TURCK

Industrial
Automation

RM-89
Ethernet/IP™ Encoder

MANUAL

MA1006
Published 12/03/2013

ABOUT THIS MANUAL

Read this chapter to learn how to navigate through this manual and familiarize yourself with the conventions used in it. The last section of this chapter highlights the manual's remaining chapters and their target audience.

Audience

This manual explains the installation and operation of TURCK's RM-89 Networked encoders. It is written for the engineer responsible for incorporating the RM-89 into a design as well as the engineer or technician responsible for its actual installation. If there are any unanswered questions after reading this manual, call the factory. An applications engineer will be available to assist you.

Manual Conventions

Three icons are used to highlight important information in the manual:

NOTES: highlight important concepts, decisions you must make, or the implications of those decisions.
CAUTIONS: tell you when equipment may be damaged if the procedure is not followed properly.
WARNINGS: tell you when people may be hurt or equipment may be damaged if the procedure is not followed properly.

Trademarks and Other Legal Stuff

Where to Go From Here

This manual contains information that is of interest to everyone from engineers to operators. The table below gives a brief description of the content of each chapter to help you find the information you need to assist you in your job.

CHP NUM.	Chapter Title	Chapter Description
1	INTRODUCTION TO THE RM-89	Intended for anyone new to the RM-89 networked encoder, this chapter gives a basic overview of the unit, including an explanation of its programmable features. The chapter also explains the RM-89 part numbering system.
2	INSTALLATION	This chapter is intended for the engineer or technician responsible for installing and wiring the RM-89 networked encoder. Information in this chapter includes mechanical drawings, installation guidelines and connector pinout.
3	RSLogix 5000 EtherNet/IP CONFIGURATION	This chapter covers how to communicate with the RM-89 using the EtherNet/IP protocol and implicit messaging. The RSLogix 5000 software is used as a programming example.
4	RSLogix 500 EtherNet/IP CONFIGURATION	This chapter covers how to communicate with the RM-89 using the EtherNet/IP protocol and explicit messaging. The RSLogix 500 software is used as a programming example.
5	Modbus TCP CONFIGURATION	This chapter covers how to communicate with the RM-89 using the Modbus TCP protocol.
A	IP ADDRESS SETUP WITH BOOTP	Older RM-89 units must use a Bootp server to set their IP address. This appendix is a step by step guide to using the Bootp server from Rockwell Automation to change the default IP address of the RM-89.
B	Modbus TCP CONFIGURATION	When using the EtherNet/IP protocol, the RM-89 conforms to the Encoder Device Profile as defined by the Common Industrial Protocol (CIP). This chapter explains the Position Sensor Object that is implemented by the RM-89 as part of this profile.

CHAPTER 1
INTRODUCTION TO THE RM-89

Overview

The RM-89 is a new line of heavy-duty resolver based encoder products from TURCK. The initial offerings in this line communicate over a standard Ethernet network.
Two different protocol stacks are built into each RM-89 Ethernet product:
An ODVA compliant EtherNet/IP protocol stack
A Modbus TCP compliant protocol stack
Having both protocols available allows the RM-89 to be used in a vast majority of applications today and allows machine builders to confidently choose a single solution that can be used regardless of the protocol their end customers are using.
Power over Ethernet (PoE) is also a standard feature on all RM-89 Ethernet products, allowing you to reduce cabling requirements if you use a network switch that supports PoE. Side connect units have only the Ethernet connector and must use PoE while end connect units have a second connector that can be used for power. This connector allows you to use these RM-89 encoders as drop in replacements for other EtherNet/IP encoders.
The RM-89 series is composed of absolute single- or multi-turn sensors in an IP67 rated, 2.5 inch diameter package. All RM-89 networked encoders offer a maximum single turn position resolution of 16 bits, 65,536 counts per turn and encodes 4,096 turns (12 bit).
Every RM-89 resolver based encoder is programmable over its Ethernet interface. Initial configuration can be accomplished with the TURCK Net Configurator software while setting the IP address of the unit. Additional configuration can be accomplished once the unit is installed on your machine through simple data reads and writes programmed into your controller. Parameters allow you to set the count direction, the number of counts per turn, the format of the velocity data, and preset the position data to any value within its range. The current version of the firmware also allows you to set the number of counts before returning to zero.
All RM-89 resolver based encoders have three status LED's to help you determine the state of the device. These LED's are always located on the back cover of the RM-89.
Module Status – Operating status of the RM-89 itself
Network Status – Operating state of the EtherNet/IP or Modbus TCP protocol
Link/Activity – Physical state of Ethernet connection

Electrical Specifications

Operating Voltage (External Supply)
12Vdc to 54Vdc

Power Requirements
2.5W max.
100mA @ 24Vdc typical

Ethernet Capability
10/100 Mbit autosense with auto-switch capability. Auto-switch eliminates the need of a crossover cable in all applications.

Power over Ethernet (PoE)
Compatible with Power over Ethernet standard. With only data pairs available, the power sourcing equipment (PSE) must be able to output power on these two pairs (Mode A)

Single Turn Resolution
Programmable from 1 to 65,536 counts per turn (16 bit resolution max.)

Multi-turn Resolution
4,096 turns (12 bit) or 16,384 (14 bit)

Direction of Increasing Counts
Default of CW increasing when looking at the shaft. Programmable to CCW increasing over the EtherNet/IP interface.

Preset Position
Position can be preset to any value within its range over the Ethernet interface. Internal Position Offset can be stored in non-volatile memory and retrieved on power up.

Positional Accuracy
±10 arc-minutes

Response Time
1 millisecond

Mechanical Specifications

Package Style
2.5 inch housing with flange or servo

Connector Location
End opposite shaft (axial)

Housing
Powder coated aluminum

Max. Starting Torque @ 25°C
2.0 oz-in

Moment of Inertia (oz-in-sec2)
6.00 X 10-4

Max. Operating Speed
6000 RPM max.

Max. Shaft Loading
Axial: 20lbs. (89N)
Radial: 40lbs. (178N)
At specified max. loads, minimum bearing life is 2X109 revolutions.

Environmental Specifications

Operating Temperature
–40°F to +185°F (–40°C to +85°C)

Shock
50g, 11 millisecond duration

Vibration
20g, 5 to 2000Hz

Enclosure Rating
IP67

Approximate Weight

Available Data

All RM-89 encoders offer position and velocity data that can be scaled with the programmable parameters as described in the following section. The position data can also be preset which allows you to align the position data with your machine position without having to physically rotate the shaft.

RM-89 encoders can transmit the following additional data:

Time Stamp: The time stamp is an unsigned double integer value with an interval of 400 nanoseconds. It will roll over every 1717.9869184 seconds. The time stamp can be used to verify active communications between the RM-89 and your host controller.

Actual Sensor Reading: This unsigned double integer value is the raw position data from the RM-89. Changing the position scaling parameters will have no effect on this value.

Programmable Parameters

The following parameters are available on all RM-89 encoders. Note that most of these parameter names are pulled from the ODVA (EtherNet/IP) specification. They are generic, and sometimes confusing, but they are what is defined in the specification. TURCK has decided to adopt these parameter names for all RM-89s to avoid additional translations between protocols with one exception. In the ODVA specification, the parameter that sets the number of counts per turn of the shaft is called the 'Measurement Units per Span'. This generic name can be applied to both rotary and linear encoders. Being that the RM-89 is a rotary encoder, this manual refers to the parameter as Counts per Turn.

Direction Counting Toggle

This parameter allows you to set the direction of shaft rotation needed to produce increasing counts. A value of "0" sets the direction of increasing counts to clockwise when looking at the shaft. A value of "1" sets the direction of increasing counts to counter-clockwise when looking at the shaft. The factory default value is clockwise increasing counts.

Scaling Function Control

The RM-89 has a maximum resolution of 65,536 counts per turn and it can be programmed to scale the counts per turn from 1 to 65,536. Scaling is only performed if this parameter is in its "enabled" state. A value of "0" disabled the scaling function and a value of "1" enables the scaling function. Note that the Scaling Function Control parameter only affects the scaling of the position data. Once this parameter is enabled, the velocity data will always be scaled by the Counts per Turn parameter until power is cycled to the unit. This is true even if the Scaling Function Control is returned to the disabled state. See Calculating Position and Velocity Data on page 7 for the reasoning behind this behavior.

Counts per Turn (ODVA: Measuring Units Per Span)

This parameter can range from 1 to 65,536. Note that this parameter is only used to scale the position value when the Scaling Function Control parameter is set to "1". If the Scaling Function Control parameter is set to "0", the RM-89 will report position data at its full resolution of 65,536 counts per turn. This parameter will always be applied to the velocity data reported by the RM-89, regardless of the state of the Scaling Function Control parameter. If you are not using the Counts per Turn parameter, set it to its default value of 65,536.

Preset Value

This parameter allows you to preset the position to any value in its single or multi-turn range without rotating the shaft. The minimum value for this parameter is zero. Its maximum value depends on the RM-89 version you have. For single turn RM-89's, the maximum value is 65,535. For 28 bit multi-turn RM-89's, the maximum value is 268,435,455. The maximum value of this parameter can be limited by the Total Measurement Range parameter. See the Total Measurement Range Parameter section on page 7 for a description of this parameter

Velocity Format

The RM-89 reports velocity data as well as position data over the network. This parameter sets the units of measure for the velocity data. This parameter has four fixed values.
0x1F04 = pulses/second
0x1F05 = pulses/millisecond
0x1F07 = pulses/minute
0x1F0F = RPM
Once the Scaling Function Control parameter is enabled, the velocity data will always be scaled by the Counts per Turn parameter until power is cycled to the unit. This is true even if the Scaling Function Control is returned to it disabled state. See Calculating Position and Velocity Data below for the reasoning behind this behavior.

Device Type

You can program how the RM-89 defines itself to the EtherNet/IP network and is only available when the RM-89 is configured to use the EtherNet/IP protocol. This parameter has a double integer (32 bit) data type and two fixed values.
0x22 = Encoder Device (factory default value)
0x00 = Generic Device

Calculating Position and Velocity Data

The maximum position resolution of an RM-89 is 65,536 counts per turn. This value is used unless the Scaling Function Control is set to its Enabled state. If this parameter is set to its enabled state, the number of counts per turn is set to the value specified by the Counts per Turn parameter.
Note that the Scaling Function Control parameter is a true enable/disable control. The Counts per Turn parameter is only used to scale the position data if the Scaling Function Control is in its enabled state. If you change the Scaling Function Control parameter to its disabled state, the RM-89 will begin to report position data with a resolution of 65,536 counts per turn as soon as the state change is accepted.
The velocity data calculation is also affected by the Counts per Turn parameter. The velocity data will always be calculated based on the last value of the Counts per Turn parameter. This is true even if the Scaling Function Control is never set to its Enabled state. Therefore, leave the Counts per Turn parameter at its default value of 65,536 if you do not want to scale the velocity data. For example, if you enable the Scaling Function Control and set the Counts per Turn parameter to 10,000, the position will be calculated at 10,000 counts per turn and the velocity will also be calculated at 10,000 counts per turn. If you then disable the Scaling Function Control, the position will be calculated at 65,536 counts per turn and the velocity will still be calculated at 10,000 counts per turn. Additionally, if you change the Counts per Turn parameter to 5,000 and do not enable the Scaling Function Control parameter, the position will still be calculated at 65,536 counts per turn and the velocity will now be calculated at 5,000 counts per turn. This behavior may be confusing to some users, but may be exactly what other users need. (One example is a packaging machine where you want the position at full resolution, but the velocity data scaled to boxes-per-minute.) If this behavior would not be beneficial to you, then the best way to avoid any issues is to always leave the Scaling Function Control parameter enabled and use the Counts per Turn parameter, even when setting the counts per turn to 65,536.

Total Measurement Range Parameter

The Total Measurement Range parameter sets the total number of counts before the position value returns to zero. It is always used when determining the position value. Its use is not affected by the state of the Scaling Function Control parameter. If the Total Measurement Range parameter is left at its default value of zero, the roll over position is determined by the Counts per Turn parameter and the number of turns the RM-89 can encode. If the Total Measurement Range is non-zero, it places an upper limit on the position value and the Preset Value parameter. Total Measurement Range parameter ranges are as follows:
Single Turn RM-89: Range of 0, 2 to 65,536
28 bit Multi-turn RM-89: Range of 0, 2 to 268,435,455
There is no fixed relationship between the Total Measurement Range and Counts per Turn parameters, which leads to interesting applications that use the two parameters.

Storage of Internal Position Offset

The Total Measurement Range parameter affects how the internal position offset, which is generated when you preset the position value, is stored. When the Total Measurement Range parameter is zero, the position offset is stored in RAM and lost when power is removed from the RM-89. You must issue a command to save the position offset to non-volatile memory. When the Total Measurement Range parameter is non-zero, the internal position offset is automatically stored in non-volatile FRAM memory. You do not need to issue a command to save the internal position offset. If you set the Total Measurement Range parameter as follows, the parameter will have no effect on the position value and the internal position offset will be stored in FRAM.

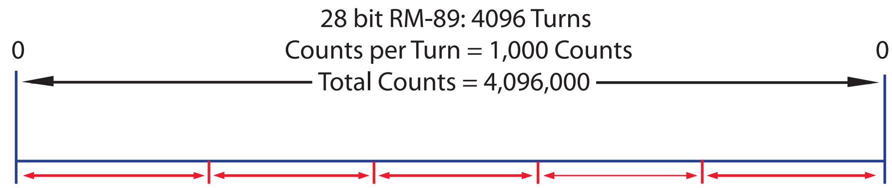
Single Turn RM-89: 65,536 or the value of the Counts per Turn parameter if the Scaling Function Control parameter is set to 'True'.

28 bit Multi-turn RM-89: 268,435,455, or the value of the Counts per Turn parameter multiplied by 4,096 if the Scaling Function Control parameter is set to 'True'.

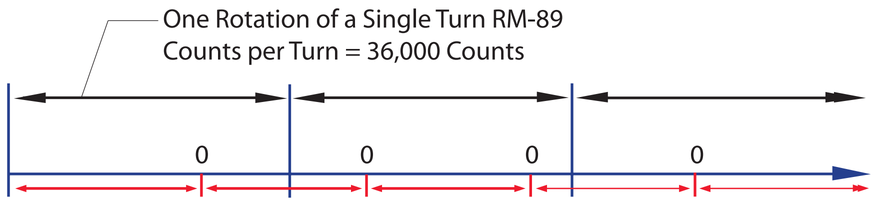
> **NOTE:**
> Using the Total Measurement Range parameter this way only affects how the internal position offset is stored. You must still issue a command to save the programmable parameters to non-volatile memory.

