POINT I/O ASCII Modules

Catalog Numbers 1734-232ASC, 1734-485ASC, 1734-485ASCK

Allen-Bradley
by ROCKWELL AUTOMATION

User Manual

Original Instructions

Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

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.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

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.

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 a hazard, and recognize the consequence.

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

Labels may also be on or inside the equipment to provide specific precautions.

SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.

BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

Preface

Chapter 1

Overview of the POINT I/O ASCII Modules

Chapter 2

Configure the ASCII Module

Chapter 3

Module Tag Definitions

Chapter 4

Diagnostics

Appendix A

ASCII Module with ControlLogix System Application Examples

About This Publication

Read this preface to familiarize yourself with the rest of the manual. It provides information concerning:

- Who should use this manual
- The purpose of this manual
- Related documentation

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Purpose of this Manual

This manual describes how to use your POINT I/O™ 1734-232ASC serial interface module (for the RS-232 network) or 1734-485ASC and 1734-485ASCK modules (for the RS-485 or RS-422 network) in Logix 5000® control systems.

Make sure that you are familiar with the following:

- Use of a controller in a Logix 5000 control system
- Use of an EtherNet/IP™ network, if the POINT I/O modules are installed in a remote location from the controller that is accessible via the EtherNet/IP network
- Studio 5000 Logix Designer® application environment

IMPORTANT Remember the following when you use POINT I/O module:

- You can use POINT I/O modules directly with CompactLogix™ 5370 controllers. However, for all other CompactLogix and ControlLogix® controllers you need to use a POINT I/O adapter.
- You must use the Studio 5000 Logix Designer application, version 17 or later, to configure POINT I/O modules.

Download Firmware, Add-on Profile, EDS, and Other Files

Download firmware, associated files (such as Add-on Profile, EDS, and DTM), and access product release notes from the Product Compatibility and Download Center at rok.auto/pcdc.

Summary of Changes

This publication contains the following new or updated information. This list includes substantive updates only and is not intended to reflect all changes.

Topic	Page
Updated template	throughout
Changed from DeviceNet® to EtherNet/IP	throughout
Added inclusive language acknowledgment	5
Removed the installation section	8
Updated configuration chapter with EtherNet/IP network and Studio 5000 Logix Designer	15
Added a module tag definition chapter	23
Removed the specifications appendix	37

Additional Resources

These documents contain additional information concerning related products from Rockwell Automation.

Resource	Description
POINT I/O RS-232 and RS-485 ASCII Modules Installation Instructions, publication 1734-IN588	Describes how to install and wire the POINT I/O ASCII modules.
POINT I/O Selection Guide, publication 1734-SG001	Provides specifications for the entire POINT I/O family of products.
System Security Design Guidelines Reference Manual, publication SECURE-RM001	Provides guidance on how to conduct security assessments, implement Rockwell Automation products in a secure system, harden the control system, manage user access, and dispose of equipment.
EtherNet/IP Network Configuration User Manual, publication ENET-UM006	Describes how to configure and use EtherNet/IP devices to communicate on the EtherNet/IP network.
Ethernet Reference Manual, publication ENET-RM002	Describes basic Ethernet concepts, infrastructure components, and infrastructure features.
Industrial Components Preventive Maintenance, Enclosures, and Contact Ratings Specifications, publication IC-TD002	Provides a quick reference tool for Allen-Bradley® industrial automation controls and assemblies.
Safety Guidelines for the Application, Installation, and Maintenance of Solid-state Control, publication SGI-1.1	Designed to harmonize with NEMA Standards Publication No. ICS 1.1-1987 and provides general guidelines for the application, installation, and maintenance of solid-state control in the form of individual devices or packaged assemblies incorporating solid-state components.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, rok.auto/certifications	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at rok.auto/literature.

Chapter 1

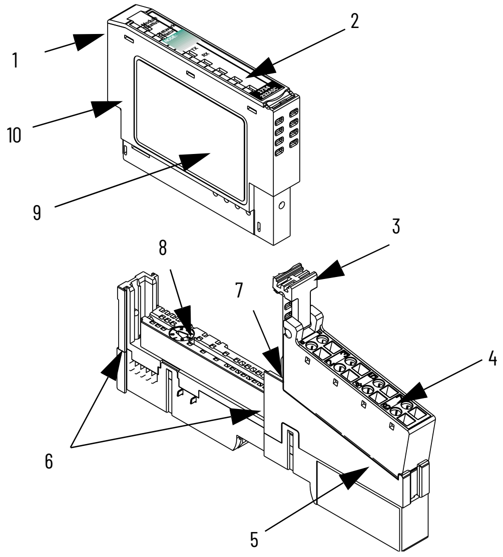
Overview of the POINT I/O ASCII Modules

General Information on the ASCII Modules

The POINT I/O ASCII modules provide a flexible EtherNet/IP interface to a wide variety of RS-232, RS-485, and RS-422 ASCII devices. The modules provide the communication connections to the ASCII device.

The 1734-232ASC module connects to the RS-232 network while the 1734-485ASC module connects to the RS-485 or RS-422 network.

POINT I/O RS-232 and RS-485 ASCII Modules



Component Description

	Description		Description
1	Module locking mechanism	6	Interlocking side pieces

Component Description (Continued)

	Description		Description
2	Slide-in writable Label	7	DIN rail locking screw (orange)
3	RTB removing handle	8	Mechanical keying (orange)
4	Removable Terminal Block (RTB)	9	Module wiring diagram
5	Mounting base	10	Insertable I/O module

Cable Pinouts for Standard DB Connectors

Table 1 shows how to connect cable pinouts for standard DB connectors.

Table 1 - ASCII Functions

	Pin Number	
Function	DB-25	DB-9
Transmit Data (TXD)	2	3
Receive Data (RXD)	3	2
Signal Ground (SG)	7	5

Install a Serial Network

The communication between your serial devices and the:

- 1734-232ASC is an RS-232 3-wire network.
- 1734-485ASC is an RS-485 2-wire network or RS-422 4-wire network.

1. Connect an appropriate cable to your device.
2. Connect the other end of the cable to the ASCII module using the designated positions on the 1734-TB terminal block.

   Notice the terminal markings on the ASCII module case.
3. Turn on power to the serial device and the ASCII module.
4. Configure the ASCII buffer sizes on the ASCII module.

The defaults are 20 and 20. If more than 20 bytes are required for the transmit or receive buffers, set the appropriate parameters. See Read Data from the ASCII Modules into ControlLogix on page 38 and Transmit Data from ControlLogix Through the ASCII Modules on page 39.

Read Serial Device Input Data from the ASCII Module

1. Add the ASCII module to a Studio 5000 Logix Designer application project, see Create a New Module on page 15.
2. Connect the serial side of the ASCII module to your computer's serial port or another serial device.
3. From the Module Information dialog box, check the module information.
4. Make sure that the ASCII module is in the default factory configuration.
5. Set the communication rate and framing format of the serial port to the communication rate and framing format of the serial device that you are using.
6. Put the controller in Run mode.
7. Direct the device that you are communicating with to send data.
   You can see the data sent from the device in the controller tag. See Read Data from the ASCII Modules into ControlLogix on page 38.
   The default assembly of the poll response message is shown in Table 2.

Table 2 - Default Receive Data Assembly Format (Default Mode)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5…23	Byte 24
Receive Record Number	Status Byte	Reserved	Length	ASCII Data	<CR> (Terminator)

Write Serial Output Data to the ASCII Module

1. Add the ASCII module to a Studio 5000 Logix Designer application project, see Create a New Module on page 15.
2. Connect the serial side of the ASCII module to your computer's serial port or another serial device.
3. From the Module Information dialog box, check the module information.
4. Make sure that the ASCII module is in the default factory configuration.
5. Set the communication rate and framing format of the serial port to the communication rate and framing format of the serial device that you are using.
6. Enter the serial data that you wish to send in the transmit data controller tag.
7. Enter the length of the data in the length byte to reflect the length you wish to send and press Enter.
8. Set the transmit record number from the controller tag, and apply. The ASCII module transmits the characters that you entered in the controller tag to the connected serial device.

Table 3 - Default Transmit Data Assembly Format (Default Mode)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5…23	Byte 24
Reserved	Transmit Record Number	Reserved	Length	ASCII Data	<CR> (Terminator)

Configure ASCII Modules

You must create a Studio 5000 Logix Designer application project for the Logix 5000 controller that owns the ASCII module. The project includes module configuration data for the ASCII module.

The Studio 5000 Logix Designer application transfers the project to the owner-controller during the program download. Data is then transferred to the ASCII module over the EtherNet/IP network.

ASCII modules can operate immediately after receiving the configuration data.

Connections

During module configuration, you must define the module. Among the Module Definition parameters, you must choose a connection type for the module. A connection is a real-time data transfer link between the owner-controller and the module that occupies the slot that the configuration references.

When you download module configuration to a controller, the controller attempts to establish a connection to each module in the configuration.

Because part of module configuration includes a slot in the POINT I/O system, the owner-controller checks for the presence of a module there. If a module is detected, the owner-controller sends the configuration. One of the following occurs:

- If the configuration is appropriate to the module detected, a connection is made and operation begins.
- If the configuration is not appropriate to the module detected, the data is rejected and the Studio 5000 Logix Designer application indicates that an error occurred.

  The configuration can be inappropriate for many reasons. For example, a mismatch in electronic keying that helps prevents normal operation.

The owner-controller monitors its connection with a module. Any break in the connection, for example, the loss of power to the backplane, causes a fault. The Studio 5000 Logix Designer application monitors the fault status tags to indicate when a fault occurs on a module.

Connection Types Available with POINT I/O ASCII Modules

When configuring an ASCII module, you must define the module. Connection is a required parameter in the Module Definition. The choice determines what data is exchanged between the owner-controller and the module.

Table 4 describes the connection types that you can use with an ASCII module.

Table 4 - Connections - POINT I/O ASCII Modules

Connection Type	Description
Data	The module returns the following to the owner-controller: • General fault data • Input data
Listen Only	When a Listen Only data connection is used, another controller owns the module. A controller that makes a Listen Only connection to the module does not write configuration for the module. It merely listens to the data exchanged with the owner-controller.

For more information on the connection choices available with ASCII module, see the Studio 5000 Logix Designer application.

Requested Packet Interval

The Requested Packet Interval (RPI) is a configurable parameter that defines a specific rate at which data is exchanged between the owner-controller and the module.

You set the RPI value during initial module configuration and can adjust it as necessary after module operation has begun. Valid RPI values are 2…750 ms.

IMPORTANT If you change the RPI while the project is online, the connection to the module is closed and reopened in one of the following ways:

- You inhibit the connection to the module, change the RPI value, and uninhibit the connection.
- You change the RPI value. In this case, the connection is closed and reopened immediately after you apply the change to the module configuration.

