DXMR90 Series Controller

Instruction Manual

Original Instructions
225859 Rev. A
1 December 2021

225859

Contents

1 Overview 4
1.1 Models 6
1.2 Hardware Overview 6
1.3 Automation Procotols 6
1.4 Modbus Overview 7
1.5 Modbus Registers 7
1.6 Dimensions 9
2 Quick Start Guide 10
2.1 Apply Power to the Controller 10
2.1.1 Wiring 10
2.2 Configuration Instructions 10
2.2.1 DXM Configuration Software 10
2.2.2 Configuring the DXMR90 Controller 11
2.3 Mechanical Installation 15
3 Controller Connections 16
3.1 Ethernet 16
3.2 Internal Local Registers (Slave ID 199) 16
3.3 Connecting to Remote Modbus Devices 19
3.3.1 Modbus Master and Slave Ports 19
3.3.2 Set the Master and Slave Port Parameters 19
4 Working with Modbus Devices 21
4.1 Assigning Modbus Slave IDs 21
4.2 Modbus Operation 22
4.3 Modbus Communication Timeouts 22
4.4 Modbus TCP Client 22
5 Configuration Instructions 23
5.1 Scheduler 23
5.1.1 Create a Weekly Event 23
5.1.2 Create a One-Time Event 23
5.1.3 Create a Holiday Event 24
5.2 Authentication Setup 24
5.2.1 Set the Controller to use Authentication 24
5.2.2 Set the Web Services to Use Authentication 25
5.2.3 Controller Configuration Authentication 25
5.3 Register Flow and Configuration 26
5.3.1 Basic Approach to Configuration 26
5.3.2 Troubleshooting a Configuration 27
5.3.3 Saving and Loading Configuration Files 27
5.3.4 Uploading or Downloading Configuration Files 27
5.4 EtherNet/IP™ Configuration 27
5.4.1 Configuring the Host PLC 27
5.4.2 Configuring the Controller 27
5.5 Set up the Email 28
5.5.1 Mail Server Authentication 28
5.5.2 Define the Network Interface Settings 29
5.5.3 Configure your Ethernet Connection 29
5.5.4 Set the Email Parameters 30
5.5.5 Define Threshold Rules for Email 31
5.5.6 Define Log File Parameters for Emailing Log Files 31
5.6 Ethernet Push Retries 31
5.6.1 Ethernet Push Retries 32
5.6.2 Event/Action Rule or Log File Push Retries 32
5.6.3 Email Message Push Retries 32
6 PROFINET® 33
6.1 General Station Description Markup Language File 33
6.2 DXM PROFINET IO Data Model 33
6.3 Configure the DXM Controller for a PROFINET IO Connection 33
6.3.1 Save and Upload the Configuration File 33
6.4 Slots and Modules 34
6.5 Configuration Instructions 35
6.5.1 Install the GSD File 35
6.5.2 Change the Device IP Address 37
6.5.3 Change the Device Name 38

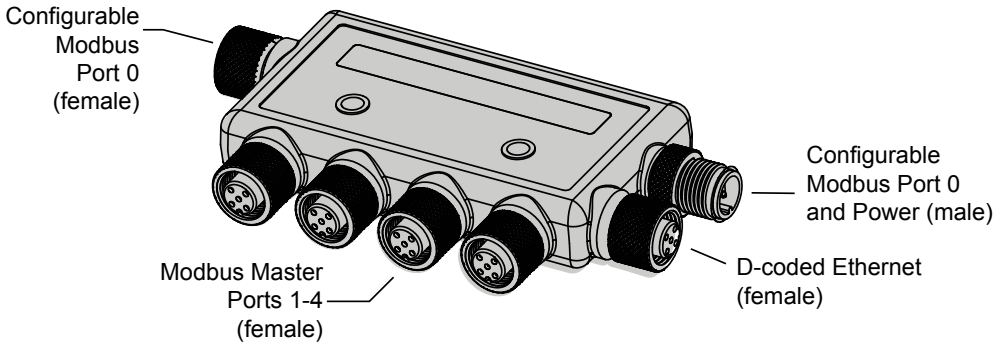
7 Accessories 40
8 Product Support and Maintenance 42
8.1 Specifications 42
8.2 File System and Archive Process 42
8.3 Update Your DXM Processor Firmware Using the Configuration Software 43
8.4 DXMR90 Support Policy 44
8.4.1 Firmware Updates 44
8.4.2 Website Information 44
8.4.3 Feature Requests 44
8.4.4 Potential DXM Issues 44
8.4.5 DXM Security 44
8.5 Contact Us 44
8.6 Warnings 45
8.7 Banner Engineering Corp Limited Warranty 45

1 Overview

Banner's DXMR90 Series Controller consolidates data from multiple sources to provide local data processing as well as accessibility for host systems as a platform for the Industrial Internet of Things (IIoT).