Roll Over on Fractional Travel

When the Total Measurement Range is less than the total counts available from the RM-89, which is (Counts per Turn multiplied by the number of turns the RM-89 can encode), the position will return to zero before the full mechanical travel is completed. Two examples are shown below.



Total Measurement Range = 819,200 Counts
RM-89 outputs five cycles of counts from 0 to 819,199 over 4,096 turns



Total Measurement Range = 22,500 Counts
RM-89 outputs one cycle of counts from 0 to 22,499 for every 225 degrees of rotation.

Figure 1.3 Fractional Turn Examples

The top half of figure 1.3 shows what occurs when the Total Measurement Range parameter is used to divide the full range of travel of the RM-89 into equal parts. In this case, a twenty-eight bit RM-89 has its 4,096 turns evenly divided into five cycles of 819.2 turns.

The bottom half of figure 1.3 shows a single turn RM-89 where the Total Measurement Range parameter is not used to divide the full range of travel into equal parts. In this case, the position value will roll over to zero after 225 degrees of rotation.

If the value of {Total Counts ÷ Total Measurement Range} is an integer, the RM-89 remains an absolute rotary sensor. You can remove power from the RM-89, rotate it as far as you want, re-apply power, and the RM-89 will give you the correct position value. The top half of figure 1.3 is an example of this setup because the division of the two parameters results in the quotient value of five.

It is also possible for the value of {Total Counts ÷ Total Measurement Range} to be a real number instead of an integer. This case is shown in the bottom half of figure 1.3, where the quotient is 1.6. In these cases, the RM-89 becomes what TURCK terms a quasi-absolute rotary sensor.

Quasi-absolute means that the RM-89 will power up with the correct position value as long as the shaft was rotated less than half of the complete span of the encoder while power was removed. In practical terms:

For 30-bit RM-89 encoders: If you remove power from the sensor and rotate the shaft less than 8,192 turns in either direction, when you re-apply power, the position reading will be correct.

For 28-bit RM-89 encoders: If you remove power from the sensor and rotate the shaft less than 2,048 turns in either direction, when you re-apply power, the position reading will be correct.

For 16-bit RM-89 encoders: If you remove power from the sensor and rotate the shaft less than 180 degrees in either direction, when you re-apply power, the position reading will be correct.

If the shaft rotates further than the limits listed above while power is removed, the position value from the RM-89 will be off by at least ±1 turn when power is applied.

Quasi-Absolute Multi-turn

When the Total Measurement Range is greater than the total counts available from the RM-89, which is (Counts per Turn multiplied by the number of turns the RM-89 can encode), multiple rotations of the shaft are required before the position value reaches the roll over count. For example, assume a single turn RM-89 that has its Counts per Turn parameter set to 360 and its Total Measurement Range parameter set to 64,800. With this setup, the shaft of the RM-89 must rotate 180 turns, {64,800 ÷ 360}, before the position returns to zero. In this application, the single turn RM-89 acts as a 180 turn encoder with one degree position resolution.

The same trade off between resolution and number of turns encoded can be made with the multi-turn RM-89 encoders. For example, if a 30-bit RM-89 encoder has its Counts per Turn parameter set to 360 and its Total Measurement Range parameter set to its maximum of 1,073,741,824, the RM-89 will encode 2,982,616.17 turns with one degree resolution.

In all of these applications, the RM-89 will act as a quasi-absolute encoder, with the same motion restrictions listed in the Roll Over on Fractional Travel section above. Exceeding these limits will result in a position value error when power is re-applied.

Effects of Reversing Count Direction

Changing the Direction Counting Toggle parameter changes the way the position value is calculated. When you reverse the count direction, the position changes from your current position value to (Maximum number of counts – current position value). For example, assume a 30 bit RM-89 with its default of 65,536 counts per turn. If the current position value is 100,000 and you change the Direction Counting Toggle parameter, the current position will change to (230 – 100,000 = 1,073,741,824 – 100,000) = 1,073,641,824. Most applications do not require you to change the count direction after the machine is setup, so the count direction is typically set before the position value is preset.

Changing the count direction on your machine while maintaining the current position value is a three step process. First, read and store the current position value from the RM-89. Second, change the Direction Counting Toggle value. Third, write the stored position value back to the RM-89 as a preset value.

CHAPTER 2
INSTALLATION

End View

The Status LED's on all RM-89's are located on the end of the unit opposite of the input shaft. Figure 2.1 below shows the layout of a unit with the connectors also located on the end. For side connect units, only the LED's will be on this surface. Also note that only the Network connector is available on side connect units and these devices must use Power over Ethernet. All end-connect units also include a screw on cap for the power connector so the unit will retain its IP67 rating if you decide to use it as a PoE device.

Status LED's

As shown in figure 2.1 above, the RM-89 has three status LED's on the rear cover. These LED's are present on side connector units as well. The tables below list the various states of the LED's and their meaning.

Network Status LED

LED State	EtherNet/IP Definition	Modbus TCP Definition
Off	No Power	No power or no TCP connections
Alternating Red/Green	Power up Self-Test	Power up Self-Test
Flashing Green	No Ethernet network connections	Indicates number of concurrent connections with 2 second delay between group. The RM-89 supports up to 3 concurrent connections.
Steady Green	Ethernet network connected	Should not occur. LED should always flash when network is connected.
Flashing Red	Network Connection Timeout	Not implemented in Modbus TCP
Steady Red	Duplicate IP address on network.	

Table 2.1 Network Status LED States

Module Status LED

LED State	Definition
Off	No Power
Alternating Red/Green	Self-Test (Run on power up.)
Steady Green	Device Operational
Steady Red	Hardware Fault. (Cycle power. If fault persists, contact TURCK for support.)

Table 2.2 Module Status LED States

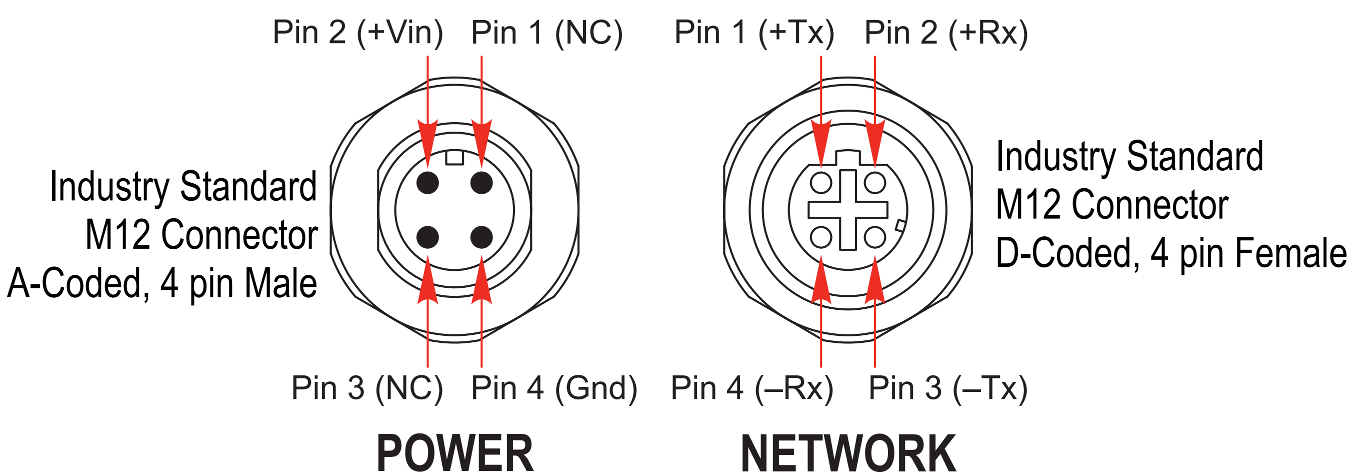
Status LED's (continued)

Link/Activity LED
This orange LED is on when a Ethernet hardware connection exists to the RM-89 and blinks when there is activity on the RM-89 Ethernet network segment. Note that this LED shows the state of the hardware network, not the EtherNet/IP or Modbus TCP protocols.

Connector Pinout

The diagram below shows the pinout of the RM-89 connectors. The Power and Network connectors are both available on units with end connectors. Side connect units only have the Network connector and must be powered using the Power over Ethernet feature of the RM-89.All end-connect units also include a screw on cap for the power connector so the unit will retain its IP67 rating if you decide to use it as a PoE device.

Figure 2.2 Connector Pinout



ANSI/TIA-568-C.2 Color Codes
There are two color codes in common use when wiring Ethernet connections with twisted pairs. Either one of these standards is acceptable. The TURCK Contact cordsets all follow the 568B standard. Note that accidently reversing the Tx/Rx pairs will not affect the operation of the RM-89. The RM-89 has an "auto-sense" port that will automatically adjust for swapped pairs.

Signal	568A Color	568B Color
+Transmit (+Tx)	White/Green Tracer	White/Orange tracer
–Transmit (–Tx)	Solid Green	Solid Orange
+Receive (+Rx)	White/Orange Tracer	White/Green Tracer
–Receive (–Rx)	Solid Orange	Solid Green

Table 2.3 ANSI/TIA Color Codes

Compatible Connectors and Cordsets

Connectors

PART NUMBER#	Description
CMB 8141-0	Mating connector for Data/PoE connector. Screw terminal connections. 6 to 8 mm dia. cable. Straight, IP67 rated when properly installed.
BM 8151-0/PG9	Mating connector for Power connector. Screw terminal connections. 6 to 8 mm dia. cable. Straight, IP67 rated when properly installed.

Table 2.4 Compatible Connectors

Ethernet Cordsets

PART NUMBER #	Description
RSSD 4410-2M	Bus system cable: 4-position, 26AWG, shielded. ANSI/TIA 568-C.2 color coded Plug: Straight M12, D-coded, free conductor end. Cable length: 2 m
RSSD 4410-5M	Bus system cable: 4-position, 26AWG, shielded. ANSI/TIA 568-C.2 color coded Plug: Straight M12, D-coded, free conductor end. Cable length: 5 m
RSSD 4410-10M	Bus system cable: 4-position, 26AWG, shielded. ANSI/TIA 568-C.2 color coded Plug: Straight M12, D-coded, free conductor end. Cable length: 10 m
RSSD 4410-15M	Bus system cable: 4-position, 26AWG, shielded. ANSI/TIA 568-C.2 color coded Plug: Straight M12, D-coded, free conductor end. Cable length: 15 m

Table 2.5 Ethernet Cordsets

NOTE:
These cordsets include the RM-89 connector, but the other end is un-terminated. This end can be punched down onto a patch panel, or an RJ45 connector can be added if the cordset is plugged directly into a switch.

Power over Ethernet (PoE)

All RM-89 Ethernet encoders can be powered using only the network connector if the network supports Power of Ethernet. Because the RM-89 connector only has the ±Tx and ±Rx pairs, the network device the RM-89 is cabled to, which is called the power source equipment (PSE) in the standard, must be able to output power on these two pairs (Mode A).

All end connect RM-89 devices have a second connector for an external supply. The RM-89 contains power supply sensing logic that will use the external power supply if power is available on that connector. No operating power will be drawn from the PoE. The RM-89 will not be damaged if power is supplied on both connectors. All end-connect units also include a screw on cap for the power connector so the unit will retain its IP67 rating if you decide to use it as a PoE device.

If an end connect RM-89 is using PoE, the power connector must be sealed to maintain the unit's IP67 rating.

NOTE:
You must use a PoE switch or injector. Do not attempt to connect an external supply to the ethernet data pins.

CHAPTER 3

RSLogix 5000 EtherNet/IP CONFIGURATION

Rockwell Automation Ethernet products use the EtherNet/IP protocol. This chapter shows how to configure communications between an RM-89 and your PLC using RSLogix 5000 software. It also shows how to use the data tags assigned to the RM-89 to read and write data to the encoder.

Implicit Messaging

Every PLC that is programmed using RSLogix 5000 software supports implicit messaging. These include the ControlLogix and CompactLogix platforms. Implicit messaging means that the PLC processor will automatically exchange data with the RM-89 at the programmed RPI time, thereby simplifying the use of the RM-89.

The other form of communication is explicit messaging. In explicit messaging, the PLC processor will only communicate with the RM-89 when explicitly told to through instructions that are programmed into your ladder logic. You can use explicit messages in your RSLogix 5000 programs, but for most users the additional complexity is not necessary. Explicit messaging is explained in the next chapter, RSLogix 500 EtherNet/IP CONFIGURATION, starting on page 55 because explicit messaging is the only form of communication supported by platforms that are programmed with the RSLogix 500 software package.

RSLogix 5000 Configuration

When using the ControlLogix and CompactLogix platforms, you have the option of using the Ethernet port that is built into some processors, or a separate Ethernet Bridge module.

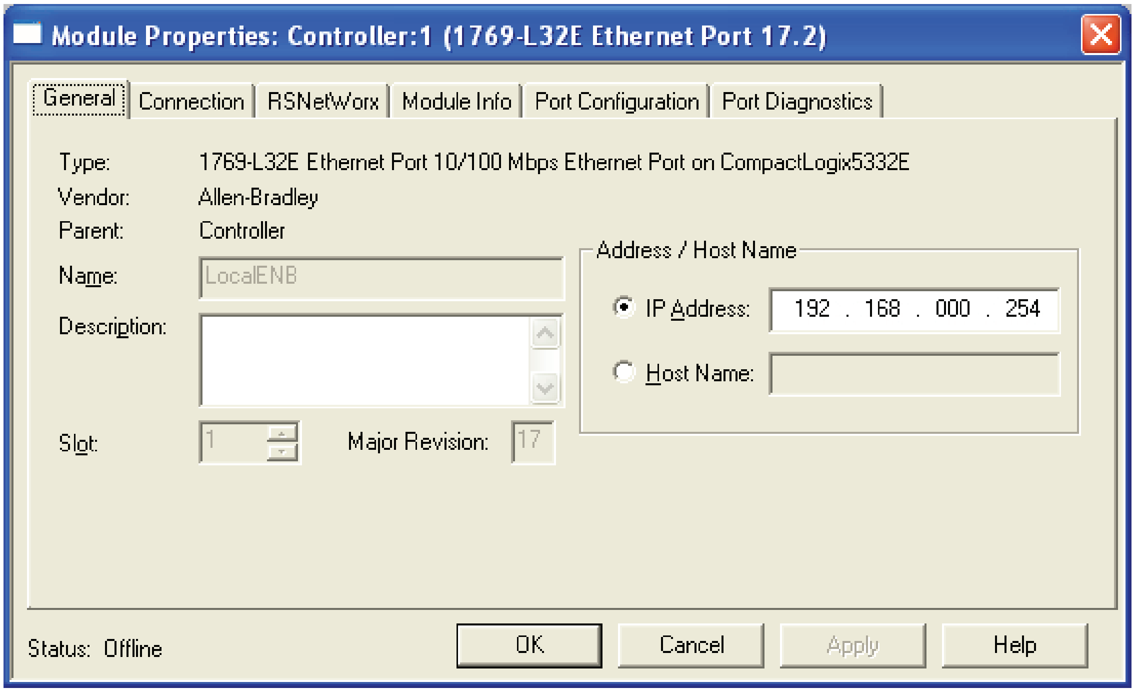
If the Ethernet port is built into processor, the only step you have to take before adding the RM-89 is to create a new project with the correct processor or modify an existing project. Once this is done, the Ethernet port will automatically appear in the Project Tree. If you are using an Ethernet Bridge module, you will have to add the module to the I/O Configuration tree and configure it before adding the RM-89 to your project.

