User Manual

Allen-Bradley

RFID Systems

Bulletin Number 56RF

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.

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: WARNING: ??????????ARNING: ?????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.
	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.
IMPORTANT	Identifies information that is critical for successful application and understanding of the product.

RSLogix 5000 Controller Tags

Commands Summary

RSLogix 5000 Code Examples

SLC Code Examples

MicroLogix 1400 Code Examples

RFID Tag Speed

Notes:

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
- conventions used in this manual

Who Should Use this Manual

Use this manual if you are responsible for designing, installing, programming, or troubleshooting control systems that use the 56RF RFID products.

You should have a basic understanding of electrical circuitry and familiarity with relay logic. If you do not, obtain the proper training before using this product.

Purpose of this Manual

This quick start guide assumes you have some familiarity with RSLogix software. It provides an example of the steps needed to get a 56RF RFID system set up and functioning. The reader should refer to appropriate user manuals for other details. This manual:

- explains how to install and wire an example RFID system
- install and setup the module in an RSLogix 5000 program
- set up a simple program to receive and transmit data to an RFID tag

Abbreviations

AFI – Application Family Identifier

AOP – Add On Profile

DFSID – Data Storage Format Identifier

DHCP – Dynamic Host Configuration Protocol

DNS – Domain Name Server

DOS – Disk Operating System

EAS – Electronic Article Surveillance

FE – Functional Earth

IEC – International Electrotechnical Commission

INT – signed, two byte integer

ISO – International Organization for Standardization

JTC – Joint Technical Committee

MACID – Media Access Control Identification

QD – Quick Disconnect

RFID – Radio Frequency Identification

SB – Sub-committee

SINT – signed, single byte integer

UID – Unique Identifier

UUID – Universally Unique Identifier

Additional Resources

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

Resource	Description
EtherNet/IP Modules in Logix5000 Control Systems User Manual, publication ENET-UM001	A manual on how to use EtherNet/IP modules with Logix5000 controllers and communicate with various devices on the EtherNet network.
Getting Results with RSLogix 5000, publication 9399-RLD300GR	Information on how to install and navigate RSLogix 5000. The guide includes troubleshooting information and tips on how to use RSLogix 5000 effectively.
Allen-Bradley Industrial Automation Glossary, AG-7.1	A glossary of industrial automation terms and abbreviations.
EtherNet/IP Embedded Switch Technology Application Guide, publication ENET-AP005	A manual on how to install, configure, and maintain linear and Device-level Ring (DLR) networks using Rockwell Automation EtherNet/IP devices with embedded switch technology.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation industrial system.
Product Certifications website, http://www.ab.com	Provides declarations of conformity, certificates, and other certification details.

You can view or download publications at http:/www.rockwellautomation.com/literature/. To order paper copies of technical documentation, contact your local Allen-Bradley distributor or Rockwell Automation sales office.

Introduction

What is RFID?

RFID stands for Radio Frequency Identification. It is a method for communicating information from one point to another point by the use of electromagnetic waves (i.e., radio waves). It has unique characteristics that make it attractive for use in industrial systems.

For example, you have a shipping carton that must be loaded with various goods to meet a customer's specific purchase order. You can attach a tag to the carton. Prior to attaching the tag, you fill the tag with the specific items that the customer wants. Then, as the carton moves to the filling stations, each station places the required objects, only if needed, into the carton. If the tag does not require something, the station is skipped.

Each filling station has an RFID transceiver. The transceiver reads and writes to the tag. When the tag approaches the RFID transceiver, the transceiver reads the contents of the tag. Based on the information received, the packaging process adds items (or skips this step) and then writes to the tag that the item(s) was added. The carton moves to the next filling station.

This is a common use of RFID technology. What makes the Bulletin 56RF product line unique is its conformance to the open international standards: ISO15693 and ISO18000-3 M1.

International Standard Compliance

ISO/IEC 15693 is an ISO standard for what are called vicinity tags. The tags, commonly referred to as ICODE tags, can be read from a greater distance than proximity tags and closed couple tags. ISO/IEC 15693 systems operate at the 13.56 MHz frequency, and offer maximum read distance of 3.3…4.9 ft (1…1.5 m), depending on the transceiver. Library applications with very large antennas are capable of these distances. Most industrial applications are less than 8 in. (20.3 cm) for a read/write range.

The ICODE compatible tags permit users to use lower cost tags than proprietary systems currently provide. Users have the ability to use tag configuration options from multiple vendors.

ISO/IEC 15693 forms part of a series of International Standards that specify non-contact tags. The tags can be attached to objects, like cartons, bags, and valuable items, which can then be tracked while in the vicinity of a reading device. ISO/IEC 15693-2:2006 defines the power and communications interface between the vicinity card and the reading device. Other parts of ISO/IEC 15693 define the physical dimensions of the card and the commands interpreted by the card and reader.

Power is coupled to the tag by an AC field produced in the transceiver. The powering field has a frequency of 13.56 MHz and is one of the industrial, scientific and medical (ISM) frequencies available for worldwide use. When sufficient power is received by the tag, it is able to respond to commands sent from the coupler. The coupler sends commands to the card by modulating the powering field and by using a modulation system known as pulse position modulation, whereby the position of a single pulse relative to a known reference point codes the value of a nibble or byte of data. This allows the card to draw the maximum energy from the field almost continuously. Tags, which have no power source, can be energized at ranges of up to 3.3 ft (1 m) from a coupler that can only transmit power within the limits permitted by international radio frequency (RF) regulations.

A tag only responds when it receives a valid command that selects a single tag from a possible collection of cards within range of the coupler. This process of collision detection and selection, also known as anti-collision, is made possible by detecting the unique identification number encoded into every tag. Anti-collision, and the commands used, are defined in ISO/IEC 15693-3. The tag responds to the transceiver by drawing more or less power from the field and generates one or two sub-carriers of around 450 kHz. These are switched on and off to provide special-encoded data that are then detected by the transceiver.

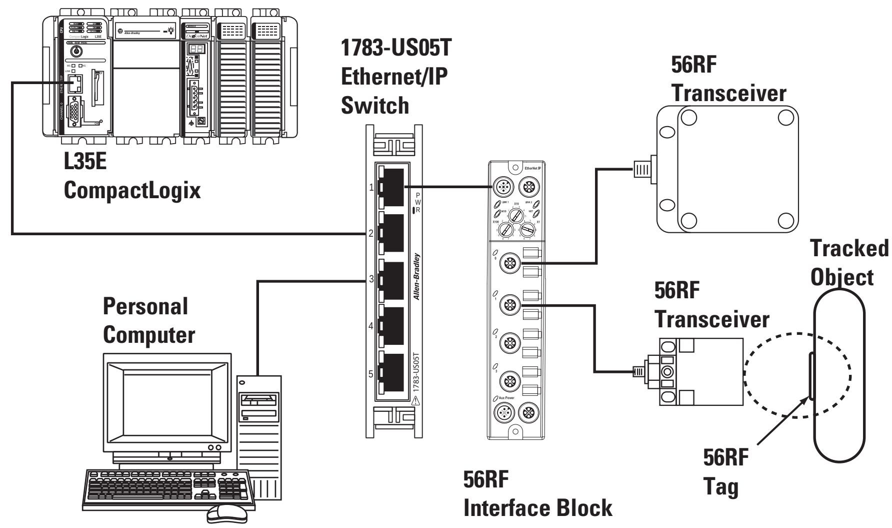
Backward Compatibility

The Bulletin 56RF RFID system will be offered initially on EtherNet/IP and is backward compatible with the previous offering of Bul. 56RF ICODE products. The transceivers and interface blocks are a matched pair so they cannot be interchanged. However, the tags can be interchanged with either system if they are ICODE tags. Both systems can read and write these tags seamlessly.

System Setup

The figure below shows a simple RFID system. This user manual describes the setup, installation, and programming required to get this system running.

Figure 1 - RFID System



Tags are attached to objects that need to be tracked. The tags hold important information about the object. An RF transceiver reads and/or writes information to the tags when the tag moves within the transmission envelope of the transceiver (dotted ellipse). The physical size of the transceiver is directly related to the size of the transmission field. The larger the transceiver, the longer and wider the antenna field is. Please see transceiver instruction sheets for antenna field sizes.

The transceivers are connected to a special RFID EIP interface block. The distribution block has an EtherNet connection to an EtherNet switch. An L35E CompactLogix controller and a personal computer also have EtherNet connections to the EtherNet switch.

Notes:

RFID Components

This chapter covers the three key components that make up the RFID system:

- the interface block
- the transceiver
- the tags

Interface Block

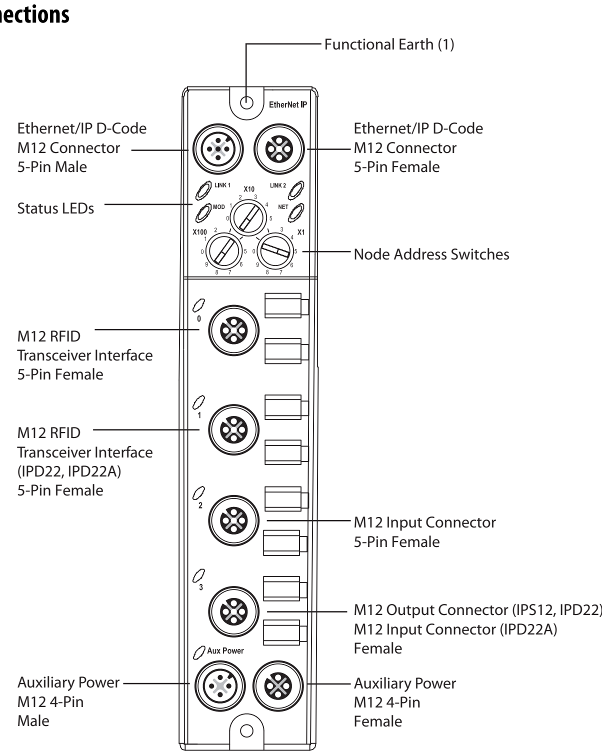
Three different interface blocks are available from which to choose. The table below shows the type of ports for each catalog number.

Table 1 - Type of Ports

Transceiver Ports	Input Ports	Output Ports	Cat. No.
1	1	1	56RF-IN-IPS12
2	1	1	56RF-IN-IPD22
2	2	0	56RF-IN-IPD22A

The figure below identifies the connections for the EtherNet/IP, RF transceivers, input devices, output devices, and power.

Figure 2 - Connections



LED Indicators

When the indicator is flashing, all flashes are 0.25 s ON and 0.25 s OFF.

This block has the seven different indicators.

Figure 3 - LED Indicators

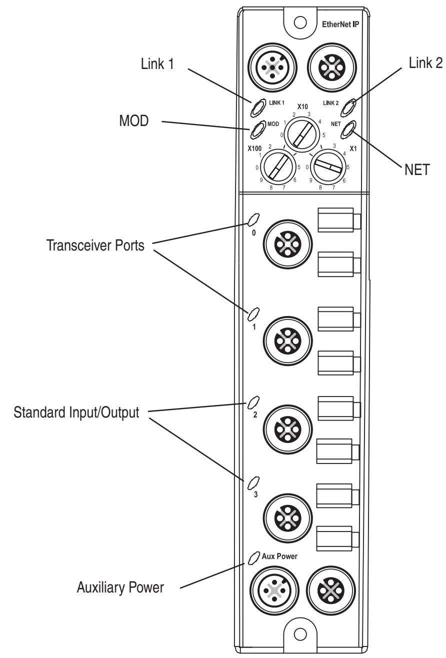


Table 2 - LED Indicators

LED Name	LED State	Indicates
Link1 and Link2	Off	No link
	Green	100 Mbps
	Flashing green	100 Mbps/active
	Yellow	10 Mbps
	Flashing yellow	10 Mbps/active
MOD (Module)	Off	There is no power applied to the block.
	Flashing red/ green	Device in self-test
	Green	The block is operating in a normal condition.
	Flashing green	Standby. The device is not communicating with the interface block. Normal state when only power has been applied to the transceiver.
	Flashing red	Recoverable fault. Most often occurs when data is corrupted between interface block and transceiver. CRC failures etc. Recommended solution is to remove electrical noise near cabling or reduce baud rate between transceiver and interface block.
	Red	The transceiver has an unrecoverable fault; may need replacing.

LED Name	LED State	Indicates
NET (Network)	Off	There is no power or no IP address.
	Flashing red/ green	Device in self-test
	Green	The block is operating in a normal condition.
	Flashing green	Standby. The device is not communicating with the interface block. Normal state when only power has been applied to the transceiver.
	Flashing red	Connection timeout. Most often occurs when data is corrupted between interface block and transceiver. CRC failures etc. Recommended solution is to remove electrical noise near cabling or reduce baud rate between transceiver and interface block.
	Red	Duplicate IP address. The transceiver has an unrecoverable fault; may need replacing.
Standard I/O	Off	Outputs inactive Inputs inactive
	Yellow	Outputs active Inputs active
	Flashing green	Outputs are idled and not faulted.
	Flashing red	Output faulted Inputs faulted
	Red	Outputs forced off Inputs unrecoverable fault
Aux Power	Off	No power is applied.
	Solid green	The applied voltage is within specifications.
	Solid yellow	The input power is out of specification.
RFID Port	Off	No power
	Flashing green	No tag present, but communicating
	Green	Communicating
	Flashing red	No transceiver connected
	Amber	Tag present

Transceivers

LED Indicators

Figure 4 - Indicators

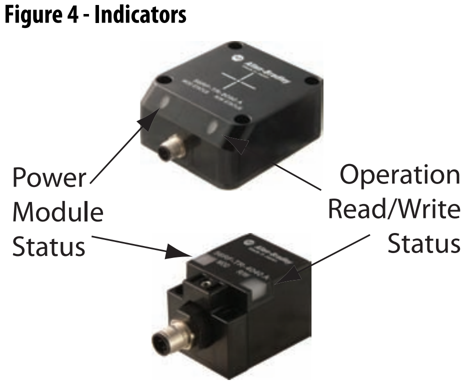


Table 3 - LEDs

LED Name	LED State	Indicates
Module Status	Off	There is no power applied to the block.
	Green	The block is operating in a normal condition.
	Red	The transceiver has an unrecoverable fault; may need replacing.
Read/Write Status	Off	There is no power applied to the device.
	Green	The EIP interface block is communicating with the transceiver, but no tag is present. No errors received.
	Amber	A tag is present within the antenna field.
	Red	A communication error has occurred. Examples are: bad read/write, corrupt CRC **Note:** If a read/write command is not completed while the tag is within the field, an error will occur.

Transceiver Power Up Sequence

1. Both LEDs OFF.
2. Power status turns green. R/W status turns green for 0.25 seconds.
3. R/W status turns red for 0.25 seconds.
4. R/W status turns off for 3…5 seconds.
5. R/W status turns amber for 0.5 seconds.
6. R/W status turns green.

RFID Tags

RF tags come in many shapes and sizes. In general, the bigger the tag, the longer the sensing distance from the transceiver. Table 4 summarizes the size of the memory for each type of tag.

Table 4 - Memory

Tag Type	Total Tag Memory	User Memory		
		No. of Bytes	No. of Blocks	Bytes per Block
SLI	128 B	112 B	28	4
SLI-S	256 B	160 B	40	4
SLI-L	64 B	32 B	8	4
FRAM	2048 B	2 kB	250	8

Tag Memory Structure

Universally Unique Identifier (UUID)

Each tag has a different 64-bit hexadecimal UUID that is programmed during the production process according to ISO/IEC 15693-3 and cannot be changed afterwards.

The numbering of the 64 bits is done according to ISO/IEC 15693-3 starting with the least significant bit (LSB) 1 and ending with the most significant bit (MSB) 64. This is in contrast to the general used bit numbering within a byte (starting with LSB 0).

Byte 5 (bit 41…48) is the tag type. Byte 6 (bit 49…56) is the manufacturer code, which coincides with the number of bytes/block.

The table below shows the RFID tags offered by Rockwell Automation and their structure.

Table 5 - Tag Structure

Byte		7	6	5	4	3	2	1	0
Name		UID 7	UID 6	UID 5	UID 4	UID 3	UID 2	UID 1	UID 0
Bit		64…57	56…49	48…41	40…1				
Value	SLI	E0	04	01	Unique Serial Number				
	SLI-S	E0	04	02	Unique Serial Number				
	SLI-L	E0	04	03	Unique Serial Number				
	FRAM	E0	08	01	Unique Serial Number				

Application Family Identifier (AFI)

The AFI represents the type of application targeted. AFI is coded on one byte, which constitutes two nibbles of 4 bits each. The most significant nibble of AFI is used to code one specific or all application families, as defined in the table below. The least significant nibble of AFI is used to code one specific or all application sub-families. Sub-family codes different from 0 are proprietary.

Table 6 - AFI Examples

AFI Most Significant Nibble	AFI Least Significant Nibble	Meaning	Examples/Notes
0	0	All families and subfamilies	No applicative preselection
X	0	All sub-families of family X	Wide applicative preselection
X	Y	Only the Yth sub-family of family X	—
0	Y	Proprietary sub-family Y only	—
1	0, Y	Transport	Mass transit, bus, airline
2	0, Y	Financial	IEP, banking, retail
3	0, Y	Indentification	Access control
4	0, Y	Telecommunication	Public telephony, GSM
5	0, Y	Medical	—
6	0, Y	Multimedia	Internet service
7	0, Y	Gaming	—
8	0, Y	Data storage	Portable files
9	0, Y	EAN-UCC (European Article Numbering-Uniform Code Council) system for application indentifiers	Managed by ISO/IECJTC 1/SC 31
A	0, Y	Data Identifiers as defined in ISO/ IEC 15418	Managed by ISO/IEC JTC 1/SC 31
B	0, Y	UPU	Managed by ISO/IEC JTC 1/SC 31
C	0, Y	IATA (International Air Transport Association)	Managed by ISO/IEC JTC 1
D	0, Y	Reserved for Future Use	Managed by ISO/IEC JTC 1/SC 17
E	0, Y	Reserved for Future Use	Managed by ISO/IEC JTC 1/SC 17
F	0, Y	Reserved for Future Use	Managed by ISO/IEC JTC 1/SC 17

X = ‘1’ to ‘F’, Y = ‘1’ to ‘F’

Data Storage Format Identifier (DSFID)

The DSFID indicates how data is structured in the tag memory. It may be programmed and locked by the respective commands. It is coded on one byte. It allows for instant knowledge on the logical organization of the data.

Electronic Article Surveillance (EAS)

EAS is a technology typically used to prevent shoplifting in retail establishments. An EAS detection system will detect active tags and set off an alarm.

EAS status is 1 bit data (LSB side), which is stored in the system area of a tag. The initial value is "1". EAS bit "1" means goods-monitoring status, and EAS bit "0" means that goods-monitoring status is cleared.

Smart Label Integrated Circuit (SLI)

SLI tags use an EEPROM (electrically erasable programmable read only memory) to store data. The 1024 bit EEPROM memory is divided into 32 blocks. Each block consists of 4 bytes (1 block = 32 bits). Bit 0 in each byte represents the least significant bit (LSB) and bit 7 the most significant bit (MSB), respectively.

Table 7 - SLI Tags

Block	Byte 0	Byte 1	Byte 2	Byte 3	Description
-4	UID0	UID1	UID2	UID3	Unique identifier (lower bytes)
-3	UID4	UID5	UID6	UID7	Unique identifier (higher bytes)
-2	Internally used	EAS	AFI	DSFID	EAS, AFI, DSFID
-1	00	00	00	00	Write access conditions
0					User Data
1					
2					
:					
:					
:					
22					
23					
27					

SLI

EAS Function

The LSB of Byte 1 in Block -2 holds the EAS bit (Electronic Article Surveillance mode active – the label responds to an EAS command)

Table 8 - EAS

Block -2, Byte 1							
MSB							**LSB**
X	X	X	X	X	X	X	e

EAS: e = 1 (EAS enabled) e = 0 (EAS disabled)

IMPORTANT	Changing of the EAS Configuration must be done in secure environment. The label must not be moved out of the communication field of the antenna during writing. We recommend to put the label close to the antenna and not to remove it during the operation.

Application Family Identifier

The ICODE system offers the feature to use an Application Family Identifier (AFI) at the inventory command and the two custom commands inventory read and fast inventory read (this allows for example the creation of label families).

This 8-bit value is located at Byte 2 in Block -2 as shown in the following figure and is only evaluated if the AFI flag is set in the reader command.

Table 9 - AFI

Block -2, Byte 2							
MSB							**LSB**
X	X	X	X	X	X	X	X

Data Storage Format Identifier

The Data Storage Format Identifier (DSFID) is located at Byte 3 in Block -2.

Table 10 - DSFID

Block -2, Byte 3							
MSB							**LSB**
X	X	X	X	X	X	X	X

Write Access Conditions

The Write Access Condition bits in block -1 determine the write access conditions for each of the 28 user blocks and the special data block. These bits can be set only to 1 with a lock command (and never be changed back to 0), i.e. already write protected blocks can never be written to from this moment on.

In block -2 each byte can be individually locked.

Table 11 - Write Access

Block -1																
Byte 0									**Byte 1**							
	MSB							**LSB**	**MSB**							**LSB**
Conditon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Write Access for Block Number	3	2	1	0	-2 (3)	-2 (2)	-2 (1)	-2 (0)	11	10	9	8	7	6	5	4

Block -1																
Byte 2									**Byte 3**							
	MSB							**LSB**	**MSB**							**LSB**
Conditon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Write Access for Block Number	19	18	17	16	15	14	13	12	27	26	25	24	23	22	21	20

IMPORTANT	Changing of the Write Access conditions must be done in secure environment. The label must not be moved out of the communication field of the antenna during writing. We recommend to put the label close to the antenna and not to remove it during operation.

Smart Label IC – Secure (SLI-S)

The 2048 bit EEPROM memory is divided into 64 blocks. A block is the smallest access unit. Each block consists of 4 bytes (1 block = 32 bits). Four blocks are summed up to one page for password protection. Bit 0 in each byte represents the least significant bit (LSB) and bit 7 the most significant bit (MSB), respectively.

The memory is divided into two parts:

- **Configuration Area:** This memory area stores all required information, such as UID, EPC data, write protection, access control information, passwords, etc. Direct access to this memory area is not possible.
- **User Memory:** This memory area stores user data. Direct read/write access to this part of the memory is possible depending on the related security and write protection conditions.

The table below shows the memory organization of an SLI-S tag.

Table 12 - SLI-S Memory Organization

Page	Block	Byte 0	Byte 1	Byte 2	Byte 3	Description
-6	-24					Configuration area for internal use
	-23					
	-22					
	-21					
:	:	:	:	:	:	
:	:	:	:	:	:	
:	:	:	:	:	:	
:	:	:	:	:	:	
-1	-4					
	-3					
	-2					
	-1					
0	0					User Memory 10 pages 4 blocks per page 4 bytes per block Total: 160 bytes
	1					
	2					
	3					
:	:	:	:	:	:	
:	:	:	:	:	:	
9	36					
	37					
	38					
	39					

Smart Label IC – Lean (SLI-L)

The SLI-L is used in applications that require smaller memory size. The 512 bit EEPROM memory is divided into 16 blocks. A block is the smallest access unit. Each block consists of 4 bytes (1 block = 32 bits). Four blocks are summed up to one page. Bit 0 in each byte represents the least significant bit (LSB) and bit 7 the most significant bit (MSB), respectively.

The memory is divided into two parts:

- **Configuration Area:** This memory area stores all required information, such as UID, write protection, passwords, etc. Direct access to this memory area is not possible.
- **User Memory:** This memory area stores user data. Direct read/write access to this part of the memory is possible depending on the related write protection conditions.

The table below shows the memory organization of an SLI-L tag.

Table 13 - SLI-L Memory Organization

Page	Block	Byte 0	Byte 1	Byte 2	Byte 3	Description
-2	-8					Configuration area for internal use
	-7					
	-6					
	-5					
-1	-4					
	-3					
	-2					
	-1					
0	0					User Memory 2 pages 4 blocks per page 4 bytes per block Total: 32 bytes
	1					
	2					
	3					
	4					
	5					
	6					
	7					

Ferroelectric Random Access Memory (FRAM)

FRAM is a non-volatile memory that uses ferroelectric film as a capacitor for storing data. FRAM offers high speed access, high endurance in write mode, low power consumption, non-volatility, and excellent tamper resistance. The FRAM tags have 2000 bytes for use as user area and 48 bytes for use as system area.

The FRAM tag memory areas consist of a total of 256 blocks (250 blocks of user area and 6 blocks of system area). Each block can store 64 bits (8 bytes) of data.

The block is the unit used for the writing and reading of FRAM data. The memory configuration of FRAM is shown below.

Table 14 - FRAM Memory Configuration

Area	Block No.	Details	Data Read	Data Write
User area (2000 bytes)	00_H to $F9_H$	User area	Yes	Yes
System area (48 bytes)	FA_H	UUID (64 bits)	Yes	No
	FB_H	AFI, DSFID, EAS, security status	Yes	Limited
	FC_H to FF_H	Block security status	Yes	No

Blocks 00_H...$F9_H$ are user area. The user area is defined as an area that can be accessed when the corresponding block address is specified. On the other hand, Blocks FA_H...FF_H are system area. The system area is defined as an area that can be accessed only with a specific command.

The system area consists of six blocks and contains UUID, AFI, DSFID, EAS bits, and security status (can write or cannot write) data for individual block. UID is fixed and cannot be updated. AFI, DSFID, and EAS bits are written at the factory, and can be updated and locked (disable to write) with commands (only EAS bit cannot be locked) .

As shown in above, FA_H holds the UUID, and FC_H...FF_H hold the security status information on individual user areas. The configuration of FB_H ...FF_H blocks is shown below. FB_H block is used for EAS status, AFI and DSFID data, the security status data of AFI and DSFID. Blocks FC_H...FF_H contain security status data.

Table 15 - Structure of FB_H

MSB											LSB
64	57	56	33	32	25	24	17	16	9	8	1
EAS Status		Reserved for future use		DSFID Lock Status		AFI Lock Status		DSFID		AFI	

Table 16 - Structure of FC_H to FF_H

	MSB									LSB
FC_H	3F	3E	3D	3C	3B	3A	39	03	02	01	00
FD_H	7F	7E	7D	7C	7B	7A	79	43	42	41	40
FE_H	BF	BE	BD	BC	BB	BA	B9	83	82	81	80
FF_H	Reserved for future use (6 bits)						F9	C3	C2	C1	C0

The security status of the user area is stored in the block security status bit in system area blocks of FC_H...FF_H per bit in each block. A user area is unlocked when the corresponding block security status bit is 0; it is locked (disable to write state) when the corresponding block security status bit is 1.

EAS bit is a single bit, and it is used for setting EAS status. It is possible to read/write data of two blocks at one time in the user area (if Read Multiple Blocks Unlimited command is used, up to 256 blocks can be accessed at one time) .

Handheld Reader/Writer

The RFID ICODE handheld interface provides a portable solution for reading/writing values to the tag data area. The handheld interface is a touch-screen operated computer with an attached RFID antenna and software that allows reading, writing, and saving tag RFID tag data. Each handheld interface comes with multiple connectivity methods, such as wireless, Bluetooth, and USB, which allow tag data to be transferred to/from a computer. The RFID ICODE handheld interface is IP65 rated for harsh industrial and outdoor environments. It is the ideal accessory for system setup, field service, fleet management, time and attendance, and any other application where transceiver mobility is required.

For more information on the RFID ICODE handheld interface, refer to Publication 57RF-UM001.

Figure 5 - Handheld Interface

Component Catalog Number Table

The following tables show the catalog numbers for the components in the Bul. 56RF product family.

EtherNet/IP Interface Blocks

Transceiver Ports	Input Ports	Output Ports	Cat. No.
1	1	1	56RF-IN-IPS12
2	1	1	56RF-IN-IPD22
2	2	0	56RF-IN-IPD22A

Transceivers

Dimensions [mm]	Recommended Sensing Distance [mm] ❶	Max. Sensing Distance [mm] ❶	Cat. No.
Rectangular (80x90)	100	168	56RF-TR-8090
Square (40x40)	50	85	56RF-TR-4040
Cylindrical M30	35	60	56RF-TR-M30
Cylindrical M18	18	30	56RF-TR-M18

❶ Range reference for a 50 mm diameter tag.

Tags

<table>
<tr><th>Outline</th><th>Type</th><th>Total Memory Size [B]</th><th>User Memory Size [B]</th><th>Dimensions [mm]</th><th>Cat. No.</th></tr>
<tr><td rowspan="6">Disc</td><td rowspan="4">SLI</td><td rowspan="4">128</td><td rowspan="4">112</td><td>16</td><td>56RF-TG-16</td></tr>
<tr><td>20</td><td>56RF-TG-20</td></tr>
<tr><td>30</td><td>56RF-TG-30</td></tr>
<tr><td>50</td><td>56RF-TG-50</td></tr>
<tr><td>SLI-S</td><td>64</td><td>32</td><td>16</td><td>56RF-TG-16-64B</td></tr>
<tr><td>SLI-L</td><td>256</td><td>160</td><td>10</td><td>56RF-TG-10-256B</td></tr>
<tr><td>Disc – High Impact Resistant</td><td>SLI</td><td>128</td><td>112</td><td>35</td><td>56RF-TG-35HIR</td></tr>
<tr><td rowspan="2">Disc – Mount on Metal</td><td rowspan="2">SLI</td><td rowspan="2">128</td><td rowspan="2">112</td><td>20</td><td>56RF-TG-20MOM</td></tr>
<tr><td>50</td><td>56RF-TG-50MOM</td></tr>
<tr><td rowspan="3">Disc – FRAM</td><td rowspan="3">FRAM</td><td rowspan="3">2048</td><td rowspan="3">2 kB</td><td>20</td><td>56RF-TG-20-2KB</td></tr>
<tr><td>30</td><td>56RF-TG-30-2KB</td></tr>
<tr><td>50</td><td>56RF-TG-50-2KB</td></tr>
<tr><td rowspan="2">Label</td><td rowspan="2">SLI</td><td rowspan="2">128</td><td rowspan="4">112</td><td>54 x 86</td><td>56RF-TG-5486</td></tr>
<tr><td>50 x 50</td><td>56RF-TG-5050</td></tr>
<tr><td>Smart Cards</td><td>SLI</td><td>128</td><td>54 x 86</td><td>56RF-TG-5486SC</td></tr>
<tr><td>Square – High Temperature</td><td>SLI</td><td>128</td><td>50 x 50</td><td>56RF-TG-50HT</td></tr>
</table>

Accessories

Transceiver

Style	Connector Type	No. of Pins	Shield	Wire Size [AWG]	Cat. No.
DC Micro (M12) Patchcords	Female straight to male straight	4	Shielded	22	889D-F5FCDM-J ❶
	Female straight to male right angle				889D-F5FCDE-J ❶
	Female right angle to male straight				889D-R5FCDM-J ❶
	Female right angle to male right angle				889D-R5FCDE-J ❶
DC Micro (M12) Cordsets	Female straight	4	Shielded	22	889D-F5FC-J ❷
	Female right angle				889D-R5FC-J ❷
	Male straight				889D-M5FC-J ❷
	Male right angle				889D-E5FC-J ❷
M12 Terminal Chambers	Female straight	4	—	18…22	871A-TS5-D1
	Female right angle				871A-TR5-D1
	Male straight				871A-TS5-DM1
	Male right angle				871A-TR5-DM1

❶ Available in 0.3, 1, 2, 5, or 10 m lengths.

❷ Available in 2, 5, or 10 m lengths.

Auxiliary Power

Style	Connector Type	No. of Pins	Shield	Wire Size [AWG]	Cat. No.
DC Micro (M12) Patchcords	Female straight to male straight	4	Unshielded	22	889D-F4ACDM-❸
	Female straight to male right angle				889D-F4ACDE-❸
	Female right angle to male straight				889D-R4ACDM-❸
	Female right angle to male right angle				889D-R4ACDE-❸
DC Micro (M12) Cordsets	Female straight	4	Unshielded	22	889D-F4AC-❹
	Female right angle				889D-R4AC-❹
	Male straight				889D-M4AC-❹
	Male right angle				889D-E4AC-❹
M12 Terminal Chambers	Female straight	4	—	22	871A-TS4-D
	Female right angle				871A-TR4-D
	Male straight				871A-TS4-DM
	Male right angle				871A-TR4-DM

❸ Available in 0.3, 1, 2, 5, or 10 m lengths.

❹ Available in 2, 5, or 10 m lengths.

EtherNet/IP

Style	Connector Type	No. of Pins	Shield	Wire Size [AWG]	Cat. No.
M12 D Code Patchcords	Male straight to male straight	4	Unshielded	24	1585D-M4TBDM-❶
	Male straight to male right angle				1585D-M4TBDE-❶
	Male right angle to male right angle				1585D-E4TBDE-❶
M12 D Code Patchcords	Male straight to male straight	4	Shielded	26	1585D-M4UBDM-❶
	Male straight to male right angle				1585D-M4UBDE-❶
	Male right angle to male right angle				1585D-E4UBDE-❶

❶ Available in lengths of 0.3, 1, 2, 5, 10, 15 m in increments of 5 m up to 75 m.

Handheld Interface

Description	Cat. No.
RFID Handheld Interface, 52-Key Directional Pad	57RF-HH-56A
RFID Handheld Interface, 45-Key Pad	57RF-HH-56B

Handheld Accessories

Description	Cat. No.
Domestic single position charging cradle with cable, USB cable, stylus	57RF-HH-56US1
Domestic wall mount power supply, serial cable, USB cable, stylus	57RF-HH-56US2
International power supply kit, serial cable, USB cable, stylus	57RF-HH-56IN
Battery pack, rechargeable	57RF-HH-56BAT
Serial cable, 15 ft, RS-232	57RF-HH-56CA
64 MB CompactFlash card	1784-CF64
128 MB CompactFlash card	1784-CF128

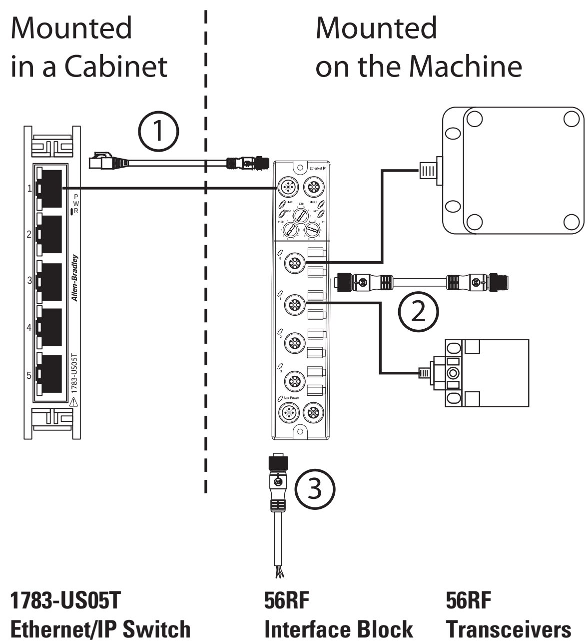
Chapter 3

Electrical Installation

Cable Overview

The EtherNet/IP switch must be mounted inside a control panel. The Bul. 56RF EIP interface block and Bul. 56RF transceivers can be mounted on the machine.

Figure 6 - Transceiver Mounting



Three types of cables are needed.

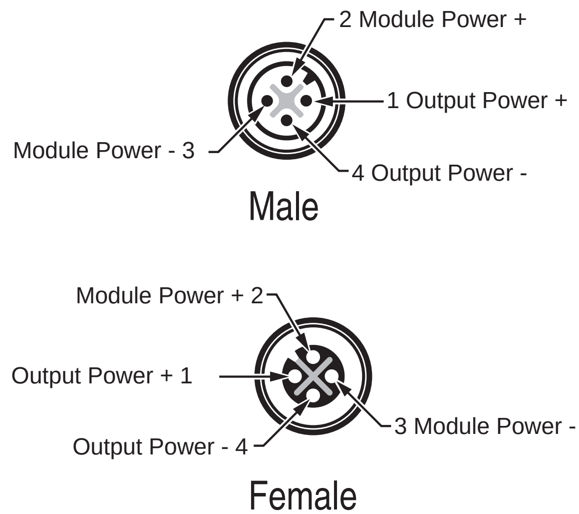
1. This is an EtherNet cable, RJ45 to M12-QD patchcord.
2. This is a 5-pin M12 to 5-pin M12 patchcord. The cable includes a shield that connects to the functional earth point on the interface block.
3. This is a 4-pin female micro QD cordset that connects power to the interface block.

Auxiliary Power Connection

Attach a micro-style 4-pin female to the micro-style 4-pin male receptacle as shown below. The female side is used to daisy chain the power to another device. The power connection is limited to 4 A. When the daisy chain approach is used, the maximum number of interface blocks that can be connected is determined by the total power consumed by each block.

IMPORTANT	Power must be connected to the male connector first. Do not connect power to the female connector and leave the male connector exposed. The pins in the male connector will have 24V DC potential for short circuit.

The pin connections for the aux power connectors are shown below:



The power for the output port is separate from the power to the remaining portions of the interface block. This allows the output device to be turned off, while maintaining power to the transceivers, the input port, and the EtherNet/IP connection. When the output is connected to the safety related portion of the machine control system, an actuator can be turned off, while diagnostic information is still available to the machine control system.

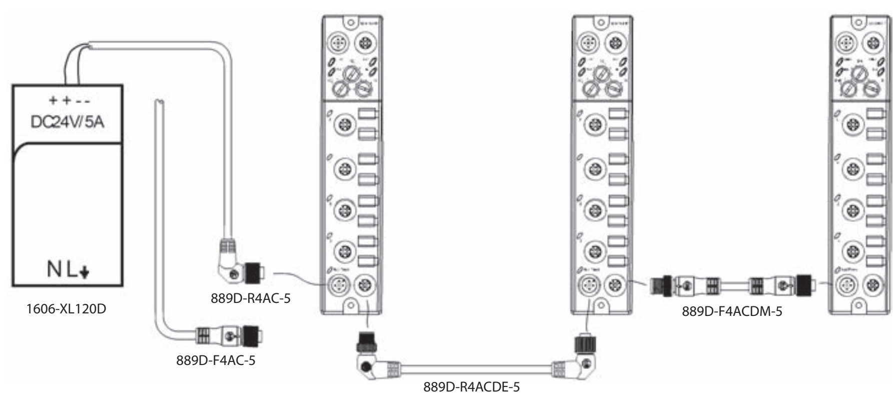
Power Connection Options

Each interface block is limited to 4 A total consumption.

Example 1: Daisy Chain the Power Connections

This example allows for a simple and easy way to distribute power to the RFID system. This approach is preferred when the total current of the RFID system is less than 4 A.