The DXMR90 contains four individual Modbus masters allowing for concurrent communication to up to four independent networks. Data is collected into the internal logic controller to facilitate edge processing, protocol conversion to Industrial Ethernet, and pushing information to web servers.

Figure 1. Overview of the DXMR90 Series Controller



One male M12 connection provides common power and ground to all M12 Modbus ports. The two port 0 Modbus connections can be configured as pass-through wiring to connect to a Modbus trunk. One 100 Mbps Ethernet port (female) using an M12 D-coded Ethernet connection.

- Modbus TCP
- EtherNet /IP
- Profinet
- Configuration/discovery port

Four Modbus master connections using female M12 connectors.

- 2-wire RS-485 physical transceiver with power/ground at each connector
- Separate Modbus master control and programmability for each connection point
- Independent and selectable baud rate and parity settings
- Individual timing and packet timing for each Modbus connection

Logic Controller

Program the DXMR90's logic controller using action rules and/or ScriptBasic or MicroPython programming languages, which can execute concurrently. The control functions allow freedom when creating custom sensing and control sequences. The logic controller supports the Modbus protocol standards for data management, ensuring seamless integration with existing automation systems. File password protection is an option.

Action Rules

- Thresholds (IF/THEN/ELSE) with timers, minimum on/off time, and logging options
- Math/Logic Rules (arithmetic and bitwise operators)
- Control Logic (logical operators and SR/T/D/JK flip flops)
- Trending (multiple averaging filters)
- Tracking (counts, on/off times)
- Email notifications
- Push data on conditions

Programming Language—ScriptBasic to create variables, arrays, functions, loops, IF/THEN/ELSE, logical and arithmetic operators, API commands, register access, string functions and operators, time commands

Scheduler

- Time/calendar-based events
- Holiday skips
- One-time events
- Dynamic scheduler updating
- Astronomical clock

Push to the Cloud

Data Logging

- Cyclic data/event logging
- Email log files

Email

Register Mapping

- Cyclical Read rules from wireless devices or local wired Modbus devices that include optional scaling, error conditions, and the ability to activate a read rule
- Cyclical or Change of State Write rules to local wired Modbus devices with scaling
- Modbus/TCP Master Read or Write rules for external devices on the network

Wired Connectivity

Ethernet: Modbus/TCP (master/slave) or Ethernet/IP

Field Bus: Modbus RS-485 Master/Slave

User Interface

API Interface—Host Initiated control and Web service integration

User-defined LED indicators—The DXMR90 has six user-configurable LED indicators to indicate the status of the DXMR90, processes, or equipment

Table 1: Modbus registers for internal local registers (Modbus slave ID 199)

Local Registers	Type	Description
1–845	32-bit integer	Local data registers
846–849	32-bit integer	Reset, Constant, Timer
851–900	32-bit non-volatile integer	Data flash, non-volatile
901–1000		Reserved for internal use
1001–5000	Floating point	Floating point registers, local data registers
5001–7000	32-bit integer	Local data registers
7001–8000	32-bit non-volatile integer	Data flash, non-volatile
> 10000		Read only virtual registers, system-level data

1.1 Models

Model	Ethernet Connection	Modbus Master Connections	Other Connections
DXMR90-X1	One female M12 D-Code Ethernet Connector	Four female M12 connections for Modbus master connections	One male M12 (Port 0) for incoming power and Modbus RS-485, one female M12 for daisy chaining Port 0 signals.

1.2 Hardware Overview

The DXMR90 Series Controller can have multiple configurations. The DXMR90 will have a model number label on the housing. Use the model number to identify which boards are included in the controller.

1.3 Automation Procotols

The DXMR90 Series Controller supports the following automation protocols.

EtherNet/IP™

By default, EtherNet/IP is disabled. Configure the DXMR90 Local Registers as EtherNet/IP input or output registers using the DXM Configuration Software. A single register can only be set as either an EtherNet/IP input or output register.