> **NOTE:**
> If you are using an Ethernet Bridge module and have difficulty communicating with the RM-89, you may have to upgrade the firmware of the Ethernet Bridge module to its latest version.

Configuring a Built-in Ethernet Port (As Needed)

You have to set an IP address for the Ethernet Port if the port is built into your processor. Right click on the port name in the I/O Configuration screen and select "Properties". A Module Properties window similar to the one shown in figure 3.1 will open. In this window you must set an IP Address for the port, not the IP address of the RM-89.

Figure 3.1 Setting Ethernet Port Parameters


Module Properties: Controller:1 (1769-L32E Ethernet Port 17.2)
General
Connection
RSNetWorx
Module Info
Port Configuration
Port Diagnostics
Type:
1769-L32E Ethernet Port 10/100 Mbps Ethernet Port on CompactLogix5332E
Vendor:
Allen-Bradley
Parent:
Controller
Name:
LocalENB
Description:
Slot:
1
Major Revision:
17
Address / Host Name
IP Address:
192 . 168 . 000 . 254
Host Name:
Status: Offline
OK
Cancel
Apply
Help

RSLogix 5000 Configuration (continued)

Configure Bridge Module (As Needed)

The first step is to create a new project or open an existing one. The 1756-L1 processor is used in the screen images below.

1. Insert a bridge module into the I/O Configuration tree. As shown in figure 3.2 on the right, right click on the I/O Configuration folder and select "New Module..." in the pop-up menu.
2. In the Select Module Type windows that opens, select the proper Ethernet Bridge module. (In this example, the 1756-ENET/B.) Click on the [OK] button.
3. Enter the following information in the Module Properties window that opens. All parameters not listed here are optional Figure 3.3 shows a completed screen.

- Name: A descriptive name for the Bridge Module.
- IP Address: Must be the address you want for the module, not the address you set for the RM-89.
- Slot: The slot the module will reside in.

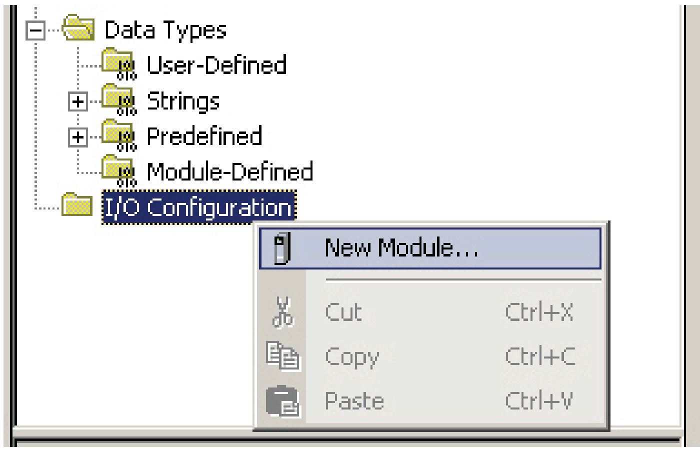


Figure 3.3 Defining A Bridge Module

4. When done, click on [Finish>>] to complete the setup of the Ethernet Bridge module.

RSLogix 5000 Configuration (continued)

Adding the RM-89 to Your Project

You can add the RM-89 to your project once the Ethernet port (Built-in or bridge module) is configured.

1.)As shown in figure 3.4 below, the Ethernet port will be listed under the I/O Configuration tree. Right click on the port and then click on "New Module..." in the pop-up menu.

2.)In the resulting window, scroll down the list until you find the entry that has a description of "Generic Ethernet Module". (Module Type is ETHERNET-MODULE in figure.) Click on the module name to select and then click the [OK] button. A Module Properties window will open.

3.)Set the following parameters in the Module Properties window. All parameters not listed here are optional. Figure 3.5 shows a completed screen.

- Name: A descriptive name for the RM-89.
- Comm Format: Data - INT (MUST be changed from the default Data - DINT.)
- IP Address: Must be the address you set for the RM-89. Refer to Chapter 3, RM-89 CONFIGURATION starting on page
- 37 for information on setting the IP Address of the RM-89.
- Input: You have four choices:

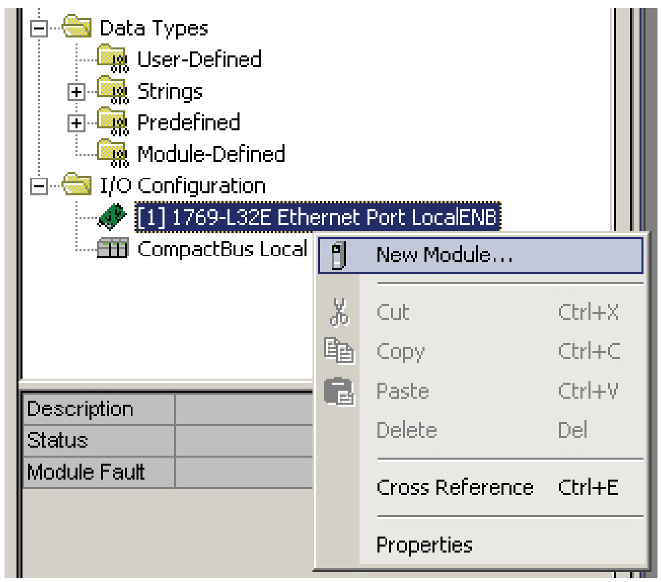


Figure 3.4

Assembly	Size	Data
1	2	Position Value
3	4	Position Value and Velocity Data
104	4	Position Value and Time Stamp
105	4	Position Value and Actual Sensor Reading

Table 3.1 Input Assembly Instances

RSLogix 5000 Configuration (continued)

Adding the RM-89 to Your Project (continued)
Output: Assembly Instance = 101, Size = 3
Configuration: You have two choices:

Assembly	Size	Data
102	8	Direction Counting Toggle, Scaling Function Control, Measuring Units per Span, Velocity Format
103	12	Direction Counting Toggle, Scaling Function Control, Measuring Units per Span, Total Measurement Range, Velocity Format

Table 3.2 Configuration Assembly Instances

NOTE:
The RM-89 must have valid data in the Configuration Registers before it will communicate with the network. This is true even if you have saved a valid configuration to the RM-89 with the TURCK Net Configurator software. You must put valid data into the Configuration Assembly Instance before the RM-89 will send data through the Input Assembly Instance. See Configuring the RM-89 on page 49.

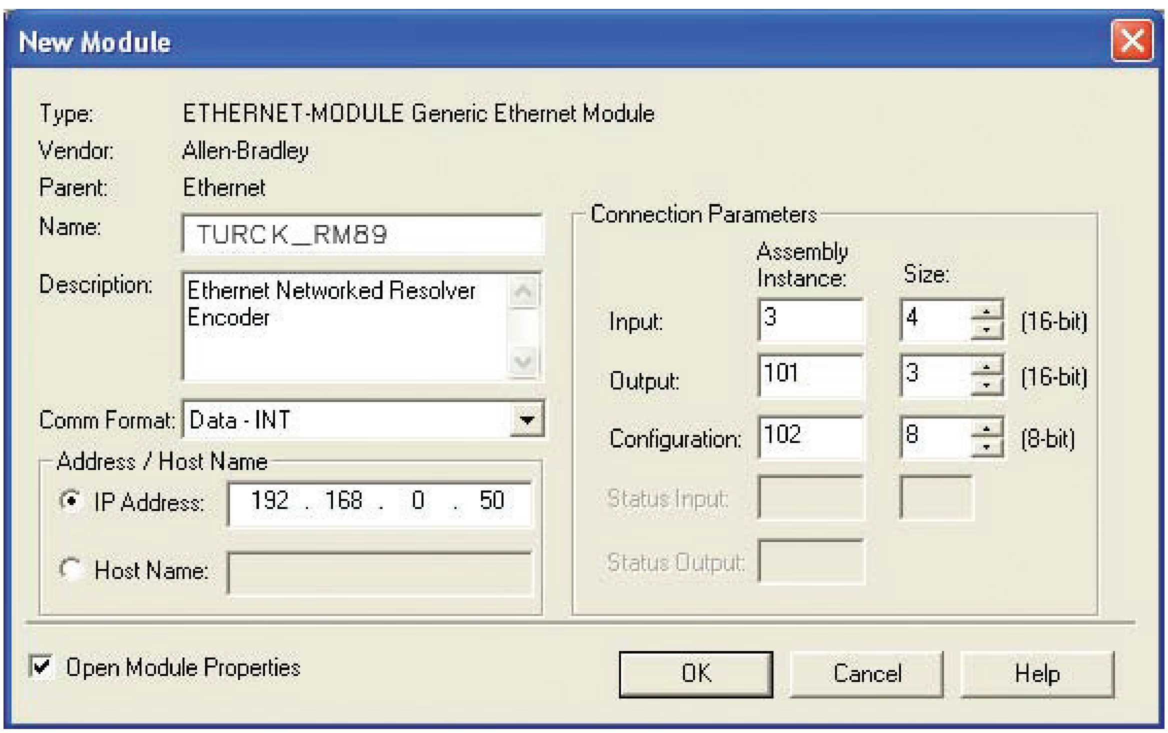


Figure 3.5 Sample RM-89 Configuration Screen

4. Click on [OK] to close the window.
5. Double click on the name you gave the RM-89 in the I/O Configuration tree. A "Module Properties" window will open Click on the "Connections" tab and set the RPI time that is required for your system. The minimum RPI time for an RM-89 is two milliseconds. The number of nodes on the network has an effect on the minimum RPI time. You may have to increase this RPI time if your network is heavily loaded. When done, click on [OK] to complete the setup.

Buffering Read Data

Ethernet Data is read asynchronously to the scan at the RPI time, therefore, you must buffer the data from the RM-89. Data must be buffered with Synchronous Copy instructions (CPS) to ensure that the data is not updated during the copy. If the data is not buffered, it can change during a program scan, resulting in logical errors that may result in machine malfunction.

Configuring the RM-89

The RM-89 Must be Configured

> **NOTE:**
> The RM-89 must have valid data in the Configuration Registers before it will communicate with the network. This is true even if you have saved a valid configuration to the RM-89 with the TURCK Net Configurator software. This requirement includes setting values for the Counts per Turn and Velocity Format parameters, even if they are not used in your application. The suggested value for the Counts per Turn parameter is 64,536 (0x0100) and Velocity Format is 0x1F04 (7,940). A non-configured RM-89 is the most common cause of technical support calls to TURCK for the RM-89 product family.

Assembly Instance = 102
The RM-89 is configured through the eight bytes in the Configuration Assembly Instance assigned to it when you added the encoder to your project. These bytes are accessed through the data tags assigned to the RM-89. Table 3.3 below shows the layout of the parameters programmed through the configuration bytes.

Byte #	Parameter	Description	
0	Direction Counting Toggle	"0" = Clockwise increasing counts looking at shaft. "1" = Counter-Clockwise increasing counts looking at shaft.	
1	Scaling Function Control	"0" = Disable Scaling Function. The full resolution of 65,536 counts per turn is used for the Measuring Units per Span. "1" = Enable Scaling Function. The number of counts per turn is set by the Measuring Units of Span parameter below.	
2	Counts per Turn	Sets the number of counts generated over a single turn if the Scaling Function Control parameter equals "1". This value requires four bytes and ranges from 1 to 65,536. A value of 39,370 (16#99CA) is shown to the right.	CA
3			99
4			00
5			00
6	Velocity Format	Format of the velocity data. Byte 7 must always equal "1F". Byte 6 = "04" for pulses/second, "05" for pulses/millisecond, "07" for pulses/minute or "0F" for revolutions/minute. A value of "1F04" to the right would set the unit of measure to pulses/second.	04
7			1F

Table 3.3 Configuration Bits

More information on these configuration parameters can be found in chapter 1, starting with the section Programmable Parameters, starting on page 13.

> **NOTE:**
> A valid Counts per Turn value must be entered even if you have disabled the Counts per Turn value by setting the Scaling Function Control set to zero. The default value of 65,536 is a suggested value. (Bytes 2, 3, 5 = 0. Byte 4 = 1.)

Configuring the RM-89 (continued)

Assembly Instance = 103

The RM-89 is configured through the twelve bytes in the Configuration Assembly Instance assigned to it when you added the encoder to your project. These bytes are accessed through the data tags assigned to the RM-89. Table 3.4 below shows the layout of the parameters programmed through the configuration bytes.

