Reference Manual
00809-0100-4023, Rev AA
May 2003

Rosemount 3420

Fieldbus Interface Module

ROSEMOUNT®

www.rosemount.com

EMERSON™
Process Management

Reference Manual
00809-0100-4023, Rev AA
May 2003

Rosemount 3420 Fieldbus Interface Module (FIM)

NOTICE

Read this manual before working with the product. For personal and system safety, and for optimum product performance, make sure you thoroughly understand the contents before installing, using, or maintaining this product.

Within the United States, Emerson Process Management has two toll-free assistance numbers:

Customer Central
Technical support, quoting, and order-related questions.

1-800-999-9307 (7:00 am to 7:00 pm CST)

North American Response Center
Equipment service needs.

1-800-654-7768 (24 hours—includes Canada)

Outside of the United States, contact your local Rosemount representative.

⚠CAUTION

The products described in this document are NOT designed for nuclear-qualified applications. Using non-nuclear qualified products in applications that require nuclear-qualified hardware or products may cause inaccurate readings.

For information on Rosemount nuclear-qualified products, contact your local Rosemount Sales Representative.

Table of Contents

Section 1 Introduction

OVERVIEW

This manual provides installation and troubleshooting instructions for the Rosemount 3420 Fieldbus Interface Module (FIM).

Using this Manual

The sections in this manual provides information on installing, operating, and maintaining the Rosemount 3420 Fieldbus Interface Module.

- Section 2: Installation contains mechanical and electrical installation instructions.
- Section 3: Configuration provides instruction on commissioning and operating the Rosemount 3420 Fieldbus Interface Module. Information on software functions, configuration parameters, and online variables are also included.
- Section 4: Operation and Maintenance contains operation and maintenance techniques.
- Section 5: Troubleshooting provides troubleshooting techniques for the most common operation.
- Appendix A: Reference Data supplies reference and specification data, as well as ordering information.
- Appendix B: Approval Information contains intrinsic safety approval information, European ATEX directive information, and approval.
- Appendix C: Fieldbus Status Values contains fieldbus function block output status codes.

Service Support

To expedite the return process outside of the United States, contact the nearest Rosemount representative.

Within the United States, call the Rosemount National Response Center using the 1-800-654-RSMT (7768) toll-free number. This center, available 24 hours a day, will assist you with any needed information or materials.

The center will ask for product model and serial numbers, and will provide a Return Material Authorization (RMA) number. The center will also ask for the process material to which the product was last exposed.

Rosemount National Response Center representatives will explain the additional information and procedures necessary to return goods exposed to hazardous substance can avoid injury if they are informed of and understand the hazard. If the product being returned was exposed to a hazardous substance as defined by OSHA, a copy of the required Material Safety Data Sheet (MSDS) for each hazardous substance identified must be included with the returned goods.

Section 2 Installation

OVERVIEW

A Quick Installation Guide (00825-0100-4023) is shipped with every Module to describe the basic mechanical and electrical procedures for initial installation. Dimensional drawings are included in Appendix A: Reference Data. A PC with an Ethernet port is required to perform the initial configuration of the device.

Safety Messages

Instructions and procedures in this section may require special precautions to ensure the safety of the personnel performing the operations. Information that raises potential safety issues is indicated by a warning symbol (⚠). Please refer to the following safety messages before performing an operation preceded by this symbol.

Warnings

⚠WARNING

Explosions could result in death or serious injury:

- Do not remove the transmitter from its mounting enclosure in explosive atmospheres when the circuit is live.
- Verify that the operating atmosphere of the transmitter is consistent with the appropriate hazardous locations certifications.

Electrical shock could cause death or serious injury. If the device is installed in a high-voltage environment and a fault condition or installation error occurs, high voltage may be present on transmitter leads and terminals.

- Use extreme caution when making contact with the leads and terminals.

Failure to follow these installation guidelines could result in death or serious injury:

- Make sure only qualified personnel perform the installation.

General Considerations

The Rosemount 3420 may be mounted in any General Purpose location. Be sure the covers are secured tightly to prevent exposure of the electronics to moisture and contamination.

CONFIGURATION NOTE
Before connecting the fieldbus segments to the 3420, you must first connect a PC and configure the Rosemount 3420. Once the Plug and Play features have been setup you can then make the final connection of the segments to the 3420 terminal strip.

INSTALLATION PROCEDURE

For dimensional drawing information refer to Appendix A: Reference Data on page A-4.

The cast aluminum housing encloses the electronics circuitry of the FIM. The front of the enclosure has two covers; an upper cover and a junction box cover.

The upper cover provides access to the electronics assembly which includes the microprocessor, fieldbus interface boards, fieldbus power conditioners/terminators, and the power supply board.

The junction box cover provides access to the terminal block. To open either cover of the enclosure, use a 1/4 inch blade screwdriver to remove the appropriate screw on the unhinged side of the enclosure.

Mounting Procedure

The FIM can be mounted to a support bracket on a wall or to a pipe.

Mounting the FIM to a Support Bracket

The following hardware and tools are needed:

- Four 15/16 inch bolts
- Mounting support
- 3/8 inch drill
- 1/2 inch socket-head wrench

Mount the FIM by doing the following:

1. Drill four 15/16 inch (7.94 mm) holes in the support bracket to which the FIM will be mounted.
2. Using a 1/2 inch socket-head wrench, attach the FIM to the support bracket with four 15/16 inch bolts.

Mounting the FIM to a Pipe

The following hardware tools are needed:

- Pipe mount with holes spaced 2.81 inch (71 mm) apart horizontally and 11.15 inch (283 mm) apart vertically.
- Two 5/16 inch U-bolts
- 1/2 inch socket-head wrench

Mount the FIM by doing the following:

1. Insert one U-bolt around the pipe and through the top mounting holes of the pipe mount and the FIM and another U-bolt through the bottom mounting holes of the pipe mount and the FIM.
2. Using a 1/2 inch socket-head wrench, fasten nuts to the U-bolts.

Grounding the FIM

If mounting the 3420 in the field, ground the FIM with a connection of 1 ohm or less leading from the external grounding lug to earth ground. If mounting the FIM in the control room, a cabinet ground is sufficient. In either location, follow local or plant electrical codes.

Wiring the FIM

FIM wiring is done in the terminal block. For access to the terminal block, open the junction box cover following the instruction “Installation Procedure” on page 2-2. The terminal block label is located on the inside of the FIM junction box cover.

At the bottom of the junction box in ¾ inch NPT conduit entries are three plastic plugs that were placed there at the factory. Three metal plugs were shipped with the FIM and are used to seal any unused ports.

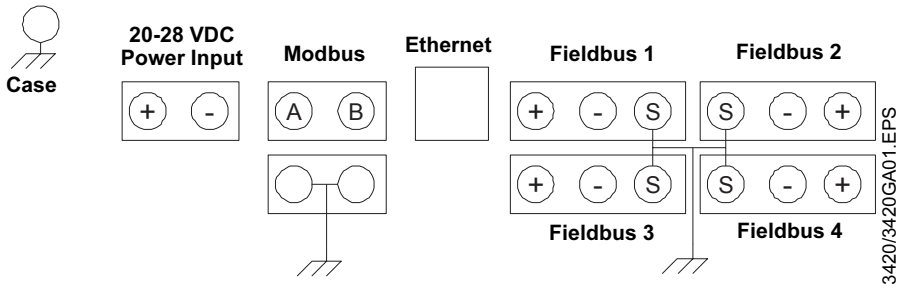
The FIM case should always be grounded in accordance with national and local electrical codes. The most effective grounding method is direct connection to earth ground with minimal impedance.

The internal Ground Connection located with the supply terminals is the Internal Ground Connection screw. This screw is identified by the following symbol:

NOTE
Grounding the FIM case via threaded conduit connection may not provide sufficient ground.

The wiring should include an external power shut-off switch or an external circuit breaker. This device should be located near the FIM.

Figure 2-1. Terminal Wiring Diagram



FIM Input Power Connection

The FIM is designed to be powered by 24 V dc power. Use a power supply suitable for 185°F with sufficient capacity to power both the FIM and all of the fieldbus devices that are connected to it. The 3420 requires 500 mA. About 300 mA of the required should be allocated for each H1 segment. The positive and negative power terminals are found on the left side of the terminal block. A case ground is also found on the left hand side of the compartment.

Connecting the Devices

Network

The FIM is equipped with a 10/100 Base-T Ethernet interface receptacle in the center of the terminal block. Connect the FIM to the PC that will be used for the configuration by creating a local network using the cross-over cable provided with the FIM connected to the RJ-45 receptacle on the FIM and the Ethernet port on a PC. Alternatively you can use a standard Ethernet cable and an Ethernet HUB to make this connection.

Fieldbus

The fieldbus terminals are found on the right side of the terminal block in 4 sets of three terminals for the positive and negative conductors and a shield. Although the FIM is not polarity sensitive other components in the segment such as junction blocks may require correct polarity.

Modbus

On the left side of the terminal block there are four sets of two wiring mounting points. There is only one Modbus interface which is the two upper most terminals on the right side. The Modbus interface is polarity sensitive. Connect the negative to the right most terminal (B) and the positive to the left terminal (A).

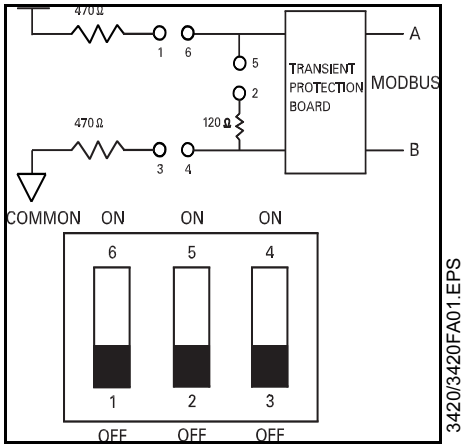
Modbus Termination Setup

NOTE:
Do not open the 3420 electronics housing in an explosive atmosphere.

Modbus RTU is transmitted on a RS485 physical layer. Three dip-switches are provided to enable the RS485 circuitry with a network terminator. The switches are found inside the electronics housing on the RS485 communication board located in the top center of the housing. Switch 2 places a 120 ohm terminator on the bus. This would be used to match cable impedance if needed to dampen reflections on long cable runs. Its use will depend on the baud rate and cable length of the Modbus network.

