RFID Systems

Bulletin Number 56RF

Allen-Bradley
by ROCKWELL AUTOMATION

User Manual

Original Instructions

Important User Information

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

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

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

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

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

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

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

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

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

⚠ **ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

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

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

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

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

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

The following icon may appear in the text of this document.

💡 Identifies information that is useful and can help to make a process easier to do or easier to understand.

Table of Contents

Chapter 1

Introduction

Chapter 2

RFID Components

Chapter 3

Electrical Installation

Chapter 4

EtherNet/IP Addressing

Chapter 5

Mechanical Installation

Chapter 6

Add Your RFID Interface Block to an RSLogix 5000 Program

Chapter 7

RSLogix 5000 Controller Tags

Chapter 8

Commands Summary

Chapter 9

RSLogix 5000 Code Examples

Chapter 10

SLC Code Examples

Chapter 11

MicroLogix 1400 Code Examples

Chapter 12

RFID Tag Speed

Chapter 13

RFID Interface Block Webpage

Appendix A

Error Codes for RFID Interface Block

Appendix B

CIP Information

Appendix C

Install the Add-on Profile

Appendix D

Troubleshooting

Notes:

Read this preface to familiarize yourself with the rest of the manual.

Summary of Changes

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

Topic	Page
Updated International Standard Compliance	9

Who Should Use this Manual

Use this manual if you are responsible for design, installation, programming, or troubleshooting of control systems that use Bulletin 56RF RFID products.

You must 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. This guide provides an example of steps to follow to configure and make functional a 56RF RFID system. The reader must refer to the appropriate user manuals for other details. You must use this manual to accomplish the following:

- Learn how to install and wire an example RFID system
- Install and configure the module in an RSLogix 5000® program
- Built a simple program to receive and transmit data to an RFID tag

Abbreviations

Abbreviation	Definition
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

Abbreviation	Definition
ISO	International Organization for Standardization
JTC	Joint Technical Committee
MAC address	Media Access Control (Ethernet) address
MACID	Media Access Control Identification
QD	Quick Disconnect
RFID	Radio Frequency Identification
SB	Subcommittee
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
High Frequency 13.56 MHz RFID EtherNet/IP Interface Block Installation Instructions, publication 56RF-IN008	Provides information that is required to install RFID interface blocks.
Bulletin 56RF RFID Square 40x40 mm Transceiver Installation Instructions, publication 56RF-IN009	Provides information that is required to install 40x40 mm transceivers.
Bulletin 56RF Rectangular 80x90 mm Transceiver Installation Instructions, publication 56RF-IN010	Provides information that is required to install 80x90 mm transceivers.
High Temperature ICODE Tag (High Frequency RFID 13.56 MHz) Installation Instructions, publication 56RF-IN011	Provides information that is required to install high temperature ICODE tags.
Bulletin 56RF RFID 30 mm Cylindrical Transceiver Installation Instructions, publication 56RF-IN013	Provides information that is required to install 30 mm cylindrical transceivers.
EtherNet/IP Network Devices User Manual, publication ENET-UM006	This manual describes how to use EtherNet/IP communication modules in Logix 5000™ control systems.
Allen-Bradley Industrial Automation Glossary, AG-7.1	A glossary of industrial automation terms and abbreviations.
EtherNet/IP Device Level Ring Application Guide, publication ENET-AP007	This publication describes DLR network operation, topologies, configuration considerations, and diagnostic methods.
Industrial Automation Wiring and Grounding Guidelines, publication 1770-4.1	Provides general guidelines for installing a Rockwell Automation® industrial system.
Product Certifications website, rok.auto/certifications	Provides declarations of conformity, certificates, and other certification details.

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

Introduction

RFID Defined

RFID (Radio Frequency Identification) is a method to communicate information from one point to another point by the use of electromagnetic waves (radio waves). RFID 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 the specific purchase order of a customer. You can attach a tag to the carton. Before 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 into the carton. If the tag does not require an item, 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 that is received, the packaging process adds items (or skips this step) and then writes to the tag that one or more items were added. The carton moves to the next filling station.

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

International Standard Compliance

ISO/IEC 15693 is an ISO standard for vicinity tags. Vicinity 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 a maximum read distance of 1…1.5 m (3.3…4.9 ft), depending on the transceiver. Library applications with large antennas are capable of these distances. Most industrial applications are less than 203.2 mm (8 in.) for a read/write range.

The ICODE compatible tags permit you to use lower-cost tags than proprietary systems currently provide. You can 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 that the card and reader interpret.

An AC field that is produced in the transceiver couples power to the tag. 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 the tag receives sufficient power, 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 that is known as pulse position modulation. The position of one pulse relative to a known reference point codes the value of a nibble or byte of data. This process 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 1 m (3.3 ft) from a coupler that can only transmit power within the limits that international radio frequency (RF) regulations permit.

A tag only responds when it receives a valid command that selects one 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 that are 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 subcarriers of around 450 kHz that are switched on and off to provide special-encoded data that the transceiver detects.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions:

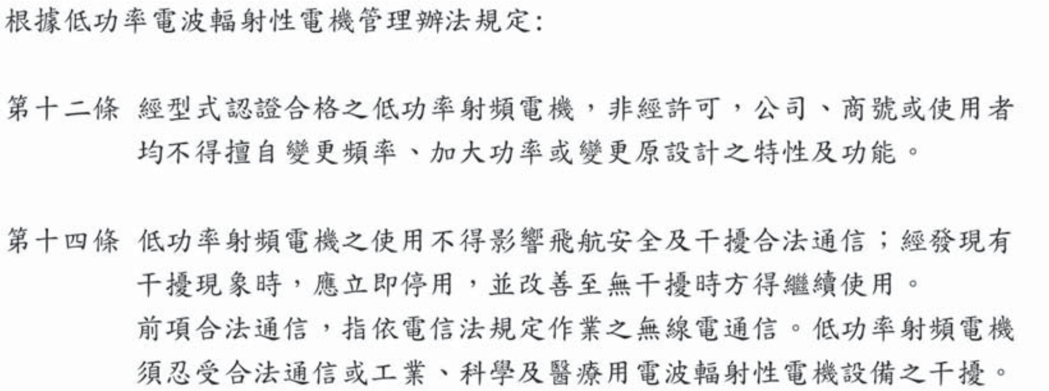
1. This device causes no harmful interference.
2. This device must accept any interference received, including interference that can cause undesired operation.

FCC Caution

Changes or modifications that are not expressly approved by the party responsible for compliance can void the authority of the user to operate the equipment.

IMPORTANT This equipment has been tested and found to comply with the limits for a Class A digital device, according to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, can cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user is required to correct the interference at their own expense. This device must be installed and operated keeping at minimum 200 mm (0.79 in.) away from bodies.

Taiwan NCC Warning Statement



根據低功率電波輻射性電機管理辦法規定：

第十二條 經型式認證合格之低功率射頻電機，非經許可，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。

第十四條 低功率射頻電機之使用不得影響飛航安全及干擾合法通信；經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。
前項合法通信，指依電信法規定作業之無線電通信。低功率射頻電機須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。

取得審驗證明之低功率射頻器材，非經核准，公司、商號或使用者均不得擅自變更頻率、加大功率或變更原設計之特性及功能。

低功率射頻器材之使用不得影響飛航安全及干擾合法通信 ；經發現有干擾現象時，應立即停用，並改善至無干擾時方得繼續使用。

前述合法通信，指依電信管理法規定作業之無線電通信。

低功率射頻器材須忍受合法通信或工業、科學及醫療用電波輻射性電機設備之干擾。

Date of manufacture (year/month) is written on package.

This device must be installed under 3m DC power cable. If the DC power cable of the radio and/or the ancillary equipment is less than or equal to 3 m (9.84 ft) in length, there is no need to conduct an emission test.

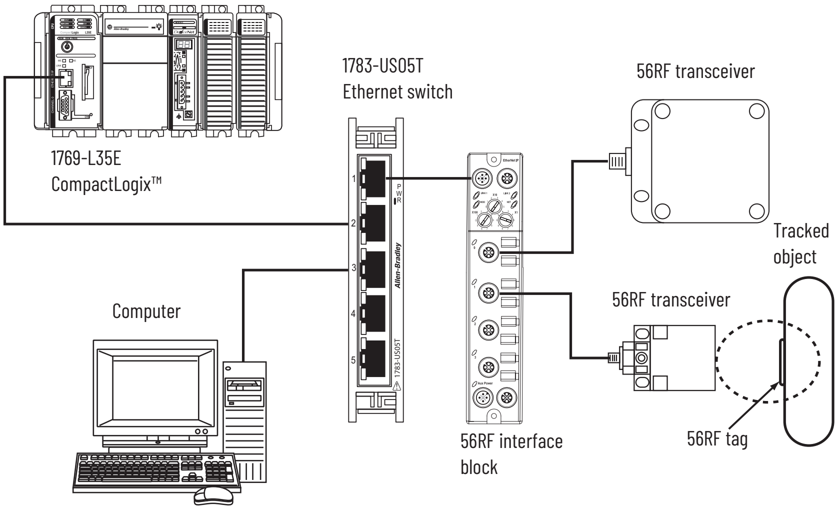
Backward Compatibility

The 56RF RFID system is offered on EtherNet/IP™ and is backward compatible with the previous offering of 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

Figure 1 shows a simple RFID system. This user manual describes the setup, installation, and programming that is required to get this system running.

Figure 1 - RFID System



Tags are attached to objects that must 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. See the transceiver installation instructions for antenna field sizes (see Additional Resources on page 8).

The transceivers are connected to a special RFID EtherNet/IP interface block. The distribution block has an Ethernet connection to an Ethernet switch. A 1759-L35E CompactLogix™ controller and a personal computer also have Ethernet connections to the Ethernet switch.

RFID Components

Interface Block

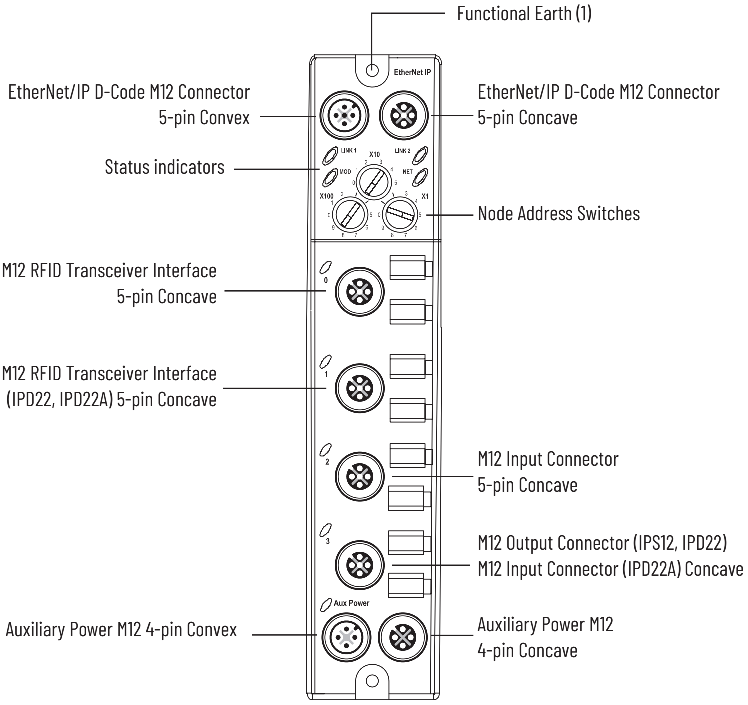
Three different interface blocks are available. Table 1 shows the type of ports for each catalog number.

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

Figure 2 identifies the connections for the EtherNet/IP™, RF transceivers, input devices, output devices, and power.

Figure 2 - Connections


Functional Earth (1)
EtherNet/IP D-Code M12 Connector 5-pin Convex
EtherNet/IP D-Code M12 Connector 5-pin Concave
Status indicators
Node Address Switches
M12 RFID Transceiver Interface 5-pin Concave
M12 RFID Transceiver Interface (IPD22, IPD22A) 5-pin Concave
M12 Input Connector 5-pin Concave
M12 Output Connector (IPS12, IPD22)
M12 Input Connector (IPD22A) Concave
Auxiliary Power M12 4-pin Convex
Auxiliary Power M12 4-pin Concave
EtherNet/IP
LINK 1
LINK 2
X10
MOD
NET
X100
X1
Aux Power

Status Indicators

When the status indicator is flashing, all flashes are 0.25 s on and 0.25 s off.

This block has the seven status indicators.

Figure 3 - Status Indicators

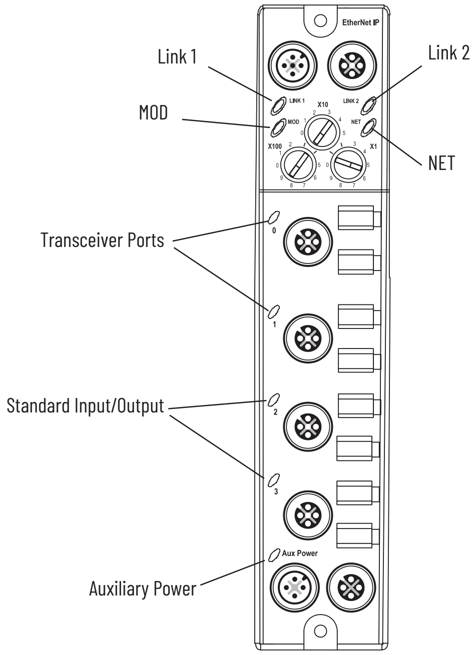


Table 2 - Status Indicators

Status Indicator Name	Status Indicator 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 and so on. Recommended solution is to remove electrical noise near cabling or reduce communication rate between transceiver and interface block.
	Red	The transceiver has an unrecoverable fault; can need replacing.
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 and so on. Recommended solution is to remove electrical noise near cabling or reduce communication rate between transceiver and interface block.
	Red	Duplicate IP address. The transceiver has an unrecoverable fault; can need replacing.

Table 2 - Status Indicators (Continued)

Status Indicator Name	Status Indicator State	Indicates
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.
	Steady green	The applied voltage is within specifications.
	Steady 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 is connected.
	Amber	Tag present

Transceivers

Status Indicators

Figure 4 - Indicators

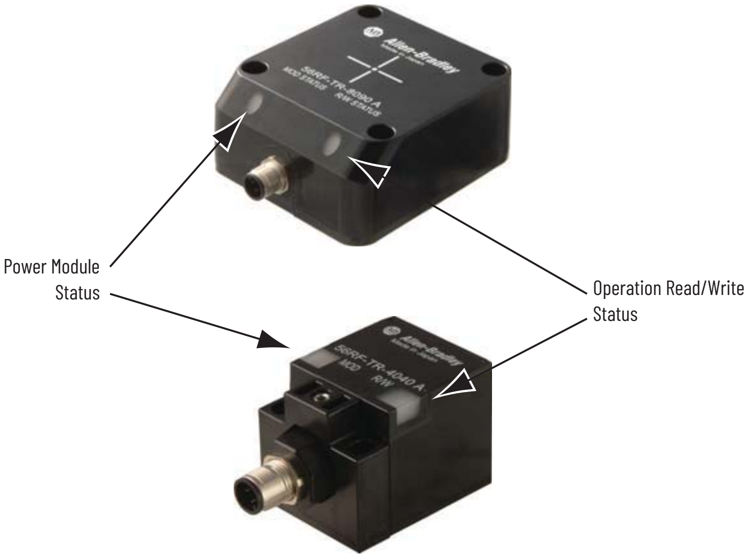


Table 3 - Status Indicators

Status Indicator Name	Status Indicator 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; can need replacing.
Read/Write Status	Off	There is no power applied to the device.
	Green	The EtherNet/IP 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[1]

[1] If a read/write command is not completed while the tag is within the field, an error occurs.

Transceiver Power-up Sequence

1. Both status indicators off.
2. Power status indicator turns green. R/W status turns green for 0.25 s.
3. R/W status indicator turns red for 0.25 s.
4. R/W status indicator turns off for 3…5 s.
5. R/W status indicator turns amber for 0.5 s.
6. R/W status indicator 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 bytes	112 bytes	28	4
SLI-S	256 bytes	160 bytes	40	4
SLI-L	64 bytes	32 bytes	8	4
FRAM	2048 bytes	2 kB	250	8

Tag Memory Structure

There are five types of tag memory structure:

- Universally Unique Identifier (UUID)
- Application Family Identifier (AFI)
- Data Storage Format Identifier (DSFID)
- Electronic Article Surveillance (EAS)
- Smart Label Integrated Circuit (SLI)

Universally Unique Identifier (UUID)

Each tag has a unique 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; numbering starts with the least significant bit (LSB) 1 and ends with the most significant bit (MSB) 64. This way is in contrast to the general used bit numbering within a byte (starts 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.

Table 5 shows the structure of our RFID tags.

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 1 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 Table 6. The least significant nibble of AFI is used to code one specific or all application subfamilies. Subfamily 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 subfamilies of family X	Wide applicative preselection
X	Y	Only the Yth subfamily of family X	—
0	Y	Proprietary subfamily Y only	—
1	0, Y	Transport	Mass transit, bus, airline
2	0, Y	Financial	IEP, banking, retail
3	0, Y	Identification	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 identifiers	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. The respective commands can program and lock it. It is coded on 1 byte. It allows for instant knowledge on the logical organization of the data.

Electronic Article Surveillance (EAS)

EAS is a technology that is typically used to help prevent shoplifting in retail establishments. An EAS detection system detects active tags and sets 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	Only change the EAS Configuration in a secure environment. The label must not be moved out of the communication field of the antenna during writing. We recommend putting 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 AFI at the inventory command and the two custom commands inventory read and fast inventory read (this feature allows, for example, the creation of label families).

This 8-bit value is at Byte 2 in Block -2 as shown in Table 9 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 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), that is, 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**
Condition	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**
Condition	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

Block -1																
Byte 2									**Byte 3**							
	MSB							**LSB**	**MSB**							**LSB**
Condition	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 Only change the Write Access conditions in a secure environment. The label must not be moved out of the communication field of the antenna during writing. We recommend putting 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. 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.

Table 12 on page 21 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. 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.

Table 13 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 nonvolatile 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 that is used for the writing and reading of FRAM data. The memory configuration of FRAM is shown in Table 14 on page 23.

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, which is defined as an area that can be accessed when the corresponding block address is specified. While Blocks FA_H…FF_H are system area, which 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 Table 14, 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 in Table and Table 16. 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 one 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).

Product Selection

Table 17...Table 22 on page 26 show the catalog numbers for the components in the Bulletin 56RF product family.

Main Components

Table 17 - 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

Table 18 - Transceivers

Dimensions [mm (in.)]	Recommended Sensing Distance [mm (in.)] [1]	Sensing Distance, Max [mm (in.)] [1]	Cat. No.
Rectangular [80 x 90 (3.14 x 3.54)]	100 (3.9)	168 (6.6)	56RF-TR-8090
Square [40 x 40 (1.57 x 1.57)]	50 (2)	85 (3.3)	56RF-TR-4040
Cylindrical M30	35 (1.4)	60 (2.4)	56RF-TR-M30
Cylindrical M18	18 (0.7)	30 (1.2)	56RF-TR-M18

[1] Range reference for a 50 mm (2 in.) diameter tag.

Table 19 - Tags

Outline	Type	Total Memory Size [B]	User Memory Size [B]	Dimensions [mm (in.)]	Cat. No.
Disc	SLI	128	112	16 (0.6)	56RF-TG-16
				20 (0.8)	56RF-TG-20
				30 (1.2)	56RF-TG-30
				50 (2)	56RF-TG-50
	SLI-S	64	32	16 (0.6)	56RF-TG-16-64B
	SLI-L	256	160	10 (0.4)	56RF-TG-10-256B
Disc – High Impact Resistant	SLI	128	112	35 (1.4)	56RF-TG-35HIR
Disc – Mount on Metal	SLI	128	112	20 (0.8)	56RF-TG-20MOM
				50 (2)	56RF-TG-50MOM
Disc – FRAM	FRAM	2048	2 kB	20 (0.8)	56RF-TG-20-2KB
				30 (1.2)	56RF-TG-30-2KB
				50 (2)	56RF-TG-50-2KB
Label	SLI	128	112	54 x 86 (2.1 x 3.4)	56RF-TG-5486
				50 x 50 (2 x 2)	56RF-TG-5050
Smart Cards	SLI	128		54 x 86 (2.1 x 3.4)	56RF-TG-5486SC
Square – High Temperature	SLI	128		50 x 50 (2 x 2)	56RF-TG-50HT

Accessories

Table 20 - Transceiver

Style	Connector Type	No. of Pins	Shield	Wire Size [mm^2 (AWG)]	Cat. No.
DC Micro (M12) Patchcords	Concave straight to convex straight	4	Shielded	0.34 (22)	889D-F5FCDM-Jx [1]
	Concave straight to convex right angle				889D-F5FCDE-Jx [1]
	Concave right angle to convex straight				889D-R5FCDM-Jx [1]
	Concave right angle to convex right angle				889D-R5FCDE-Jx [1]
DC Micro (M12) Patchcords	Concave straight	4	Shielded	0.34 (22)	889D-F5FC-Jx [2]
	Concave right angle				889D-R5FC-Jx [2]
	Convex straight				889D-M5FC-Jx [2]
	Convex right angle				889D-E5FC-Jx [2]
M12 Terminal Chambers	Concave straight	4	–	0.34…0.75 (22…18)	871A-TS5-D1
	Concave right angle				871A-TR5-D1
	Convex straight				871A-TS5-DM1
	Convex right angle				871A-TR5-DM1

[1] Replace *x* with 0M3 (0.3 m [1 ft]), 1 (1 m [3.3 ft]), 2 (2 m [6.6 ft]), 5 (5 m [16.4]), or 10 (10 m [32.8 ft]).
[2] Replace *x* with 2 (2 m [6.6 ft]), 5 (5 m [16.4]), or 10 (10 m [32.8 ft]).

Table 21 - Auxiliary Power

Style	Connector Type	No. of Pins	Shield	Wire Size [mm^2 (AWG)]	Cat. No.
DC Micro (M12) Patchcords	Concave straight to convex straight	4	Unshielded	0.34 (22)	889D-F4ACDM-x [1]
	Concave straight to convex right angle				889D-F4ACDE-x [1]
	Concave right angle to convex straight				889D-R4ACDM-x [1]
	Concave right angle to convex right angle				889D-R4ACDE-x [1]
DC Micro (M12) Patchcords	Concave straight	4	Unshielded	0.34 (22)	889D-F4AC-x [2]
	Concave right angle				889D-R4AC-x [2]
	Convex straight				889D-M4AC-x [2]
	Convex right angle				889D-E4AC-x [2]
M12 Terminal Chambers	Concave straight	4	–	0.34 (22)	871A-TS4-D
	Concave right angle				871A-TR4-D
	Convex straight				871A-TS4-DM
	Convex right angle				871A-TR4-DM

[1] Replace *x* with 0M3 (0.3 m [1 ft]), 1 (1 m [3.3 ft]), 2 (2 m [6.6 ft]), 5 (5 m [16.4]), or 10 (10 m [32.8 ft]).
[2] Replace x with 2 (2 m [6.6 ft]), 5 (5 m [16.4]), or 10 (10 m [32.8 ft]).

Table 22 - EtherNet/IP

Style	Connector Type	No. of Pins	Shield	Wire Size [mm^2 (AWG)]	Cat. No.
M12 D Code Patchcords	Convex straight to convex straight	4	Unshielded	0.25 (24)	1585D-M4TBDM-x [1]
	Convex straight to convex right angle				1585D-M4TBDE-x [1]
	Convex right angle to convex right angle				1585D-E4TBDE-x [1]
M12 D Code Patchcords	Convex straight to convex straight	4	Shielded	0.14 (26)	1585D-M4UBDM-x [1]
	Convex straight to convex right angle				1585D-M4UBDE-x [1]
	Convex right angle to convex right angle				1585D-E4UBDE-x [1]

[1] Replace x with 0M3 (0.3 m [1 ft]), 1 (1 m [3.3 ft]), 2 (2 m [6.6 ft]), 5 (5 m [16.4]), 10 (10 m [32.8 ft]), or 15 (15 m [49.2 ft]). Increments of 5 m (16.4 ft) up to 75 m (246.1 ft) are also available.

Chapter 3

Electrical Installation

Cable Overview

The Ethernet switch must be mounted inside a control panel. The Bulletin 56RF interface block and Bulletin 56RF transceivers can be mounted on the machine.