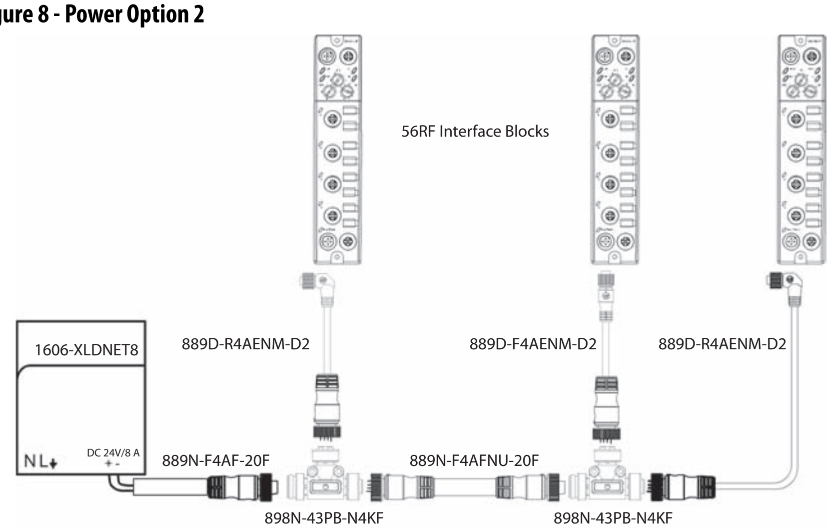
Figure 7 - Power Option 1



Example 2: System Needs More Than 4 A

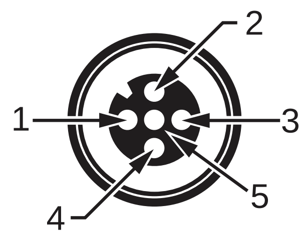
If multiple blocks are required on a machine and the current consumption exceeds 4 A, then a combination of mini-style and micro-style connections can be used to distribute the power. In the example below, mini-style cordsets, patchcords and tees are used to set up the power. A mini-to-micro style patchcord connects each Bul. 56RF interface block with the tee. In this example, the power supply is a Cat. No. 1606-XLDNET8, which can supply up to 8 A to the RFID system.

Figure 8 - Power Option 2



Transceiver Connection

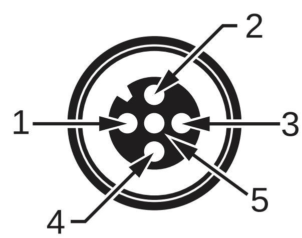
The M12 QD female connector for the transceivers is shown below. Pin 5 is the cable shield connection and is connected only at the block to functional earth (FE).



Pin	Function
1	24V DC power
2	Data +
3	24V common
4	Data -
5	Shield/FE

Digital Input Connection

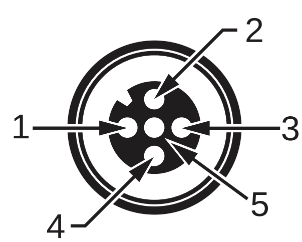
The female M12 QD input connector is shown below.



Pin	Function
1	24V DC power
2	Not used
3	24V common
4	Digital input
5	Shield/FE

Digital Output Connection

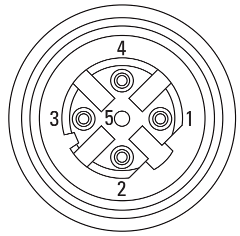
The female M12 QD output connector is shown below.



Pin	Function
1	Not used
2	Not used
3	24V common
4	Digital output
5	Shield/FE

EtherNet I/P Connection

The D-Code M12 connector on the interface block is shown in the figure below.



Pin	Function
1	Tx+
2	Rx+
3	Tx-
4	Rx-
5	Connector shell connected to FE

Use the Cat. No. 1585D-M4DC-H (polyamide small body unshielded) or the Cat. No. 1585D-M4DC-SH (zinc die-cast large body shielded) mating connectors for the D-Code M12 female network connector.

Use two twisted pair Cat 5E UTP or STP cables.

D-Code M12 Pin	Wire Color	Signal	8-Way Modular RJ45 Pin
1	White-Orange	Tx+	1
2	White-Green	Rx+	3
3	Orange	Tx-	2
4	Green	Rx-	6

The 56RF interface block encoders can be connected in either of three network topologies: star, linear or Device Level Ring (DLR).

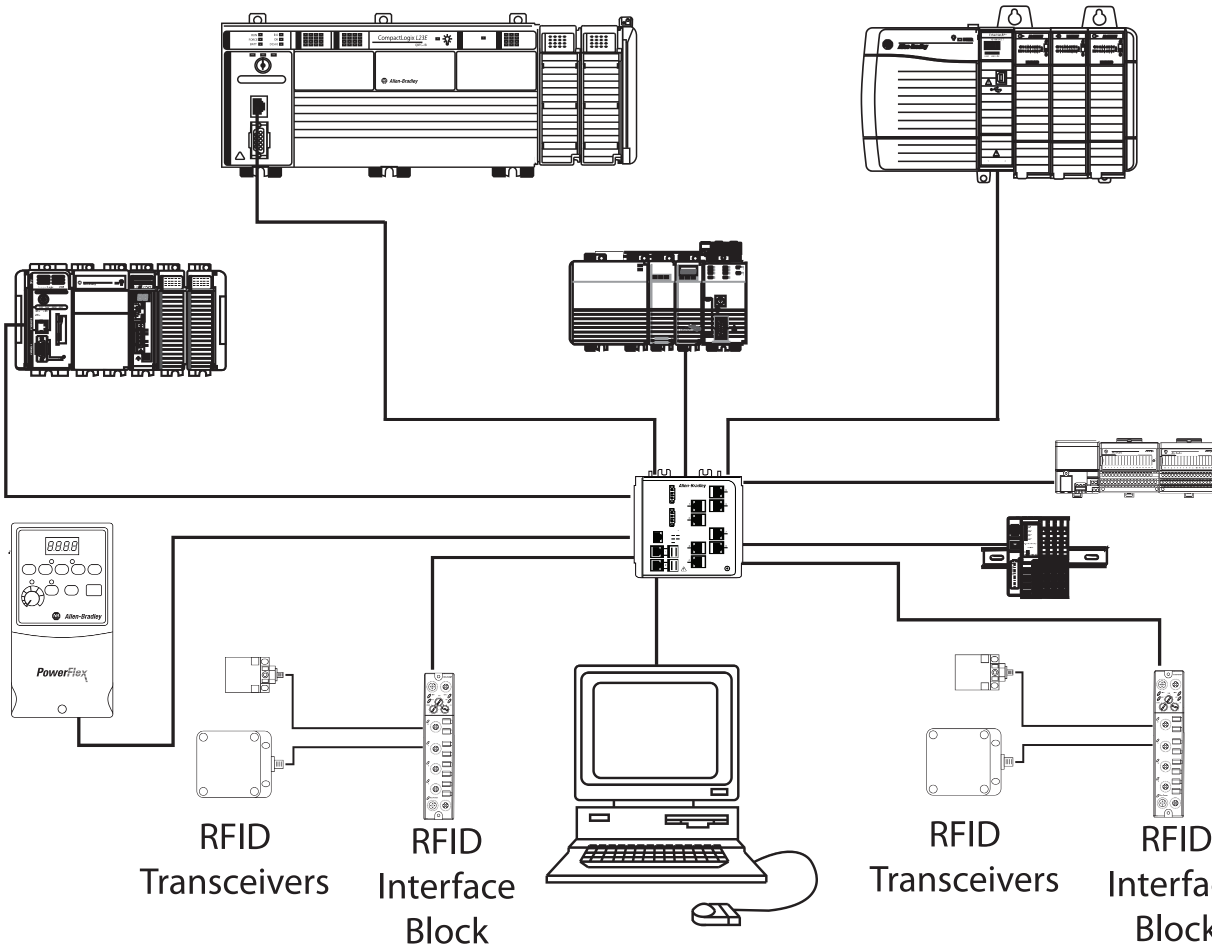
Notes:

EtherNet/IP Addressing

Star Topology

The star structure consists of a number of devices connected to central switch. When this topology is used, only one EtherNet connection can be made to the Bul. 56RF interface block – this connection is made to the Link 1 connector. The Link 2 connection must remain unused.

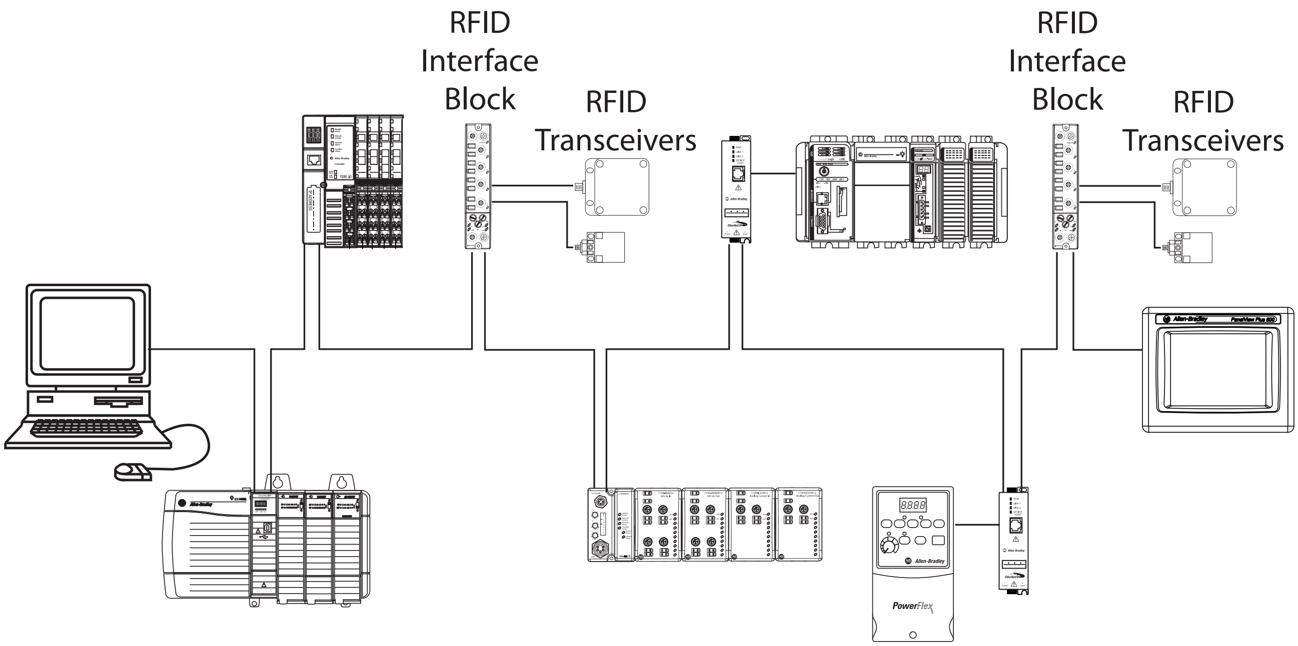
Figure 9 - Star Topology



Linear Topology

The linear topology uses the embedded switching capability to form a daisy-chain style network that has a beginning and an end. Linear topology simplifies installation and reduces wiring and installation costs, but a break in the network disconnects all devices downstream from the break. When this topology is used, both EtherNet connections are used. The network connection to Link 1 or Link 2 does not matter.

Figure 10 - Linear Topology



Device Level Ring (DLR) Topology

A DLR network is a single-fault tolerant ring network intended for the interconnection of automation devices. DLR topology is advantageous as it can tolerate a break in the network. If a break is detected, the signals are sent out in both directions. When this topology is used, both EtherNet connections are used. The network connection to Link 1 or Link 2 does not matter.

Rockwell Automation recommends that you use no more than 50 nodes on a single DLR, or linear, network. If your application requires more than 50 nodes, we recommend that you segment the nodes into separate, but linked, DLR networks.

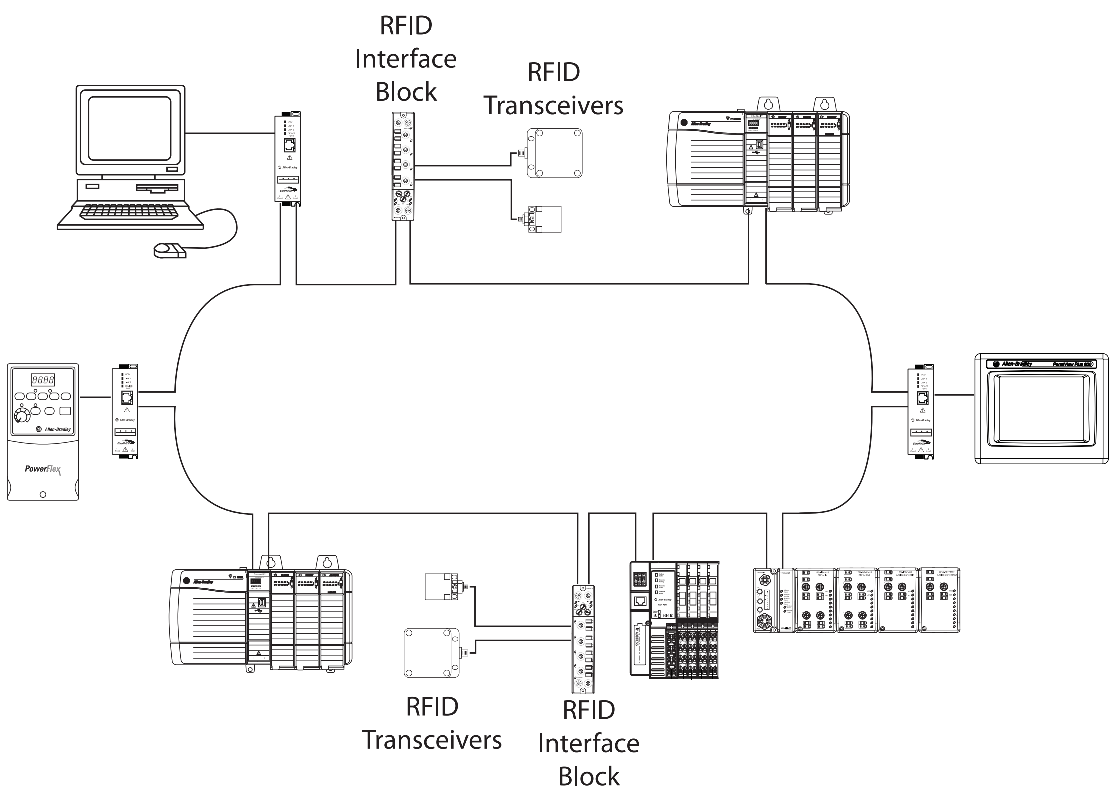
With smaller networks:

- there is better management of traffic on the network.
- the networks are easier to maintain.
- there is a lower likelihood of multiple faults.

Additionally, on a DLR network with more than 50 nodes, network recovery times from faults are higher. The maximum cable length between devices cannot exceed 100 m (328 ft).

For more information on setting up and configuring a DLR network, please refer to ENET-AP005.

Figure 11 - DLR Topology



Setting the Network Address

Before using the Bul. 56RF interface block in an EtherNet/IP network, configure it with an IP address, subnet mask, and optional Gateway address. This chapter describes these configuration requirements and the procedures for providing them. The address can be set in one of three ways:

- Use the Network Address switches.
- Use the Rockwell BootP/DHCP utility (version 2.3 or greater), which ships with RSLogix™ 5000.
- Use RSLinx® software.

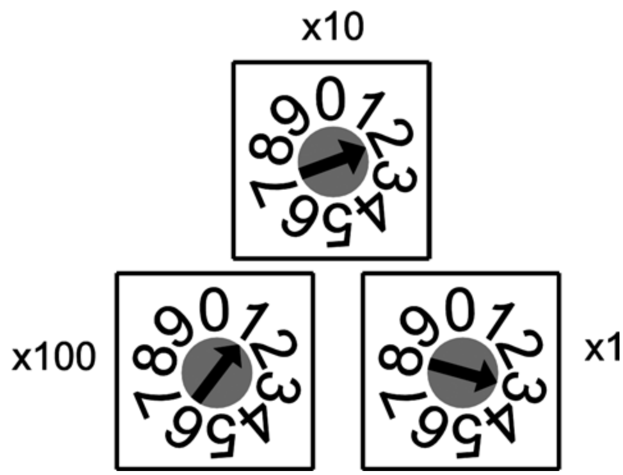
IP network addresses have a format of xxx.xxx.xxx.xxx. The user must know what values are being used for the network. If your network has the fundamental 192.168.1.xxx scheme, then you can simply use the three network address switches. If your network is something other than 192.168.1.xxx, you must use advanced tools, such as the BootP/DHCPserver, to assign an IP address. After the address is set, you can use RSLinx to change the address.

Fundamental IP Addresses: 192.168.1.xxx

If your network scheme is 192.168.1.xxx, then you can adjust the network address switches to set the IP address. Remove the covers of the three network address screws. Use a small blade screwdriver to rotate the switches. Line up the small notch on the switch with the number setting you wish to use. Valid settings range from 001…254

When the switches are set to a valid number, the IP address of the interface block is 192.168.1.xxx (where xxx represents the number set on the switches). Cycle the power and the valid setting becomes effective immediately.

The example below shows an address setting of 192.168.1.123.



The subnet mask of the interface block is automatically set to 255.255.255.0 and the gateway address is set to 0.0.0.0. When the interface block uses the network address set on the switches, the interface block does not have a host name assigned to it or use a Domain Name Server (DNS).

Advanced IP Addresses

The following steps show how to change the IP address from the fundamental 192.168.1.xxx to an advanced address. This assumes the Bul. 56RF interface block was already set up with an IP address using the network address switches. The examples below show the change process using specific addresses. The user is not limited to these addresses; the user can select any address that meets their needs. In the example below, we change from 192.168.**1**.115 to 192.168.**2**.115.

1. Set address switches to 888 and cycle the power.

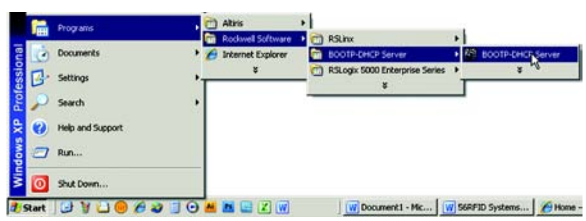
   On the Bul. 56RF interface block, the address switches had previously been to 115. Set the address switch settings to 888. Cycle the power and wait until the MOD indicator is blinking red. The MOD indicator blinks red once, green once, then solid red for a short while, then blinks green once, and finally blinks red continuously (about once each second). This takes about 10 seconds after power is restored. The interface block is reset to its factory setting.

2. Set the address switches to 999 and cycle the power.

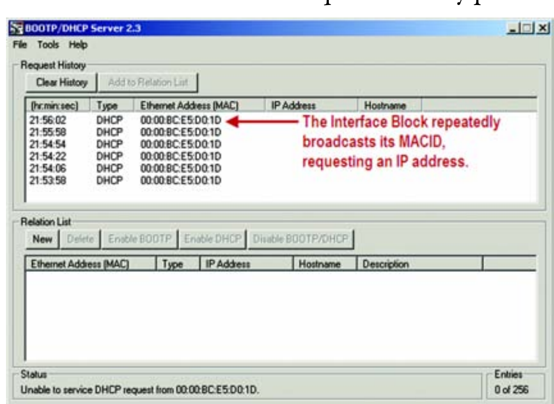
   On the Bul. 56RF interface block, set the address switch settings to 999. Cycle the power and wait until the MOD indicator is solid green. The MOD indicator blinks red once, green once, solid red for a short while and finally turns solid green. This takes about 10 seconds after power is restored. The interface block IP address is reset.

3. Use BootP/DHCP Server to set new address

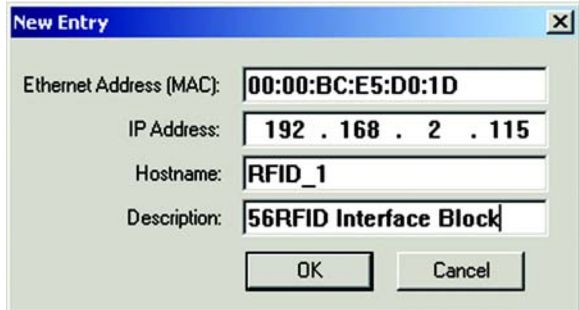
   Use the Rockwell BootP/DHCP Server utility to assign a valid address to the interface block. From the Start button, select Programs > Rockwell Software > BOOTP-DHCP Server > BOOTP-DHCP Server.



   When power is restored, the interface block repeatedly broadcasts its MACID and requests an IP address. The BOOTP-DHCP server displays the MACID in the Request History panel.



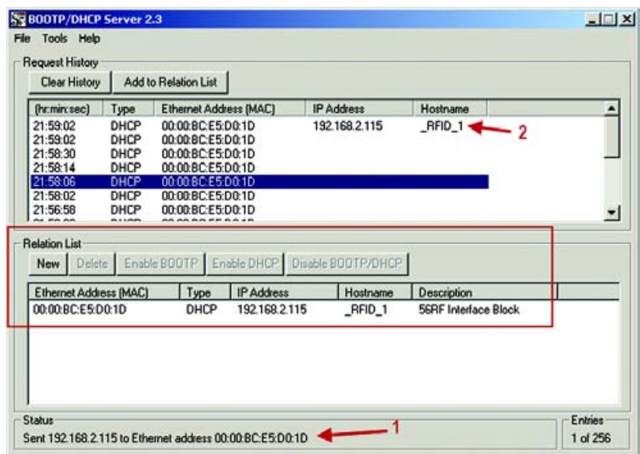
Double-click on one of the EtherNet Address (MAC) of the device. The New Entry dialog appears showing the EtherNet Address (MAC) of the device.



Type in the IP Address, Hostname and Description and click **OK**. The Hostname and Description are optional fields; they can be left blank.

The device is added to the Relation List, displaying the EtherNet Address (MAC) and corresponding IP Address, Hostname, and Description.

When the address is assigned to the Bul. 56RF interface block, 1) the Status message is updated, and 2) the IP address appears in the Request History window.

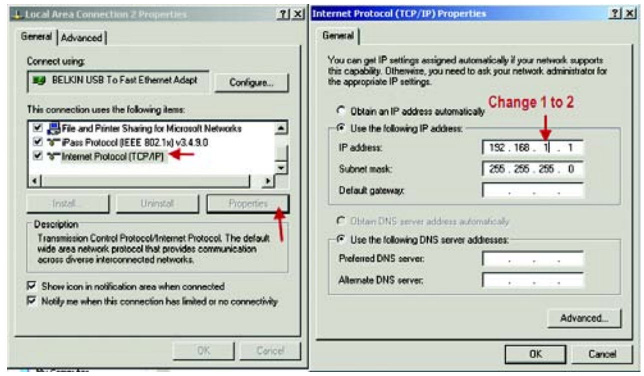


IMPORTANT	Wait for the Status message to show "Sent 192.168.2.115 to EtherNet address 00:00:BC:E5:D0:1D." This may take a few seconds to 30 seconds.

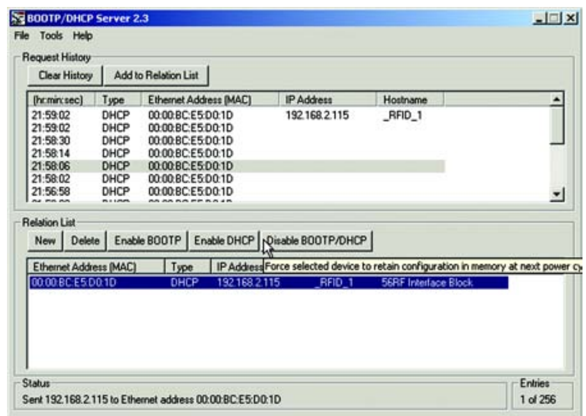
4. At this point, the IP addresses of other devices should be changed.

5. Change the Network Adaptor to 192.168.2.1.

Open the network connections of the host computer. Highlight the Internet Protocol (TCP/IP) connection. Click **Properties**. In the IP Address field, set the IP Address to 192.168.2.1. Click **OK**. Click **Close** to close the Local Area Connection window (this window must be closed to apply the new address).

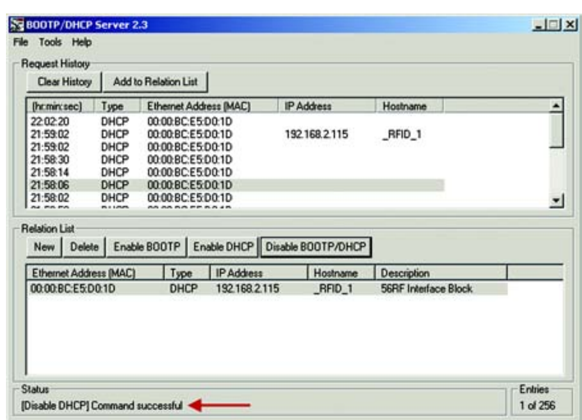


6. Disable DHCP.



Click on (only once) the interface block in the Relation List to highlight it. Then click Disable BOOTP/DHCP. This instructs the Bul. 56RF interface block to retain the IP Address at the next power cycle.

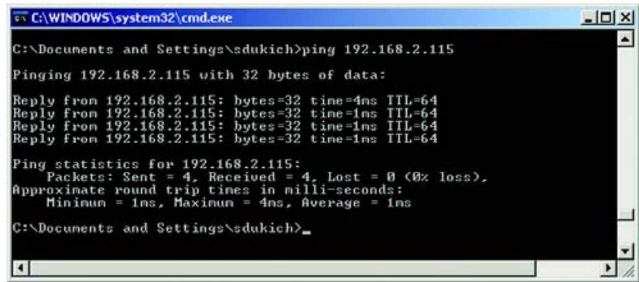
Wait for the Status message to show that the command was successfully sent. If not, repeat this step.



Click **File** > **Save As** to save the relationship, if desired.

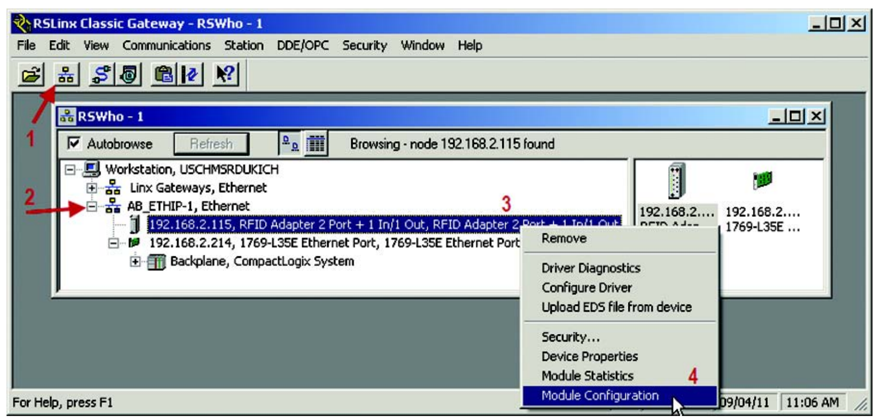
Cycle the power to the Bul. 56RF interface block. You should no longer see the Bul. 56RF interface block appear in the Request History panel.

From a DOS prompt, you can ping the new address. The response should be 4 packets sent, 4 packets received and 0 lost.



Change IP Address from One Advanced Address to another Advanced Address

The easiest way to change the IP Address from one non-simple address to another non-simple address is to use RSLinx. In this case, the three network switches on the Bul. 56RF interface block are set to 999, and the address has been previously set using the BootP/DHCP server. The example below shows how to change the IP address from 192.168.2.115 to 192.168.3.115.

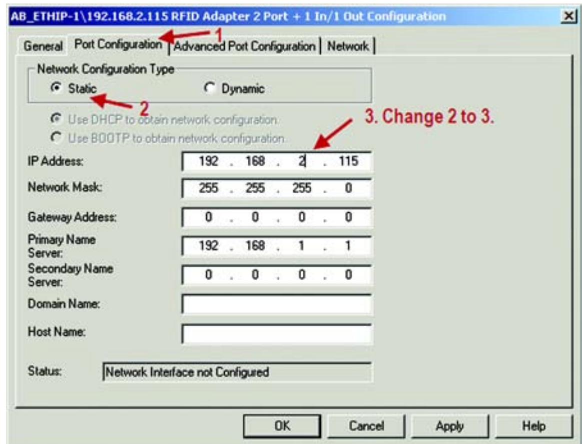


Open RSLinx.

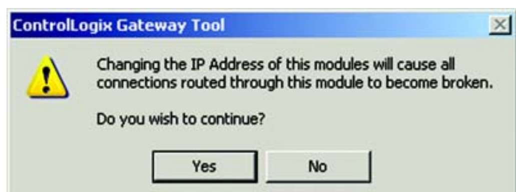
1. Click on the RS-Who icon.
2. Expand the EtherNet connection.
3. Right click on the RFID Adaptor.
4. Click Module Configuration.

The Configuration window appears.

1. Click on the Port Configuration tab.
2. Set the Network Configuration Type to Static (if not already done).
3. Change the IP Address to the new address. In this example, the address will be changed form **192.168.2.115** to **192.168.3.115**.

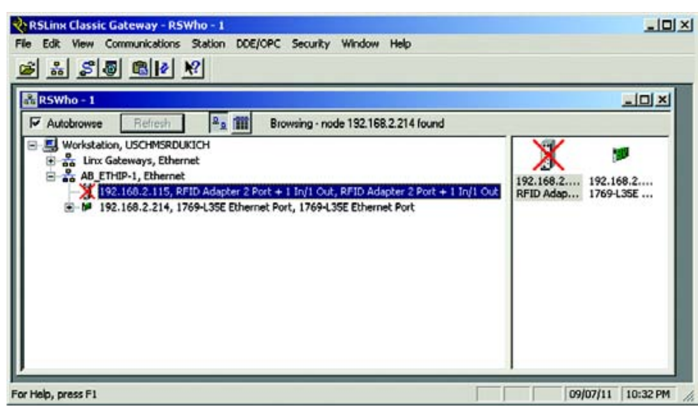


Click **Yes** to confirm the change.



Click **OK** to close the configuration window.

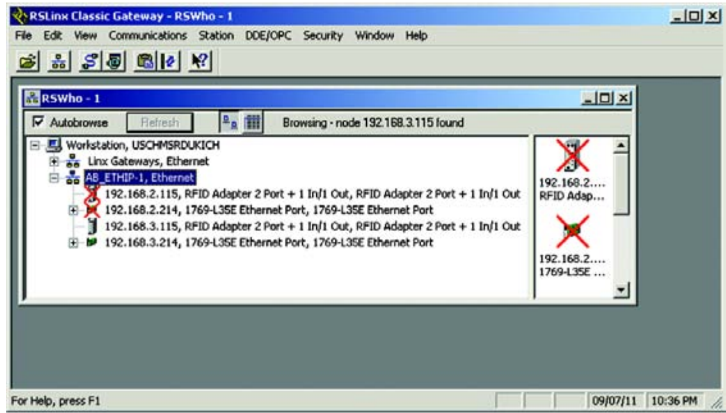
RSLinx places an X over the RFID adaptor because it can no longer communicate with it.



Use the same steps to change the IP address of the other devices on the network.

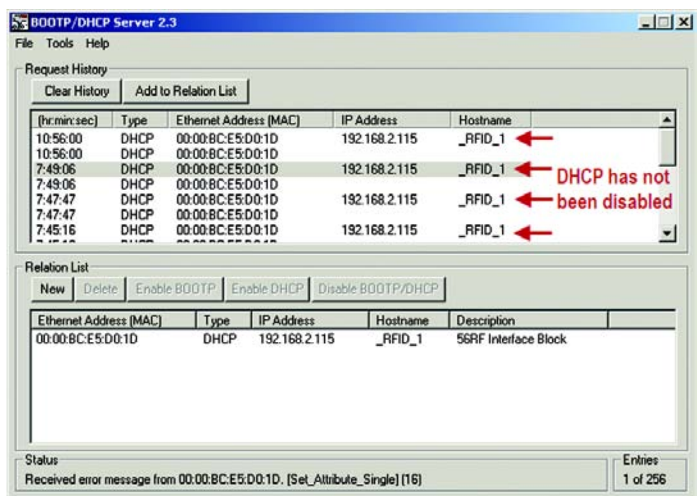
Change the Network adaptor address to 192.168.3.1.

Close and re-open the RSWho window. The older addresses are not available and the new addresses (192.168.3.115 and 192.168.3.214) appear.



Note: If DHCP is not disabled, the Bul. 56RF interface block will show two requests in the DHCP Server at each power up.

In the picture below, power was cycled to the Bul. 56RF interface block at 7:45:16, 7:47:47, 7:49:06 and again at 10:56:00. Each time power was applied, the Bul. 56RF interface block notified the BootP/DHCP server of its IP Address. This indicates that DHCP has not been disabled. If DHCP is disabled, the Bul. 56RF interface block would show nothing.



IP Address 888

Address 888 is used to reset the interface block to the factory defaults. Rotate the address switches to 888 and cycle the power. The interface block clears out the current assigned IP Address.

The MOD indicator blinks the following pattern: blinks red once, green once, then solid red, then blinks green once, and final blinks continuous red about once each second. The reset process takes about 10 seconds.

Mechanical Installation

Each of the transceivers has a similar but unique RF field that it generates.

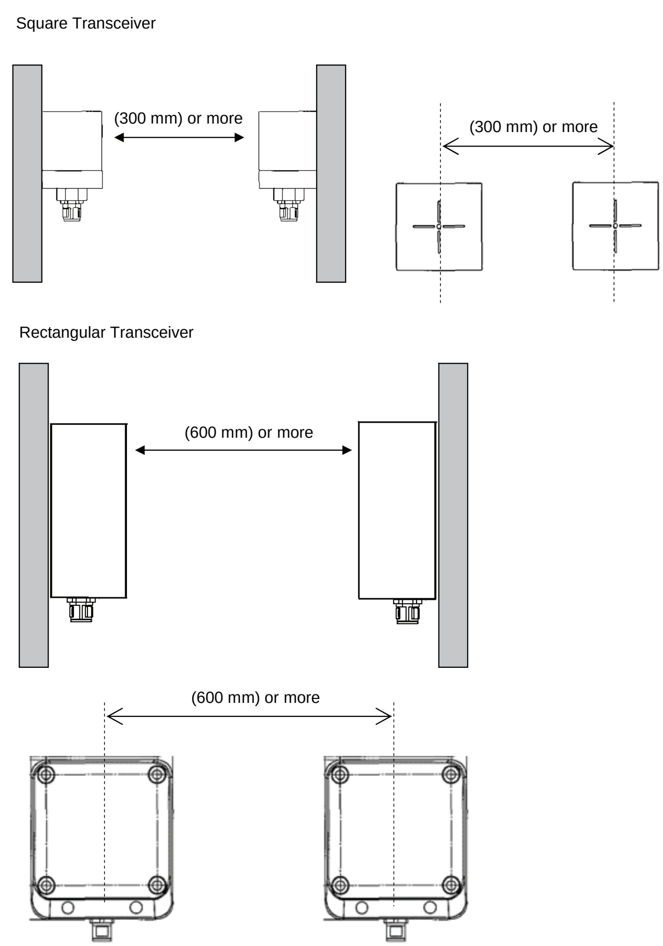
Fastening

Attach the transceiver to the flat plate with M5 screws. The tightening torque must be 1.5 N•m for the M5 screw.

Spacing Between Transceivers

Installing more than one transceiver causes radio frequency interference and may result in the difficulty of the tag communication. Keep a sufficient distance between the transceivers as shown in Figure 12.

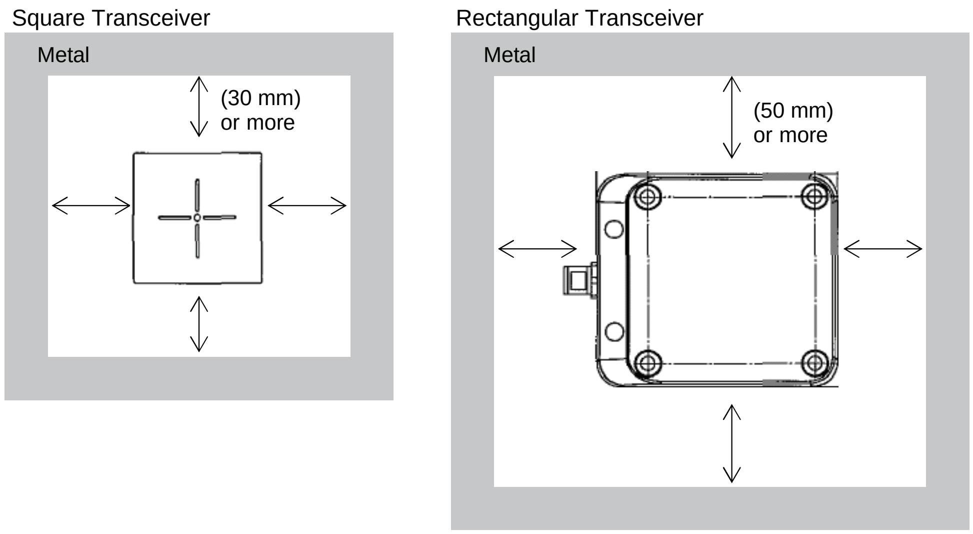
Figure 12 - Spacing Between Transceivers



Spacing Next to Metal Surfaces

For the square transceiver, the communication distance will drop significantly when the distance between the transceiver and any surrounding metal is 30 mm (1.2 in.) or less. For the rectangular transceiver, the communication distance will drop significantly when the distance between the transceiver and any surrounding metal is 50 mm (2 in.) or less.

Figure 13 - Tranceiver Spacing with Metal Surfaces



Transceiver Field Maps

The transceiver has a three-dimensional RF field emanating from its sensing surface. The field consists of a main center lobe and a secondary side lobe.

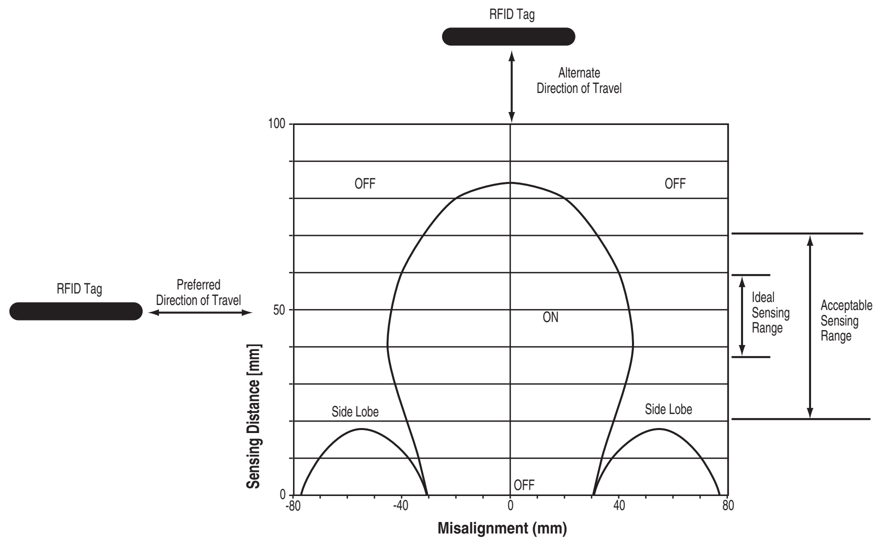
The RF tags must enter the RF field once, stay long enough to complete the read and write cycles, and then to leave the field smoothly and efficiently.