Switches 1 and 3 are connected to pull-up and pull-down resisters on the Modbus network. These resisters are used to prevent noise from being interpreted as valid communications during periods when no communications is occurring on the network. Only one set is required on a RS-485 network.

Figure 2-2. Modbus Setup



Section 3 Configuration

⚠WARNING

Explosions could result in death or serious injury:

- In an Explosion-Proof and Flame-Proof environment, do not open the Rosemount 3420 electronic housing in an explosive atmosphere.
- Cover must be fully engaged to meet Explosion-Proof requirements.

Private Networks

To configure the 3420, a private network between a PC and the Rosemount 3420 must first be established. This can be done with a PC dedicated to the 3420 or a PC used for another purpose can be temporarily configured for the task. If a PC from another network is used, carefully record the IP address and other settings so the PC can be returned to it's original network when finished with 3420. If using a PC attached to another network, shutdown the PC and remove it from the network before proceeding to set up the 3420 private network.

Configuration of the FIM is done through its web interface. To access the device, you must create a private LAN with a subnet of 192.168.1.XX. The FIM will appear on this LAN at the IP address 192.168.1.10. The PC that is used for configurating the device should have another IP address on the same LAN (i.e. 192.168.1.12).

NOTE
Before leaving the webpage that you make changes to, click **Submit**, or all your changes will be lost.

1. On your PC, install the Java Plug-in found on the CD provided with the FIM. You can also find the Java Plug-in at: http://java.sun.com/j2se.
2. Select Windows JRE and install it on your computer.
3. Under Network and Dial-Up setting in your Control Panel:
   a. Select your network port
   b. Select TCP/IP and then Properties
   c. Check the manual IP address button
   d. Set your IP address to 192.168.1.12
   e. Network to 255.255.255.0
   f. Restart your computer to use the new settings.
4. Using the cross-over Ethernet cable attach your PC to the Rosemount 3420. Add an entry in your hosts for the fim3420.

NOTE
This will only need to be done once. Skip this step if you connect to the FIM again.

5. Check the Network connection using the Command Prompt (usually found under Accessories) type: PING 192.168.1.10. If you see replies the network is setup properly.

Adding an entry in your hosts for the fim3420

6. In Windows 2000 or Windows NT go to the following: c:\winnt\system32\drivers\etc\hosts

   In Windows XP go to the following: c:\windows\system32\drivers\etc\hosts

   In Window 95 or Windows 98 go to the following directory: c:\windows\...(find a template called hosts.sam).

NOTE
If there is not already a file named *hosts* in c:\windows, copy *hosts.sam* to hosts and edit the c:\windows\hosts file.

7. Using a text editor (example: Notepad) add the following line to the end of the file:
   192.168.1.10 fim3420.
8. Save your file. Save the changed file as hosts (with no extensions) if you used the host.sam template.

9. Check the Network connection type: PING fim3420. If you see replies you know you have the hosts file setup properly.
10. Open your Internet Explorer browser.
11. Select LAN settings (located under **Tools>Internet Options> Connections**) and turn off the box saying to use the Proxy Server.
12. In your browser address bar type: http://fim3420.
13. Press **Enter**. (You should get the main page.)
14. At the security LogOn box:
    a. LogOn as User: **admin**
    b. Password of: **fieveladmin**

Network

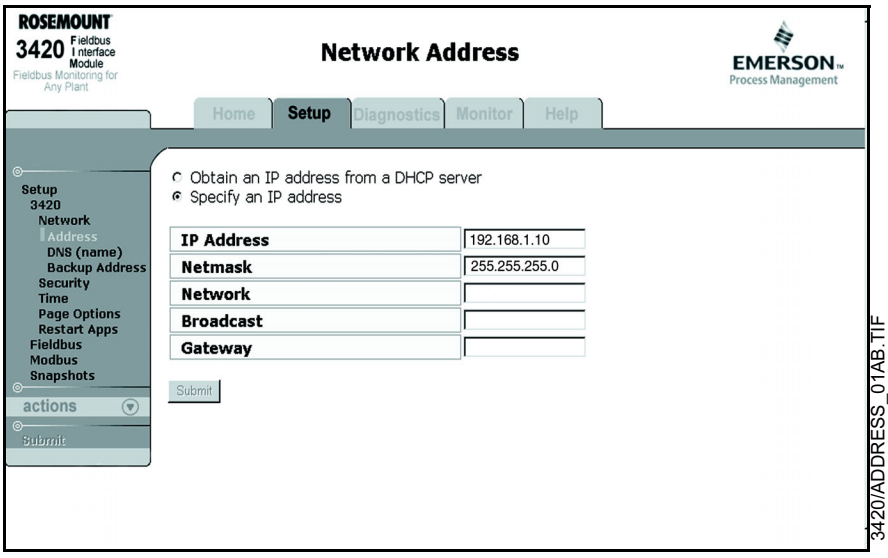
Click **Setup>3420>Network** to enter network parameters. If the FIM is connected to a LAN or if more than one FIM will be used on a private network, the unit will need to be given a new IP address.

Address

Select to have the device attain an IP address via DHCP or be statically assigned an IP address (Figure 3-1). Contact your network administrator if you are not sure which selection is appropriate.

NOTE
If you accidentally misconfigure the network settings and can not reach the device at the new IP address, return the device to the private LAN you used for initial configuration. You can still access the FIM by its default IP address (192.168.1.10) in this environment.

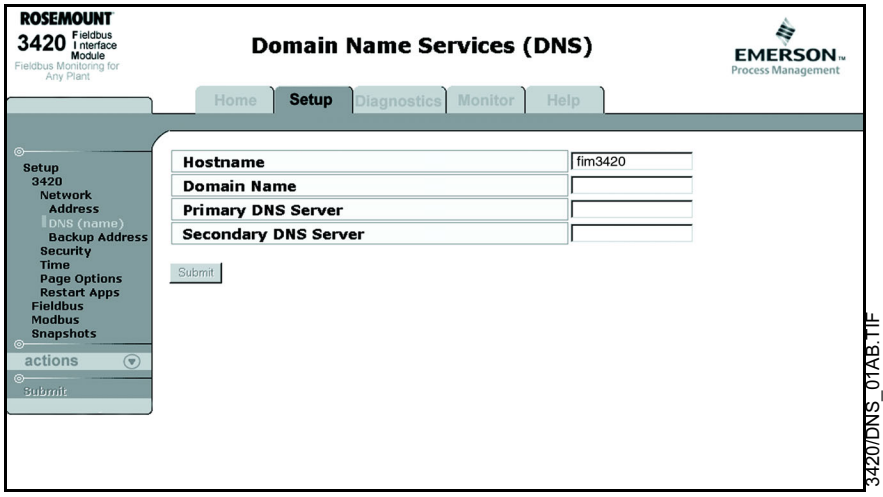
Figure 3-1. Network Address



Domain Name Services (DNS)

Click **Setup>3420>Network>DNS** to enter DNS settings. DNS settings are not required for the device to be operational but it will provide better diagnostic information if you configure these settings. If you do not want the FIM to use a nameserver or Domain name, you should clear the appropriate fields.

Figure 3-2. Domain Name Services (DNS)



Backup Address

This address should only need to be changed if your internal corporate network uses non-routable IP addresses for its internal use and they use the 192.168.1.xxx subnet. If this is the case, you will want to change the default IP address to an address that does not conflict with an address that is in use. Please consult your network administrator if you can not make that determination yourself.

Click **Setup>3420>Network>Backup Address** to configure the backup IP Address settings.

Be very careful when changing these settings. The device can be rendered unusable if these values are modified incorrectly.

Security

Click **Setup>3420>Security** to change the administration and administration passwords. These passwords allow for varying levels of application access. The administrator can modify any system or field device setting. In contrast, the operator is only able to modify some Fieldbus parameters. Use caution when changing the administrator password. If the administrator password is lost, you will not be able to setup the Rosemount 3420. The FIM is shipped with the following default passwords:

Table 3-1. Default Passwords

ID	PASSWORD
Executive (exec)	showme
Operator (oper)	runit
Maintenance (maint)	keepitgoing
Administrator (admin)	fieveladmin

Table 3-2. Access Table

Role	HTML Access	Tree view Access
Executive (exec)	With the exception of factory settings (Setup/factory.html), can get any page (Read-Only access).	Read-only access to AI and MAI blocks. (VFDs and other blocks are not visible.
Operator (oper)	No additional privileges	Same values as executive, but with read-write access.
Maintenance (maint)	• Can set device PD TAGs • Can set block tags • Can configure Modbus communications • Can configure Modbus register map • Can configure snapshots • Can configure Plug and Play and Operating Modes	All parameters of all blocks (Read-Write).
Administrator (admin)	• Can configure network settings (address, dns, default). • Can set passwords • Can set time settings • Can set home page options • Can restart applications	No additional privileges

System Time Setup

Click **Setup>3420>Time** to configure the system time. If your Rosemount 3420 is connected to a network and you want to use this feature, you should select a timeserver at your facility or one near you geographically to insure accurate time adjustments. The device will function properly with this feature disabled but data time stamps will be less accurate and time updates must be entered for each Rosemount 3420.

NOTE
If you do not provide proper DNS information, you must provide the IP address of the timeserver instead of its hostname.

Alternately you can set the time manually. This is accomplished by unchecking the “Enable Network Time Protocol” check box. This will enable you to enter information into the “Date” and “Time” fields. Also set the timezone and daylight savings time setting appropriate for your local area.

Page Options

Click **Setup>3420>Page Options** to choose the look of the Home Page. You can select Menu Overview, Fieldbus Diagnostics, or Fieldbus Explorer.

Restart App

Click **Setup>3420>Restart Apps** to restart the Rosemount 3420 application software. This is not needed during normal operation but may be required during troubleshooting of a system issue. Simply select **Yes** to restart or **No** to abort. The restart will take about 2 minutes. Click on a different screen to refresh the screen and continue working.