Figure 5 - Transceiver Mounting

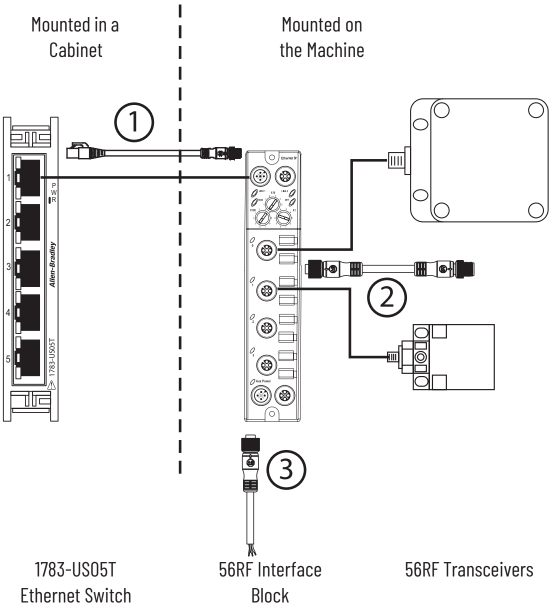


Figure 5 shows the three types of cables that are needed.

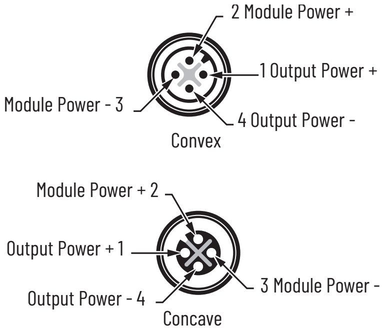
1. An Ethernet cable, RJ45 to M12-QD patchcord.
2. A 5-pin M12 to 5-pin M12 patchcord. The cable includes a shield that connects to the functional earth point on the interface block.
3. A 4-pin concave micro QD cordset that connects power to the interface block.

Auxiliary Power Connection

Attach a micro-style 4-pin concave to the micro-style 4-pin convex receptacle as shown in Figure 6 on page 28. The concave 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 total power that is consumed by each block determines the maximum number of interface blocks that can be connected.

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

Figure 6 - Pin Connections for the Aux Power Connectors



The power for the output port is separate from the power to the remaining portions of the interface block. This configuration 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.

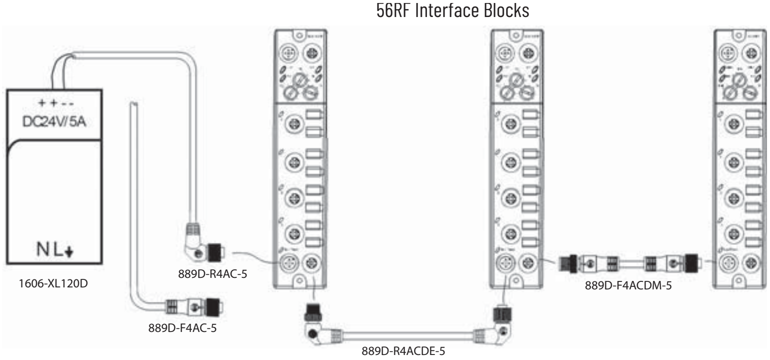
Power Connection Options

Each interface block is limited to 4 A total consumption.

Example 1: Daisy Chain the Power Connections

The example in Figure 7 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.

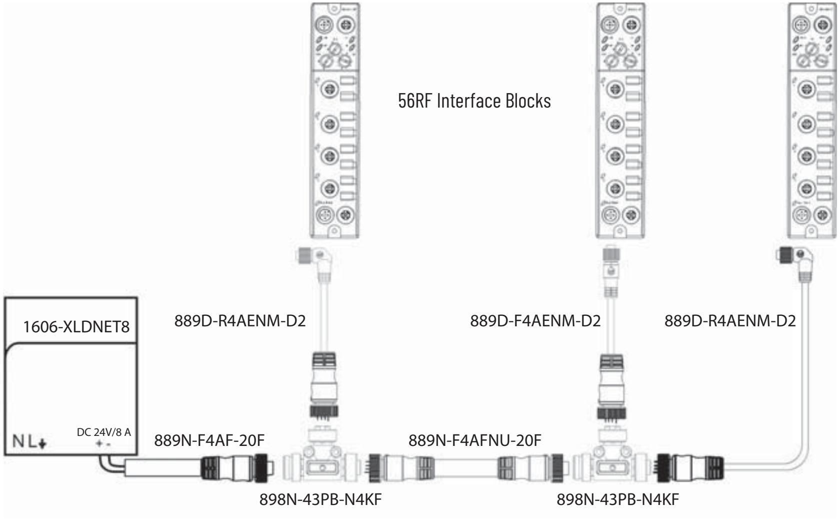
Figure 7 - Power Option 1



Example 2: System Needs More Than 4 A

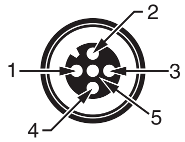
If multiple blocks are required on a machine and the current consumption exceeds 4 A, then a combination of mini-style and micro-style connections can be used to distribute the power. In the example shown in Figure 8 on page 29, mini-style cordsets, patchcords, and tees are used to configure the power. A mini-to-micro style patchcord connects each 56RF interface block with the tee. In this example, the power supply is a catalog number 1606-XLDNET8, which can supply up to 8 A to the RFID system.

Figure 8 - Power Option 2



Transceiver Connection

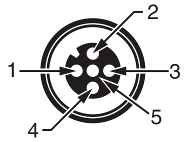
The following shows the M12 QD concave connector for the transceivers. Pin 5 is the cable shield connection and is connected only at the block to functional earth (FE).



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

Digital Input Connection

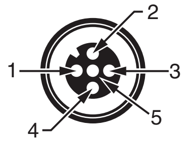
The following shows the concave M12 QD input connector.



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

Digital Output Connection

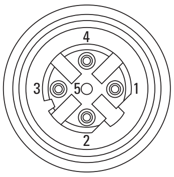
The following shows the concave M12 QD output connector.



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

EtherNet/IP Connection

The following shows the D-Code M12 connector on the interface block.



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

Use the catalog number 1585D-M4DC-H (polyamide small body unshielded) or catalog number 1585D-M4DC-SH (zinc die-cast large body shielded) mating connectors for the D-Code M12 concave 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 the following network topologies:

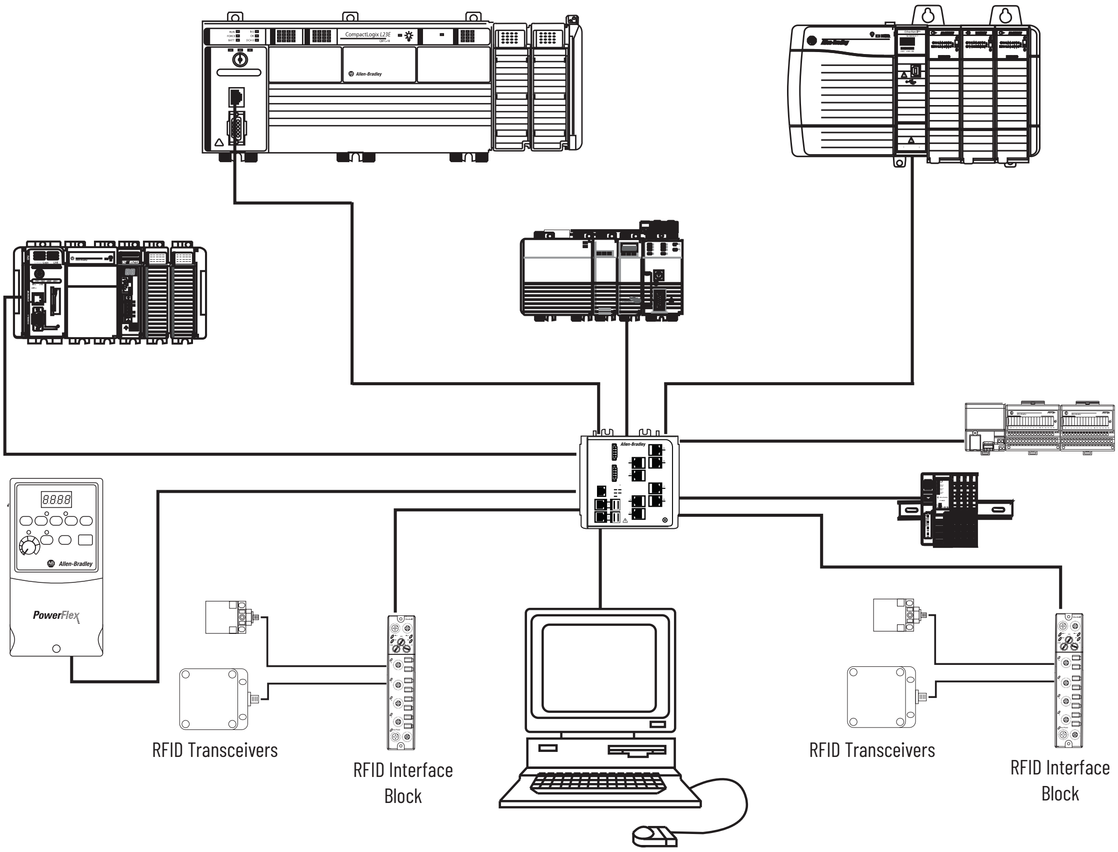
- Star Topology on page 31
- Linear Topology on page 31
- Device Level Ring (DLR) Topology on page 32

EtherNet/IP Addressing

Star Topology

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

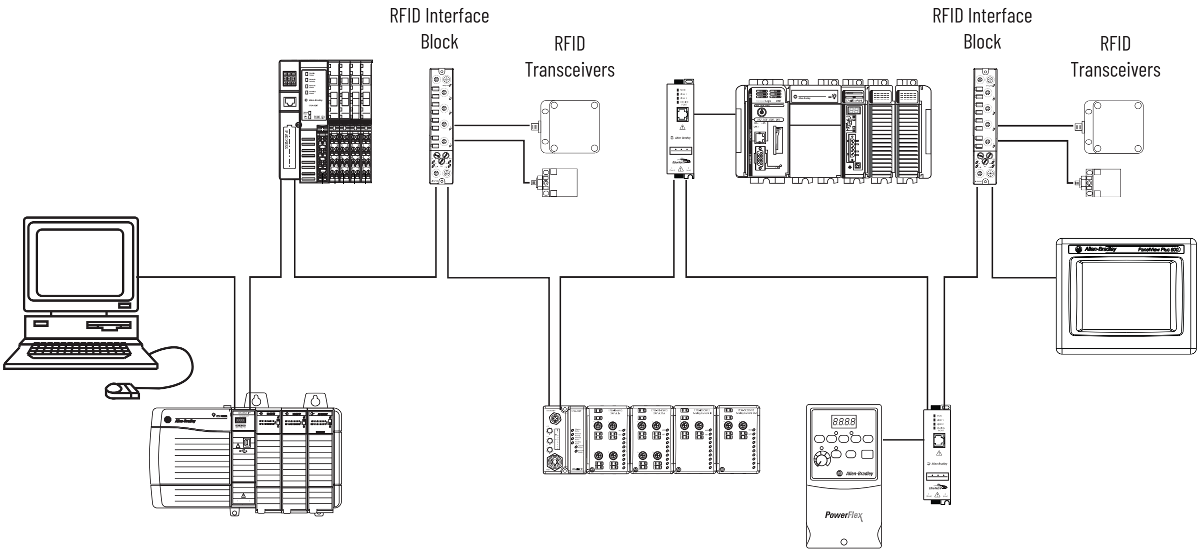
Figure 9 - Star Topology



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 that is 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.

We recommend that you use no more than 50 nodes on one 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.

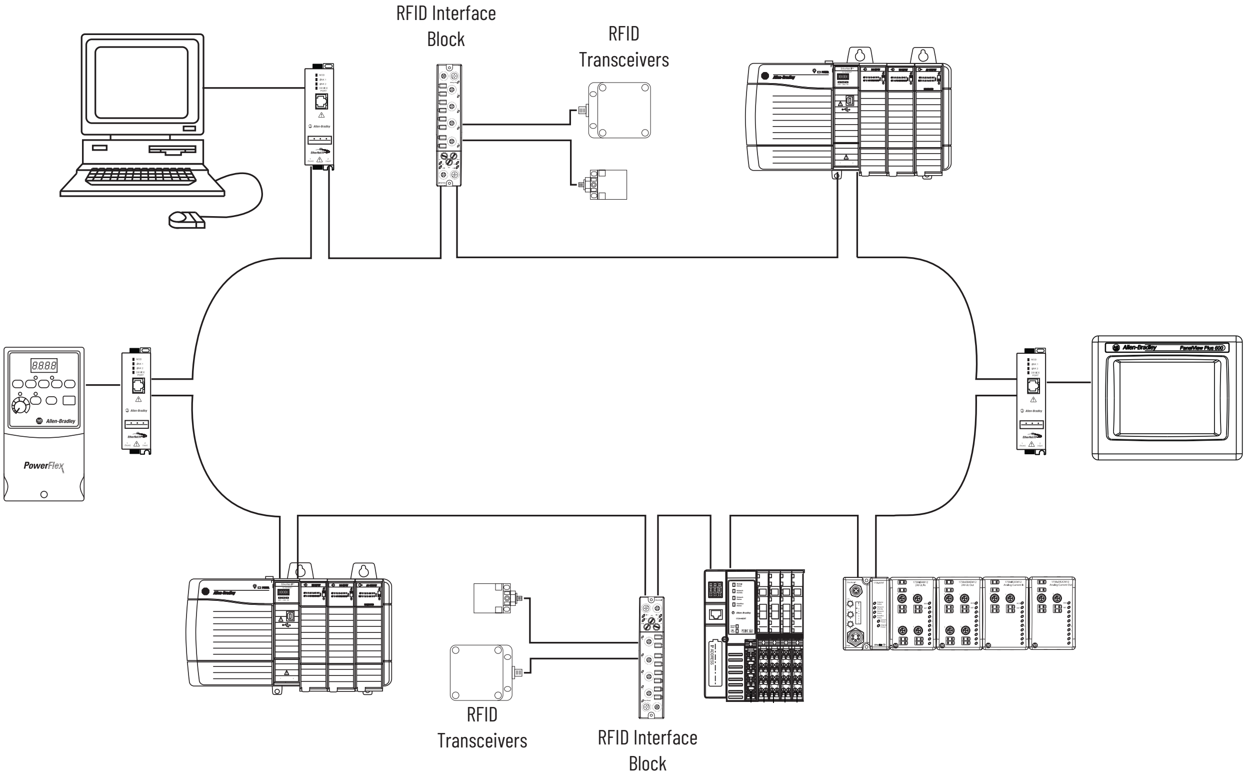
Smaller networks provide the following benefits:

- 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 DLR network design and configuration, see publication ENET-AP007.

Figure 11 - DLR Topology



Setting the Network Address

Before using the 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 BootP/DHCP utility (version 2.3 or greater), which ships with RSLogix 5000® software.
- Use RSLinx® software.

IP network addresses have a format of xxx.xxx.xxx.xxx. You 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 software to change the address.

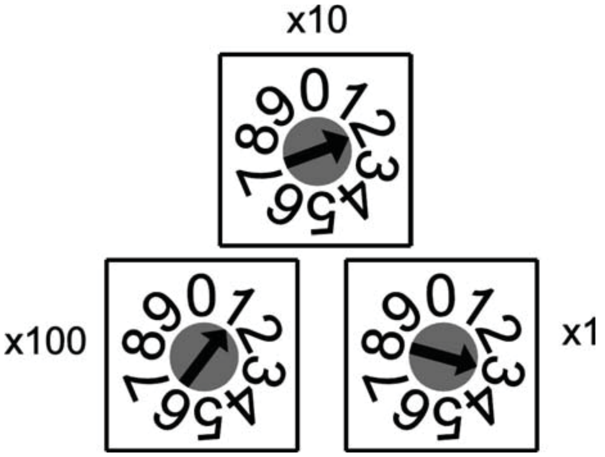
Fundamental IP Addresses: 192.168.1.*xxx*

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 screwdriver to rotate the switches. Align 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.

Figure 12 on page 34 shows an address setting of 192.168.1.123.

Figure 12 - Address Setting Example



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 an assigned host name or use a Domain Name Server (DNS).

Advanced IP Addresses

Step 1…Step 6 show how to change the IP address from the fundamental 192.168.1.*xxx* to an advanced address. This procedure assumes that the 56RF interface block was already configured with an IP address using the network address switches. The following examples show the change process using specific addresses. You are not limited to these addresses; you can select any address that meets their needs. In the following example, we change from 192.168.1.115 to 192.168.2.115.

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

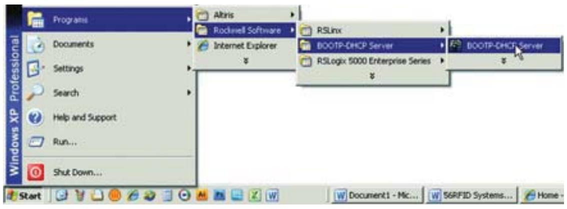
   On the 56RF interface block, the address switches had previously been to 115. Set the address switch settings to 888. Cycle the power and wait until the MOD indicator is flashing red. The MOD indicator flashes red once, green once, then steady red for a short while, then flashes green once, and finally flashes red continuously (about once each second). This process 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.

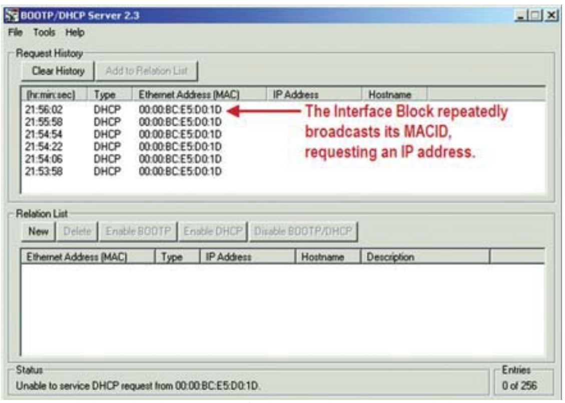
   On the 56RF interface block, set the address switch settings to 999. Cycle the power and wait until the MOD indicator is steady green. The MOD indicator flashes red once, green once, steady red for a short while and finally turns steady green. This process takes about 10 seconds after power is restored.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

3. Use BootP/DHCP Server to set new valid address**.**

   a. From the Start menu, select Programs > Rockwell Software > BOOTP-DHCP Server > BOOTP-DHCP Server.



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



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
21:56:02	DHCP	00:00:BC:E5:D0:1D		
21:55:58	DHCP	00:00:BC:E5:D0:1D		
21:54:54	DHCP	00:00:BC:E5:D0:1D		
21:54:22	DHCP	00:00:BC:E5:D0:1D		
21:54:06	DHCP	00:00:BC:E5:D0:1D		
21:53:58	DHCP	00:00:BC:E5:D0:1D		

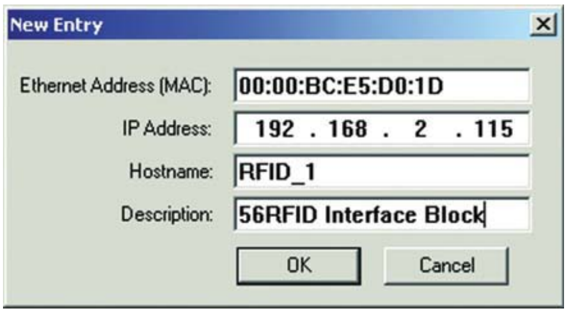
The Interface Block repeatedly broadcasts its MACID, requesting an IP address.

Ethernet Address (MAC)	Type	IP Address	Hostname	Description

Status: Unable to service DHCP request from 00:00:BC:E5:D0:1D.

Entries: 0 of 256

c. Double-click one of the Ethernet addresses (MAC) of the device. The New Entry dialog appears, which shows the Ethernet address (MAC) of the device.



Ethernet Address (MAC): 00:00:BC:E5:D0:1D

IP Address: 192 . 168 . 2 . 115

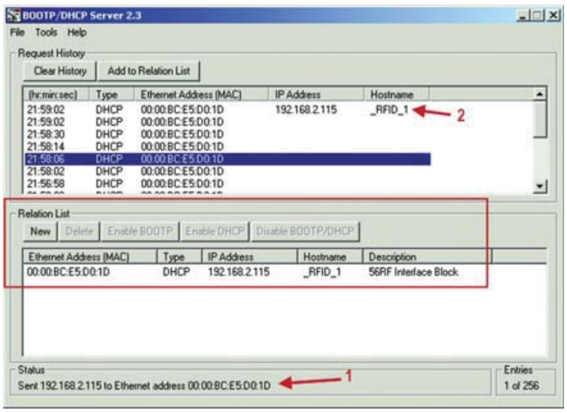
Hostname: RFID_1

Description: 56RFID Interface Block

d. Type in the IP address, host name, and description and click OK. The host name and description are optional fields; they can be left blank.

The device is added to the Relation List, which displays the Ethernet address (MAC) and corresponding IP address, host name, and description.

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



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
21:59:02	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1
21:59:02	DHCP	00:00:BC:E5:D0:1D		
21:58:30	DHCP	00:00:BC:E5:D0:1D		
21:58:14	DHCP	00:00:BC:E5:D0:1D		
21:58:06	DHCP	00:00:BC:E5:D0:1D		
21:58:02	DHCP	00:00:BC:E5:D0:1D		
21:56:58	DHCP	00:00:BC:E5:D0:1D		

Ethernet Address (MAC)	Type	IP Address	Hostname	Description
00:00:BC:E5:D0:1D	DHCP	192.168.2.115	_RFID_1	56RF Interface Block

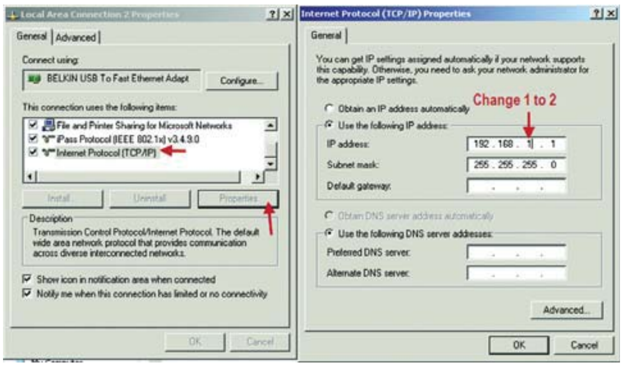
Status: Sent 192.168.2.115 to Ethernet address 00:00:BC:E5:D0:1D

Entries: 1 of 256

IMPORTANT Wait for the Status message to show “Sent 192.168.2.115 to Ethernet address 00:00:BC:E5:D0:1D.” This process can take up to 30 seconds.

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

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



Connect using: BELKIN USB To Fast Ethernet Adapt

This connection uses the following items:

☑ File and Printer Sharing for Microsoft Networks
☑ iPass Protocol (IEEE 802.1x) v3.4.9.0
☑ Internet Protocol (TCP/IP)

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☑ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

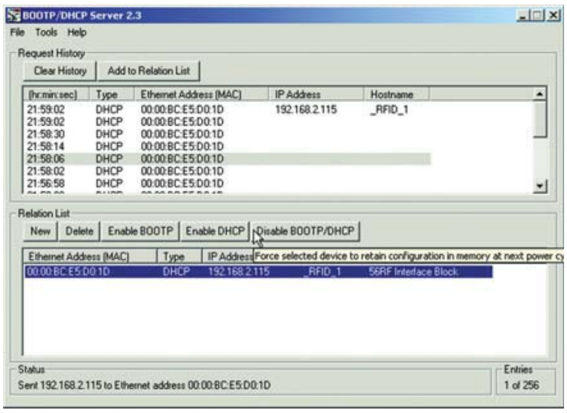
Change 1 to 2

IP address: 192 . 168 . 1 . 1
Subnet mask: 255 . 255 . 255 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

6. Disable DHCP.
   a. Set the rotary switches to 0 0 0.



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
21:59:02	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1
21:59:02	DHCP	00:00:BC:E5:D0:1D		
21:58:30	DHCP	00:00:BC:E5:D0:1D		
21:58:14	DHCP	00:00:BC:E5:D0:1D		
21:58:06	DHCP	00:00:BC:E5:D0:1D		
21:58:02	DHCP	00:00:BC:E5:D0:1D		
21:56:58	DHCP	00:00:BC:E5:D0:1D		

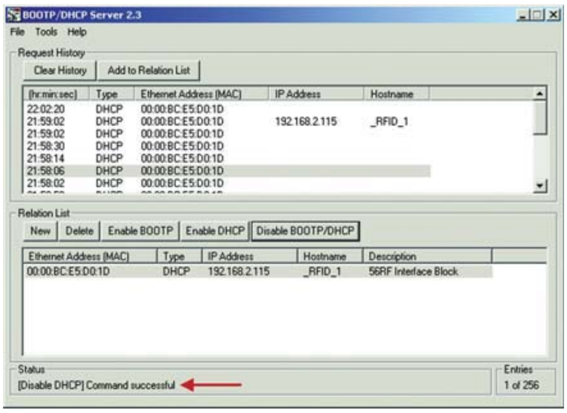
Ethernet Address (MAC)	Type	IP Address		
00:00:BC:E5:D0:1D	DHCP	192.168.2.115	_RFID_1	56RF Interface Block

Status: Sent 192.168.2.115 to Ethernet address 00:00:BC:E5:D0:1D

Entries: 1 of 256

   b. To highlight the interface block, single-click it in the Relation List.
   c. Then, click Disable BOOTP/DHCP. This action instructs the 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.