For more information on guidelines for specifying RPI rates, see the Logix 5000 Controllers Design Considerations Reference Manual, publication 1756-RM094.

Listen Only Mode

Any controller in the system can listen to the data from an ASCII module. An owner-controller exchanges data with the ASCII module.

Other controllers can use a Listen Only connection with the module. In this case, the 'listening' controller can only listen to input data or 'echoed' output data. The listening controller does not own the module configuration or exchange other data with the module.

During the I/O configuration process, you can specify a Listen Only connection. For more information on Connection options, see Module Definition on page 17.

IMPORTANT Remember the following:

- If a controller attempts to use a Listen Only connection to a module but the owner-controller connection uses the Unicast option, the attempt at a Listen Only connection fails.

The 'Listen Only' controller receives data from the module as long as a connection between an owner-controller and module is maintained.

- If the connection between an owner-controller and the module is broken, the module stops sending data and connections to all 'listening controllers' are also broken.

Operating Mode Selections

The ASCII module has several different operating modes, some of which are available only in certain combinations. Major options are summarized below, with additional details provided in Chapter 2.

Transmit Handshake and Transmit Immediate Option

This option defines when the ASCII module transmits data out of its serial port.

With Transmit Handshake active (default), the ASCII modules keys on the second byte (Transmit Record Number) of the poll command or explicit message to determine whether the consumed data from controller is to be transmitted out the serial port. While this Transmit Record Number is static, the module does not transmit the command data. Upon sensing a change in value of the Transmit Record Number byte, the module transmits the data. It initiates one serial transmission for each change in the Transmit Record Number value, regardless of the data string value.

In **Transmit Immediate** mode, the ASCII module ignores the Transmit Record Number byte value and initiates a new serial port transmission with each message that is received from the controller.

Carefully consider when you select this option for I/O messaging. See POINT I/O Parameters on page 25 for details.

These actions take place with either a Poll command or an explicit message that writes to the consume Assembly Object or the serial port object.

Produce Immediate and Master/Slave Handshake Option

This option defines when the ASCII module sends new data to controller.

In Produce Immediate mode (default), the ASCII module sends its most current serial port data to the controller in response to each Poll command or explicit message, or in response to an event.

With Master/Slave Handshake active, the ASCII module keys on the first byte (next Transmit Record Number) of the Poll command or explicit message to determine whether the new data from its serial port is sent to the controller. This technique is useful when the adapter must ensure that some specific actions have taken place before receiving the new serial data.

In this operating mode, the module sets the New Data Flag in its Serial Status byte when a new string is received into the serial port. When the controller is ready to receive the new data, it changes the value of the next Transmit Record Number in the next Poll command or explicit message. Upon detection of the next Transmit Record Number value change, the module updates its produce buffer with the most recent serial input string.

Pad and No Pad Option

If the Pad option is selected (default), the ASCII module always sends a fixed number of data bytes to the master. It is typically used when a Terminating Character trigger is used to stop receiving ASCII data. This is useful if the ASCII device that is connected to the ASCII module has varying data lengths and the master cannot accept varying length I/O messages. The ASCII module fills a short message with the pre-defined Pad Character. The Max Receive Length parameter defines the data length.

If the No Pad option is selected, the ASCII module’s produce I/O message resizes itself to the length of the received ASCII data. This conserves bandwidth, but causes scanners that do not support variable I/O sizes to malfunction. Do not turn off Pad Mode if you do not have (or do not know if you have) a scanner that supports variable I/O sizes. Pad Mode is ON by default.

Module Inhibiting

Module inhibiting lets you indefinitely suspend a connection, including Listen Only connections, between an owner-controller and an I/O module without removing the module from the configuration. This process lets you temporarily disable a module, such as to perform maintenance.

You can use module inhibiting in the following ways:

- You write a configuration for an I/O module but inhibit the module to help prevent it from communicating with the owner-controller. The owner does not establish a connection and the configuration is not sent to the module until the connection is uninhibited.

- In your application, a controller already owns a module, has downloaded the configuration to the module, and is exchanging data over the connection between the devices.

  In this case, you can inhibit the module and the connection to the module does not exist.

IMPORTANT	Whenever you inhibit an output module that is ProgMode enabled, it enters Program mode, and all outputs change to the state configured for Program mode. For example, if an output module is configured so that the state of the outputs transition to zero during Program mode, whenever that module is inhibited, outputs transition to zero.

You can use module inhibiting in these instances:

- You want to update an analog I/O module, for example, update the module firmware revision. Use the following procedure.
  a. Inhibit the module.
  b. Perform the update.
  c. Uninhibit the module.
- You use a program that includes a module that you do not physically possess yet. You do not want the controller to look for a module that does not yet exist. In this case, you can inhibit the module in your program until it physically resides in the proper slot.

To see where to inhibit a POINT I/O ASCII module, see Connection Category on page 18.

Electronic Keying

Electronic Keying reduces the possibility that you use the wrong device in a control system. It compares the device that is defined in your project to the installed device. If keying fails, a fault occurs. These attributes are compared.

Attribute	Description
Vendor	The device manufacturer.
Device Type	The general type of the product, for example, ASCII module.
Product Code	The specific type of the product. The Product Code maps to a catalog number.
Major Revision	A number that represents the functional capabilities of a device.
Minor Revision	A number that represents behavior changes in the device.

Table 5 shows the Electronic Keying options that are available.

Table 5 - Electronic Keying Options

Keying Option	Description
Compatible Module	Lets the installed device accept the key of the device that is defined in the project when the installed device can emulate the defined device. With Compatible Module, you can typically replace a device with another device that has the following characteristics: • Same catalog number • Same or higher Major Revision • Minor Revision as follows: – If the Major Revision is the same, the Minor Revision must be the same or higher. – If the Major Revision is higher, the Minor Revision can be any number.
Disable Keying	Indicates that the keying attributes are not considered when attempting to communicate with a device. With Disable Keying, communication can occur with a device other than the type specified in the project. **ATTENTION:** Be extremely cautious when using Disable Keying; if used incorrectly, this option can lead to personal injury or death, property damage, or economic loss. We strongly recommend that you do not use Disable Keying. If you use Disable Keying, you must take full responsibility for understanding whether the device being used can fulfill the functional requirements of the application.
Exact Match	Indicates that all keying attributes must match to establish communication. If any attribute does not match precisely, communication with the device does not occur.

Carefully consider the implications of each keying option when selecting one.

Configure the ASCII Module

This chapter describes how to configure your POINT I/O ASCII module in a Studio 5000 Logix Designer application project. You can use the default module configuration or edit the module configuration.

IMPORTANT Consider the following:

- You must use the Studio 5000 Logix Designer application, version 17 or later, to configure POINT I/O modules. Version 31 or later is slightly different from previous software versions. For example, in some cases, instead of tabs across the top of the Module Properties dialog box, the application uses categories on the left side of the dialog box.
- This chapter does not explain each user-configurable module feature that you can edit on different screens in your Studio 5000 Logix Designer application project.

Before You Begin

You must complete the following tasks before you can configure the module:

1. Create a Studio 5000 Logix Designer application project.
2. Add a POINT I/O EtherNet/IP adapter to the project.

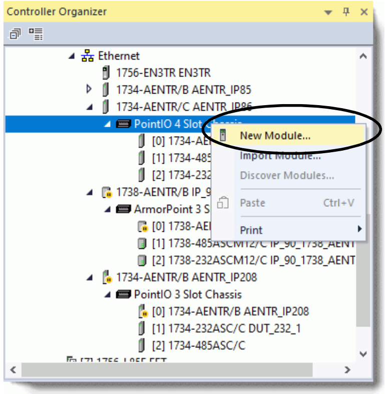
   For more information on how to add a POINT I/O EtherNet/IP adapter to a Studio 5000 Logix Designer application project, see the POINT I/O and ArmorPOINT® I/O Dual Port EtherNet/IP Adapters User Manual, publication 1734-UM017, or the POINT I/O EtherNet/IP Adapters User Manual, publication 1734-UM018.

Create a New Module

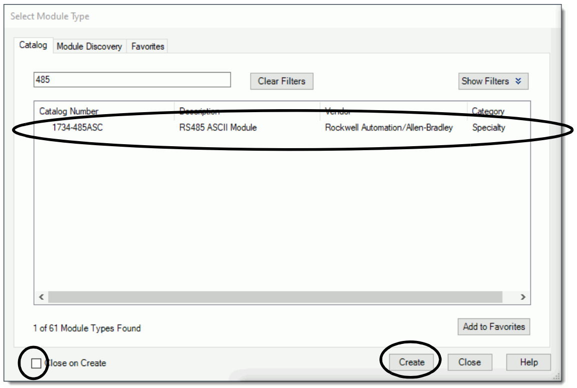
After you create a Studio 5000 Logix Designer application project and add a POINT I/O EtherNet/IP adapter to the project, complete these steps to add modules to the project.

This example shows how to add a 1734-485ASC module when the Studio 5000 Logix Designer application project is offline.

1. Right-click on the POINT I/O Chassis node and choose New Module.



2. Select the module and click Create.
You can use filters to minimize the number of modules displayed as shown.



485

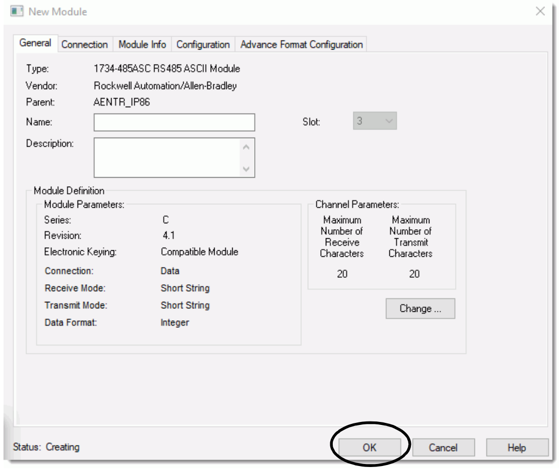
Catalog Number	Description	Vendor	Category
1734-485ASC	RS485 ASCII Module	Rockwell Automation/Allen-Bradley	Specialty

1 of 61 Module Types Found

□ Close on Create

Clear the Close on Create checkbox if you intend to add more than one module.

The New Module dialog box appears. You can click OK to use the default configuration as shown or edit the module configuration.



Type: 1734-485ASC RS485 ASCII Module
Vendor: Rockwell Automation/Allen-Bradley
Parent: AENTR_IP86
Name:
Slot: 3
Description:

Module Definition

Module Parameters:
Series: C
Revision: 4.1
Electronic Keying: Compatible Module
Connection: Data
Receive Mode: Short String
Transmit Mode: Short String
Data Format: Integer

Channel Parameters:

Maximum Number of Receive Characters	Maximum Number of Transmit Characters
20	20

Status: Creating

To add additional I/O modules in the same rack, complete one of the following:

- If you cleared the Close on Create checkbox when you created the first I/O module, repeat step 2.
- If you did not clear the Close on Create checkbox when you created the first I/O module, repeat steps 1…2.