Byte #	Word #	Parameter	Description	
0	0.0	Direction Counting Toggle	"0" = Clockwise increasing counts looking at shaft. "1" = Counter-Clockwise increasing counts looking at shaft.	
1	0.8	Scaling Function Control	"0" = Disable Scaling Function. The full resolution of 65,536 counts per turn is used for the Measuring Units per Span. "1" = Enable Scaling Function. The number of counts per turn is set by the Measuring Units of Span parameter below.	
2	1	Measuring Units per Span (Counts per Turn)	Sets the number of counts generated over a single turn if the Scaling Function Control parameter equals "1". This value requires four bytes and ranges from 1 to 65,536. A value of 39,370 (16#99CA) is shown to the right.	CA
3				99
4	2			00
5				00
6	3	Total Measurement Range	Sets the number of counts before returning to zero. This value is used regardless of the state of the Scaling Function Control parameter. Parameter ranges: • Single Turn RM-89: Range of 0, 2 to 65,536 • 28 bit Multi-turn RM-89: Range of 0, 2 to 268,435,455 • 30 bit Multi-turn RM-89: Range of 0, 2 to 1,073,741,823 A value of 648,000 (16#0009 E340) is shown to the right.	40
7				E3
8	4			09
9				00
10	5	Velocity Format	Format of the velocity data. Byte 11 must always equal "1F". Byte 10 = "04" for pulses/second, "05" for pulses/millisecond, "07" for pulses/minute or "0F" for revolutions/minute. A value of "1F04" to the right would set the unit of measure to pulses/second.	04
11				1F

Table 3.4 Configuration Bits

More information on these configuration parameters can be found in chapter 1, starting with the section Programmable Parameters, starting on page 13.

NOTE:

A valid Counts per Turn value must be entered even if you have disabled the Counts per Turn value by setting the Scaling Function Control set to zero. The default value of 65,536 is a suggested value. (Bytes 2, 3, 5 = 0. Byte 4 = 1.)

Configuring the RM-89 (continued)

Module Fault Code 16#0110

If any of the parameter values are incorrect or missing, the RM-89 will respond with a Module Fault Code of 16#0110, Connection Request Error. As shown in the figure to the right, this error can be viewed under the Connections tab of the Module Properties window.

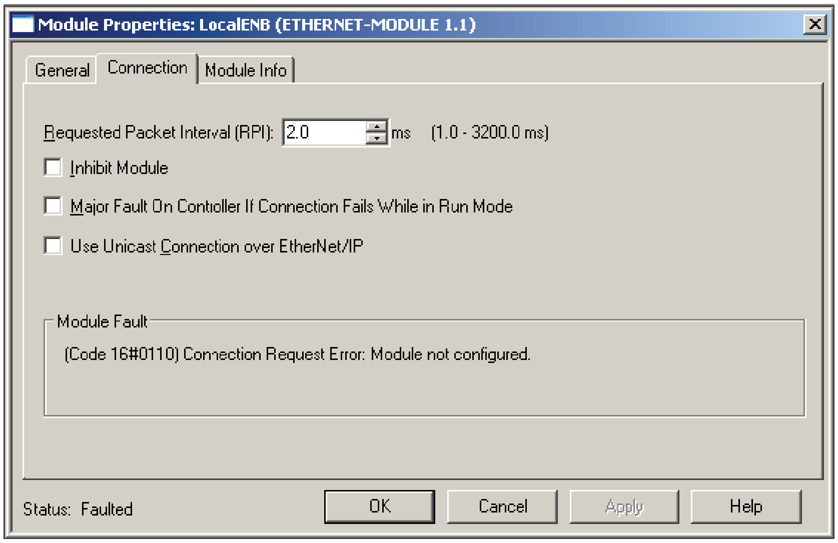


Figure 3.6 Configuration ...?

> **NOTE:**
> The RM-89 will not communicate with the network if the configuration data is incorrect. The only error indication that you will receive is the 16#0110 Fault Code.

Reading Data from the RM-89

All RM-89 encoders offer the following Input Assembly Instances:

Assembly	Size	Data
1	2	32 bit Position Value
3	4	32 bit Position Value and 32 bit Velocity Data

Table 3.5 Input Assembly Instances

RM-89 encoders that are revision 2.3 and above offer the following additional Input Assembly Instances:

Assembly	Size	Data
104	4	32 bit Position Value and 32 bit Time Stamp
105	4	32 bit Position Value and 32 bit Actual Sensor Reading

Table 3.6 Additional Input Assembly Instances

> **NOTE:**
> If you plan to preset the position value and store the resulting internal position offset, then TURCK strongly suggests using Assembly Instances 3, 104, or 105. The RM-89 uses the second data register to notify the processor when the data has been stored in the non-volatile memory of the RM-89. See Storing Configuration Data and the Internal Position Offset on page 54 for more information.

Reading Data from the RM-89 (continued)

Assembly Instance = 1

As shown in the table below, when you set the Input Assembly Instance to 1, the input data consists of the position value transferred as two 16 bit words.

Word #	Description	
0	Position Value. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013

Table 3.7 Input Data, Position Only

Assembly Instance = 3

As shown in the table below, when you set the Input Assembly Instance to 3, the input data consists of the position value and velocity data transferred in a total of four 16 bit words.

Word #	Description	
0	Position Data. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013
2	Velocity Data. The units of measure of the velocity data is set by the Velocity Format parameter in the Configuration Data. If the Scaling Function Control bit is ever set to a "1", the position data used to calculate the velocity data is always scaled by the Measuring Units per Span parameter. Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the velocity value is 461,725 (16# 0007 0B9D)	16#0B9D
3		16#0007

Table 3.8 Input Data, Position and Velocity

Assembly Instance = 104

As shown in the table below, when you set the Input Assembly Instance to 104, the input data consists of the position value and time stamp transferred in a total of four 16 bit words.

Word #	Description	
0	Position Data. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013
2	Time Stamp. The time stamp is an unsigned double integer value with an interval of 400 nanoseconds. It will roll over every 1717.9869184 seconds. The time stamp can be used to verify active communications between the RM-89 and your host controller. The values on the right show the register values in hexadecimal if the time stamp is 204,813,002 (16# 0C35 32CA)	16#32CA
3		16#0C35

Table 3.9 Input Data, Position and Time Stamp

Assembly Instance = 105

As shown in the table below, when you set the Input Assembly Instance to 105, the input data consists of the position value and the actual sensor position reading transferred in a total of four 16 bit words.

Word #	Description	
0	Position Data. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	
1		16#0013
2	Actual Sensor Reading. This unsigned double integer value is the raw position data from the RM-89. Changing the position scaling parameters have no effect on this value. The values on the right show the register values in hexadecimal if the Actual Sensor Reading value is 637,091,550 (16# 25F9 3EDE)	
3		16#25F9

Table 3.10 Input Data, Position and Actual Sensor Reading

Writing Data to the RM-89

The three output words assigned to the RM-89 are used to preset the position value and to save the resulting offset into non-volatile memory. Configuration data is also saved. The format of the output assembly words is shown below.

Word #	Description	
0	Command Word. Transitions on bits in this word will either preset the position value or store the resulting position offset in non-volatile memory.	
1	Preset Value. The value that you want the position to become when you issue the Preset Command. The Preset Value can be any number between 0 and the configured full scale count of the encoder. The values on the right show the register values in hexadecimal if the Preset Value is 704,303 (16# 000A BF2F)	16#BF2F
2		16#000A

Table 3.11 Output Assembly Instance Data Format

Presetting the Position Value

The position value is preset when the RM-89 detects the proper bit transitions in the Command Word. You begin the process by writing the desired Preset Value into words 1 and 2 and setting the Command Word to 16#0002. (2 decimal, the last four bits in binary are 0010). You must then hold these values for a length of time greater than the RPI time you programmed for the RM-89 when you added it to your project. This is to guarantee that these values have been written to the RM-89. After the RPI time has elapsed, change the Command Word value to 16#000D. (13 in decimal, the last four bits in binary are 1101.) The RM-89 will respond by changing the position value to the Preset Value by calculating and applying an internal position offset.

NOTE:

The internal position offset is stored in volatile RAM memory and is lost when power is cycled to the RM-89. This is acceptable in some applications because the machine has to be aligned on every power up. If you want to preset the position value once and have it apply the internal position offset on every power up, then you must command the RM-89 to store the internal offset in non-volatile memory. RM-89 encoders will automatically store the internal position offset to non-volatile memory if the value of the Total Measurement Range parameter is non-zero.

Storing Configuration Data and the Internal Position Offset

The data sent to the RM-89 through the configuration tags, as well as the internal position offset, is stored when the RM-89 detects the proper bit transitions in the Command Word. The parameters that are set with the configuration data are listed in the Configuring the RM-89 section of this chapter, starting on page 49.

You begin the process by setting the Command Word to 16#0020. (32 decimal, the last eight bits in binary are 0010 0000). You must then hold this value for a length of time greater than the RPI time you programmed for the RM-89 when you added it to your project. This guarantees that the value is written to the RM-89.

After the RPI time has elapsed, change the Command Word value to 16#00D0. (208 in decimal, the last eight bits in binary are 1101 0000.)

The RM-89 will store the values. If you are using Assembly Instances 3, 104, or 105 the RM-89 responds by changing the Velocity, Time Stamp, or Actual Sensor Reading to a value of 16#EEEE EEEE on a successful write or a value of 16#AAAA AAAA on an error. The data in the second 32 bit word will remain at this value until power to the RM-89 is cycled.

NOTE:

Once the command to store the internal position offset is accepted, the RM-89 will not respond to any further commands, including another save command, until power is cycled to the unit. This is to prevent damage to the non-volatile memory of the RM-89 by attempting to write to it too many times.

If you are using Assembly Instance 1 for your input tags, then you are only receiving the position data and the RM-89 will not be able to indicate that the store command was completed successfully. Because of this, TURCK strongly suggests that you use Assembly Instances 3, 104, or 105 for your input tags if your application requires you to store the configuration data or internal position offset.

The RM-89 encoder will automatically store the internal position offset in non-volatile memory if the Total Measurement Range parameter is non-zero.

CHAPTER 4
RSLogix 500 EtherNet/IP CONFIGURATION

Rockwell Automation Ethernet products use the EtherNet/IP protocol. This chapter shows you how to configure communications between an RM-89 and your PLC using RSLogix 500 software. A MicroLogix 1400 will be used as an example. This chapter also shows you how to use the registers you create for the RM-89 to read and write data to the encoder.

Explicit Messaging
Every PLC that is programmed using RSLogix 500 software, such as the MicroLogix platform, must use explicit messaging to communicate with the RM-89. In explicit messaging, the PLC processor will only communicate with the RM-89 when explicitly told to through Message Instructions that are programmed into your ladder logic.
The other form of communication is implicit messaging. Implicit messaging is only supported by Allen Bradley PLC platforms that are programmed using the RSLogix 5000 software. Implicit messaging means that the PLC processor will automatically exchange data with the RM-89 at the programmed RPI time, thereby simplifying the use of the RM-89. Implicit messaging is explained in the previous chapter, RSLogix 5000 EtherNet/IP CONFIGURATION, which started on page 45.

NOTE:
You can use explicit messaging on platforms that are programmed using the RSLogix 5000 software. The configuration and use of Message Instructions given here are also applicable to the RSLogix 5000 software.

RSLogix 500 Configuration
When using the MicroLogix platforms, you have to configure the Ethernet port that is built into some processors before adding the RM-89 to your project.

NOTE:
Only RSLogix 500 version 8.0 or above can be used to configure Message Instructions to communicate with an Ethernet IP device.

Configuring a Built-in Ethernet Port
You have to set an IP address for the built in Ethernet port before communicating with the RM-89. Right click on the port name in the I/O Configuration screen and select "Properties". A Module Properties window similar to the one shown in figure 4.1 will open. In this window you must set the IP Address for the port, not the IP address of the RM-89.

Read Data Format
The Assembly Instance of a Message Instruction that is used to read data from an RM-89 defines the data that is transferred. All RM-89 encoders respond to the following Input Assembly Instances:

Assembly	Size	Data
1	2	32 bit Position Value
3	4	32 bit Position Value and 32 bit Velocity Data

Table 4.1 Input Assembly Instances

Read Data Format (continued)

RM-89 encoders respond to the following additional Input Assembly Instances:

Assembly	Size	Data
104	2	32 bit Position Value and 32 bit Time Stamp
105	4	32 bit Position Value and 32 bit Actual Sensor Reading

Table 4.2 Additional Input Assembly Instances

Assembly Instance = 1
When you use an Assembly Instance to 1, the input data consists of the position value transferred as two 16 bit words.

Word #	Description	
0	**Position Value.** The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013

Table 4.3 Input Data, Position Only

Assembly Instance = 3
When you use an Assembly Instance to 3, the input data consists of the position value and velocity data transferred in a total of four 16 bit words.

Word #	Description	
0	Position Data. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013
2	Velocity Data. The units of measure of the velocity data is set by the Velocity Format parameter in the Configuration Data. If the Scaling Function Control bit is ever set to a "1", the position data used to calculate the velocity data is always scaled by the Measuring Units per Span parameter. Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the velocity value is 461,725 (16# 0007 0B9D)	16#0B9D
3		16#0007

Table 4.4 Input Data, Position and Velocity

Read Data Format (continued)

Assembly Instance = 104
When you use an Assembly Instance to 104, the input data consists of the position value and time stamp transferred in a total of four 16 bit words.

Word #	Description	
0	Position Data. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013
2	Time Stamp. The time stamp is an unsigned double integer value with an interval of 400 nanoseconds. It will roll over every 1717.9869184 seconds. The time stamp can be used to verify active communications between the RM-89 and your host controller. The values on the right show the register values in hexadecimal if the time stamp is 204,813,002 (16# 0C35 32CA)	16#32CA
3		16#0C35

Table 4.5 Input Data, Position and Time Stamp

Assembly Instance = 105
When you use an Assembly Instance to 105, the input data consists of the position value and the actual sensor position reading transferred in a total of four 16 bit words.

Word #	Description	
0	Position Data. The maximum position value depends on your RM-89 model and the programmed counts per turn. The maximum value in all cases is 1,073,741,823 (16#3FFF FFFF). Note that the two 16 bit registers are combined into a single 32 bit data word. The values on the right show the register values in hexadecimal if the position value is 1,274,237 (16# 0013 717D)	16#717D
1		16#0013
2	Actual Sensor Reading. This unsigned double integer value is the raw position data from the RM-89. Changing the position scaling parameters have no effect on this value. The values on the right show the register values in hexadecimal if the Actual Sensor Reading value is 637,091,550 (16# 25F9 3EDE)	16#3EDE
3		16#25F9

Table 4.6 Input Data, Position and Actual Sensor Reading

Write Data Format

The Assembly Instance of a Message Instruction that is used to write configuration data to an RM-89 defines the data that is transferred. All RM-89 encoders respond to the following Input Assembly Instances:

As-sembly	Size	Data
102	8	Direction Counting Toggle, Scaling Function Control, Counts per Turn, Velocity Format

Table 4.7 Write Assembly Instances

Write Data Format (continued)

RM-89 encoders respond to the following additional Input Assembly Instances:

Assembly	Size	Data
103	12	Direction Counting Toggle, Scaling Function Control, Counts per Turn, Total Measurement Range, Velocity Format

Table 4.8 Additional Write Assembly Instances

Assembly Instance = 102
Table 4.9 below shows the layout of the programmed parameters when you use an Assembly Instance of 102. For the RM-89 encoder, the Total Measurement Range parameter stays at its last value.