FIELDBUS FIELD DEVICES

The FIM collects data based on the TAG of the function blocks in the devices. If the devices are not pre-configured with this information it can be edited using the FIM web interface.

Devices

Click **Setup>Fieldbus>Device** to rename the PD (Physical Devices) TAG for individual devices. The PD TAG is changed by entering up to 32 characters in the field and clicking **Submit**. Allow 2 minutes for the update to take effect.

Device Explorer

Click **Setup>Fieldbus>Explorer** to display the fieldbus tree view. This view allows you to see all the segments and devices attached to the segments of the Rosemount 3420. It provides an overview of the devices, blocks, and parameters and provides a means to write parameter values.

NOTE
You can also reach the fieldbus tree view by selecting the **Monitor** tab.

Blocks

Click **Setup>Fieldbus>Blocks** to set up your fieldbus blocks. If the device's function blocks are not already identified with a TAG then you can use this display to enter them into the device. This TAG information is used to assign Modbus registers and other functions in the FIM. Block names (TAGs) may be 32 characters in length and are case sensitive. Once all the changes have been made, click **Submit**. Allow 2 minutes for the update to take effect if several TAGs are changed at once.

NOTE
All Analog Input (AI) and Multiple Analog Input (MAI) block Tags must be unique on any 3420.

Advanced

Click **Setup>Fieldbus>Advanced** to access the advanced fieldbus setup features.

Plug and Play

Click **Setup>Fieldbus>Advanced>Plug and Play** to schedule your fieldbus blocks. Function blocks in a device are not normally scheduled to run when they are shipped from the factory. The host application is expected to schedule the function blocks that it needs for the application. In this case the Plug and Play feature provides this service.

The Rosemount 3420 provides automatic field device configuration for the typical monitoring applications. When a new field device is detected on the segment, the plug and play software checks each AI/MAI block in the device to determine if it is scheduled to execute. If the block is already scheduled, the 3420 will leave that block alone.

If a block is found that is not scheduled, it will be scheduled and the CHANNEL, XD_SCALE, OUT_SCALE, and L_TYPE will be set for the default values. The default units for XD_SCALE and OUT_SCALE are set through the plug and play configuration. If you desire something other than the default values for a specific device, any function block parameter can be changed using the Fieldbus Tree View accessed under the Monitor tab.

A portion of the plug and play is device specific. For example, a Rosemount 3244MVF device channel configuration is set so AI1 is set for sensor 1, AI2 is set for sensor2. For the Rosemount 3051, AI1 is setup for pressure. Also, the units are set according to the plug and play configuration as stated above.

When the Rosemount 3420 and all of the fieldbus devices have been setup the way you want them, check the box next to **Operating Mode**. In Operating Mode the 3420 will only scan the blocks that are required for the fieldbus mapping and Snapshot files for optimal performance. If the unit is in Setup Mode (Operating Mode unchecked) all of the blocks are scanned to facilitate the configuration and commissioning of the 3420 and attached devices.

Figure 3-3. Plug and Play

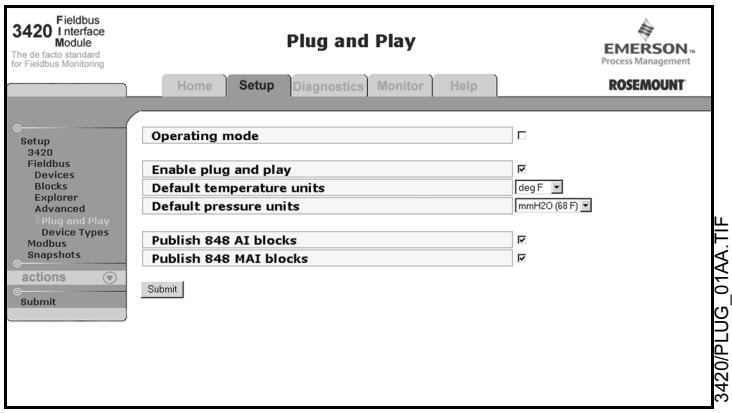


Table 3-3. Plug and Play Default Settings

Description	Default Setting
Operating Mode	unchecked
Enable Plug and Play	check
Default Temperature Units	deg F
Default Pressure Units	mmH_2O 68°F
Publish 848 AI Blocks	check
Publish 848 MAI Blocks	check

The Rosemount 848T temperature transmitter provides 8 AI blocks and 1 MAI block. A MAI block process all eight inputs at once. The speed at which the 3420 scans all of the measurements in devices on the fieldbus segments is dependent on the number of AI and MAI blocks being polled. For example if 13 Rosemount 848T's were polled with MAI blocks the scan time would be approximately 1 second for all 104 values. If the same 13 devices were polled for with the AI blocks the scan time would be about 6 seconds (See Table 3-4 on page 3-8).

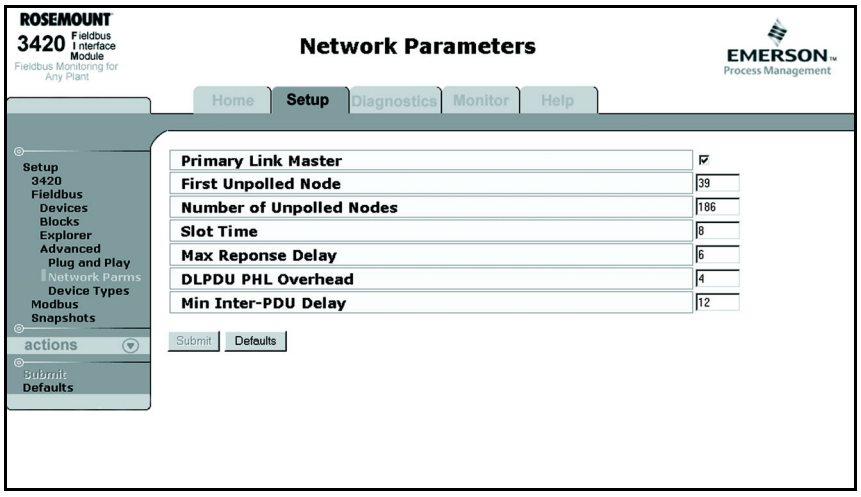
Table 3-4. Scan Rate

Number of AI or MAI Blocks scanned on each segment	Scan rate
1	0.3 seconds
2	0.4 seconds
4	0.5 seconds
8	0.7 seconds
16	1.0 seconds
32	2.7 seconds
64	3.1 seconds
128	7.0 seconds

Network Parameters

Click Setup>Fieldbus>Advanced>Network Parms to display the fieldbus network parameters. Under normal circumstances these values should not have to be changed. However if a new device is added to a segment with significantly different communication parameters it may require that these values be adjusted. Contact Rosemount technical support before making any changes.

Figure 3-4. Network Parameters

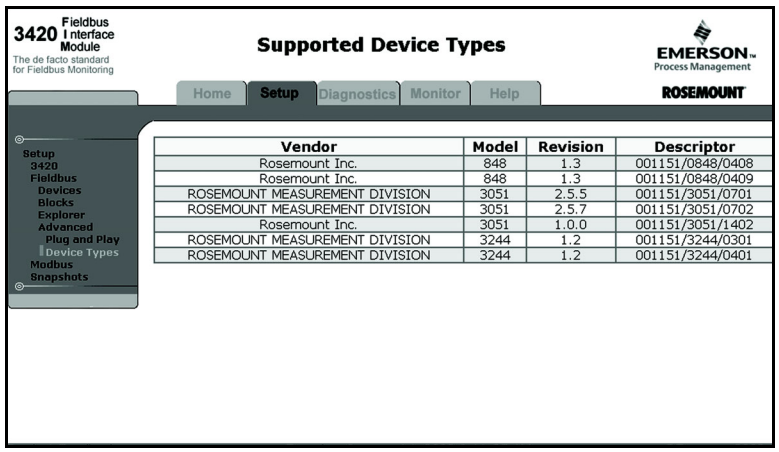


3420/NETPARAMETERS.TIF

Device Types

Click **Setup>Fieldbus>Advanced>Device Types** to display all your Rosemount 3420 supported device types that are currently loaded and supported by the 3420.

Figure 3-5. Example of Supported Device Types



3420/DEVICES_01AA.TIF

MODBUS

Click **Setup>Modbus** to configure the Modbus Interface. Most of these settings are self-explanatory and are related to configurating the serial port to match the settings used by the Modbus Master.

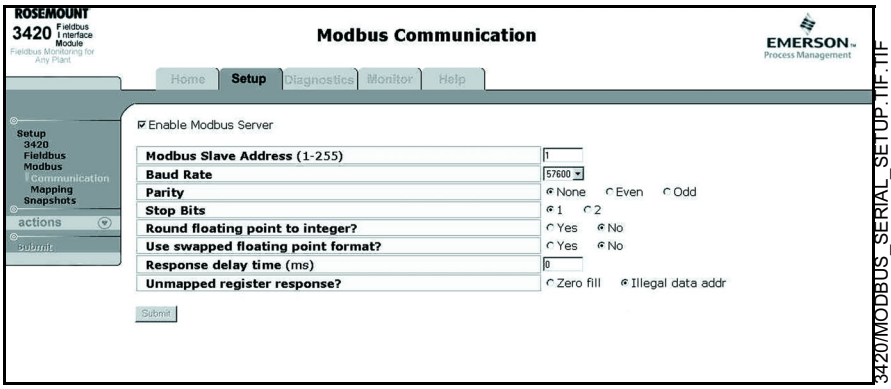
Communication

Click **Setup>Modbus>Communication** to configure the Modbus Master. The measured values can be represented as either a single register integer number or a two-register (standard or swapped) floating point number. One common difference in Modbus masters is the representation of a floating point number. The default used by the FIM3420 is a Standard Floating Point but this configuration page also allows you to use Swapped Floating Point which reverses the order in which the data in the floating point registers are sent.

The “Response delay time” entry allows you to have the 3420 wait for the specified amount of time before it outputs its response to the master request. Some master devices may not be able to immediately receive the response due to receiver setup time. This setting accommodates those master devices.

The unmapped register response setting allows the selection of the value entered into a register if no TAG is assigned to it.