Once added to the network, you must configure the module for use. The 1734-485ASC and 1734-232ASC modules are configured identically.

IMPORTANT	This chapter shows configuration in the offline mode. Changes set in this mode take effect when you download to the individual module.

Edit the Module Configuration Categories

You click the category names in the New Module dialog box to view and change the configuration parameters that are associated with that module.

IMPORTANT	This chapter shows how to edit configuration when you add the module to the Studio 5000 Logix Designer application project. If you access the module configuration after it has been added to the project, the dialog box is named Module Properties. The Module Properties dialog box shows the same categories as the New Module dialog box.

Some new module configuration categories apply to all POINT I/O modules. Some categories are specific to the module type.

For example purposes, the figures in this section are from a 1734-485ASC module.

The following categories apply to all POINT I/O ASCII modules and are described in this section:

- General Category
- Connection Category
- Module Info Category
- Configuration Category
- Advance Format Configuration Category

General Category

The General category appears first when you create a module. The parameters in this category are the same for all POINT I/O ASCII modules.

You use this category to complete the following tasks:

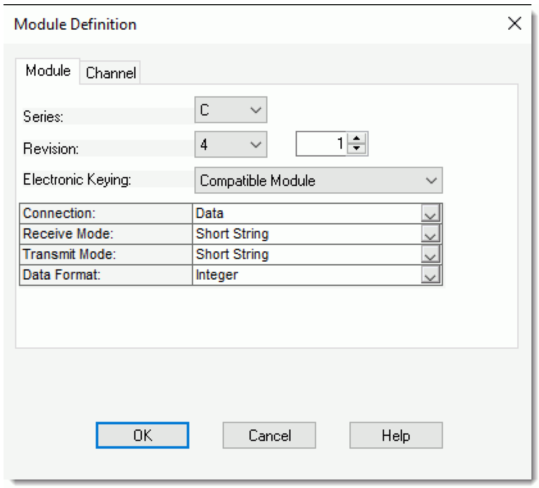
- Name the module
- Assign a slot number (required)
- Describe the module
- Access the Module Definition

Module Definition

Module Definition parameters are available on the General tab of the Module Properties dialog box in the Studio 5000 Logix Designer application project.

Table 6 describes the parameters on the Module Definition dialog box.

IMPORTANT	The graphic is an example of a Module Definition dialog box. The same set of fields and options are not available on all POINT I/O modules.

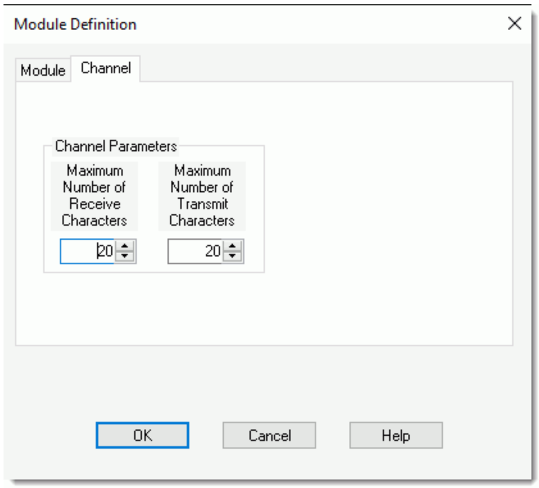


Series: C

Revision: 4 1

Electronic Keying: Compatible Module

Connection:	Data
Receive Mode:	Short String
Transmit Mode:	Short String
Data Format:	Integer



Channel Parameters

Maximum Number of Receive Characters	Maximum Number of Transmit Characters
20	20

Table 6 - Module Definition Parameters

Parameter	Definition	Available Choices[1]
Series	Module hardware series	Module-specific
Revision	Module firmware revision, including major and minor revision levels	Module-specific
Electronic Keying	Software method by which you reduce the possibility of using the wrong device in a control system. For more information, see the following: • Electronic Keying on page 13 • Electronic Keying in Logix 5000 Control Systems Application Technique, publication LOGIX-AT001	Exact Match Compatible Module Disable Keying
Connection	Determines the following for the module type you configure: • Available configuration parameters • Data type transferred between the module and the controller • Which tags are generated when configuration is complete	Data Listen Only[2]
Receive Mode	Input data for the module that is being defined.	Short_String String Array
Transmit Mode	Output data for the module that is being defined.	Short_String String Array
Data Format	Module data format transferred between module and controller.	Integer
Channel Parameters	Maximum number of characters that the ASCII module expects to receive and transmit.	20 (default) 1…127

[1] The range of available choices varies by module type.

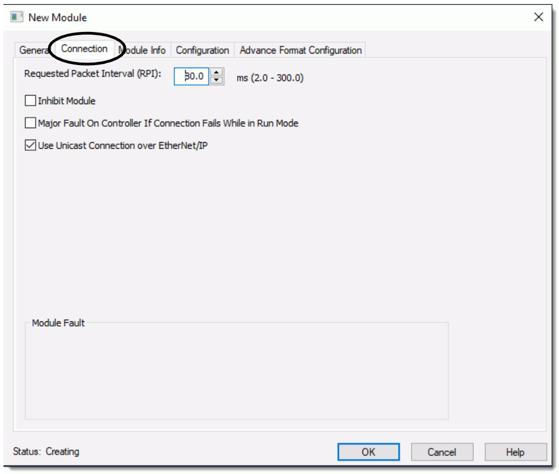
[2] In Listen Only connection, the controller and module establish communication without the controller sending any configuration or output data to the module. A full input data connection is established but depends on the connection between the owner-controller and the module.

Connection Category

The Connection category lets you complete the following tasks:

- Set the RPI rate. For more information about the RPI, see Requested Packet Interval on page 10.
- Inhibit the module. For more information on how to inhibit the module, see Module Inhibiting on page 12.
- Configure whether a connection failure while the controller is in Run mode causes a major or minor fault.
- Configure whether to use a Unicast connection over the EtherNet/IP network.

The Module Fault area of the Connection category shows a description that includes an error code that is associated with the specific fault type.



New Module

Requested Packet Interval (RPI): 50.0 ms (2.0 - 300.0)

□ Inhibit Module

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

☑ Use Unicast Connection over EtherNet/IP

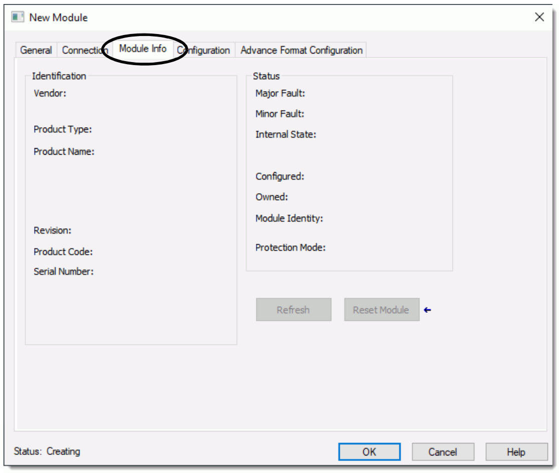
Module Fault

Status: Creating

Module Info Category

The Module Info category displays module and status information about the module when the project is online. You can use this category to complete the following:

- Identify the module identification information
- Access module status information
- Determine the system configuration and identity of the module
- Refresh the data on the screen
- Reset the module



New Module

Identification

Vendor:

Product Type:

Product Name:

Revision:

Product Code:

Serial Number:

Status

Major Fault:

Minor Fault:

Internal State:

Configured:

Owned:

Module Identity:

Protection Mode:

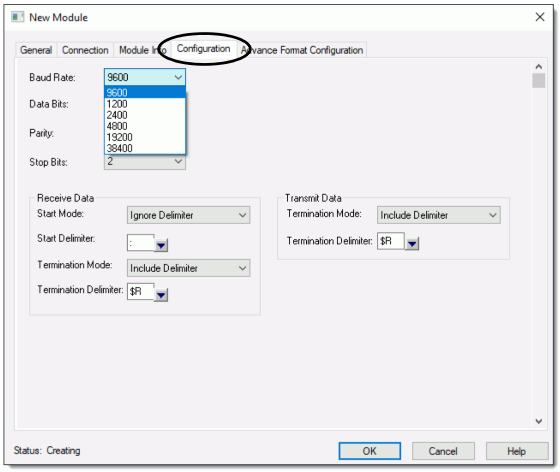
Status: Creating

Configuration Category

The Configuration category lets you complete the following tasks:

- Set the Baud rate. For more information about the baud rate, see Parameter List on page 25.

- Set the data bits, stop bits, and parity in data character frames. For more information about data character frames, see Parameter List on page 25.
- Configure serial port receive and transmit data modes and delimiters. For more information about receive and transmit data settings, see Parameter List on page 25.



Baud Rate: 9600
Data Bits:
Parity:
Stop Bits: 2

Receive Data
Start Mode: Ignore Delimiter
Start Delimiter: :
Termination Mode: Include Delimiter
Termination Delimiter: $R

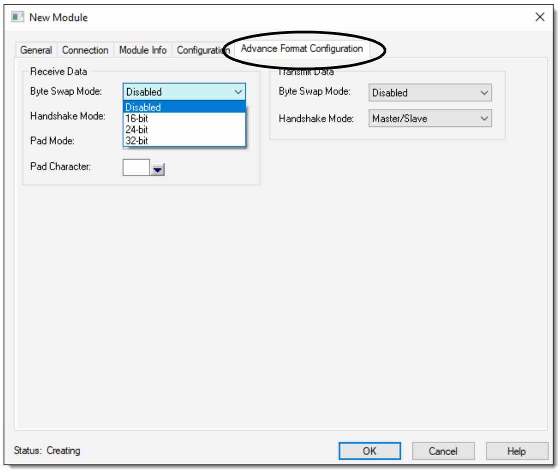
Transmit Data
Termination Mode: Include Delimiter
Termination Delimiter: $R

Status: Creating

Advance Format Configuration Category

The Advance Format Configuration category lets you complete the following tasks:

- Set the receive and transmit data swap bytes modes. For more information about Swap Bytes modes, see Set Up and Use the Swap Bytes Mode on page 28.
- Set the receive and transmit handshake modes. For more information about Handshake modes, see Produce Immediate and Master/Slave Handshake Option on page 12.
- Enable or disable Pad Mode. For more information about pad mode, see Pad and No Pad Option on page 12.
- Define the receive pad character. For more information about pad characters, see Pad and No Pad Option on page 12.