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
22:02:20	DHCP	00:00:BC:E5:D0:1D		
21:59:02	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1
21:59:02	DHCP	00:00:BC:E5:D0:1D		
21:58:30	DHCP	00:00:BC:E5:D0:1D		
21:58:14	DHCP	00:00:BC:E5:D0:1D		
21:58:06	DHCP	00:00:BC:E5:D0:1D		
21:58:02	DHCP	00:00:BC:E5:D0:1D		

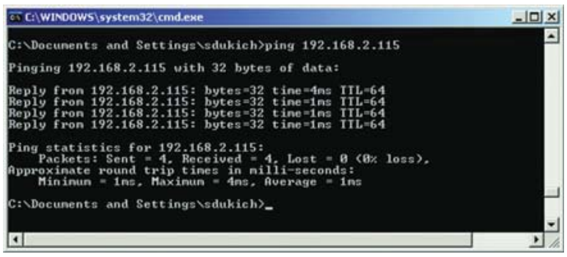
Ethernet Address (MAC)	Type	IP Address	Hostname	Description
00:00:BC:E5:D0:1D	DHCP	192.168.2.115	_RFID_1	56RF Interface Block

Status: [Disable DHCP] Command successful

Entries: 1 of 256

   d. Click File > Save As to save the relationship, if desired.
   e. Cycle the power to the 56RF interface block. You no longer see the 56RF interface block in the Request History panel.

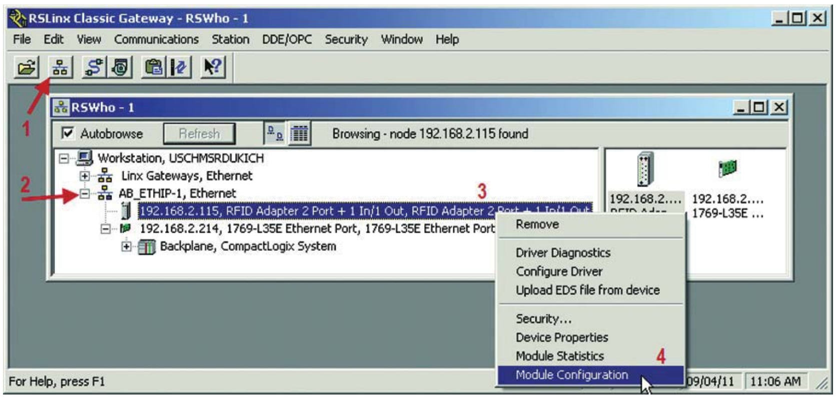
f. From a DoS prompt, you can ping the new address. The response must be four packets sent, four packets received, and zero lost.



```
C:\Documents and Settings\sdukich>ping 192.168.2.115

Pinging 192.168.2.115 with 32 bytes of data:

Reply from 192.168.2.115: bytes=32 time=4ms TTL=64
Reply from 192.168.2.115: bytes=32 time=1ms TTL=64
Reply from 192.168.2.115: bytes=32 time=1ms TTL=64
Reply from 192.168.2.115: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.2.115:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 4ms, Average = 1ms

C:\Documents and Settings\sdukich>_
```

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 56RF interface block are set to 999, and the address has been previously set using the BootP/DHCP server. The following example shows how to change the IP address from 192.168.2.115 to 192.168.3.115.



☑ Autobrowse   Refresh   Browsing - node 192.168.2.115 found

- Workstation, USCHMSRDUKICH
  - Linx Gateways, Ethernet
  - AB_ETHIP-1, Ethernet
    - 192.168.2.115, RFID Adapter 2 Port + 1 In/1 Out, RFID Adapter 2 Port + 1 In/1 Out
    - 192.168.2.214, 1769-L35E Ethernet Port, 1769-L35E Ethernet Port
      - Backplane, CompactLogix System

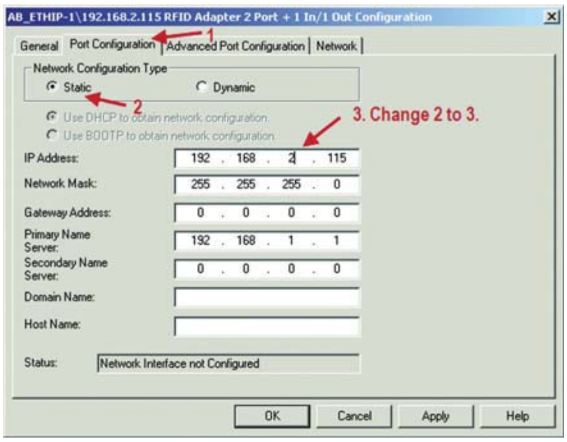
For Help, press F1   09/04/11   11:06 AM

After you open RSLinx:

1. Click the RS-Who icon.
2. Expand the Ethernet connection.
3. Right-click the RFID Adapter.
4. Click Module Configuration.

After the Configuration window appears:

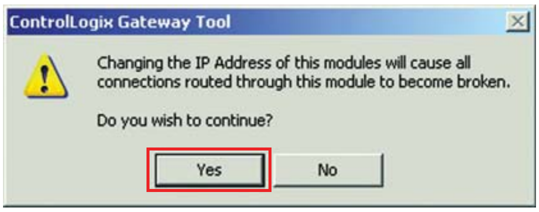
1. Click the Port Configuration tab.
2. Set the Network Configuration Type to Static (if not already done).
3. Change the IP address to the new address. In this example, the address is changed from 192.168.2.115 to 192.168.3.115.


AB_ETHIP-1\192.168.2.115 RFID Adapter 2 Port + 1 In/1 Out Configuration
General
Port Configuration
Advanced Port Configuration
Network
1
2
3. Change 2 to 3.
OK
Cancel
Apply
Help

Network Configuration Type
● Static   ○ Dynamic
● Use DHCP to obtain network configuration.
○ Use BOOTP to obtain network configuration.

IP Address:	192 . 168 . 2 . 115
Network Mask:	255 . 255 . 255 . 0
Gateway Address:	0 . 0 . 0 . 0
Primary Name Server:	192 . 168 . 1 . 1
Secondary Name Server:	0 . 0 . 0 . 0
Domain Name:	
Host Name:	
Status:	Network Interface not Configured

a. Click Yes to confirm the change.

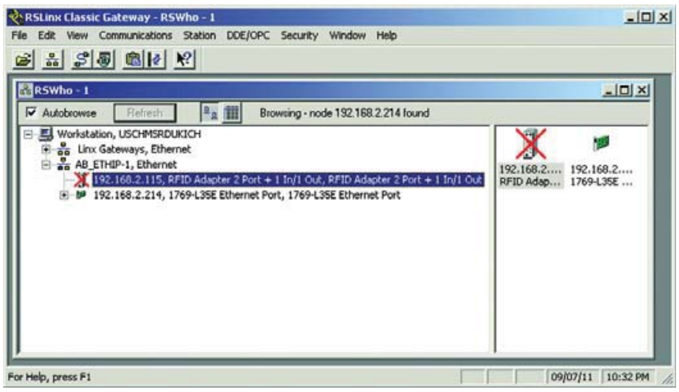


Changing the IP Address of this modules will cause all connections routed through this module to become broken.

Do you wish to continue?

b. To close the configuration window, click OK.

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



RSLinx Classic Gateway - RSWho - 1
File Edit View Communications Station DDE/OPC Security Window Help
RSWho - 1
Autobrowse
Refresh
192.168.2.... RFID Adap...
192.168.2.... 1769-L35E ...
For Help, press F1
09/07/11
10:32 PM

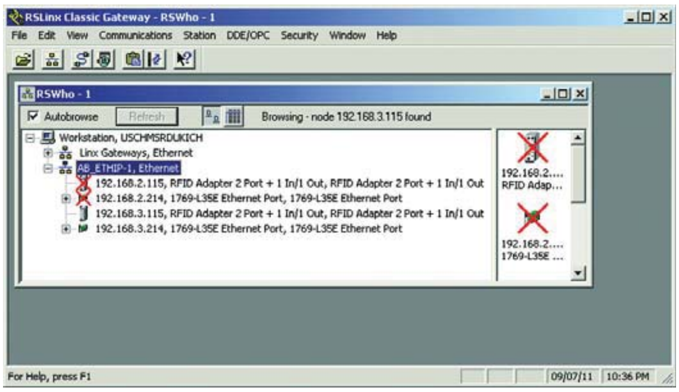
Browsing - node 192.168.2.214 found

- Workstation, USCHMSRDUKICH
  - Linx Gateways, Ethernet
  - AB_ETHIP-1, Ethernet
    - 192.168.2.115, RFID Adapter 2 Port + 1 In/1 Out, RFID Adapter 2 Port + 1 In/1 Out
    - 192.168.2.214, 1769-L35E Ethernet Port, 1769-L35E Ethernet Port

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

Change the Network adapter address to 192.168.3.1.

d. Close and reopen the RSWho window. The older addresses are not available and the new addresses (192.168.3.115 and 192.168.3.214) appear.



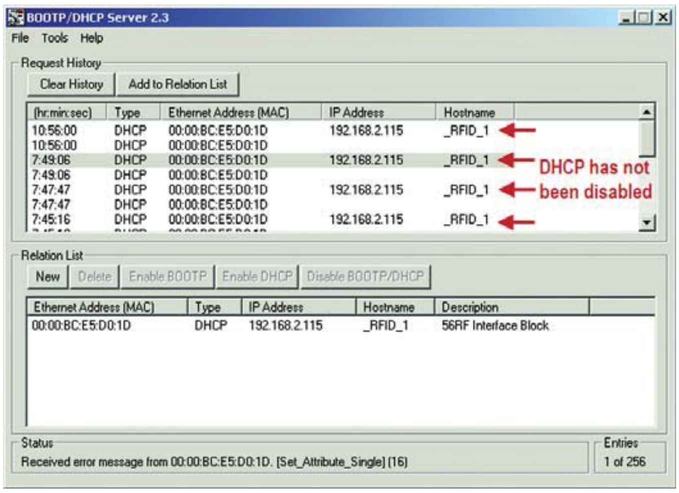
RSLinx Classic Gateway - RSWho - 1
File Edit View Communications Station DDE/OPC Security Window Help
RSWho - 1
Autobrowse
Refresh
192.168.2.... RFID Adap...
192.168.2.... 1769-L35E ...
For Help, press F1
09/07/11
10:36 PM

Browsing - node 192.168.3.115 found

- Workstation, USCHMSRDUKICH
  - Linx Gateways, Ethernet
  - AB_ETHIP-1, Ethernet
    - 192.168.2.115, RFID Adapter 2 Port + 1 In/1 Out, RFID Adapter 2 Port + 1 In/1 Out
    - 192.168.2.214, 1769-L35E Ethernet Port, 1769-L35E Ethernet Port
    - 192.168.3.115, RFID Adapter 2 Port + 1 In/1 Out, RFID Adapter 2 Port + 1 In/1 Out
    - 192.168.3.214, 1769-L35E Ethernet Port, 1769-L35E Ethernet Port

IMPORTANT	If DHCP is not disabled, the 56RF interface block shows two requests in the DHCP Server at each 56RF interface block power-up.

In the following example, power was cycled to the 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 56RF interface block notified the BootP/DHCP server of its IP address, which indicates that DHCP has not been disabled. If DHCP is disabled, the 56RF interface block would show nothing.



(hr:min:sec)	Type	Ethernet Address (MAC)	IP Address	Hostname
10:56:00	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1
10:56:00	DHCP	00:00:BC:E5:D0:1D		
7:49:06	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1
7:49:06	DHCP	00:00:BC:E5:D0:1D		
7:47:47	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1
7:47:47	DHCP	00:00:BC:E5:D0:1D		
7:45:16	DHCP	00:00:BC:E5:D0:1D	192.168.2.115	_RFID_1

Ethernet Address (MAC)	Type	IP Address	Hostname	Description
00:00:BC:E5:D0:1D	DHCP	192.168.2.115	_RFID_1	56RF Interface Block

Received error message from 00:00:BC:E5:D0:1D. [Set_Attribute_Single] (16)

1 of 256

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 flashes the following pattern: flashes red once, green once, then steady red, then flashes green once, and final flashes continuous red about once each second. The reset process takes about 10 seconds.

Notes:

Chapter 5

Mechanical Installation

Each transceiver generates a similar but unique RF field.

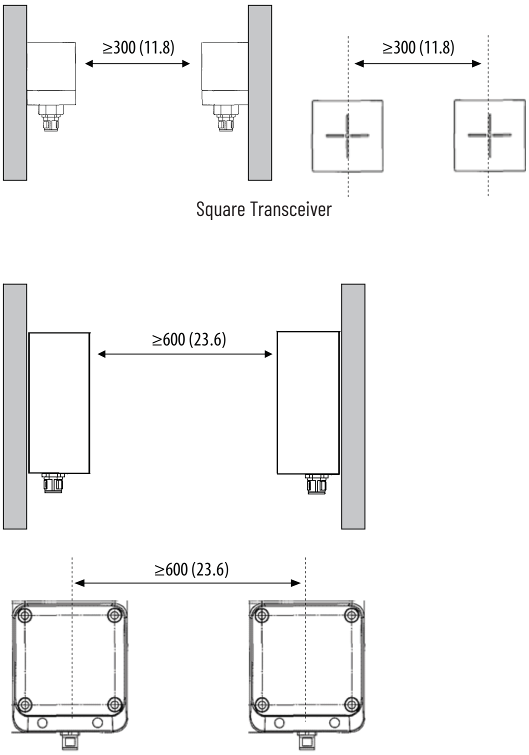
Fastening

Attach the transceiver to the flat plate with M5 screws. The tightening torque must be 1.5 N•m (13.3 lb•in) for the M5 screw.

Spacing Between Transceivers

The installation of multiple transceivers causes radio frequency interference and can result in tag communication difficulty. Keep a sufficient distance between the transceivers as shown in Figure 13.

Figure 13 - Spacing Between Transceivers [mm (in.)]



Square Transceiver

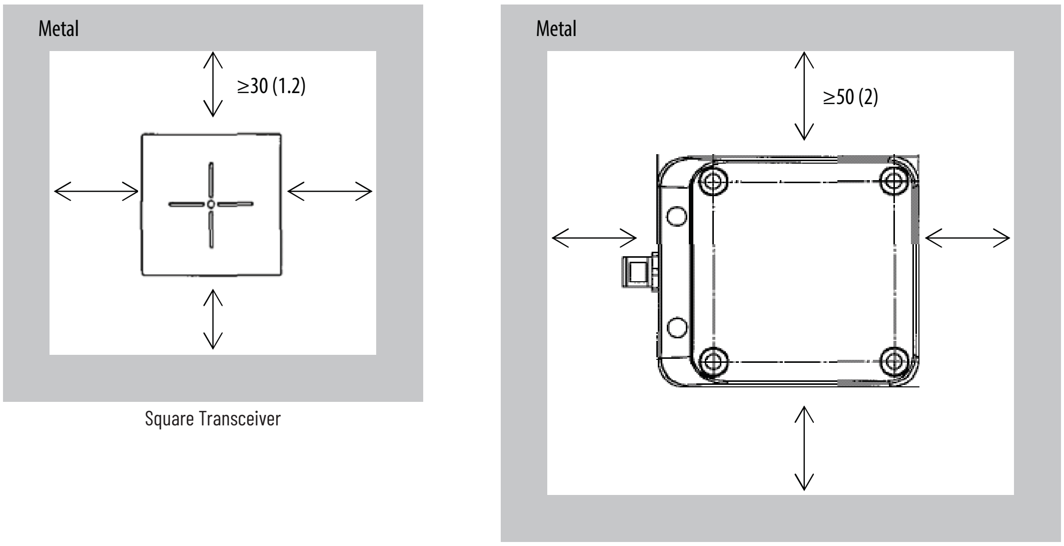
Rectangular Transceiver

Spacing Next to Metal Surfaces

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

Figure 14 - Transceiver Spacing with Metal Surfaces



Square Transceiver

Rectangular Transceiver

Transceiver Field Maps

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

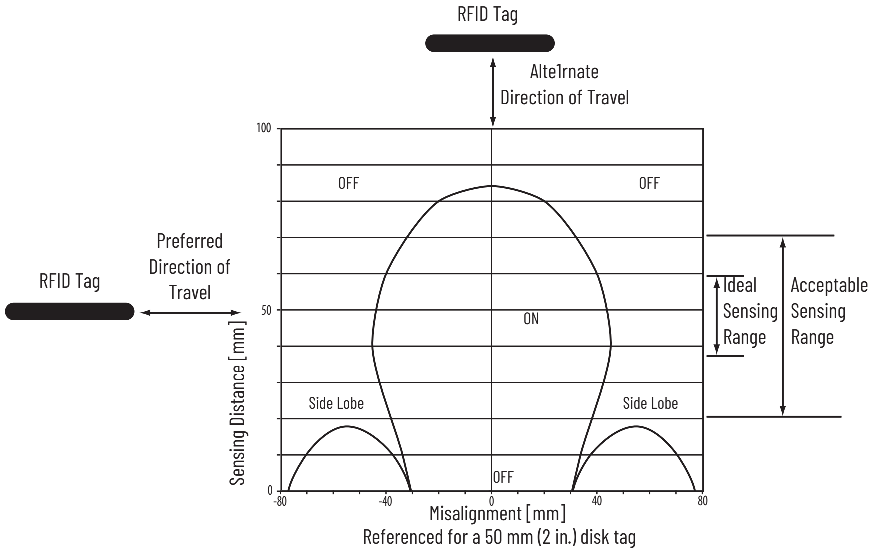
The RF tags must enter the RF field once, stay long enough to complete the read and write cycles, and then to leave the field smoothly and efficiently.

Ideally, the RFID tag must pass through the widest section of the main lobe. This arrangement maximizes the time that 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.

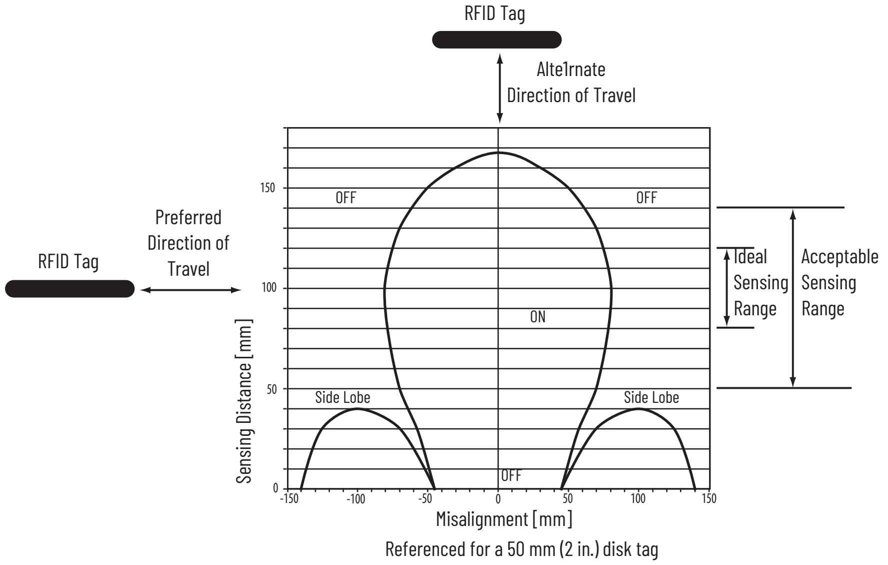
Figure 15 on page 43 shows the field map of the 65 x 65 mm (2.6 x 2.6 in.) transceiver.

Figure 15 - 65 x 65 mm (2.6 x 2.6 in.) Transceiver



The field map for the 80 x 90 mm (3.1 x 3.5 in.) transceiver, which is shown in Figure 16, is similar.

Figure 16 - 80 x 90 mm (3.1 x 3.5 in.) Transceiver



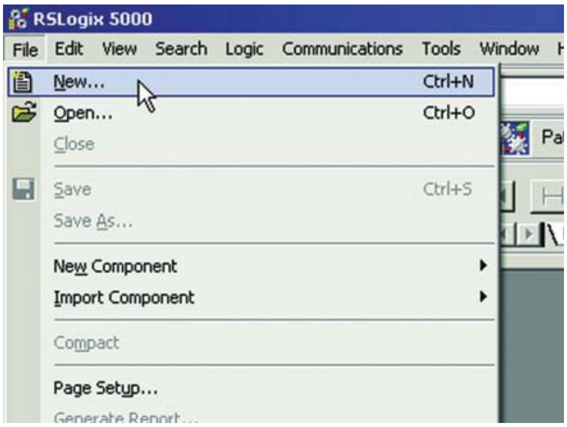
Notes:

Chapter 6

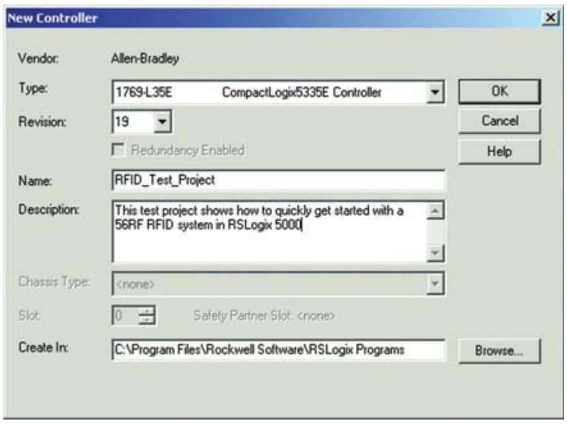
Add Your RFID Interface Block to an RSLogix 5000 Program

Procedure

1. Open RSLogix 5000® software.
2. Click File>New.



3. Enter the new controller information.



Vendor: Allen-Bradley

Type: 1769-L35E CompactLogix5335E Controller

Revision: 19

□ Redundancy Enabled

Name: RFID_Test_Project

Description: This test project shows how to quickly get started with a 56RF RFID system in RSLogix 5000

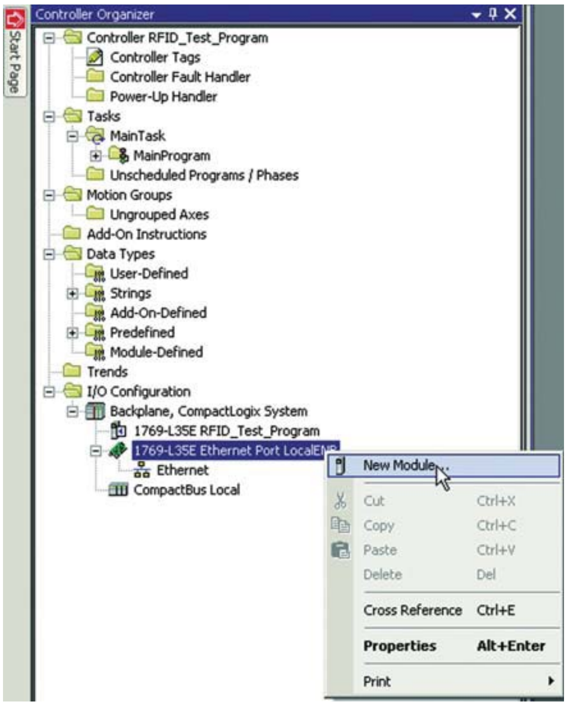
Chassis Type: <none>

Slot: 0 Safety Partner Slot: <none>

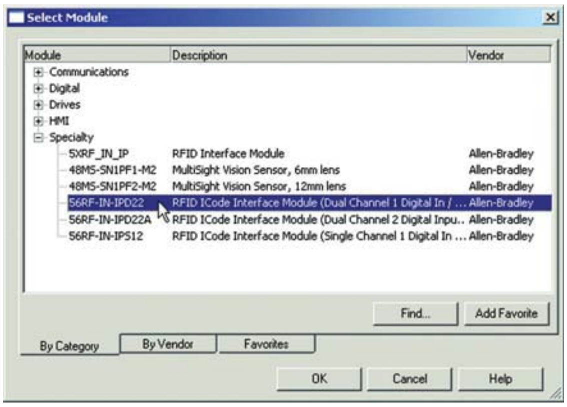
Create In: C:\Program Files\Rockwell Software\RSLogix Programs

4. Right-click the Ethernet port of the controller.

5. Click New Module.



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

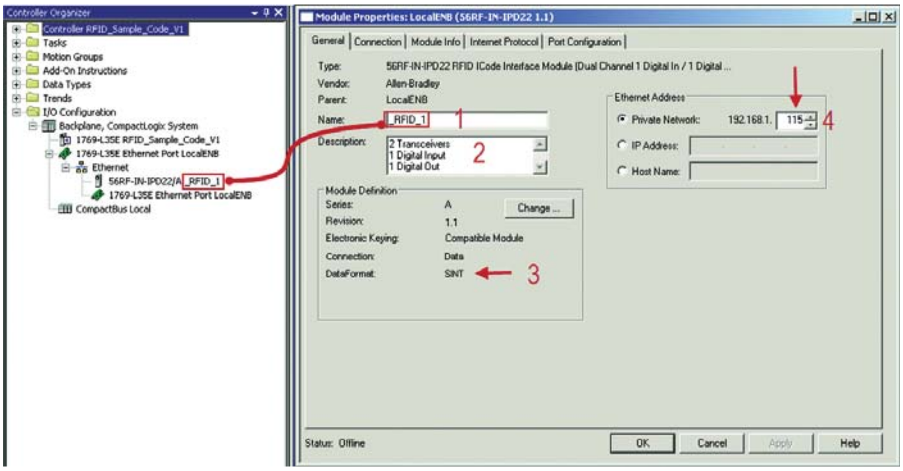


Module	Description	Vendor
Communications		
Digital		
Drives		
HMI		
Specialty		
5XRF_IN_IP	RFID Interface Module	Allen-Bradley
48MS-SN1PF1-M2	MultiSight Vision Sensor, 6mm lens	Allen-Bradley
48MS-SN1PF2-M2	MultiSight Vision Sensor, 12mm lens	Allen-Bradley
56RF-IN-IPD22	RFID ICode Interface Module (Dual Channel 1 Digital In / ...	Allen-Bradley
56RF-IN-IPD22A	RFID ICode Interface Module (Dual Channel 2 Digital Inpu..	Allen-Bradley
56RF-IN-IPS12	RFID ICode Interface Module (Single Channel 1 Digital In ...	Allen-Bradley

General Tab

The General tab describes the device, its definition, and its IP address. Make the changes that are shown Figure 17 and click Apply.