Byte#	Word #	Parameter	Description	
0	0.0	Direction Counting Toggle	"0" = Clockwise increasing counts looking at shaft. "1" = Counter-Clockwise increasing counts looking at shaft.	
1	0.8	Scaling Function Control	"0" = Disable Scaling Function. The full resolution of 65,536 counts per turn is used for the Measuring Units per Span. "1" = Enable Scaling Function. The number of counts per turn is set by the Measuring Units of Span parameter below.	
2	1	Measuring Units per Span (Counts per Turn)	Sets the number of counts generated over a single turn if the Scaling Function Control parameter equals "1". This value requires four bytes and ranges from 1 to 65,536. A value of 39,370 (16#99CA) is shown to the right.	CA
3				99
4	2			00
5				00
6	3	Velocity Format	Format of the velocity data. Byte 7 must always equal "1F". Byte 6 = "04" for pulses/second, "05" for pulses/millisecond, "07" for pulses/minute or "0F" for revolutions/minute. A value of "1F04" to the right would set the unit of measure to pulses/second.	04
7				1F

Table 4.9 Configuration Data: Assembly Instance 102

More information on these configuration parameters can be found in chapter 1, starting with the section Programmable Parameters, starting on page 13.

NOTE:
A valid Counts per Turn value must be entered even if you have disabled the Counts per Turn value by setting the Scaling Function Control set to zero. The default value of 65,536 is a suggested value. (Bytes 2, 3, 5 = 0. Byte 4 = 1.)

Write Data Format (continued)

Assembly Instance = 103

Table 4.10 below shows the layout of the programmed parameters when you use an Assembly Instance of 103.

Byte #	Word #	Parameter	Description	
0	0.0	Direction Counting Toggle	"0" = Clockwise increasing counts looking at shaft. "1" = Counter-Clockwise increasing counts looking at shaft.	
1	0.8	Scaling Function Control	"0" = Disable Scaling Function. The full resolution of 65,536 counts per turn is used for the Measuring Units per Span. "1" = Enable Scaling Function. The number of counts per turn is set by the Measuring Units of Span parameter below.	
2	1	Measuring Units per Span (Counts per Turn)	Sets the number of counts generated over a single turn if the Scaling Function Control parameter equals "1". This value requires four bytes and ranges from 1 to 65,536. A value of 39,370 (16#99CA) is shown to the right.	CA
3				99
4	2			00
5				00
6	3	Total Measurement Range	Sets the number of counts before returning to zero. This value is used regardless of the state of the Scaling Function Control parameter. Parameter ranges: Single Turn RM-89: Range of 0, 2 to 65,536 28 bit Multi-turn RM-89: Range of 0, 2 to 268,435,455 30 bit Multi-turn RM-89: Range of 0, 2 to 1,073,741,823 A value of 648,000 (16#0009 E340) is shown to the right.	40
7				E3
8	4			09
9				00
10	5	Velocity Format	Format of the velocity data. Byte 11 must always equal "1F". Byte 10 = "04" for pulses/second, "05" for pulses/millisecond, "07" for pulses/minute or "0F" for revolutions/minute. A value of "1F04" to the right would set the unit of measure to pulses/second.	04
11				1F

Table 4.10 Configuration Data: Assembly Instance 103

More information on these configuration parameters can be found in chapter 1, starting with the section Programmable Parameters, starting on page 13.

NOTE:

A valid Counts per Turn value must be entered even if you have disabled the Counts per Turn value by setting the Scaling Function Control set to zero. The default value of 65,536 is a suggested value. (Bytes 2, 3, 5 = 0. Byte 4 = 1.)

Presetting the Position Value

A custom Message Instruction with the Generic Set_Attribute_Single service type is used to preset the position value of the RM-89. The format of the data for this Message Instruction is shown below.

Byte #	Word #	Description	
0	1	Preset Value. The value that you want the position to become when you issue this command. The Preset Value can be any number between 0 and the configured full scale count of the encoder. The values on the right show the register values in hexadecimal if the Preset Value is 704,303 (16# 000A BF2F)	2F
1			BF
2	2		0A
3			00

Table 4.11 Preset Position Data Format

NOTE:

- The internal position offset is stored in volatile RAM memory and is lost when power is cycled to the RM-89
- This is acceptable in some applications because the machine has to be aligned on every power up. If you
- want to preset the position value once and have it apply the internal position offset on every power up, then you
- must issue a Message Instruction with the Save command. (See the Saving Configuration Data and Position
- Offset section below.) This command stores the configuration data and the internal position offset in non-volatile memory.
- The FRAM memory in the RM-89 does not have a limited number of write cycles.

RM-89 encoders will automatically store the internal position offset to non-volatile memory if the value of the Total Measurement Range parameter is non-zero. You do not have to issue a Save command to store the internal position offset if the Total Measurement Range parameter is non-zero.

Saving Configuration Data and Position Offset

The internal position offset, as well as the configuration data, is stored in non-volatile memory when the RM-89 detects a Save command. The Message Instruction that sends the Save command is a custom instruction that has no data. (The Class and Service Code of the Message Instruction are used by the RM-89 to determine that the instruction is a Save command.).
Once the command is issued, the data is stored when the Done bit of the Message Instruction is set. You must cycle power to the RM-89 before another Save command will be accepted.

NOTE:
The FRAM memory in the RM-89 does not have a limit on the number of permitted write cycles.

Determining Needed Registers

Before adding the logic needed to communicate with the RM-89 and use the data from it, you have to assign registers that will be used to hold the data. The top of the following table lists the types and sizes of all of the data that can be read from or written to the RM-89.

The center section of the table lists the additional registers needed to buffer the data from the Message Instructions that read data from the RM-89. These registers are required because the Message Instructions complete asynchronously to the program scan.

WARNING:
If data read with Message Instructions is not buffered, this data can change during a program scan, resulting in logical errors that may result in machine malfunction.

The bottom section of the table lists the Message (MG) data file and Extended Routing Information (RIX) data file types needed to control the Message Instructions. Each RM-89 requires a separate file of each type, and these files must have one element for each Message Instruction associated with the RM-89.

Value	Type	Size	Availability
Position	Integer	2 Words	
Position and Velocity	Integer	4 Words	
Position and Time Stamp	Integer	4 Words	
Position and Actual Sensor Reading	Integer	4 Words	
Direction Counting Toggle	Upper byte of Integer	1/2 Word	
Scaling Function Control	Lower byte of Integer	1/2 Word	
Measuring Units per Span	Integer	2 Words	
Total Measurement Range	Integer	2 Words	
Velocity Format	Integer	2 Words	
Preset Value	Integer	2 Words	
Buffered Position	Integer	2 Words	
Buffered Velocity	Integer	2 Words	
Buffered Time Stamp	Integer	2 Words	
Buffered			
Actual Sensor Reading	Integer	2 Words	
Message Element	Message (MG)	–	Each RM-89 requires a separate file, and each Message Instruction requires a separate element in that file.
RIX Element	(RIX)	–	Each RM-89 requires a separate file, and each Message Instruction requires a separate element in that file.

Table 4.12 Suggested Register Allocation

Configuring a Message Instruction

Figure 4.2 shows Message Instructions as they appear in your ladder logic.

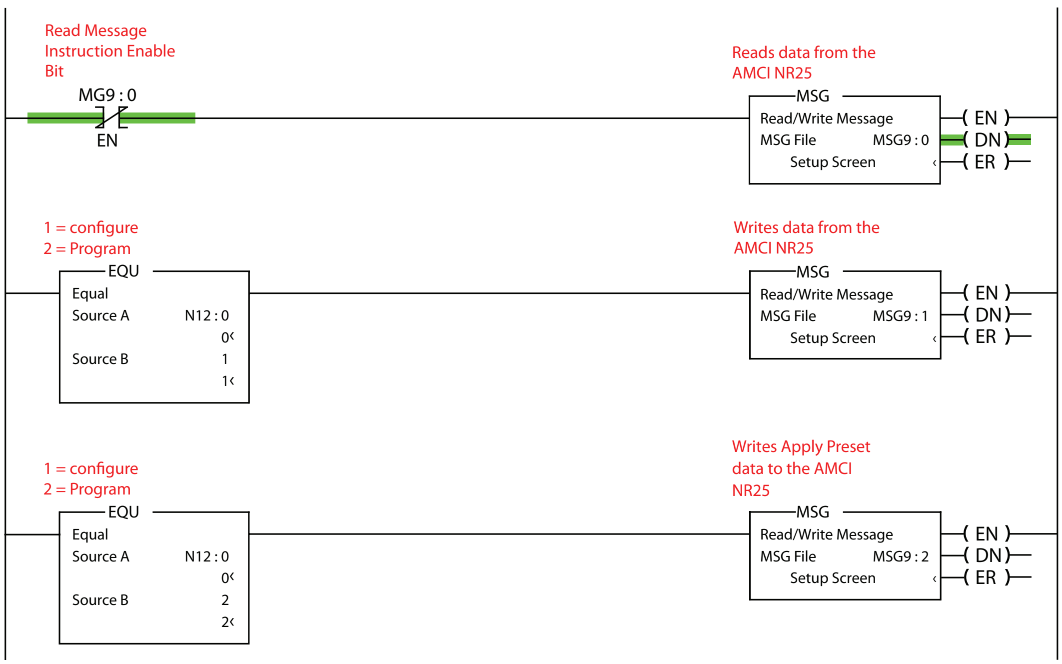


Figure 4.2 Message Instruction Example

1. Start to configure the Message Instruction by double clicking on the Setup Screen text that is inside the Message Instruction. The window in figure 4.3 will open. Note that this is the default window and its appearance will change considerably as you progress through these steps.

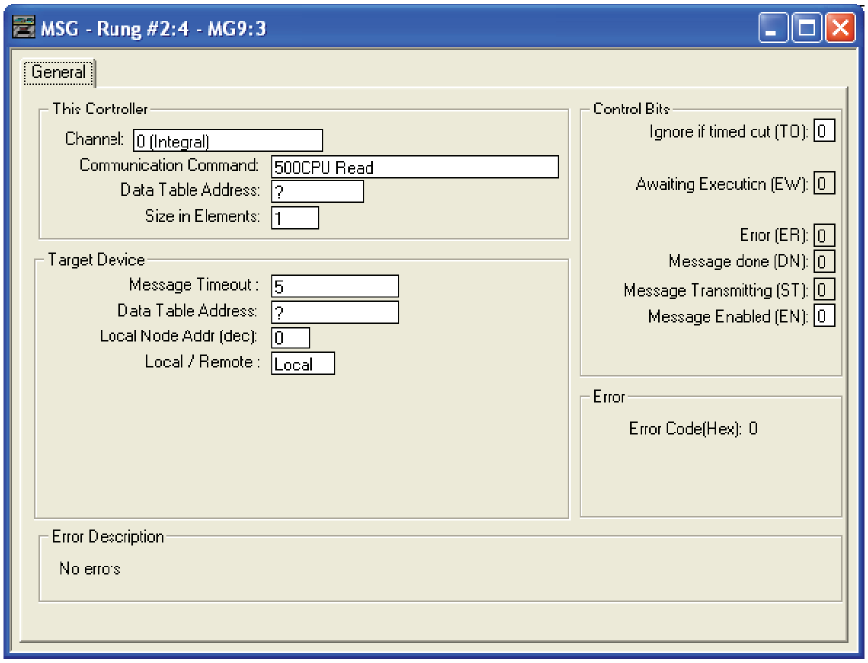


Figure 4.3 Default Message Instruction Setup Screen

2. Double click in the Channel field, click on the , select "1 (Integral)", and press Enter.
3. Double click in the Communication Command field, click on the , select "CIP Generic" and press Enter.

NOTE:
The rest of this section is broken down into four parts. Continue with the sub-section that is specific to the Message Instruction you are programming.

- Read Message Instructions: See below
- Write Configuration Message Instructions: Starts on page 64
- Apply Preset Message Instruction: Starts on page 66.
- Save Configuration and Offset Message Instruction: Starts on page 67.

Read Message Instructions

As shown in the table below, the RM-89 will respond to four different Read Message Instructions.

	Position Value Only	Position Value and Velocity Data	Position Value and Time Stamp	Position Value and Actual Sensor Reading
Size in Bytes	4 bytes	8 bytes	8 bytes	8 bytes
Instance	1 (decimal)	3 (decimal)	104 (decimal)	105 (decimal)

Table 4.13 Attributes: Explicit Read Message Instructions

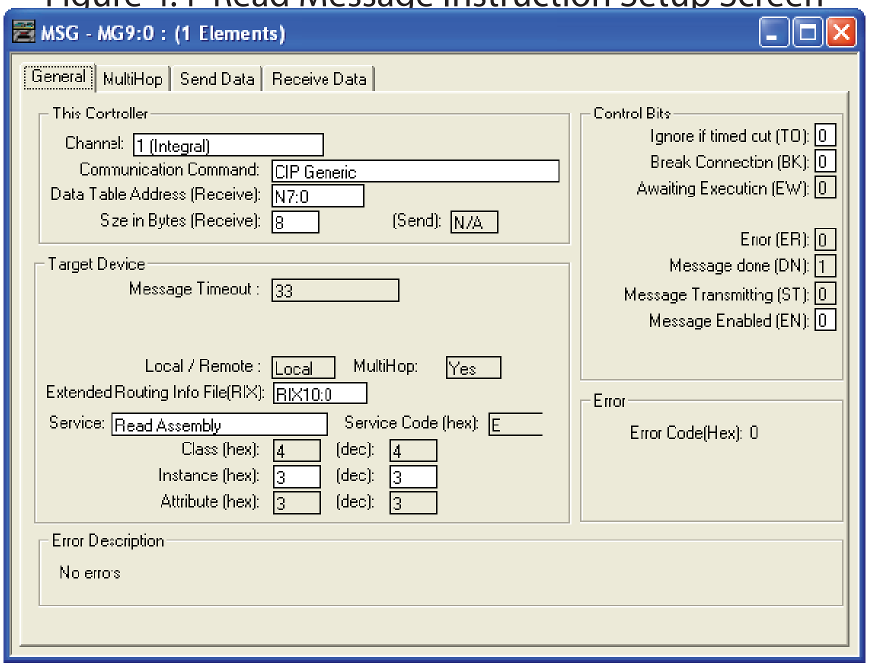
1. Enter the integer file address where the data will be placed in the Data Table Address (Received) field and press Enter.
2. Enter the correct value in the 'Size in Bytes (Receive)' field. The size is determined by the data you wish to transfer from the RM-89. Refer to table 4.13 above to determine the correct value for this field.
3. Enter a RIX address in the Extended Routing Info field. Please note that each Message Instruction must have its own RIX address.
4. Double click in the Service field, select "Read Assembly" for a Message Instruction type and press Enter.
5. The Service Code field will change to "E" (hex), the Class field will change to "4" (hex), and the Attribute field will change to "3" (hex).