Receive Data
Byte Swap Mode: Disabled
Handshake Mode:
Pad Mode:
Pad Character:

Transmit Data
Byte Swap Mode: Disabled
Handshake Mode: Master/Slave

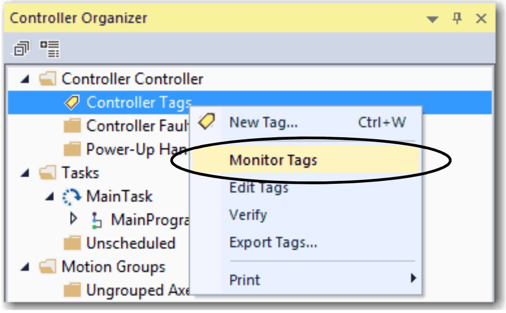
Status: Creating

View the Module Tags

When you create a module, the Studio 5000 Logix Designer application creates a set of tags that you can view in the Controller Tags. Each configured feature on your module has a distinct tag that is available for use in the controller program logic.

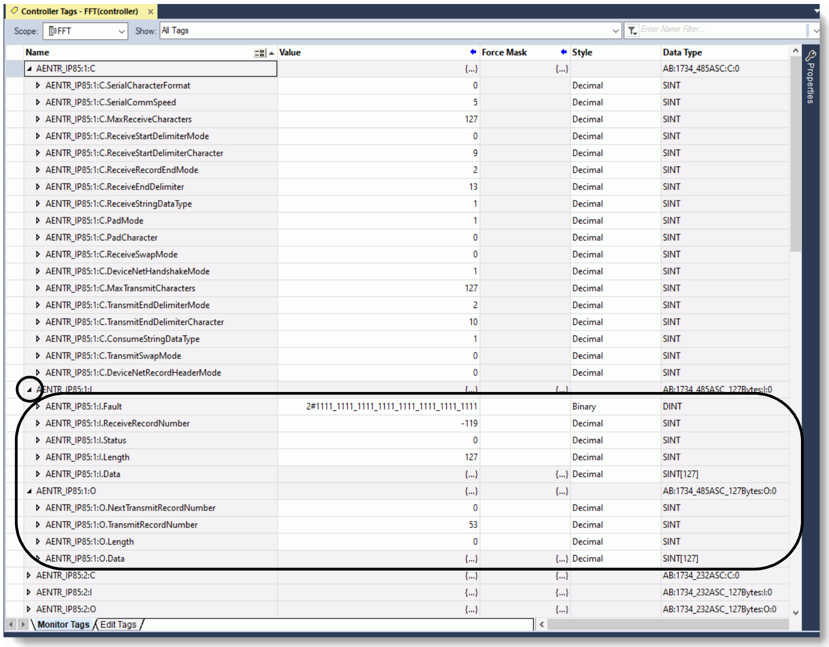
Complete the following steps to access the module tags.

1. In the Controller Organizer, right-click Controller Tags and choose Monitor Tags.



The Controller Tags dialog box appears with data.

2. To view module tags as shown, click the ▶ symbols.



Name	Value	Force Mask	Style	Data Type
AENTR_IP85:1:C	{...}	{...}		AB:1734_485ASC:C:0
AENTR_IP85:1:C.SerialCharacterFormat	0		Decimal	SINT
AENTR_IP85:1:C.SerialCommSpeed	5		Decimal	SINT
AENTR_IP85:1:C.MaxReceiveCharacters	127		Decimal	SINT
AENTR_IP85:1:C.ReceiveStartDelimiterMode	0		Decimal	SINT
AENTR_IP85:1:C.ReceiveStartDelimiterCharacter	9		Decimal	SINT
AENTR_IP85:1:C.ReceiveRecordEndMode	2		Decimal	SINT
AENTR_IP85:1:C.ReceiveEndDelimiter	13		Decimal	SINT
AENTR_IP85:1:C.ReceiveStringDataType	1		Decimal	SINT
AENTR_IP85:1:C.PadMode	1		Decimal	SINT
AENTR_IP85:1:C.PadCharacter	0		Decimal	SINT
AENTR_IP85:1:C.ReceiveSwapMode	0		Decimal	SINT
AENTR_IP85:1:C.DeviceNetHandshakeMode	1		Decimal	SINT
AENTR_IP85:1:C.MaxTransmitCharacters	127		Decimal	SINT
AENTR_IP85:1:C.TransmitEndDelimiterMode	2		Decimal	SINT
AENTR_IP85:1:C.TransmitEndDelimiterCharacter	10		Decimal	SINT
AENTR_IP85:1:C.ConsumeStringDataType	1		Decimal	SINT
AENTR_IP85:1:C.TransmitSwapMode	0		Decimal	SINT
AENTR_IP85:1:C.DeviceNetRecordHeaderMode	0		Decimal	SINT
AENTR_IP85:1:I	{...}	{...}		AB:1734_485ASC_127Bytes:I:0
AENTR_IP85:1:I.Fault	2#1111_1111_1111_1111_1111_1111_1111_1111		Binary	DINT
AENTR_IP85:1:I.ReceiveRecordNumber	-119		Decimal	SINT
AENTR_IP85:1:I.Status	0		Decimal	SINT
AENTR_IP85:1:I.Length	127		Decimal	SINT
AENTR_IP85:1:I.Data	{...}	{...}	Decimal	SINT[127]
AENTR_IP85:1:O	{...}	{...}		AB:1734_485ASC_127Bytes:O:0
AENTR_IP85:1:O.NextTransmitRecordNumber	0		Decimal	SINT
AENTR_IP85:1:O.TransmitRecordNumber	53		Decimal	SINT
AENTR_IP85:1:O.Length	0		Decimal	SINT
AENTR_IP85:1:O.Data	{...}	{...}	Decimal	SINT[127]
AENTR_IP85:2:C	{...}	{...}		AB:1734_232ASC:C:0
AENTR_IP85:2:I	{...}	{...}		AB:1734_232ASC_127Bytes:I:0
AENTR_IP85:2:O	{...}	{...}		AB:1734_232ASC_127Bytes:O:0

For more information on module tags, see Module Tag Definitions on page 23.

Notes:

Module Tag Definitions

Module tags are created when you add a module to the Studio 5000 Logix Designer application project.

The set of module tags that are associated with a module depends on the module type and Module Definition choices that are made during module configuration. For example, if you use a Listen Only Connection in the Module Definition, the Studio 5000 Logix Designer application creates only Input tags for that module.

The following types of tags are available with POINT I/O modules:

- Configuration
- Input
- Output

The tables contained in this section list all tags available with a module. Not all tags in the list are used when that module type is added to a project. Tag use varies by module configuration.

Tag Name Conventions

The module tag names use defined naming conventions. The conventions are as follows:

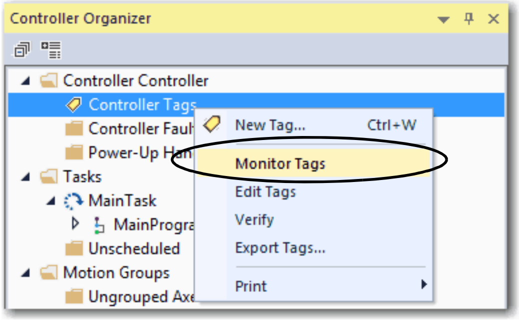
Example tag name = AENTR_IP85:1:I.Data

- AENTR_IP85 = name of the POINT I/O EtherNet/IP adapter in the POINT I/O system.
- 1 = slot number
- I = tag type
  The possible POINT I/O analog module tag types are C (configuration), I (input), and O (output).
- Data = tag function
  In this case, Data represents the input data that is returned to the owner-controller.

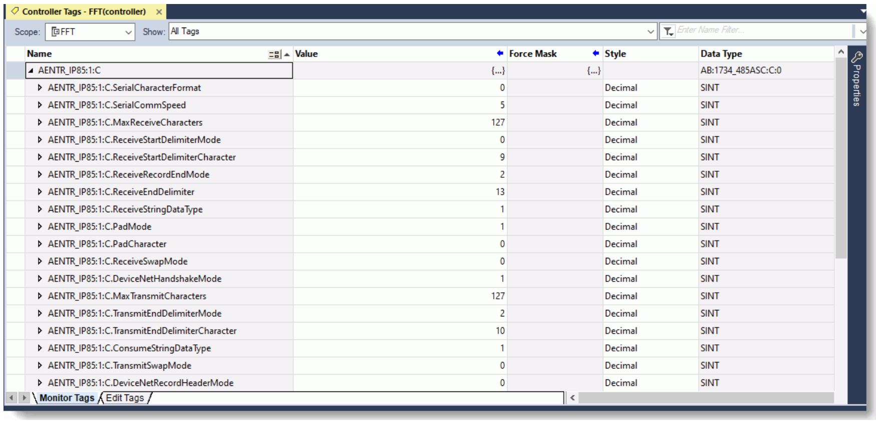
Access the Tags

You view tags from the Configuration Tags.

1. Open your Studio 5000 Logix Designer application project.
2. Right-click Controller Tags and choose Monitor Tags.



3. Open the tags as necessary to view specific tags.



Scope: FFT Show: All Tags

Name	Value	Force Mask	Style	Data Type
AENTR_IP85:1:C	{...}	{...}		AB:1734_485ASC:C:0
AENTR_IP85:1:C.SerialCharacterFormat	0		Decimal	SINT
AENTR_IP85:1:C.SerialCommSpeed	5		Decimal	SINT
AENTR_IP85:1:C.MaxReceiveCharacters	127		Decimal	SINT
AENTR_IP85:1:C.ReceiveStartDelimiterMode	0		Decimal	SINT
AENTR_IP85:1:C.ReceiveStartDelimiterCharacter	9		Decimal	SINT
AENTR_IP85:1:C.ReceiveRecordEndMode	2		Decimal	SINT
AENTR_IP85:1:C.ReceiveEndDelimiter	13		Decimal	SINT
AENTR_IP85:1:C.ReceiveStringDataType	1		Decimal	SINT
AENTR_IP85:1:C.PadMode	1		Decimal	SINT
AENTR_IP85:1:C.PadCharacter	0		Decimal	SINT
AENTR_IP85:1:C.ReceiveSwapMode	0		Decimal	SINT
AENTR_IP85:1:C.DeviceNetHandshakeMode	1		Decimal	SINT
AENTR_IP85:1:C.MaxTransmitCharacters	127		Decimal	SINT
AENTR_IP85:1:C.TransmitEndDelimiterMode	2		Decimal	SINT
AENTR_IP85:1:C.TransmitEndDelimiterCharacter	10		Decimal	SINT
AENTR_IP85:1:C.ConsumeStringDataType	1		Decimal	SINT
AENTR_IP85:1:C.TransmitSwapMode	0		Decimal	SINT
AENTR_IP85:1:C.DeviceNetRecordHeaderMode	0		Decimal	SINT

Configuration Assembly

The ASCII modules support a configuration assembly that is accessed through the Assembly Object (Class 4), Instance 103. The configuration assembly is 18 bytes.