Figure 17 - General Tab



Type: 56RF-IN-IPD22 RFID ICode Interface Module (Dual Channel 1 Digital In / 1 Digital ...
Vendor: Allen-Bradley
Parent: LocalENB
Name: RFID_1
Description: 2 Transceivers
1 Digital Input
1 Digital Out

Module Definition
Series: A
Revision: 1.1
Electronic Keying: Compatible Module
Connection: Data
DataFormat: SINT

Ethernet Address
● Private Network: 192.168.1. 115
○ IP Address:
○ Host Name:

Status: Offline

1. Enter a name for the module. In this example, the name is RFID_1. You can 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 also appears in the description of the tags, which are described later.
2. Enter a description of the module or its function.
3. The Data Format can be left as SINT (preferred) or changed to INT (for compatibility with non-Rockwell Automation® RFID tags).

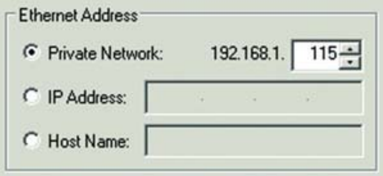
A SINT is a signed single-byte integer, which can represent numbers from -255…255 in decimal format (-F…FF in hexadecimal format). An INT is a signed 2 byte integer, which can represent numbers from -65535…65535 in decimal format (-FFFF…FFFF in hexadecimal format).

4. Set the MAC 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 Bulletin 56RF interface block.

MAC Address

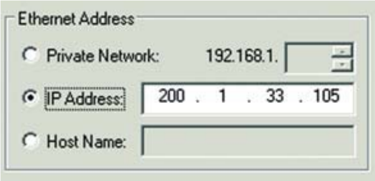
When the controller is offline, the MAC address can be set. You have three options.

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



Ethernet Address
● Private Network: 192.168.1. 115
○ IP Address:
○ Host Name:

- When multiple networks exist, you can elect to set the address to some other value. When offline, simply click IP address and enter the desired address.



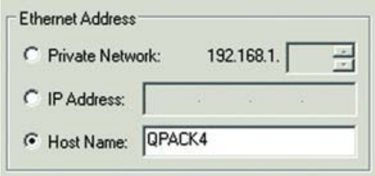
Ethernet Address

○ Private Network: 192.168.1.

● IP Address: 200 . 1 . 33 . 105

○ Host Name:

- Click Host Name and type in the name of the host. In the following example, the host name is QPACK4.



Ethernet Address

○ Private Network: 192.168.1.

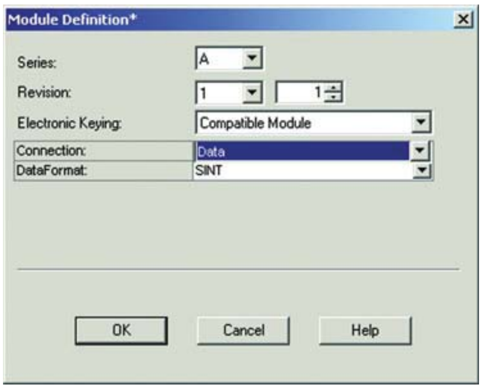
○ IP Address:

● Host Name: QPACK4

Module Definition

You do not need to change the default values. If necessary, changes can be made by clicking Change.

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



Module Definition*

Series: A

Revision: 1 1

Electronic Keying: Compatible Module

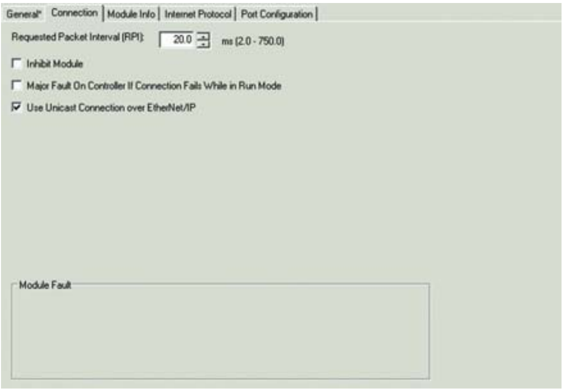
Connection: Data

DataFormat: SINT

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

Connection Tab

You do not need to change any settings on this tab.



Requested Packet Interval (RPI): 20.0 ms (2.0 - 750.0)

□ Inhibit Module

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

☑ Use Unicast Connection over EtherNet/IP

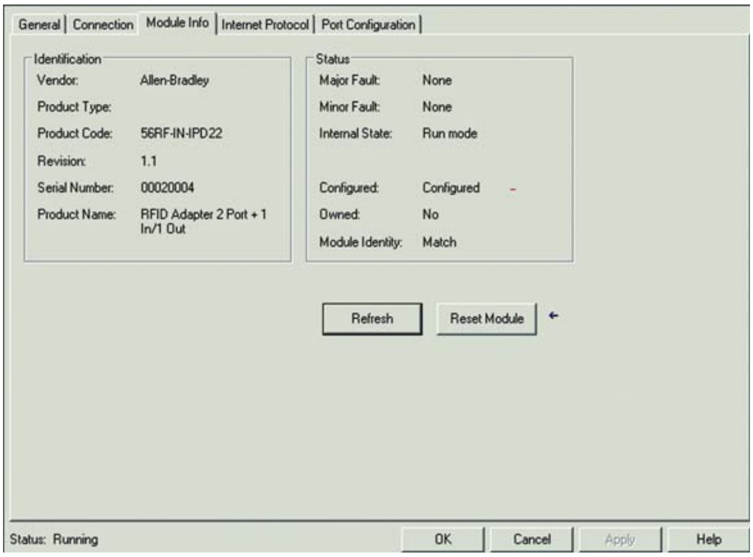
Module Fault

Setting	Description
Requested Packet Interval	Specify the number of milliseconds between requests for information from the controller to the RFID block. The block can 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 the controller ignores any information that is provided.
Major Fault on Controller If Connection Fails While In Run Mode	Check this box if a connection failure is 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 appear in this box.

Module Info Tab

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

In the left panel, the Add-on Profile (AOP) shows the vendor, product type, product code. Revision level, serial number, and product name.



Identification
Vendor: Allen-Bradley
Product Type:
Product Code: 56RF-IN-IPD22
Revision: 1.1
Serial Number: 00020004
Product Name: RFID Adapter 2 Port + 1 In/1 Out

Status
Major Fault: None
Minor Fault: None
Internal State: Run mode

Configured: Configured
Owned: No
Module Identity: Match

Status: Running

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

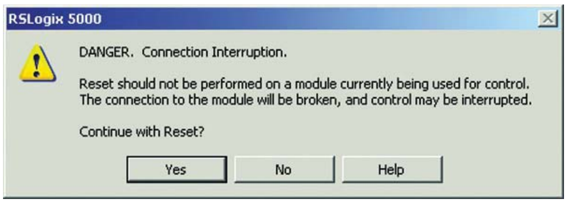
The Refresh and Reset Module buttons are active when the controller is online.

- Refresh

  Click to refresh the data in the window.

- Reset Module

  Click with care as it disconnects the module momentarily and control is interrupted. The following warning window appears.

  RSLogix 5000

  DANGER. Connection Interruption.

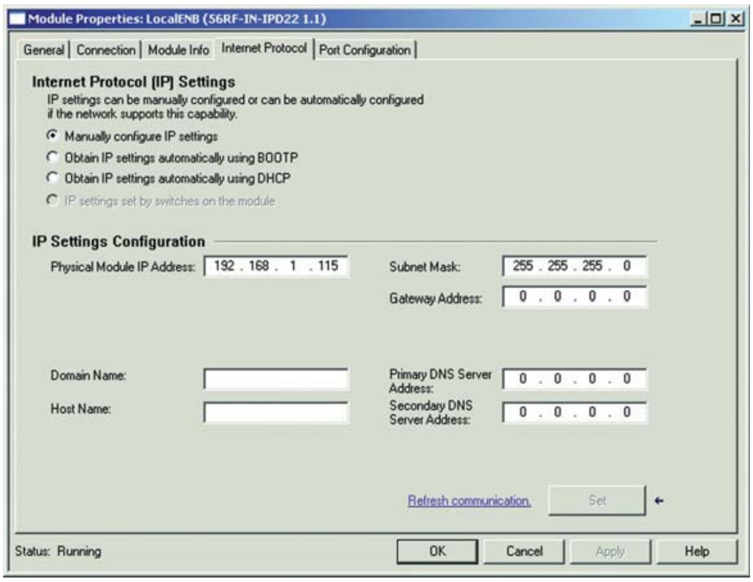
  Reset should not be performed on a module currently being used for control. The connection to the module will be broken, and control may be interrupted.

  Continue with Reset?

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

Internet Protocol Tab

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


Module Properties: LocalENB (56RF-IN-IPD22 1.1)
General | Connection | Module Info | Internet Protocol | Port Configuration

Internet Protocol (IP) Settings

IP settings can be manually configured or can be automatically configured if the network supports this capability.

● Manually configure IP settings
○ Obtain IP settings automatically using BOOTP
○ Obtain IP settings automatically using DHCP
○ IP settings set by switches on the module

IP Settings Configuration

Physical Module IP Address: 192 . 168 . 1 . 115
Subnet Mask: 255 . 255 . 255 . 0
Gateway Address: 0 . 0 . 0 . 0
Domain Name:
Host Name:
Primary DNS Server Address: 0 . 0 . 0 . 0
Secondary DNS Server Address: 0 . 0 . 0 . 0

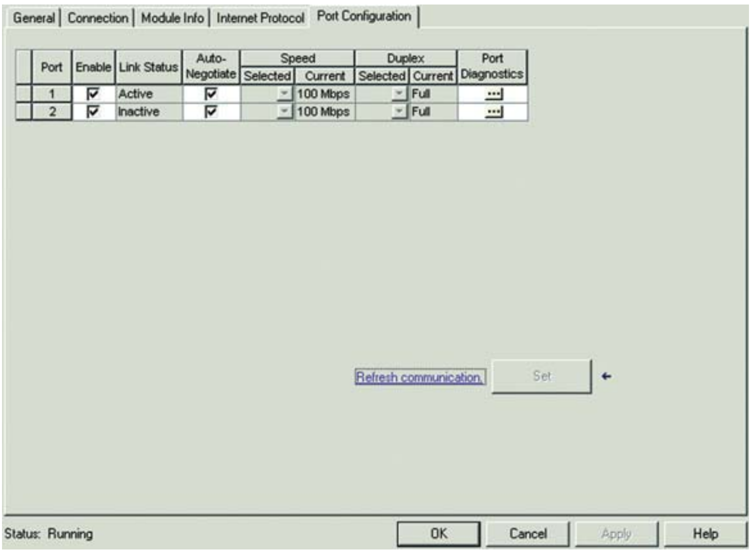
Status: Running

Port Configuration Tab

Changes to the fields on the Port Configuration tab are not required for the Quick Start process. These fields only become active when the controller is online.

The number of ports that are shown in this window varies depending on the block used. There are either one or two ports.

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

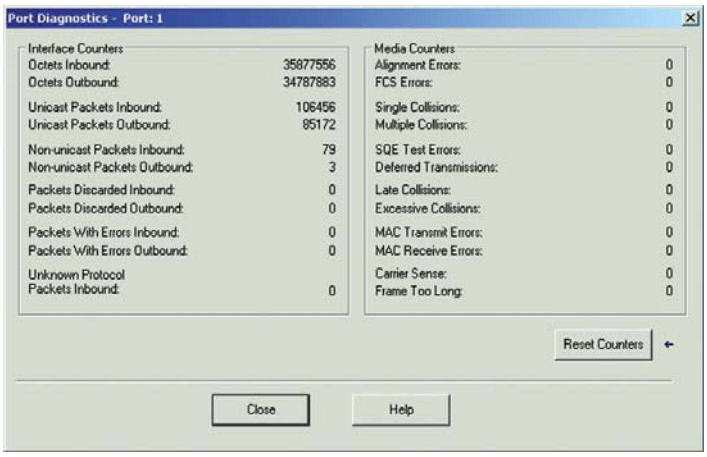

General | Connection | Module Info | Internet Protocol | Port Configuration

Port	Enable	Link Status	Auto-Negotiate	Speed		Duplex		Port Diagnostics
				Selected	Current	Selected	Current	
1	☑	Active	☑		100 Mbps		Full	...
2	☑	Inactive	☑		100 Mbps		Full	...

Status: Running

Refresh communication.
Set
OK
Cancel
Apply
Help

Click the ellipsis (...) under the Port Diagnostics. The following window appears, which shows the communication that takes place between the controller and the transceiver that is connected to the port.



Port Diagnostics - Port: 1

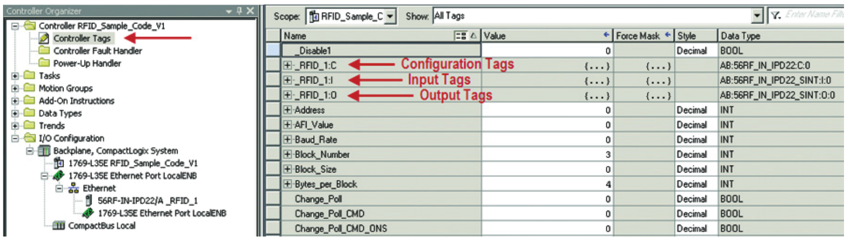
Interface Counters	
Octets Inbound:	35877556
Octets Outbound:	34787883
Unicast Packets Inbound:	106456
Unicast Packets Outbound:	85172
Non-unicast Packets Inbound:	79
Non-unicast Packets Outbound:	3
Packets Discarded Inbound:	0
Packets Discarded Outbound:	0
Packets With Errors Inbound:	0
Packets With Errors Outbound:	0
Unknown Protocol Packets Inbound:	0

Media Counters	
Alignment Errors:	0
FCS Errors:	0
Single Collisions:	0
Multiple Collisions:	0
SQE Test Errors:	0
Deferred Transmissions:	0
Late Collisions:	0
Excessive Collisions:	0
MAC Transmit Errors:	0
MAC Receive Errors:	0
Carrier Sense:	0
Frame Too Long:	0

RSLogix 5000 Controller Tags

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

In the Controller Organizer, click the Controller Tags.



Scope: RFID_Sample_C Show: All Tags

Name	Value	Force Mask	Style	Data Type
_Disable1	0		Decimal	BOOL
_RFID_1:C	{...}	{...}		AB:56RF_IN_IPD22:C:0
_RFID_1:I	{...}	{...}		AB:56RF_IN_IPD22_SINT:I:0
_RFID_1:O	{...}	{...}		AB:56RF_IN_IPD22_SINT:O:0
Address	0		Decimal	INT
AFI_Value	0		Decimal	INT
Baud_Rate	0		Decimal	INT
Block_Number	3		Decimal	INT
Block_Size	0		Decimal	INT
Bytes_per_Block	4		Decimal	INT
Change_Poll	0		Decimal	BOOL
Change_Poll_CMD	0		Decimal	BOOL
Change_Poll_CMD_ONS	0		Decimal	BOOL

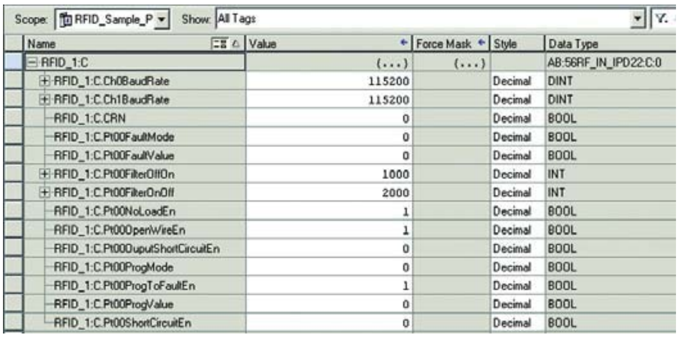
Configuration Tags
Input Tags
Output Tags

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

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

Configuration Image Table and Tags

Expand the RFID_1:C by clicking "+" to show the configuration image table, which has the following tags:



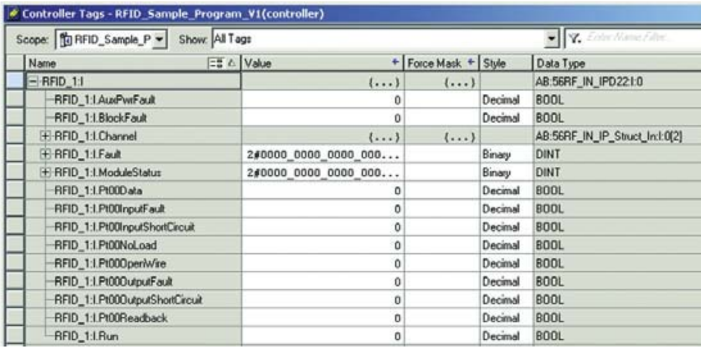
Scope: RFID_Sample_P Show: All Tags

Name	Value	Force Mask	Style	Data Type
⊟ RFID_1:C	{...}	{...}		AB:56RF_IN_IPD22:C:0
⊞ RFID_1:C.Ch0BaudRate	115200		Decimal	DINT
⊞ RFID_1:C.Ch1BaudRate	115200		Decimal	DINT
RFID_1:C.CRN	0		Decimal	BOOL
RFID_1:C.Pt00FaultMode	0		Decimal	BOOL
RFID_1:C.Pt00FaultValue	0		Decimal	BOOL
⊞ RFID_1:C.Pt00FilterOffOn	1000		Decimal	INT
⊞ RFID_1:C.Pt00FilterOnOff	2000		Decimal	INT
RFID_1:C.Pt00NoLoadEn	1		Decimal	BOOL
RFID_1:C.Pt00OpenWireEn	1		Decimal	BOOL
RFID_1:C.Pt00OuputShortCircuitEn	0		Decimal	BOOL
RFID_1:C.Pt00ProgMode	0		Decimal	BOOL
RFID_1:C.Pt00ProgToFaultEn	1		Decimal	BOOL
RFID_1:C.Pt00ProgValue	0		Decimal	BOOL
RFID_1:C.Pt00ShortCircuitEn	0		Decimal	BOOL

Tag	Description
Ch0BaudRate	The communication rate for Channel 0 from the RFID block to the RFID transceiver is stored in this tag. Allowable communication rates are 9600, 19200, 38400, and 115200. The default value is 115200.
Ch1BaudRate	The communication rate for Channel 1 from the RFID block to the RFID transceiver is stored in this tag. Allowable communication 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. You do not need to use this tag.
Pt00FaultMode	The Pt00FaultMode is used with FaultValue to configure the state of output 0 when a communications fault occurs. A value of 0 means that, if there is a communications fault, the value in FaultValue is used (OFF or ON). A value of 1 means that the last state is held. By default this value is 0.
Pt00FaultValue	The Pt00FaultValue is used with FaultMode to configure the state of output 0 when a communications fault occurs. A value of 0 is OFF; 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 that 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 that 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 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) is used when the controller is in Program mode. A value of 1 means that the last state is held. By default this value is 0.
Pt00ProgValue	The Pt00ProgValue is used with ProgMode to configure the state of output 0 when the controller is in Program mode. A value of 0 is OFF; 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 "+" to show the input image table, which has the following tags:

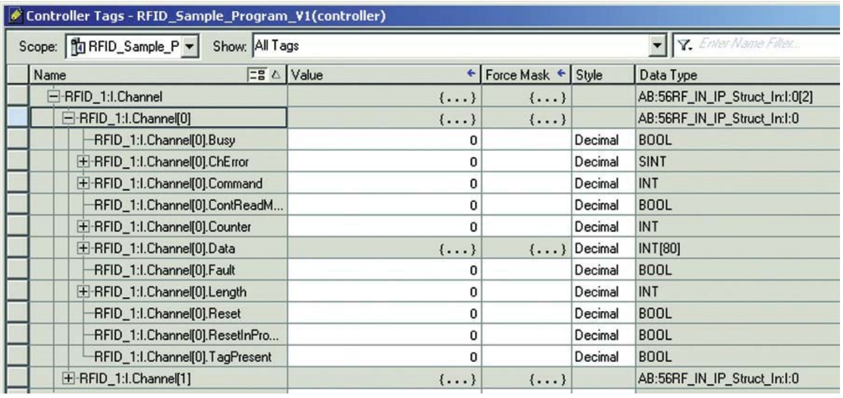


Name	Value	Force Mask	Style	Data Type
RFID_1:I	{...}	{...}		AB:56RF_IN_IPD22:I:0
RFID_1:I.AuxPwrFault	0		Decimal	BOOL
RFID_1:I.BlockFault	0		Decimal	BOOL
RFID_1:I.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_In:I:0[2]
RFID_1:I.Fault	2#0000_0000_0000_000...		Binary	DINT
RFID_1:I.ModuleStatus	2#0000_0000_0000_000...		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

Tag	Description
AuxPwrFault	The AuxPwrFault bit indicates 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 indicates if any of the RFID channels or input/output points is in a fault condition. A value of 0 indicates that 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 stores the connection status between the interface and the controller. A value of 0 indicates that a connection has been established; a value of -1 indicates no connections.
ModuleStatus	The Module status is a 4-byte value that contains the overall status of the module. A value of 0 or 1 indicates that the module is functioning with no faults; a value greater than 1 indicates that a fault condition exists. The ModuleStatus word varies slightly based on the configured unit.
Pt00Data	The Pt00Data bit indicates if the status of input point 0. A value of 0 indicates open; a value of 1 indicates closed.
Pt00InputFault	The Pt00InputFault bit indicates 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; a value of 1 indicates a fault condition.
Pt00InputShortCircuit	The Pt00InputShortCircuit bit indicates if the input point 0 has a 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 indicates if the output point 0 has a no load condition; No load detection 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 indicates 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 indicates 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 indicates 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 indicates 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 contains the value of 1.
Run	The Run bit indicates if the block is in run or program mode. A value of 1 indicates that the block is in run mode; a value of 0 indicates that the block is in program mode.