6. Enter the correct value in the 'Instance' field. This value determines the data that will be transferred by the RM-89. Refer to table 4.13 above to determine the correct value for this field.

Configuring a Message Instruction (continued)

Read Message Instructions
The figure below shows a typical configuration for a Message Instruction that reads data from an RM-89. Please note that the Data Table Address (Receive), Size in Bytes (Receive), and RIX fields may be different in your application.

Figure 4.4 Read Message Instruction Setup Screen



7. Jump to the section, Setting the MultiHop Address, which is on page 68, to finish configuring the Message Instruction.

Write Configuration Message Instructions
Write Message Instructions are used to write programmable parameters to the RM-89. As shown in the table below, the RM-89 will respond to two different Write Message Instructions.

	Direction Counting Toggle Scaling Function Control Measuring Units per Span Velocity Format	Direction Counting Toggle Scaling Function Control Measuring Units per Span Total Measurement Range Velocity Format
Length	8 bytes	12 bytes
Instance	102 (decimal)	103 (decimal)

Table 4.14 Attributes: Explicit Write Message Instructions

1. Enter the integer file address where the source data is located into the 'Data Table Address (Send)' field and press Enter.
2. Enter the correct value in the 'Size in Bytes (Send)' field. The size is determined by the data you wish to transfer to the RM-89.
3. Enter a RIX address in the Extended Routing Info field. Please note that each Message Instruction must have its own RIX address.

Write Configuration Message Instructions (continued)

4. Double click in the Service field, select "Write Assemble", and press Enter.
5. The Service Code field will change to "10" (hex). The Class field will change to "4" (hex), and the Attribute field will change to "3" (hex).
6. Enter the correct value in the 'Instance' field. This value determines the data that will be transferred to the RM-89. Refer totable 4.14 above to determine the correct value for this field.

The figure below show a typical Message Instruction for writing configuration data to an RM-89. Please note that the Data Table Address (Send) field and RIX field may be different in your application.

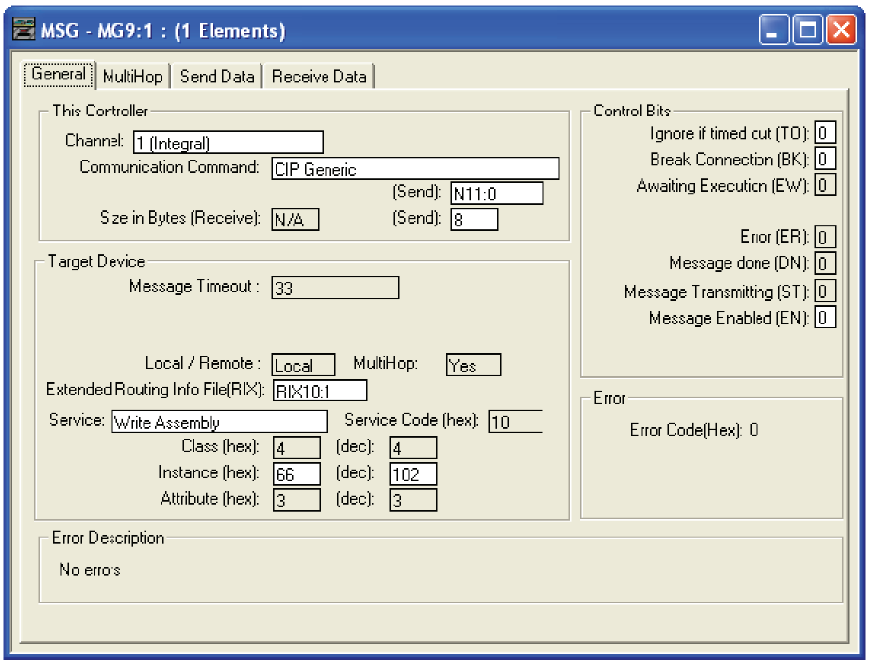


Figure 4.5 Configuration Message Instruction Setup Screen

7. Jump to the section, Setting the MultiHop Address, which is on page 68, to finish configuring the Message Instruction.

Apply Preset Message Instruction

A Generic Message Instruction is used to preset the position value of the RM-89.

1. Enter the integer file address where the desired position preset value is located into the 'Data Table Address (Send)' field and press Enter.
2. Enter a value of "4" in the Size In Bytes (Send) field.
3. Enter a RIX address in the Extended Routing Info field. Please note that each Message Instruction must have its own RIX address.
4. Double click in the Service field, select "Generic Set Attribute Single", and press Enter.
5. The Service Code field will change to "10" (hex).
6. Enter a value of 16#23 into the Class field.
7. Enter a value of 1 into the Instance field.
8. Enter a value of 16#13 into the Attribute field.

The figure below show a typical Message Instruction for issuing a Preset Command to an RM-89. Please note that the Data Table Address (Send) field and RIX field may be different in your application.

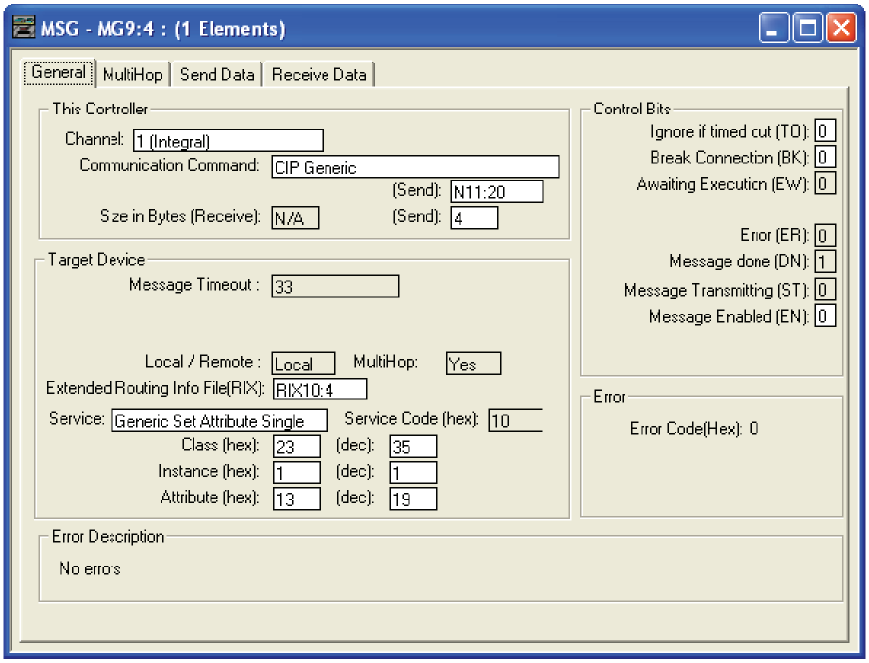


Figure 4.6 Preset Message Instruction Setup Screen

9. Jump to the section, Setting the MultiHop Address, which is on page 68, to finish configuring the Message Instruction.

Save Configuration and Offset Message Instruction

A Custom Message Instruction is used to save the present configuration of the RM-89. The instruction will also save the internal position offset that is modified by an Apply Preset Message Instruction.

NOTE:
Setting the Total Measurement Range parameter to a non-zero value will force the RM-89 to automatically store the internal position offset to non-volatile FRAM. Using the Total Measurement Range parameter this way only affects how the internal position offset is stored. You must still issue this Message Instruction to save the programmable parameters to non-volatile memory. See Storage of Internal Position Offset found on page 15 for information on how to set the Total Measurement Range parameter.

1. The Save Configuration and Offset Message Instruction does not write any data to the RM-89 so you can ignore the Data Table Address and Size in Byte parameters. Once you save this instruction and re-open the window, you will see that the address field has been removed and the length fields are set to zero.
2. Enter a RIX address in the Extended Routing Info field. Please note that each Message Instruction must have its own RIX address.
3. Double click in the Service field, select "Custom", and press Enter.
4. Change the Service Code field to 16#16.
5. Change the Class field to 16#23 (35 decimal).
6. Change the Instance field and Attribute field to zero.

The figure below show a typical Message Instruction for issuing a Save Command to an RM-89. Please note that the RIX field may be different in your application.

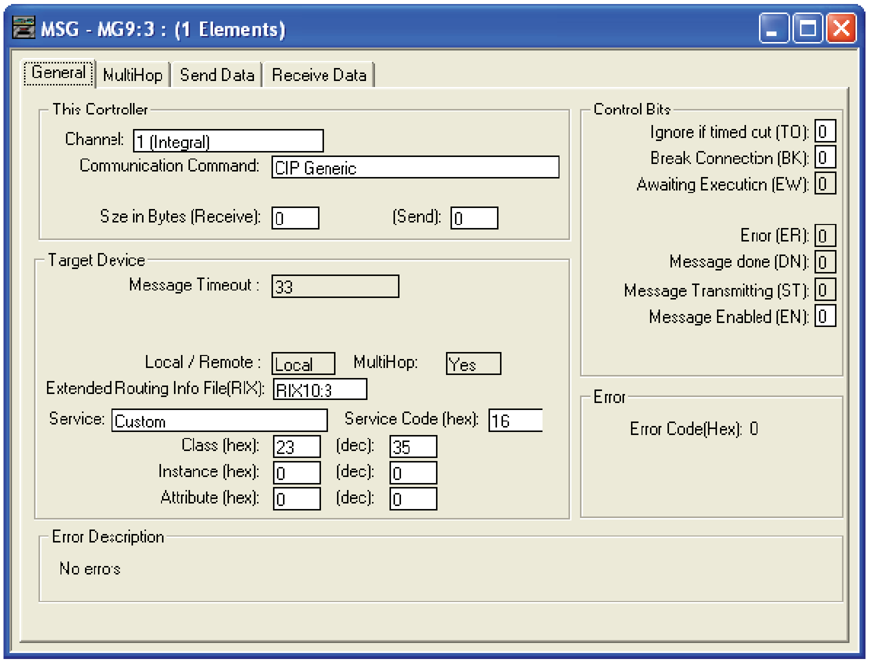


Figure 4.7 Save Message Instruction Setup Screen

7. Continue to the section, Setting the MultiHop Address, which is on the next page, to finish configuring the Message Instruction.

Setting the MultiHop Address

After setting the fields on the General tab, click on the MultiHop tab in the setup window. You must enter the IP address of the RM-89 in the "To Address" field. The default RM-89 address is show in the figure below.

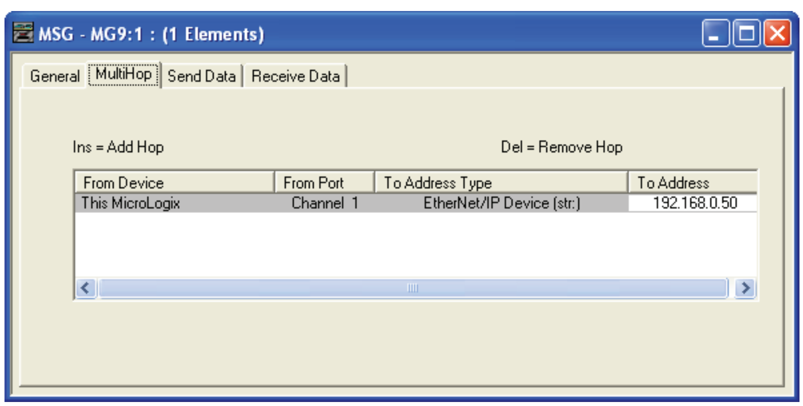


Figure 4.8 Message Instruction MultiHop Settings

Troubleshooting Message Instructions

These are the main causes of communication errors when using an RM-89:

- The IP address or netmask are not set correctly when the processor's Ethernet was configured.
- The Message Instruction Instance or Length parameters are not set correctly.
- The To Address field of the Message Instruction, found under the MultiHop tab, is not set to the address the RM-89 is
- configured for.
- The Configuration data is not formatted correctly. The proper format of the Configuration data is shown in the next section.

Configuration Error Response

If any of the parameter values are incorrect, the RM-89 will respond by setting the Message Instruction's Error bit. As shown in the figure below, the error can be viewed under the General tab of the Message Instruction Setup Window. The following message is in response to an invalid Velocity Format parameter.

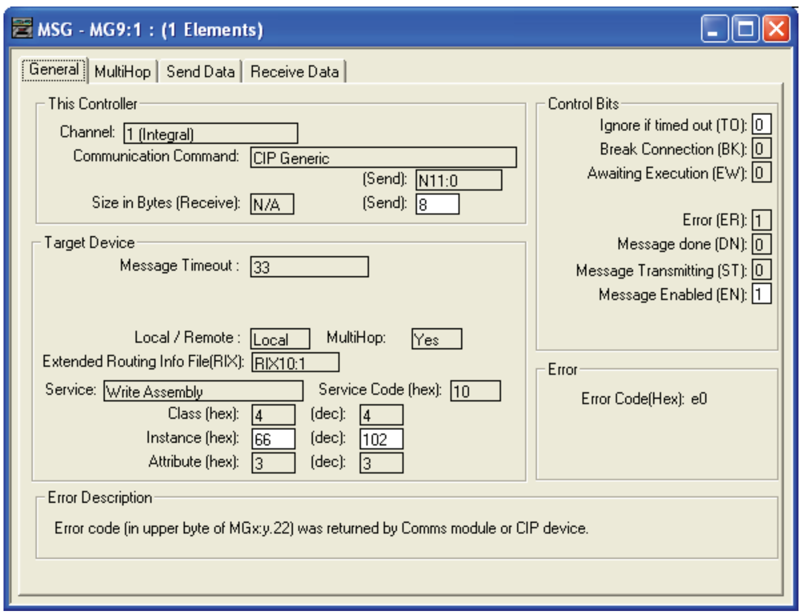


Figure 4.9 Configuration Error Response

CHAPTER 5
Modbus TCP CONFIGURATION

RM-89 Memory Layout

All RM-89 encoders, regardless of revision, use six 16-bit input registers and seven 16-bit output registers to communicate through the Modbus protocol. RM-89 encoders require eight 16-bit input registers and nine 16-bit output registers to access the time stamp value and to program the Total Measurement Range parameter. Figure 5.1 shows how these registers are mapped to the Modbus data reference. The complete specification for the Modbus protocol can be downloaded at http://www.modbus.org/specs.php.