Table 7 - Configuration Assembly

Byte	Parameter	Parameter Instance[1]
0	ASCII Serial Character Format	1
1	ASCII Serial Comm Speed	2
2	ASCII Max Number of Receive Characters	3
3	ASCII Receive Record Start Mode	4
4	ASCII Receive Start Delimiter	5
5	ASCII Receive Record End Mode	6
6	ASCII Receive End Delimiter	7
7	ASCII Receive String Data Type	8
8	ASCII Pad Mode	9
9	ASCII Pad Character	10
10	ASCII Receive Swap Mode	11
11	Ethernet Handshake Mode	12
12	ASCII Max Number of Transmit Characters	17

Table 7 - Configuration Assembly (Continued)

Byte	Parameter	Parameter Instance[1]
13	ASCII Transmit End Delimiter Mode	18
14	ASCII Transmit End Delimiter Character	19
15	ASCII Consume String Data Type	20
16	ASCII Transmit Swap Mode	21
17	Ethernet Record Header Mode	22

[1] See Table 8.

POINT I/O Parameters

Use the information in Table 8 to configure your ASCII module parameters.

Table 8 - Parameter List

Parameter	Parameter Instance	Access	Description	Parameter Choices	Default Setting	Default Value	Data Type
Serial Port Parameters							
Serial Character Framing	1	Get/Set	Defines the number of data bits, stop bits, and parity in data character frames	0 = 7N2 1 = 7E1 2 = 7O1 3 = 8N1 4 = 8N2 5 = 8E1 6 = 8O1 7 = 7E2 8 = 7O2	7N2	0	USINT
Serial Port Comm Speed	2	Get/Set	Defines the communication rate of the serial port	0 = 9600 1 = 1200 2 = 2400 3 = 4800 4 = 19.2k 5 = 38.4k	9600 baud	0	USINT
Serial Port Receive from ASCII Device							
Max Number of Receive Chars	3	Get/Set	Maximum number of characters the 1734-232ASC module expects to receive into its ASCII port from the serial device	1…128	20 chars	20	USINT
Receive Record Start Mode	4	Get/Set	Selects whether the start delimiter is included with the receive data	0 = No Start Delimiter 1 = Exclude Start Delimiter 2 = Include Start Delimiter	No Start Delimiter	0	USINT
Receive Start Delimiter	5	Get/Set	Character that identifies the beginning of the data string from the ASCII device when the length is specified as 0	Any valid standard ASCII character (0…127, 0…255) Dependent on number of bits chosen.	Colon	0x3A	USINT
Receive Record End Mode	6	Get/Set	Selects whether the End delimiter is included with the received data	0 = No End Delimiter 1 = Exclude End Delimiter 2 = Include End Delimiter	Include End Delimiter	2	USINT
Receive End Delimiter	7	Get/Set	Character that identifies the end of the data string from the ASCII device when the length is specified as 0	Any valid standard ASCII character (0…127, 0…255)	Carriage return	D_{hex}	USINT
Send (Produce) over Ethernet to Master							
Receive String Data Type	8	Get/Set	Defines the format of the data string to the master	0 = Array 1 = Short_String 2 = String	Short_String	1	USINT
Pad Mode	9	Get/Set	Indicates whether to pad the invalid data region after the delimiter with the pad character, or to use variable length I/O responses	0 = Pad Mode Disabled 1 = Pad Mode Enabled	Enabled	1	USINT
Pad Character	10	Get/Set	Value to use to pad the invalid data portion of the poll response	Any valid standard ASCII character (0…127, 0…255)	NULL	0	USINT
Receive Swap Mode	11	Get/Set	If enabled, the position of the bytes in the serial messages is swapped every 2, 3, or 4	0 = Disabled 1 = 16 bit Swap Enabled 2 = 24 bit Swap Enabled 3 = 32 bit Swap Enabled	Disabled	0	USINT

Table 8 - Parameter List (Continued)

Parameter	Parameter Instance	Access	Description	Parameter Choices	Default Setting	Default Value	Data Type
Master/Slave Handshake Mode	12	Get/Set	If enabled, master must acknowledge it is ready for next new data before 1734-232ASC module sends the new data.	0 = Master/Slave Handshake 1 = Produce Immediate	Produce Immediate	1	USINT
Produce Assembly Size	13	Get	Total number of bytes of I/O data that are sent to the master from the 1734-232ASC module. This should be the RX size of your scanner.	5…132	20 bytes of array data and 4 header bytes	24	USINT
Serial Data	14	Get	Serial data in the receive buffer	Any data string 0…128 bytes long	Empty	Empty	SHORT_STRING
Receive Data Size	15	Get	Number of characters in the receive data buffer	0…128	0	0	USINT
Receive Record Number	16	Get/Set	The receive record number sent from the master	0…255	0	0	USINT
Serial Port Transmit to ASCII Device							
Max Number of Transmit Chars	17	Get/Set	Maximum number of characters the 1734-232ASC module expects to transmit out its serial port to the serial device	1…128	20 chars	14_{hex}	USINT
Transmit End Delimiter Mode	18	Get/Set	Selects whether the End delimiter is included with the received data	0 = No End Delimiter 1 = Exclude End Delimiter 2 = Include End Delimiter	Include	2	USINT
Transmit End Delimiter Character	19	Get/Set	Character that identifies the end of the transmit data string from Ethernet to the ASCII device when the length is specified as 0	Any valid standard ASCII character (0…127 with 7-bit data, 0…255 with 8-bit data)	Carriage return	D_{hex}	USINT
Consume over Ethernet from Master							
Consume String Data Type	20	Get/Set	Defines the format of the data string that is received from the master	0 = Array 1 = Short_String 2 = String	Short_ String	1	USINT
Transmit Swap Mode	21	Get/Set	If enabled, the position of the bytes in the serial messages is swapped every 2 bytes or 4 bytes	0 = Disabled 1 = 16 bit Swap Enabled 2 = 24 bit Swap Enabled 3 = 32 bit Swap Enabled	Disabled	0	USINT
Transmit Handshake Mode	22	Get/Set	Selects the source of the event that triggers the transmission of data over the serial link	0 = Transmit Handshake 1 = Transmit Immediate	Transmit Handshake	0	USINT
Consume Assembly Size	23	Get	Total number of bytes of I/O data that are received from the master. This should be the TX size of your scanner.	5…132	20 bytes of array data and 4 header bytes	24	USINT
Serial Port Transmit/Explicit Messages from Configuration Tool							
Transmit Serial Data String	24	Get/Set	Serial data to be sent to the serial transmit buffer	ASCII Block Data 0…128 bytes long	Empty	Empty	SHORT_STRING
Transmitted Serial Data Length	25	Get/Set	Length of the transmit serial data	0…128	0	0	USINT
Transmit Record Number	26	Get/Set	The transmit record number of the current transmit data buffer	0…255	0	0	USINT
Status	27	Get	The combined status byte for the serial port object, the receive record object and the transmit record object.	1 = TX FIFO Overflow 2 = RX FIFO Overflow 4 = RX Parity Error 64 = Handshake Error 128 = New Data Flag	No Status	0	USINT

Receive Serial Data from the ASCII Device

The ASCII modules receive a number of characters and transmit them to the EtherNet/IP adapter via:

- I/O – poll, COS, or cyclic messages
- Explicit message

The received character string is captured when:

- The specific number of bytes defined (Receive Character Buffer Length) is received, or
- The defined End-of-String Terminator character is detected.

When either of these events occur, the ASCII modules store the received message string into its internal buffer and then transmits (produce) it onto Ethernet at the next appropriate opportunity.

Set Up the Receive Character Buffer Length

The Receive Character Buffer Length is the number of characters that the ASCII modules can receive from your I/O device into its buffer at one time.

If the module receives more characters than the Receive Character Buffer Length, it internally generates an overflow and forces the data into the ASCII module transmit buffer to be sent to the adapter. The subsequent received characters are then received into the buffer and are handled at the start of the next incoming message string. The overflow bit in the status byte is set as well.

IMPORTANT Incoming characters could be missed in the process of handling a string longer than the defined max length.

This value can be set and retrieved by using the standard set and get services on Class 15 (F_{hex}), Instance 3, Attribute 1.

Set Up and Use Pad Mode

Pad Mode operation adds extra characters to the end of its received data string (after the delimiter character). These characters are added by the module to the received characters from the external I/O device before sending to the controller.

The quantity that is added is such that the data string that is returned to the scanner is always a constant length, and that length is the number that is specified in the maximum Receive Character Length parameter plus the 4-byte header. The quantity of pad characters that are sent can vary from message to message, depending upon the size of the incoming string.

Pad Mode Selection

Pad Mode allows for compatibility with scanners that cannot receive variable length I/O messages. For such scanners, you must turn Pad mode ON (a value of 1). Turning Pad mode ON does not harm Scanners that do support variable length receive messages.

The default value for Pad Mode is ON. If your scanner does support variable I/O messaging lengths, you can turn OFF the Pad Mode option (a value of 0) to conserve some network bandwidth.

The selection of Pad Mode is valid only for the Ethernet message that the ASCII module produces. It has no effect on Ethernet messages that are sent from the scanner to the ASCII module. This value can be set and retrieved by using the standard set and get services on Class 15 (F_{hex}), Instance 9, Attribute 1.

Pad Mode Character

The ASCII modules allow you to specify the character that Pad Mode uses to pad the received serial data. This can be set to any valid I/O value (0…127 in 7-bit modes, 0…255 in 8-bit modes). This value can be set and retrieved by using the standard set and get services on Class 15 (F_{hex}), Instance 10, Attribute 1.

Set Up and Use the Swap Bytes Mode

This option is helpful if the ASCII module is connected to a controller that organizes the data string characters into data type elements that are larger than 1 byte each. In such cases, the bytes of the data in the master's memory organization can be reversed from the order in which they are sent or received in the controller's serial link to the ASCII device. This can cause errors in some cases. Thus, the received string "ABCDEFGH" may appear in memory as "BADCFEHG" for 2-byte word organization, and "DCBAHGFE" for 4-byte word organization.

Transmit Byte Swapping

By setting Parameter 21 (Class F_{hex}, Instance 21, Attribute 1), the ASCII module swaps the bytes from the master before transmitting the string to the ASCII device.

To swap bytes from the master, set the Byte Swap mode from the Transmit Data section of the Advance Format Configuration Category. See Advance Format Configuration Category on page 20.

Receive Byte Swapping

By setting Parameter 11 (Class F_{hex}, Instance 11, Attribute 1), the ASCII module reorders the bytes received from your ASCII device before sending the string to the master.

To swap bytes received from the ASCII device, set the Byte Swap mode from the Receive Data section of the Advance Format Configuration Category. See Advance Format Configuration Category on page 20.

Maximum Number of Receive Characters Parameter Rules for Usage

- Swap Bytes Mode is set for transmit and receive independently.
- The byte swapping works best if the max RX and TX lengths are multiples of the byte-swap size.