Input Channel Tags

Expand the RFID_1:Channel by clicking “+” to show that two channels exist (Channel[0] and Channel[1]). Expand the RFID_1:Channel[0] by clicking “+”. 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:I.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_In:I:0[2]
⊟-RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_IN_IP_Struct_In: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	0		Decimal	INT
RFID_1:I.Channel[0].ContReadM...	0		Decimal	BOOL
⊞-RFID_1:I.Channel[0].Counter	0		Decimal	INT
⊞-RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	INT[80]
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].ResetInPro...	0		Decimal	BOOL
RFID_1:I.Channel[0].TagPresent	0		Decimal	BOOL
⊞-RFID_1:I.Channel[1]	{...}	{...}		AB:56RF_IN_IP_Struct_In:I:0

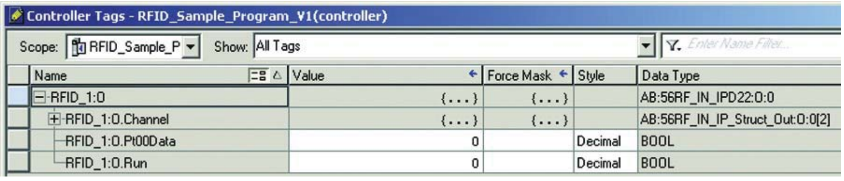
Tag	Description
Busy	The channel Busy bit indicates the status of an RFID channel. A value of 0 indicates that the RFID channel is not executing a command. A value of 1 indicates that a command is in the process of executing on that channel.
ChError	The channel ChError is a 1-byte word that contains the last error code for that channel. A value of 0 indicates no error, a value >0 indicates some error. See Error Codes for RFID Interface Block on page 123 for a list of the error codes.
Command	The channel command word is a 2-byte value that stores the last command that the channel received; at power-up this value must be 0. The allowable commands are listed in Table 23 on page 57.
ContReadMode	The channel ContReadMode bit indicates 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 ignores all other commands except a Stop Continuous Read.
Counter	The channel counter word is a 2-byte value that increments its value by 1 after the interface has completed execution of a command. This value rolls over to 0 after it counts to 65535 and starts 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 that is returned from the RFID interface. Upon completion a command, reply data is deposited in this array and the length of the reply (in 16-bit word increments) is placed within the associated length field; at power-up this value must be 0.
Fault	The channel fault bit indicates the fault status of the RFID channel. A value of 0 indicates that the channel is operating normally; a value of 1 indicates that the channel has faulted.
Length	The channel length word is a 2-byte value that indicates the data length for specific commands. Upon completion of a command, this word is populated with the number of 16-bit words that are returned to the data field; at power-up this value must be 0.
Reset	The channel reset bit indicates the reset status of the RFID channel. A value of 0 indicates that the channel is not in reset; a value of 1 indicates that the channel has completed a reset.
ResetInProgress	The channel ResetInProg bit indicates 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 indicates the status of a tag at the RFID channel. A value of 0 indicates that there is not tag present at the transceiver; a value of 1 indicates one or more tags have been detected at the transceiver.

Table 23 - Allowable Commands

Value	Command	Description
1	Read Single Block	Reads one 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	Allow you to set the best time to start reading in continuous read mode automatically.
10	Write SingleBlock	Writes one 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 communication rate, Device ID,Retry Time, and Gain.
33	Get Version Information	Retrieves the firmware revision 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.

Output Image Table and Tags

Expand the RFID_1:O by clicking “+” to show the output image table, which has the following tags:

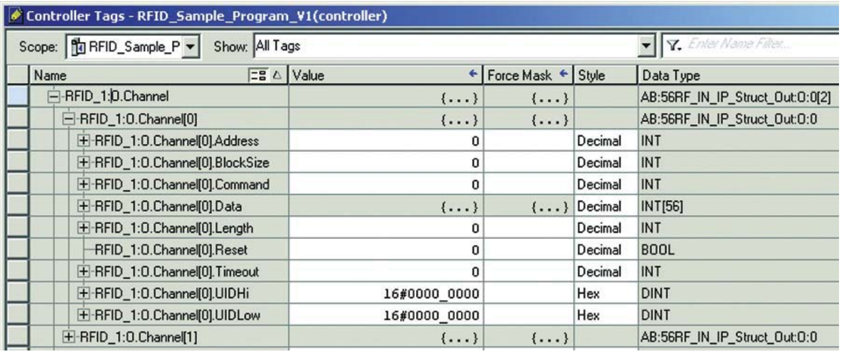


Name	Value	Force Mask	Style	Data Type
RFID_1:O	{...}	{...}		AB:56RF_IN_IPD22:O:0
RFID_1:O.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_Out:O:0[2]
RFID_1:O.Pt00Data	0		Decimal	BOOL
RFID_1:O.Run	0		Decimal	BOOL

Tag	Description
Channel	See Output Channel Tags.
Pt00Data	The Pt00Data bit is used to turn output point 0 either ON or OFF. A value of 0 is used to turn OFF the output point; a value of 1 is used to turn ON the output point.
Run	The Run bit is used to place the RFID block into run or program mode. A value of 0 is used for program mode; a value of 1 is used for run mode. When in program mode, the interface maintains the connection to the processor but does not execute commands. The discrete output point follows the mode of the processor and the Run bit, with the Run bit overriding.

Output Channel Tags

Expand the RFID_1:Channel by clicking "+" to show that two channels exist (Channel[0] and Channel[1]). Expand the RFID_1:Channel[0] by clicking "+". 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_1:0.Channel	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0[2]
⊟-RFID_1:0.Channel[0]	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0
⊞-RFID_1:0.Channel[0].Address	0		Decimal	INT
⊞-RFID_1:0.Channel[0].BlockSize	0		Decimal	INT
⊞-RFID_1:0.Channel[0].Command	0		Decimal	INT
⊞-RFID_1:0.Channel[0].Data	{...}	{...}	Decimal	INT[56]
⊞-RFID_1:0.Channel[0].Length	0		Decimal	INT
-RFID_1:0.Channel[0].Reset	0		Decimal	BOOL
⊞-RFID_1:0.Channel[0].Timeout	0		Decimal	INT
⊞-RFID_1:0.Channel[0].UIDHi	16#0000_0000		Hex	DINT
⊞-RFID_1:0.Channel[0].UIDLow	16#0000_0000		Hex	DINT
⊞-RFID_1:0.Channel[1]	{...}	{...}		AB:56RF_IN_IP_Struct_Out:0:0

Tag	Description
Address	The channel Address word is a 2-byte value that contains the address or block value within the RFID tag that the command executes on.
BlockSize	The channel BlockSize word is a 2-byte value that stores the expected Block Size for the tag. Valid values are 0 bytes, 4 bytes, or 8 bytes per block. A value of 0 defaults to a Block Size of 4 bytes per block.
Command	The channel Command word is a 2-byte value that stores the next command for the interface to process. The RFID interface executes 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. The allowable commands are listed in Table 24.
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 is directed to the RFID interface. Some commands, such as reading, do not require the use of this data field. Writing to tags uses this information with the length field to inform the RFID interface what values it must 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 indicates the data length for specific commands. Upon completion of a command, this word is populated with the number of 16-bit words that are 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 that 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 waits for a command response from the transceiver before indicating a message timeout. The default value is 0, which sets the timeout at 750 ms. You can enter a timeout value in milliseconds. **IMPORTANT** 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 contains the UUID information for specific commands that allows the command to be targeted to a specific tag in the field. Under normal circumstances, this value is 0, which tells the RFID interface to perform an action regardless of what tag it is. Any value other than 0 attempts 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.

Table 24 - Allowable Commands

Value	Command	Description
1	Read Single Block	Reads one 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	Allows you to set the best time to start reading in continuous read mode automatically.
10	Write SingleBlock	Writes one 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 communication rate, Device ID and Retry Time.
33	Get Version Information	Retrieves the firmware revision 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.

Notes:

Commands Summary

RFID Commands

This section provides a summary of the commands that the RFID transceiver supports. Detail of the commands can be found in RSLogix 5000 Code Examples on page 63. This guide assumes familiarity with RSLogix 5000®. The *.ACD file must already be downloaded into the PLC and working properly.

Table 25 assumes the following:

- You have configured the RSLogix 5000 Add-on Profile (AOP) with Data Format set to SINT.
- The RFID tag has blocks that are only 4 bytes each.
- The Universally Unique Identifier (UUID) is set to zero (unless specified).

A 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 operates on the first tag in the transceiver field. All other Output values must be set to 0 where not specified.

Table 25 - Commands

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 four 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 four tags

Read Single Block	Option Flag 0
	Reads one block of user data from a tag
		Command = 1
		Data[0] = 0
			Data[0…3] = User data (4 bytes)

	Option Flag 1
	Reads one 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 one 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 one 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 AFI area that is contained within block -2	Command = 41
		Length = 1
		Data[0] = 00xx
			All data bytes are zero

Table 25 - Commands (Continued)

Command	Description	Output xx.O.Channel[0]	Input xx.I.Channel[0]

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 four 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 first tag Data[10…*] = User data of first tag Data[*…*] = UUID of second tag Data[*…*] = User data of second tag

Multi-tag Block Write	Writes multiple blocks of user data to up to four 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 first tag
			Data[10…17] = UUID of second tag
			Data[18…25] = UUID of third tag
			Data[26…33] = UUID of fourth tag

Read Transceiver Settings	Retrieves the following information from the transceiver: Communication rate Device ID Retry time	Command = 31	Data[0…1] = Device ID Data[2…5] = Communication rate Data[6…7] = Retry setting Data[8…9] = Gain
Get Version Information	Retrieves the firmware revision from the transceiver	Command = 33	Data = firmware revision

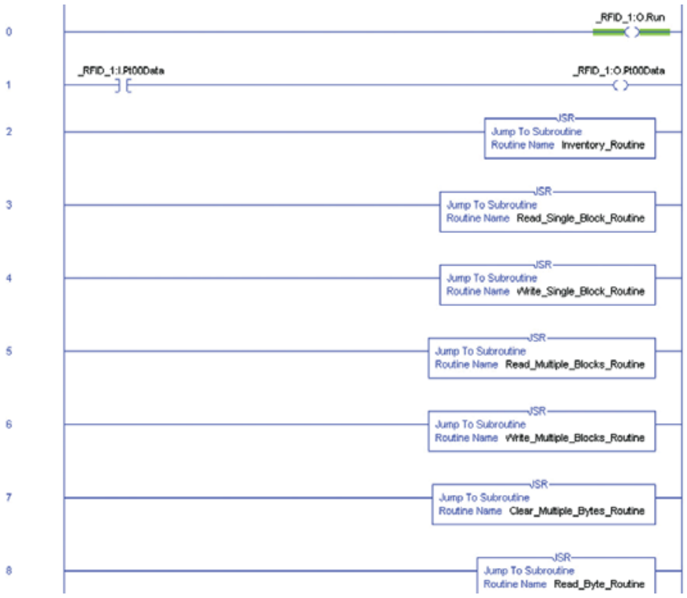
RSLogix 5000 Code Examples

This chapter contains examples of routines that run in the RSLogix 5000® program.

The examples are written for an RF transceiver that is 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 you to repeat the routine easily.

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

Main Routine


0
_RFID_1:O.Run
1
_RFID_1:I.Pt00Data
_RFID_1:O.Pt00Data
2
JSR
Jump To Subroutine
Routine Name Inventory_Routine
3
JSR
Jump To Subroutine
Routine Name Read_Single_Block_Routine
4
JSR
Jump To Subroutine
Routine Name Write_Single_Block_Routine
5
JSR
Jump To Subroutine
Routine Name Read_Multiple_Blocks_Routine
6
JSR
Jump To Subroutine
Routine Name Write_Multiple_Blocks_Routine
7
JSR
Jump To Subroutine
Routine Name Clear_Multiple_Bytes_Routine
8
JSR
Jump To Subroutine
Routine Name Read_Byte_Routine

A partial listing of the Main Routine is shown in the Example Command Routines - Overview. 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 a status indicator to confirm that you have pressed the momentary switch.

Example Command Routines - Overview

Many of the example routines (not the Main Routine) use the same Ladder Logic. Rung 0 on page 64…Rung 4 on page 65 explain the Ladder Logic.

Rung 0

Rung 0 initiates the routine. A sensor or momentary switch, which is 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 Examine If Closed (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 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 that is 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 – 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 reset it to the same desired value.

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

Unlatch – 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 nonzero value into the output command byte instructs the RFID block to execute the command.

Rung 3

Rung 3 verifies 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 is 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 an RFID tag. You can specify the number of bytes to clear and the address from which to begin. Similar to a “copy” command, it copies the value that you specify in the output data image Data[0] location to the addresses you specify.

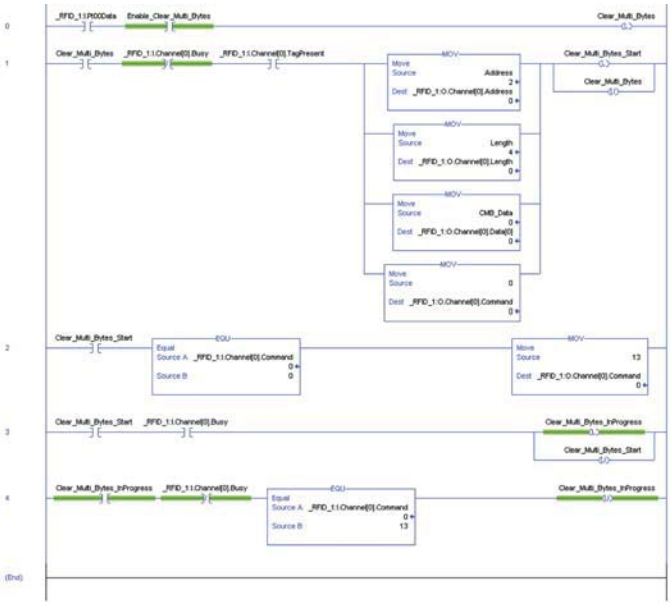
Set the following values in the output image table:

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 that is 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 operates 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.

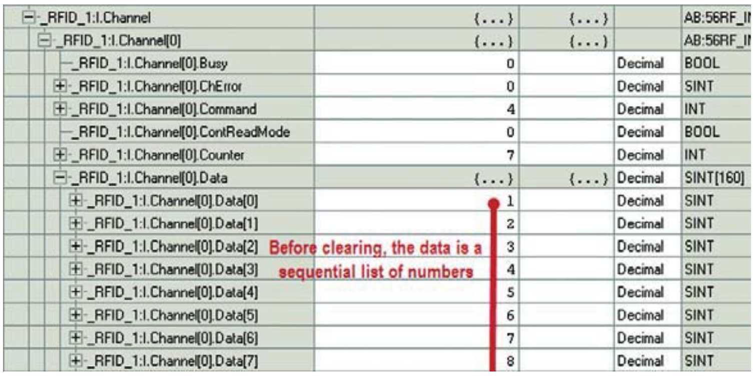
Example Routine

In the following example routine, the initialization in Rung 1 sets the address, length data, the value that is 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 following example, CMB_Data is set to 0, but you can set this value to be any valid SINT value.


0
_RFID_1:I.Pt00Data
Enable_Clear_Multi_Bytes
Clear_Multi_Bytes
1
Clear_Multi_Bytes
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
2
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
4
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source CMB_Data
0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Clear_Multi_Bytes_Start
Clear_Multi_Bytes
2
Clear_Multi_Bytes_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 0
MOV
Move
Source 13
Dest _RFID_1:O.Channel[0].Command
0
3
Clear_Multi_Bytes_Start
_RFID_1:I.Channel[0].Busy
Clear_Multi_Bytes_InProgress
Clear_Multi_Bytes_Start
4
Clear_Multi_Bytes_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 13
Clear_Multi_Bytes_InProgress
(End)

Example Results

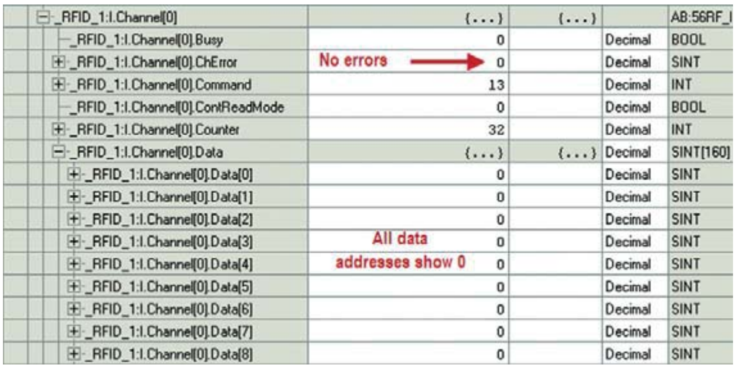
To demonstrate the results, the Read Byte command was executed on an RFID tag. The data in this tag was a simple list of numbers starting from 1. The counter is 31.



⊟-_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

Before clearing, the data is a sequential list of numbers

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

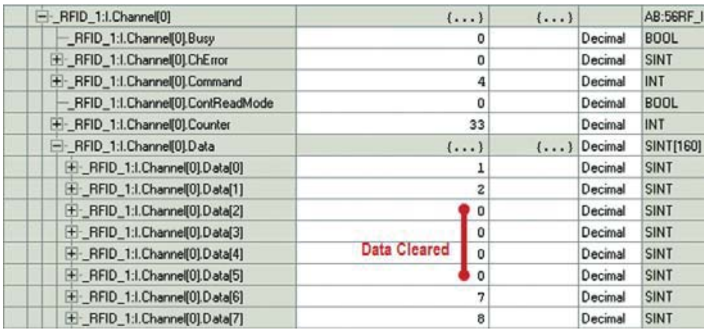


⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_I
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	13		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
⊞-_RFID_1:I.Channel[0].Counter	32		Decimal	INT
⊟-_RFID_1:I.Channel[0].Data	{...}	{...}	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

No errors

All data addresses show 0

The tag is read again (command = 4) to confirm the clearing. Data bytes 2…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 also displays 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

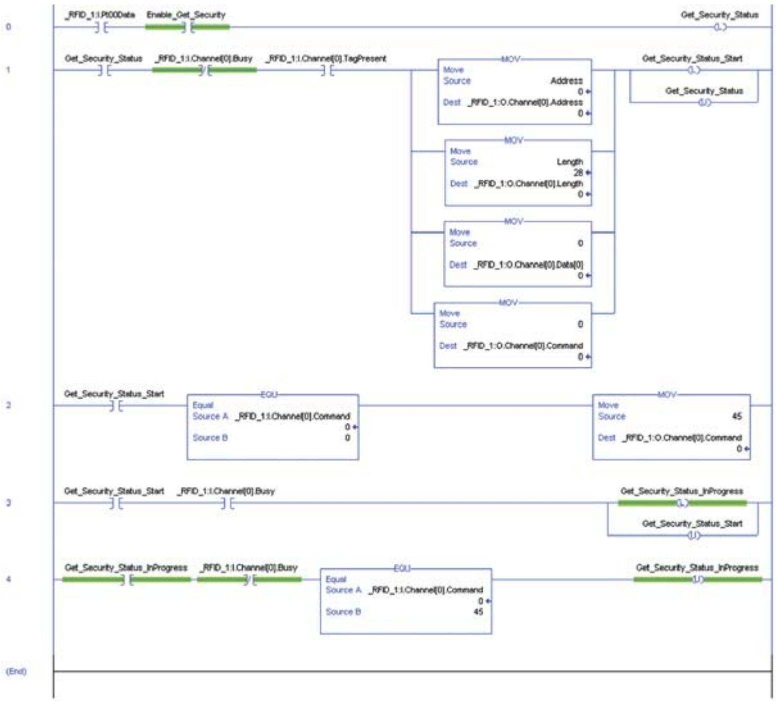
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)

i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command operates 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.

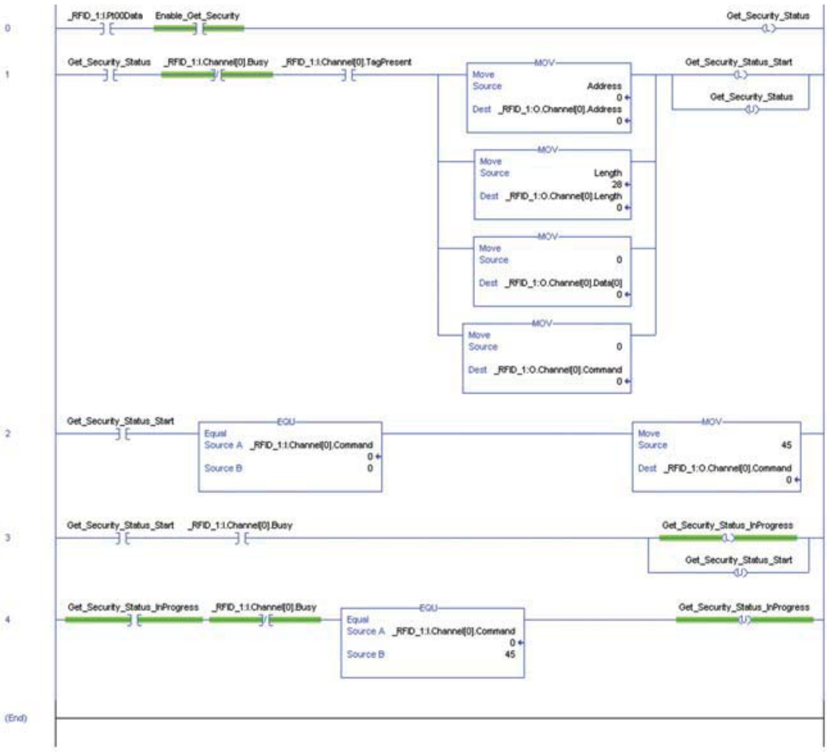
Example Routine

In the following example routine, the initialization in Rung 1 sets the address, length data, the Data[0] value that is 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 blocks of this RFID tag).


0
_RFID_1:I.Pt00Data
Enable_Get_Security
Get_Security_Status
L
1
Get_Security_Status
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
0
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
28
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Get_Security_Status_Start
L
Get_Security_Status
U
2
Get_Security_Status_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 0
MOV
Move
Source 45
Dest _RFID_1:O.Channel[0].Command
0
3
Get_Security_Status_Start
_RFID_1:I.Channel[0].Busy
Get_Security_Status_InProgress
L
Get_Security_Status_Start
U
4
Get_Security_Status_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 45
Get_Security_Status_InProgress
U
(End)

Example Results

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


0
_RFID_1:I.Pt00Data
Enable_Get_Security
Get_Security_Status
L
1
Get_Security_Status
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
0
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
28
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Get_Security_Status_Start
L
Get_Security_Status
U
2
Get_Security_Status_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 0
MOV
Move
Source 45
Dest _RFID_1:O.Channel[0].Command
0
3
Get_Security_Status_Start
_RFID_1:I.Channel[0].Busy
Get_Security_Status_InProgress
L
Get_Security_Status_Start
U
4
Get_Security_Status_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 45
Get_Security_Status_InProgress
U
(End)

The following information is 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 returns the following RFID tag information:

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

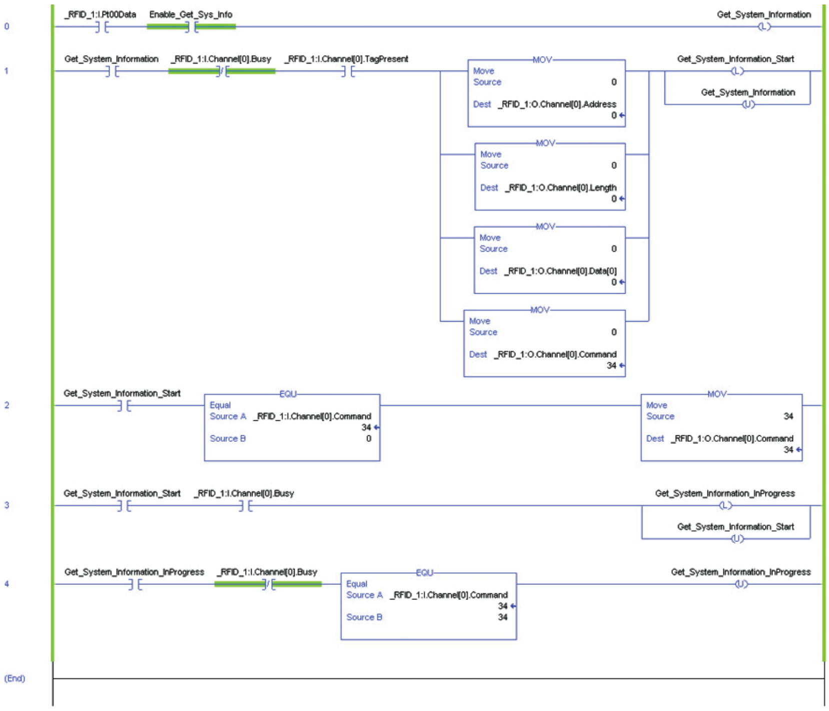
Set the following values in the output image table:

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 operates 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 in Rung 1 sets the address, length data, the Data[0] value that is 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.