Figure 3-6. Modbus Communication

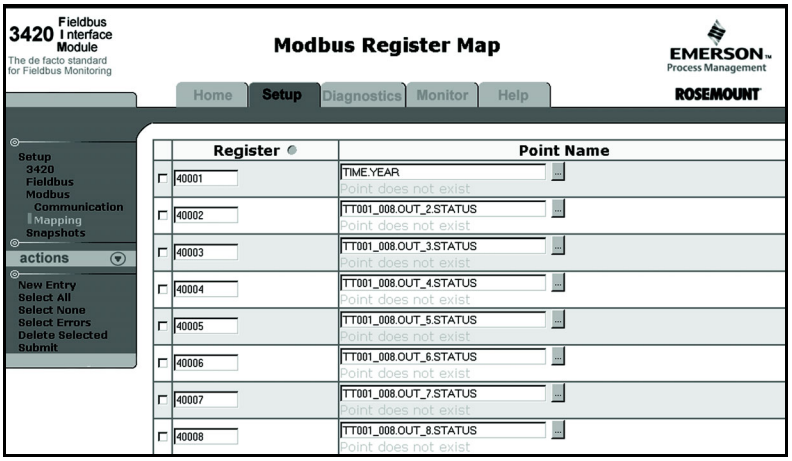


Mapping Fieldbus Tags to Modbus Registers (Web Interface)

Click **Setup>Modbus>Mapping** to map Fieldbus tags to Modbus Registers. This allows a Modbus master to read a given register on the FIM and effectively be reading a parameter from a device on the Fieldbus. The mapping webpage provides the ability to assign a register number to any AI or MAI function block output by selecting the Block Tag from a drop down list.

Click the header for a column to sort the data by register number or point name. If there is a device with duplicate Block Names, only the block for the first device found will be displayed. Check the Fieldbus/Block Setup display to verify that there are no Analog Input or multiple Analog Input Block TAGs that were duplicated.

Figure 3-7. Register Mapping



3420/REGISTERMAP.TIF

Mapping Fieldbus Tags to Modbus Registers (FTP)

This mapping is contained in comma-separated-value (or csv) file on the FIM. As an alternative to using the mapping webpage, this file can be read, modified, and re-written to the FIM. This file is named modbus.csv and is located on the FIM at the path:

/home/fievel/config/modbus.csv

If you ftp in as fievel, this can be shortened to: config/modbus.csv.

Here is an example of a small modbus.csv file:

TT001.OUT.STATUS, 40001

TT002.OUT.STATUS, 40002

TT003.OUT.STATUS, 40003

TT004.OUT.STATUS, 40004

TT848.OUT_1.STATUS, 40005

TT848.OUT_2.STATUS, 45006

TT001.OUT.VALUE, 45001

TT002.OUT.VALUE, 45003

TT003.OUT.VALUE, 45005

TT004.OUT.VALUE, 45007

TT848.OUT_1.VALUE, 45009

TT848.OUT_2.VALUE, 45011

Modbus Register Rules

The small example file above is compliant with some rules that you must follow when changing the mappings. These rules are:

- When the AI and MAI Function Block Output data is in floating point format it requires two registers. Therefore nothing may be mapped into the next register specified for a floating point measurement value.
- Status information uses 1 register and can be located in adjacent registers of the status information of other Tags.
- Contiguous registers must all be of the same type.
- Do not use registers 49001 through 49011 (see “Predefined Diagnostic Registers” on page 3-13).

Modbus Register Guidelines

The Modbus protocol allows for reading contiguous registers of the same datatype in one read request from the Modbus Master (up to 127 registers can be communicated in one read request). To take advantage of efficiently reading registers, the following guidelines are suggested:

- Fieldbus output STATUS tags should be mapped to registers in one contiguous block starting at register 40001. Each tag requires one register.
- Fieldbus output VALUE tags should be mapped to registers in one contiguous block starting at register 45001. Each tag requires two registers.

Modifying the Fieldbus to Modbus Mapping

The mapping from tags to registers is modified as follows:

1. Retrieve the modbus.csv file from the FIM via FTP (username: fievel, password: fievel).
2. Open the modbus.csv file using Excel.
3. Modify entries in the table as desired.
4. Save the changes in the csv format to the file modbus.csv.
5. Download the modified modbus.csv file to the FIM via FTP.

After the download has been completed, the FIM will detect the changes and will start using the new mappings within 20 seconds.

Predefined Diagnostic Registers

The following is a table of predefined diagnostic registers. Do not use any of these registers in the Modbus register map file (modbus.csv).

Table 3-5. Predefined Diagnostic Registers

Description	Register
Current Year (1)	49001
Current Month (1)	49002
Current Day (1)	49003
Current Hour (1)	49004
Current Minute (1)	49005
Current Second (1)	49006
Messages Received	49007
Corrupt Messages Received	49008
Messages Sent with Exception (error responses)	49009
Messages Sent Count	49010
Valid Messages Ignored (when in listen only mode)	49011

(1) Time is reported in GMT.

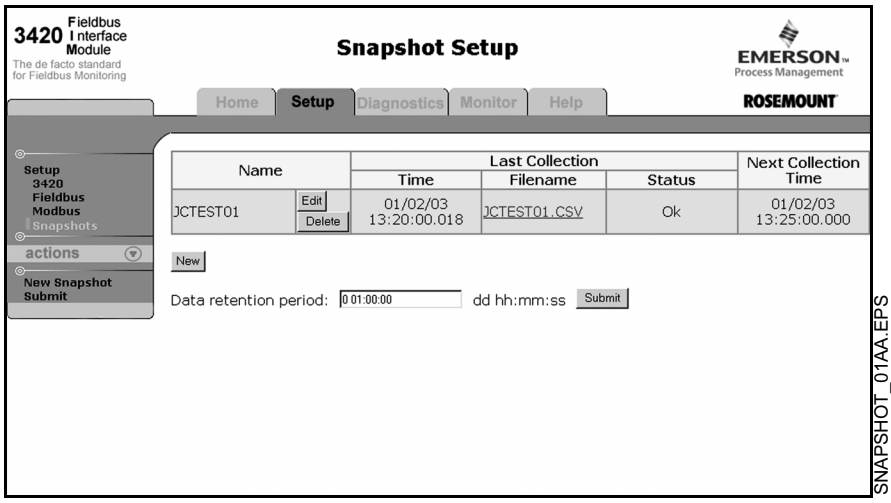
The Message Sent Count is the most useful data for determining if the Modbus slave is finding errors in the messages or is rejecting the messages. The message received and messages sent count should be the same if no errors are encountered.

SNAPSHOT FILES

Snapshot File Setup

Click **Setup>Snapshot** to display the current snapshot collection information. Snapshots are files in the specified format (CSV or XML) that contain periodically captured data and optional time stamp of selected device values.

Figure 3-8. Snapshot Setup



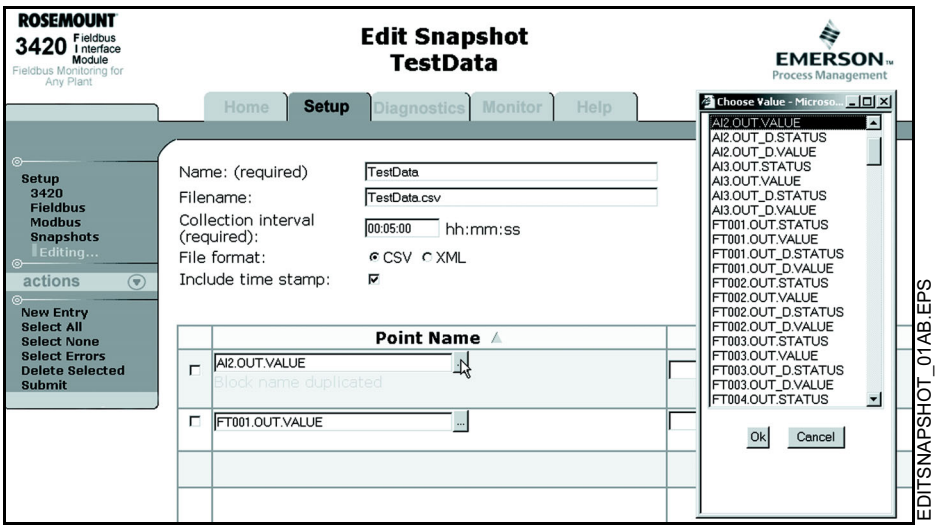
Data can be accessed by a host system by reading a file using FTP over Ethernet. The files are called Snapshot files.

New or Edit Snapshot Data

New Snapshot will take you to an edit screen that will allow you to select the name of the snapshot, the interval of collection, the file format, an optional timestamp and selected device values. Add values to be collected by selecting **New Entry** and then selecting the tag from the drop down menu.

Once the file has been setup you can click on the file name to display the current contents of the file.

Figure 3-9. New or Edit Snapshot Data



Snapshot File Format

The file is in Comma Separated Value (CSV) or XML format as follows:

- The file consists of series of lines that contain two fields: the label for the value and the value itself.
- The first record which contains the data timestamp is optional. If enabled, it is always the first value reported. Time values are reported in GMT time.
- Any valid AI or MAI block output can be configured.
- The label for each value is the fully qualified name of the value. This is: <block tag>. <parameter name>.<subparameter name>

  Example: TT-800-1.OUT.STATUS
- The label is user definable.
- The value is a textual representation of the point being collected. If the point cannot be collected (the device is offline, etc.), a value of the form: “error: <descriptive message>” is written.

The following is an example of the records for reading information:

Example of Readings
Timestamp,07/05/01 18:00:00.009
TT-800X.FB VFD.TT-800-1.OUT.STATUS,Good_NonCascade: NonSpecific:NotLimited
TT-800-1.OUT.VALUE,27.66525
TT-800-1.MODE_BLK.ACTUAL,Auto
TT-800-1.OUT_SCALE.UNITS_INDEX,%

- If a label is specified it will be used to label the value in the file. If a label is not specified, the file will report the point name.