Ideally, the RFID tag should pass through the widest section of the main lobe. This maximizes the time the transceiver has for reading and writing. Avoid the top of the field, and avoid the side lobes.

The preferred direction of travel is for the tag to pass across the RFID sensor surface. The tag can also approach the sensor surface directly and then move away directly backwards or to the side.

Figure 14 shows the field map of the 65mm x 65mm transceiver.

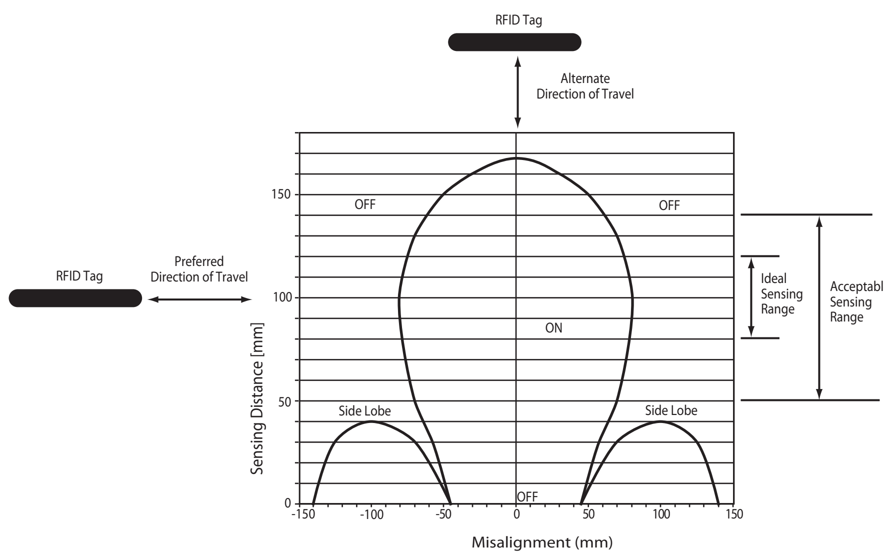
Figure 14 - 65 x 65 mm Transceiver



Referenced for a 50 mm disc tag

The field map for the 80 mm x 90 mm transceiver, shown in Figure 15, is very similar.

Figure 15 - 80 x 90 mm Transceiver

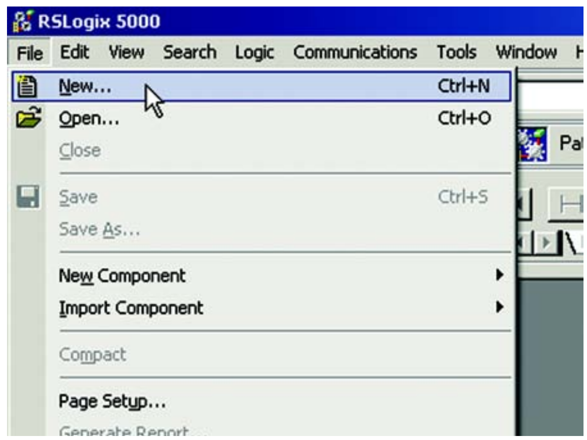


Referenced for a 50 mm disc tag

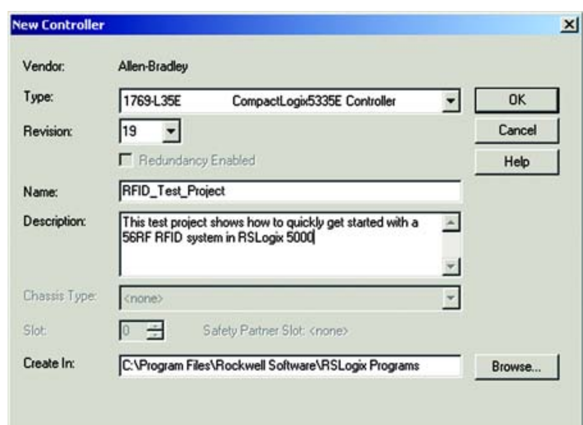
Notes:

Add Your RFID Interface Block to an RSLogix 5000 Program

1. Open RSLogix 5000.
2. Click **File**>**New**.

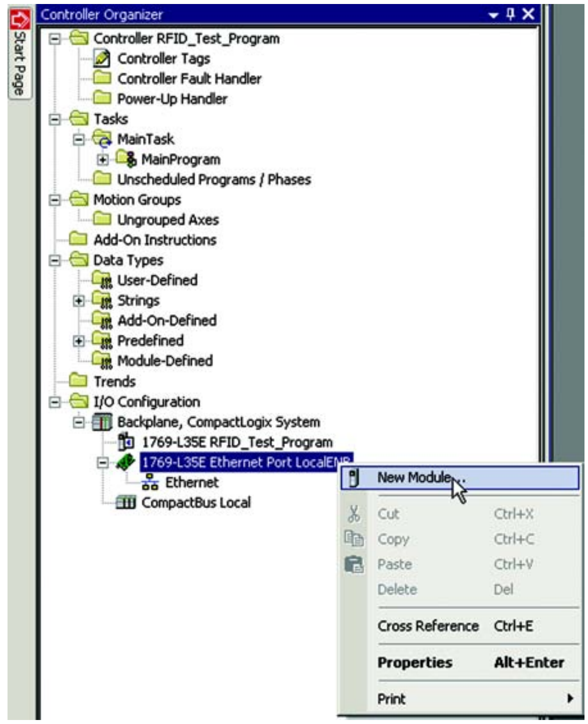


3. Enter the new controller information.

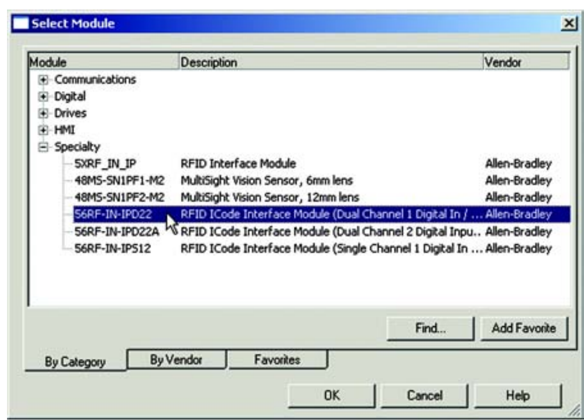


4. Right click on the EtherNet port of the controller.

5. Click **New Module**.



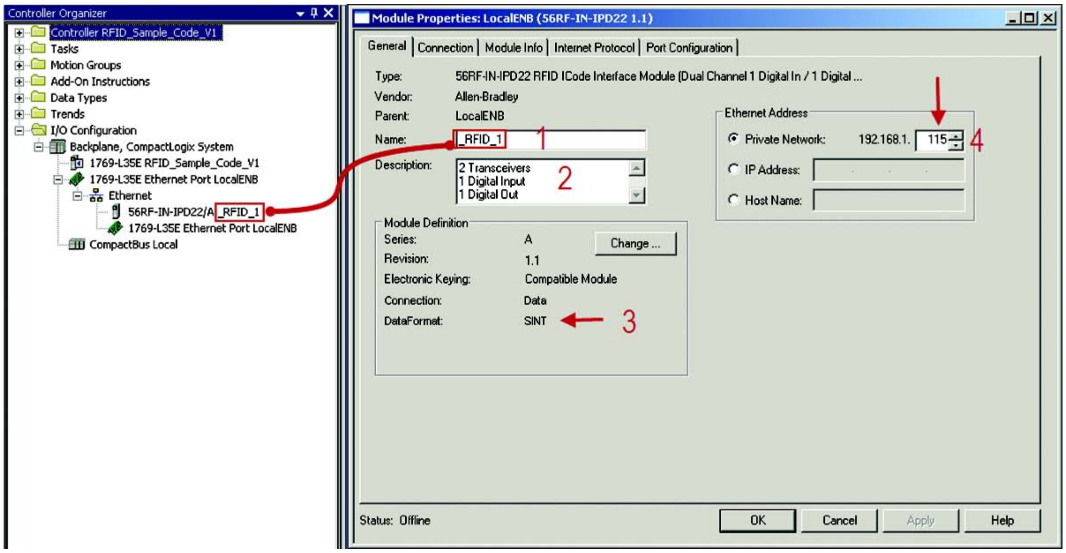
6. Select the desired 56RF module and click **OK**.



General Tab

The general panel describes the device, its definition and its IP address.

1. Make the four changes shown below and click **Apply**.



a. Enter a name for the module. In this example, the name is RFID_1. You may have multiple modules, so be sure to give it a brief but descriptive name. The name that you assign to the module appears in the Controller Organizer navigation pane. The name will also appear in the description of the tags, which are described below.

b. Enter a description of the module or its function.

c. Set the EtherNet Address for the module. In this example, the address is 192.168.1.115. The 115 reflects the address of the three rotary switches on the Bul. 56RF interface block.

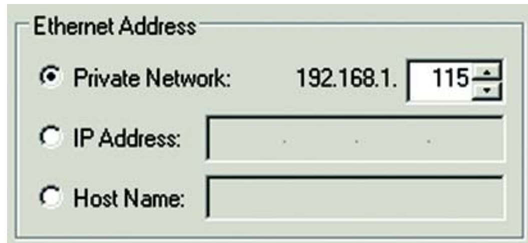
d. The Data Format can be left as SINT (preferred) or changed to INT (for compatibility with non-Rockwell RFID tags).

TIP A SINT is a signed single byte integer, which can represent numbers from -255…255 in decimal format (-F…FF in hexadecimal format). An INT is a signed two byte integer, which can represent numbers from -65535…65535 in decimal format (-FFFF…FFFF in hexadecimal format).

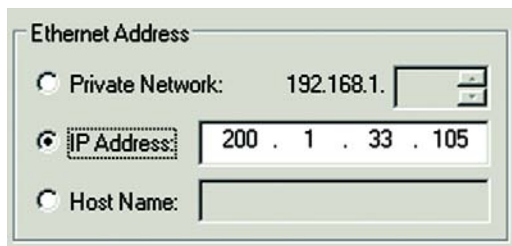
EtherNet Address

When the controller is offline, the EtherNet address can be set. The user has three options.

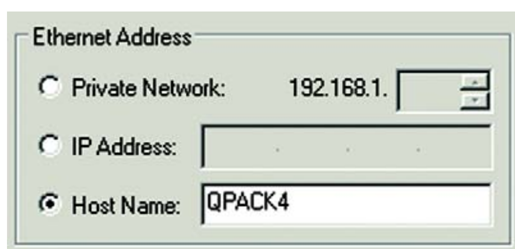
1. When a Private Network is used, click on the Private Network radio button. Enter a value for the last octet between 1…254. Be sure not to duplicate the address of an existing device. In preceding example, the address of the RFID block is 192.168.1.115.



2. When multiple networks exist, the user may elect to set the address to some other value. When offline, simply click the IP Address radio button and enter the desired address.



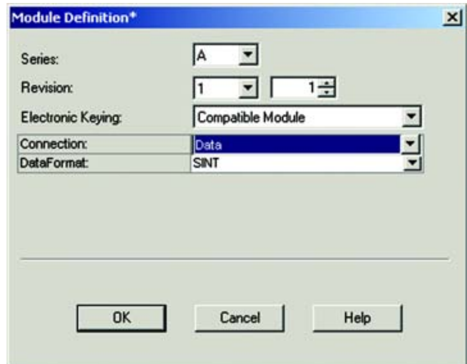
3. Click the Host Name radio button and type in the name of the host. In the example below, the Host name is QPACK4.



Module Definition

The user should not have to make changes to the default values. If necessary, changes can be made by clicking the **Change** button.

The user can change the Series, Revision, Electronic Keying, Connection and Data Format. Click the down arrow on the Data Format field and select SINT.



Click **OK** to accept the changes (or **Cancel** to retain the original settings). Click **Help** for more info.

Connection Tab

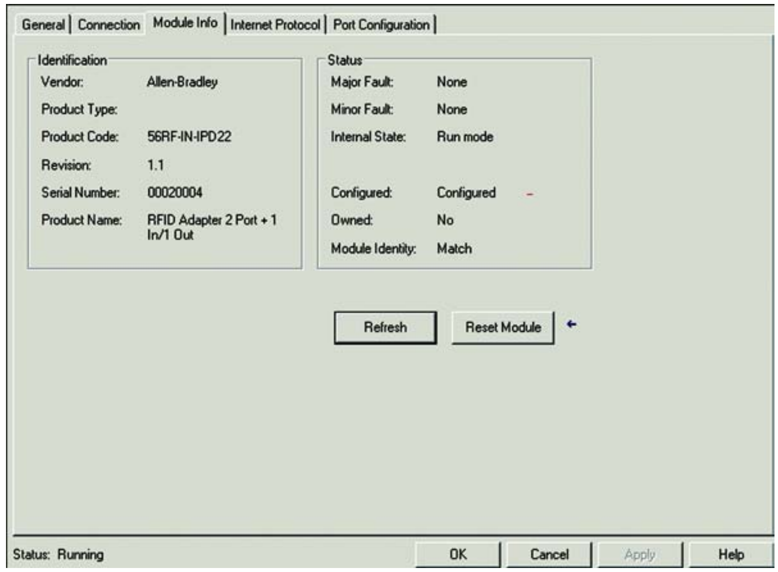
You should not have to change any settings here.

- **Requested Packet Interval** – Specify the number of milliseconds between requests for information from the controller to the RFID block. The block may provide data on a shorter interval, but if no data is received, the controller asks the RFID block for a status update. Minimum setting is 2. Maximum setting is 750.
- **Inhibit Module** – When checked, the RFID block is not polled for information, and any information provided will be ignored by the controller.
- **Major Fault on Controller If Connection Fails While In Run Mode** – Check this box if a connection failure should be considered a major fault.
- **Use Unicast Connection over EtherNet/IP** –Unicast connections are point-to-point connections. Multicast connections are considered one-to-many. Unicast reduces the amount of network bandwidth used.
- **Module Fault** – Fault messages will appear in this box.

Module Info

The Module Info tab contains read only data that is populated when the controller goes on-line (a program is downloaded to or uploaded from the controller).

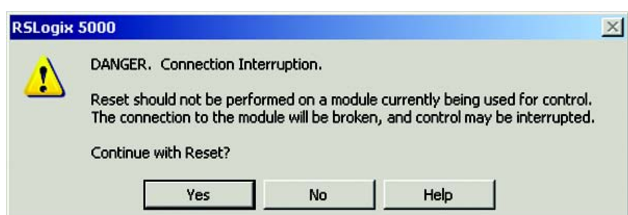
In the left panel, the AOP shows the vendor, product type, product code. Revision level, serial number, and product name.



In the right panel, the AOP shows the fault status, internal state (i.e., Run mode), and whether the file is owned and Module Identity.

The Refresh and Reset Module button are active when the controller is online.

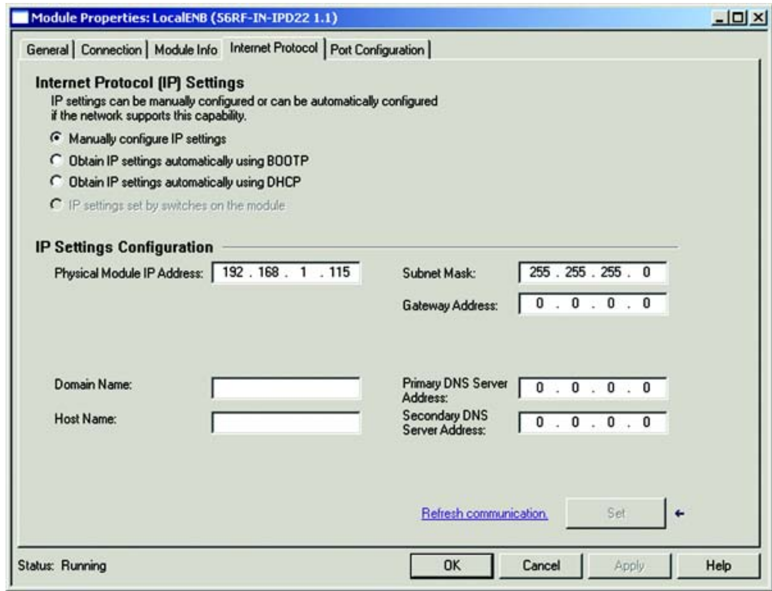
- **Refresh** – Click this button to refresh the data in the window.
- **Reset Module** – Click this button with care as it disconnects the module momentarily and control will be interrupted. The following warning window appears.



Click **Yes** or **No** as needed. Click **Help** for further information.

Internet Protocol Tab

For the purposes of this user manual, the user is expected to use a Private Address, that is, and address of 192.168.1.xxx. This window is automatically populated with the data.

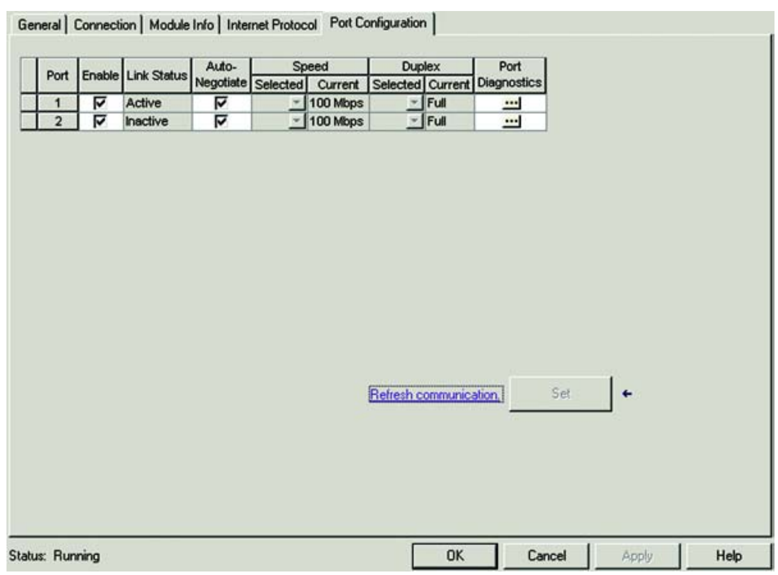


Port Configuration Tab

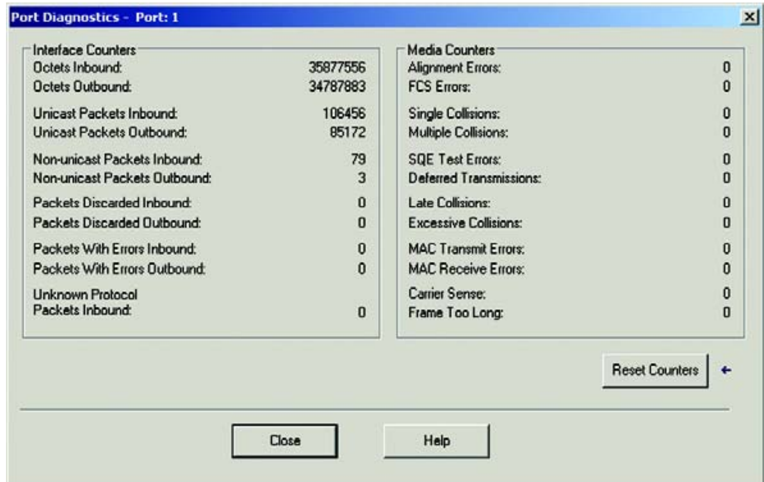
The Port Configuration fields should not need to be changed for the Quick Start process. These fields only become active when the controller is on line.

The number of ports showing in this window will vary depending on the block used. There should be either one or two ports.

The following window shows two ports. Port 1 is active, while Port 2 is inactive.



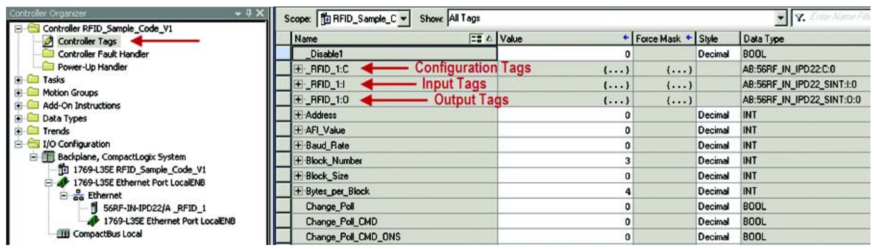
Click on the ellipsis (...) under the Port Diagnostics. A window pops up showing the communications taking place between the controller and the transceiver connected to the port.



RSLogix 5000 Controller Tags

During the module installation, the RFID_1 tags are automatically loaded as controller tags. This makes the tags available to all programs.

In the Controller Organizer, click on the **Controller Tags**.

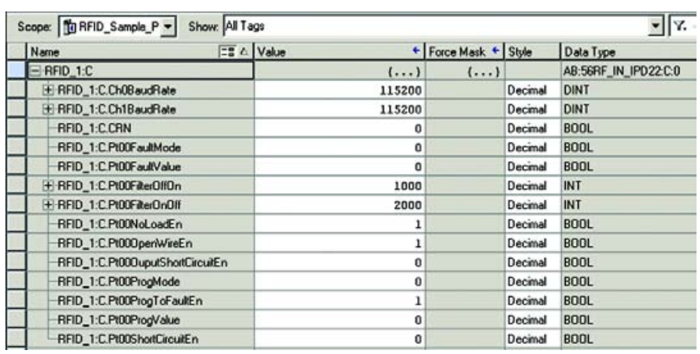


Three categories of tags appear. The tag name is comprised of the module name followed by a:

- ":C" for Configuration
- ":I" for Input
- ":O" for Output.

Configuration Image Table and Tags

Expand the RFID_1:C by clicking the "+" box. This shows the configuration image table, which has the following tags:

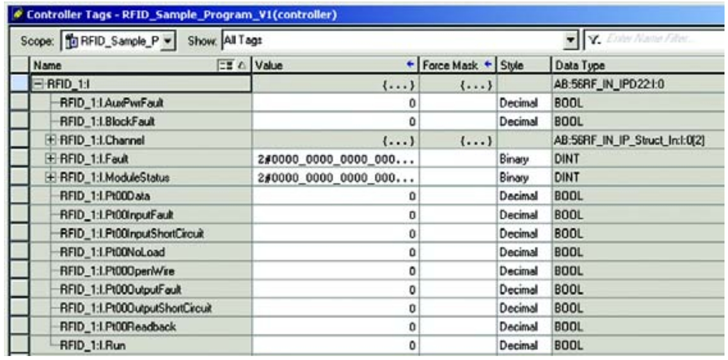


- **Ch0BaudRate** – The baud rate for communication for Channel 0 from the RFID block to the RFID transceiver is stored in this tag. Allowable baud rates are 9600, 19200, 38400, and 115200. The default value is 115200.
- **Ch1BaudRate** – The baud rate for communication for Channel 1 from the RFID block to the RFID transceiver is stored in this tag. Allowable baud rates are 9600, 19200, 38400, and 115200. The default value is 115200.

- **CRN** – The Configuration Revision Number is used internally with RSLogix for configuration information. The user does not need to use this tag.
- **Pt00FaultMode** – The Pt00FaultMode is used in conjunction with FaultValue to configure the state of output 0 when a communications fault occurs. A value of 0 means that, in the case of a communications fault, the value in FaultValue will be used (Off or On). A value of 1 means that the last state will be held. By default this value is 0.
- **Pt00FaultValue** – The Pt00FaultValue is used in conjunction with FaultMode to configure the state of output 0 when a communications fault occurs. A value of 0 is Off, and a value of 1 is On. By default the value is 0.
- **Pt00FilterOffOn** – The Pt00FilterOffOn is used to determine the Off to On delay time for input point 0 before the interface considers the input point on or True. A value of 0 indicates there is no delay from an off condition to an on condition; the only delay would be a hardware delay. A value >0 would delay the input turning on by the configured value in milliseconds. By default this value is 0.
- **Pt00FilterOnOff** – The Pt00FilterOnOff is used to determine the On to Off delay time for input point 0 before the interface considers the input point off or False. A value of 0 indicates there is no delay from an on to off condition; the only delay would be a hardware delay. A value >0 would delay the input turning off by the configured value in milliseconds. By default this value is 0.
- **Pt00NoLoadEn** – The Pt00NoLoadEn is used to enable or disable No Load diagnostic detection for output 0. A value of 1 means that No Load diagnostic detection is enabled. A value of 0 means that No Load diagnostic detection is disabled. By default this value is 0.
- **Pt00OpenWireEn** – The Pt00OpenWireEn is used to enable or disable the open wire detection for input point 0. A value of 1 means that open wire detection is enabled. A value of 0 means that open wire detection is disabled. By default this value is 1.
- **Pt00OutputShortCircuitEn** – The Pt00OutputShortCircuitEn is used to enable or disable the short circuit detection for output point 0. A value of 1 means that short circuit detection is enabled. A value of 0 means that short circuit detection is disabled. By default this value is 0.
- **Pt00ProgMode** – The Pt00ProgMode is used in conjunction with ProgValue to configure the state of output 0 when the controller is in Program mode. A value of 0 means that the ProgValue (Off or On) will be used when the controller is in Program mode. A value of 1 means that the last state will be held. By default this value is 0.
- **Pt00ProgValue** – The Pt00ProgValue is used in conjunction with ProgMode to configure the state of output 0 when the controller is in Program mode. A value of 0 is Off, and a value of 1 is On. By default this value is 0.
- **Pt00ShortCircuitEn** – The Pt00ShortCircuitEn is used to enable or disable the short circuit detection for input point 0. A value of 1 means that short circuit detection is enabled. A value of 0 means that short circuit detection is disabled. By default this value is 0.

Input Image Table and Tags

Expand the RFID_1:I by clicking the "+" box. This shows the input image table, which has the following tags:



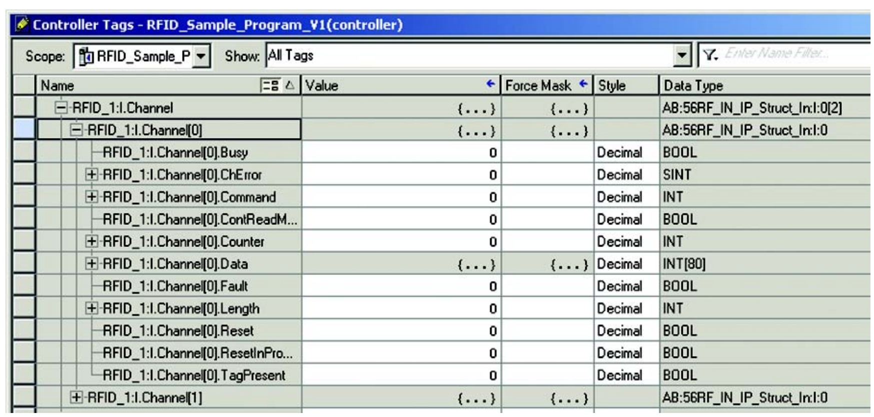
Name	Value	Force Mask	Style	Data Type
RFID_1:I	{...}	{...}		AB:56RF_IN_IPD22:I:0
RFID_1:I.AuxPwrFault	0		Decimal	BOOL
RFID_1:I.BlockFault	0		Decimal	BOOL
RFID_1:I.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_In:I:0[2]
RFID_1:I.Fault	2#0000_0000_0000_000...		Binary	DINT
RFID_1:I.ModuleStatus	2#0000_0000_0000_000...		Binary	DINT
RFID_1:I.Pt00Data	0		Decimal	BOOL
RFID_1:I.Pt00InputFault	0		Decimal	BOOL
RFID_1:I.Pt00InputShortCircuit	0		Decimal	BOOL
RFID_1:I.Pt00NoLoad	0		Decimal	BOOL
RFID_1:I.Pt00OpenWire	0		Decimal	BOOL
RFID_1:I.Pt00OutputFault	0		Decimal	BOOL
RFID_1:I.Pt00OutputShortCircuit	0		Decimal	BOOL
RFID_1:I.Pt00Readback	0		Decimal	BOOL
RFID_1:I.Run	0		Decimal	BOOL

- **AuxPwrFault** – The AuxPwrFault bit will indicate if there is no auxiliary power detected. A value of 0 indicates no fault, a value of 1 indicates a fault condition.
- **BlockFault** – The Block Fault bit will indicate if any of the RFID channels or input/output points is in a fault condition. A value of 0 indicates the RFID channels and input/output points are functioning correctly, a value of 1 indicates one or more of the RFID channels and/or input/output points are in a fault condition. Individual RFID channel fault bits are contained within each associated Channel[x] input word.
- **Channel** – See Input Channel Tags.
- **Fault** – The Fault word is a 4-byte value that will store the connection status between the interface and the controller. A value of 0 indicates a connection has been established, and value of -1 indicates no connections.
- **ModuleStatus** – The Module status is a 4-byte value that will contain the overall status of the module. A value of 0 or 1 will indicate the module is functioning with no faults, a value greater than 1 indicates a fault condition exists. The ModuleStatus word will vary slightly based on the configured unit.
- **Pt00Data** – The Pt00Data bit will indicate if the status of input point 0. A value of 0 indicates open, a value of 1 indicates closed.
- **Pt00InputFault** – The Pt00InputFault bit will indicate if the input point 0 has a fault condition. Input faults would be Open Wire and/or Short Circuit. A value of 0 indicates no fault condition, whereas a value of 1 indicates a fault condition.
- **Pt00InputShortCircuit** – The Pt00InputShortCircuit bit will indicate if the input point 0 has an short condition. A value of 0 indicates no fault; a value of 1 indicates a fault condition. Short circuit detection can be enabled or disabled during configuration.
- **Pt00NoLoad** – The Pt00NoLoad bit will indicate if the output point 0 has a no load condition; No load detectioin only occurs when the output point is OFF. A value of 0 indicates no fault; a value of 1 indicates a fault condition. No load detection can be enabled or disabled during configuration.

- **Pt00OpenWire** – The Pt00OpenWire bit will indicate if the input point 1 has an open wire condition. A value of 0 indicates no fault, a value of 1 indicates a fault condition. Open wire detection can be enabled or disabled during configuration.
- **Pt00OutputFault** – The Pt00OutputFault bit will indicate if the output point 0 has a fault condition. Output faults would be No Load and/or Short Circuit. A value of 0 indicates no fault; a value of 1 indicates a fault condition.
- **Pt00OutputShortCircuit** – The Pt00OutputShortCircuit bit will indicate if the output point 0 has a short condition. A value of 0 indicates no fault; a value of 1 indicates a fault condition; output short-circuit detection only occurs when the output is ON. Short circuit detection can be enabled or disabled during configuration.
- **Pt00Readback** – The Pt00Readback bit will indicate the status of the output point Pt00Data. If the output bit Pt00Data is 1, indicating that the output has been commanded to turn ON, then when the output point turns ON Pt00Readback will contain the value of 1.
- **Run** – The Run bit will indicate if the block is in run or program mode. A value of 1 indicates the block is in run mode, a value of 0 indicates the block is in program mode.

Input Channel Tags

Expand the RFID_1:Channel by clicking the "+" box. This shows that two channels exist: Channel[0] and Channel[1]. Expand the RFID_1:Channel[0] by clicking the "+" box. Each channel has the following tags:



- **Busy** – The channel Busy bit will indicate the status of an RFID channel. A value of 0 indicates that the RFID channel is not executing a command, a value of 1 indicates a command is in the process of executing on that channel.
- **ChError** – The channel ChError is a 1-byte word that will contain the last error code for that channel. A value of 0 indicates no error, a value >0 indicates some error. Refer to Error Codes for RFID Interface Block on page 129 for a list of the error codes.
- **Command** – The channel command word is a 2-byte value that will store the last command that the channel received; at power up this value must be 0. The allowable commands are listed in Table 17:

Table 17 - Allowable Commands

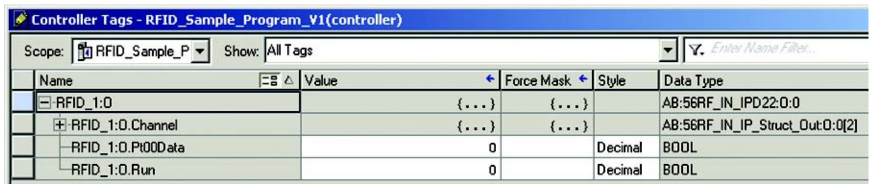
Value	Command	Description
1	Read Single Block	Reads a single block of user data.
2	Read Multiple Blocks	Reads multiple blocks of user data from a tag.
3	Multi-tag Block Read	Reads information from up to four tags.
4	Read Byte	Reads bytes of user data from a tag.
5	Start Continuous Read	Initiates continuous read mode.
6	Stop Continuous Read	Stops continuous read mode.
8	Teach Continuous Read	Provides the ability to automatically set the best time to start reading in continuous read mode.
10	Write SingleBlock	Writes a single block of user data.
11	Write Multiple Blocks	Writes multiple blocks of user data to a FRAM tag.
12	Multi-tag Block Write	Writes multiple blocks of user data to up to four tags.
13	Clear Multiple Bytes	Clears multiple bytes of user data in a tag.
14	Write Byte	Writes bytes of data to a tag.
20	Inventory	Counts the number of blocks in the field (up to four) and returns the UUID of the first tag in the field.
31	Read Transceiver Settings	Read Baud Rate, Device ID,Retry Time, and Gain.
33	Get Version Information	Retrieves the firmware version from the transceiver.
34	Get System Information	Gets Info Flags,UUID, DSFID, AFI,Memory Size, and IC Reference from Tag.
40	Lock Block	Locks blocks of memory.
41	Write AFI	Write the AFI byte to the tag.
42	Lock AFI	Locks the AFI byte from future changes.
43	Write DSFID	Writes the DSFID byte to the tag.
44	Lock DSFID	Locks the DSFID byte from future changes.
45	Get Multiple Block Security Status	Retrieves that security status of multiple blocks within a tag.

- **ContReadMode** – The channel ContReadMode bit will indicate the status of Continuous Read Mode for an RFID channel. A value of 0 indicates that the RFID channel is not in continuous read mode; a value of 1 indicates that the RFID channel is in continuous read mode. While in Continuous Read Mode, the interface will ignore all other commands except a Stop Continuous Read.
- **Counter** – The channel counter word is a 2-byte value that will increment its value by 1 after the interface has completed execution of a command. This value will roll over to 0 after it counts to 65535 and start again; at power up this value must be 0.
- **Data** – Depending on the Data Format, the channel Data word is an array of either 2-byte values or an array of 1-byte values that total 160 bytes in length. This array is used to store information returned from the RFID interface. Upon completion a command, reply data will be deposited in this array and the length of the reply (in 16-bit word increments) will be placed within the associated length field; at power up this value must be 0.
- **Fault** – The channel fault bit will indicate the fault status of the RFID channel. A value of 0 indicates the channel is operating normally, a value of 1 indicates the channel has faulted.
- **Length** – The channel length word is a 2-byte value that will indicate the data length for specific commands. Upon completion of a command, this word will be populated with the number of 16-bit words returned to the data field; at power up this value must be 0.

- **Reset** – The channel reset bit will indicate the reset status of the RFID channel. A value of 0 indicates the channel is not in reset, a value of 1 indicates the channel has completed a reset.
- **ResetInProgress** – The channel ResetInProg bit will indicate the status of an RFID channel reset. A value of 0 indicates that the RFID channel is not currently undergoing a reset; a value of 1 indicates a reset in progress on that channel.
- **TagPresent** – The channel TagPresent bit will indicate the status of a tag at the RFID channel. A value of 0 indicates there is not tag present at the transceiver; a value of 1 indicates one or more tags have been detected at the transceiver.

Output Image Table and Tags

Expand the RFID_1:O by clicking the "+" box. The shows the output image table, which has the following tags:



Controller Tags - RFID_Sample_Program_V1(controller)

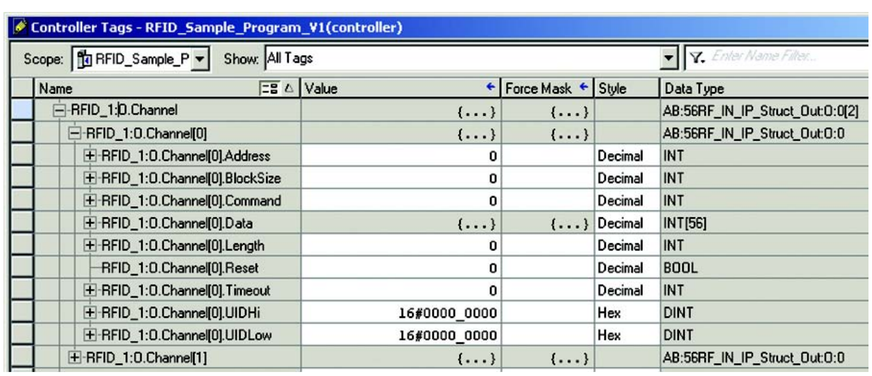
Scope: RFID_Sample_P Show: All Tags

Name	Value	Force Mask	Style	Data Type
RFID_1:O	{...}	{...}		AB:56RF_IN_IPD22:0:0
RFID_1:O.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0[2]
RFID_1:O.Pt00Data	0		Decimal	BOOL
RFID_1:O.Run	0		Decimal	BOOL

- **Channel** – See Output Channel Tags.
- **Pt00Data** – The Pt00Data bit is used to turn output point 0 either on or off. A value of 0 will be used to turn the output point off, a value of 1 will be used to turn the output point on.
- **Run** – The Run bit is used to place the RFID block into run or program mode. A value of 0 will be used for program mode, a value of 1 will be used for run mode. When in program mode, the interface will maintain the connection to the processor but will not execute commands. The discrete output point will follow the mode of the processor as well as the Run bit, with the Run bit overriding.