0
_RFID_1:I.Pt00Data
Enable_Get_Sys_Info
Get_System_Information
L
1
Get_System_Information
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
34
Get_System_Information_Start
L
Get_System_Information
U
2
Get_System_Information_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
34
Source B 0
MOV
Move
Source 34
Dest _RFID_1:O.Channel[0].Command
34
3
Get_System_Information_Start
_RFID_1:I.Channel[0].Busy
Get_System_Information_InProgress
L
Get_System_Information_Start
U
4
Get_System_Information_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
34
Source B 34
Get_System_Information_InProgress
U
(End)

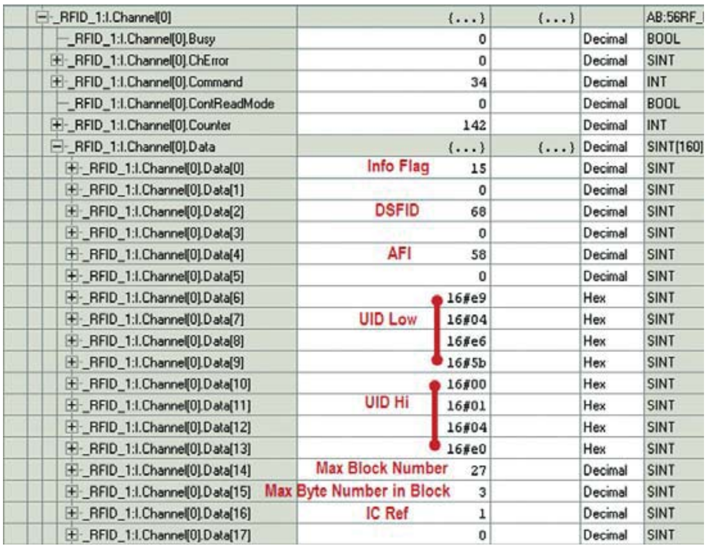
Example Results

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

The DSFID, AFI, and UUID follow.

The tag being read was catalog number 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

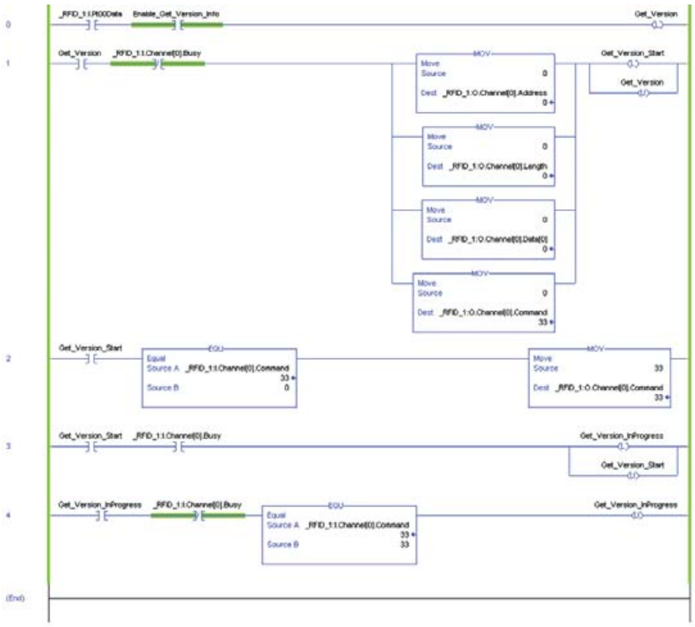
The Get Version Information command retrieves the firmware revision information from the transceiver.

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

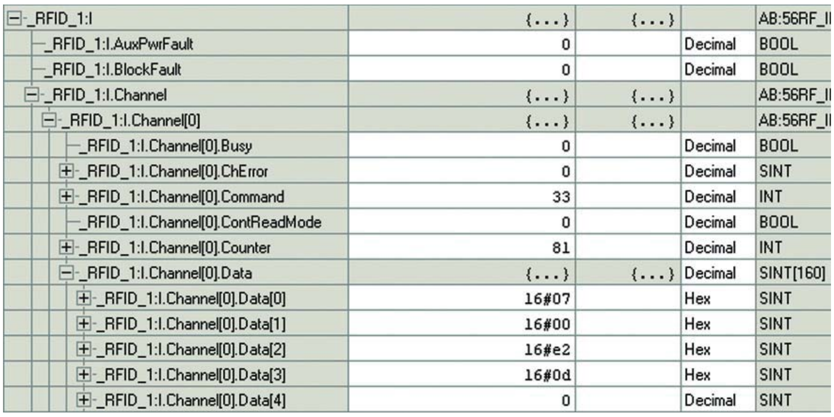
Example Routine

In the following example routine, the initialization in Rung 1 sets the address, length data, the Data[0] value that is 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_I
_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	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 must 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 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) [(a)]

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

d. xx:O.Channel[0].Data[0] = 0 (or 1) [(b)]

e. xx:O.Channel[0].Length = 0 (or 1) [(c)]

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

[(a)] Set Address = 0 to get all tags in the RF field.
Set Address = 1 to get all tags that have the AFI value specified in the Data[0] location.

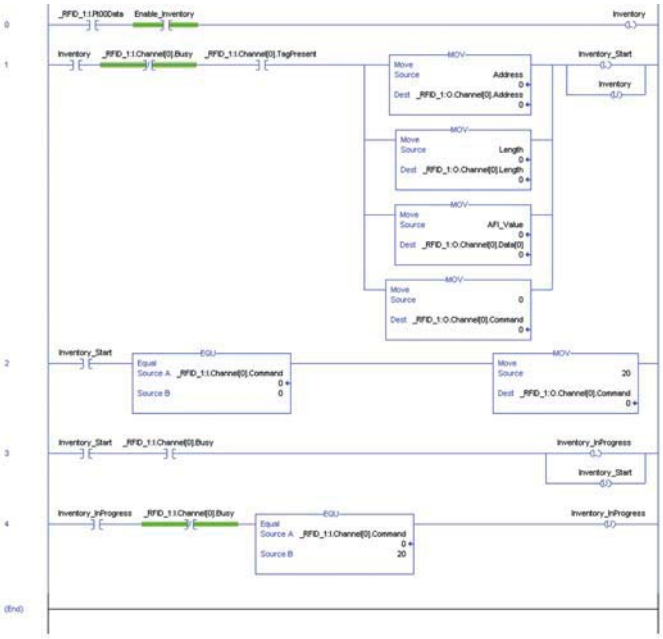
[(b)] Set Data[0] = 0 to return all tags in the RF field.
Set Data[0] = AFI value (but not zero) to return only those tags that have that AFI value

[(c)] Set Length = 0 to get only the UUID for each tag.
Set Length = 1 to get both the UUID and the DSFID for each tag.

Example Routine

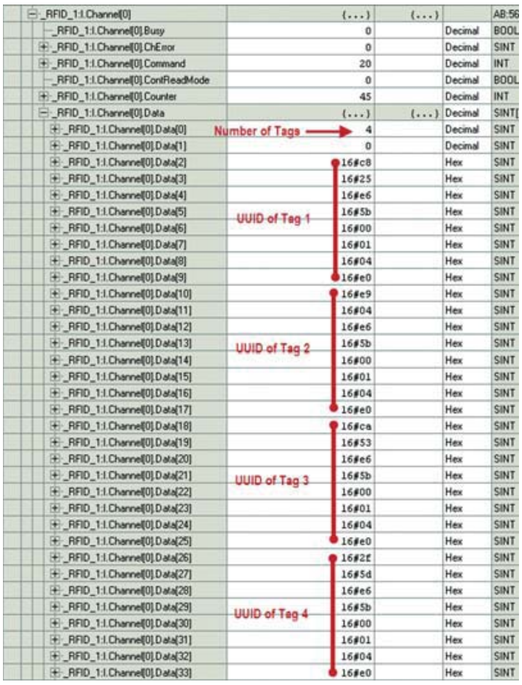
In the following example routine, the initialization in Rung 1 sets the address, length data, the Data[0] value that is used to clear the fields and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.

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



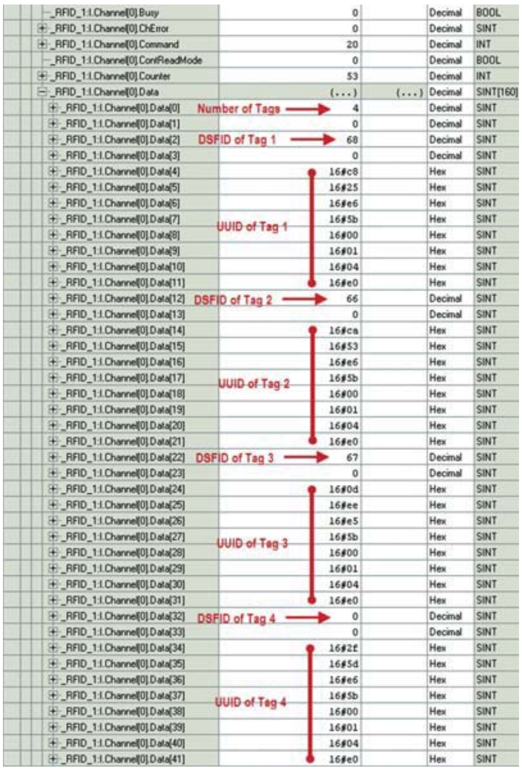
Example Results

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 in the following example. The data shows the number of tags in the RF field and the UUID for each tag.



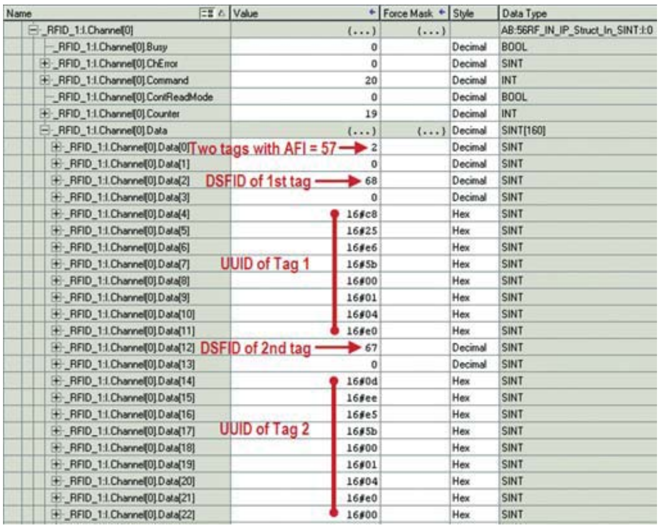
_RFID_1:I.Channel[0]	{...}	{...}		AB:56
_RFID_1:I.Channel[0].Busy	0		Decimal	BOOL
_RFID_1:I.Channel[0].ChError	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	20		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	45		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[
_RFID_1:I.Channel[0].Data[0]	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]	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]	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]	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]	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 in the following example. 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]	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	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]	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]	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]	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]	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]	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]	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]	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 that 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 in the following example. The data shows the number of tags in the RF field, the DSFID, and the UUID for each of these tags.



Name	Value	Force Mask	Style	Data Type
_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	20		Decimal	INT
_RFID_1:I.Channel[0].ContReadMode	0		Decimal	BOOL
_RFID_1:I.Channel[0].Counter	19		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]	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]	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]	67		Decimal	SINT
_RFID_1:I.Channel[0].Data[13]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[14]	16#0d		Hex	SINT
_RFID_1:I.Channel[0].Data[15]	16#ee		Hex	SINT
_RFID_1:I.Channel[0].Data[16]	16#e5		Hex	SINT
_RFID_1:I.Channel[0].Data[17]	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]	16#00		Hex	SINT

Lock AFI

The Lock AFI command locks the 1 byte of information for the AFI, preventing it from being modified in the future.

IMPORTANT Once the AFI byte is locked, it cannot be unlocked.

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

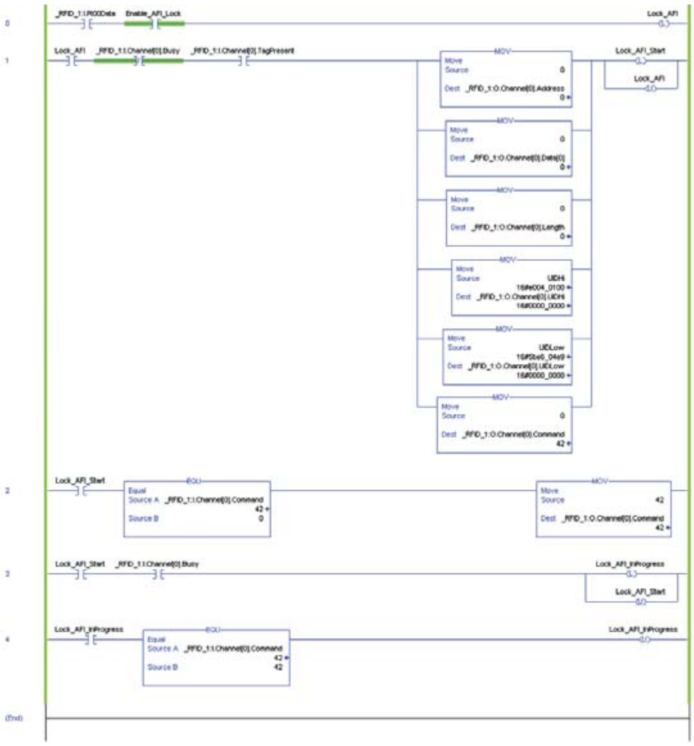
Set the following values in the output image table:

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

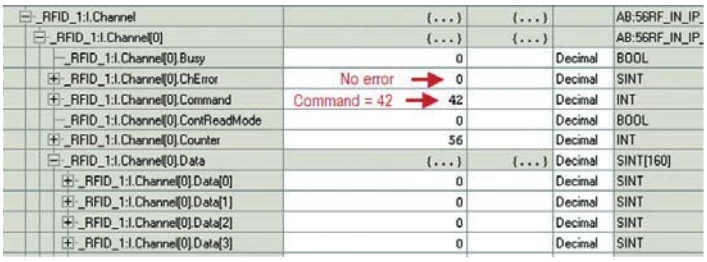
In the following example routine, the initialization in Rung 1 sets the address, length, the Data[0, UIDLow and UIDHi values used to lock the AFI and sets the command value to 0. The BlockSize, Reset, and Timeout are set to 0 in the output image table.



Example Results

Figure 18 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.

Figure 18 - Input Image Table



⊟-_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 are 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. The transceiver automatically determines the block size of the RFID tag.

IMPORTANT Once the block is locked, the block cannot be unlocked.

Set the following values in the output image table:

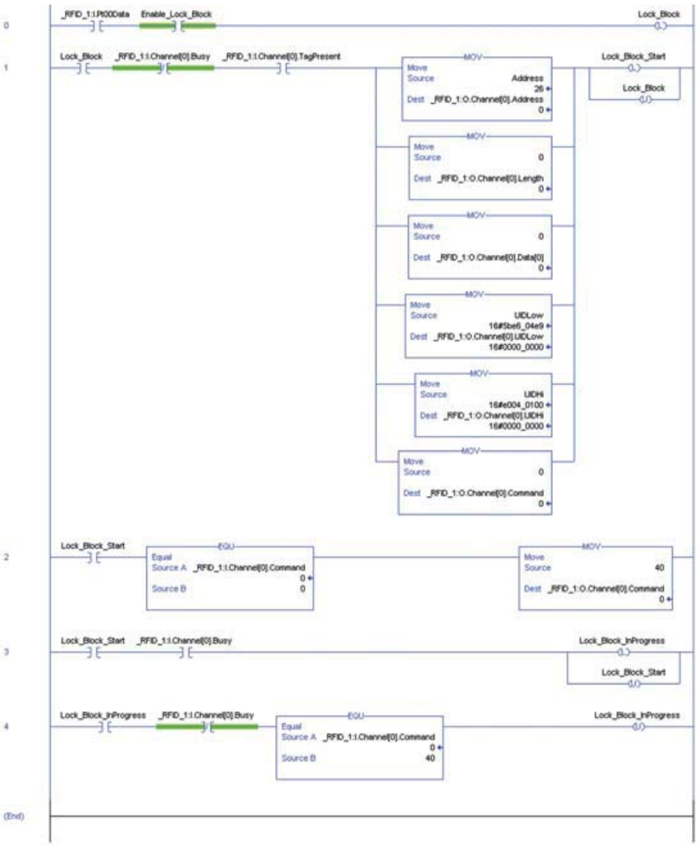
a. xx:O.Channel[0].Command = 40
b. xx:O.Channel[0].Address = the number of the block to lock
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = UIDLow
i. xx:O.Channel[0].UIDHi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

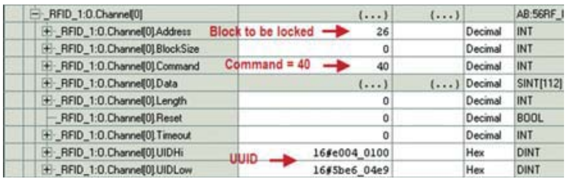
In the example routine, rung 1 initializes the output image table. The UUID is stored in a controllers tags UIDLow and UIDHi. Block 26 is locked. This tag has a total of 27 blocks.



Example Results

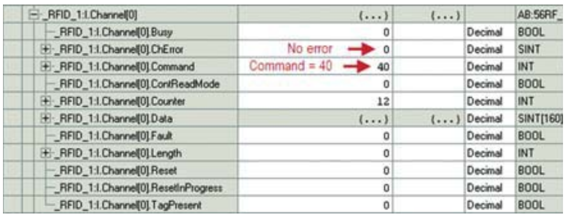
Figure 19 shows address 26, which is the second to last block of the catalog number 56RF-TG-30 tag. The command is 40. The UUID must be specified to lock any blocks.

Figure 19 - Input Image Table



⊟-_RFID_1:O.Channel[0]	{...}	{...}		AB:56RF_I
⊞-_RFID_1:O.Channel[0].Address	26		Decimal	INT
⊞-_RFID_1:O.Channel[0].BlockSize	0		Decimal	INT
⊞-_RFID_1:O.Channel[0].Command	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	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 shows that 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	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Command	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 is 8 if you try to lock a block that is already locked.

Lock DSFID

The Lock DSFID command locks the 1 byte of information for the Data Storage Format Identifier (DSFID) area of the tag, preventing it from being modified.

IMPORTANT Once the DSFID byte is locked, it cannot be unlocked.

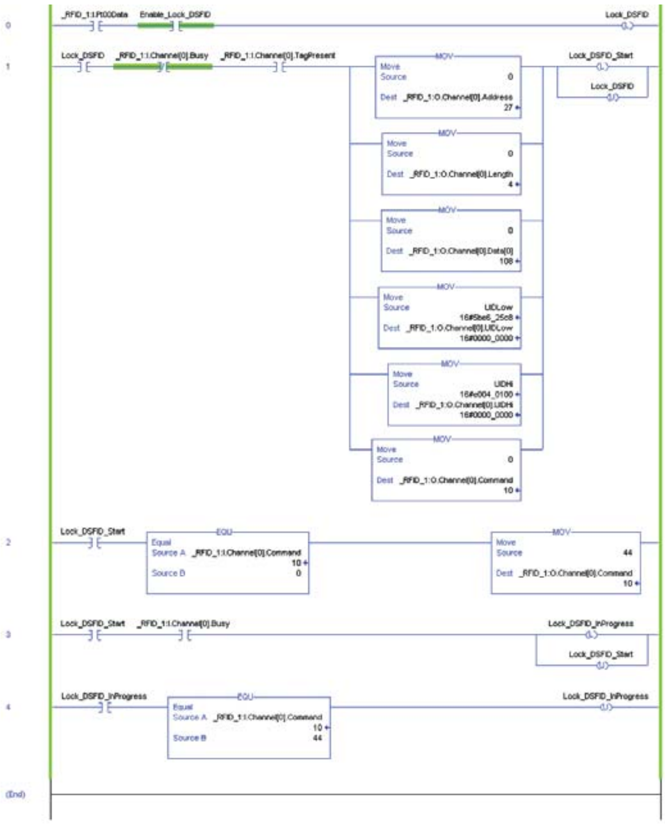
Set the following values in the output image table:

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

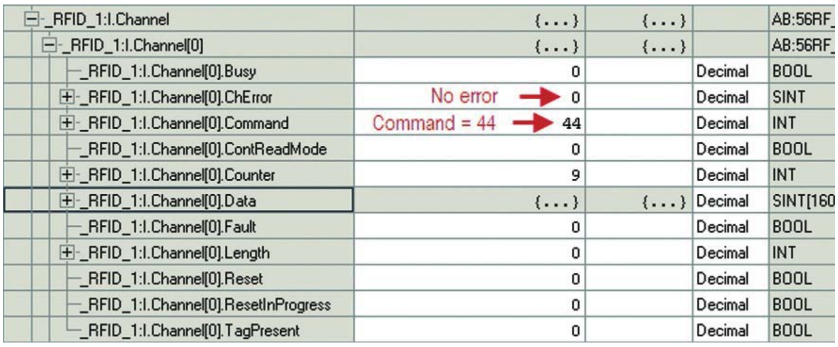
In the following example routine, the initialization in Rung 1 sets the address, length, the Data[0], UIDLow, and UIDHi values used to lock the DSFID and sets the command value to 0. The BlockSize, Reset, and Timeout are set to 0 in the output image table.



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 that is already locked, the ChError is equal to 8.



⊟-_RFID_1:I.Channel	{...}	{...}		AB:56RF_
⊟-_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	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

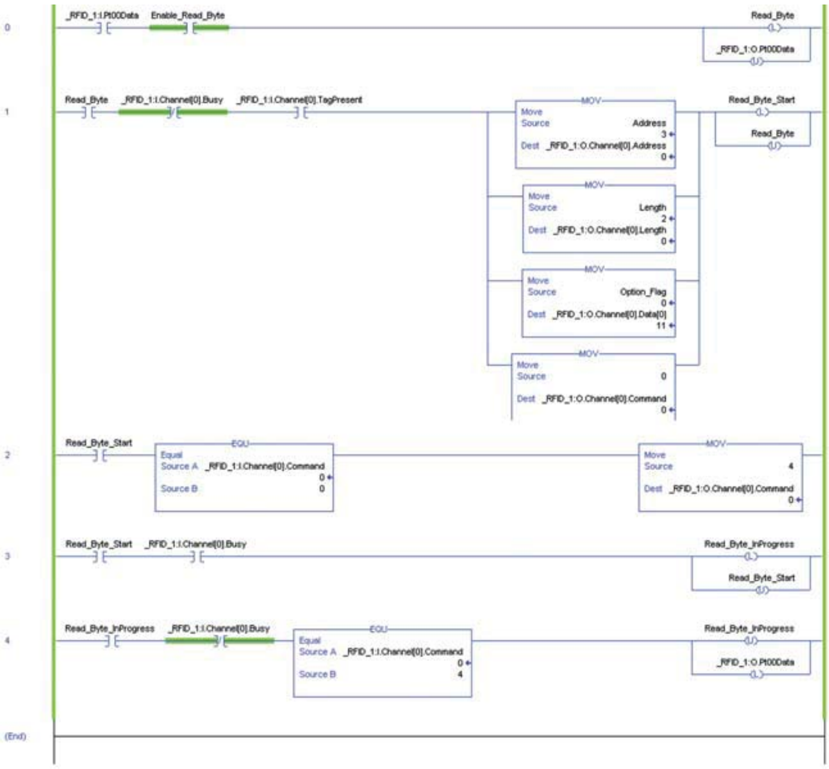
This command operates only on the first tag in the field.

Data[1] must also be set to 0.

Example Routine

The following example routine is to read all data and the UUID in a catalog number 56RF-TG-30 ICODE tag. This tag holds a maximum of 112 bytes of data.

In the following example routine, 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.

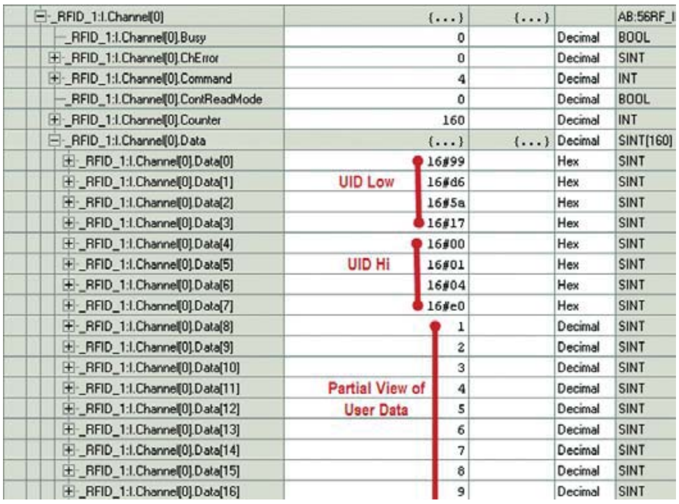


Example Results

Figure 20 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]. Figure 20 only shows a partial listing of the user data. The command read in 112 bytes of data.