Set Up and Use Delimiter Operation

When receiving data strings from your serial device, the ASCII module can take advantage of both Start and Stop (End) delimiters. The Start Delimiter is the start-of-string indicator and the End Delimiter is the end-of-string indicator.

When you select Start Delimiter operation, you define a character that prompts the ASCII module to start storing the incoming data string. All characters up to this Start Delimiter (after the previous message was completed) are ignored. Once the Start Delimiter is received, all characters are stored until either the End Delimiter is received or the Max_Receive_Char_Length is reached. Once the End Delimiter is reached, the data string is updated and the ASCII module ignores all subsequent data until the start delimiter is received again.

If either delimiter is used, you can also elect whether to include those characters in the data string. If the start delimiter mode is set to Disabled, then the ASCII module always adds data to the string.

Receive String Data Type

Receive String Data Type is the format of the data – Array, Short_String, or String. Which one you choose depends on your application, and modifies the format of the header field in the data string.

The Array data type does not have a length that is associated with it. It is equivalent to specifying a length of zero using a String or Short_String data type.

The Short_String data type is the default data type of the device. This suffices for most applications. The Short_String data type value has only 1 byte of length, and the rest of the data bytes are appended after the length.

The String data type value has 2 bytes of length. The String data type is useful in communicating to some controllers or other devices that have a data file made to handle this data type. The length is little-endian (low byte, high byte), and the high-order byte is always set to zero. The ASCII module receives up to 128 bytes of information, so the extra byte, although required for this data type, is always 0.

Table 9 - Receive Data Format — *Array* Data Type

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Receive Record Number	Status Byte	Reserved	Reserved	ASCII Data (max 128 bytes)	<CR> (Terminator)

Table 10 - Receive Data Format — *Short_String* Data Type

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Receive Record Number	Status Byte	Reserved	Length	ASCII Data (max 128 bytes)	<CR> (Terminator)

Table 11 - Receive Data Format — *String* Data Type

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Receive Record Number	Status Byte	Length (Low Byte)	Length (High Byte)	ASCII Data (max 128 bytes)	<CR> (Terminator)

Status Byte Description

The Serial Status, shown in Table 12, byte is an OR'd bit field of a number of status and exception flags.

Table 12 - Serial Status Byte

Bit	Exception
0	TX Buffer Overflow
1	RX Buffer Overflow
2	RX Parity Error
6	Handshake Error
7	New Data Flag

TX Buffer Overflow (Bit 0)

The transmit queue has overflowed resulting in a loss of data. The transmit I/O is full of data waiting to be transmitted. Some of the data added has been lost. When space becomes available in the TX, this bit is reset.

RX Buffer Overflow (Bit 1)

The receive queue has overflowed resulting in a loss of data. The receive is full of data waiting to be processed. The data has been lost. When space becomes available in the RX, this bit is reset.

RX Parity Error (Bit 2)

This bit is set if a parity error is detected on the received data.

Handshake Error (Bit 6)

This error occurs only in Master-Slave Handshake Mode. It indicates that the master has requested a new data record from the ASCII module, but the ASCII module has not indicated new data is available to be sent.

New Data (Bit 7)

This bit is used only when the Master-Slave Handshake option is active. When the ASCII module receives a new data string into its serial port, it sets this flag in its response message. The bit remains set until the master changes the Receive Record Number in its produced data.

Transmit Serial Data to the ASCII Device

To transmit data to your serial device, the data must first be sent to the ASCII module and then the ASCII module must send the data to the serial device.

The ASCII module transmits a number of characters from the controller to your serial device using:

- I/O messages
- Explicit messages

The character string is transmitted to your serial device when:

- The specific number of bytes defined (Transmit Character Buffer Length) is reached, or
- The defined End-of-String Terminator character is detected and active.

When either of these events occur, the ASCII module stores the string data into its internal buffer and then transmits it out its serial port.

Set Up the Transmit Character Buffer Length

The Transmit Character Buffer Length is the number of characters that the ASCII module can receive in its transmit buffer from the controller.

Set Up and Use the Transmit Delimiter

The Transmit Delimiter is an end-of-string character that determines how many bytes to transmit over the serial link to your ASCII device.

IMPORTANT This Transmit Delimiter is used only if the length is 0. If this length is not 0, the ASCII module ignores the transmit delimiter and sends the defined number of characters.

The ASCII module transmits up to and (optionally) including the delimiter when the above condition is met. The Transmit Delimiter can be set to any valid binary value. Be careful not to set the delimiter to a value outside of the valid range for your data bits. A data bit size setting of 7 only allows you a delimiter range of 0…127 dec., 00…7F$_{\text{hex}}$.

If the length is 0 and if you do not have a valid delimiter in your string, the module sends all data characters from the I/O field to your serial device.

Transmit String Data Assemblies

Transmit String Data Assemblies are the formats of the data you send from the ASCII module – Array, Short_String, or String – to the controller. Which one you choose depends on your application, and modifies the format of the data field.

Table 13 - Transmit Data Format — *Array* Data Type

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Next Transmit Record Number (Handshake Mode Only)	Transmit Record Number	Reserved	Reserved	ASCII Data (max 128 bytes)	End of String Delimiter or Last Data Byte

Table 14 - Transmit Data Format — *Short_String* Data Type

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Next Transmit Record Number (Handshake Mode Only)	Transmit Record Number	Reserved	Length	ASCII Data (max 128 bytes)	End of String Delimiter or Last Data Byte

Table 15 - Transmit Data Format — *String* Type

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Next Transmit Record Number (Handshake Mode Only)	Transmit Record Number	Length (LSB)	Length (MSB)	ASCII Data (max 128 bytes)	End of String Delimiter or Last Data Byte

Transmit Handshake or Transmit Immediate Mode

This option defines when the ASCII module transmits data out of the serial port. Transmit Handshake mode is used primarily in Polled I/O to help prevent the ASCII module from repeatedly sending its data buffer to the serial device each time an I/O command is received.

With **Transmit Handshake** active (default), the master changes the Transmit Record Number value in the message header whenever it wants the ASCII module to send a serial string to the target serial device. The module monitors the second byte (Transmit Record Number) of the Poll command or explicit message to determine whether the consumed data from the controller is to be transmitted out of the serial port.

The module does not transmit the command data until it receives a new Transmit Record Number value. Upon sensing a change in value of the Transmit Record Number, the module transmits the data. It initiates one serial transmission for each change in the Transmit Record Number value, regardless of the data string value.

In **Transmit Immediate** mode, the ASCII module ignores the Transmit Record Number value and initiates a new serial port transmission with each message that is received from the controller. This technique can be useful when the target ASCII device is not affected by receiving the same data multiple times and operates quickly enough so that message overflows do not occur. This option is most useful with explicit messages.

A transmission of the serial data from the ASCII module to your ASCII device is initiated in two ways. If an I/O connection is used, serial data is sent from the controller to the ASCII module based on the RPI rate set.

- In Transmit Handshake Mode, changing the transmit record number always initiates a transmission on the network. Even if you do not set new data into the data string, the old data is transmitted.
- In Transmit Immediate Mode, the ASCII module transmits data out of its serial port every time it receives an I/O command or explicit message to its transmit buffer or changes the transmit record number.

⚠ **ATTENTION:** An I/O message is repeatedly sent and generates internal writes to Class 71, Instance 1, Attribute 3. This causes you to send large amounts of data to your device and can cause TX Buffer Overflows. You may experience corrupted data and improper information that is sent to your serial device when using data only mode with I/O messaging.

⚠ **ATTENTION:** You should take care when activating Transmit Immediate mode while I/O Polling is active. In this mode of operation, the module continuously transmits serial data to the target serial device. If the data string represents a command message to your target device, these actions could result in unintended mis-operation of your device or system.

Transmit Serial Data

The length of the string that is sent determines the use of a delimiter in transmitting data to a serial device from the ASCII module.

- If the string length is zero (as dynamically defined by the third and fourth bytes in the message header), or if the data type is Array:

  The ASCII module receives data that is sent from the controller and uses the delimiter to determine how much data is sent to the serial device. The ASCII module computes the length and then stores this as the new length in the string attribute. This does not show up if the data type is Array, you only see the string that is truncated, and the length is in the background.
- If the string length > 0 or the data type is String or Short_String:

  The ASCII module receives data that is sent from the controller ignoring any embedded terminator. It stores the number of characters that are defined in Max_Number_of_Transmit_Chars, or the total that is sent by the master, whichever is less.

The ASCII module sends the information immediately in Transmit Immediate mode. You can always transmit the data by changing the Transmit Record Number.

Produce Immediate or Master/Slave Handshake Mode

Selecting the mode option defines when the ASCII module sends new data to the controller.

In **Produce Immediate** mode (default), there is no gating by the ASCII module of data that is sent from the ASCII module to the controller. The module sends its most current serial port data to the master in response to each Poll command or explicit message, or in response to a COS or Cyclic event. It is the master’s responsibility to be ready to accept and process new data strings as they are received.

In **Master/Slave Handshake** mode, the ASCII module refrains from updating the new ASCII data until the master requests it. This technique is useful when the master must ensure that some specific actions take place before receiving the new serial data.

IMPORTANT Master/Slave Handshake mode is only available if the Transmit Immediate mode is also active.

When Master/Slave Handshake mode is active, two data fields in the message headers are used to initiate and complete the transaction:

- The **New Data Available Flag** is set in the Serial Status byte of the ASCII Poll response message.

  This informs the master that a new data string is received and the module is waiting for the OK from the master to send it.
- The next **Transmit Record Number** is updated by the master to the ASCII module that it is now ready to receive the new data string.

Table 16 - ASCII Modules Consume Assembly Message with Handshake Mode (Short_String)

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5-X	Byte X+1 (Max = 132)
Next Transmit Record Number (Handshake Mode Only)	Transmit Record Number	Reserved	Length	ASCII Data (max 127 bytes)	End of String Delimiter or Last Data Byte

The master monitors the new data flag and when the master is ready to receive new serial data, it sets a new number in the new Receive Record Number of the I/O message. This applies only to data being sent from the ASCII module to the master.

The operation proceeds as follows:

- The ASCII module receives a new data string.
- The ASCII module sets the New Data Flag in the Status byte of its next produce message.
- When the master is ready to receive the new data string, it changes the next Transmit Record Number to any value different than what it had been sending.
- The ASCII module sends the new data when the next Transmit Record Number changes. The ASCII module resets the new data available bit.
- If the next Transmit Record Number changes, and the module has no new data, it sets the Handshake Error bit in its Produce Status byte.

You can set the next **Transmit Record Number** to any value. It may reflect the Receive Record Number that is sent from the module, if desired, or any other meaningful number. The ASCII module only looks for a change from the value since it set the New Data Flag.