Output Channel Tags

Expand the RFID_1:Channel by clicking the "+" box. This shows that two channels exist: Channel[0] and Channel[1]. Expand the RFID_1:Channel[0] by clicking the "+" box. Each channel has the following tags:



Controller Tags - RFID_Sample_Program_V1(controller)

Scope: RFID_Sample_P Show: All Tags

Name	Value	Force Mask	Style	Data Type
RFID_1:O.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0[2]
RFID_1:O.Channel[0]	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0
RFID_1:O.Channel[0].Address	0		Decimal	INT
RFID_1:O.Channel[0].BlockSize	0		Decimal	INT
RFID_1:O.Channel[0].Command	0		Decimal	INT
RFID_1:O.Channel[0].Data	{...}	{...}	Decimal	INT[56]
RFID_1:O.Channel[0].Length	0		Decimal	INT
RFID_1:O.Channel[0].Reset	0		Decimal	BOOL
RFID_1:O.Channel[0].Timeout	0		Decimal	INT
RFID_1:O.Channel[0].UIDHi	16#0000_0000		Hex	DINT
RFID_1:O.Channel[0].UIDLow	16#0000_0000		Hex	DINT
RFID_1:O.Channel[1]	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0

- **Address** – The channel Address word is a 2-byte value that will contain the address or block value within the RFID tag that the command will execute on.
- **BlockSize** – The channel BlockSize word is a 2-byte value that will store the expected Block Size for the tag. Valid values are 0, 4, or 8 bytes per block. A value of 0 will default to a Block Size of 4 bytes per block.
- **Command** – The channel Command word is a 2-byte value that will store the next command for the interface to process. The RFID interface will execute the command once when this value changes. If a command must be repeated then set the value to zero first and then change it again to the desired command. Use a MOV or COP instruction to store the command value in this tag.

Value	Command	Description
1	Read Single Block	Reads a single block of user data.
2	Read Multiple Blocks	Reads multiple blocks of user data from a tag.
3	Multi-tag Block Read	Reads information from up to four tags.
4	Read Byte	Reads bytes of user data from a tag.
5	Start Continuous Read	Initiates continuous read mode
6	Stop Continuous Read	Stops continuous read mode
8	Teach Continuous Read	Provides the ability to automatically set the best time to start reading in continuous read mode.
10	Write SingleBlock	Writes a single block of user data.
11	Write Multiple Blocks	Writes multiple blocks of user data to a FRAM tag
12	Multi-tag Block Write	Writes multiple blocks of user data to up to four tags.
13	Clear Multiple Bytes	Clears multiple bytes of user data in a tag.
14	Write Byte	Writes bytes of data to a tag.
20	Inventory	Counts the number of blocks in the field (up to four) and returns the UUID of the first tag in the field.
31	Read Transceiver Settings	Read Baud Rate, Device ID and Retry Time.
33	Get Version Information	Retrieves the firmware version from the transceiver.
34	Get System Information	Gets Info Flags,UUID, DSFID, AFI,Memory Size and IC Reference from Tag
41	Write AFI	Write the AFI byte to the tag
42	Lock AFI	Locks the AFI byte from future changes.
43	Write DSFID	Writes the DSFID byte to the tag.
44	Lock DSFID	Locks the DSFID byte from future changes.
45	Get Multiple Block Security Status	Retrieves that security status of multiple blocks within a tag.

- **Data** – Depending on the Data Format, the channel Data word is either an array of 2-byte values or an array of 1-byte values that total 112 bytes in length per channel. This array is used to store information that will be directed to the RFID interface. Some commands, such as reading, do not require the use of this data field. Writing to tags will utilize this information in conjunction with the length field to inform the RFID interface what values it needs to write. The size of this word allows the writing of up to 28 blocks of data to a tag at a time, with each block being 4-bytes in length.
- **Length** – The channel length word is a 2-byte value that will indicate the data length for specific commands. Upon completion of a command, this word will be populated with the number of 16-bit words returned to the data field; at power up this value must be 0.

- **Reset** – The channel reset bit is used to command an RFID channel reset. A value of 0 indicates the channel is not being commanded to reset, a value of 1 indicates a request to reset the channel.
- **Timeout** – This value determines how long the interface will wait for a command response from the transceiver before indicating a message time out. The default value is 0, which sets the timeout at 750 ms. The user can enter a timeout value in milliseconds, keeping in mind that a low timeout value can cause command failures by timing out before the command would otherwise have successfully completed.
- **UIDHi** – The channel UID word is an 8-byte value that will contain the UUID information for specific commands that will allow the command to be targeted to a specific tag in the field. Under normal circumstances this value will be 0, which tells the RFID interface to perform an action regardless of what tag it is. Any value other than 0 will attempt to direct the command to that specific tag. The UIDHi value contains bytes 0…1 and 6…7 of the UID.
- **UIDLow** – The UIDLow value contains bytes 2-5 of the UID.

Commands Summary

This section provide a summary of the commands supported by the RFID transceiver. Detail of the commands can be found in Chapter 9. This guide assumes familiarity with RSLogix 5000. The *.ACD file should already be downloaded into the PLC and working properly.

The table below assumes the following:

- The user has set up the RSLogix5000 AOP with Data Format set to SINT.
- The RFID tag has blocks that are only 4 bytes each.
- The UUID is set to zero (unless specified).

TIP A Universally Unique Identifier (UUID) can be specified in xx.0.Channel[0].UIDLow and xx.0.Channel[0].UIDHi for most commands to operate on a specific tag. If xx.0.Channel[0].UIDLow and xx.0.Channel[0].UIDHi are set to 0, the command will operate on the first tag in the transceiver field. All other Output values should be set to 0 where not specified.

Command	Description	Output xx.0.Channel[0]	Input xx.I.Channel[0]
Inventory	Option Flag 0 Returns number of tags in field Returns Universally Unique Identifier (UUID) of first tag in field	Command = 20 Length = 0 Data[0] = 0	Data[0] = # of tags Data[2…9, 10…17, 18…25, 26…33] = UUID of up to 4 tags
	Option Flag 1 Returns number of tags in field Returns Application Family Identifier (AFI) of first tag in field Returns Universally Unique Identifier (UUID) of first tag in field	Command = 20 Length = 1 Data[0] = 1	Data[0] = # of tags Data[2, 12, 22, 32] = AFI of up to 4 tags Data[4…11, 13…21, 24…31, 34…41] = UUID of up to 4 tags
Read Single Block	Option Flag 0 Reads a single block of user data from a tag	Command = 1 Data[0] = 0	Data[0…3] = User data (4 bytes)
	Option Flag 1 Reads a single block of user data from a tag Returns security status of the block	Command = 1 Data[0] = 1	Data[0…3] = User data (4 bytes) Data[4] = Security status
Write Single Block	Writes a single block of user data to a tag	Command = 10 Length =Block size BlockSize = Block size Data[0-1] = User data (4 bytes)	All data bytes are zero
Lock Block	Locks a single block of user data, preventing writing	Command = 40 UIDLow = UIDLow UIDHi = UIDHi	All data bytes are zero
Read Multiple Blocks	Option Flag 0 Reads multiple blocks of user data from a tag	Command = 2 Length = Number of blocks Data[0] = 0	Data[0…3] = Block x Data[4…7] = Block x+1
	Option Flag 1 Reads multiple blocks of user data from a tag Returns security status of the blocks	Command = 2 Length = Number of blocks Data[0] = 1	Data[0…3] = Block x Data[4] = Security status of block x Data[6…9] = Block x+1 Data[10] = Security status of block x+1

Write Multiple Blocks	Writes multiple blocks of user data to an FRAM tag	Command = 11 Length = Number of bytes (multiple of 8) BlockSize = Block size Data[0-3] = User data (8 bytes)	All data bytes are zero
Write AFI	Writes 1 byte of information into the Application Family Identifier (AFI) area contained within block -2	Command = 41 Length = 1 Data[0] = 00xx	All data bytes are zero
Lock AFI	Locks the 1 byte of information for the AFI area, preventing it from being modified	Command = 42 UIDLow = UIDLow UIDHi = UIDHi	All data bytes are zero
Write DSFID	Writes 1 byte of information in the DSFID area	Command = 43 Length = 1 Data[0] = 00xx	All data bytes are zero
Lock DSFID	Locks the 1 byte of information for the DSFID area, preventing it from being modified	Command = 44 UIDLow = UIDLow UIDHi = UIDHi Data[0] = 00xx	All data bytes are zero
Get System Information	Returns the following system information of the tag: Info_Flags UUID DSFID AFI Memory Size (Max Block Number +1 * Max Byte per Block +1) IC Reference	Command = 34	Data[0] = Info_Flag Data[2] = DSFID Data[4] = AFI Data[6-13] = UUID Data[14] = Max Block Number Data[15] = Max Byte Number in Block Data[16] = IC Ref
Get Multiple Block Security Status	Retrieves the security status of multiple blocks within a tag	Command = 45 Length = Number of blocks	Data[0…7] = UUID Data[8] = Security status of block x Data[10] = Security status of block x+1
Read Byte	Option Flag 0 Reads bytes of user data from a tag	Command = 4 Address = Starting byte Length = Number of bytes to read Data[0] = 0	Data[0…] = User data
	Option Flag 1 Reads the UUID from a tag Reads bytes of user data from a tag	Command = 4 Address = Starting byte Length = Number of bytes to read Data[0] = 1	Data[0…7] = UUID Data[8…] = User data
Write Byte	Writes bytes of user data to a tag	Command = 14 Address = Starting byte Length = Number of bytes to write Data[0] = Start of User data	Data[0…7] = UUID
Clear Multiple Bytes	Clears multiple bytes of user data in a tag	Command = 13 Address = Starting byte Length = Number of bytes to clear Data[0] = Cleared byte value	All data bytes are the cleared byte value
Multi-Tag Block Read	Reads the following information from up to 4 tags in the field: Number of tags UUID Multiple blocks of user data	Command = 3 Address = First block to read Length = Number of blocks to read for each tag	Data[0] = Number of tags Data[2…9] = UUID of 1st tag Data[10…*] = User data of 1st tag Data[*…*] = UUID of 2nd tag Data[*…*] = User data of 2nd tag
Multi-Tag Block Write	Writes multiple blocks of user data to up to 4 tags in the field Returns number of tags in the field Retrieves UUID of tags	Command = 12 Length = Number of bytes to write to each tag BlockSize = Block size Data[0] = Block x Data[4…7] = Block x+1	Data[0] = Number of tags Data[2…9] = UUID of 1st tag Data[10…17] = UUID of 2nd tag Data[18…25] = UUID of 3rd tag Data[26…33] = UUID of 4th tag
Read Transceiver Settings	Retrieves the following information from the transceiver: Baud rate Device ID Retry time	Command = 31	Data[0…1] = Device ID Data[2…5] = Baud rate Data[6…7] = Retry setting Data[8…9] = Gain
Get Version Information	Retrieves the firmware version from the transceiver	Command = 33	Data = Firmware version

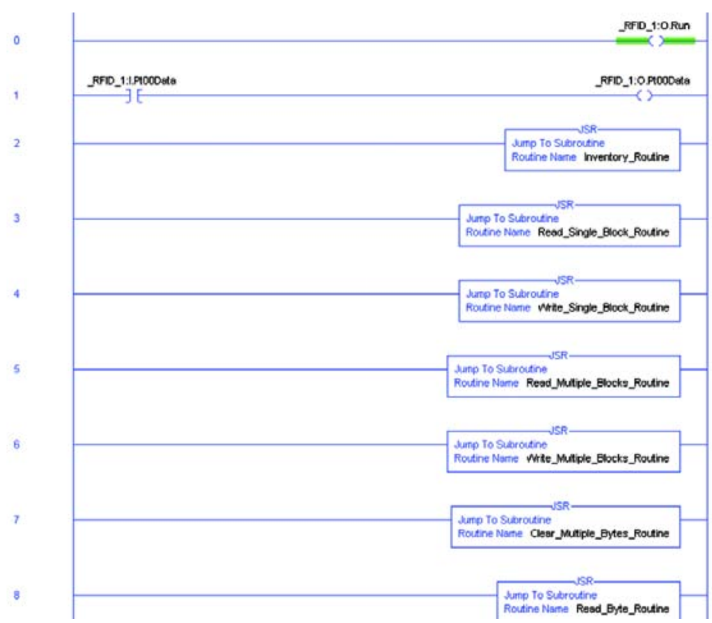
RSLogix 5000 Code Examples

This chapter contains examples of routines that will run in RSLogix 5000.

The examples are written for an RF transceiver connected to the “0” connector on the RF interface block. A momentary switch is connected to the Digital Input connector. The switch is used to enable the routine to allow the user to repeat the routine easily.

In the examples, the RFID block is identified as “_RFID1”

Main Routine



A partial listing of the Main Routine is shown below. The Main Routine sets the run bit. In program mode, the run bit is 0; and 1 for run mode. The remaining blocks jump to the various subroutines to execute the commands. In Rung 1, the momentary switch turns on Digital Output 0, which turns on an LED to confirm that the user has pressed the momentary switch.

Example Command Routines - Overview

Many of the example routines (not the Main Routine) use the same ladder logic. The ladder logic is explained below.

Rung 0

Rung 0 initiates the routine. A sensor or momentary switch, connected to the input connection of the RFID interface block, senses that an object (with an RFID tag attached) is approaching and enables the execution of the read routine. The sensor is the XIC bit labeled _RFID_1:I:Pt00Data. When the sensor detects the object, the instruction latches ON.

Rung 1

Rung 1 initializes the output image table in preparation for command. Execution begins when the transceiver is not already busy reading a tag and a tag is present in the RF field.

This Examine If Closed (XIC) instruction is latched ON by the sensor in Rung 0.

RFID_1:I:Channel[0]Busy – This Examine If Open (XIO) instruction prevents the rung from executing when the transceiver is busy executing a command.

RFID_1:I:Channel[0].TagPresent – This XIC instruction closes when a tag is present in the RF field of the transceiver connected to Channel[0].

MOV variable to RFID_1:O:Channel[0]:variable – Moves data from a Controller tag to the output image table variable.

MOV 0 to RFID_1:0:Channel[0].Command – This initializes the output command to 0.

IMPORTANT	The transceiver executes a command when the command value changes. When repeating a command, set the command value to 0 first and then re-set it to the same desired value.

Start – Latches a tag that indicates the function has started.

Unlatch – This unlatches (turns OFF) the instruction from Rung 0 and readies the routine for the next RFID tag.

Rung 2

Start – With the output channel properly initialized, the Start bit enables the rung to begin execution.

EQU RFID_1:I:Command[0].Command =0 – When an output command is updated, the interface block returns that command back to the input command. If the input command is zero (it was set in Rung 1), then the EQU output goes HI and enables the subsequent MOV command.

MOV x to RFID_1:O:Command[0].Command – Moving a non-zero value into the output command byte instructs the RFID block to execute the command.

Rung 3

Rung 3 ensures that another command is not initiated while a command is busy.

Start – The Start bit enables the rung to begin execution.

RFID_1:I:Channel[0].Busy – When the command begins execution, the Busy bit goes HI. This contact closes and the rung is executed.

InProgress – When command begins execution, an In-Progress bit is latched ON.

Start – This contact is opened, as the command has transitioned from start to busy.

Rung 4

Rung 4 confirms the completion of the command, as the interface block moves a value into the input channel command location.

InProgress – This contact closes when the read command begins execution.

RFID_1:I:Channel[0].Busy – This contact will be open while the command is in process.

EQU RFID_1:I:Channel[0].Command = Upon completion of the command the interface block copies the value from output command to the input command. If the input command value equals the value of the command, the EQU output goes HI.

InProgress – This bit is unlatched when the command is successfully completed. The routine is now ready for the next RFID tag or other routine.

Clear Multiple Bytes

The Clear Multiple Bytes command clears multiple bytes of user data in a RFID tag. The user can specify the number of bytes to clear and the address from which to begin. This is very similar to a "copy" command. It copies the value you specify in the output data image Data[0] location to the addresses you specify.

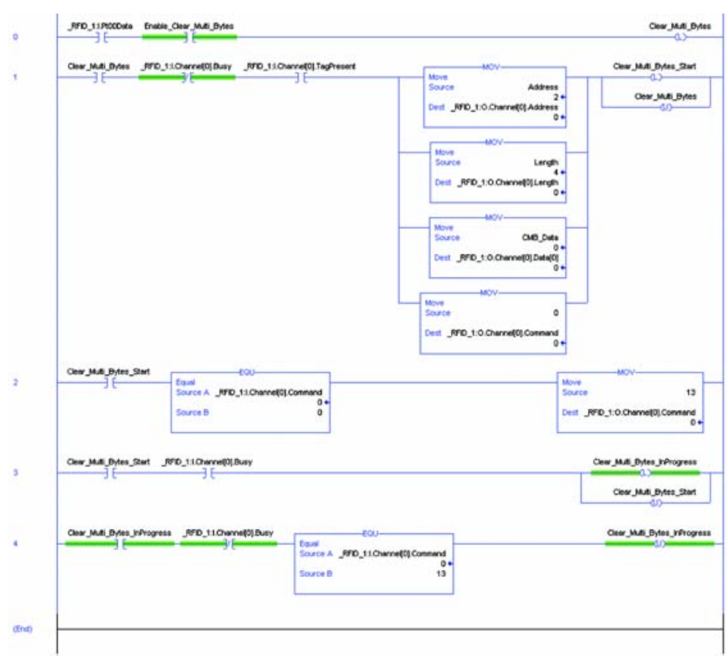
Set the following values in the output image table:

a. xx:O.Channel[0].Command = 13
b. xx:O.Channel[0].Address = starting address
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0 (or value used to clear the byte)
e. xx:O.Channel[0].Length = the number of bytes to clear
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

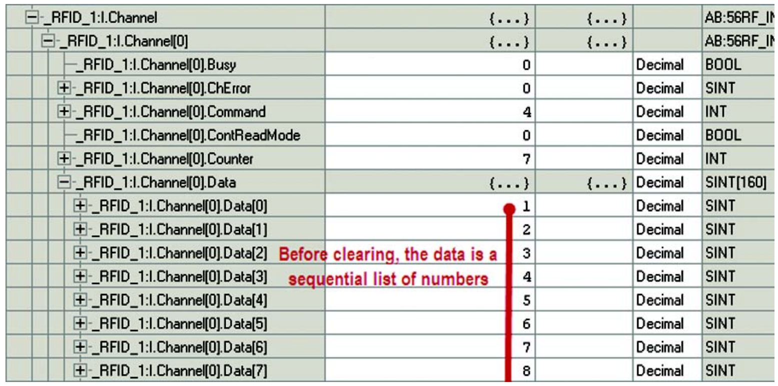
Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length data, the value used to clear the fields and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table. The value to be copied is initially stored in the controller tag CMB_Data. In the example below, CMB_Data is set to 0, but the user can set this to be any valid SINT value.



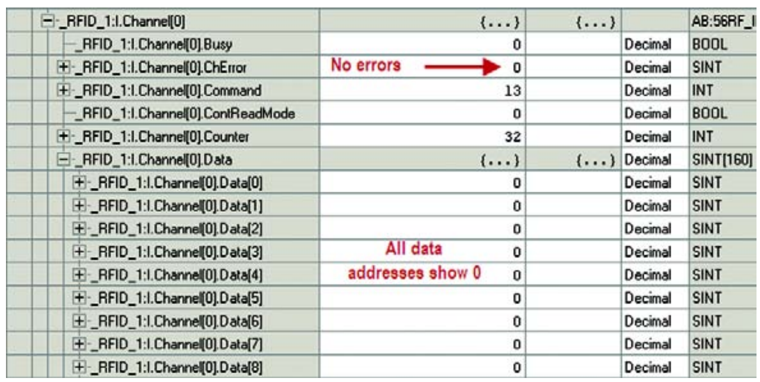
Example Results

To demonstrate the results, the Read Byte command was executed on an RFID tag. The data in this tag was a simple list of numbers starting from 1. Note the counter is 31.



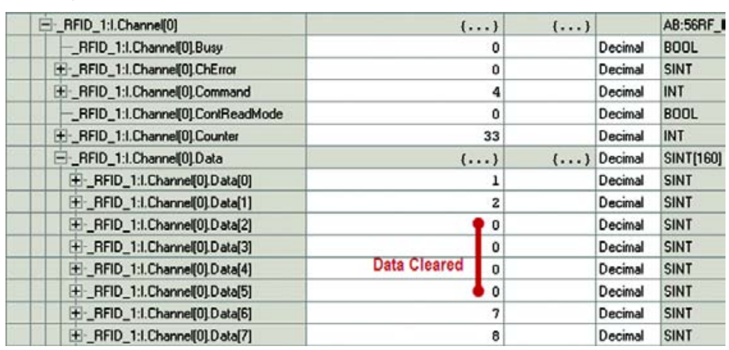
_RFID_1:I.Channel	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	4		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	7		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	1		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	2		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	3		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	5		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	6		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	7		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	8		Decimal	SINT

The Clear Multiple Byte command is executed successfully as the ChError = 0 and all the data bytes are zero. The counter increments to 32.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	13		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	32		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[8]	0		Decimal	SINT

The tag is read again (command = 4) to confirm the clearing. Data bytes 2 through 4 are successfully set to 0.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	4		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	33		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	1		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	2		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	7		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	8		Decimal	SINT

Get Multiple Block Security Status

The Get Multiple Block Security Status command retrieves the security status of multiple blocks within a tag. It will also display the Universally Unique Identifier (UUID) of the RFID tag.

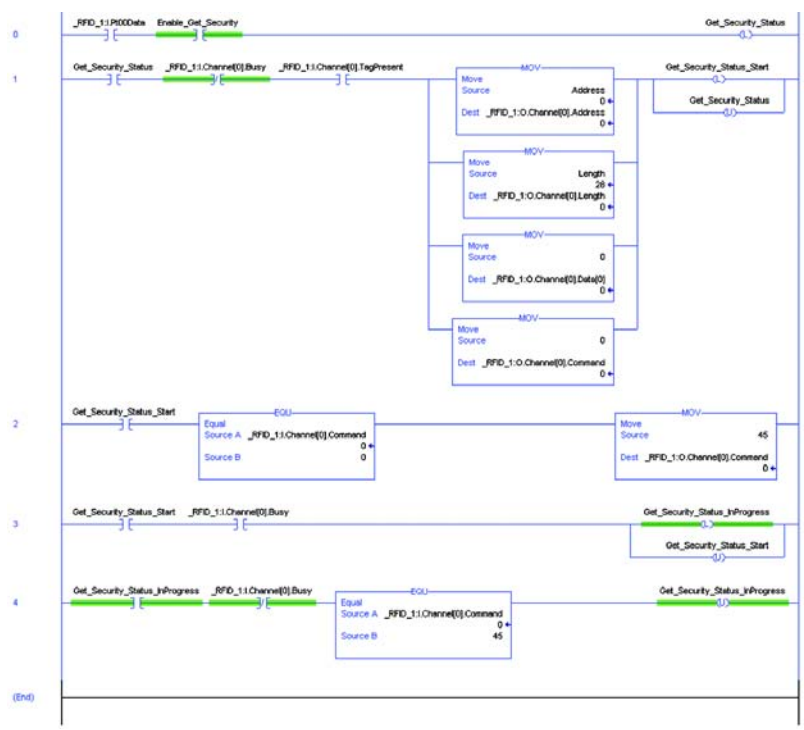
Set the following values in the output image table:

a. xx:O.Channel[0].Command = 45
b. xx:O.Channel[0].Address = the first block to read
c. xx:O.Channel[0].Block = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = the number of blocks to read.
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

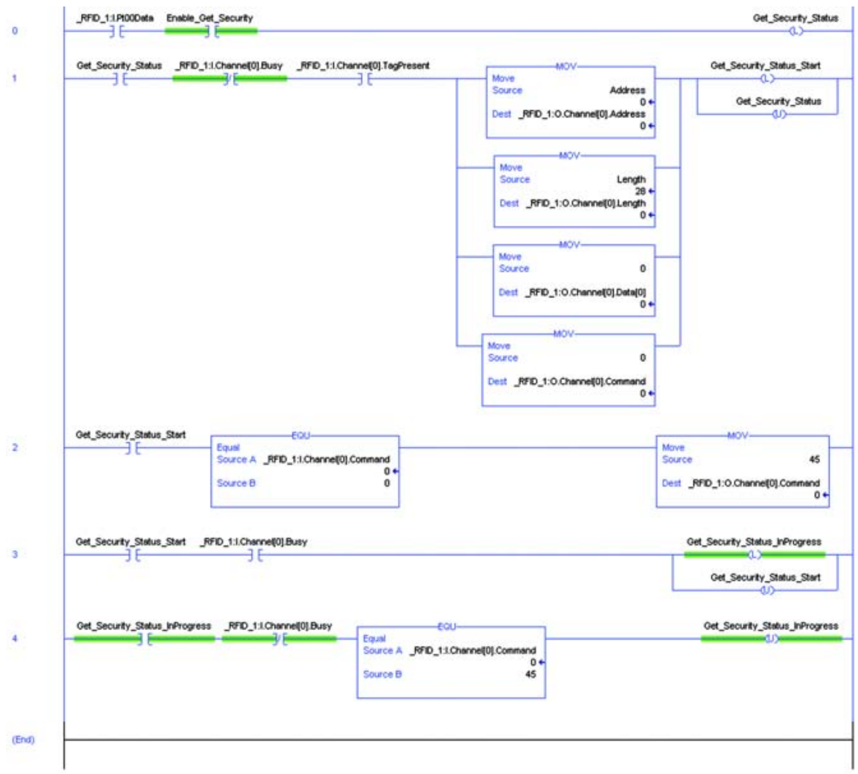
Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length data, the Data[0] value used to clear the fields and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table. The starting address is block 0. The command reads 28 blocks (all the blocks of this RFID tag).



Example Results

The figure below shows the security status for the first three blocks. Blocks 0 and 2 are locked. Block 1 is not locked.



The following information will be displayed:

- xx:I.Channel[0].Data[0-7] = UUID
- xx:I.Channel[0].Data[8-9] = Security status of block x
- xx:I.Channel[0].Data[10-11] = Security status of block x+1

Get System Information

The Get System Information command will return the following RFID tag information:

- Info_Flag
- Data Storage Format Identifier (DSFID)
- Application Family Identifier (AFI)
- Universally Unique Identifier (UUID)
- Memory Size
- IC Reference

Set the following values in the output image table:

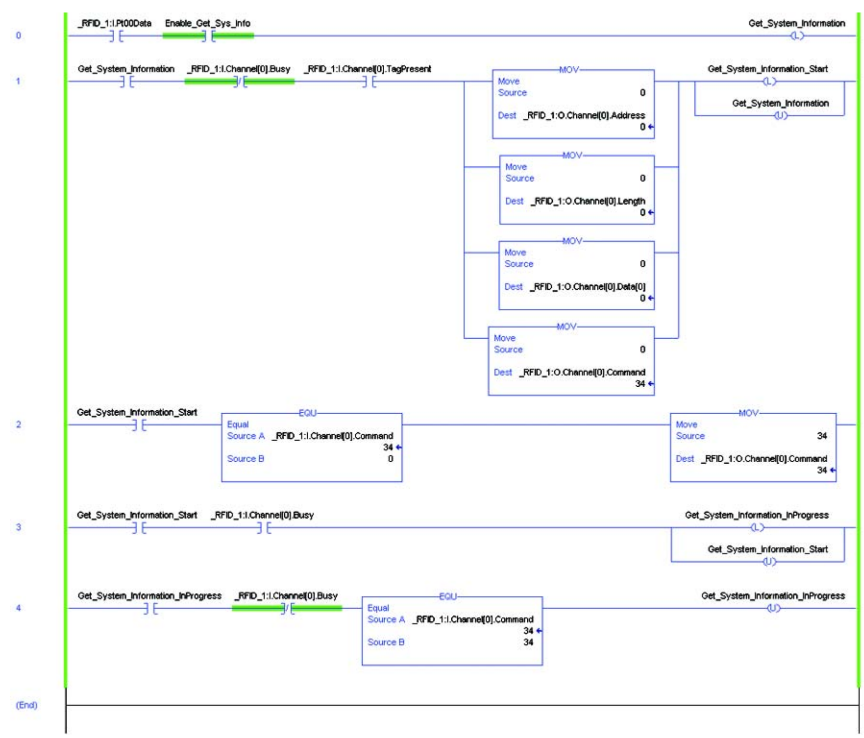
a. xx:O.Channel[0].Command = 34
b. xx:O.Channel[0].Address = 0
c. xx:O.Channel[0].BlockSize = 0

d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length data, the Data[0] value used to clear the fields and sets the command value to 0. Because the address, length and data[0] can only be 0, the source in the MOV instruction can be set to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table.



Example Results

The Info Flag contains data used to determine what parameters are passed back.

The DSFID, AFI and UUID follow.

The tag being read was Cat. No. 56RRF-TG-30. This tag has 28 blocks. The maximum block number is 27, as the first block is 0. Each block has 4 bytes. The maximum byte number is 3, as the first byte is 0.

The IC Ref is the last byte reported.

_RFID_1:I.Channel[0]		{...}	{...}		AB:56RF_
_RFID_1:I.Channel[0].Busy		0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError		0		Decimal	SINT
_RFID_1:I.Channel[0].Command		34		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode		0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter		142		Decimal	INT
_RFID_1:I.Channel[0].Data		{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	Info Flag	15		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	DSFID	68		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	AFI	58		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]		16#e9		Hex	SINT
_RFID_1:I.Channel[0].Data[7]	UID Low	16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[8]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[9]		16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[10]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[11]	UID Hi	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[12]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[13]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[14]	Max Block Number	27		Decimal	SINT
_RFID_1:I.Channel[0].Data[15]	Max Byte Number in Block	3		Decimal	SINT
_RFID_1:I.Channel[0].Data[16]	IC Ref	1		Decimal	SINT
_RFID_1:I.Channel[0].Data[17]		0		Decimal	SINT

Get Version Information

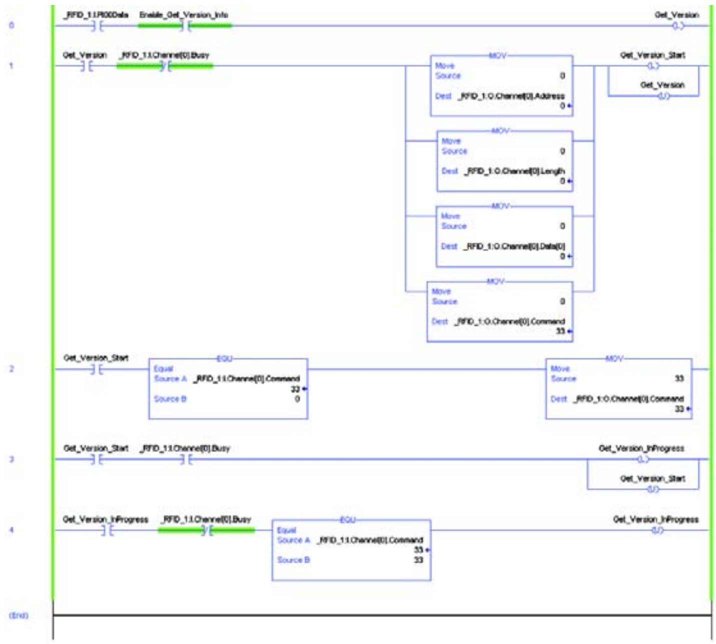
The Get Version Information command will retrieve the firmware version information from the transceiver.

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 33
b. xx:O.Channel[0].Address = 0
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0
i. xx:O.Channel[0].UIDHi = 0

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length data, the Data[0] value used to clear the fields and sets the command value to 0. Because the address, length and data[0] can only be 0, the source in the MOV instruction can be set to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table.



Example Results

The results are stored in Data [0…3]. In this example, the version is “de20007”. (version 2.07).

⊟-_RFID_1:I	{...}	{...}		AB:56RF_IN_
_RFID_1:I.AuxPwrFault	0		Decimal	BOOL
_RFID_1:I.BlockFault	0		Decimal	BOOL
⊟-_RFID_1:I.Channel	{...}	{...}		AB:56RF_IN_
⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN_
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	33		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	81		Decimal	INT
⊟-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞-_RFID_1:I.Channel[0].Data[0]	16#07		Hex	SINT
⊞-_RFID_1:I.Channel[0].Data[1]	16#00		Hex	SINT
⊞-_RFID_1:I.Channel[0].Data[2]	16#e2		Hex	SINT
⊞-_RFID_1:I.Channel[0].Data[3]	16#0d		Hex	SINT
⊞-_RFID_1:I.Channel[0].Data[4]	0		Decimal	SINT

Inventory

The inventory command returns the UUID and DSFID information from the RFID tags in the field. This command can read up to a maximum of four tags. The more tags in the field, the more time the tags need to be in the field to complete the inventory command. By setting the output image fields to specific values, the Inventory command returns the following information:

1. Returns the number of tags in the field and the UUID of each tag. Set Address =0, Length = 0 and Data[0] = 0
2. Returns the number of tags in the field, the UUID and the DSFID of each tag. Set Address =0, Length = 1 and Data[0] = 0
3. Returns the number of tags in the field, the UUID and the DSFID of each tag that meets the specified AFI. Set Address =1, Length = 1 and Data[0] = AFI value. If the AFI value is 0, then all the tags are reported.

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 20
b. xx:O.Channel[0].Address = 0 (or 1, see below)
c. xx:O.Channel[0].Block = 0
d. xx:O.Channel[0].Data[0] = 0 (or 1, see below)
e. xx:O.Channel[0].Length = 0 (or 1, see below)
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0
i. xx:O.Channel[0].UIDHi = 0

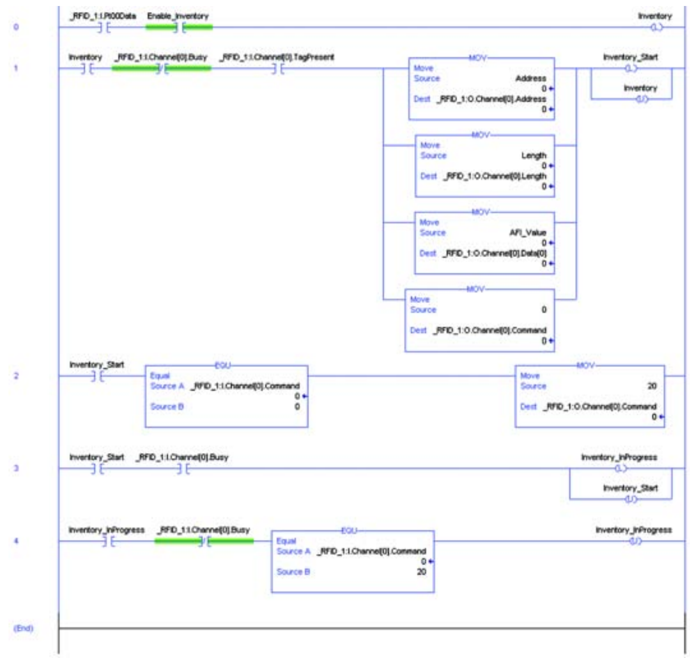
Notes:

1. Set Address = 0 to get all tags in the RF field.
2. Set Address = 1 to get all tags that have the AFI value specified in the Data[0] location.
3. Set Length = 0 to get only the UUID for each tag.
4. Set Length = 1 to get both the UUID and the DSFID for each tag.
5. Set Data[0] = 0 to return all tags in the RF field.
6. Set Data[0] = AFI value (but not zero) to return only those tags that have that AFI value

Example Routine

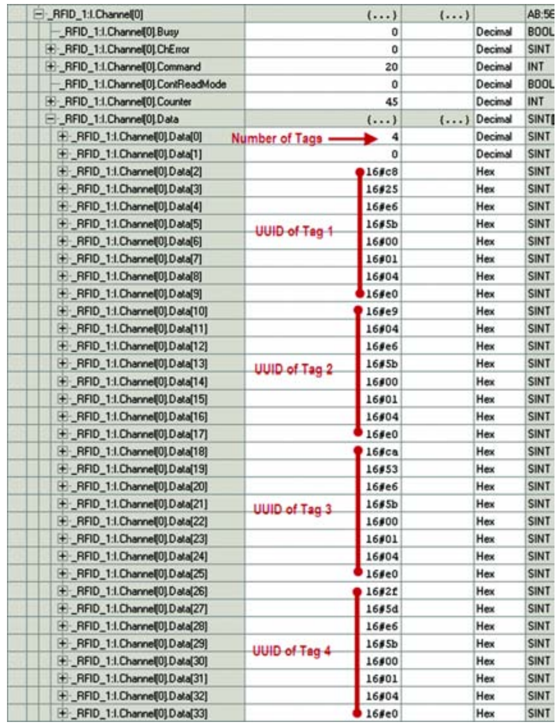
In the example routine below, the initialization in Rung 1 sets the address, length data, the Data[0] value used to clear the fields and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table.

The example ladder diagram is initially set for Address =0, Length = 0 and Data[0] = 0. These values are then changed to get obtain example results for the three versions of the Inventory command.



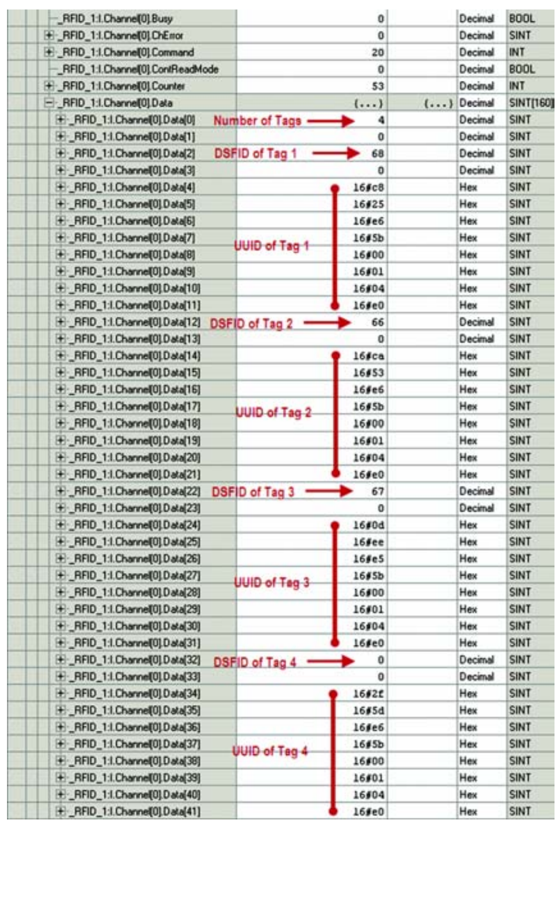
Example Results

In example 1, the Address = 0, Length = 0 and Data[0] = 0. Four RFID tags were in the RF field at the time the read command was executed. The controller tag values are shown below. The data shows the number of tags in the RF field and the UUID for each tag.