Retrieving Snapshot File

- Use a program or a script that you have created to fetch the snapshot data and then your program will parse the file and use the data. A control system is not required.
  a. ftp account: user = fievel, password = fievel
  b. directory: **/home/fievel/snapshot**
  c. filename: determined by snapshot setup
- To get a file manually using the ftp client on windows, perform the following steps:
  a. Open a command prompt window and cd to the directory where you want to store the file.
  b. Enter the command **ftp fim3420** (where fim3420 is the name of the 3420 you wish to connect to).
  c. At the login prompts for user and password, enter **fievel** for both.
  d. At the ftp prompt, enter **cd snapshot**.
  e. The windows client defaults to ascii mode. Enter the command **dir** at the ftp prompt to see a list of files. A program or script would already know the name of the file, so it would skip this and go right to the next step.
  f. At the ftp prompt, enter **get data.csv** (filename). This copies the data onto your PC.
  g. At the ftp prompt in your command window, enter **quit**.

Section 4 Operation and Maintenance

OVERVIEW

On power up, the FIM will determine what devices are attached to each of the fieldbus segments. If Plug and Play has been selected in the Setup each device will be interrogated and any unscheduled function blocks will be initialized. The first time the unit is powered up it is recommended that no devices be connected so that the Plug and Play settings can be configured. The FIM will poll all of the Blocks specified in the Modbus register map file or in the FTP configuration file and update the values every **x** seconds and update the values at a frequency depending on the number of function blocks being polled on each segment according to Table 3-4 on page 3-8.

DIAGNOSTICS

Select the **Diagnostics** menu tab to view the status of your fieldbus segments, fieldbus communications, Modbus communications, and the client server parameters.

Fieldbus Segment

Fieldbus segment diagnostics are provided to give you an quick view of what is attached to the Rosemount 3420. It shows how many segments are active and how many devices are attached to each segment. It also shows the current state of operating mode and plug and play.

Advanced

Client/Server

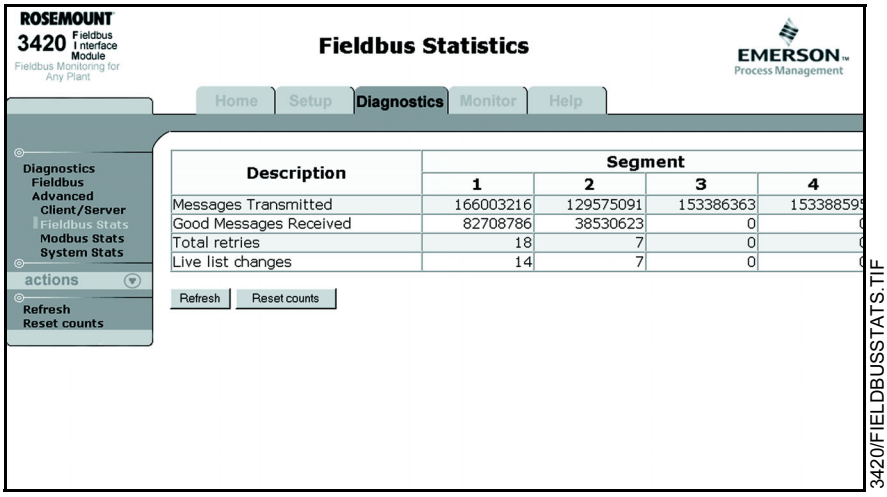
The client server diagnostics provide detailed information about the Rosemount 3420 application server and the client browser that is being used to view the 3420.

Fieldbus Communication Statistics

The fieldbus communication statistics provide information on fieldbus packets and details on the status of the internal communication link between the 3420 CPU and the host stack card (see Appendix C: Fieldbus Status Values).

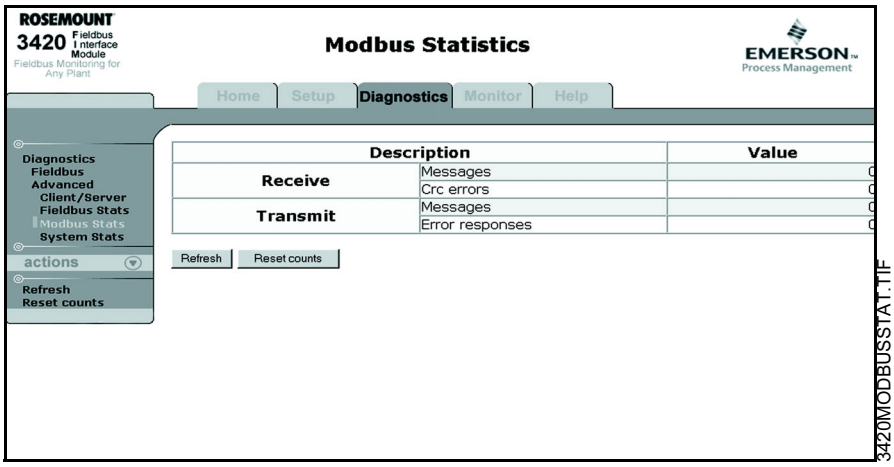
The **Messages Transmitted** and **Good Messages Received** should be incrementing steadily on segments that have devices installed. The **Total retries** should be a small number and should only increment rarely. The **Live list changes** should only change when devices are added and removed from the segment. Click **Refresh** periodically to view that the interface is communicating correctly.

Figure 4-1. Fieldbus Statistics



Modbus Communication Statistics

Figure 4-2. Fieldbus Statistics



The modbus communications statistics provide information on the data and packets received and transmitted by the modbus slave interface.

The following is a table of predefined diagnostic registers. Do not use any of these registers in the Modbus register map file (modbus.csv).

Table 4-1. Predefined Diagnostic Registers

Description	Register
Current Year (1)	49001
Current Month (1)	49002
Current Day (1)	49003
Current Hour (1)	49004
Current Minute (1)	49005
Current Second (1)	49006
Messages Received	49007
Corrupt Messages Received	49008
Messages Sent with Exception (error responses)	49009
Messages Sent Count	49010
Valid Messages Ignored (when in listen only mode)	49011

(1) Time is reported in GMT.

A Modbus host has access to digital information. The Message Sent Count is the most useful data for determining if the Modbus slave is finding errors in the messages or is rejecting the messages. The message received and messages sent count should be the same if no errors are encountered.

System Statistics

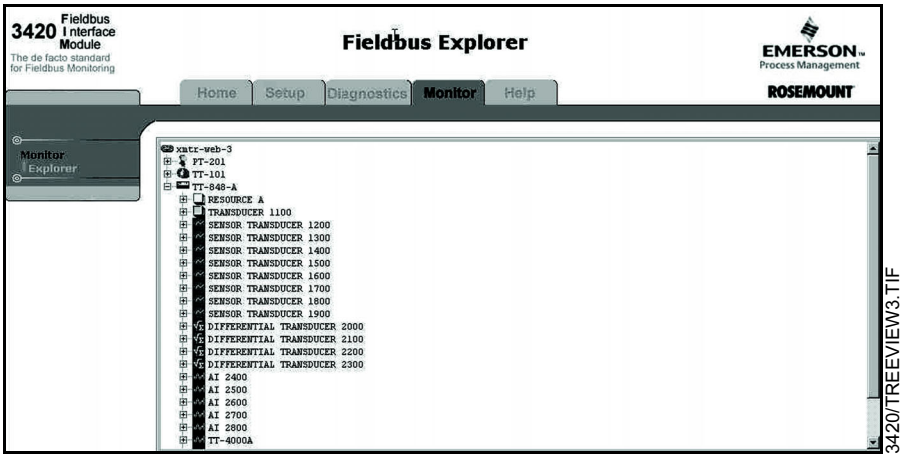
The system statistics provide an overview of various internal CPU resources that are being used by the 3420.

MONITOR

Select the **Monitor** menu tab to view data from each of the field devices. The fieldbus explorer provides a tree view of:

- fieldbus segments
- devices on a segment
- block in a device
- parameters in a block

Figure 4-3. Tree View



To change a value of a parameter select and click on the specific parameter. A dialog box will appear allowing you to either make a selection or to enter the data directly. Numerous parameters in a function block may not be changed while the block is running, in which case the MODE of the block must first be set to OUT OF SERVICE. Remember to change the MODE back to the original setting after making any changes.

Section 5 Troubleshooting

Select the **Help** menu tab to view information on product installation, setup and configuration, diagnostics, and monitoring. In addition, fieldbus, field device, and specific Rosemount 3420 facts are available.

Contact your local representative for additional service support, see “Service Support” on page 1-1.

Table 5-1. General Troubleshooting

Symptom [1]	Cause	Recommended Actions
PC does not communicate with the Rosemount 3420		• Check to see if you are using a cross-over cable • Use the PING command described in Section 3: Configuration to verify the IP address and the FIM name • Check that the browser has the proxy server turned off • Check that the IP address of the PC is set as indicated in Section 3: Configuration • Use the backup address as indicated in Section 3: Configuration Restart the computer to be sure these changes have taken effect
Fieldbus device is not detected		• Check your installation for correct wiring, grounding and that a field terminator is installed at each end of the segment.
Device does not show up on segment	Unknown	Recycle power to device
	No power to device	1. Ensure the device is connected to the segment. 2. Check voltage at terminals. There should be 9–32Vdc. 3. Check to ensure the device is drawing current. There should be approximately 17 mA.
	Segment problems	
	Electronics failing	1. Electronics board loose in housing. 2. Replace electronics.
	Incompatible network settings	Change host network parameters. Refer to host documentation for procedure.
Device does not stay on segment[2]	Incorrect signal levels. Refer to host documentation for procedure.	1. Check for two terminators. 2. Excess cable length. 3. Bad Power supply or conditioner
	Excess noise on segment. Refer to host documentation for procedure.	1. Check for incorrect grounding. 2. Check for correct shielded wire. 3. Tighten wire connections. 4. Check for corrosion or moisture on terminals. 5. Check for Bad power supply.
	Electronics failing	1. Tighten electronics board. 2. Replace electronics.
	Other	1. Check for water in the terminal housing.

(1) The corrective actions should be done with consultation of your system integrator.
(2) Wiring and installation 31.25 kbit/s, voltage mode, wire medium application guide AG-140 available from the fieldbus Foundation.