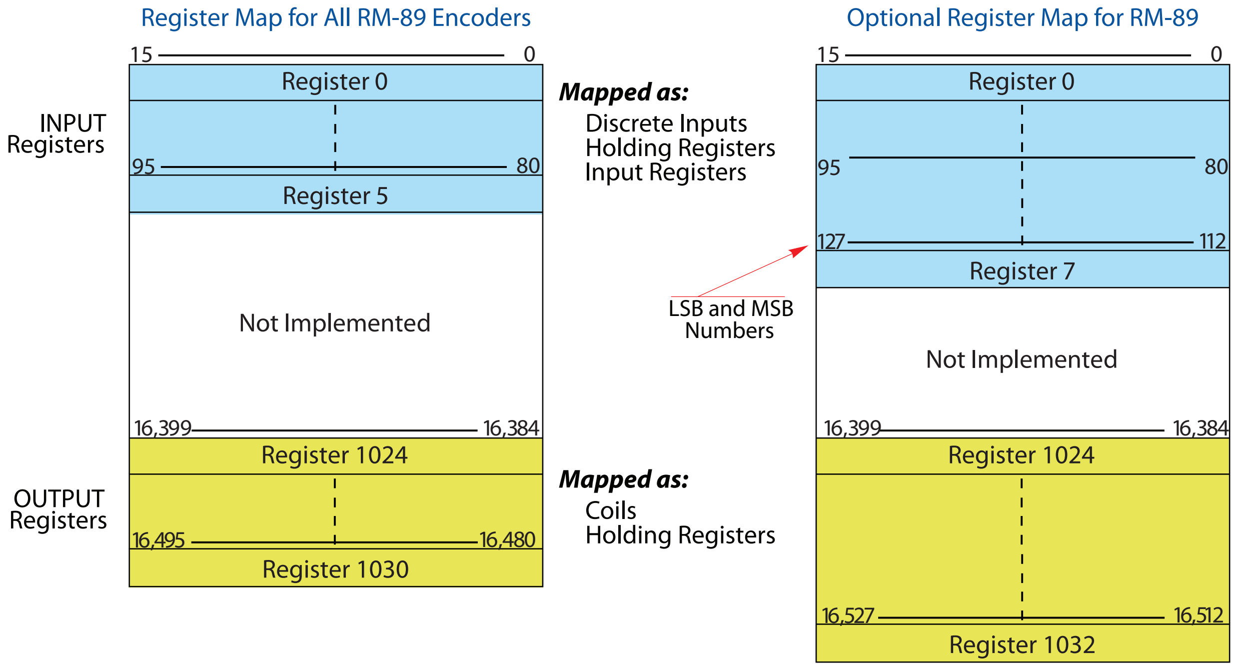


Figure 5.1 Modbus Data Reference Map

Supported Modbus Functions

Function Code	Function Name	NR25 Register	Addressing method
1	Read Coils	OUTPUT	Bit: Address 16,384 through 16,527
2	Read Discrete Inputs	INPUT	Bit: Address 0 through 127
3	Read Holding Registers	OUTPUT & INPUT	Word: Out Regs. 1024 through 1032 In Regs. 0 though 7
4	Read Input Registers	INPUT	Word: Registers 0 through 7
5	Write Single Coil	OUTPUT	Bit: Address 16,384 through 16,527
6	Write Single Register	OUTPUT	Word: Registers 1024 through 1032
15	Write Multiple Coils	OUTPUT	Bit: Address 16,384 through 16,527
16	Write Multiple Registers	OUTPUT	Word: Registers 1024 through 1032
22	Mask Write Register	OUTPUT	Word: Registers 1024 through 1032
23	Read/Write Registers	INPUT/OUTPUT	Word: Out Regs. 1024 through 1032 In Regs. 0 though 5

Table 5.1 Supported Modbus Functions

Supported Modbus Exceptions

Code	Name	Description
01	Illegal function	The RM-89 does not support the function code in the query
02	Illegal data address	The data address received in the query is outside the initialized memory area
03	Illegal data value	The data in the request is illegal

Table 5.2 Supported Modbus Exceptions

Multi-Word Format

The Modbus protocol uses 16 bit registers, which limits the range of values from -32,768 to 32,767 or 0 to 65,535. Many parameters and data values from the RM-89 exceed this range. These parameters are transmitted in two separate registers. The table below shows how values are split.

Value	First Register	Second Register
12 (0x0000 000C)	12 (0x000C)	0 (0x0000)
1,234,567 (0x0012 D687)	54,919 (0xD687)	18 (0x0012)

Table 5.3 Multi-Word Format Examples

Output Data Format

Table 5.4 shows the correct format for the Network Output Registers that are written to the RM-89.

Register	Description	Example
1024	**Command Word**	See Description below
1025	**Preset Value:** The value that you want the position to become when you issue this command. The Preset Value can be any number between 0 and the maximum count of the encoder. The values on the right show the register values in hexadecimal if the Preset Value is 704,303 (0x 000A BF2F)	0xBF2F
1026		0x000A
1027	**Configuration Word**	See Description Below
1028	**Counts per Turn:** Sets the number of counts generated over a single turn if the Scaling Function Control parameter equals "1". This value requires two registers and ranges from 2 to 65,536. A value of 39,370 (16#99CA) is shown to the right.)	0x99CA
1029		0x0000
1030	**Velocity Format:** Format of the velocity data. 0x1F04 for pulses/second, 0x1F05 for pulses/millisecond, 0x1F07 for pulses/minute or 0x1F0F for revolutions/minute. The value of "1F04" to the right would set the unit of measure to pulses/second.	0x1F04
1031	**Total Measurement Range:** Sets the number of counts before the position value returns to zero. If this parameter is left at its default value of zero, the roll over position is determined by the Measuring Units per Span parameter and the number of turns the RM-89 can encode. The Total Measurement Range can be any number between 0 and the maximum count of the encoder. The values on the right show the register values in hexadecimal if the Total Measurement Range is 8,073,859 (0x 007B 3283)	0x3283
1032		0x007B

Table 5.4 Output Registers Data Format

Command Word

15	14	13	12	11	10	09	08	07 06 05 04	03 02 01 00
0	0	0	0	0	0	0	0	**Store Parameters to Flash**	**Apply Preset Value**

RESERVED: Bit must equal zero.

Figure 5.2 Command Word Format

Apply Preset Value: These four bits control when the Preset Value in registers 1025 and 1026 is applied to the position value. In order for the position to be preset, the value in these four bits must transition from 0x2 (0b0010), to 0xD (0b1101). When these four bits make this transition, the RM-89 calculates the position offset needed to bring the position to the Preset Value.

NOTE:
The RM-89 will not respond to an error in the Preset Value. Specifically, a Modbus Exception Code 03 is not returned. The only response from the RM-89 is to ignore the value and not preset the position. After issuing a preset command, read back the position value and verify that the position has been preset correctly.

Store Parameter to Flash: These four bits control when the programmable parameters and the internal position offset are stored to non-volatile memory. These values are not automatically written to this memory whenever they are changed. In order to store these parameter values and internal position offset, the value in these four bits must transition from 0x2 (0b0010), to 0xD (0b1101). Note that these bits are in locations 04 - 07. The actual register values when issuing this command are 0x20 and 0xD0.

NOTE:
- The FRAM memory in the RM-89 does not have a limit on the number of permitted write cycles.

Setting the Total Measurement Range parameter to a non-zero value will force the RM-89 to automatically store the internal position offset to non-volatile FRAM. Using the Total Measurement Range parameter this way only affects how the internal position offset is stored. You must still issue this Message Instruction to save the programmable parameters to non-volatile memory. See Storage of Internal Position Offset found on page 15 for information on how to set the Total Measurement Range parameter.

Output Data Format (continued)

Configuration Word

15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
0	0	0	0	0	0	0	SF CTRL	0	0	0	0	0	0	0	DCTog

RESERVED: Bit must equal zero.

Figure 5.3 Configuration Word Format

DCTog: Direction Control Toggle bit. When this bit equals "0", the position value will increase with clockwise rotation when looking at the front of the shaft. When this bit equals "1" the position value will increase with counter-clockwise rotation when looking at the front of the shaft.

SFCtrl: Scaling Function Control bit. When this bit equals "0", the position resolution will be 65,536 counts per turn. When this bit equals "1", the position resolution will be set by the Measuring Units per Span parameter value contained in registers 1028 and 1029. As explained in the Calculating Position and Velocity Data section found on page 14, once the Measuring Units per Span parameter is applied, the velocity data will always be scaled by this parameter, regardless of the state of the Scaling Function Control bit. RM-89 encoders have the Total Measurement Range parameter. This parameter is not affected by the state of the Scaling Function Control bit.

Input Data Format

Table 5.5 shows the format of the data read from the RM-89.

Register	Data Value	Example
0	**32 bit Scaled Position Value.** This data is the calculated position data. Its value is affected by the Measuring Units per Span parameter if the Scaling Function Control bit equals "1". This value can also be preset to any value within its range by using the Apply Preset Value command. The lower 16 bits of this value are in register 0. A Position Value of 84,742,977 (0x050D 1341) is shown as an example.	0x1341
1		0x050D
2	**32 bit Velocity data.** This data is the calculated change in position over time. The unit of measure is set with the Velocity Format parameter. If the Scaling Function Control bit is ever set to a "1", the position data used to calculate the velocity data is always scaled by the Measuring Units per Span parameter. The lower 16 bits of this value are in register 2. A Velocity reading of 76,754 (0x0001 2BD2) is shown as an example.	0x2BD2
3		0x0001
4	**32 bit Raw Position Value.** This data is the actual position value read from the resolver. The resolution is always 65,536 counts per turn. This data is not affected by the value of the Measuring Units per Span parameter nor the Scaling Function Control bit. This value is also not affected by the Preset Value. The lower 16 bits of this value are in register 4. A Raw Position Value of 571,942,153 (0x2217 2509) is shown as an example.	0x2509
5		0x2217
6	**32 bit Time Stamp data.** This register is incremented every 400 nanoseconds while power is applied to the RM-89. This register rolls over every 1717.9869184 seconds. The time stamp can be used to verify active communications between the RM-89 and your host controller.	0x2BD2
7		0x0001

Table 5.5 Modbus Input Data Format

APPENDIX A
IP ADDRESS SETUP WITH BOOTP

You must use a Bootp server to set the IP address of these units. This appendix explains how to use the Bootp server from Rockwell Automation to set the IP address.

Initial Configuration

Starting at the beginning of chapter 3, RM-89 CONFIGURATION, follow the instructions up to the point where you have the RM-89 attached to your computer. This is explained in the Attach the RM-89 section on page 39.

Start Your Bootp Server

If needed, start your Bootp server. The Bootp-DHCP server software, version 2.3, from Rockwell Automation is used in this example. As shown in figure 5.4, the R.A. Bootp server window is broken down into two panes, "Request History" and "Relation List". "Request History" tells you what responses come over the network and the "Relation List" shows the setup data you have entered.

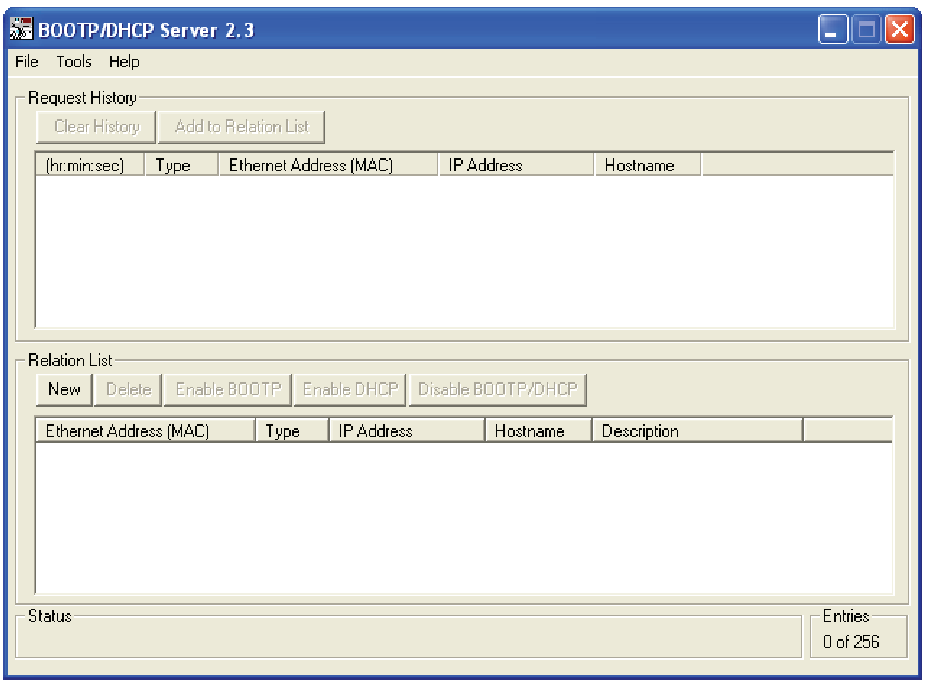


Figure 5.4 Rockwell Automation Bootp Server

Changing the IP Address

Changing the IP address of the RM-89 requires you to enable the Bootp protocol on the encoder before you can change the IP address. The RM-89 has the Bootp protocol disabled by default. This decreases the boot time by about 30 seconds when power is applied to the device because it doesn't have to wait for the Bootp request to time out before continuing with its stored address.