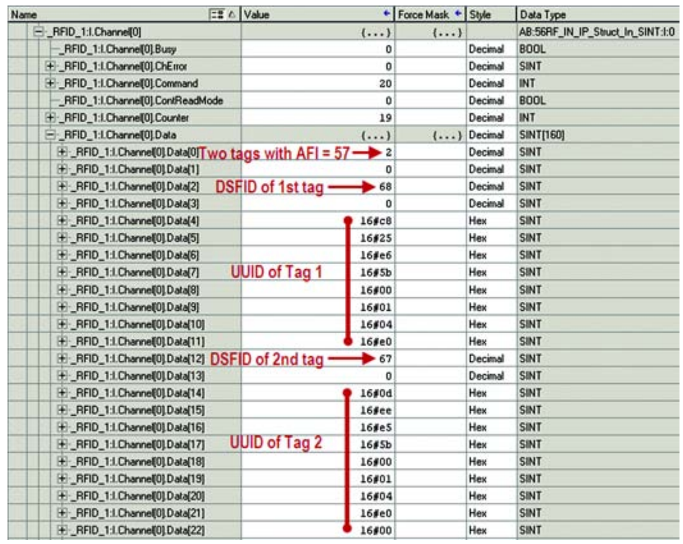
_RFID_1:I.Channel[0]		(...)	(...)		AB:5E
_RFID_1:I.Channel[0].Busy		0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError		0		Decimal	SINT
_RFID_1:I.Channel[0].Command		20		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode		0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter		45		Decimal	INT
_RFID_1:I.Channel[0].Data		(...)	(...)	Decimal	SINT[
_RFID_1:I.Channel[0].Data[0]	Number of Tags	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]		16#c8		Hex	SINT
_RFID_1:I.Channel[0].Data[3]		16#25		Hex	SINT
_RFID_1:I.Channel[0].Data[4]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[5]	UUID of Tag 1	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[6]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[7]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[8]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[9]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[10]		16#e9		Hex	SINT
_RFID_1:I.Channel[0].Data[11]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[12]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[13]	UUID of Tag 2	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[14]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[15]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[16]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[17]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[18]		16#ca		Hex	SINT
_RFID_1:I.Channel[0].Data[19]		16#53		Hex	SINT
_RFID_1:I.Channel[0].Data[20]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[21]	UUID of Tag 3	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[22]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[23]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[24]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[25]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[26]		16#2f		Hex	SINT
_RFID_1:I.Channel[0].Data[27]		16#5d		Hex	SINT
_RFID_1:I.Channel[0].Data[28]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[29]	UUID of Tag 4	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[30]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[31]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[32]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[33]		16#e0		Hex	SINT

In example 2, the length was changed to 1. the Address = 0, Length = 1 and Data[0] = 0. Four RFID tags were in the RF field at the time the read command was executed. The controller tag values are shown below. The data shows the number of tags in the RF field, the DSFID and the UUID for each tag.



_RFID_1:I.Channel[0].Busy		0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError		0		Decimal	SINT
_RFID_1:I.Channel[0].Command		20		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode		0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter		53		Decimal	INT
_RFID_1:I.Channel[0].Data		(...)	(...)	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	Number of Tags	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	DSFID of Tag 1	68		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]		16#c8		Hex	SINT
_RFID_1:I.Channel[0].Data[5]		16#25		Hex	SINT
_RFID_1:I.Channel[0].Data[6]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[7]	UUID of Tag 1	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[8]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[9]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[10]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[11]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[12]	DSFID of Tag 2	66		Decimal	SINT
_RFID_1:I.Channel[0].Data[13]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[14]		16#ca		Hex	SINT
_RFID_1:I.Channel[0].Data[15]		16#53		Hex	SINT
_RFID_1:I.Channel[0].Data[16]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[17]	UUID of Tag 2	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[18]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[19]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[20]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[21]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[22]	DSFID of Tag 3	67		Decimal	SINT
_RFID_1:I.Channel[0].Data[23]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[24]		16#0d		Hex	SINT
_RFID_1:I.Channel[0].Data[25]		16#ee		Hex	SINT
_RFID_1:I.Channel[0].Data[26]		16#e5		Hex	SINT
_RFID_1:I.Channel[0].Data[27]	UUID of Tag 3	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[28]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[29]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[30]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[31]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[32]	DSFID of Tag 4	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[33]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[34]		16#2f		Hex	SINT
_RFID_1:I.Channel[0].Data[35]		16#5d		Hex	SINT
_RFID_1:I.Channel[0].Data[36]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[37]	UUID of Tag 4	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[38]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[39]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[40]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[41]		16#e0		Hex	SINT

In example 3, we get the tag information for only those tags the have a specific AFI. In this example the AFI is 57. Address = 1, Length = 1 and Data[0] = 57. Two of the four RFID tags that were present in the RF field at the time the read command was executed had AFI set to 57. The controller tag values are shown below. The data shows the number of tags in the RF field, the DSFID and the UUID for each of these tags.



Lock AFI

The Lock AFI command will lock the one byte of information for the Application Family Identifier (AFI), preventing it from being modified in the future. **Once the AFI byte is locked, it cannot be unlocked.**

The AFI is used to group RFID tags by application. This allows the transceiver to send out an AFI and target only the tags that meet the application criteria.

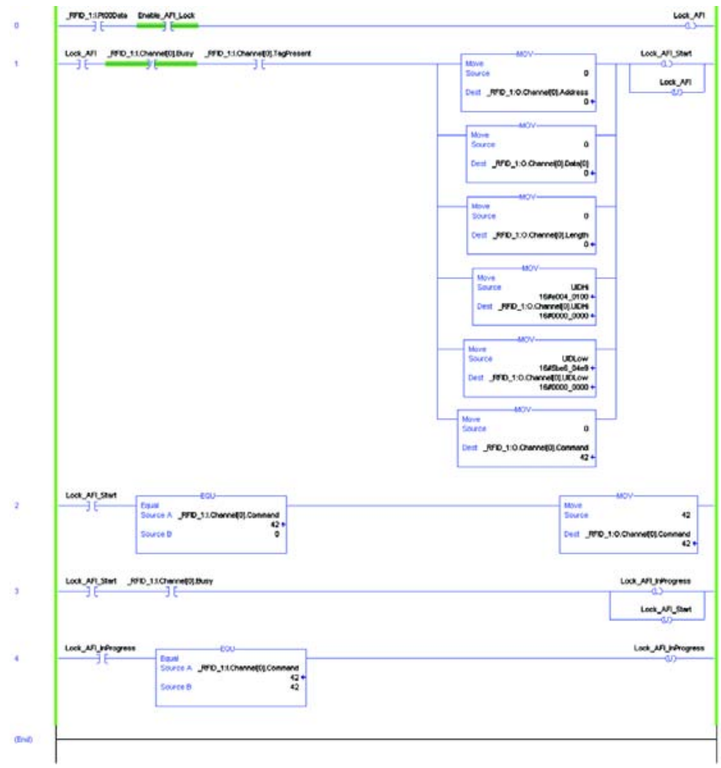
Set the following values in the output image table:

a. xx:O.Channel[0].Command = 42
b. xx:O.Channel[0].Address = 0
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = UIDLow
i. xx:O.Channel[0].UIDHi = UIDHi

The UIDLow and UIDHi bytes must be specified to lock the AFI value. The UUID can be found by performing the Inventory command.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, the Data[0, UIDLow and UIDHi values used to lock the AFI and sets the command value to 0. The BlockSize, Reset, and Timeout are set to 0 in the output image table.



Example Results

The following figure shows an example of results on the input image table. The Command is showing 42 and the ChError is showing 0. The input data bytes are all zero.

-_RFID_1:I.Channel	{...}	{...}		AB:56RF_IN_IP_
-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN_IP_
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
+_RFID_1:I.Channel[0].ChError	No error → 0		Decimal	SINT
+_RFID_1:I.Channel[0].Command	Command = 42 → 42		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
+_RFID_1:I.Channel[0].Counter	56		Decimal	INT
-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
+_RFID_1:I.Channel[0].Data[0]	0		Decimal	SINT
+_RFID_1:I.Channel[0].Data[1]	0		Decimal	SINT
+_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT
+_RFID_1:I.Channel[0].Data[3]	0		Decimal	SINT

Errors

The following ChErrors will be generated:

- 0 – AFI was successfully locked.
- 4 – A tag with the wrong UUID entered the RF field.
- 8 – A tag that has already been locked entered the RF field.

Lock Block

The Lock Block command locks one block of user data, preventing future writing. **Once the block is locked, the block cannot be unlocked**. The transceiver automatically determines the block size of the RFID tag.

Set the following values in the output image table:

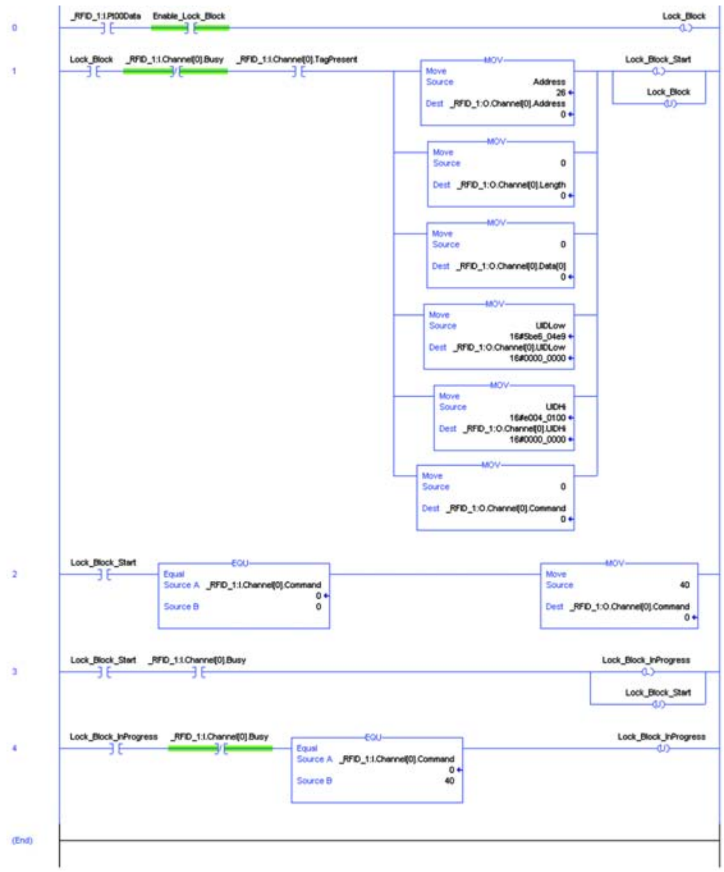
a. xx:O.Channel[0].Command = 40
b. xx:O.Channel[0].Address = the number of the block to lock
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = UIDLow
i. xx:O.Channel[0].UIDHi = UIDHi

The UIDLow and UIDHi bytes must be specified to lock the block values. The UUID can be found by performing the Inventory command.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, the Data[0], UIDLow and UIDHi values used to lock the block and sets the command value to 0. The BlockSize, Reset, and Timeout are set to 0 in the output image table.

In the example routine, rung 1 initializes the output image table. The UUID is stored in a controllers tags UIDLow and UIDHi. The block that is being locked is block 26. This tag has a total of 27 blocks.



Example Results

The output image table shows address 26. This is the second to last block of the Cat. No. 56RF-TG-30 tag. The command is 40. The UUID must be specified to lock any blocks.

⊟-_RFID_1:O.Channel[0]	{...}	{...}		AB:56RF_I
⊞-_RFID_1:O.Channel[0].Address	Block to be locked → 26		Decimal	INT
⊞-_RFID_1:O.Channel[0].BlockSize	0		Decimal	INT
⊞-_RFID_1:O.Channel[0].Command	Command = 40 → 40		Decimal	INT
⊞-_RFID_1:O.Channel[0].Data	{...}	{...}	Decimal	SINT[112]
⊞-_RFID_1:O.Channel[0].Length	0		Decimal	INT
-_RFID_1:O.Channel[0].Reset	0		Decimal	BOOL
⊞-_RFID_1:O.Channel[0].Timeout	0		Decimal	INT
⊞-_RFID_1:O.Channel[0].UIDHi	UUID → 16#e004_0100		Hex	DINT
⊞-_RFID_1:O.Channel[0].UIDLow	16#5be6_04e9		Hex	DINT

After completion of the lock block command, the input image table should show the command is 40 and the ChError is 0.

⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_
-_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	No error → 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	Command = 40 → 40		Decimal	INT
-_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	12		Decimal	INT
⊞-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
-_RFID_1:I.Channel[0].Fault	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Length	0		Decimal	INT
-_RFID_1:I.Channel[0].Reset	0		Decimal	BOOL
-_RFID_1:I.Channel[0].ResetInProgress	0		Decimal	BOOL
-_RFID_1:I.Channel[0].TagPresent	0		Decimal	BOOL

Errors

The ChErrorfield will show will be 8 if you try to lock a block that is already locked.

Lock DSFID

The Lock DSFID command will lock the 1 byte of information for the Data Storage Format Identifier (DSFID) area of the tag, preventing it from being modified.**Once the DSFID byte is locked, it cannot be unlocked.**

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 44

b. xx:O.Channel[0].Address = 0

c. xx:O.Channel[0].Data[0] = 0

d. xx:O.Channel[0].Length = 0

e. xx:O.Channel[0].Reset = 0

f. xx:O.Channel[0].Timeout = 0

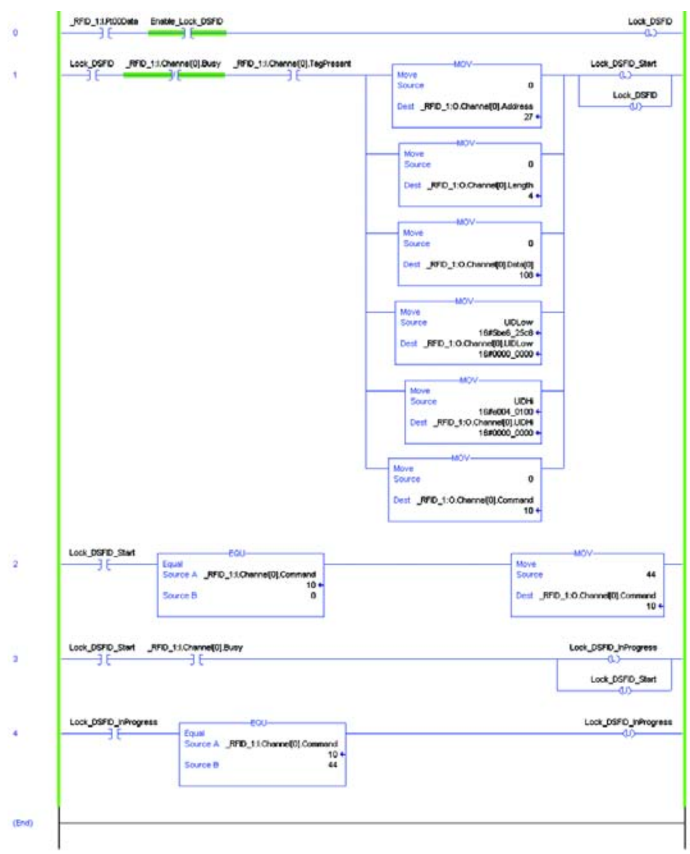
g. xx:O.Channel[0].UIDLow = UIDLow

h. xx:O.Channel[0].UIDHi = UIDHi

The UIDLow and UIDHi bytes must be specified to lock the DSFID value. The UUID can be found by performing the Inventory command.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, the Data[0], UIDLow and UIDHi values used to lock the DSFID and sets the command value to 0. The BlockSize, Reset, and Timeout are set to 0 in the output image table.



Example Results

When successful, the results shown in the input image table show ChError = 0 and the Command number =44.

If you try to lock the DSFID on an RFID tag already locked, the ChError will be equal to 8.

⊟-_RFID_1:I.Channel	{...}	{...}		AB:56RF_I
⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	No error 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	Command = 44 44		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	9		Decimal	INT
⊞-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Fault	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Length	0		Decimal	INT
_RFID_1:I.Channel[0].Reset	0		Decimal	BOOL
_RFID_1:I.Channel[0].ResetInProgress	0		Decimal	BOOL
_RFID_1:I.Channel[0].TagPresent	0		Decimal	BOOL

Read Byte Command

The Read Byte command reads a user-specified number of bytes from a tag, starting at a user-specified address. An Option Flag can be set to return the UUID of the tag. The maximum number of bytes that can be read at a time is 160 bytes using option flag 0, and 152 bytes using option flag 1.

- **Option Flag 0** – Returns the specified user data. Set xx:O.Channel[0].Data[0] = 0.
- **Option Flag 1** – Returns the UUID of the RFID tag and the specified user data. Set xx:O.Channel[0].Data[0] = 1.

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 4
b. xx:O.Channel[0].Address = starting address to read
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = Option Flag
e. xx:O.Channel[0].Length = the number of bytes to read
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0
i. xx:O.Channel[0].UIDHi = 0

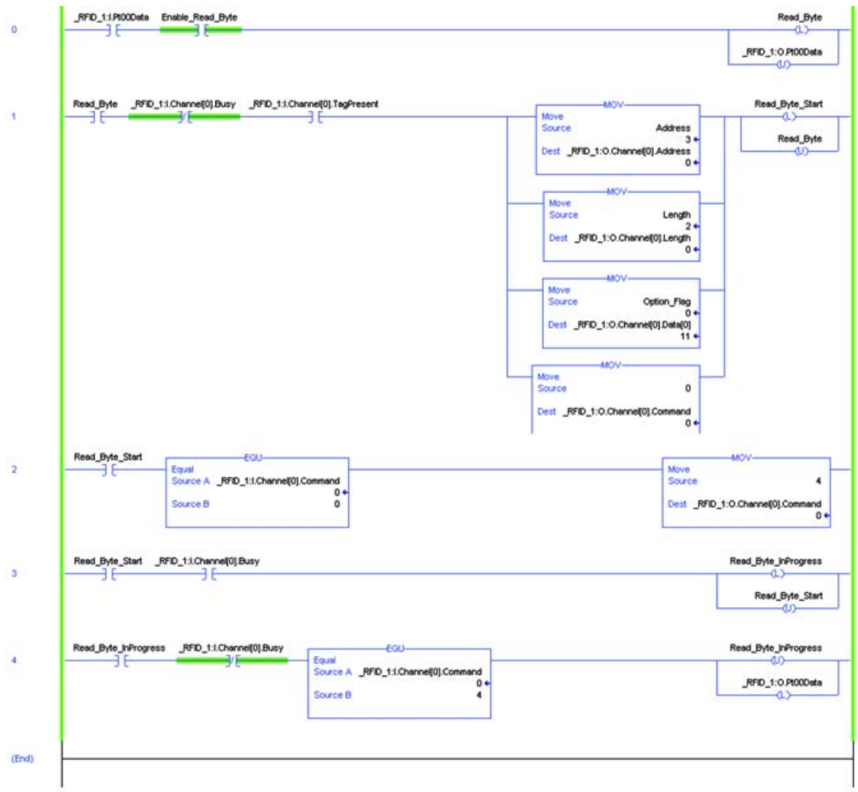
This command operates only on the first tag in the field.

Note that Data[1] must also be set to 0.

Example Routine

Below is example routine to read all the data and the UUID in a Cat. No. 56RF-TG-30 ICODE tag. This tag holds a maximum of 112 bytes of data.

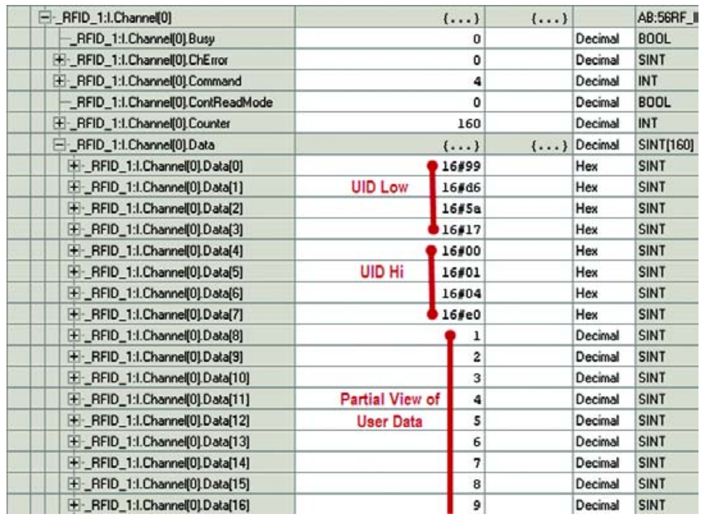
In the example routine below, the initialization in Rung 1 sets the address, length, the Data[0]to the Option Flag, and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.



Example Results

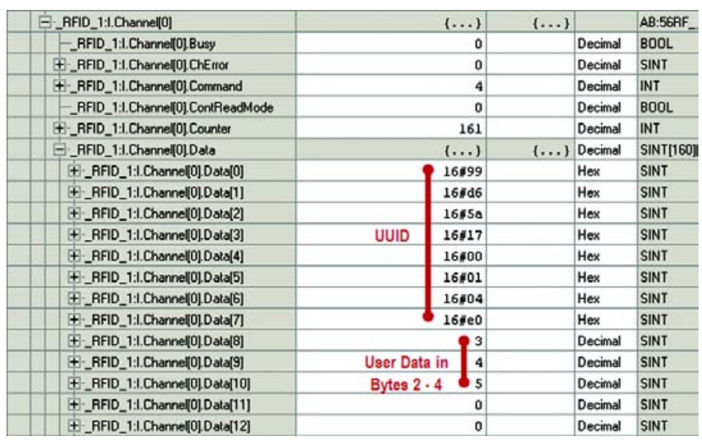
The following figure shows an example of results where the Option Flag was set to 1, which reads the UUID.

The UUID is loaded into Data[0] through Data[7]. The user data (1, 2, 3, 4, 5, 6...) begins in Data[8]. The figure below only shows a partial listing of the user data. The command read in 112 bytes of data.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	4		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	160		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	16#99		Hex	SINT
_RFID_1:I.Channel[0].Data[1]	16#d6		Hex	SINT
_RFID_1:I.Channel[0].Data[2]	16#5a		Hex	SINT
_RFID_1:I.Channel[0].Data[3]	16#17		Hex	SINT
_RFID_1:I.Channel[0].Data[4]	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[5]	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[6]	16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[7]	16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[8]	1		Decimal	SINT
_RFID_1:I.Channel[0].Data[9]	2		Decimal	SINT
_RFID_1:I.Channel[0].Data[10]	3		Decimal	SINT
_RFID_1:I.Channel[0].Data[11]	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[12]	5		Decimal	SINT
_RFID_1:I.Channel[0].Data[13]	6		Decimal	SINT
_RFID_1:I.Channel[0].Data[14]	7		Decimal	SINT
_RFID_1:I.Channel[0].Data[15]	8		Decimal	SINT
_RFID_1:I.Channel[0].Data[16]	9		Decimal	SINT

In the figure below, the command was repeated with the Starting Address set to 2 and the number of bytes set to 3.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	4		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	161		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	16#99		Hex	SINT
_RFID_1:I.Channel[0].Data[1]	16#d6		Hex	SINT
_RFID_1:I.Channel[0].Data[2]	16#5a		Hex	SINT
_RFID_1:I.Channel[0].Data[3]	16#17		Hex	SINT
_RFID_1:I.Channel[0].Data[4]	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[5]	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[6]	16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[7]	16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[8]	3		Decimal	SINT
_RFID_1:I.Channel[0].Data[9]	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[10]	5		Decimal	SINT
_RFID_1:I.Channel[0].Data[11]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[12]	0		Decimal	SINT

Multi-Tag Block Read

The Multi-Tag Block Read command reads multiple blocks of user data from multiple tags in the RF field. The transceiver automatically determines the block size. All RFID tags in the field should have the same block size.

This command can read up to four tags. Adequate time must be allowed to read all the tags in the RF field.

Set the following values in the output image table:

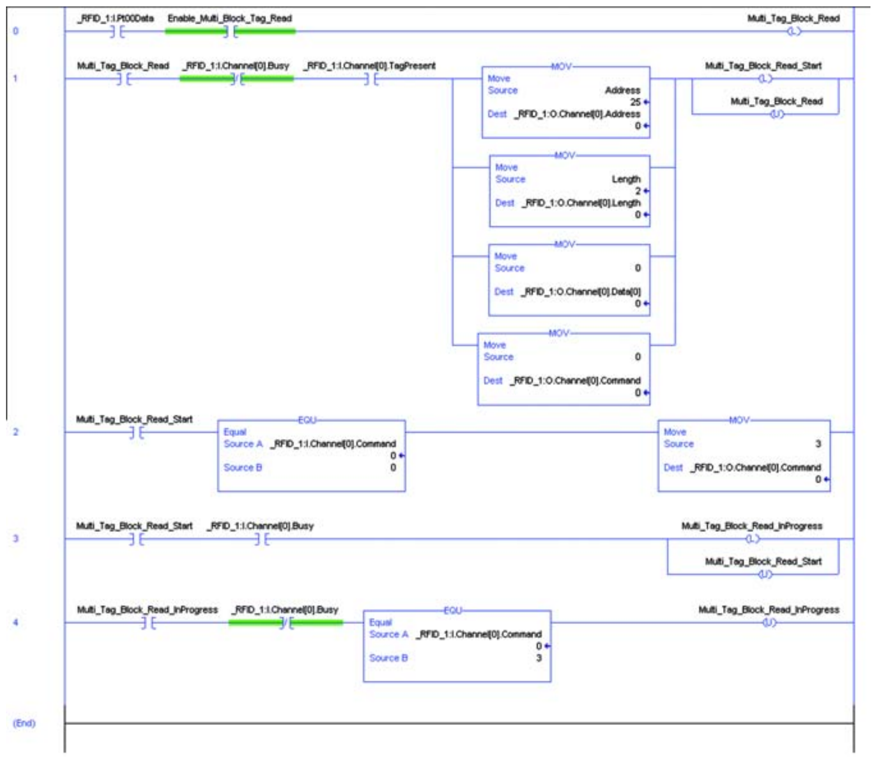
a. xx:O.Channel[0].Command = 3
b. xx:O.Channel[0].Address = the first block to read
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = the number of blocks to read
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first four tags in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, the Data[0] value used to read multiple tags and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.

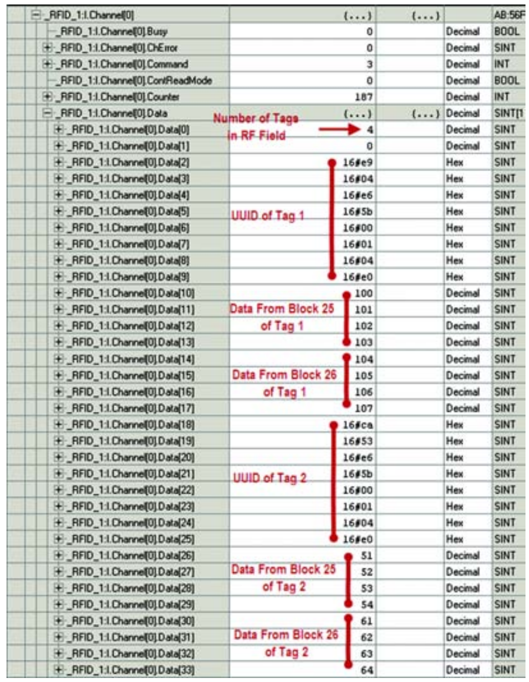
The example ladder diagram is initially set for Address =25 and the Length = 2. The command will read blocks 25 and 26.



Example Results

The input image data fields are populated with the number of tags, followed by the UUID and block data of each tag.

In the example below, four Cat. No. 56RF-TG-30 RFID tags were read. These tags hold 4 bytes per block. Since two blocks (25 and 26) were read, a total of eight data fields are used to store the user data. The figure only shows the information from two of the four RFID tags.



Read Multiple Blocks

The Read Multiple Blocks command reads multiple blocks of user data from an RFID tag. Option Flags can be set to return just the data in the blocks or return the data and the security status for each block of data. The maximum number of blocks that can be read at one time is 10.

- **Option Flag 0** – Returns multiple blocks of user data. Set xx:O.Channel[0].Data[0] = 0.
- **Option Flag 1** – Returns multiple blocks of user data and the security status of each block. Set xx:O.Channel[0].Data[0] = 1.

Set the following values in the output image table:

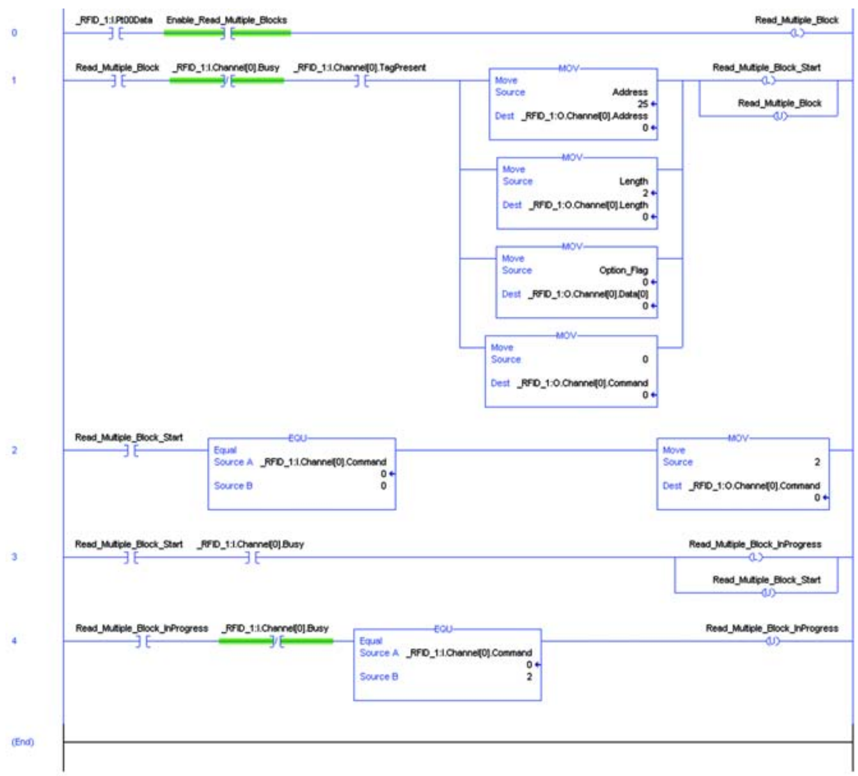
a. xx:O.Channel[0].Command = 2
b. xx:O.Channel[0].Address = the first block to read
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = the Option Flag
e. xx:O.Channel[0].Length = the number of blocks to read
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

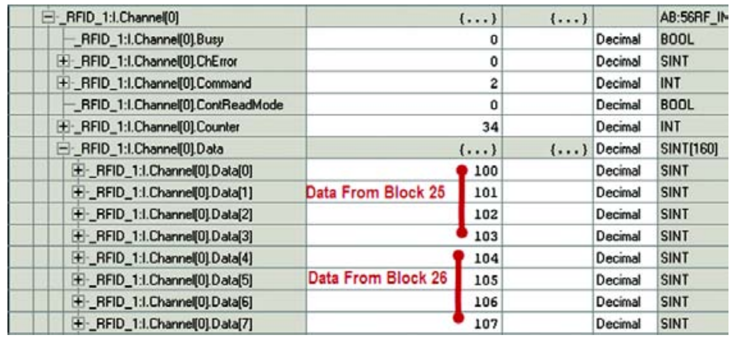
In the example routine below, the initialization in Rung 1 sets the address, length, and Data[0] values used to read multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table.

The example ladder diagram is initially set for Address =25, the Length = 2. Data[0] is set to Option Flag 0 (return just the data) The command will read blocks 25 and 26. The example is repeated with Option Flag set to 1.



Example Results

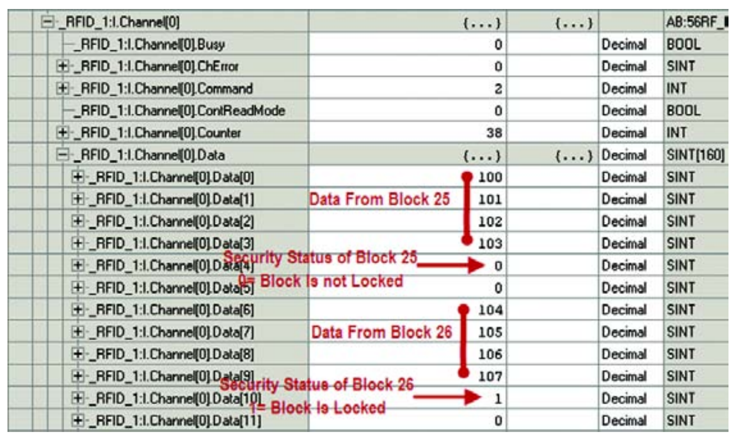
This first example uses Option Flag = 0; return only the data in the blocks. With a starting block number of 25 and two blocks to read, data from Blocks 25 and 26 are returned. The tag was a Cat. No. 56RF-TG-30 which has only 4 bytes per block. The data appears in the input channel Data[0…7].



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	2		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	34		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	100		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	101		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	102		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	103		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	104		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	105		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	106		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	107		Decimal	SINT

This second example shows the results for Option Flag = 1; return the data and the security status. With a starting block number of 25 and two blocks to read, data from Blocks 25 and 26 are returned. The tag was a Cat. No. 56RF-TG-30 which has only 4 bytes per block.

The data for the first block appears in the input channel Data[0…3]. The security status appears in Data[4]. The value of 0 indicates that the block is not locked.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	2		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	38		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	100		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	101		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	102		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	103		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	104		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	105		Decimal	SINT
_RFID_1:I.Channel[0].Data[8]	106		Decimal	SINT
_RFID_1:I.Channel[0].Data[9]	107		Decimal	SINT
_RFID_1:I.Channel[0].Data[10]	1		Decimal	SINT
_RFID_1:I.Channel[0].Data[11]	0		Decimal	SINT

The data for the second block appears in the input channel Data[6…9]. The security status appears in Data[10]. The value of 1 indicates that the block is locked.

Read Single Block

The Read Single Block command reads a single block of user data from a tag. Option Flags can be set to return information the UUID and security status of the block.

- **Option Flag 0** – Returns a single block of user data.
  Set xx:O.Channel[0].Data[0] = 0.
- **Option Flag 1** – Returns a single block of user data and the security status of that block. Set xx:O.Channel[0].Data[0] = 1.

Set the following values in the output image table:

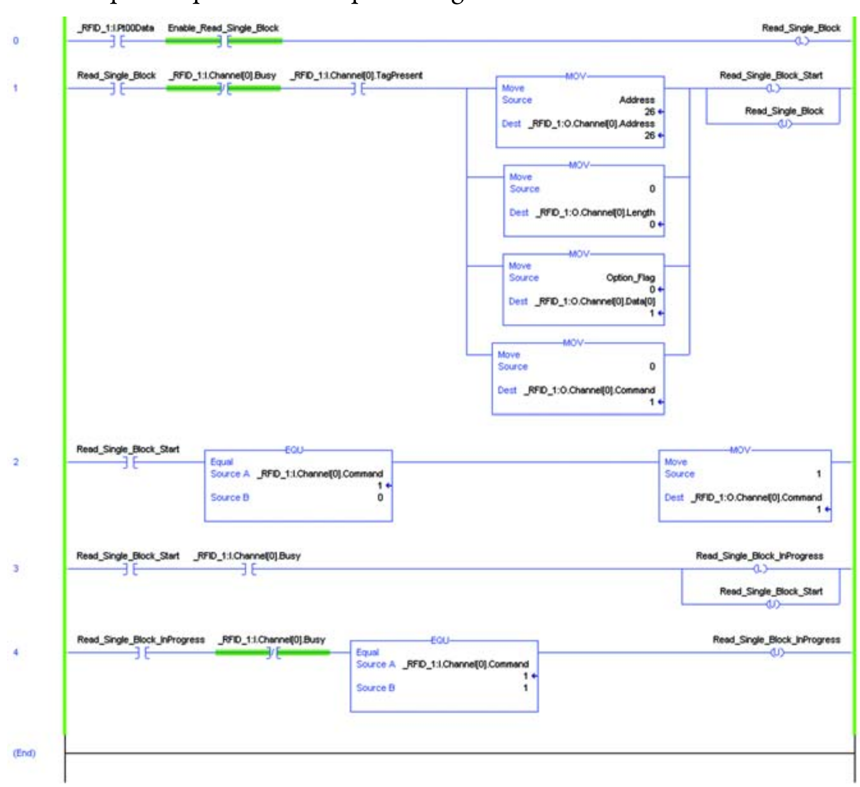
a. xx:O.Channel[0].Command = 1
b. xx:O.Channel[0].Address = the block number to read.
c. xx.O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = the Option Flag value
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

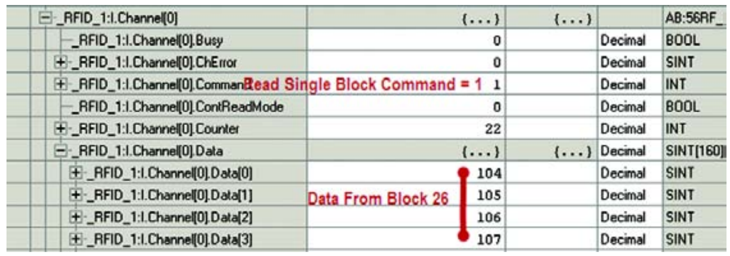
In the example routine below, the initialization in Rung 1 sets the address, length, and Data[0] values used to read multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.

The example ladder diagram is initially set for Address =26. Data[0] is set to Option Flag 0 (return just the data) The command will read blocks 25 and 26. The example is repeated with Option Flag set to 1.



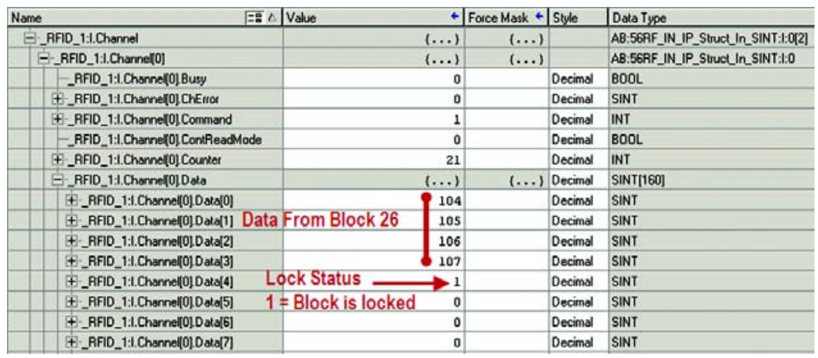
Example Results

- **Option Flag 0** —This first example uses Option Flag = 0; return only the data in the block. The block number is 26. The tag was a Cat. No. 56RF-TG-30 which has only 4 bytes per block. The data appears in the input channel Data[0...3].



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Comman	1		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	22		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	104		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	105		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	106		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	107		Decimal	SINT

- **Option Flag 1**—The second example demonstrates the results when Option Flag = 1. Data[0] shows the security status of the block. The 1 indicates the block has been locked. A zero indicates the block is unlocked. The data appears in Data[1...4].