Table 5-2. Fieldbus Status Troubleshooting

Returned Value	Action
OUT.STATUS is Bad:: OutOfService	• Verify that the MODE.TARGET of the block in question is in Auto. • Verify that the MODE.TARGET of the transducer block or function block that is connected to the block in question is in Auto. • Verify that the MODE.TARGET of the resource block of the device containing the block is set to Auto. • If BLOCK_ERR is Power_up then the block is not scheduled to run and the mode can not change. • If BLOCK_ERR is OutOfService, then the block is scheduled and the input value is not good or the block mode is not set to auto.
MODE.ACTUAL=OOS, MODE.TARGET=AUTO, and BLOCK_ERR=Power_up	• The block is not scheduled. • The Rosemount 3420 will schedule AI and MAI blocks automatically but will not schedule other blocks. • Detailed FF knowledge is required to schedule a block. This situation should be rare but when encountered, Rosemount Customer Central or a Rosemount Service Center should be called for assistance.
Error Dialog on Parameter Write	Most parameters require the block be placed in OOS for the write to be accepted: • Edit the MODE.TARGET to OOS • Wait for the MODE.ACTUAL to transition to OOS Make desired parameter change: • Edit the MODE.TARGET to Auto • Wait for the MODE.ACTUAL to transition to Auto Parameter change is complete.
OUT.STATUS=Bad::ConfigurationError	For AI/MAI blocks: • LTYPE must be set correctly for the application • CHANNEL must be set to a valid transducer channel • XD_SCALE and OUT_SCALE must have valid UNITS and EU_0 and EU_100 values for the application For transducer blocks: • Sensor configuration must be valid

Appendix A Reference Data

SPECIFICATIONS

Functional Specifications

Power Input Options

20-28 V dc

500 milliamps required to power the Rosemount 3420 module.

Additional current is required to power the field devices.

The Rosemount 3420 will distribute power to the devices on the H1 segments. The optional integrated power conditioners are capable of providing up to 288 milliamps to each fieldbus segment.

Environmental

Operating Temperature Range:
-40 to 60°C (-40 to 140°F) with internal power conditioners
-40 to 70°C (-40 to 158°F) with power conditioners mounted externally

Operating Humidity Range:
0-95% relative humidity (non-condensing)

Lightning and Surge Protection

Surge protection on the Rosemount 3420 RS-485 and Ethernet communication lines meet the requirements of EN61000-4-5 category B.

RFI/EMI

Per EN610004-3 -1978 Class 2

Performance Specifications

The rate at which the individual inputs are scanned on each H1 segment depends on the number of devices and the type of Analog Input Block configured according to the following table:

Number of AI or MAI Blocks scanned on each segment	Scan rate
1	0.3 seconds
2	0.4 seconds
4	0.5 seconds
8	0.7 seconds
16	1.0 seconds
32	2.7 seconds
64	3.1 seconds
128	7.0 seconds

Example: A segment with 13 848T temperature transmitters using the MAI block will have all 104 inputs scanned every 1.0 seconds. If the individual AI blocks are scanned the update rate would be approximately 6 seconds.

Physical Specifications

Weight
10.7 lb (4.85 kg)

Material of Construction

Housing
Low-copper aluminum, NEMA 4X and IEC 529 IP65

Pollution Degree 2

Paint
Polyurethane

Cover Gasket
Rubber

Communication Specifications

RS485

2-wire communication link for Modbus multidrop connections

Baud rate: 57600, 38400, 19200, or 9600

Protocol: Modbus RTU

Wiring: Single twisted shielded pair, 18 AWG. Wiring distance is approximately 5,000 ft. (1,524 m)

Ethernet

10baseT/ 100MBS Ethernet communication port

Modbus

- Supports Modbus RTU with 32 bit floating point values and single register integers.
- Modbus registers are assigned to measurement inputs by the TAG of the analog input block.
- Modbus register numbers are specified by the user.
- The status of each variable is available in a 16 bit register.
- The configuration of the Modbus interface is accomplished using web pages generated by the Rosemount 3420.

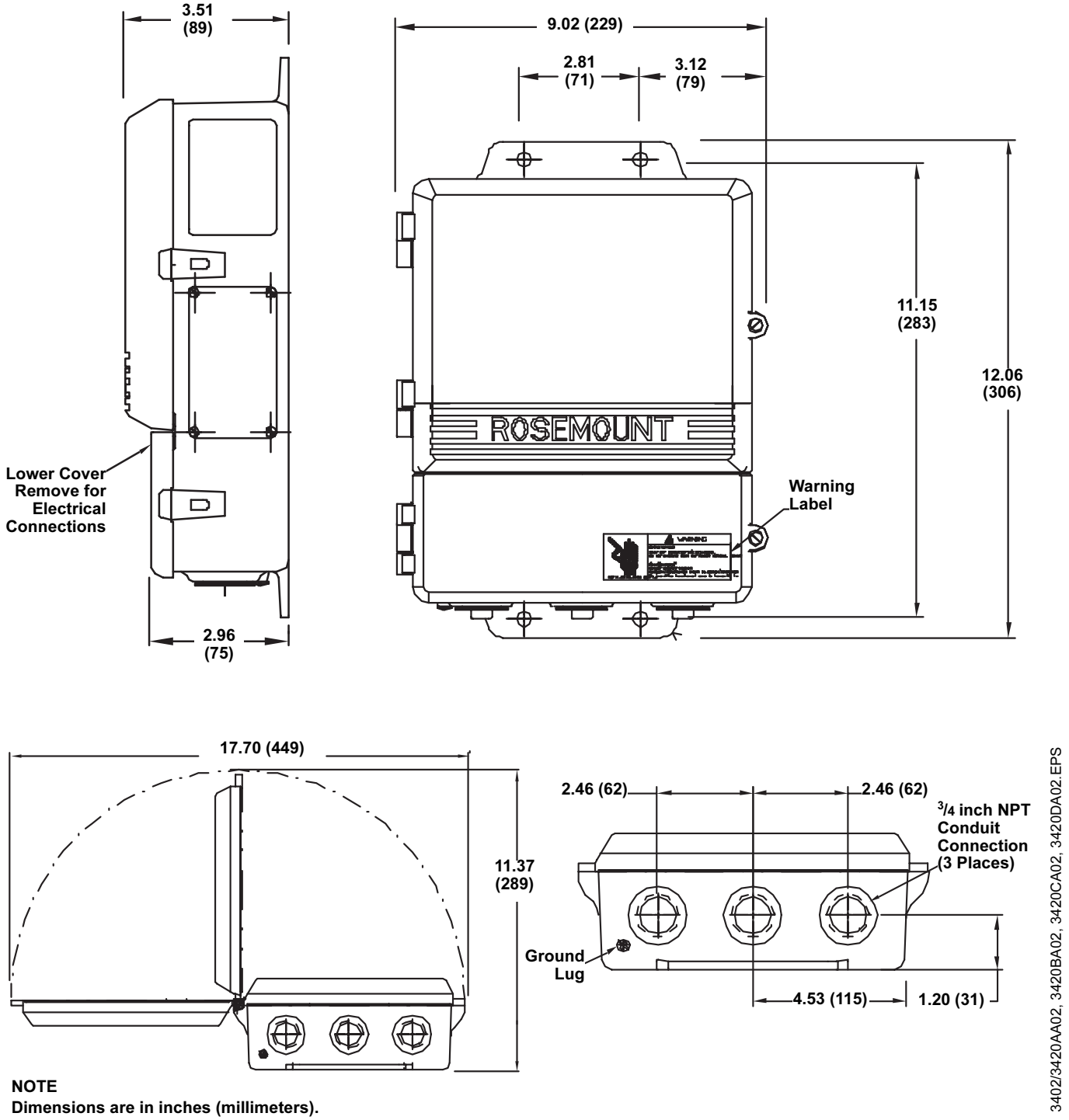
H1 Fieldbus

Up to four H1 FOUNDATION fieldbus segments are supported.
Up to 16 fieldbus devices can be connected to each H1 segment. The number of devices will depend on the power consumption of each device and the type of cable used. The 3420 internal power conditioners supply each H1 segment with 288 milliamps of current. Temperature applications using the 848T 8-input temperature transmitter exclusively, can have up to 13 848T transmitters on each segment using the internal power conditioners.

Each segment optionally includes a power conditioner and terminator. The user is required to provide a second terminator for the field end of the segment. If external power conditioners are used the user is responsible for providing both terminators for each segment and their mounting and wiring to the 3420.

DIMENSIONAL DRAWINGS

Figure A-1. Rosemount 3420
Fieldbus Interface Module



NOTE
Dimensions are in inches (millimeters).

ORDERING INFORMATION

Model	Product Description
3420	Fieldbus Interface Module
Code	**Power Input**
A	24 V dc
Code	**Output**
1	RS485 + Ethernet
Code	**Fieldbus Input**
A	One H1 Fieldbus Segment
B	Two H1 Fieldbus Segments
C	Three H1 Fieldbus Segments
D	Four H1 Fieldbus Segments
Code	**Power Conditioner + Terminator for each segment**
0	No power conditioner or terminators (must be supplied by others)
1	One power conditioner and terminator mounted in the 3420 housing
2	Two power conditioner and terminators mounted in the 3420 housing
3	Three power conditioner and terminators mounted in the 3420 housing
4	Four power conditioner and terminators mounted in the 3420 housing
Code	**RS-485 Communication Options**
N	No RS-485 Communication
A	Modbus RTU
Code	**Ethernet Communication Options**
0	Webserver
Code	**Other Options**
	Product Certifications (Pending)
N1	CENELEC Type n (ATEX)
N5	Factory Mutual (FM) Division 2 Approval (non-incendive)
N6	Canadian Standards Association (CSA) Division 2 Approval
	Adapters
J1	CM 20 Conduit Adapter
J2	PG 13.5 Conduit Adapter
Typical Model Number: 3420 A 1 A 1 N 0	

Appendix B Approval Information

PRODUCT CERTIFICATIONS (PENDING)

CE CE Marking

Compliance with European Union EMC

Factory Mutual (FM) Approval

N5 Approved for Class I, Division 2, Groups A, B, C, and D; Dust-Ignition Proof, Class II and III, Division 1, Groups E, F, and G, hazardous locations. Indoor and outdoor
(NEMA 4X). Temperature code T4.