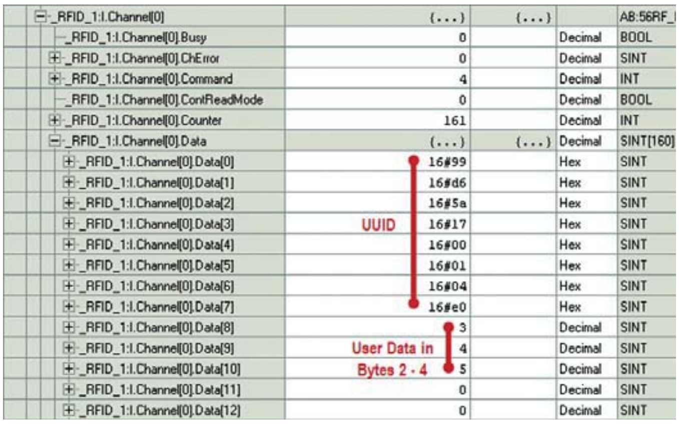
Figure 20 - Input Image Table - User Data



⊟-_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 Figure 21, the command was repeated with the Starting Address set to 2 and the number of bytes set to 3.

Figure 21 - Input Image Table - Repeated Command



⊟-_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 will have the same block size.

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

Set the following values in the output image table:

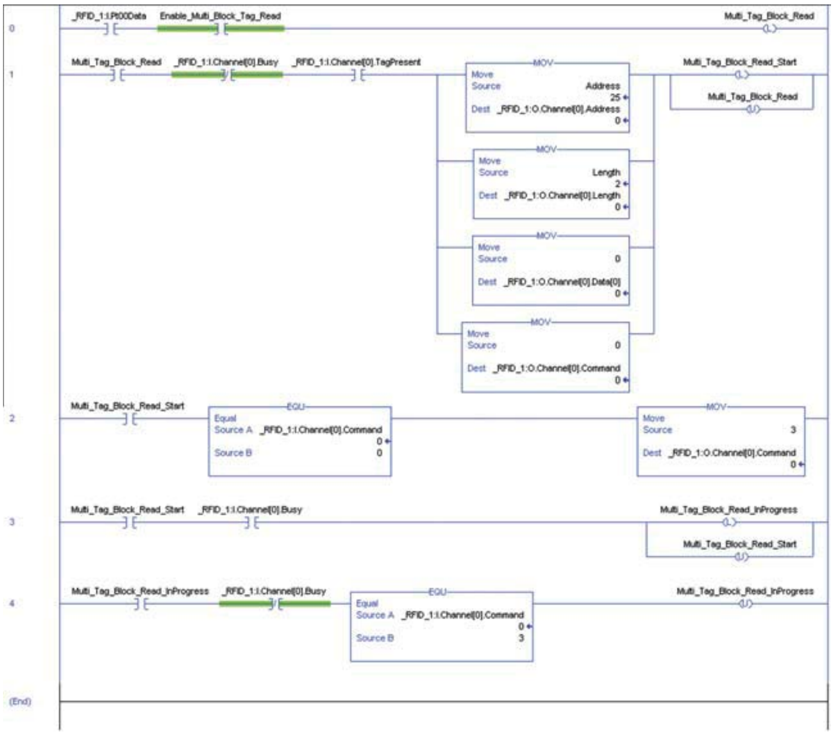
a. xx:O.Channel[0].Command = 3
b. xx:O.Channel[0].Address = the first block to read
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = 0
e. xx:O.Channel[0].Length = the number of blocks to read
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)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

Unless a UUID is specified, this command operates 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 following example routine, the initialization in Rung 1 sets the address, length, the Data[0] value that is 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.

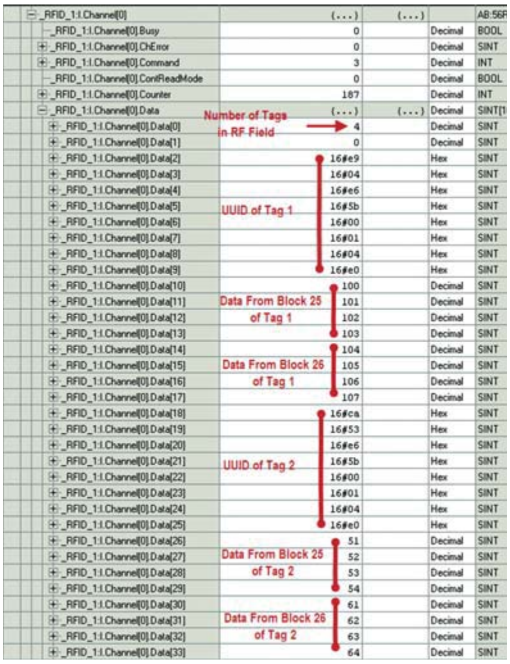
The example ladder diagram is initially set for Address = 25 and the Length = 2. The command reads blocks 25 and 26.


0
_RFID_1:I.Pt00Data
Enable_Multi_Block_Tag_Read
Multi_Tag_Block_Read
1
Multi_Tag_Block_Read
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
25
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
2
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Multi_Tag_Block_Read_Start
Multi_Tag_Block_Read
2
Multi_Tag_Block_Read_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 0
MOV
Move
Source 3
Dest _RFID_1:O.Channel[0].Command
0
3
Multi_Tag_Block_Read_Start
_RFID_1:I.Channel[0].Busy
Multi_Tag_Block_Read_InProgress
Multi_Tag_Block_Read_Start
4
Multi_Tag_Block_Read_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 3
Multi_Tag_Block_Read_InProgress
(End)

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 following example, four catalog number 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 image only shows the information from two of the four RFID tags.



_RFID_1:I.Channel[0]	{...}	{...}		AB:56R
_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	187		Decimal	INT
_RFID_1:I.Channel[0].Data	{...}	{...}	Decimal	SINT[1
_RFID_1:I.Channel[0].Data[0]	4		Decimal	SINT
_RFID_1:I.Channel[0].Data[1]	0		Decimal	SINT
_RFID_1:I.Channel[0].Data[2]	16#e9		Hex	SINT
_RFID_1:I.Channel[0].Data[3]	16#04		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]	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]	100		Decimal	SINT
_RFID_1:I.Channel[0].Data[11]	101		Decimal	SINT
_RFID_1:I.Channel[0].Data[12]	102		Decimal	SINT
_RFID_1:I.Channel[0].Data[13]	103		Decimal	SINT
_RFID_1:I.Channel[0].Data[14]	104		Decimal	SINT
_RFID_1:I.Channel[0].Data[15]	105		Decimal	SINT
_RFID_1:I.Channel[0].Data[16]	106		Decimal	SINT
_RFID_1:I.Channel[0].Data[17]	107		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]	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]	51		Decimal	SINT
_RFID_1:I.Channel[0].Data[27]	52		Decimal	SINT
_RFID_1:I.Channel[0].Data[28]	53		Decimal	SINT
_RFID_1:I.Channel[0].Data[29]	54		Decimal	SINT
_RFID_1:I.Channel[0].Data[30]	61		Decimal	SINT
_RFID_1:I.Channel[0].Data[31]	62		Decimal	SINT
_RFID_1:I.Channel[0].Data[32]	63		Decimal	SINT
_RFID_1:I.Channel[0].Data[33]	64		Decimal	SINT

Number of Tags in RF Field
UUID of Tag 1
Data From Block 25 of Tag 1
Data From Block 26 of Tag 1
UUID of Tag 2
Data From Block 25 of Tag 2
Data From Block 26 of Tag 2

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:

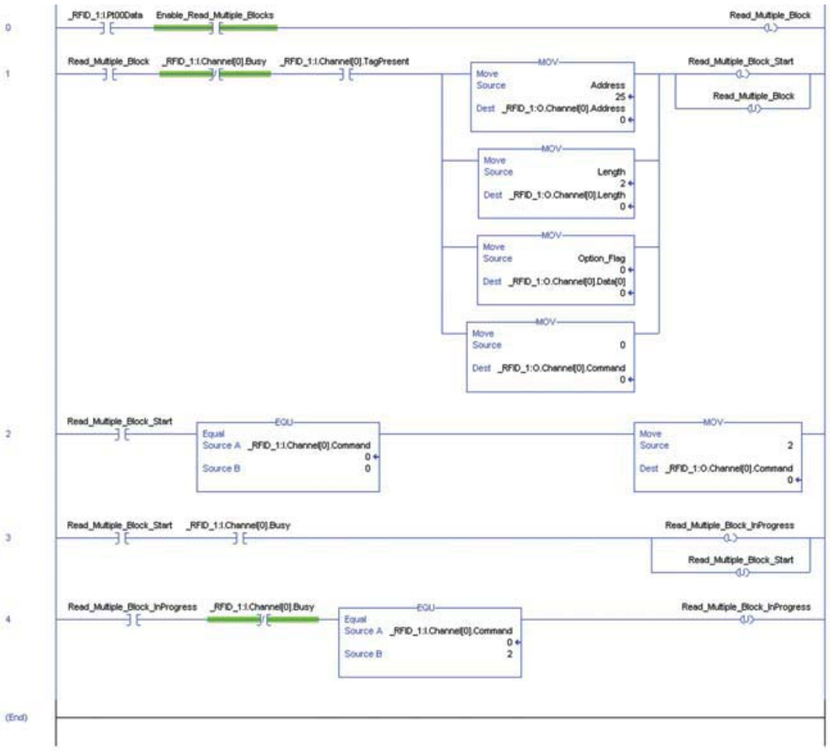
a. xx:O.Channel[0].Command = 2
b. xx:O.Channel[0].Address = the first block to read
c. xx:O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = the Option Flag
e. xx:O.Channel[0].Length = the number of blocks to read
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow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

Unless a UUID is specified, this command operates 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

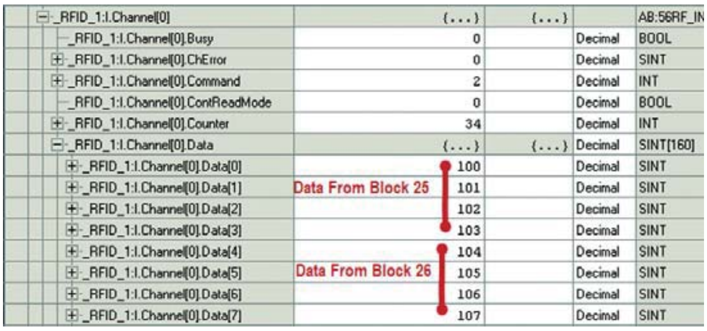
In the following example routine, the initialization in Rung 1 sets the address, length, and Data[0] values used to read multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.

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 reads blocks 25 and 26. The example is repeated with Option Flag set to 1.


0
_RFID_1:I.Pt00Data
Enable_Read_Multiple_Blocks
Read_Multiple_Block
(L)
1
Read_Multiple_Block
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
25
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
2
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source Option_Flag
0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Read_Multiple_Block_Start
(L)
Read_Multiple_Block
(U)
2
Read_Multiple_Block_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 0
MOV
Move
Source 2
Dest _RFID_1:O.Channel[0].Command
0
3
Read_Multiple_Block_Start
_RFID_1:I.Channel[0].Busy
Read_Multiple_Block_InProgress
(L)
Read_Multiple_Block_Start
(U)
4
Read_Multiple_Block_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 2
Read_Multiple_Block_InProgress
(U)
(End)

Example Results

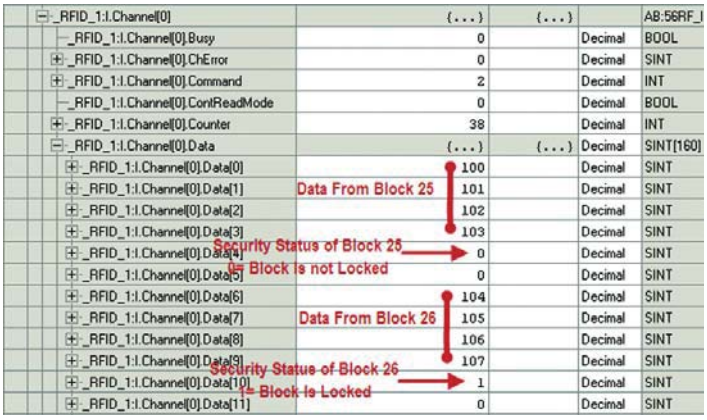
This first example uses Option Flag = 0; return only the data in the blocks. With a starting block number of 25 and two blocks to read, data from Blocks 25 and 26 are returned. The tag was a catalog number 56RF-TG-30, which has only 4 bytes per block. The data appears in the input channel Data[0…7].



⊟-_RFID_1:I.Channel[0]	{...}	{...}		AB:56RF_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	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 catalog number 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 one 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 one block of user data. Set xx:O.Channel[0].Data[0] = 0.
- Option Flag 1

  Returns one 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:

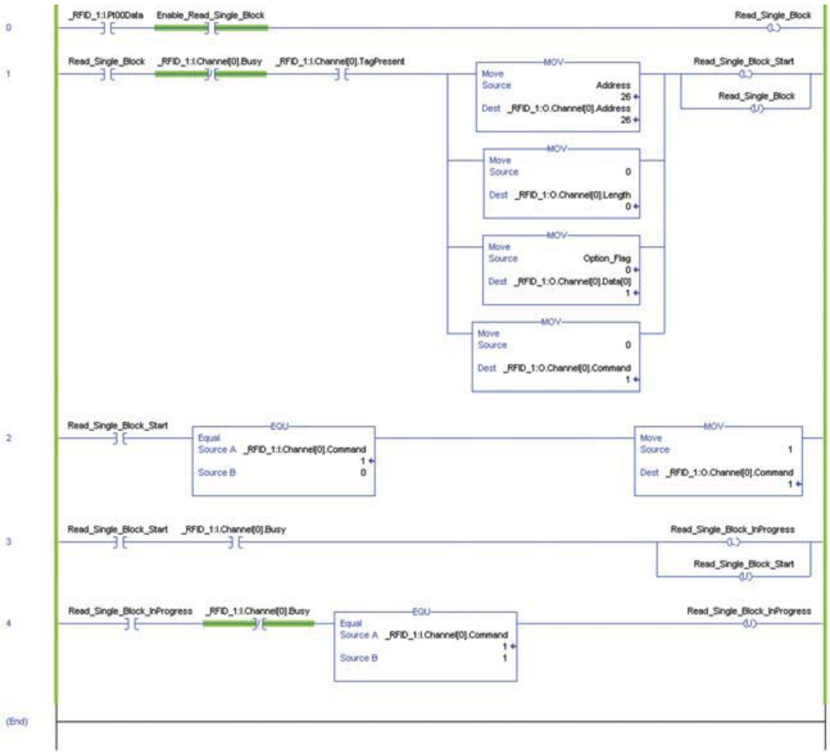
a. xx:O.Channel[0].Command = 1
b. xx:O.Channel[0].Address = the block number to read.
c. xx.O.Channel[0].BlockSize = 0
d. xx:O.Channel[0].Data[0] = the Option Flag value
e. xx:O.Channel[0].Length = 0
f. xx:O.Channel[0].Reset = 0
g. xx:O.Channel[0].Timeout = 0
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)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

Unless a UUID is specified, this command operates 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 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.

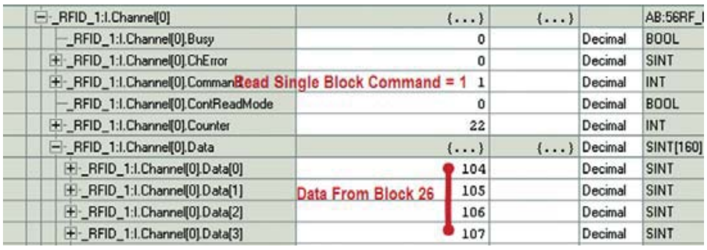
The example ladder diagram is initially set for Address =26. Data[0] is set to Option Flag 0 (return just the data). The command reads blocks 25 and 26. The example is repeated with Option Flag set to 1.


0
_RFID_1:I.Pt00Data
Enable_Read_Single_Block
Read_Single_Block
L
1
Read_Single_Block
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
26
Dest _RFID_1:O.Channel[0].Address
26
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source Option_Flag
0
Dest _RFID_1:O.Channel[0].Data[0]
1
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
1
Read_Single_Block_Start
L
Read_Single_Block
U
2
Read_Single_Block_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 0
MOV
Move
Source 1
Dest _RFID_1:O.Channel[0].Command
1
3
Read_Single_Block_Start
_RFID_1:I.Channel[0].Busy
Read_Single_Block_InProgress
L
Read_Single_Block_Start
U
4
Read_Single_Block_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 1
Read_Single_Block_InProgress
U
(End)

Example Results

- Option Flag 0

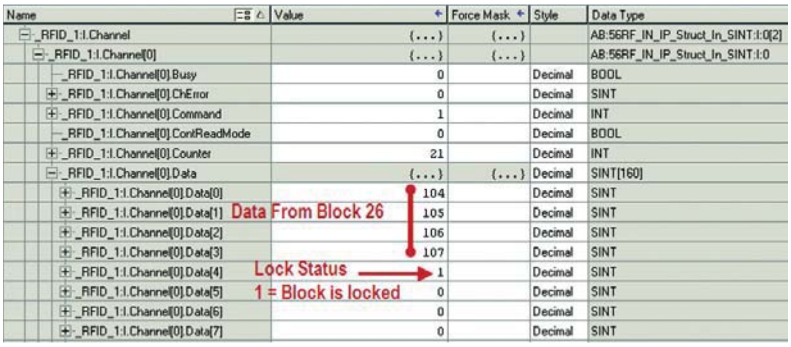
  This first example uses Option Flag = 0; return only the data in the block. The block number is 26. The tag was a catalog number 56RF-TG-30, which has only 4 bytes per block. The data appears in the input channel Data[0…3].



⊟-_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 that the block has been locked. A zero indicates that 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 retrieves the following information from the transceiver:

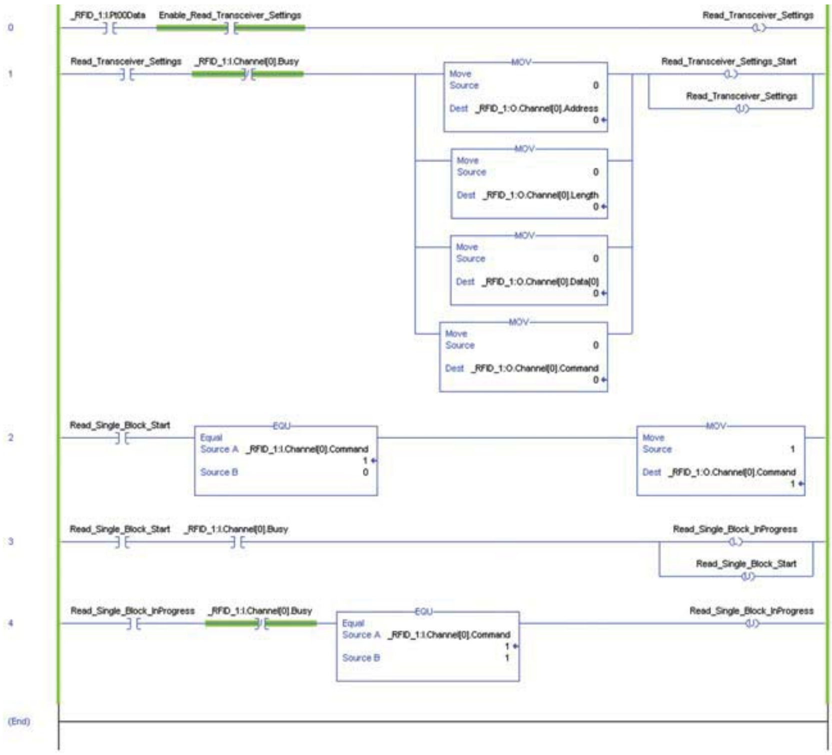
- Device ID
- Communication rate
- Retry time
- Gain

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 following example routine, 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.

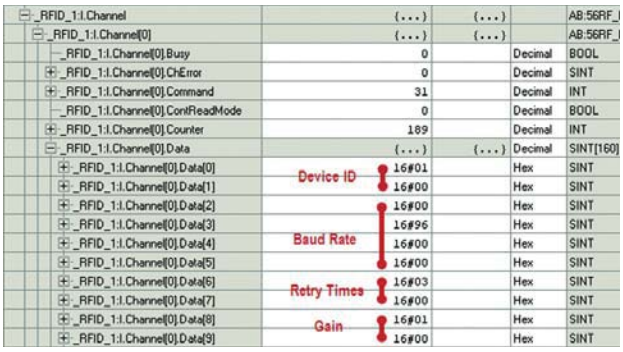

0
_RFID_1:I.Pt00Data
Enable_Read_Transceiver_Settings
Read_Transceiver_Settings
L
1
Read_Transceiver_Settings
_RFID_1:I.Channel[0].Busy
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Read_Transceiver_Settings_Start
L
Read_Transceiver_Settings
U
2
Read_Single_Block_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 0
MOV
Move
Source 1
Dest _RFID_1:O.Channel[0].Command
1
3
Read_Single_Block_Start
_RFID_1:I.Channel[0].Busy
Read_Single_Block_InProgress
L
Read_Single_Block_Start
U
4
Read_Single_Block_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 1
Read_Single_Block_InProgress
U
(End)

Example Results

The following information is displayed:

- xx:I.Channel[0].Data[0…1] = Device ID
- xx:I.Channel[0].Data[2…5] = Communication 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_
_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	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]	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[1]	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[2]	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]	16#03		Hex	SINT
_RFID_1:I.Channel[0].Data[7]	16#00		Hex	SINT
_RFID_1:I.Channel[0].Data[8]	16#01		Hex	SINT
_RFID_1:I.Channel[0].Data[9]	16#00		Hex	SINT

Write AFI

The Write AFI command writes 1 byte of information into the AFI. The AFI is used to group RFID tags by application. This configuration 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:

a. xx:O.Channel[0].Command = 41

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

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

d. xx:O.Channel[0].Data[0] = AFI value

e. xx:O.Channel[0].Length = 1

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

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

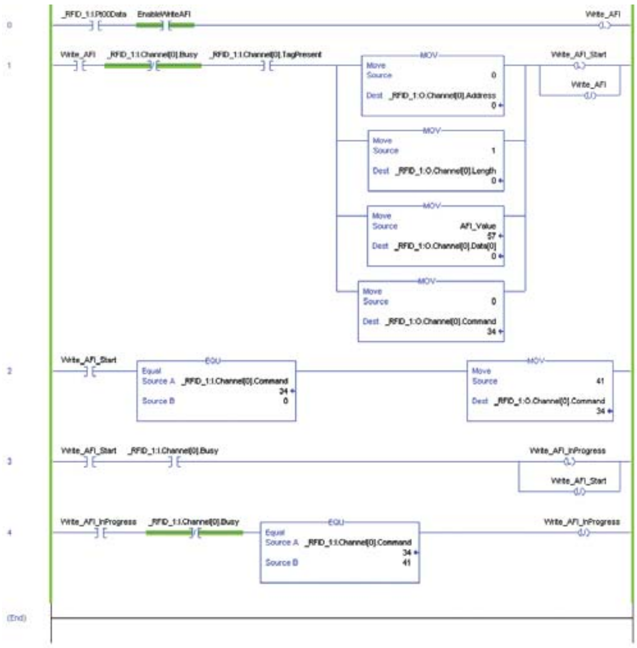
h. xx:O.Channel[0].UIDLow = 0 (or UIDLow)

i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command operates 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

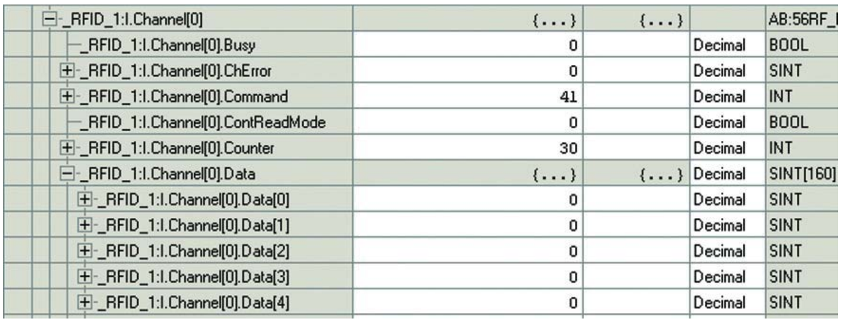
In the following example routine, the initialization sets the address, length data, and command. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.