Name	Value	Force Mask	Style	Data Type
_RFID_1:I.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_In_SINT:I:0[2]
_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN_IP_Struct_In_SINT:I:0
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	1		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	21		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	104		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	105		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	106		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	107		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	1		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	0		Decimal	SINT

Read Transceiver Settings

The Read Transceiver Settings command will retrieve the following information from the transceiver:

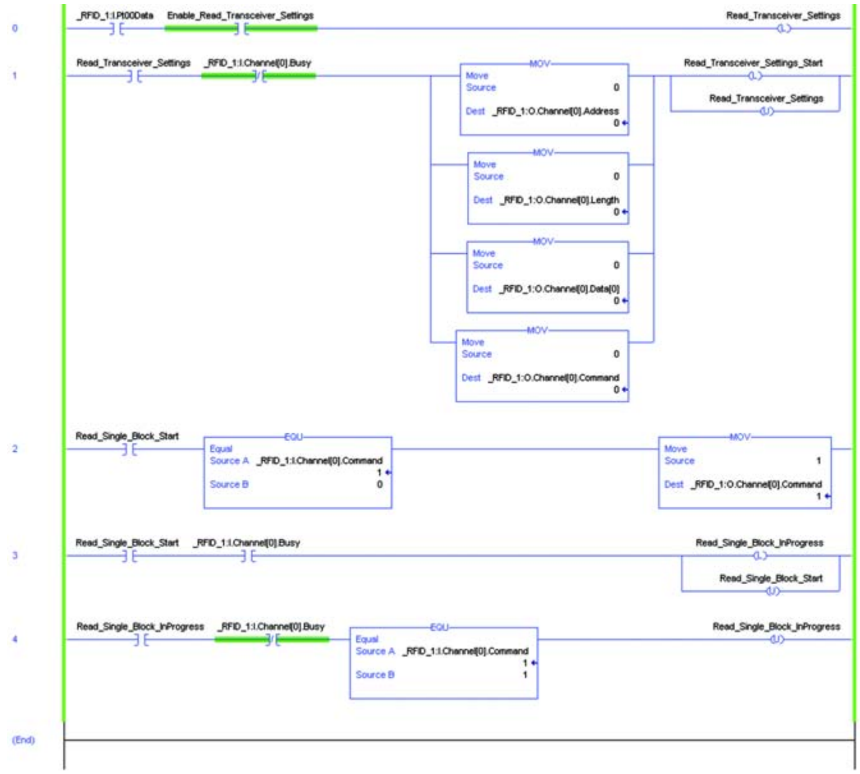
- Device ID
- Baud rate
- Retry time
- Gain

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 31
b. xx:O.Channel[0].Address = 0
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0
i. xx:O.Channel[0].UIDHi = 0

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, data, and command. Because the address, length and Data[0] can only be 0, the source in the MOV instruction can be set to 0. The UIDLow, UIDHi, BlockSize, Reset and Timeout are set to 0 in the output image table.



Example Results

The following information will be displayed:

- xx:I.Channel[0].Data[0…1] = Device ID
- xx:I.Channel[0].Data[2…5] = Baud rate
- xx:I.Channel[0].Data[6…7] = Retry setting
- xx:I.Channel[0].Data[8…9] = Gain

Gain is 0…3, with 0 being the highest gain.

_RFID_1:I.Channel		{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0]		{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy		0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError		0		Decimal	SINT
_RFID_1:I.Channel[0].Command		31		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode		0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter		189		Decimal	INT
_RFID_1:I.Channel[0].Data		{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	Device ID	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[1]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[2]	Baud Rate	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[3]		16#96		Hex	SINT
_RFID_1:I.Channel[0].Data[4]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[5]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[6]	Retry Times	16#03		Hex	SINT
_RFID_1:I.Channel[0].Data[7]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[8]	Gain	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[9]		16#00		Hex	SINT

Write AFI

The Write AFI command writes one byte of information into the Application Family Identifier (AFI).The AFI is used to group RFID tags by application. This allows the transceiver to read and write only to those tags with the specified AFI value.

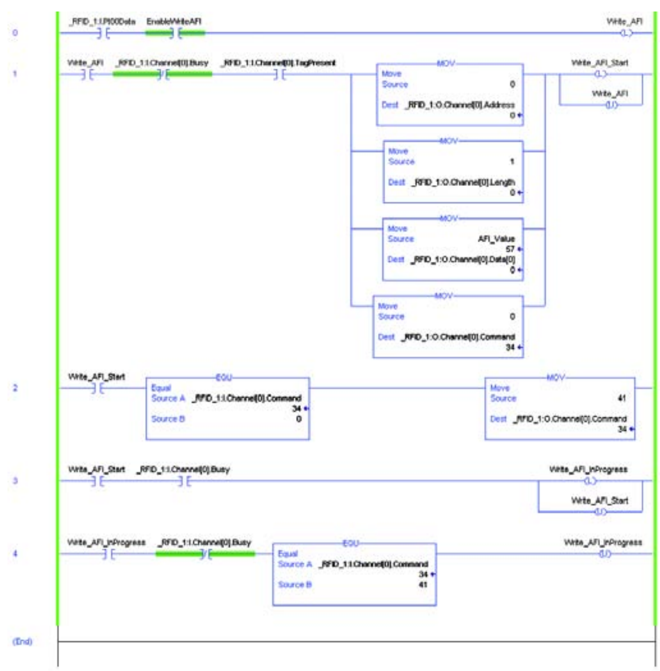
Set the following values in the output image table:

1. xx:O.Channel[0].Command = 41
2. xx:O.Channel[0].Address = 0
3. xx:O.Channel[0].BlockSize = 0
4. xx:O.Channel[0].Data[0] = AFI value
5. xx:O.Channel[0].Length = 1
6. xx:O.Channel[0].Reset = 0
7. xx:O.Channel[0].Timeout = 0
8. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
9. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

In the following example routine, the initialization sets the address, length data and command. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.



Example Results

The following figure shows an example of results on the input image table. The Command is showing 41 and theChError is showing 0. The data bytes are all zero. Confirmation that the AFI was written can be observed in the Get_System_Information_Routine.

⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN_
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	41		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	30		Decimal	INT
⊟-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞-_RFID_1:I.Channel[0].Data[0]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[1]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[3]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[4]	0		Decimal	SINT

Write Byte Command

The Write Byte command writes bytes of user data to a tag. The user must specify the data, the start byte, and the number of bytes to write.

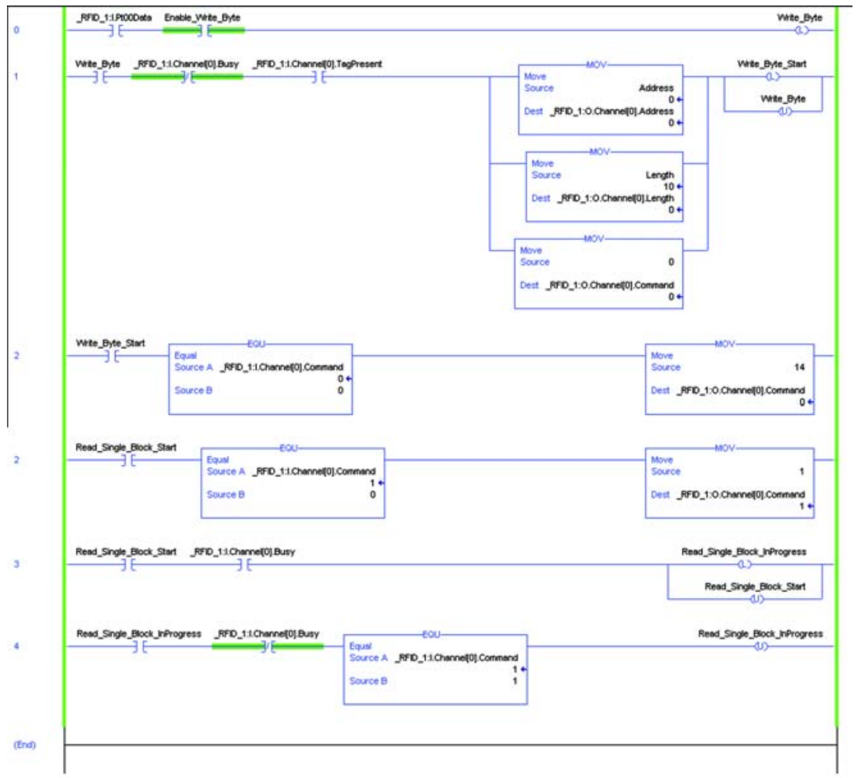
a. xx:O.Channel[0].Command = 14
b. xx:O.Channel[0].Address = starting addressto write
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0…111] = the data to write
e. xx:O.Channel[0].Length = the number of bytes to write
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, and Data[0] values used to read multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow and UIDHi are set to 0 in the output image table.

The example ladder diagram is initially set for Address =0, the Length = 10. Data[0…9] are set to a sequential list of numbers starting with 11.

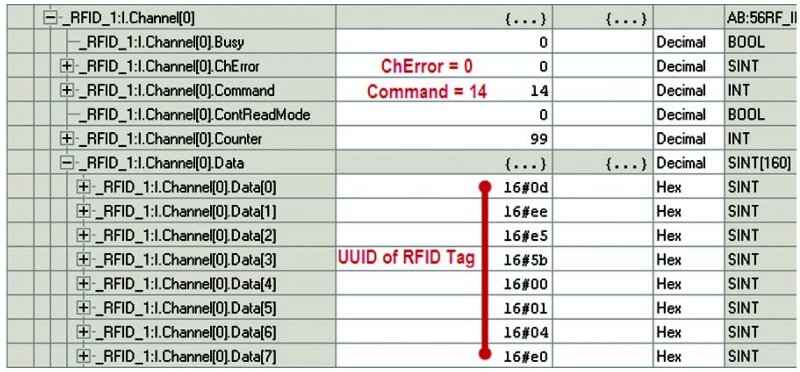


Example Results

The figure below shows the output image table with the 10 bytes of data that will be written to the RFID tag. The sequence is 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20.

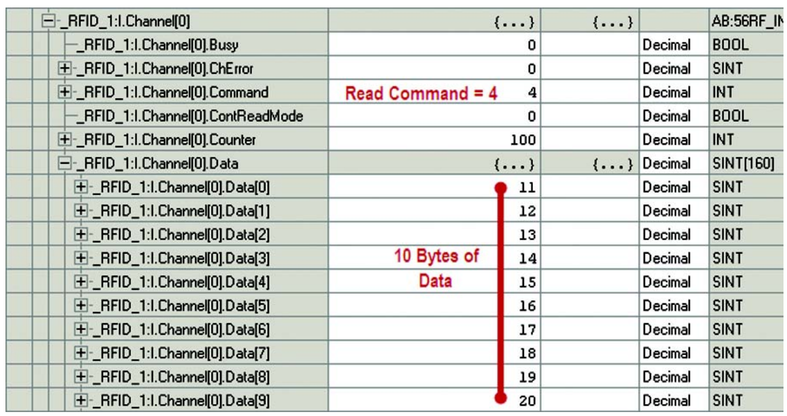
⊟-_RFID_1:O.Channel[0]	{...}	{...}		AB:56RF_IN
⊞-_RFID_1:O.Channel[0].Address	0		Decimal	INT
⊞-_RFID_1:O.Channel[0].BlockSize	0		Decimal	INT
⊞-_RFID_1:O.Channel[0].Command	14		Decimal	INT
⊟-_RFID_1:O.Channel[0].Data	{...}	{...}	Decimal	SINT[112]
⊞-_RFID_1:O.Channel[0].Data[0]	11		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[1]	12		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[2]	13		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[3]	10 Bytes of 14		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[4]	Data to Write 15		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[5]	16		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[6]	17		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[7]	18		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[8]	19		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[9]	20		Decimal	SINT

After successful completion of the Write Byte command, the input image table shows the UUID of the tag.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	14		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	99		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	16#0d		Hex	SINT
_RFID_1:I.Channel[0].Data[1]	16#ee		Hex	SINT
_RFID_1:I.Channel[0].Data[2]	16#e5		Hex	SINT
_RFID_1:I.Channel[0].Data[3]	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[4]	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[5]	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[6]	16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[7]	16#e0		Hex	SINT

The Read_Byte_Routine can be used to read the data. The data is stored in the input channel data, starting at location 0.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	4		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	100		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	11		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	12		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	13		Decimal	SINT
_RFID_1:I.Channel[0].Data[3]	14		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	15		Decimal	SINT
_RFID_1:I.Channel[0].Data[5]	16		Decimal	SINT
_RFID_1:I.Channel[0].Data[6]	17		Decimal	SINT
_RFID_1:I.Channel[0].Data[7]	18		Decimal	SINT
_RFID_1:I.Channel[0].Data[8]	19		Decimal	SINT
_RFID_1:I.Channel[0].Data[9]	20		Decimal	SINT

Write DSFID

The Write DSFID (Data Storage Format Identifier)command will write one byte of information in the Data Storage Format Identifier (DSFID) of the RFID tag.

Set the following values in the output image table:

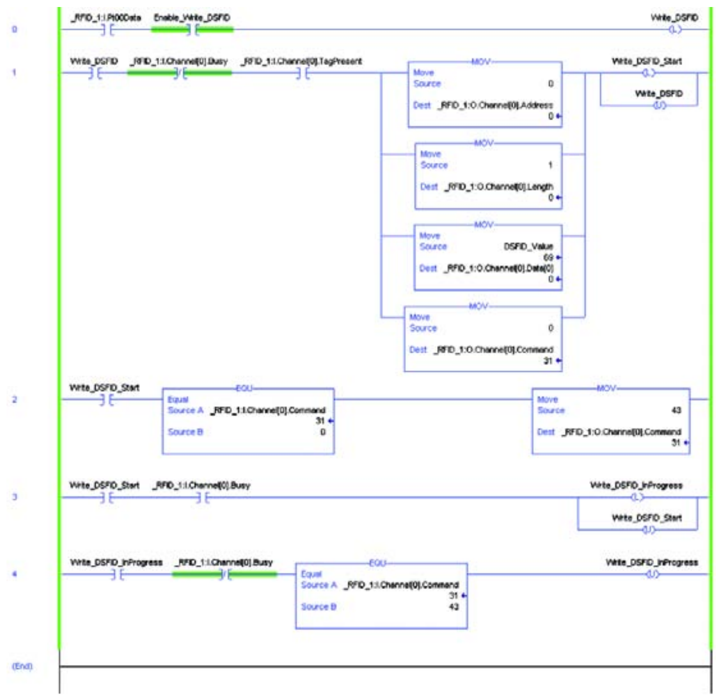
a. xx:O.Channel[0].Command = 43
b. xx:O.Channel[0].Address = 0
c. xx:O.Channel[0].Data[0] = DSFID value
d. xx:O.Channel[0].Length = 1
e. xx:O.Channel[0].Reset = 0
f. xx:O.Channel[0].Timeout = 0
g. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
h. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

If UIDLow and UIDHI are set to 0, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, and Data[0] values used to read multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.

The example ladder diagram is initially set for Address =0, the Length = 0. Data[0] is set to the DSFID value.



Example Results

The command is executed successfully if the ChError =0, the Command value = 43 and all the Data bytes are 0.

Use the Get System Information command or the Inventory command to read the DSFID.

⊟-_RFID_1:I.Channel	{...}	{...}		AB:56RF_IN
⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	ChError = 0 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	Command = 43 43		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	199		Decimal	INT
⊟-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞-_RFID_1:I.Channel[0].Data[0]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[1]	All Data Bytes are 0 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT

Write Multiple Blocks

The Write Multiple Blocks command writes to either one or two blocks of user data to a FRAM tag. This command will only work on FRAM tags.
Cat. No. 56RF-TG-2KB is a FRAM tag.

a. xx:O.Channel[0].Command = 11
b. xx:O.Channel[0].Address = starting block to write
c. xx:O.Channel[0].BlockSize = number of bytes per block
d. xx:O.Channel[0].Data[0…xxx] = data to write
e. xx:O.Channel[0].Length =the number of blocks to write
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

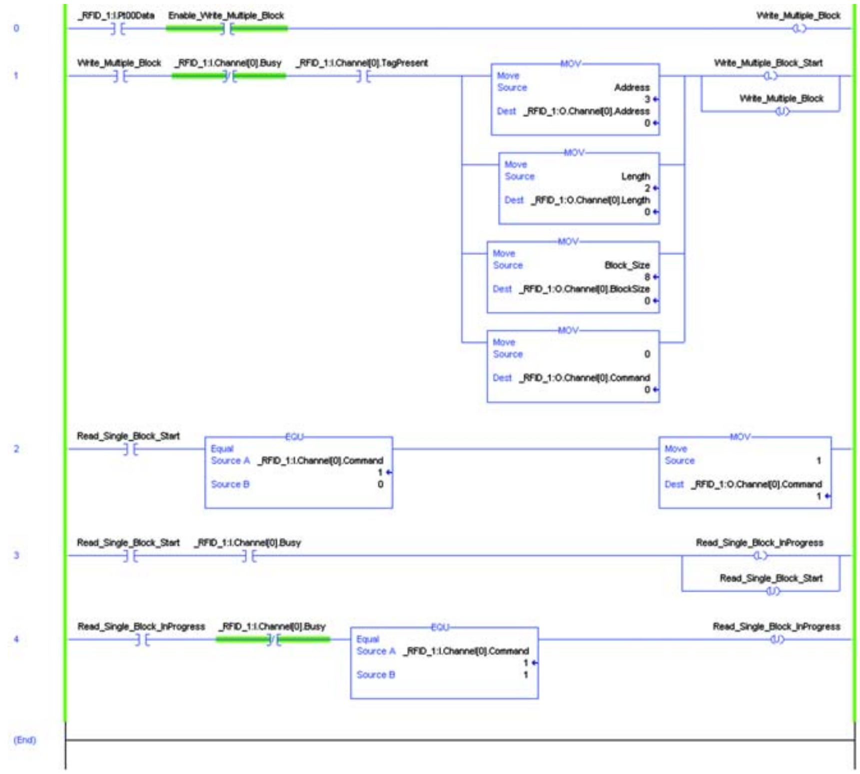
If UIDLow and UIDHi are set to 0, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

The table below shows the valid values for length, block size and the number of bytes written or each combination.

Length	1	1	1	2	2	2
Block Size	0	4	8	0	4	8
Bytes Written	4	4	8	8	8	16

Example Routine

In the example routine below, the initialization in Rung 1 sets the address, length, and block size values used to write multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.



Example Results

The figure below shows the output image table with the data that will be written (a simple numeric sequence starting at 2). Two blocks of 8 bytes each will be written to the tag. The data will be written to address locations 3 and 4.

⊟-_RFID_1:O.Channel[0]		{...}	{...}		AB:56RF_I
⊞-_RFID_1:O.Channel[0].Address	Place Data in Address 3	3		Decimal	INT
⊞-_RFID_1:O.Channel[0].BlockSize	Bytes per Block = 8	8		Decimal	INT
⊞-_RFID_1:O.Channel[0].Command		11		Decimal	INT
⊟-_RFID_1:O.Channel[0].Data		{...}	{...}	Decimal	SINT[112]
⊞-_RFID_1:O.Channel[0].Data[0]		2		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[1]		3		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[2]		4		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[3]	Data for Block 1	5		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[4]		6		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[5]		7		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[6]		8		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[7]		9		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[8]		10		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[9]		11		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[10]		12		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[11]	Data for Block 2	13		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[12]		14		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[13]		15		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[14]		16		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[15]		17		Decimal	SINT

If the write multiple blocks command is executed properly, the input table image results shows ChError = 0, Command = 11 and Data[0-xxx] =0.

_RFID_1:I.AuxPwrFault	0		Decimal	BOOL
_RFID_1:I.BlockFault	0		Decimal	BOOL
⊟ _RFID_1:I.Channel	{...}	{...}		AB:56RF_I
⊟ _RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞ _RFID_1:I.Channel[0].ChError	ChError = 0 0		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Command	Command = 11 11		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞ _RFID_1:I.Channel[0].Counter	521		Decimal	INT
⊟ _RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞ _RFID_1:I.Channel[0].Data[0]	0		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[1]	Data Bytes are 0 0		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[3]	0		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[4]	0		Decimal	SINT

Use the Read Multiple Block command (=2) to read the data.

⊟ _RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞ _RFID_1:I.Channel[0].ChError	No errors 0		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Command 2 = Read Multiple Blocks	2		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞ _RFID_1:I.Channel[0].Counter	18		Decimal	INT
⊟ _RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞ _RFID_1:I.Channel[0].Data[0]	2		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[1]	3		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[2]	4		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[3]	Data in 5		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[4]	Block 3 6		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[5]	7		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[6]	8		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[7]	9		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[8]	10		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[9]	11		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[10]	12		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[11]	Data in 13		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[12]	Block 4 14		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[13]	15		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[14]	16		Decimal	SINT
⊞ _RFID_1:I.Channel[0].Data[15]	17		Decimal	SINT

Multi-Tag Block Write

The Write Multi-Tag Block command writes one or more blocks of user data to multiple tags in the transceiver field. The maximum number tags in the RF field is limited to four and all tags must have the same block size.

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 12
b. xx:O.Channel[0].Address = starting address to write
c. xx:O.Channel[0].BlockSize = number of bytes/block
d. xx:O.Channel[0].Data[0...xxx] = data to write
e. xx:O.Channel[0].Length = number of blocks to write
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

If UIDLow and UIDHi are set to 0, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

Note: Length must be in 4-byte increments (e.g., 4, 8, 12…) for ISO15693 tags or 8-byte increments (e.g., 8, 16, 24…) for FRAM tags.

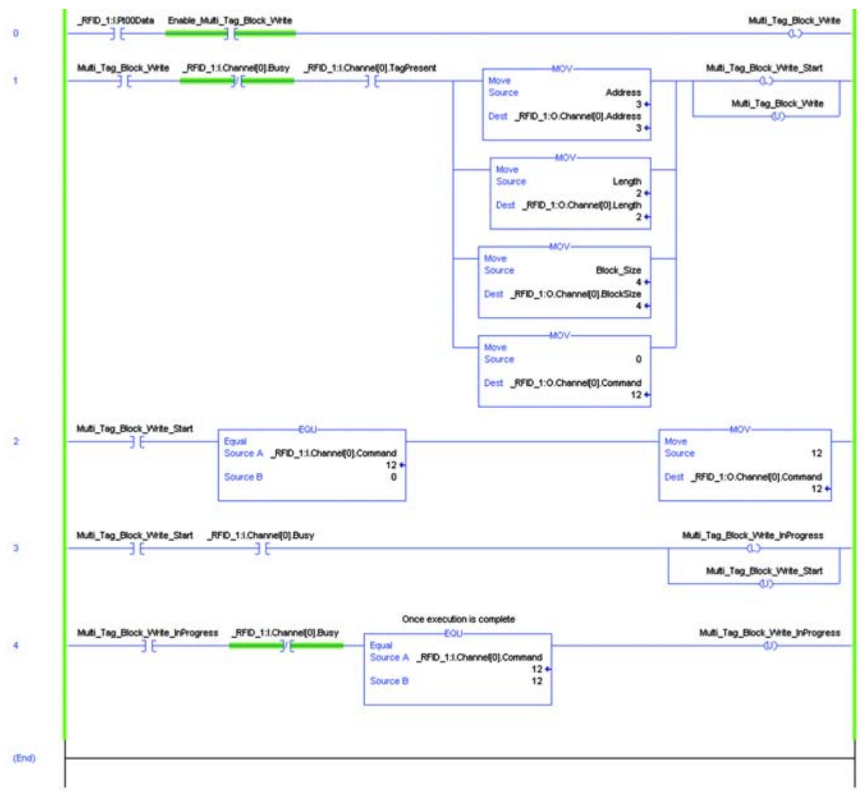
Note: The BlockSize field is used to specify the number of bytes/block of the tag. Valid values are:

- 0 = 4 bytes/block
- 4 = 4 bytes/block
- 8 = 8 bytes/block

Typically, ISO15693 tags have a block size of 4 bytes/block, and FRAM tags have a block size of 8 bytes/block.

Example Routine

In the example below, data will be written to two blocks, starting with Block 3. The data is loaded into the output channel image table. Block three will be populated with Data[0…3] = 11, 13, 15 and 17. Block 4 will be populated with Data[4…7] = 19, 21, 23, 25.



Example Results

The input channel image table will show the number of RFID tags that were written and the UUID of each RFID tag.

_RFID_1:I.Channel[0].Busy		0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError		0		Decimal	SINT
_RFID_1:I.Channel[0].Command		12		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode		0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter		286		Decimal	INT
_RFID_1:I.Channel[0].Data		{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]	Number of Tags	2		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	in RF Field	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]		16#c8		Hex	SINT
_RFID_1:I.Channel[0].Data[3]		16#25		Hex	SINT
_RFID_1:I.Channel[0].Data[4]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[5]		16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[6]	UUID for Tag 1	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[7]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[8]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[9]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[10]		16#ca		Hex	SINT
_RFID_1:I.Channel[0].Data[11]		16#53		Hex	SINT
_RFID_1:I.Channel[0].Data[12]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[13]	UUID for Tag 2	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[14]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[15]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[16]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[17]		16#e0		Hex	SINT

Use the Read Multi Tag Block command (=3) to read the blocks and confirm the data was written.

_RFID_1:I.Channel[0]		{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy		0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError		0		Decimal	SINT
_RFID_1:I.Channel[0].Command		3		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode		0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter		300		Decimal	INT
_RFID_1:I.Channel[0].Data		{...}	{...}	Decimal	SINT[160]
_RFID_1:I.Channel[0].Data[0]		2		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]		0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]		16#c8		Hex	SINT
_RFID_1:I.Channel[0].Data[3]		16#25		Hex	SINT
_RFID_1:I.Channel[0].Data[4]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[5]	Tag 1 UUID	16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[6]		16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[7]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[8]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[9]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[10]		11		Decimal	SINT
_RFID_1:I.Channel[0].Data[11]	Tag 1 Block 3 Data	13		Decimal	SINT
_RFID_1:I.Channel[0].Data[12]		15		Decimal	SINT
_RFID_1:I.Channel[0].Data[13]		17		Decimal	SINT
_RFID_1:I.Channel[0].Data[14]		19		Decimal	SINT
_RFID_1:I.Channel[0].Data[15]	Tag 1 Block 4 Data	21		Decimal	SINT
_RFID_1:I.Channel[0].Data[16]		23		Decimal	SINT
_RFID_1:I.Channel[0].Data[17]		25		Decimal	SINT
_RFID_1:I.Channel[0].Data[18]		16#ca		Hex	SINT
_RFID_1:I.Channel[0].Data[19]		16#53		Hex	SINT
_RFID_1:I.Channel[0].Data[20]		16#e6		Hex	SINT
_RFID_1:I.Channel[0].Data[21]		16#5b		Hex	SINT
_RFID_1:I.Channel[0].Data[22]	Tag 2 UUID	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[23]		16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[24]		16#04		Hex	SINT
_RFID_1:I.Channel[0].Data[25]		16#e0		Hex	SINT
_RFID_1:I.Channel[0].Data[26]		11		Decimal	SINT
_RFID_1:I.Channel[0].Data[27]	Tag 1 Block 3 Data	13		Decimal	SINT
_RFID_1:I.Channel[0].Data[28]		15		Decimal	SINT
_RFID_1:I.Channel[0].Data[29]		17		Decimal	SINT
_RFID_1:I.Channel[0].Data[30]		19		Decimal	SINT
_RFID_1:I.Channel[0].Data[31]	Tag 1 Block 4 Data	21		Decimal	SINT
_RFID_1:I.Channel[0].Data[32]		23		Decimal	SINT
_RFID_1:I.Channel[0].Data[33]		25		Decimal	SINT

Write Single Block

The Write Single Block command writes a single block of user data to an RFID tag.

Set the following values in the output image table:

a. xx:O.Channel[0].Command = 10
b. xx:O.Channel[0].Address = starting address to write
c. xx:O.Channel[0].BlockSize = 0, 4, or 8
d. xx:O.Channel[0].Data[0...112] = data to write
e. xx:O.Channel[0].Length = 0, 4, or 8
f. xx:O.Channel[0].BlockSize = 0, 4, or 8
g. xx:O.Channel[0].Reset = 0
h. xx:O.Channel[0].Timeout = 0
i. xx:O.Channel[0].UIDLow = 0 (or UIDLow)
j. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

If UIDLow and UIDHi are set to 0, this command will operate on the first tag in the field. Specify a UUID in xx:O.Channel[0].UIDLow and xx:O.Channel[0].UIDHi to perform the command on a specific tag.

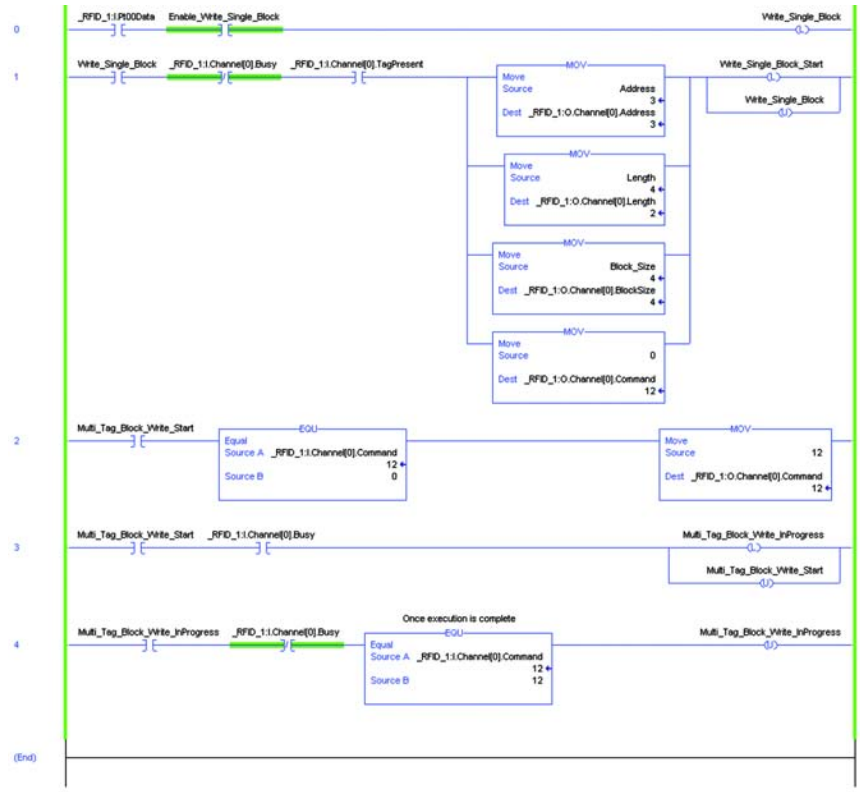
The Length and Block Size fields are used to specify the number of bytes/block of the tag. Valid values are:

- 0 = 4 bytes/block
- 4 = 4 bytes/block
- 8 = 8 bytes/block

Typically, ISO15693 tags have a block size of 4 bytes/block, and FRAM tags have a block size of 8 bytes/block.

Example Routine

In the example below, 4 bytes of data will be written to Block 3. The data is loaded into the output channel image table. Block three will be populated with Data[0…3] = 41, 42, 43, and 44.



Example Results

The output image table shows that the address is set to Block 3; the block size is 4 and the command is 10. The data to be written to block 3 is 41, 42, 43, and 44.

⊟-_RFID_1:O.Channel[0]	{...}	{...}		AB:56RF_IN_
⊞-_RFID_1:O.Channel[0].Address	Write to Block 3 3		Decimal	INT
⊞-_RFID_1:O.Channel[0].BlockSize	Block Size is 4 4		Decimal	INT
⊞-_RFID_1:O.Channel[0].Command	10		Decimal	INT
⊟-_RFID_1:O.Channel[0].Data	{...}	{...}	Decimal	SINT[112]
⊞-_RFID_1:O.Channel[0].Data[0]	41		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[1]	4 Bytes of Data 42		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[2]	to Write to Block 43		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[3]	44		Decimal	SINT

Upon successful completion of the write block command, the Input Image table shows that Command = 10 and ChError = 0. The input channel data fields are all zero.

⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	No errors 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	10		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	5		Decimal	INT
⊟-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞-_RFID_1:I.Channel[0].Data[0]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[1]	Data Bytes are 0 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT

Use the Read Single Block command (=1) , with option flag set to zero, to read the contents of the tag in block 3.

⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	No Errors 0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	1 = Read Block Cmd 1		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	6		Decimal	INT
⊟-_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[160]
⊞-_RFID_1:I.Channel[0].Data[0]	41		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[1]	Data From 42		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[2]	Block 3 43		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[3]	44		Decimal	SINT

Continuous Read Mode

The Continuous Read command is used for specialty applications requiring high line speeds (up to 3 m/s). Refer to Continuous Read Mode on page 120 for details on this command.

Stop Continuous Read

The Stop Continuous Read command is used in conjunction with the Continuous Read command for specialty applications requiring high line speeds (up to 3 m/s). Refer to Continuous Read Mode on page 120 for details on this command.

Teach Continuous Read

The Teach Continuous Read command is used to train the interface for Continuous Read operations. Refer to Teach Continuous Read on page 123 for details on this command.

SLC Code Examples

This sample code is an example using a SLC-5/05 with a Cat. No. 56RF-IN-IPD22 interface block.

Read Byte Routine

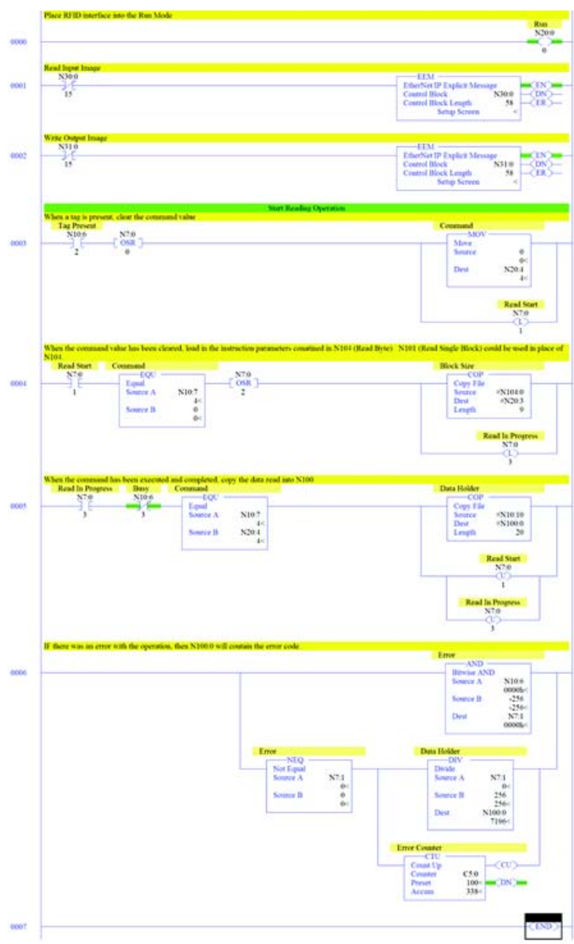
The Read Byte command (value =4) reads a user-specified number of bytes from a tag, starting at a user-specified address. Additionally, an Option Flag can be set to return the UUID of the tag.

- **Option Flag 0** – Returns the specified user data
- **Option Flag 1** – Returns the UUID of the tag and the specified user data

Note: This command operates only on the first tag in the field.

Example Routine

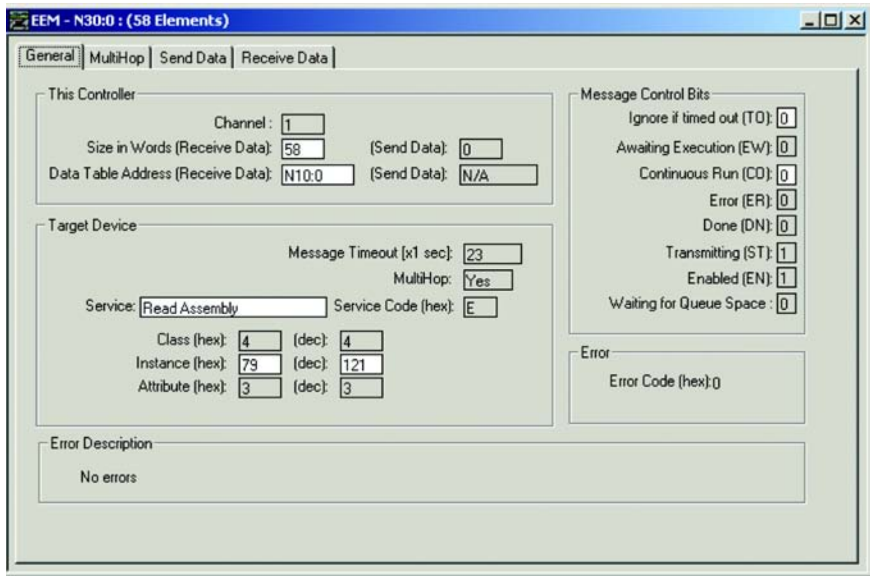
The example code below is for an SLC-5/05.



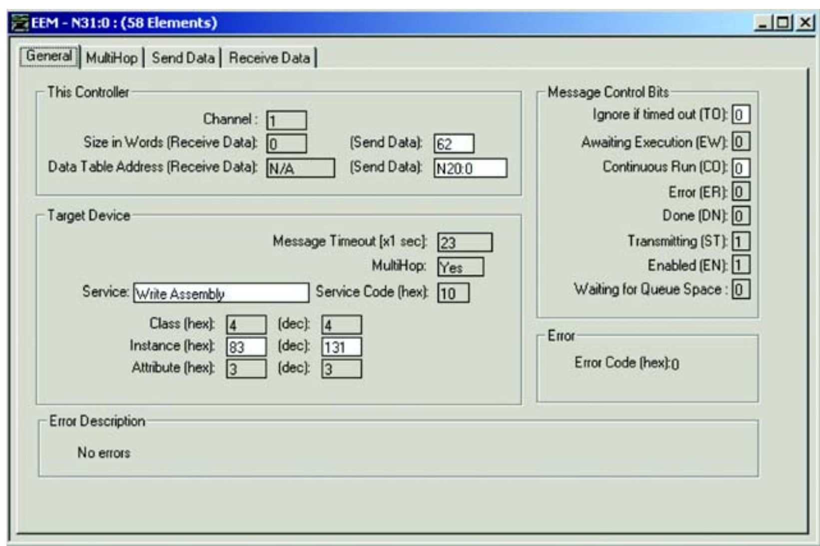
Example Routine

Rung 0000: Place RFID interface into the **Run** mode. The bit should be highlighted in green. If the bit is not green, right-click it and click "Toggle Bit".

Rung 0001: Read Input Image. Double-click the EEM box to enter the setup screen. Input Size is 116 bytes (58 words). Click on the MultiHop tab to set up an EtherNet/IP Device.