Canadian Standards Association (CSA) Approval

N6 Approved for Class I, Division 2, Groups A, B, C, and D; Dust-Ignition Proof, Class II and III, Division 1, Groups E, F, and G, hazardous locations. Indoor and outdoor
(CSA enclosure 4). Temperature code T4A.

N1 CENELEC Type n (ATEX)

Appendix C Fieldbus Status Values

This section contains the list of possible values for the Fieldbus output STATUS along with their meaning and a possible reason the status may have this value. This is included as a convenience for interpreting the status from the Fieldbus.

Status	Description	Possible Cause
0	Bad::NonSpecific:NotLimited	• There is no specific reason why the value is bad • Used for propagation
1	Bad::NonSpecific:LowLimited	• There is no specific reason why the value is bad • Used for propagation
2	Bad::NonSpecific:HighLimited	• There is no specific reason why the value is bad • Used for propagation
3	Bad::NonSpecific:Constant	• There is no specific reason why the value is bad • Used for propagation
4	Bad::ConfigurationError:NotLimited	• Set if the value is not useful because there is some other problem with the block, depending on what a specific manufacturer can detect
5	Bad::ConfigurationError:LowLimited	• Set if the value is not useful because there is some other problem with the block, depending on what a specific manufacturer can detect
6	Bad::ConfigurationError:HighLimited	• Set if the value is not useful because there is some other problem with the block, depending on what a specific manufacturer can detect
7	Bad::ConfigurationError:Constant	• Set if the value is not useful because there is some other problem with the block, depending on what a specific manufacturer can detect
8	Bad::NotConnected:NotLimited	• Set if the value is required to be connected and is not connected
9	Bad::NotConnected:LowLimited	• Set if the value is required to be connected and is not connected
10	Bad::NotConnected:HighLimited	• Set if the value is required to be connected and is not connected
11	Bad::NotConnected:Constant	• Set if the value is required to be connected and is not connected
12	Bad::DeviceFailure:NotLimited	• Set if the source of the value is affected by a device failure
13	Bad::DeviceFailure:LowLimited	• Set if the source of the value is affected by a device failure
14	Bad::DeviceFailure:HighLimited	• Set if the source of the value is affected by a device failure
15	Bad::DeviceFailure:Constant	• Set if the source of the value is affected by a device failure
16	Bad::SensorFailure:NotLimited	• Set if the device can determine this condition • The Limits define which direction has been exceeded

Status	Description	Possible Cause
17	Bad::SensorFailure:LowLimited	• Set if the device can determine this condition • The Limits define which direction has been exceeded
18	Bad::SensorFailure:HighLimited	• Set if the device can determine this condition • The Limits define which direction has been exceeded
19	Bad::SensorFailure:Constant	• Set if the device can determine this condition • The Limits define which direction has been exceeded
20	Bad::NoComm_WithLastUsableValue:NotLimited	• Set if this value had been set by communication, which has now failed
21	Bad::NoComm_WithLastUsableValue:LowLimited	• Set if this value had been set by communication, which has now failed
22	Bad::NoComm_WithLastUsableValue:HighLimited	• Set if this value had been set by communication, which has now failed
23	Bad::NoComm_WithLastUsableValue:Constant	• Set if this value had been set by communication, which has now failed
24	Bad::NoComm_WithNoUsableValue:NotLimited	• Set if there has never been any communication with this value since it was last Out of Service
25	Bad::NoComm_WithNoUsableValue:LowLimited	• Set if there has never been any communication with this value since it was last Out of Service
26	Bad::NoComm_WithNoUsableValue:HighLimited	• Set if there has never been any communication with this value since it was last Out of Service
27	Bad::NoComm_WithNoUsableValue:Constant	• Set if there has never been any communication with this value since it was last Out of Service
28	Bad::OutOfService:NotLimited	• The value is not reliable because the block is not being evaluated, and may be under construction by a configurer • Set if the block mode is O/S
29	Bad::OutOfService:LowLimited	• The value is not reliable because the block is not being evaluated, and may be under construction by a configurer • Set if the block mode is O/S
30	Bad::OutOfService:HighLimited	• The value is not reliable because the block is not being evaluated, and may be under construction by a configurer • Set if the block mode is O/S
31	Bad::OutOfService:Constant	• The value is not reliable because the block is not being evaluated, and may be under construction by a configurer • Set if the block mode is O/S
64	Uncertain::NonSpecific:NotLimited	• There is no specific reason why the value is uncertain • Used for propagation
65	Uncertain::NonSpecific:LowLimited	• There is no specific reason why the value is uncertain • Used for propagation
66	Uncertain::NonSpecific:HighLimited	• There is no specific reason why the value is uncertain • Used for propagation
67	Uncertain::NonSpecific:Constant	• There is no specific reason why the value is uncertain • Used for propagation
68	Uncertain::LastUsableValue:NotLimited	• Whatever was writing this value has stopped doing so
69	Uncertain::LastUsableValue:LowLimited	• Whatever was writing this value has stopped doing so
70	Uncertain::LastUsableValue:HighLimited	• Whatever was writing this value has stopped doing so
71	Uncertain::LastUsableValue:Constant	• Whatever was writing this value has stopped doing so
72	Uncertain::SubstituteValue:NotLimited	• Set when the value is written when the block is not Out of Service

Status	Description	Possible Cause
73	Uncertain::SubstituteValue:LowLimited	• Set when the value is written when the block is not Out of Service
74	Uncertain::SubstituteValue:HighLimited	• Set when the value is written when the block is not Out of Service
75	Uncertain::SubstituteValue:Constant	• Set when the value is written when the block is not Out of Service
76	Uncertain::InitialValue:NotLimited	• Set when the value of an input parameter is written while the block is Out of Service
77	Uncertain::InitialValue:LowLimited	• Set when the value of an input parameter is written while the block is Out of Service
78	Uncertain::InitialValue:HighLimited	• Set when the value of an input parameter is written while the block is Out of Service
79	Uncertain::InitialValue:Constant	• Set when the value of an input parameter is written while the block is Out of Service
80	Uncertain::SensorConversionNotAccurate:NotLimited	• Set if the value is at one of the sensor limits • The Limits define which direction has been exceeded
81	Uncertain::SensorConversionNotAccurate:LowLimited	• Set if the value is at one of the sensor limits • The Limits define which direction has been exceeded
82	Uncertain::SensorConversionNotAccurate:HighLimited	• Set if the value is at one of the sensor limits • The Limits define which direction has been exceeded
83	Uncertain::SensorConversionNotAccurate:Constant	• Set if the value is at one of the sensor limits • The Limits define which direction has been exceeded
84	Uncertain::EngUnitRangeViolation:NotLimited	• Set if the value lies outside of the range of values defined for this parameter • The Limits define which direction has been exceeded
85	Uncertain::EngUnitRangeViolation:LowLimited	• Set if the value lies outside of the range of values defined for this parameter • The Limits define which direction has been exceeded.
86	Uncertain::EngUnitRangeViolation:HighLimited	• Set if the value lies outside of the range of values defined for this parameter • The Limits define which direction has been exceeded
87	Uncertain::EngUnitRangeViolation:Constant	• Set if the value lies outside of the range of values defined for this parameter • The Limits define which direction has been exceeded
88	Uncertain::Subnormal:NotLimited	• Set if a value derived from multiple values has less than the required number of Good sources
89	Uncertain::Subnormal:LowLimited	• Set if a value derived from multiple values has less than the required number of Good sources.
90	Uncertain::Subnormal:HighLimited	• Set if a value derived from multiple values has less than the required number of Good sources
91	Uncertain::Subnormal:Constant	• Set if a value derived from multiple values has less than the required number of Good sources
128	Good_NonCascade::NonSpecific:NotLimited	• There is no specific reason why the value is good • No error or special condition is associated with this value
129	Good_NonCascade::NonSpecific:NotLimited	• There is no specific reason why the value is good • No error or special condition is associated with this value
130	Good_NonCascade::NonSpecific:HighLimited	• There is no specific reason why the value is good • No error or special condition is associated with this value

Status	Description	Possible Cause
131	Good_NonCascade::NonSpecific:Constant	• There is no specific reason why the value is good • No error or special condition is associated with this value.
132	Good_NonCascade::ActiveBlockAlarm:NotLimited	• Set if the value is good and the block has an active Block Alarm
133	Good_NonCascade::ActiveBlockAlarm:LowLimited	• Set if the value is good and the block has an active Block Alarm.
134	Good_NonCascade::ActiveBlockAlarm:HighLimited	• Set if the value is good and the block has an active Block Alarm
135	Good_NonCascade::ActiveBlockAlarm:Constant	• Set if the value is good and the block has an active Block Alarm
136	Good_NonCascade::ActiveAdvisoryAlarm:NotLimited	• Set if the value is good and the block has an active Alarm with a priority less than 8
137	Good_NonCascade::ActiveAdvisoryAlarm:LowLimited	• Set if the value is good and the block has an active Alarm with a priority less than 8
138	Good_NonCascade::ActiveAdvisoryAlarm:HighLimited	• Set if the value is good and the block has an active Alarm with a priority less than 8
139	Good_NonCascade::ActiveAdvisoryAlarm:Constant	• Set if the value is good and the block has an active Alarm with a priority less than 8
140	Good_NonCascade::ActiveCriticalAlarm:NotLimited	• Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8
141	Good_NonCascade::ActiveCriticalAlarm:LowLimited	• Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8
142	Good_NonCascade::ActiveCriticalAlarm:HighLimited	• Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8
143	Good_NonCascade::ActiveCriticalAlarm:Constant	• Set if the value is good and the block has an active Alarm with a priority greater than or equal to 8
144	Good_NonCascade::UnacknowledgedBlockAlarm:NotLimited	• Set if the value is good and the block has an unacknowledged Block Alarm
145	Good_NonCascade::UnacknowledgedBlockAlarm:LowLimited	• Set if the value is good and the block has an unacknowledged Block Alarm
146	Good_NonCascade::UnacknowledgedBlockAlarm:HighLimited	• Set if the value is good and the block has an unacknowledged Block Alarm
147	Good_NonCascade::UnacknowledgedBlockAlarm:Constant	• Set if the value is good and the block has an unacknowledged Block Alarm
148	Good_NonCascade::UnacknowledgedAdvisoryAlarm:NotLimited	• Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8
149	Good_NonCascade::UnacknowledgedAdvisoryAlarm:LowLimited	• Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8
150	Good_NonCascade::UnacknowledgedAdvisoryAlarm:HighLimited	• Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8
151	Good_NonCascade::UnacknowledgedAdvisoryAlarm:Constant	• Set if the value is good and the block has an unacknowledged Alarm with a priority less than 8
152	Good_NonCascade::UnacknowledgedCriticalAlarm:NotLimited	• Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8
153	Good_NonCascade::UnacknowledgedCriticalAlarm:LowLimited	• Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8

