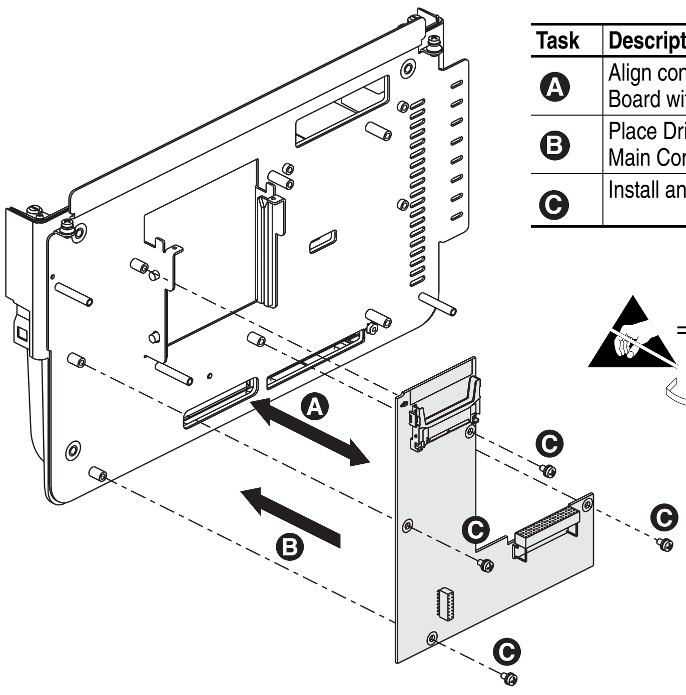
Installing the Embedded EtherNet/IP Option Board



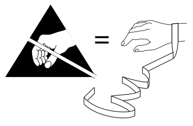
Task	Description
A	Install and tighten standoffs (7-10 lb.-in.)
B	Align connector on Embedded EtherNet/IP Option Board with connector on Main Control Board
C	Place Embedded EtherNet/IP Option Board on the Main Control Board
D	Install and tighten screws (7-10 lb.-in.)



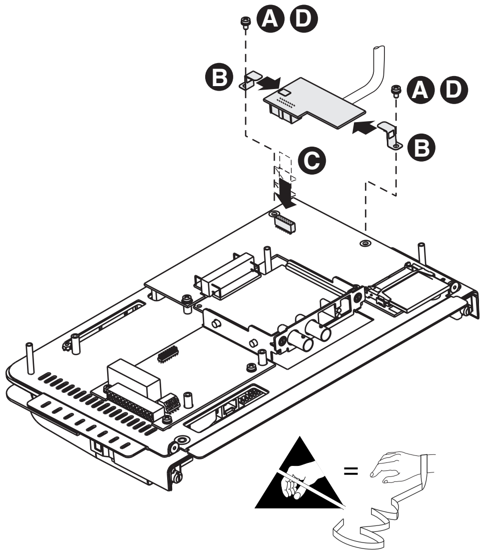
Installing the DriveLogix5730 Expansion Board



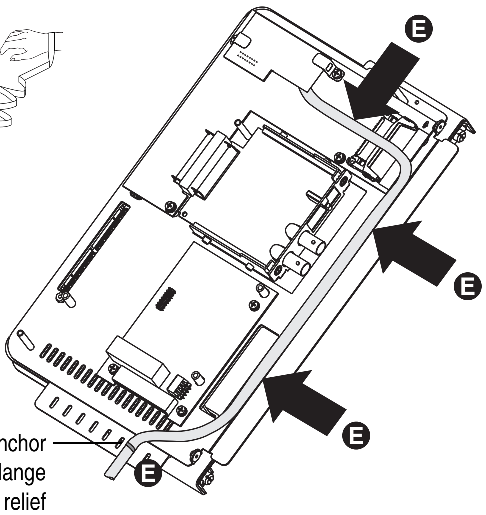
Task	Description
A	Align connector on DriveLogix5730 Expansion Board with connector on Main Control Board
B	Place DriveLogix5730 Expansion Board on the Main Control Board
C	Install and tighten screws (7-10 lb.-in.)



Installing the Compact I/O Cable

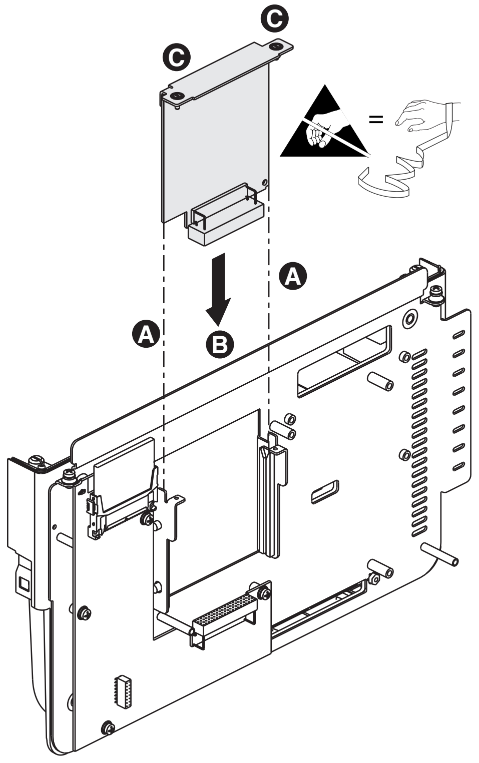


Task	Description
A	Remove screws
B	Install clips on controller end of cable
C	Plug controller end of cable into mating connector on the Logix Expansion Board
D	Install and tighten screws (7-10 lb.-in.)
E	Route and secure cable


E
E
E
E
Use cable tie to anchor cable to slots on this flange for strain relief

Installing the Communications Daughtercard

Task	Description
A	Align circuit board in Communications Daughtercard with rail on Daughtercard slot
B	Insert the Communications Daughtercard
C	Tighten screws (7-10 lb.-in.)



Important: This procedure can be performed without removing the cassette, if the Logix Expansion Board is already installed.

Notes:

Publication 20D-UM003A-EN-P – July 2004

Allen-Bradley *DriveLogix 5730* **User Manual**