Changing the IP Address (continued)

Enabling Bootp Protocol

1. Make sure power is removed from the RM-89.
2. In the "Relation List" pane of the RA Bootp Server software, click on [New]. In the window that opens, enter the MAC address of the RM-89 which is printed on a white label near the serial number tag. You do not have to enter the "-" characters when entering the address on the screen. You must also enter the current IP address of the RM-89. This is 192.168.0.50 by default. The hostname and Description fields can be left blank. Click [OK].
3. Apply power to the RM-89 and wait for the Module Status LED to come on solid green and the Network Status LED to be flashing green.
4. Click on your new entry in the "Relation List". This will activate the buttons in the pane. Click on the [Enable BOOTP] buton. The message "[Enable BOOTP] Command successful" should appear instantly in the status line at the bottom of the window.
5. The BOOTP protocol is now enabled on the RM-89. Remove power from the encoder before continuing.

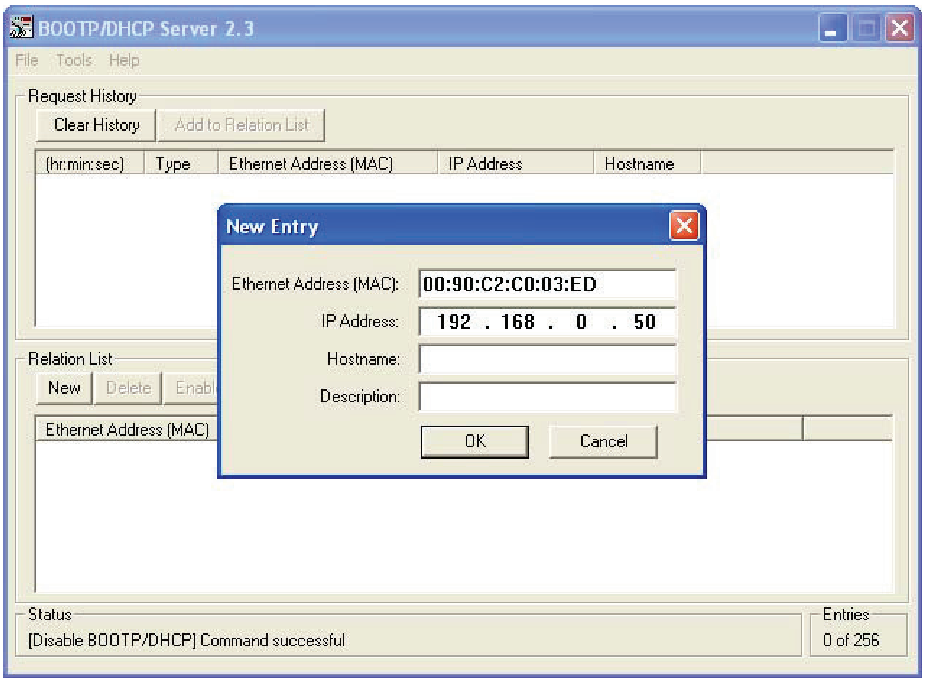


Figure 5.5 Add New Relation Entry

Changing the IP Address (continued)

Setting the IP Address
With the Bootp protocol now enabled, you can now change the IP address of the RM-89.

1. Double click on your new entry in the "Relation List" This will bring up the Properties window again. Enter the new IP address for the RM-89 and click [OK].
2. Apply power to the RM-89 and wait for the Module Status LED to come on solid green and the Network Status LED to be flashing green. At this point, you should also have a message in the "Request History" pane that lists the MAC address of the RM-89 along with the IP address you requested.

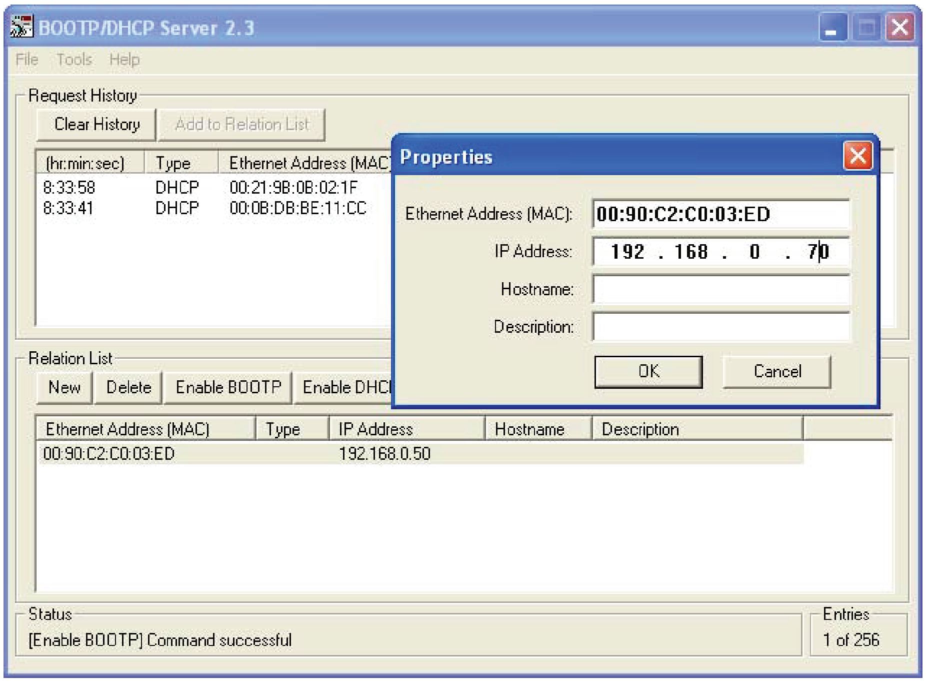


Figure 5.6 Setting New IP Address

Disabling the Bootp Protocol
Even though not strictly necessary, disabling the Bootp protocol will allow the RM-89 to boot up faster and prevent inadvertent changes to the IP address of the RM-89 if there is a network misconfiguration on your machine or plant floor.

1. With power still applied to the RM-89, click on your new entry in the "Relation List". This will enable the buttons above it.
2. Click on the [Disable BOOTP/DHCP] button. The message "[Disable BOOTP] Command successful" should appear instantly inthe status line at the bottom of the window. The new IP address for the RM-89 is now configured.

Testing the New IP Address

The easiest way to test the new address of the RM-89 is with the "ping" command. Before you can use the command, you have to be sure the RM-89 and your computer are still on the same subnet. For example, if the new address of the RM-89 is 192.168.0.42 and your computer has an address of 192.168.0.1, with a subnet mask of 255.255.255.0, then the two pieces of equipment are on the same subnet. (In this case, the first three numbers of the IP address must match.) If the new address of the RM-89 is 192.168.50.50, then the computer and RM-89 are not on the same subnet and you must go back into the Network Configuration panel and change your adapter's TCP/IP settings.

Once you are sure your computer and RM-89 are on the same subnet, open the DOS terminal if necessary by clicking on the [Start] button, and clicking on [Run...]. A dialog box will open. Enter 'cmd' on the text line and press [Enter] on the keyboard. Once the terminal is open, type in 'ping aaa.bbb.ccc.ddd' where 'aaa.bbb.ccc.ddd' in the new IP address of the RM-89. The computer will ping the RM-89 and the message "Reply from aaa.bbb.ccc.ddd: bytes=32 time<10ms TTL=128" should appear four times.

If the message "Request timed out." or "Destination host unreachable" appears, then one of three things has occurred:

- You did not enter the correct address in the ping command.
- The new IP address of the RM-89 was not set correctly.
- The RM-89 and the computer are not on the same subnet and the gateway setting on the computer are not configured to correctly forward packets to the subnet the RM-89 is on.

Continue with Chapter 3

If you haven't already, you can now use the RM-89 Configurator software to perform the initial setup of your RM-89. Refer to Using the TURCK Net Configurator starting on page 39 for additional instructions.

APPENDIX B
CIP POSITION SENSOR OBJECT

Common Industrial Protocol

EtherNet/IP is a protocol stack that implements the Common Industrial Protocol (CIP) over Ethernet using TCP/IP. The CIP is sponsored by the Open DeviceNet Vendors Association (ODVA) and is implemented over a variety of networks. The RM-89 follows the Encoder Device Profile that is defined in the CIP specification. The Configuration and Programming instances explained in chapters 3 and 4 are actually custom instances that simplify configuring and programming the encoder when using implicit messaging.

In addition to these custom instances, the RM-89 implements the Position Sensor Object, which is a mandatory object for every product that implements the Encoder Device Profile as defined in the specification. The explicit messages that are used to preset the position value and save the programmed parameters are two commands defined in the Position Sensor Object. The RM-89 implements the CIP revision 2 definition of the Position Sensor Object.

NOTE:
Using the Position Sensor Object to communicate with the RM-89 is completely optional. Most applications should communicate with the RM-89 using the custom instances as explained in the previous two chapters because it will greatly simplify your PLC programming. The only reasons to use the Position Sensor Object is if you need extremely fine grain control over communications with the RM-89 or if you use EtherNet/IP encoders from multiple vendors and you decide to write code that can be used with any of these sensors.

Supported Services

The following table lists the common services implemented by the RM-89 for the Position Sensor Object.

Service Code	Implemented		Service Name	Description of Service
	Class	Instance		
16#05	Yes	No	Reset	Resets all parameter values to the factory default
16#0E	Yes	Yes	Get_Attribute_Single	Returns the contents of the specified attribute
16#10	No	Yes	Set_Attribute_Single	Modifies an attribute value
16#15	Yes	No	Restore	Restores all parameter values from non-volatile storage
16#16	Yes	No	Save	Saves all programmable parameters to the non-volatile storage including the position offset derived from setting the Preset Value, (Attribute 16#13)

The services that are implemented only on the Class level (not on the Instance) should address Instance 0.

Table 5.6 Supported Services

- Service Code 16#0E, Get_Attribute_Single is used to read data from the Position Sensor Object class.
- Service Code 16#10, Set_Attribute_Single is used to write data to the Position Sensor Object class.

Supported Class Attributes

The only supported Class attribute is 1 and it returns the revision number of the definition of the object. Because the RM-89 implements the CIP revision 2 definition of the Position Sensor Object, this attribute will always return a value of "2".

Supported Instance Attributes

Table 5.8 on the following two pages lists all of instance attributes implemented by the RM-89. Table 5.7 below describes the Data Type values used in this table.

Data Type	Length	Description
BOOLEAN	8 bits	Holds single on/off (true/false) value
BYTE	8 bits	Holds up to 8 bits of data which should not be considered to be a scalar value
USINT	8 bits	Unsigned 8 bit value
WORD	16 bits	Holds up to 16 bits of data which should not be considered to be a scalar value
UINT	16 bits	Unsigned 16 bit integer value
DINT	32 bits	Signed 32 bit integer value
UDINT	32 bits	Unsigned 32 bit integer value

Table 5.7 Explanation of Data Types

Attrib. ID	Access	Name	Data Type	Description
16#01 - 1	Get	Number of Attributes	USINT	Number of supported Attributes = 21
16#02 - 2	Get	Attribute List	Array of BYTE	List of supported Attributes = 01, 02, 0A, 0B, 0C …71hex
16#0A - 10	Get	Position Value Signed	DINT	Current position value
16#0B - 11	Get	Position Sensor Type	WORD	Specifies the device type 1 = Single turn absolute rotary encoder 2 = Multi-turn absolute rotary encoder
16#0C - 12	Set	Direction Counting Toggle	BOOLEAN	Controls the counting direction: 0 = CW 1 = CCW
16#0E - 14	Set	Scaling Function Control	BOOLEAN	Enables Scaling function 0 = OFF (65,536 counts per turn) 1 = ON (Scaling set by Measuring Units per Span, attribute 10hex)
16#10 - 16	Set	Measuring Units per Span (Counts per Turn)	UDINT	Resolution for one revolution: 1 to 65,536 counts per turn
16#11 - 17	Set	Total Measurement Range	UDINT	Counts before roll over to zero. Single Turn RM-89: Range of 0, 2 to 65,536 28 bit Multi-turn RM-89: Range of 0, 2 to 268,435,455 30 bit Multi-turn RM-89: Range of 0, 2 to 1,073,741,823
16#13 - 19	Set	Preset Value	DINT	Sets the position to the specified value. Calculates an internal offset that will be saved to the non-volatile storage if Save service (code 16#16) is issued.
16#18 - 24	Get	Velocity Value	DINT	Current speed. The value is in the format specified by attribute 16#19

(Table is continued on next page)

Attrib. ID	Access	Name	Data Type	Description
16#19 - 25	Set	Velocity Format	WORD	Format of the velocity attribute: 16#1F04 = pulses/s 16#1F05 = pulses/ ms 16#1F07 = steps/min 16#1F0F = RPM
16#29 - 41	Get	Operating Status	BYTE	Encoder diagnostic operating status. Bit 0 = Value of attribute 16#0C (12) Bit 1 = Value of attribute 16#0E (14)
16#2A - 42	Get	Physical Resolution Span	UDINT	Physical resolution of the single-turn re-solver sensor
16#2B - 43	Get	Number of Spans	UINT	Maximum number of revolutions that could be measured.
16#2C - 44	Get	Alarms	WORD	Indicates a malfunction has occurred.
16#2D - 45	Get	Supported Alarms	WORD	Information about supported alarms
16#2E - 46	Get	Alarm Flag	BOOLEAN	Indicates that an alarm error occurred: 0 = No errors 1 = Alarm Error
16#33 - 51	Get	Offset Value	DINT	The internal position offset that is cal-culated after applying the Preset Value through attribute 13hex (19)
16#64 - 100	Set	Device Type	DINT	The way the device identifies itself: 16#22 (default) = Encoder device 16#00 = Generic device
16#70 - 112	Get	Actual Sensor Reading	UDINT	Raw position value read from RM-89
16#71 - 113	Get	Time Stamp	UDINT	Value increments every 400 nanosec-onds.

Table 5.8 Supported Instance Attributes

NOTE:
For detailed description of the Attributes, see the CIP definition.

Supported Alarms

The RM-89 supports the following operational alarm.

- Diagnostic Error

This alarm is set when the RM-89 fails its power up diagnostics. The Position Error alarm is also set to indicate that the position data may be incorrect.

Attributes 16#2D, Supported Alarms, 16#2C, Alarms, and 16#2E Alarm Flag indicate something about the alarms supported by the RM-89.

- 16#2D: Supported Alarms – Reading this attribute returns a value of 3, indicating that the Position Error alarm and Diagnostic Error alarms are both used.
- 16#2C: Alarms – Reading this attribute will return a value of zero if no alarms have occurred and a value of three if an alarm has occurred.
- 16#2E: Alarm Flag – Reading this attribute will return a value of zero if no alarms have occurred and a value of one if an alarm has occurred.