Status	Description	Possible Cause
154	Good_NonCascade::UnacknowledgedCriticalAlarm:HighLimited	• Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8
155	Good_NonCascade::UnacknowledgedCriticalAlarm:Constant	• Set if the value is good and the block has an unacknowledged Alarm with a priority greater than or equal to 8
192	Good_Cascade::NonSpecific:NotLimited	• There is no reason why the value is good • No error or special condition is associated with this value
193	Good_Cascade::NonSpecific:LowLimited	• There is no reason why the value is good • No error or special condition is associated with this value
194	Good_Cascade::NonSpecific:HighLimited	• There is no reason why the value is good • No error or special condition is associated with this value.
195	Good_Cascade::NonSpecific:Constant	• There is no reason why the value is good. No error or special condition is associated with this value
196	Good_Cascade::InitializationAcknowledge:NotLimited	• The value is an initialized value from a source (cascade input, remote-cascade-in, and remote-output in parameters)
197	Good_Cascade::InitializationAcknowledge:LowLimited	• The value is an initialized value from a source (cascade input, remote-cascade-in, and remote-output in parameters)
198	Good_Cascade::InitializationAcknowledge:HighLimited	• The value is an initialized value from a source (cascade input, remote-cascade-in, and remote-output in parameters)
199	Good_Cascade::InitializationAcknowledge:Constant	• The value is an initialized value from a source (cascade input, remote-cascade-in, and remote-output in parameters)
200	Good_Cascade::InitializationRequest:NotLimited	• The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or in the wrong mode
201	Good_Cascade::InitializationRequest:LowLimited	• The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or in the wrong mode
202	Good_Cascade::InitializationRequest:HighLimited	• The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or in the wrong mode
203	Good_Cascade::InitializationRequest:Constant	• The value is an initialization value for a source (back calculation input parameter), because the lower loop is broken or in the wrong mode
204	Good_Cascade::NotInvited:NotLimited	• The value is from a block which does not have a target mode that would use this input
205	Good_Cascade::NotInvited:LowLimited	• The value is from a block which does not have a target mode that would use this input
206	Good_Cascade::NotInvited:HighLimited	• The value is from a block which does not have a target mode that would use this input
207	Good_Cascade::NotInvited:Constant	• The value is from a block which does not have a target mode that would use this input
208	Good_Cascade::NotSelected:NotLimited	• The value is from a Control Selector which has not selected the corresponding input • This tells the upper block to limit in one direction, not to initialize
209	Good_Cascade::NotSelected:LowLimited	• The value is from a Control Selector which has not selected the corresponding input • This tells the upper block to limit in one direction, not to initialize

Status	Description	Possible Cause
210	Good_Cascade::NotSelected:HighLimited	• The value is from a Control Selector which has not selected the corresponding input • This tells the upper block to limit in one direction, not to initialize
211	Good_Cascade::NotSelected:Constant	• The value is from a Control Selector which has not selected the corresponding input • This tells the upper block to limit in one direction, not to initialize
212	Good_Cascade::DoNotSelect:NotLimited	• The value is from a block which should not be selected by a control selector block, due to conditions in or above the block
213	Good_Cascade::DoNotSelect:LowLimited	• The value is from a block which should not be selected by a control selector block, due to conditions in or above the block
214	Good_Cascade::DoNotSelect:HighLimited	• The value is from a block which should not be selected by a control selector block, due to conditions in or above the block
215	Good_Cascade::DoNotSelect:Constant	• The value is from a block which should not be selected by a control selector block, due to conditions in or above the block
216	Good_Cascade::LocalOverride:NotLimited	• The value is from a block that has been overriden by a local key switch • This also implies Not Invited
217	Good_Cascade::LocalOverride:LowLimited	• The value is from a block that has been overriden by a local key switch • This also implies Not Invited
218	Good_Cascade::LocalOverride:HighLimited	• The value is from a block that has been overriden by a local key switch • This also implies Not Invited
219	Good_Cascade::LocalOverride:Constant	• The value is from a block that has been overriden by a local key switch • This also implies Not Invited
220	Good_Cascade::FaultStateActive:NotLimited	• The value is from a block that has FAULT-STATE active • This also implies Not Invited
221	Good_Cascade::FaultStateActive:LowLimited	• The value is from a block that has FAULT-STATE active • This also implies Not Invited
222	Good_Cascade::FaultStateActive:HighLimited	• The value is from a block that has FAULT-STATE active • This also implies Not Invited
223	Good_Cascade::FaultStateActive:Constant	• The value is from a block that has FAULT-STATE active • This also implies Not Invited
224	Good_Cascade::InitiateFaultState:NotLimited	• The value is from a block that wants its downstream output block to go to Fault State
225	Good_Cascade::InitiateFaultState:LowLimited	• The value is from a block that wants its downstream output block to go to Fault State
226	Good_Cascade::InitiateFaultState:HighLimited	• The value is from a block that wants its downstream output block to go to Fault State
227	Good_Cascade::InitiateFaultState:Constant	• The value is from a block that wants its downstream output block to go to Fault State

Glossary

NETWORKING DEFINITIONS

Term	Definitions
DHCP	Dynamic Host Configuration Protocol: Used to configure the network parameters automatically. This device contains a DHCP Client to retrieve the network configuration parameters from a DHCP server on the network.
NTP/SNTP	Network or Simple Network Time Protocol: Used to set the system time. This device contains an NTP client for keeping the system time synchronized with a network time server.
DNS	Domain Name System: A system that maps IP addresses into more meaningful strings of text called the domain.
Domain Name	A unique designator on the internet composed of symbols separated by dots such as: this.domain.com
FTP	File Transfer Protocol: A method for transferring files to and from remote computers on the network.
Host Name	A unique designator in a domain associated with the IP address of a device such as: device.this.domain.com. In that example the hostname is device.
HTML	Hyper Text Markup Language: The file format used to define pages viewed with a web browser.
HTTP	Hyper Text Transfer Protocol: The protocol that defines how a web server sends and receives data to and from a web browser.
IP	Internet Protocol: The protocol that specifies how data is transmitted over the internet.
Netmask	A string of 1's and 0's that mask out or hide the network portion of an IP address leaving only the host component.
Network	The portion of the network that the device resides on.
Broadcast	The address that a station can send to that will be received by all devices on the network.
Gateway	The address of the node on the network that serves as an entrance to other networks.

FIELDBUS DEFINITIONS

Term	Definitions
Terminator	A device attached to the end of a communications bus or network to absorb signals so they do not reflect back.
Resource Block	The resource block defines device specific characteristics such as software revisions, hardware revision and materials of construction.
Transducer Block	The transducer block defines device specific input/output characteristics for function block application purposes. It contains universal parameters and device specific parameters such as trim limits, sensor type, sensor serial number and sensor diagnostic information.
Function Block	Function blocks define the capabilities of the high level measurement and control available in the device. There are many possible function block capabilities contained in a device such as analog input, discrete input, discrete output, signal characterizer, arithmetic, integrator, PD or PID control, input selector or analog output.
Analog Input (AI) Function Block	Scanned by the Rosemount 3420.
Multiple Analog Input (MAI) Function Block	Provides up to 8 variable in one function block, reducing scan times.
Mode	**Target** - The mode requested by the operator **Actual** - The current mode of the block which may differ from the target based on the current operating conditions. The actual mode is calculated during block execution. **Permitted** - The modes allowed by target during operation **Normal** - The desired operating mode of the block.
Mode: Out of Service	In this mode the algorithm of the block is no longer being executed and any outstanding alarms will be cleared. This is the highest priority mode.
Mode: Auto	In this mode the block algorithm is being executed with new value and status as well as mode being computed each Macrocycle or execution of the block
Mode: Manual	In this mode the block output is not being calculated although it may be limited. It is set directly by the operator through an interface device.
Macrocycle	The period of execution in which the function block schedule is defined.
Schedule	The rate and relative time which function blocks process their inputs and generate their outputs. The schedule defines when a function block is triggered to start execution
Link Master	A link master device is a device that is capable of becoming the LAS. There is a bid procedure that is followed that selects the link master device with the lowest node address.
Link Active Schedule Device	The LAS performs five functions: • Maintains the schedule to send compel data (CD) messages (DLPDU's) to devices on the network. This is the highest priority function. • Sends probe node (PN) messages to unused addresses to detect new devices added to the link/segment and adds them to the Live List when found. • Periodically distributes data link time and link schedule time • Send pass token (PT) messages to devices to provide them the opportunity to send unscheduled data. • Monitor responses to pass token (PT) messages and removes devices from the Live List when a device fails to use or return the token.
Device Tag	The 32 character name of the device.
Block Tag	The 32 character name of the function block.
Parameter Name	The name of the function block parameter.
Subparameter Name	The name of the function block subparameter.

Reference Manual
00809-0100-4023, Rev AA
May 2003

Emerson Process Management

Rosemount Inc.
8200 Market Boulevard
Chanhassen, MN 55317 USA
T (U.S.) 1-800-999-9307
T (International) (952) 906-8888
F (952) 949-7001

Rosemount Temperature GmbH
Frankenstrasse 21
63791 Karlstein
Germany
T 49 (6188) 992 0
F 49 (6188) 992 112

Emerson Process Management Asia Pacific Private Limited
1 Pandan Crescent
Singapore 128461
T (65) 6777 8211
F (65) 6777 0947/65 6777 0743

www.rosemount.com

EMERSON™
Process Management