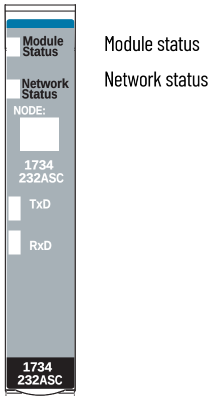
Chapter 4

Diagnostics

Interpret Status Indicators

Use the status indicators to help you troubleshoot any problems with your ASCII modules.

RS-232



RS-485

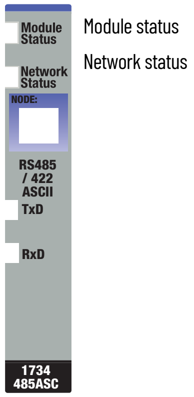


Table 17 - Module Status Indicators

Indication	Probable Cause
Module Status	
Off	No power applied to device.
Green	Device operating normally.
Flashing green	Device needs commissioning due to configuration missing, incomplete, or incorrect.
Flashing red	Recoverable fault.
Red	Unrecoverable fault may require device replacement.
Flashing red/green	Device is in self-test.

Table 18 - Network Status Indicators

Indication	Probable Cause
Network Status	
Off	Device is not online. - Device has not completed dup_MAC_id test. - Device not powered - check module status indicator.
Flashing green	Device is online but is not allocated to a master.
Green	Device online and has connections in the established state.
Flashing red	One or more I/O connections in timed-out state.
Red	Critical link failure - failed communication device. Device detected error that helps prevent it communicating on the network.

Table 19 - Transmit/Receive Indicators

Indication	Probable Cause
Transmit/Receive Status	
Flashing transmit/off receive	Check wiring, ground, and RX connection. User parameter object to view record numbers.
Flashing receive/off transmit	Check wiring. Watch TX light. If it does not flash, check that you are properly initiating transmission. Use EDS parameter object to try transmitting and watch the light. If it flashes, you are not properly initiating transmission via I/O messaging. If it does flash, check the remote device.
Off transmit/green receive	Check connections as you may have wired the device backwards.

ASCII Module with ControlLogix System Application Examples

About These Examples

For these application examples, the POINT I/O ASCII modules are used directly with a CompactLogix 5370 controller over an Ethernet network. The ASCII modules act as a normal slave to the master scanner. The data that is sent to and from the scanner contains the ASCII data and some specialty bytes.

Serial Port Connector

The ASCII devices are connected to the ASCII modules via a:

- 3-wire communications cable for the RS-232 network.
- 2-wire communications cable for the RS-485 network.
- 4-wire communications cable for the RS-422 network.

Table 20 - Module Terminations

Terminal	RS-232	RS-485	RS-422
0	Tx[1]	S+[1]	Tx+[1]
1	Rx[2]	NC	Rx+[1]
2	NC	S-[2]	Tx-[2]
3	NC	NC	Rx-[2]
4	NC	NC	NC
5	NC	NC	NC
6	NC	CG	CG
7	SG	SG	SC

[1] S+ and Tx+ are transmit from the module
[2] S- and Rx- are receive into the module

Send Data Through the ASCII Modules

To send data from the master scanner through the ASCII modules, the desired data should be written into the data area, starting at byte 5 after the header. Next, the number of bytes to be transmitted should be put into byte 4 of the header. Lastly, the Transmit Record Number byte 2 must be changed from its previous value to trigger the transmission (for instance, by incrementing the ID value by 1 each time a message is to be sent).

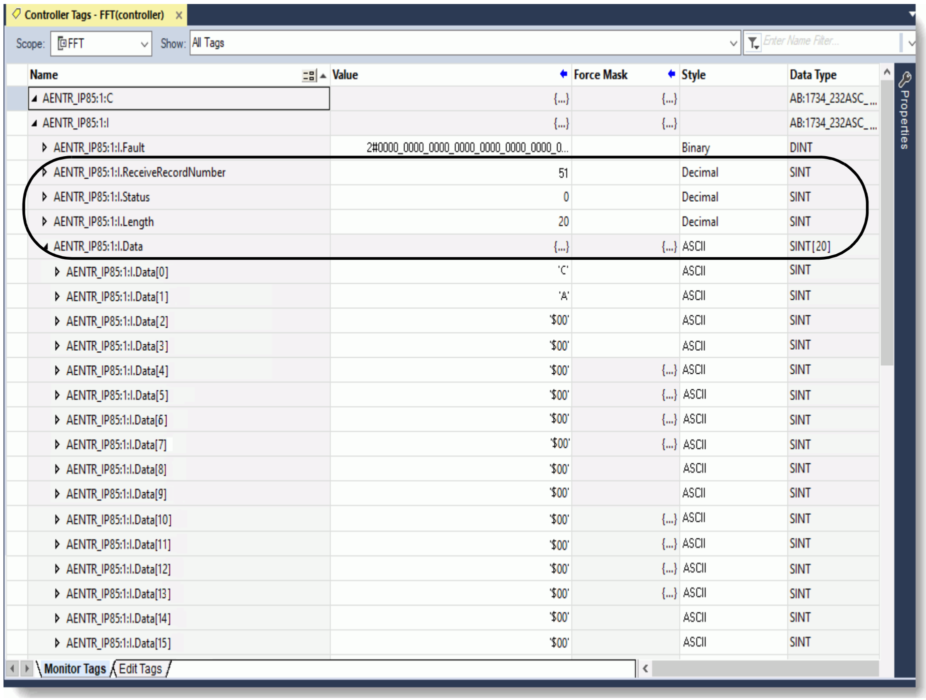
Receive Data From the ASCII Modules

To receive data from the master scanner through the ASCII modules, look for the Receive Record Number in byte 1 to change in value. When this happens, a new data buffer is available for storage. Read the length of the data from byte 4 and copy that number of bytes from the data buffer starting at byte 5 into a file.

Read Data from the ASCII Modules into ControlLogix

For this example, Controller Tags Edit Tags function is used to separate the header bytes into their own data locations. Figure 1 illustrates that the receive record number is in AENTR_IP85:1:I.ReceiveRecordNumber and the length is in AENTR_IP85:1:I.Length. The ASCII data starts in AENTR_IP85:1:I.Data[0].

Figure 1 - Receive Record Number Location in Controller Tags Dialog

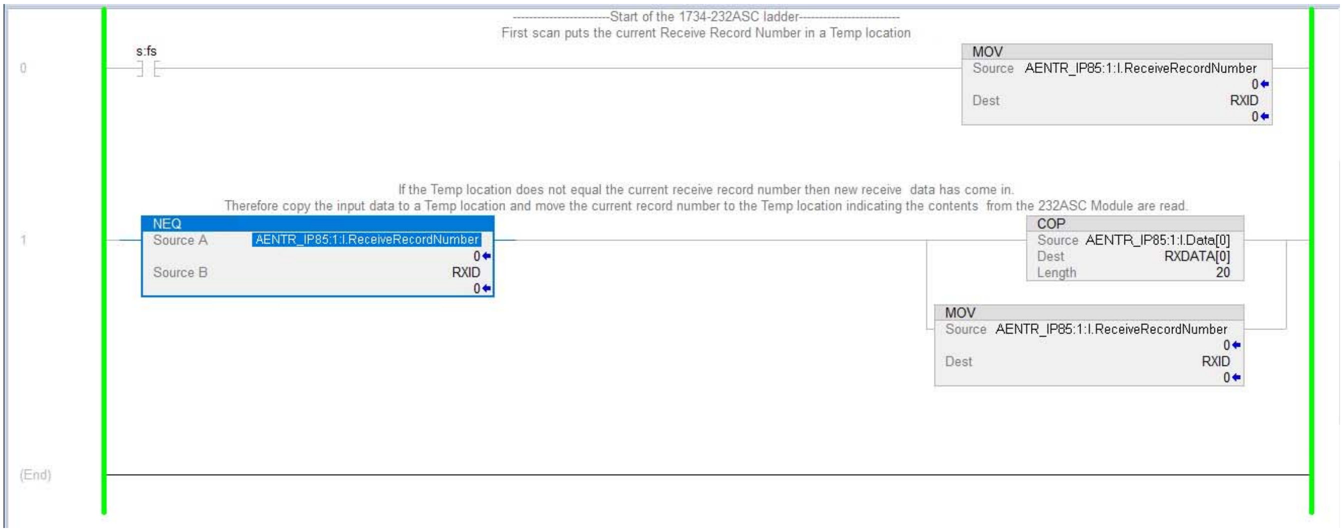


Scope: FFT Show: All Tags

Name	Value	Force Mask	Style	Data Type
AENTR_IP85:1:C	{...}	{...}		AB:1734_232ASC_...
AENTR_IP85:1:I	{...}	{...}		AB:1734_232ASC_...
AENTR_IP85:1:I.Fault	2#0000_0000_0000_0000_0000_0000_0000_0...		Binary	DINT
AENTR_IP85:1:I.ReceiveRecordNumber	51		Decimal	SINT
AENTR_IP85:1:I.Status	0		Decimal	SINT
AENTR_IP85:1:I.Length	20		Decimal	SINT
AENTR_IP85:1:I.Data	{...}	{...}	ASCII	SINT[20]
AENTR_IP85:1:I.Data[0]	'C'		ASCII	SINT
AENTR_IP85:1:I.Data[1]	'A'		ASCII	SINT
AENTR_IP85:1:I.Data[2]	'$00'		ASCII	SINT
AENTR_IP85:1:I.Data[3]	'$00'		ASCII	SINT
AENTR_IP85:1:I.Data[4]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[5]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[6]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[7]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[8]	'$00'		ASCII	SINT
AENTR_IP85:1:I.Data[9]	'$00'		ASCII	SINT
AENTR_IP85:1:I.Data[10]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[11]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[12]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[13]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:I.Data[14]	'$00'		ASCII	SINT
AENTR_IP85:1:I.Data[15]	'$00'		ASCII	SINT

In Figure 2 on page 39 the ControlLogix ladder logic checks to see if the receive record number in AENTR_IP85:1:I.ReceiveRecordNumber has changed from a previously stored value. If it has changed, the data starting in AENTR_IP85:1:I.Data[0] is copied to a file and the current receive record number is stored into a data location.

Figure 2 - Receive Record Ladder Logic Example



------------------------Start of the 1734-232ASC ladder------------------------
First scan puts the current Receive Record Number in a Temp location

If the Temp location does not equal the current receive record number then new receive data has come in.
Therefore copy the input data to a Temp location and move the current record number to the Temp location indicating the contents from the 232ASC Module are read.