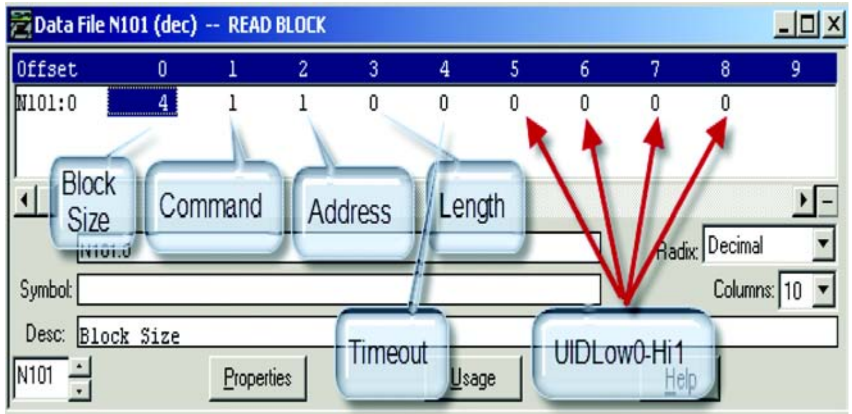


Rung 0002: Write Output Image. Double-click the MSG box to enter the setup screen. Output size is 124 bytes (62 words). Click on the MultiHop tab to set up an EtherNet/IP Device.



Rung 0003: The **Tag Present** bit will be highlighted in green when a tag is present. When a tag is present, clear the command value.

Rung 0004: When the command value has been cleared, load in the instruction parameters contained in N104 (Read Byte). N101 (Read Single Block) could be used in place of N104.



Rung 0005: Wait for the read command to run. The **Read in Progress** bit will be highlighted in green when the command is running. When the command has completed, the **Read in Progress** bit will return to its original state. When the command has been executed and completed, copy the data read into N100.

Rung 0006: If there was an error with the operation, then N100:0 will contain the error code.

Notes:

MicroLogix 1400 Code Examples

Read Byte

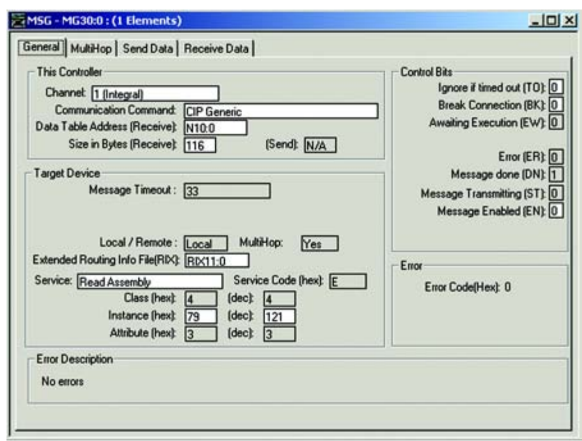
The Read Byte command (value =4) reads a user-specified number of bytes from a tag, starting at a user-specified address. Additionally, an Option Flag can be set to return the UUID of the tag.

- **Option Flag 0** – Returns the specified user data
- **Option Flag 1** – Returns the UUID of the tag and the specified user data

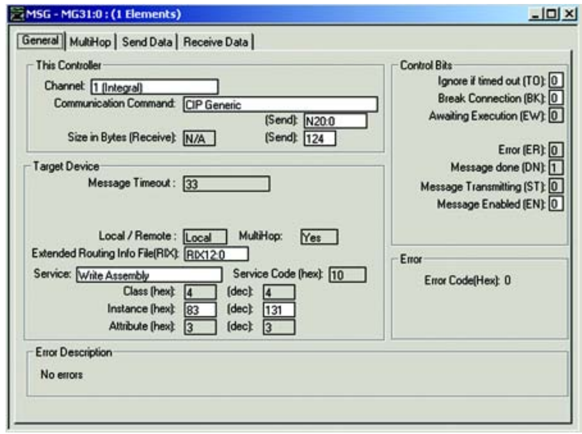
Note: This command operates only on the first tag in the field.

Example Routine

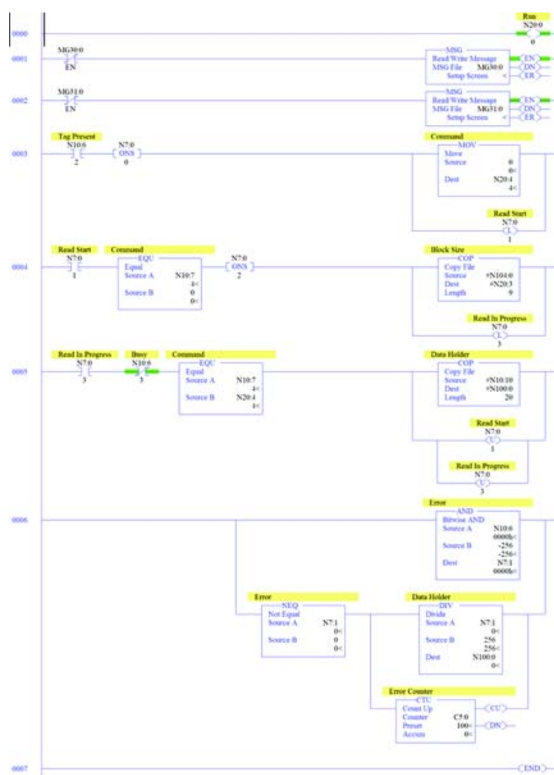
- **Rung 0000:** Place RFID interface into the Run Mode. The bit should be highlighted in green. If the bit is not green, right-click it and click **Toggle Bit**.
- **Rung 0001:** Read Input Image. Double-click the MSG box to enter the setup screen. Input size is 116 bytes (58 words). Click on the MultiHop tab to set up an EtherNet/IP Device.



- **Rung 0002:** Write Output Image. Double-click the MSG box to enter the setup screen. Output size is 124 bytes (62 words). Click on the MultiHop tab to set up an EtherNet/IP Device.



- **Rung 0003:** The **Tag Present** bit will be highlighted in green when a tag is present. When a tag is present, clear the command value.
- **Rung 0004:** When the command value has been cleared, load in the instruction parameters contained in N104 (Read Byte). N101 (Read Single Block) could be used in place of N104.
- **Rung 0005:** Wait for the read command to run. The **Read in Progress** bit will be highlighted in green when the command is running. When the command has completed, the **Read in Progress** bit will return to its original state. When the command has been executed and completed, copy the data read into N100.
- **Rung 0006:** If there was an error with the operation, then N100:0 will contain the error code.



Write Byte

The Write Byte command (value = 14) writes bytes of user data to a tag. The user can specify the data, the start byte, and the number of bytes to write.

Note: This command operates only on the first tag in the field.

Example Routine

- **Rung 0000:** Place RFID interface into the Run Mode. The bit should be highlighted in green. If the bit is not green, right-click it and click **Toggle Bit**.
- **Rung 0001:** Read Input Image. Double-click the MSG box to enter the Setup Screen. Input Size is 116 bytes (58 words). Click on the MultiHop tab to set up an EtherNet/IP Device.
- **Rung 0002:** Write Output Image. Double-click the MSG box to enter the Setup Screen. Output Size is 124 bytes (62 words). Click on the MultiHop tab to set up an EtherNet/IP Device.
- **Rung 0003:** The Tag Present bit will be highlighted in green when a tag is present. When a tag is present, clear the command value.
- **Rung 0004:** When the command value has been cleared, load in the instruction parameters contained in N114 (Write Byte). N110 (Write Single Block) could be used in place of N114.
- **Rung 0005:** Wait for the write command to run. The Write in Progress bit will be highlighted in green when the command is running. When the command has completed, the Write in Progress bit will return to its original state. When the command has been executed and completed, copy the data read into N100.
- **Rung 0006:** If there was an error with the operation, then N100:0 will contain the error code.

Read Multiple Blocks

The Read Multiple Blocks command (value = 2) reads multiple blocks of user data from a tag. Additionally, Option Flags can be set to return information such as the Universally Unique Identifier (UUID) or the Data Storage Format Identifier (DSFID) of the tag.

- **Option Flag 0** – Returns multiple blocks of user data
- **Option Flag 1** – Returns multiple blocks of user data and the security status of each block

Note: Unless a UUID is specified, this command will operate on the first tag in the field.

Example Routine

- **Rung 0000:** Place RFID interface into the Run Mode. The bit should be highlighted in green. If the bit is not green, right-click it and click “Toggle Bit”.

- **Rung 0001:** Read Input Image. Double-click the MSG box to enter the Setup Screen. Input Size is 116 bytes (58 Words.) Click on the MultiHop tab to set up an EtherNet/IP Device.
- **Rung 0002:** Write Output Image. Double-click the MSG box to enter the Setup Screen. Output Size is 124 bytes (62 Words). Click on the MultiHop tab to set up an EtherNet/IP Device.
- **Rung 0003:** The Tag Present bit will be highlighted in green when a tag is present. When a tag is present, clear the command value.
- **Rung 0004:** When the command value has been cleared, load in the instruction parameters contained in N102 (Read Multiple Blocks).
- **Rung 0005:** Wait for the read command to run. The Read in Progress bit will be highlighted in green when the command is running. When the command has completed, the Read in Progress bit will return to its original state. When the command has been executed and completed, copy the data read into N100.
- **Rung 0006:** If there was an error with the operation, then N100:0 will contain the error code.

Write Multiple Blocks

The Write Multiple Blocks command (value = 11) writes multiple blocks of user data to an FRAM tag.

Note: This command will only work on FRAM tags. Unless a UUID is specified, this command will operate on the first tag in the field.

Example Routine

- **Rung 0000:** Place RFID interface into the Run Mode. The bit should be highlighted in green. If the bit is not green, right-click it and click “Toggle Bit”.
- **Rung 0001:** Read Input Image. Double-click the MSG box to enter the Setup Screen. Input Size is 116 bytes (58 Words.) Click on the MultiHop tab to set up an EtherNet/IP Device.
- **Rung 0002:** Write Output Image. Double-click the MSG box to enter the Setup Screen. Output Size is 124 bytes (62 Words). Click on the MultiHop tab to set up an EtherNet/IP Device.
- **Rung 0003:** The Tag Present bit will be highlighted in green when a tag is present. When a tag is present, clear the command value.
- **Rung 0004:** When the command value has been cleared, load in the instruction parameters contained in N111 (Write Multiple Blocks).
- **Rung 0005:** Wait for the write command to run. The Write in Progress bit will be highlighted in green when the command is running. When the command has completed, the Write in Progress bit will return to its original state. When the command has been executed and completed, copy the data read into N100.
- **Rung 0006:** If there was an error with the operation, then N100:0 will contain the error code.

Input Image Layout

Refer to Appendix B (Class 4 Assembly Object) for details on the Input Image Layout.

Output Image Layout

Refer to Appendix B (Class 4 Assembly Object) for details on the Output Image Layout.

Notes:

RFID Tag Speed

The tables below are to be used as a guide to help determine the amount of information that can be written to/read from an RFID tag based on the speed of your application. For example, in order to consistently read 8 bytes from a tag using the square transceiver, your line speed should be 0.827 m/s or slower.

If you have a high speed application, it is best to choose the largest transceiver, larger tag, which will provide the largest antenna range. This will provide the longest time the tag is in the field for read/write functions. This will also help with tag misalignment issues.

If your tag will be stopped when all read/write functions occur, and tag misalignment is not an issue, smaller transceivers can be used. It is recommended that the tag be stopped if large amounts of data will be written to/read from the tag.

Table 18 - Rectangular (80x90) Transceiver

	Max Tag Speed (m/s)	
Bytes	**Read**	**Write**
4	1.488095	1.328609
8	1.378676	1.121915
16	1.202887	0.8566533
32	0.9578544	0.5811701
64	0.6802721	0.3535235
112	0.4743833	0.2227833
160	0.3641661	0.1626369
2000	0.03674939	0.01432665

Table 19 - Square (40x40) Transceiver

	Max Tag Speed (m/s)	
Bytes	**Read**	**Write**
4	0.8928571	0.7971656
8	0.8272058	0.6731489
16	0.7217322	0.513992
32	0.5747126	0.348702
64	0.4081633	0.2121141
112	0.28463	0.13367
160	0.2184996	0.09758213
2000	0.02204964	0.008595988

Table 20 - M18 Transceiver

	Max Tag Speed (m/s)	
Bytes	**Read**	**Write**
4	0.1984127	0.1771479
8	0.1838235	0.1495886
16	0.1603849	0.1142204
32	0.1277139	0.07748935
64	0.09070295	0.04713646
112	0.06325111	0.02970444
160	0.04855547	0.02168492
2000	0.004899919	0.00191022

Table 21 - M30 Transceiver

	Max Tag Speed (m/s)	
Bytes	**Read**	**Write**
4	0.3373016	0.3011515
8	0.3125	0.2543007
16	0.2726544	0.1941748
32	0.2171137	0.1317319
64	0.154195	0.08013199
112	0.1075269	0.05049755
160	0.0825443	0.03686436
2000	0.008329863	0.003247374

Continuous Read Mode

Command Objective

Perform tag read operations as fast as possible.

Operation

Command 5 will be issued from the controller to place an interface RFID channel into continuous read mode; no additional commands will be required from the controller in order to retrieve information from a tag. The read type issued would be a Read Multiple Block or a Read Single Block depending on the number of blocks requested. The maximum number of blocks that can be read at one time is 10. Each time the interface reads a tag successfully, the counter value will increment by 1. If there was an issue reading the tag the counter value will not increment and the ChError will indicate the error code value.

While the interface is in this mode, it will reject all other commands sent to it for that channel except a Stop Continuous Read. The interface will not perform its normal poll cycle on that channel while it is in this mode of operation. During Continuous Read Mode, the ContReadMode and Busy bit will be set to true.

When the interface receives a stop command, **Command 6**, it will revert back to the normal mode of operation and resume the polling cycle. Continuous Read mode can also be canceled by issuing a channel reset (reset bit in the output image word set to 1).

When using a 50 mm disc tag, Cat. No. 56RF-TR-8090 transceiver, and reading 4 bytes of data it may be possible to achieve a line speed of up to 3 m/s.

Modes of Operation

Only one type of mode of operation can be used on each channel. To change modes you will need to issue a Stop Continuous Read, and then reissue a Start Continuous Read with the new mode. Both channels can be setup for the same mode or different modes simultaneously. Modes of operation are limited based on the model number of the interface.

56RF-IN-IPS12

- 1 RFID Channel (Channel 0)
- 1 discrete input and 1 discrete output
- Support modes 0 and 1 only

56RF-IN-IPD22

- 2 RFID Channels (Channel 0, Channel 1)
- 1 discrete input and 1 discrete output
- Support modes 0, and 1 only.

The single input can be used for either channel.

56RF-IN-IPD22A

- 2 RFID Channels (Channel 0, Channel 1)
- 2 discrete inputs
- Support modes 0, 1, 2, and 3

The same input can be used for either channel.

Mode Overview

1. Mode 0

The interface waits for the delay time, sends out a read, obtains data, and returns that data back to the PLC. This cycle repeats until a Stop Continuous Read command is issued.

2. Mode 1

The interface waits for input point 0 to turn ON, waits for the delay timer to expire then sends out a read, obtains data, and returns that data back to the PLC. This cycle repeats until a Stop Continuous Read command is issued.

3. Mode 2

The interface waits for input point 1 to turn ON, waits for the delay timer to expire then sends out a read, obtains data, and returns that data back to the PLC. This cycle repeats until a Stop Continuous Read command is issued.

4. Mode 3

The interface waits for both input point 0 and 1 to turn ON, waits for the delay timer to expire then sends out a read, obtains data, and returns that data back to the PLC. This cycle repeats until a Stop Continuous Read command is issued.

Command Structure

a. xx:O.Channel[0].Reset =0
b. xx:O.Channel[0].Command = 5
c. xx:O.Channel[0].BlockSize = Bytes per Block in the tag
d. xx:O.Channel[0].Address = Starting Block
e. xx:O.Channel[0].Length = Number of blocks to read
f. xx:O.Channel[0].Timeout = Delay time between sending commands
g. xx:O.Channel[0].UIDLow = 0
h. xx:O.Channel[0].UIDHi = 0
i. xx:O.Channel[0].Data[0] = Mode x
j. xx:O.Channel[0].Data[1] = Option Flag

Address – Block within the tag to start read operations from.

BlockSize – Size in bytes per block of the tag.

Length – Number of blocks to read

Timeout – Delay time between sending command attempts in Mode 0.
Delay time after input condition is true before sending commands in Mode 1-3.

UIDLow/UIDHigh – Can be used to target only a specific tag for read operations, otherwise this value would be 0 to read any tag.

Mode x – Specifies the mode of operation for the Continuous Read.

Option Flag – Used to specify the mode of the Read Multiple/ Read Single Block(s) command.
A 0 value would only read the data requested starting at the address specified, for the number of blocks specified in the Length field. A value of 1 would read and return both the security block status and the tag data.
For modes 1…3, users can either set the delay time on their own or they can train the interface and the transceiver so that the value is determine automatically based on their system setup and line speed. A delay time of 0 will cause the interface to send out the command as soon as it sees the input condition go true.
For mode 0, there is no ability to train the system.

Teach Continuous Read

Command Objective

This operation is valid only for modes 1…3 and is used to train the interface to the approximate delay time that should be used before it sends out the read command based on input conditions and tag speeds.

Operation

Command 8 will be issued from the Controller to place an RFID interface channel into teach mode.

When first entering Teach Mode (Phase 1), the interface will wait for the input condition(s) to go true, and then poll for tag detection. Once 10 good detections have occurred the unit will enter phase 2.

During Phase 2, the unit will wait for the input condition(s) to go true, then issue the Read Multiple/Read Single Block command after the predetermined time delay and adjust the delay time as necessary. Once 10 good reads in a row have occurred, the unit will exit teach mode and report back the average and recommended delay time in milliseconds.

If the interface is unable to obtain 10 good reads in a row, it will decrement the delay time by 1ms and start again in phase 2. If the delay time has been decremented more than 30 ms from the average the interface will exit teach mode and report back the recommended delay time of -1. A -1 value indicates that the interface cannot determine what the best delay time would be due to variations in tag speed.

Phase progression in teach mode can be monitored by viewing the counter value in the input image table. Phase 1 will always be a value <10, Phase 2 will always be a value >10. Once the counter hits 20 the interface will exit teach mode and report the average and recommended delay times. The user will need to load the recommended delay time value into the Timeout field prior to initiating a continuous read.

During Teach Mode, the ContReadMode and Busy bit will be set to true.

Teach mode can canceled by issuing a channel reset (reset bit in the output image word set to 1).

Command Structure

a. xx:O.Channel[0].Reset =0
b. xx:O.Channel[0].BlockSize =Bytes per Block in the tag
c. xx:O.Channel[0].Command = 8
d. xx:O.Channel[0].Address = Starting Block
e. xx:O.Channel[0].Length = Number of Blocks
f. xx:O.Channel[0].Timeout = 0
g. xx:O.Channel[0].UIDLow = 0
h. xx:O.Channel[0].UIDHi = 0
i. xx:O.Channel[0].Data[0] = Mode x
j. xx:O.Channel[0].Data[1] = Option Flag

RFID Interface Block Web Page

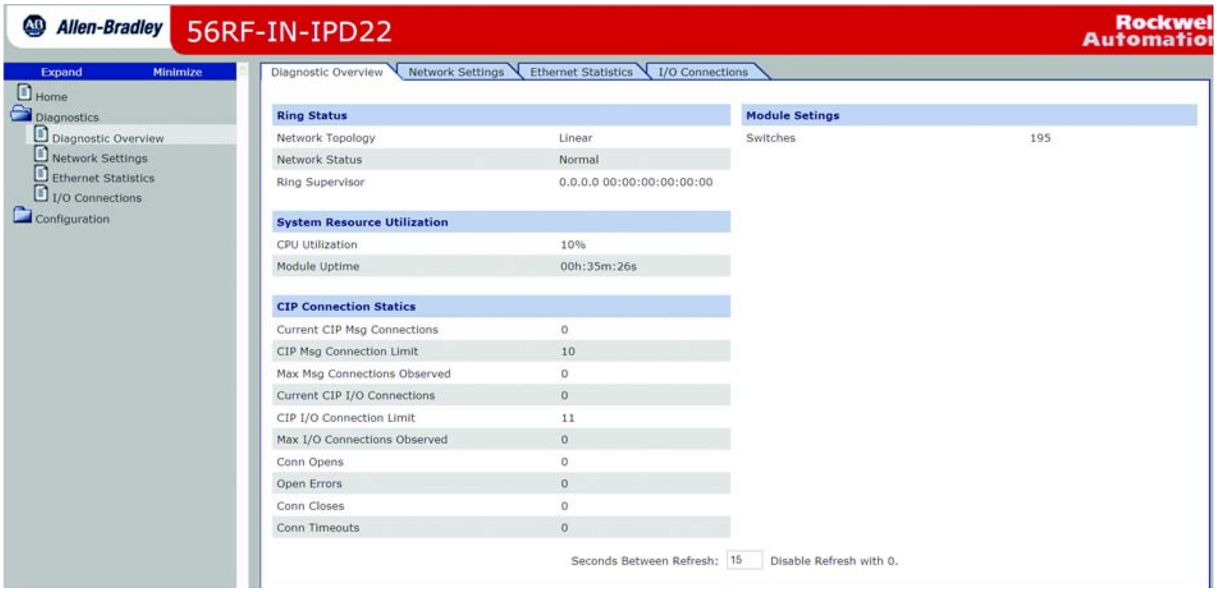
The RFID interface block web page is accessible by entering the IP address of the interface block into a web browser. The interface block must have EtherNet connectivity and power to be viewable on the web page. The web page provides diagnostic and configuration for the RFID interface block.

Home

The Home page allows the user to view basic information about the interface block. Data cannot be changed on the home page. The Device Description and Device Location is specified and can be changed on the Device Identity tab in the Configuration section.

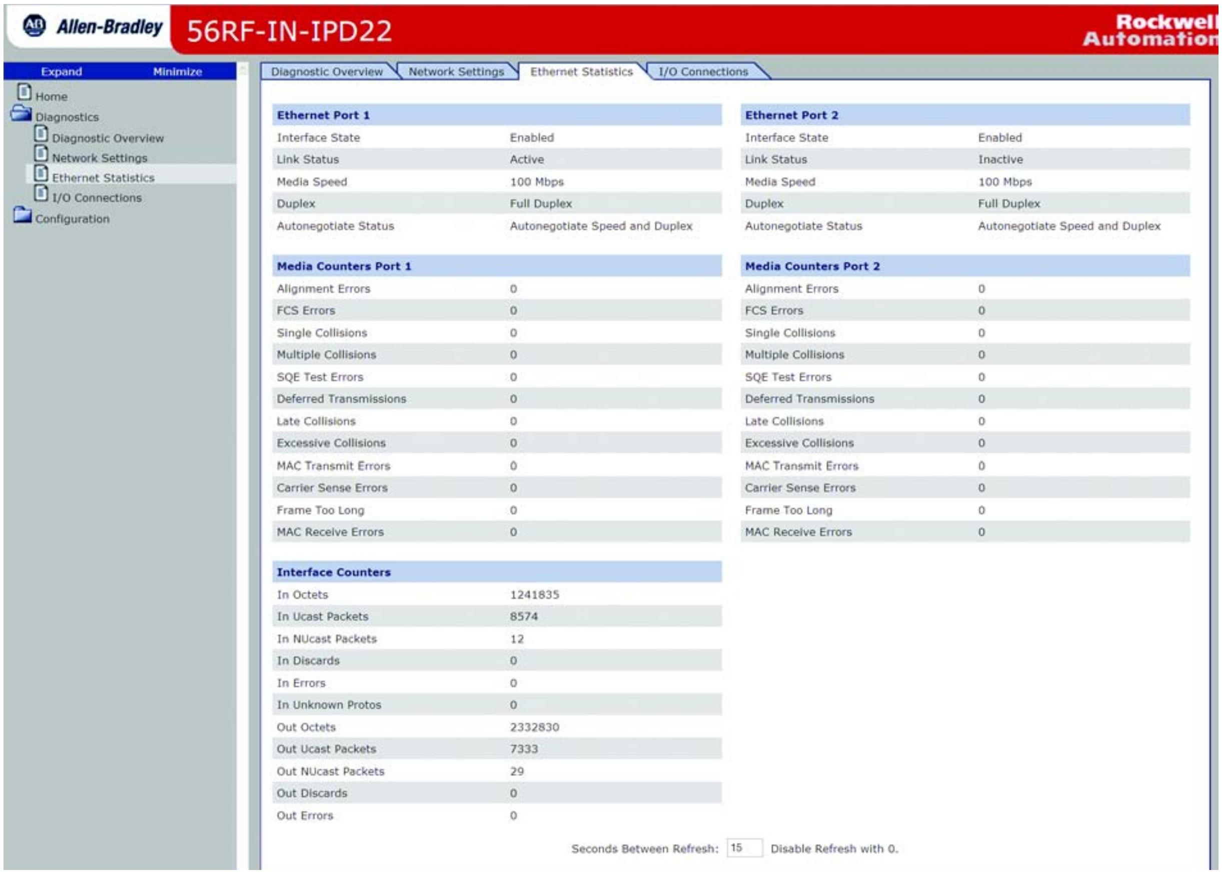
Diagnostics

The Diagnostic section has three tabs of "view only" detailed information on the status of the interface block. The tabs show Diagnostic Overview, Network Settings, and EtherNet Statistics. The I/O Connections tab contains a field that allows the user to change the web page refresh rate.



Network Settings

EtherNet Statistics



I/O Connections

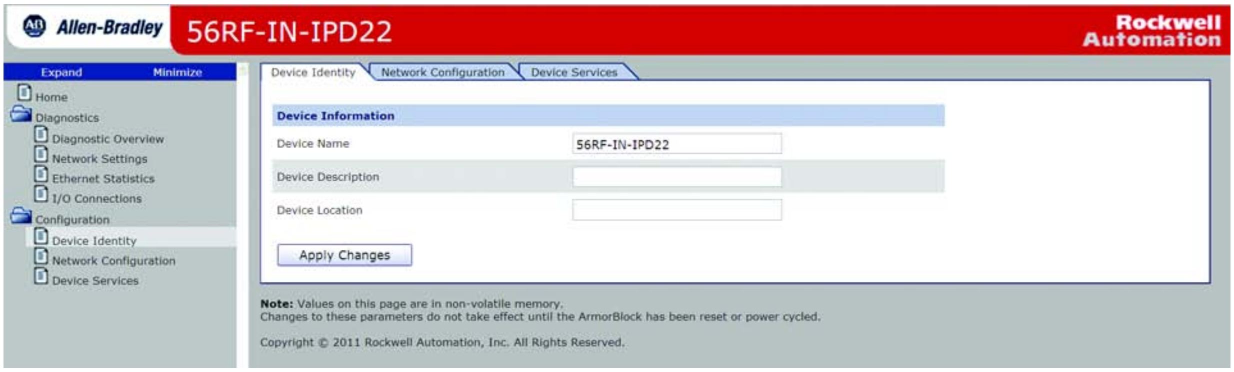
Configuration

To access the configuration section of the RFID interface block web page, a username and password are required. The default username is "Admin", and there is no password by default. The username and password can be changed on the Device Services tab.

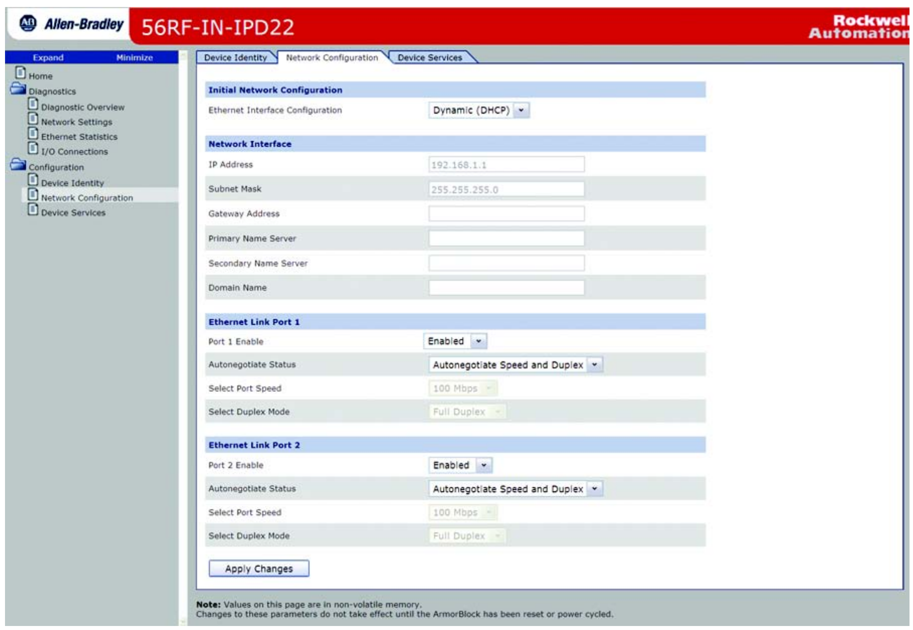
Note: If the username and password are lost, the interface block must be reset to default before it can be accessed again. This will reset the username and password to the default values above.

Device Identity

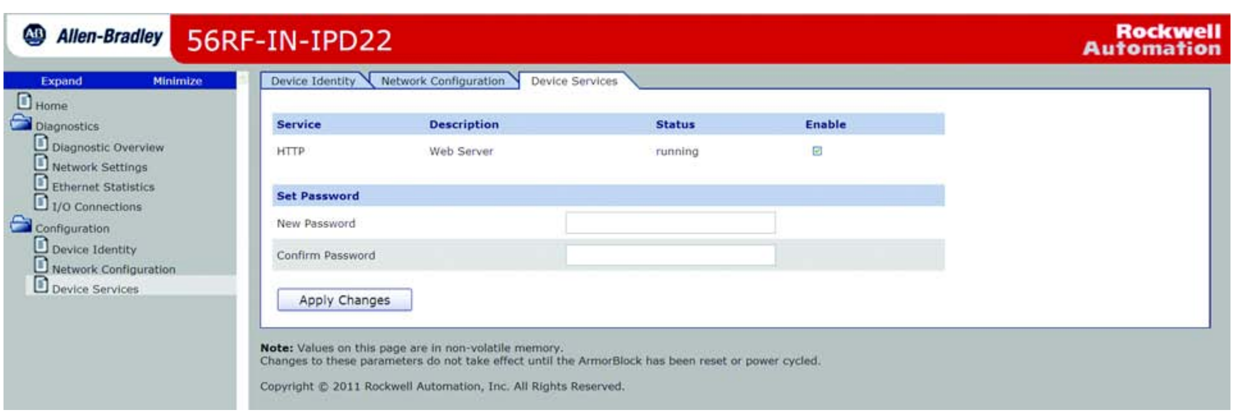
Change the device name, description, or location. Changes will take place after the interface block has been power cycled.



Network Configuration



Device Services



Error Codes for RFID Interface Block

Error Codes

The error codes for the RFID interface block are stored in the input for each channel. In the examples in the manual, the error codes are stored in the image table RFID_1:I:Channel[0].ChError and RFID_1:I:Channel[1].ChError.

Error Codes	Status Word	Binary
0	OK	0000
1	Transceiver not found	0001
2	Invalid Response	0010
3	Invalid Parameter	0011
4	No Tag Detected	0100
5	Instruction Timed Out	0101
6	Block Access Error	0110
7	Format Error	0111
8	Tag Communications Error	1000
9	Address Error	1001
10	Mismatch Error	1010
11	Internal Channel Error	1011
12	Malformed Packet	1100
13	Unit in Program Mode	1101
14	Reserved	1110
15	Module Error	1111

- **OK (Decimal 0)** – Indicates there are no issues with the channel in question when the decimal value of these bits is equal to zero.
- **Transceiver not found (Decimal 1)** – Indicates that communications with the transceiver for the specified channel has been lost.
- **Invalid Response (Decimal 2)**– Indicates that the response to a command is not what was expected.
- **Invalid Parameter (Decimal 3)** – Indicates that either a parameter passed or received was out of bounds.
- **No Tag Detected (Decimal 4)**– Indicates that a command was attempted on a channel but there was no tag detected in the field.
- **Instruction Timed Out (Decimal 5)**- Indicates that the timeout value associated with a command was exceeded before a response could be obtained.
- **Block Access Error (Decimal 6)** – Indicates that either:
  - A read command attempted to read a block but was denied access.
  - A write command attempted to write to a block but was denied access.
- **Format Error (Decimal 7)**– Indicates that the format of the command or response was invalid.

- **Tag Communications Error (Decimal 8)**– Indicates that the interface block was not able to complete execution of a command with a tag before the tag left the field or the Output Channel Timeout is set too short. For example, set the Output Channel Timeout to 100 ms and then try to read 112 byte of data from a Cat. No. 56RF-TG-30 tag.
- **Address Error (Decimal 9)**– Indicates that the block address value was out of bounds for the tag.
- **Mismatch Error (Decimal 10)** – Indicates that there are more tags detected in the field than the unit can process.
- **Internal Channel Error (Decimal 11)** – Indicates that there is some internal issue with channel (hardware fault).
- **Malformed Packet (Decimal 12)** –Indicates an issue with the command packet received by the transceiver.
- **Unit in Program Mode (Decimal 13)** –Indicates that a command was issued but the module is in program mode.
- **Module Error (Decimal 15) –** Indicates that there is some internal issue interface block (hardware fault).

CIP Information

Product Codes and Name Strings

The following table lists the product codes and name strings for the EtherNet/IP interface block.

Product Type	Product Code	Cat. No.	Identity Object Name String
139	4	56RF-IN-IPS12	RFID Adapter 1 Port + 1In/1 Out
139	5	56RF-IN-IPD22	RFID Adapter 2 Port + 1In/1 Out
139	6	56RF-IN-IPD22A	RFID Adapter 2 Port + 2In/0 Out

CIP Explicit Connection Behavior

The RFID interface block allows user outputs to be driven by connected explicit messages when no I/O connection exists, or when an I/O connection exists in the idle state. A single EtherNet/IP Class 3 explicit connection will be allowed to send “explicit control” messages via an “Active Explicit” connection. An EtherNet/IP Class 3 explicit connection becomes the “explicit control” connection when it becomes the first EtherNet/IP Class 3 explicit connection to send a “set” service to one of the following:

- The “value” attribute of any DOP instance (class code 0x09).
- The “data” attribute of any output (consumed) Assembly instance (class code 0x04).
- Attribute 3 or 4 of the Control Supervisor Object (class code 0x29).

CIP Objects

The following CIP objects will be covered in the following subsections. CIP objects provide a window into the devices properties that can be read/written to. Each CIP Class contains instances (copies of a class structure), and attributes for each instance. Most devices will have only one instance of a class.

Class	Object
0x0001	Identity Object
0x0004	Assembly Object
0x0008	Discrete Input Point Object
0x0009	Discrete Output Point Object

Identity Object Class Code 0x0001

This Identity Object provides identification of and general information about the device.

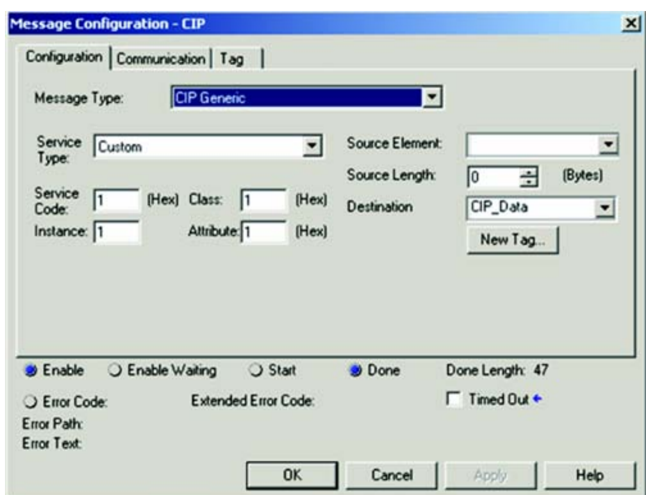
Instance 1 of the Identity Object will contain the following attributes:

Attribute ID	Access Rule	Name	Data Type	Value
1	Get	Vendor	UINT	1
2	Get	Device Type	UINT	139
3	Get	Product Code	UINT	4,5, or 6
4	Get	Revision Major Revision Minor Revision	Structure of: USINT USINT	The initial release is Major Rev. 1, Minor Rev. 1.
5	Get	Status	WORD	Refer to Device Status (CIP_Data[8]-[9]) table below.
6	Get	Serial Number	UDINT	Unique number for each device
7	Get	Product Name String Length ASCII String	Structure of: USINT STRING	Product Code specific

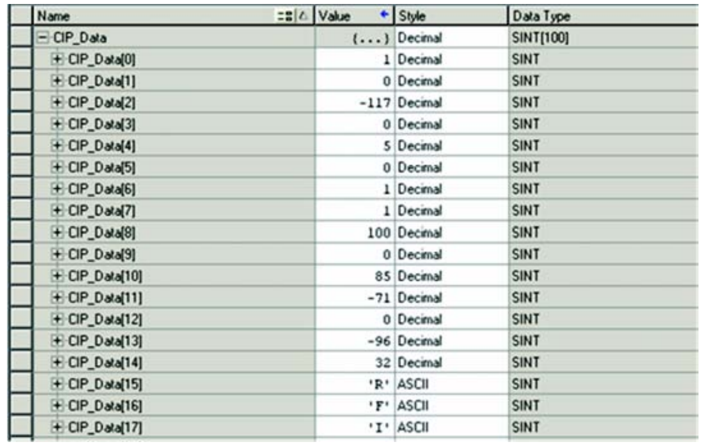
The following common services will be implemented for Instance 1.

Service Code	Implemented for:		Service Name
	Class	Instance	
0x01	Yes	Yes	Get_Attributes_All
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attributes_Single

Accessing the Identity Object will require the creation of a Message Instruction (MSG) to be configured as a CIP Generic type.