0
_RFID_1:I.Pt00Data
EnableWriteAFI
Write_AFI
L
1
Write_AFI
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source 1
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source AFI_Value
57
Dest _RFID_1:O.Channel[0].Data[0]
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
34
Write_AFI_Start
L
Write_AFI
U
2
Write_AFI_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
34
Source B 0
MOV
Move
Source 41
Dest _RFID_1:O.Channel[0].Command
34
3
Write_AFI_Start
_RFID_1:I.Channel[0].Busy
Write_AFI_InProgress
L
Write_AFI_Start
U
4
Write_AFI_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
34
Source B 41
Write_AFI_InProgress
U
(End)

Example Results

Figure 22 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.

Figure 22 - Input Image Table



⊟-_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	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. You 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 address to 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)

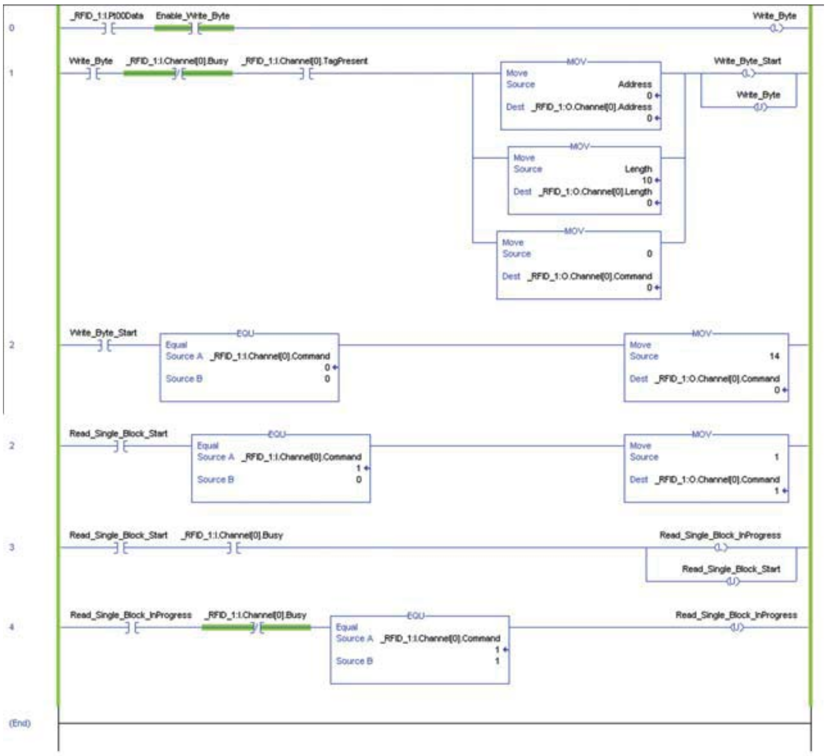
i. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

Unless a UUID is specified, this command operates 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 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.

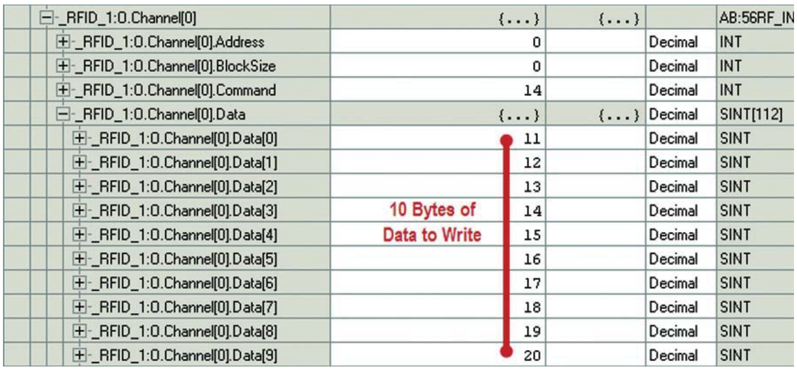
The example ladder diagram is initially set for Address =0, the Length = 10. Data[0…9] are set to a sequential list of numbers starting with 11.


0
_RFID_1:I.Pt00Data
Enable_Write_Byte
Write_Byte
(L)
1
Write_Byte
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
0
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
10
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Write_Byte_Start
(L)
Write_Byte
(U)
2
Write_Byte_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
0
Source B 0
MOV
Move
Source 14
Dest _RFID_1:O.Channel[0].Command
0
2
Read_Single_Block_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 0
MOV
Move
Source 1
Dest _RFID_1:O.Channel[0].Command
1
3
Read_Single_Block_Start
_RFID_1:I.Channel[0].Busy
Read_Single_Block_InProgress
(L)
Read_Single_Block_Start
(U)
4
Read_Single_Block_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 1
Read_Single_Block_InProgress
(U)
(End)

Example Results

Figure 23 shows the output image table with the 10 bytes of data that is written to the RFID tag. The sequence is 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20.

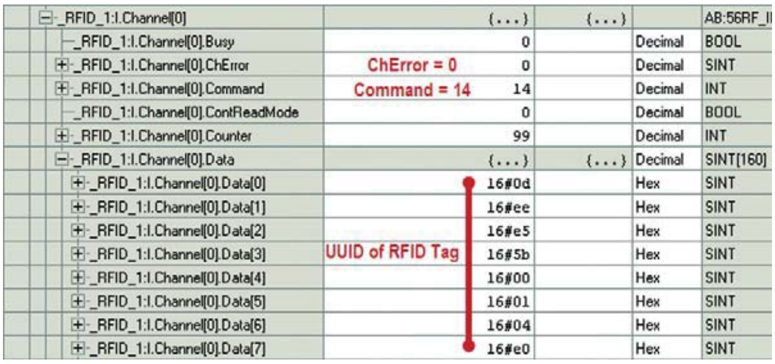
Figure 23 - Output Image Table



⊟-_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]	14		Decimal	SINT
⊞-_RFID_1:O.Channel[0].Data[4]	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

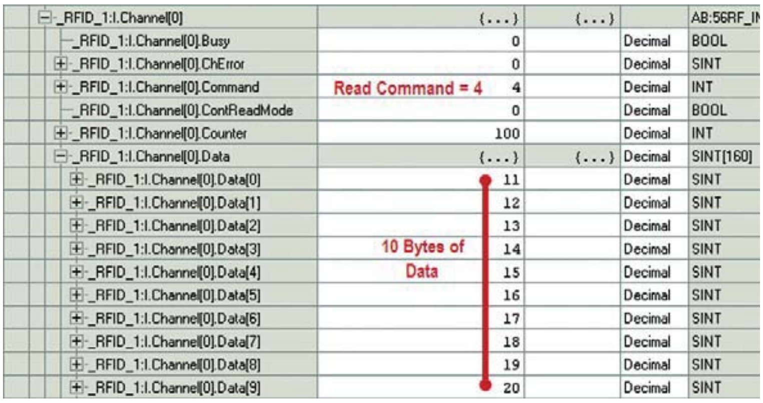
10 Bytes of Data to Write

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_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	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 writes 1 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)

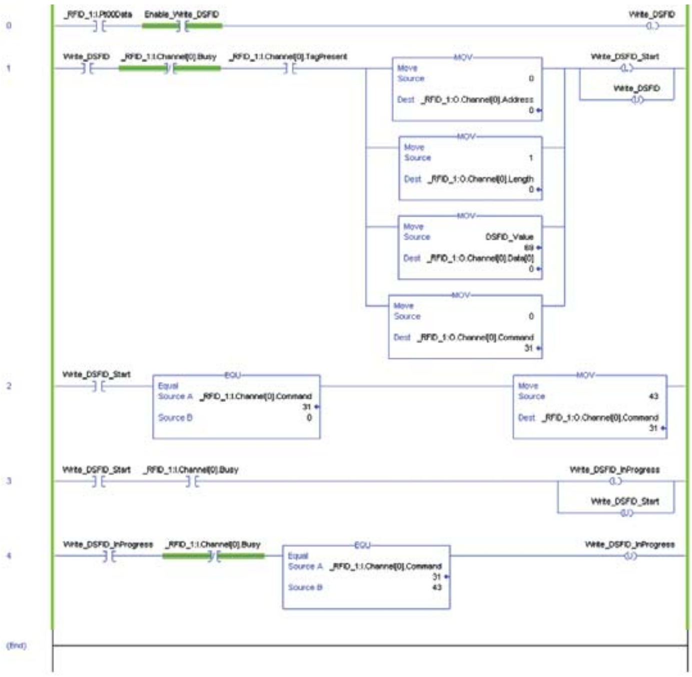
h. xx:O.Channel[0].UIDHi = 0 (or UIDHi)

If UIDLow and UIDHI are set to 0, this command operates 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 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.

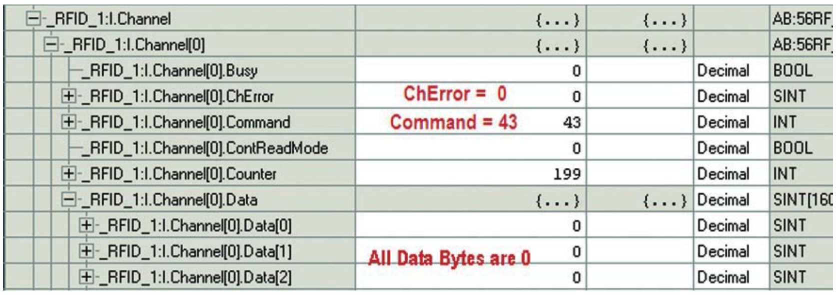
The example ladder diagram is initially set for Address =0, the Length = 0. Data[0] is set to the DSFID value.



Example Results

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

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



⊟-_RFID_1:I.Channel	{...}	{...}		AB:56RF_
⊟-_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	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]	0		Decimal	SINT
⊞-_RFID_1:I.Channel[0].Data[2]	0		Decimal	SINT

ChError = 0

Command = 43

All Data Bytes are 0

Write Multiple Blocks

The Write Multiple Blocks command writes to either one or two blocks of user data to a FRAM tag. This command only works on FRAM tags. Catalog number 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 operates 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.

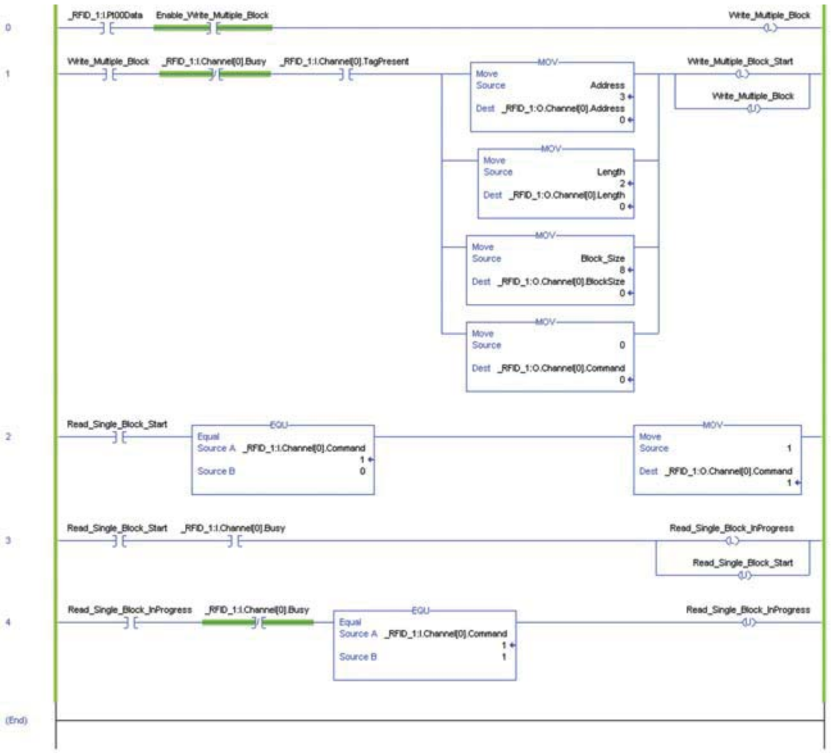
Table 26 shows the valid values for length, block size, and the number of bytes written or each combination.

Table 26 - Valid Values

Description	Value					
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

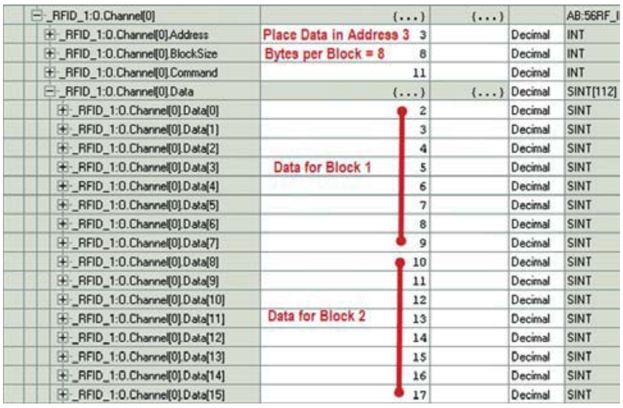
In the following example routine, the initialization in Rung 1 sets the address, length, and block size values that are used to write multiple blocks and sets the command value to 0. The BlockSize, Reset, Timeout, UIDLow, and UIDHi are set to 0 in the output image table.


0
_RFID_1:I.Pt00Data
Enable_Write_Multiple_Block
Write_Multiple_Block
(L)
1
Write_Multiple_Block
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
3
Dest _RFID_1:O.Channel[0].Address
0
MOV
Move
Source Length
2
Dest _RFID_1:O.Channel[0].Length
0
MOV
Move
Source Block_Size
8
Dest _RFID_1:O.Channel[0].BlockSize
0
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
0
Write_Multiple_Block_Start
(L)
Write_Multiple_Block
(U)
2
Read_Single_Block_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 0
MOV
Move
Source 1
Dest _RFID_1:O.Channel[0].Command
1
3
Read_Single_Block_Start
_RFID_1:I.Channel[0].Busy
Read_Single_Block_InProgress
(L)
Read_Single_Block_Start
(U)
4
Read_Single_Block_InProgress
_RFID_1:I.Channel[0].Busy
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
1
Source B 1
Read_Single_Block_InProgress
(U)
(End)

Example Results

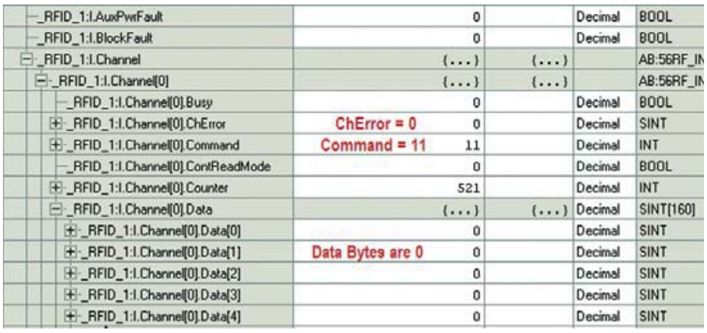
Figure 24 shows the output image table with the data that is written (a simple numeric sequence starting at 2). Two blocks of 8 bytes each is written to the tag. The data is written to address locations 3 and 4.

Figure 24 - Output Image Table



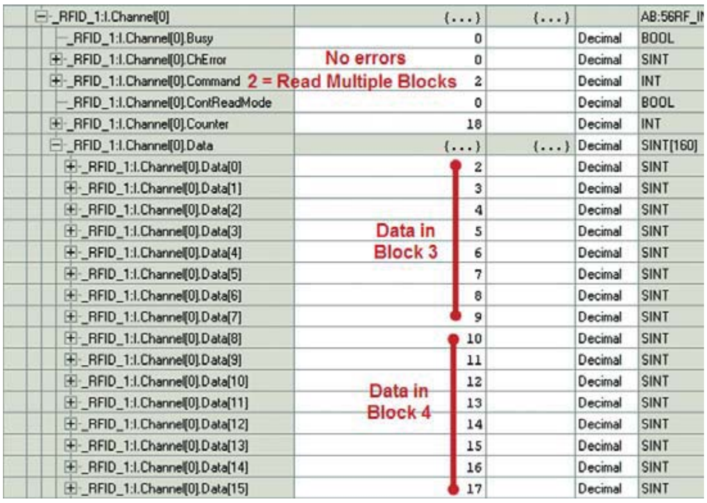
⊟ _RFID_1:O.Channel[0]		{...}	{...}		AB:56RF_I
⊞ _RFID_1:O.Channel[0].Address	Place Data in Address 3	3		Decimal	INT
⊞ _RFID_1:O.Channel[0].BlockSize	Bytes per Block = 8	8		Decimal	INT
⊞ _RFID_1:O.Channel[0].Command		11		Decimal	INT
⊟ _RFID_1:O.Channel[0].Data		{...}	{...}	Decimal	SINT[112]
⊞ _RFID_1:O.Channel[0].Data[0]		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 show 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	0		Decimal	SINT
_RFID_1:I.Channel[0].Command	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]	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	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	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]	5		Decimal	SINT
_RFID_1:I.Channel[0].Data[4]	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]	13		Decimal	SINT
_RFID_1:I.Channel[0].Data[12]	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 Multi-tag Block Write command writes one or more blocks of user data to multiple tags in the transceiver field. The maximum number of 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 operates 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.

IMPORTANT	Length must be in 4-byte increments (for example, 4, 8, 12…) for ISO15693 tags or 8-byte increments (for example, 8, 16, 24…) for FRAM tags.

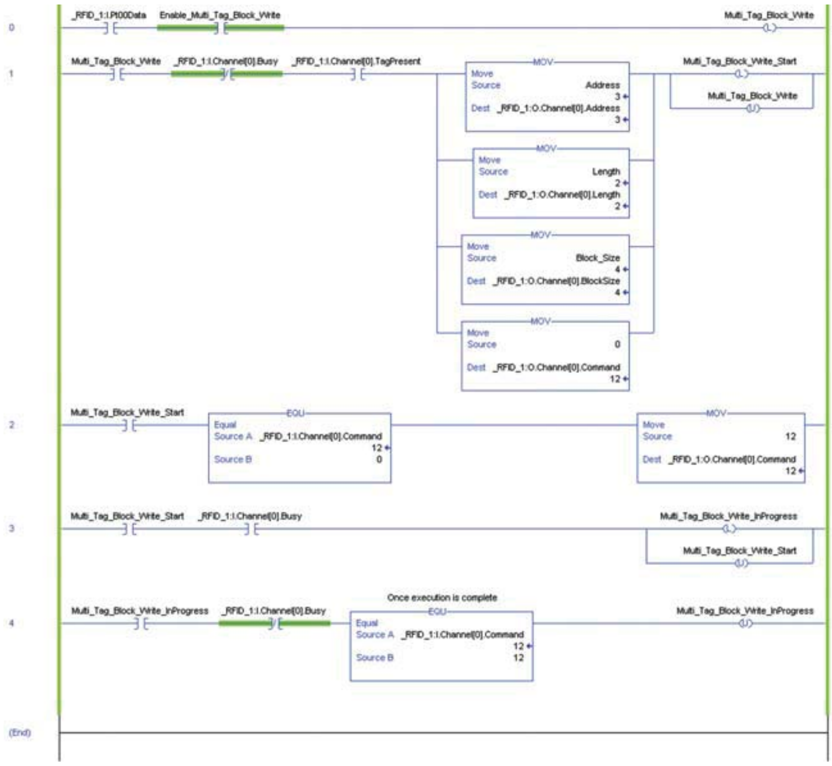
IMPORTANT The BlockSize field is used to specify the number of bytes/block of the tag. Valid values are:

- 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.

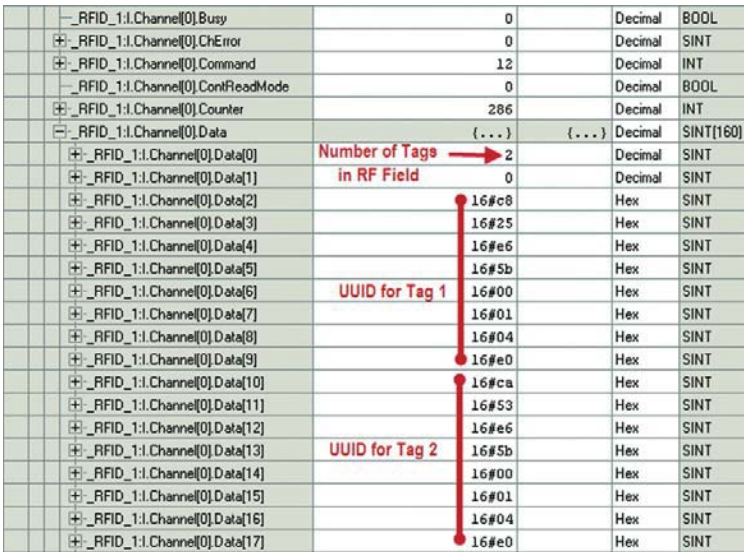
Example Routine

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


0
_RFID_1:I.Pt00Data
Enable_Multi_Tag_Block_Write
Multi_Tag_Block_Write
(L)
1
Multi_Tag_Block_Write
_RFID_1:I.Channel[0].Busy
_RFID_1:I.Channel[0].TagPresent
MOV
Move
Source Address
3
Dest _RFID_1:O.Channel[0].Address
3
MOV
Move
Source Length
2
Dest _RFID_1:O.Channel[0].Length
2
MOV
Move
Source Block_Size
4
Dest _RFID_1:O.Channel[0].BlockSize
4
MOV
Move
Source 0
Dest _RFID_1:O.Channel[0].Command
12
Multi_Tag_Block_Write_Start
(L)
Multi_Tag_Block_Write
(U)
2
Multi_Tag_Block_Write_Start
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
12
Source B 0
MOV
Move
Source 12
Dest _RFID_1:O.Channel[0].Command
12
3
Multi_Tag_Block_Write_Start
_RFID_1:I.Channel[0].Busy
Multi_Tag_Block_Write_InProgress
(L)
Multi_Tag_Block_Write_Start
(U)
4
Multi_Tag_Block_Write_InProgress
_RFID_1:I.Channel[0].Busy
Once execution is complete
EQU
Equal
Source A _RFID_1:I.Channel[0].Command
12
Source B 12
Multi_Tag_Block_Write_InProgress
(U)
(End)

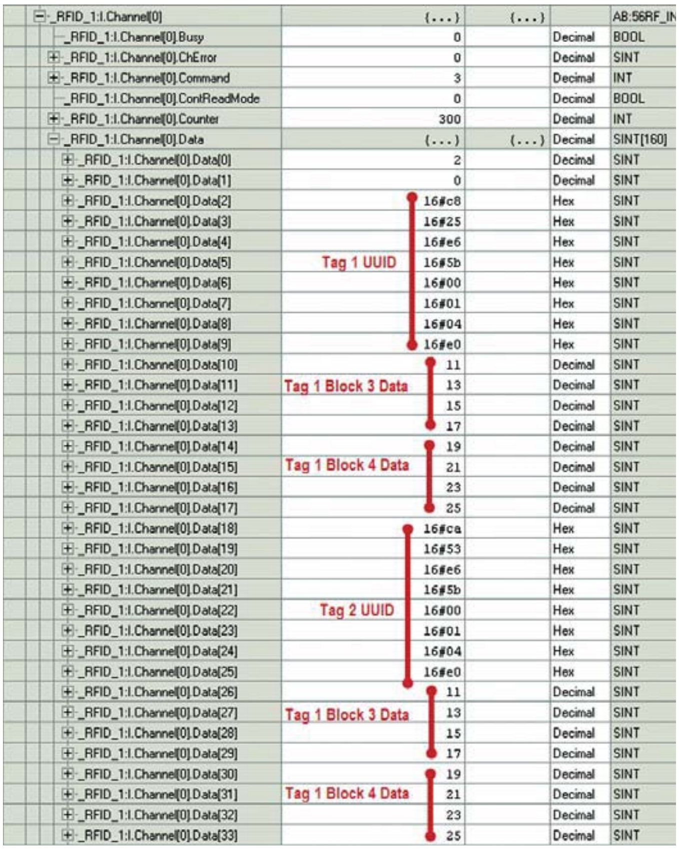
Example Results

The input channel image table shows 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]	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]	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#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]	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 that the data was written.



_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	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]	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]	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]	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]	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]	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]	21		Decimal	SINT
_RFID_1:I.Channel[0].Data[32]	23		Decimal	SINT
_RFID_1:I.Channel[0].Data[33]	25		Decimal	SINT