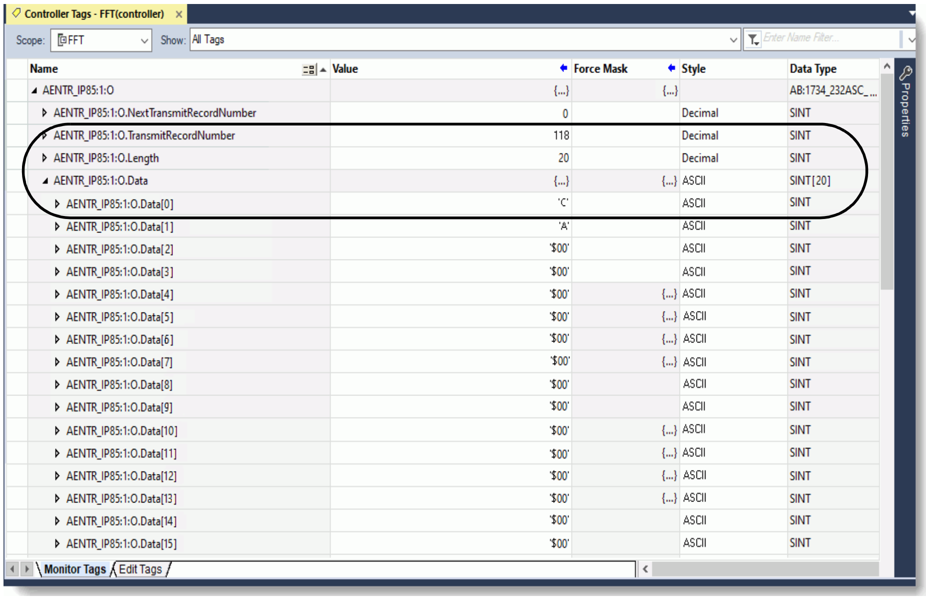
0
s:fs
MOV
Source AENTR_IP85:1:I.ReceiveRecordNumber
0
Dest RXID
0
1
NEQ
Source A AENTR_IP85:1:I.ReceiveRecordNumber
0
Source B RXID
0
COP
Source AENTR_IP85:1:I.Data[0]
Dest RXDATA[0]
Length 20
MOV
Source AENTR_IP85:1:I.ReceiveRecordNumber
0
Dest RXID
0
(End)

If data is received frequently, we recommend storing the receive data into some kind of large rotary buffer so as not to overwrite an older packet before it is decoded.

Transmit Data from ControlLogix Through the ASCII Modules

For this example, Controller Tags Edit Tags function is used to separate the header bytes into their own data locations. Figure 3 illustrates that the transmit record number is in AENTR_IP85:1:O.TransmitRecordNumber and the length is in AENTR_IP85:1:O.Length. The ASCII data starts in AENTR_IP85:1:O.Data[0].

Figure 3 - Transmit Record Number Location in Controller Tags Dialog



Controller Tags - FFT(controller)

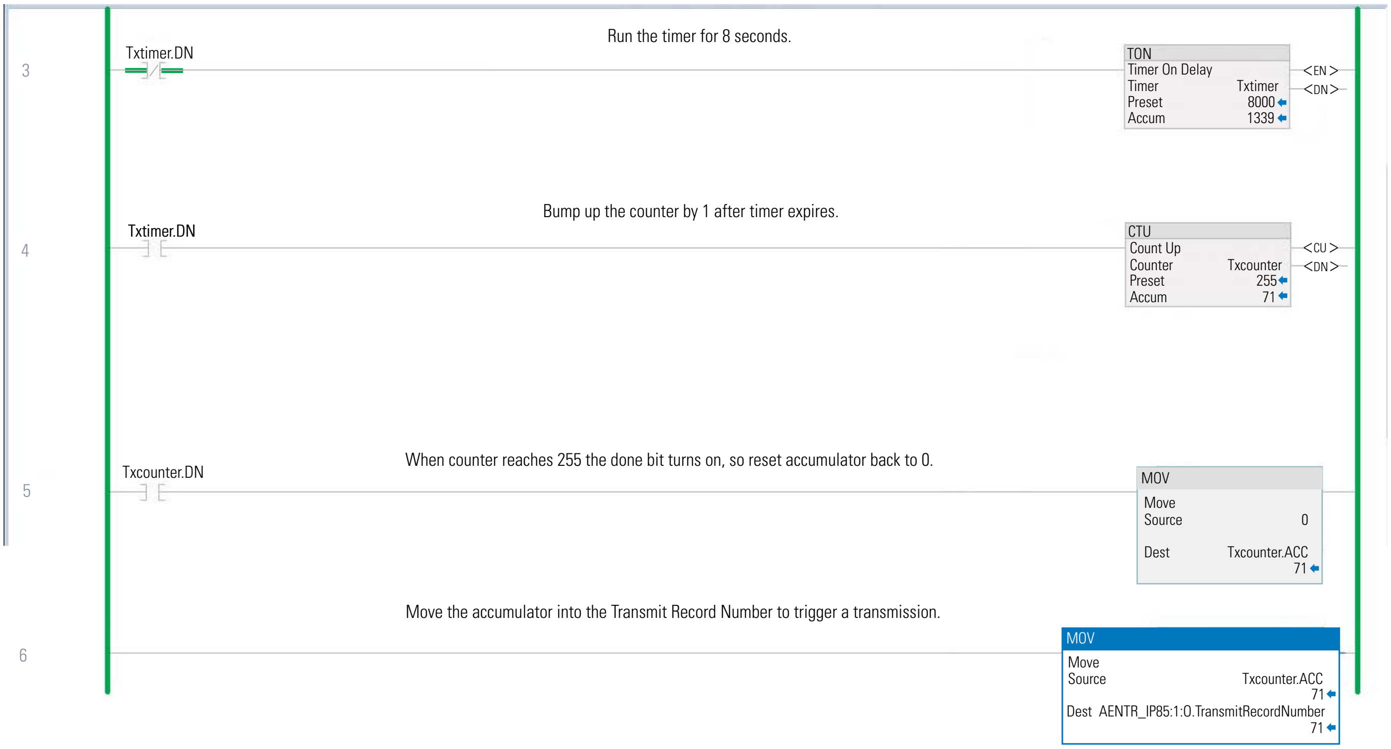
Scope: FFT Show: All Tags

Name	Value	Force Mask	Style	Data Type
AENTR_IP85:1:O	{...}	{...}		AB:1734_232ASC_...
AENTR_IP85:1:O.NextTransmitRecordNumber	0		Decimal	SINT
AENTR_IP85:1:O.TransmitRecordNumber	118		Decimal	SINT
AENTR_IP85:1:O.Length	20		Decimal	SINT
AENTR_IP85:1:O.Data	{...}	{...}	ASCII	SINT[20]
AENTR_IP85:1:O.Data[0]	'C'		ASCII	SINT
AENTR_IP85:1:O.Data[1]	'A'		ASCII	SINT
AENTR_IP85:1:O.Data[2]	'$00'		ASCII	SINT
AENTR_IP85:1:O.Data[3]	'$00'		ASCII	SINT
AENTR_IP85:1:O.Data[4]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[5]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[6]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[7]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[8]	'$00'		ASCII	SINT
AENTR_IP85:1:O.Data[9]	'$00'		ASCII	SINT
AENTR_IP85:1:O.Data[10]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[11]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[12]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[13]	'$00'	{...}	ASCII	SINT
AENTR_IP85:1:O.Data[14]	'$00'		ASCII	SINT
AENTR_IP85:1:O.Data[15]	'$00'		ASCII	SINT

The ladder logic in Figure 4 on page 40 increments a counter every 8 seconds and then copies the accumulator value of the counter into the Transmit Record

Number location AENTR_IP85:1:O.TransmitRecordNumber. When the counter reaches 255, it is automatically wrapped around to 0.

Figure 4 - Transmit Record Ladder Logic Example



3 Run the timer for 8 seconds.

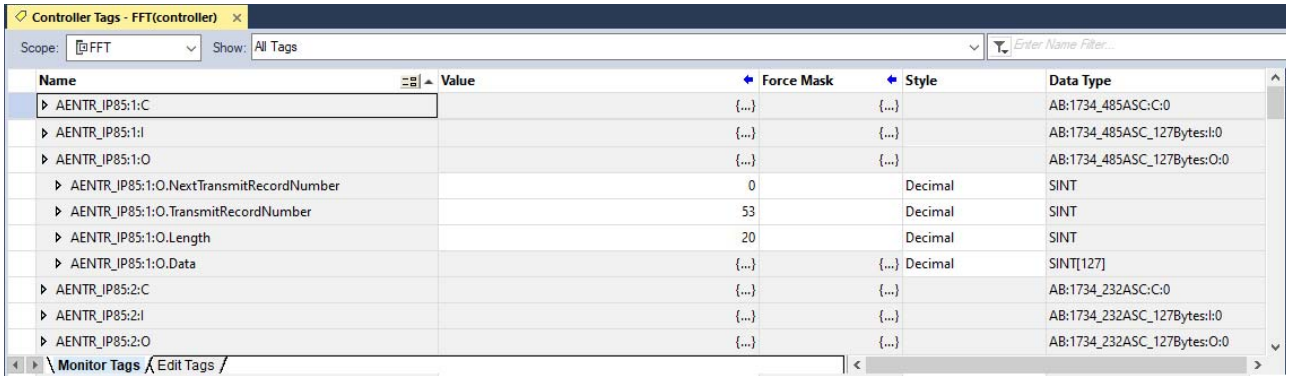
4 Bump up the counter by 1 after timer expires.

5 When counter reaches 255 the done bit turns on, so reset accumulator back to 0.

6 Move the accumulator into the Transmit Record Number to trigger a transmission.

In the Controller Tags output table shown in Figure 5, the data being transmitted is already inside the data table starting at AENTR_IP85:1:O.Data and a constant length of 20 is in AENTR_IP85:1:O.Length. Under normal conditions, the transmit data and length is loaded by the ladder program before incrementing the Transmit Record Number.

Figure 5 - Controller Tags Output Table



Name	Value	Force Mask	Style	Data Type
AENTR_IP85:1:C	{...}	{...}		AB:1734_485ASC:C:0
AENTR_IP85:1:I	{...}	{...}		AB:1734_485ASC_127Bytes:I:0
AENTR_IP85:1:O	{...}	{...}		AB:1734_485ASC_127Bytes:O:0
AENTR_IP85:1:O.NextTransmitRecordNumber	0		Decimal	SINT
AENTR_IP85:1:O.TransmitRecordNumber	53		Decimal	SINT
AENTR_IP85:1:O.Length	20		Decimal	SINT
AENTR_IP85:1:O.Data	{...}	{...}	Decimal	SINT[127]
AENTR_IP85:2:C	{...}	{...}		AB:1734_232ASC:C:0
AENTR_IP85:2:I	{...}	{...}		AB:1734_232ASC_127Bytes:I:0
AENTR_IP85:2:O	{...}	{...}		AB:1734_232ASC_127Bytes:O:0

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Download firmware, associated files (such as AOP, EDS, and DTM), and access product release notes.	rok.auto/pcdc

Documentation Feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752, İçerenkÖy, İstanbul, Tel: +90 (216) 5698400 EEE YÖnetmeliğine Uygundur

Connect with us.

rockwellautomation.com — expanding **human possibility**®

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846