Service Code: 1- Get Attribute All
Class: 1 - Identity Object
Instance: 1 - First instance
Attribute: 1 - First attribute
Destination: CIP_Data - a SINT[100] array to hold the data



Name	Value	Style	Data Type
CIP_Data	{...}	Decimal	SINT[100]
CIP_Data[0]	1	Decimal	SINT
CIP_Data[1]	0	Decimal	SINT
CIP_Data[2]	-117	Decimal	SINT
CIP_Data[3]	0	Decimal	SINT
CIP_Data[4]	5	Decimal	SINT
CIP_Data[5]	0	Decimal	SINT
CIP_Data[6]	1	Decimal	SINT
CIP_Data[7]	1	Decimal	SINT
CIP_Data[8]	100	Decimal	SINT
CIP_Data[9]	0	Decimal	SINT
CIP_Data[10]	85	Decimal	SINT
CIP_Data[11]	-71	Decimal	SINT
CIP_Data[12]	0	Decimal	SINT
CIP_Data[13]	-96	Decimal	SINT
CIP_Data[14]	32	Decimal	SINT
CIP_Data[15]	'R'	ASCII	SINT
CIP_Data[16]	'F'	ASCII	SINT
CIP_Data[17]	'I'	ASCII	SINT

CIP_Data[0]...[1]= Vendor (1=Allen-Bradley)
CIP_Data[2]...[3]= Device Type (139=RFID)
CIP_Data[4]...[5]=Device Code (5=56RF-IN-IPS12)
CIP_Data[6]= Major Revision (1)
CIP_Data[7]= Minor Revision (1)
CIP_Data[8]...[9]= Status (100 decimal, 000000001100100 binary)
CIP_Data[10]...[13]= Serial Number (A000B955)
CIP_Data[14]= Product Name Length (32 bytes)
CIP_Data[15]-[n]= Product Name

Table 22 - Device Status (CIP_Data[8]…[9])

Bits	Name	Description
0	Owned	0=Not Owned, 1=Owned by a Master
1	Reserved	Reserved
2	Configured	0=Not configured, 1=Configured
3	Reserved	Reserved
4…7	Extended Device Status	See table below
8	Minor Recoverable Fault	1=Detected a recoverable minor fault
9	Minor Unrecoverable Fault	1=Detected a non-recoverable minor fault
10	Major Recoverable Fault	1=Detected a recoverable major fault
11	Major Unrecoverable Fault	1=Detected a non-recoverable major fault
12…15	Reserved	Reserved

Table 23 - Values for the Extended Device Status (bits 4…7)

Value	Description
0	Self-Testing or Unknown
1	Firmware Update in Progress
2	At least one faulted I/O connection
3	No I/O connections established
4	Non-Volatile Configuration Bad
5	Major Fault
6	At least one I/O connection in run mode
7	At least one I/O connection established, all in idle mode
8 & 9	Reserved
10…15	Vendor specific

Assembly Object Class Code 0x0004

The Assembly Object binds attributes of multiple objects, which allows data to or from eachobject to be sent or received over a single connection. Controllers that do not have the ability to create and establish a class 1 (scheduled) connection can utilize the Assembly Object in a Message Instruction to obtain both the input and output assemblies of the RFID interface.

The following services will be implemented for the Assembly Object:

Sevice Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single
0x18	No	Yes	Get_Member

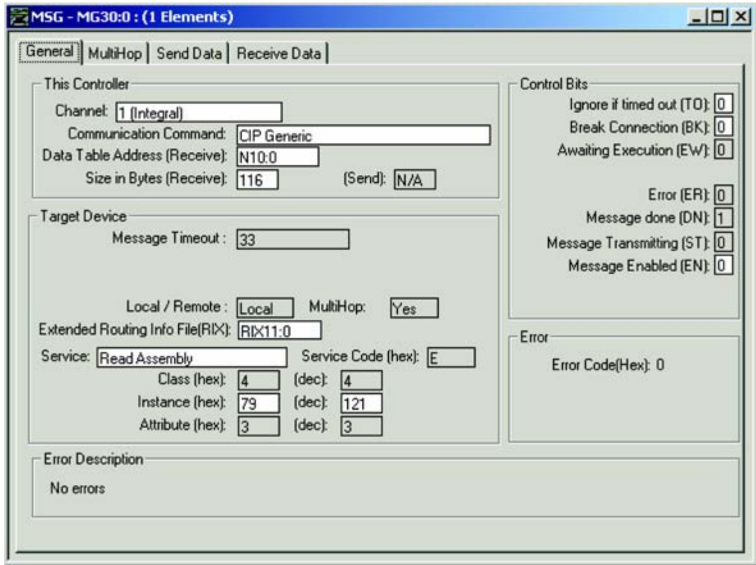
Different connection instances are needed for each RFID interface based on the model. These class 3 connection instances are different than the class 1 instances used by a ControlLogix or CompactLogix processor due to the limitations within the SLC and Micrologix for handling Send and Receive data.

Use the table below to determine the class 3 connection instance and Send / Receive size for your unit:

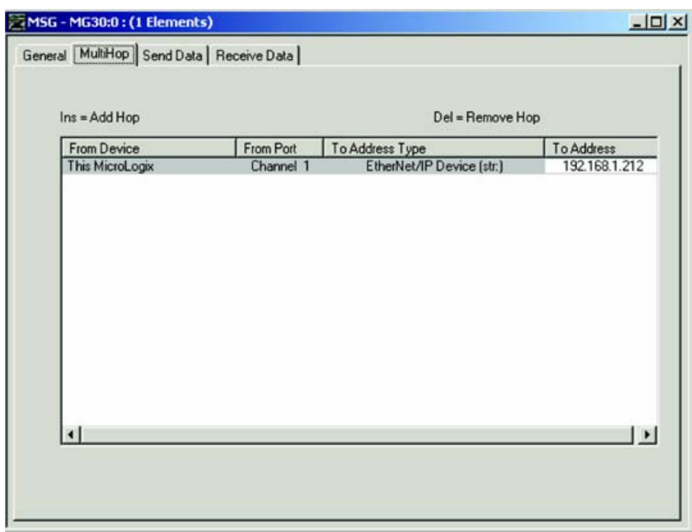
Table 24 - Class 3 Connection Instances with size (in bytes)

Cat. No.	Input	Size	Output	Size	Config	Size
56RF-IN-IPS12	120	64	130	64	103	16
56RF-IN-IPD22	121	116	131	124	109	20
56RF-IN-IPD22A	122	116	132	124	112	24

Reading the Input Image Table of a 56RF-IN-IPD22 with a MicroLogix 1400



- N10:0 is the data table address where the input image will be stored and will span N10:0 thru N10:57.
- The number of bytes to receive is 116 (58 words).
- The extended routing file (RIX11:0) is used to store the Multi-Hop routing information.
- Service is type Read Assembly
- Class 4 is the Assembly Instance Class
- Instance 79h is the input image connection instance.
- Attribute 3 is the assembly attribute for the input image table



The Multi-Hop information is used to configure the communications path from the MicroLogix to the RFID interface.

Input Image (56RF-IN-IPD22)

Word	Description	Word	Description
N10:0 – N10:1	Module Connection Status	N10:9	Length
N10:2	Module Status	N10:10 – N10:31	Data
N10:3	Reserved	N10:32	Channel[1] Diagnostics
N10:4	Block Status	N10:33	Command Value
N10:5	I/O Data	N10:34	Counter Value
N10:6	Channel[0] Diagnostics	N10:35	Length
N10:7	Command Value	N10:36 – N10:57	Data
N10:8	Counter Value		

Module Status

Bit	Definition	Bit	Definition
0	Run Status	8	Reserved
1	Block Fault	9	Reserved
2	Aux Power Fault	10	Reserved
3	Reserved	11	Reserved
4	Pt00 Input Fault	12	Pt00 Output Fault
5	Pt00 Open Wire	13	Pt00No Load
6	Pt00 Input Short Circuit	14	Pt00 Output Short Circuit
7	Reserved	15	Reserved

I/O Data

Bit	Definition	Bit	Definition
0	Pt00 Data	8	Pt00 Readback
1	Reserved	9	Reserved
2	Reserved	10	Reserved
3	Reserved	11	Reserved
4	Reserved	12	Reserved
5	Reserved	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

Channel[n] Diagnostics

Bit	Definition	Bit	Definition
0	Reset	8	Error Code
1	Fault	9	Error Code
2	Tag Present	10	Error Code
3	Busy	11	Error Code
4	Reset in Progress	12	Reserved
5	Continuous Read Mode	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

Input Image (56RF-IN-IPD22A)

Word	Description	Word	Description
N10:0 – N10:1	Module Connection Status	N10:9	Length
N10:2	Module Status	N10:10 – N10:31	Data
N10:3	Reserved	N10:32	Channel[1] Diagnostics
N10:4	Block Status	N10:33	Command Value
N10:5	I/O Data	N10:34	Counter Value
N10:6	Channel[0] Diagnostics	N10:35	Length
N10:7	Command Value	N10:36 – N10:57	Data
N10:8	Counter Value		

Module Status

Bit	Definition	Bit	Definition
0	Run Status	8	Pt01 Input Fault
1	Block Fault	9	Pt01 Open Wire
2	Aux Power Fault	10	Pt01 Input Short Circuit
3	Reserved	11	Reserved
4	Pt00 Input Fault	12	Reserved
5	Pt00 Open Wire	13	Reserved
6	Pt00 Input Short Circuit	14	Reserved
7	Reserved	15	Reserved

I/O Data

Bit	Definition	Bit	Definition
0	Pt00 Data	8	Reserved
1	Pt01 Data	9	Reserved
2	Reserved	10	Reserved
3	Reserved	11	Reserved
4	Reserved	12	Reserved
5	Reserved	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

Channel[n] Diagnostics

Bit	Definition	Bit	Definition
0	Reset	8	Error Code
1	Fault	9	Error Code
2	Tag Present	10	Error Code
3	Busy	11	Error Code
4	Reset in Progress	12	Reserved
5	Continuous Read Mode	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

Input Image (56RF-IN-IPS12)

Word	Description	Word	Description
N10:0 – N10:1	Module Connection Status	N10:6	Channel[0] Diagnostics
N10:2	Module Status	N10:7	Command Value
N10:3	Reserved	N10:8	Counter Value
N10:4	Block Status	N10:9	Length
N10:5	I/O Data	N10:10 – N10:31	Data

Module Status

Bit	Definition	Bit	Definition
0	Run Status	8	Reserved
1	Block Fault	9	Reserved
2	Aux Power Fault	10	Reserved
3	Reserved	11	Reserved
4	Pt00 Input Fault	12	Pt00 Output Fault
5	Pt00 Open Wire	13	Pt00 No Load
6	Pt00 Input Short Circuit	14	Pt00 Output Short Circuit
7	Reserved	15	Reserved

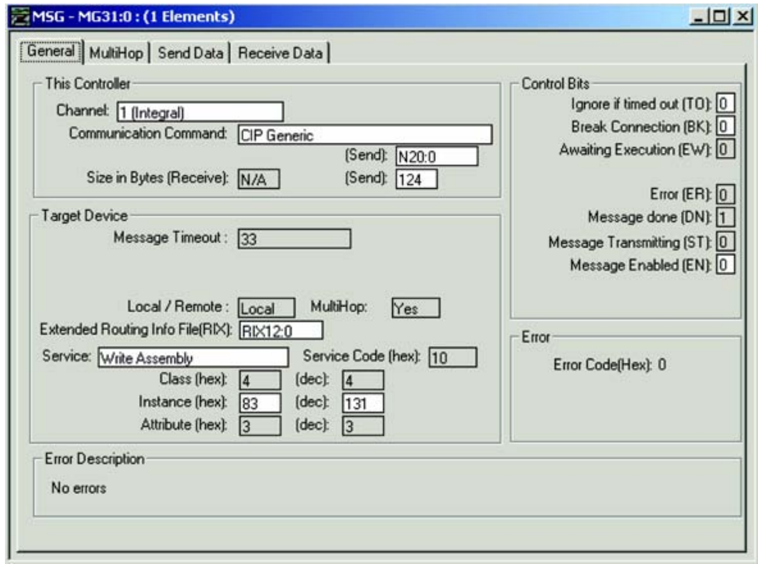
I/O Data

Bit	Definition	Bit	Definition
0	Pt00 Data	8	Pt00 Readback
1	Reserved	9	Reserved
2	Reserved	10	Reserved
3	Reserved	11	Reserved
4	Reserved	12	Reserved
5	Reserved	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

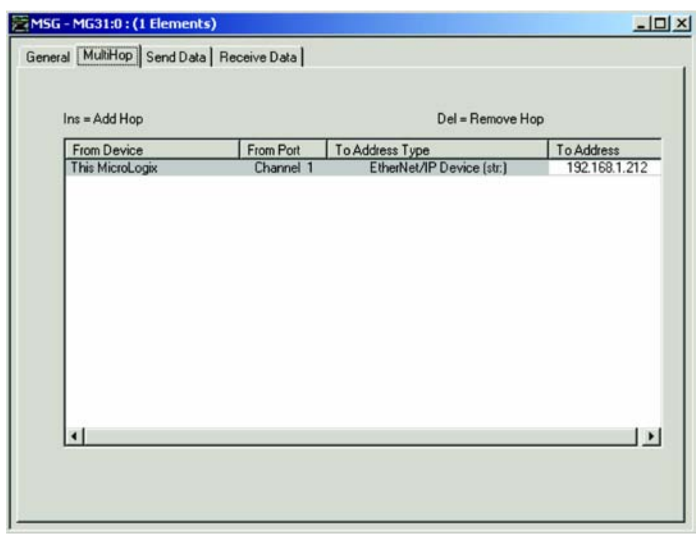
Channel[n] Diagnostics

Bit	Definition	Bit	Definition
0	Reset	8	Error Code
1	Fault	9	Error Code
2	Tag Present	10	Error Code
3	Busy	11	Error Code
4	Reset in Progress	12	Reserved
5	Continuous Read Mode	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

Writing to the Output Image Table of a 56RF-IN-IPD22 with a MicroLogix 1400



- N20:0 is the data table address to store the output image and will span N20:0…N20:61.
- The number of bytes to send is 124 (62 words).
- The extended routing file (RIX12:0) is used to store the Multi-Hop routing information.
- Service is type Write Assembly
- Class 4 is the Assembly Instance Class
- Instance 83h is the output image connection instance.
- Attribute 3 is the assembly attribute for the output image table



The Multi-Hop information is used to configure the communications path from the MicroLogix to the RFID interface.

Input Image (56RF-IN-IPD22)

Word	Description	Word	Description
N20:0	Module Data	N20:12…N10:31	Data
N20:1	Reserved	N20:32	Channel[1] Reset
N20:2	Channel[0] Reset	N20:33	Block Size
N20:3	Block Size	N20:34	Command
N20:4	Command	N20:35	Address
N20:5	Address	N20:36	Length
N20:6	Length	N20:37	Timeout
N20:7	Timeout	N20:38…N20:39	UIDLow
N20:8…N20:9	UIDLow	N20:40…N20:41	UIDHi
N20:10…N20:11	UIDHi	N20:42…N20:61	Data

Module Data

Bit	Definition	Bit	Definition
0	Run Mode	8	Pt00 Data
1	Reserved	9	Reserved
2	Reserved	10	Reserved
3	Reserved	11	Reserved
4	Reserved	12	Reserved
5	Reserved	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

Input Image (56RF-IN-IPD22A)

Word	Description	Word	Description
N20:0	Module Data	N20:12…N10:31	Data
N20:1	Reserved	N20:32	Channel[1] Reset
N20:2	Channel[0] Reset	N20:33	Block Size
N20:3	Block Size	N20:34	Command
N20:4	Command	N20:35	Address
N20:5	Address	N20:36	Length
N20:6	Length	N20:37	Timeout
N20:7	Timeout	N20:38…N20:39	UIDLow
N20:8…N20:9	UIDLow	N20:40…N20:41	UIDHi
N20:10…N20:11	UIDHi	N20:42…N20:61	Data

Module Data

Bit	Definition	Bit	Definition
0	Run Mode	8	Reserved
1	Reserved	9	Reserved
2	Reserved	10	Reserved
3	Reserved	11	Reserved
4	Reserved	12	Reserved
5	Reserved	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

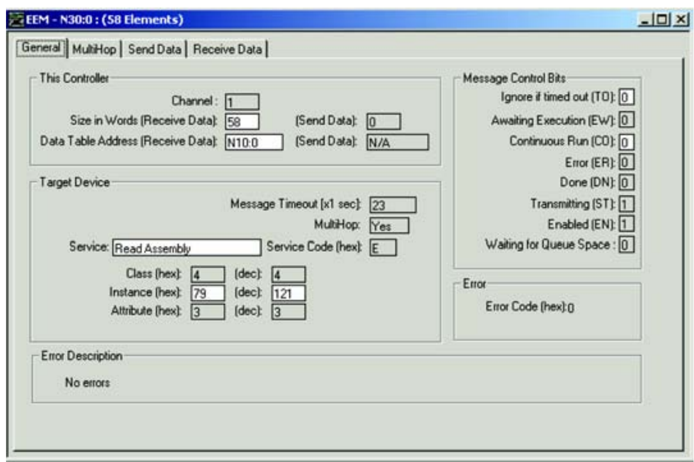
Input Image (56RF-IN-IPS12)

Word	Description	Word	Description
N20:0	Module Data	N20:6	Length
N20:1	Reserved	N20:7	Timeout
N20:2	Channel[0] Reset	N20:8…N20:9	UIDLow
N20:3	Block Size	N20:10…N20:11	UIDHi
N20:4	Command	N20:12…N10:31	Data
N20:5	Address		

Module Data

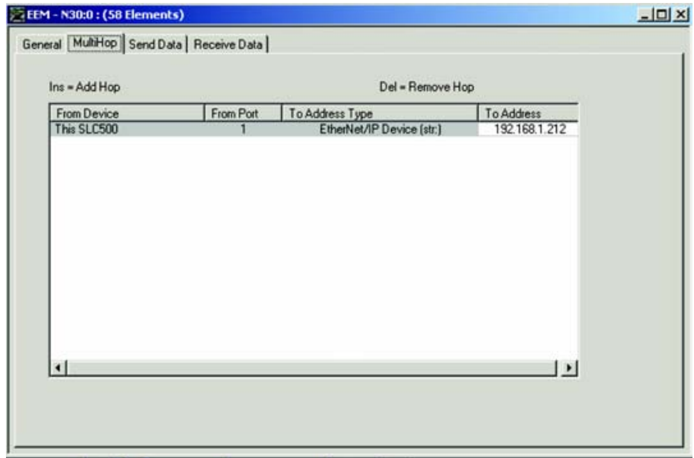
Bit	Definition	Bit	Definition
0	Run Mode	8	Pt00 Data
1	Reserved	9	Reserved
2	Reserved	10	Reserved
3	Reserved	11	Reserved
4	Reserved	12	Reserved
5	Reserved	13	Reserved
6	Reserved	14	Reserved
7	Reserved	15	Reserved

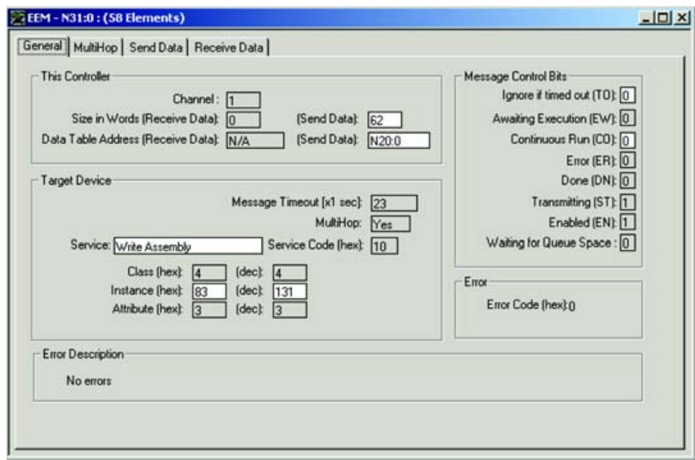
Reading the Input Image Table of a 56RF-IN-IPD22 with a SLC-5/05



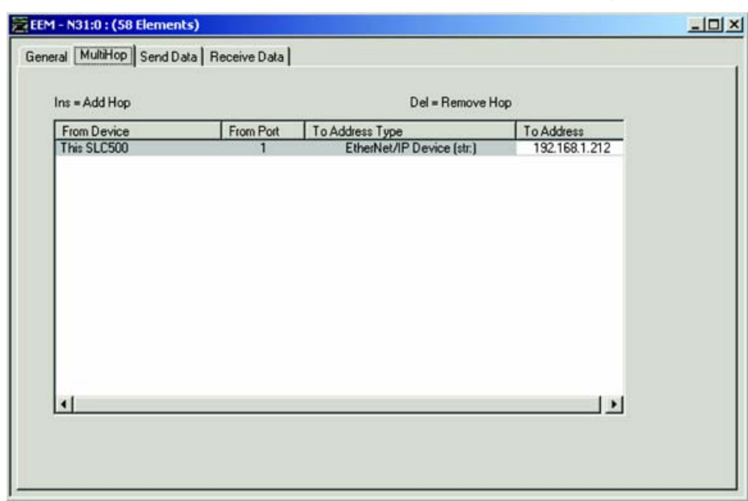
The biggest difference between the MicroLogix1400 and the SLC-5/05 is that the SLC uses an EEM instruction instead of a MSG instruction, but the setup is basically the same. The routing information for the EEM is stored within the Control Block address (N30:0)

- N10:0 is the data table address where the input image will be stored and will span N10:0 thru N10:57.
- The size in words is 58 (116 bytes).
- Service is type Read Assembly
- Class 4 is the Assembly Instance Class
- Instance 79h is the input image connection instance.
- Attribute 3 is the assembly attribute for the input image table





- N20:0 is the data table address to store the output image and will span N20:0…N20:61.
- The Send Data size is 62 (124 bytes).
- Service is type Write Assembly
- Class 4 is the Assembly Instance Class
- Instance 83h is the output image connection instance.
- Attribute 3 is the assembly attribute for the output image table



Class 1 Connections

Class 1 connections are used to transfer I/O data, and can be established to the assembly object instances. Each Class 1 connection establishes two data transports, one consuming and one producing. The heartbeat instances are used for connections that shall access only inputs. Class 1 uses UDP transport.

- Total numbers of supported Class 1 connections equals 2 (total for: exclusive owner + input only + listen only)
- Supported API: 2…3200 ms (Note that the minimum API can be higher if processor resources become a problem)
- T->O Connection type: point-to-point, multicast
- O->T Connection type: point-to-point
- Supported trigger type: cyclic, change-of-state

The producing instance can be assigned to multiple transports, using any combination of multicast and point-to-point connection types.

Only one Exclusive-owner connection will be supported at each time. If an Exclusive-owner connection is already established and an originator tries to establish a new Exclusive-owner connection an "Ownership conflict" (general status = 0x01, extended status = 0x0106) error code will be returned.

For a connection to be established the requested data sizes must be an exact match of the connections points that the connection tries to connect to. If the requested and actual sizes don't match, an "Invalid connection size" (general status = 0x01, extended status = 0x0109) error code will be returned.

Exclusive Owner Connection

This connection type is used for controlling the outputs of the module and shall not be dependent on any other condition. Only one exclusive owner connection can be opened against the module.

If an exclusive owner connection is already opened "Connection in use" (general status = 0x01, extend status = 0x0100) shall be returned an error code.

- Connection point O -> T shall be Assembly object, Instance 3, 162 or 166 (162 for product codes <= 0x100 only, 166 for product codes > 0x100 only).
- Connection point T -> O shall be Assembly object, Instance 52, 150 or 151 (150 for product codes <= 0x100 only, 151 for product codes > 0x100 only).

Input Only Connection

This connection is used to read data from the module without controlling the outputs. This connection is not dependent on any other connection.

It is recommended that the originator sets the data size in the O->T direction of the Forward_Open be zero.

IMPORTANT	If an exclusive owner connection has been opened against the module and times out, the input only connection shall time-out as well. If the exclusive owner connection is properly closed, the input only connection shall not be affected.

- Number of supported input only connections equals two (shared with exclusive owner and listen only connection).
- Connection point O -> T shall be Assembly object, Instance 191 (Input only heartbeat).
- Connection point T -> O shall be Assembly object, Instance 52, 150, or 151 (150 for product codes <= 0x100 only, 151 for product codes > 0x100 only).

Listen Only Connection

This connection is dependent on another connection to exist. If that connection(exclusive owner or input only) is closed, the listen only connection shall be closed as well.

It is recommended that the originator sets the data size in the Forward_Open be zero.

- Number of supported listen only connections equals two (shared with exclusive owner and listen only connection).
- Connection point O -> T shall be Assembly object, Instance 192 (Listen only heartbeat)
- Connection point T -> O shall be Assembly object, Instance 52, 150 or 151 (150 for product codes <= 0x100 only, 151 for product codes > 0x100 only)

Class 3 Connections

Class 3 connections are used to establish connections to the message router. The connection is used for explicit messaging. Class 3 connections use TCP connections.

- Three concurrent encapsulation sessions will be supported
- Six concurrent Class 3 connections will be supported
- More than one Class 3 connection per encapsulation session will be supported
- Supported API: 100…10000 ms
- T->O Connection type: point-to-point
- O->T Connection type: point-to-point
- Supported trigger type: application

Discrete Input Point Object Class Code 0x0008

The following class attributes are currently supported for the Discrete Input Point Object:

Attribute ID	Access Rule	Name	Data Type	Value
1	Get	Revision	0xC7	2
2	Get	Max Instance	UINT	4

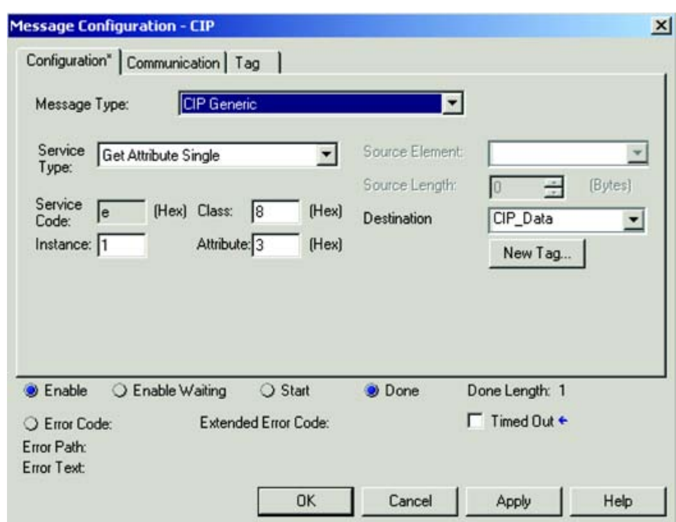
Two instances of the Discrete Input Point Object are supported. All instances contain the following attributes.

Attribute ID	Access Rule	Name	Data Type	Value
3	Get	Value	BOOL	0 = OFF, 1 = ON
5		FilterOffOn	0xC7	0 = No Delay 1000 = 1 ms 2000 = 2 ms 4000 = 4 ms 8000 = 8 ms 16000 = 16 ms
6		FilterOnOff	0xC7	0 = No Delay 1000 = 1 ms 2000 = 2 ms 4000 = 4 ms 8000 = 8 ms 16000 = 16 ms

The following common services will be implemented for the Discrete Input Point Object.

Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

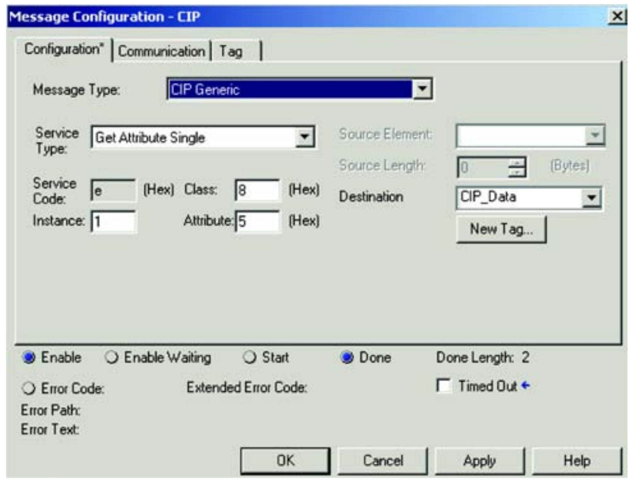
To obtain the status of an input point (ON or OFF), configure a CIP message as shown below:



Instance 1 is the first input (Pt00), if the RFID interface supports two inputs, then Pt01 would be instance 2.

The return value in CIP_Data[0] will be either 0 (Input OFF) or 1 (Input ON).

To obtain the Input Filter OffOn value of an input point, configure a CIP message as shown below:



Instance 1 is the first input (Pt00), if the RFID interface supports two inputs, then Pt01 would be instance 2.

The return value will contain the filter time in milliseconds.

Discrete Output Point Object Class Code 0x0009

The following class attributes are supported:

Attribute ID	Access Rule	Name	Data Type	Value
1	Get	Revision	0xC1	1
2	Get	Max Instance	UINT	4 or 10

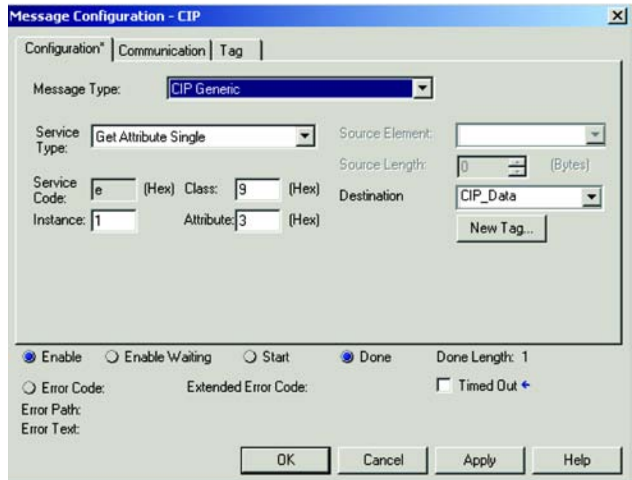
Two instances of the Discrete Output Point Object are supported. All instances contain the following attributes.

Attribute ID	Access Rule	Name	Data Type	Value
3	Get	Value	BOOL	0 = OFF, 1 = ON
5	Get/Set	FaultMode	BOOL	0 = Use Fault Value 1 = Hold Last State
6	Get/Set	FaultValue	BOOL	0 = OFF 0 = ON
7	Get/Set	ProgMode	BOOL	0 = Use Program Value 1 = Hold Last State
8	Get/Set	ProgValue	BOOL	0 = OFF 1 = ON

The following common services are implemented for the Discrete Output Point Object.

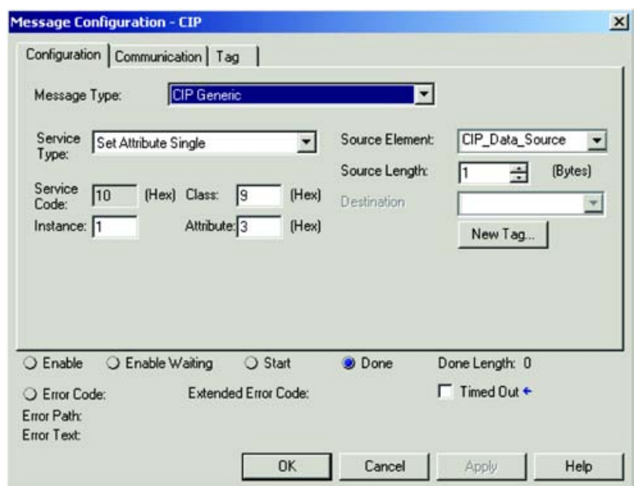
Service Code	Implemented for:		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

To obtain the state of an output point, configure a CIP message as shown below:



The return value will contain the state of the output (0=Off, 1=On)

To set the state of an output point, configure a CIP message as shown below:



CIP_Data_Source is a SINT that will contain the value to set the output too (0=Off, 1=On).

Notes:

Install the AOP (Add-On Profile)

Introduction

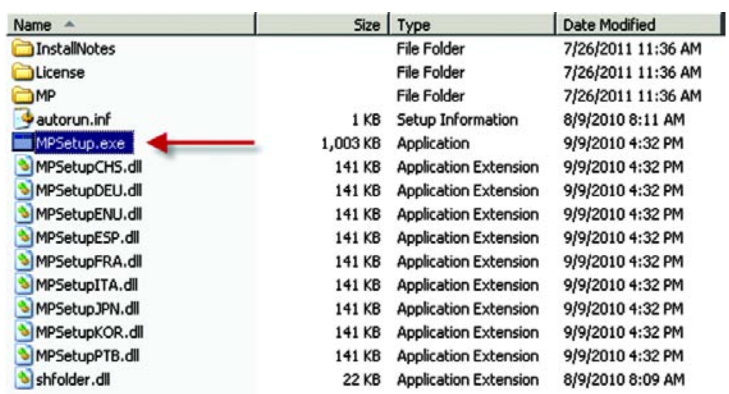
This chapter goes through the add-on profile (AOP) of the RFID transceivers with the RSLogix 5000 program. Add-On Profiles are files that users add to their Rockwell Automation library. These files contain the pertinent information for configuring a device that will be added to the Rockwell Automation network.

The Add-On Profile simplifies the setup of devices because it presents the necessary fields in an organized fashion, which allows users to set up and configure their system in a quick and efficient manner.

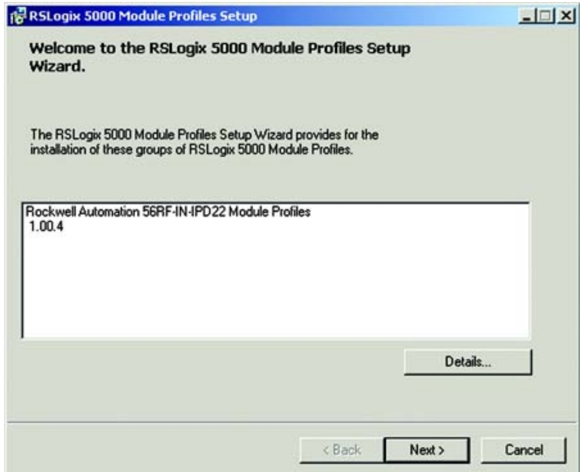
The AOP is a folder containing numerous files for the device. It will come as an installation package. Install the AOP following the on-screen instructions.

In the File Explorer, locate the directory where the installation files were extracted.

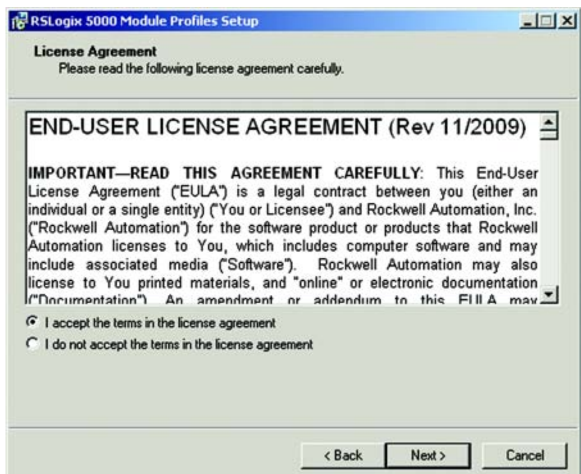
Click **MPSetup.exe**



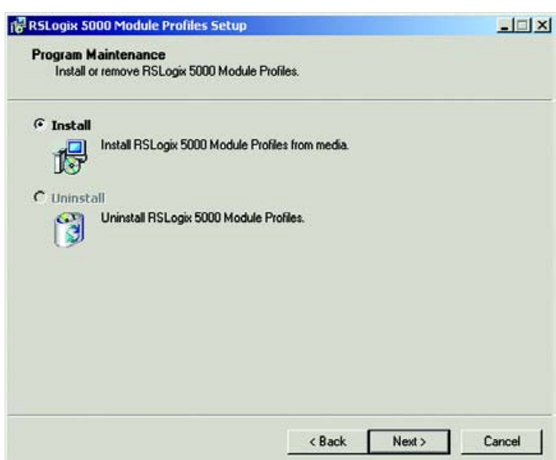
The window identifies the module profiles and the firmware version.



Click **Next**

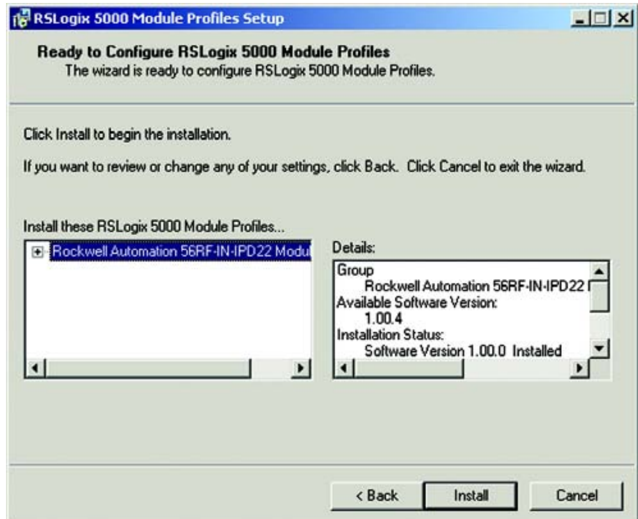


Accept the terms of the license agreement and click **Next**.



With Install selected, click **Next**.

The profile name appears in the left-hand box and its details appear in the right-hand box. Verify that the module name is correct.



Click **Install**.

Notes:

Troubleshooting

The following lists common problems and solutions for the RFID system.

Problem

I just hooked this unit up out of the box and cannot see the RFID interface in RSLinx.

Solution

The RFID interface is shipped with DHCP/BootP enabled and will not have an EtherNet IP address assigned unless the MAC address of the RFID is in the relationship list. There are three rotary switches on the RFID interface (all set to 0 by default), adjust the switches to a valid IP address in the range of 192.168.1.xxx where xxx is the position of the three rotary switches. Once the switches are in place, cycle power to the RFID interface.

Problem

I am getting a yellow triangle in RSLogix5000 for my RFID interface.

Solution

Open the properties of the RFID interface in RSLogix5000 and verify:

- The Inhibit Module box in the connection tab is not checked.
- The IP Address in the General Tab is the same as the IP address configured in the RFID interface.
- The IP Address of the RFID interface is on the same subnet as the EtherNet module in the Logix rack.

Also, verify the RFID interface has power by checking that the Aux Power LED is on solid green and the MOD LED is solid green, the Link 1 LED is flashing green, the NET LED is solid green.

Problem

My RFID channel[x] LED is flashing red on the interface.

Answer

Flashing red indicates no communications between the interface and the transceiver. Check cables between the RFID interface and transceiver. Ensure the power LED on the transceiver is green.

Problem

When I put a tag in the RFID field the LED on my transceiver and interface turns amber.

Answer

When one or more RFID tags are detected in the field, the LEDs on the interface and transceiver will turn amber indicating tag presence. When no tags are detected the LEDs will turn green indicating no tags detected but communications are healthy.

Problem

When I put a tag in the RFID field the power LED on the transceiver is solid green, the R/W Status LED is solid green, and the LED indicator for that channel is solid green.

Answer

Verify that the RFID tag is an ICODE compatible or SL2 style tag. The RFID interface may not be able to detect proprietary tag types.

Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products.
At http://www.rockwellautomation.com/support/, you can find technical manuals, a knowledge base of FAQs, technical and application notes, sample code and links to software service packs, and a MySupport feature that you can customize to make the best use of these tools.

For an additional level of technical phone support for installation, configuration, and troubleshooting, we offer TechConnect[SM] support programs. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/support/.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator at http://www.rockwellautomation.com/support/americas/phone_en.html, or contact your local Rockwell Automation representative.

New Product Satisfaction Return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.

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

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

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

Publication 56RF-UM001A-EN-P - October 2011