EtherNet/IP registers are limited to 228 registers set as **E/IP Originator to DXM** and 228 registers set as **DXM to Originator**

Modbus RTU

The DXMR90 manages five separate physical ports running the Modbus RTU protocol. The DXMR90 is the Modbus Master when operating the Modbus master RTU port (port 1–4). The DXMR90 uses the master Modbus RTU bus to communicate with locally connected Modbus slave devices.

The other Modbus RTU port (port 0) is used by a host system to access the DXMR90 as a slave device. The slave Modbus RTU port allows access all the internal local registers concurrently with the master RTU port. Port 0 can be configured as a Modbus Master Port using the DXM Configuration Software but is defined as a slave port by default.

Configure the port parameters using the DXM Configuration Software.

Modbus TCP/IP

A host system acting as a Modbus master (Client) can access the DXMR90 using the Modbus TCP/IP protocol over Ethernet. Standard Modbus TCP port 502 is used by the DXMR90 for all Modbus TCP/IP requests.

All internal local registers are available to the host system concurrently with Modbus TCP.

By default, the DXMR90 is configured as a Modbus TCP/IP Server. To configure the DXMR90 as a Modbus TCP Client, Modbus TCP must be enabled in the DXM Configuration Software and sockets must be defined to point the DXMR90 to up to 5 Servers.

Profinet

By default, Profinet is disabled on the DXMR90. To configure the DXMR90 for Profinet communications, Profinet must be enabled using the DXM Configuration Software. The DXMR90 uses fixed Slot sizes and locations in the Local Registers for the Input and Output values.

Module sizes supported are 64, 128, 256, and 512 bytes which range from 32 to up to 256 Local Registers in the DXMR90.

1.4 Modbus Overview

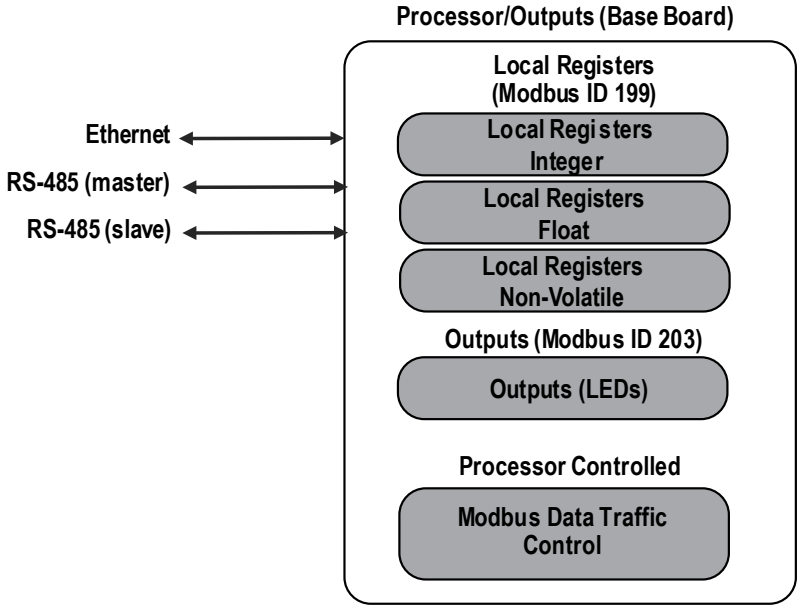
The DXMR90 Series Controller uses internal 32-bit registers to store information. The processor's internal Local Registers serve as the main global pool of registers and are used as the common data exchange mechanism. External Modbus device registers can be read into the Local Registers or written from the local data registers.

The DXMR90, as a Modbus master device or slave device, exchanges data using the Local Registers. Modbus over Ethernet (Modbus/TCP) uses the Local Registers as the accessible register data.

Using Action, Read/Write, and Threshold Rules allows you to manipulate the processor's Local Registers. The MicroPython or ScriptBasic programming capabilities extends the use of Local Registers with variables to create a flexible programming solution for more complex applications.

The processor's Local Registers are divided into three different types: integer, floating point, and non-volatile. When using Local Registers internally, the user can store 32-bit numbers. Using Local Registers with external Modbus devices follows the Modbus standard of a 16-bit holding register. Local Registers are accessible as Modbus ID 199 when using ScriptBasic or MicroPython.

Figure 2. DXMR90 Modbus overview



1.5 Modbus Registers

The DXMR90 Series Controller may have up to two internal Modbus slave addresses:

Table 2: Internal Modbus slave IDs (factory default)

Modbus Slave ID	Device
199	Local Registers—Internal storage registers
203	LED indicators

All Modbus registers are defined as 16-bit Modbus Holding Registers. The local register slave ID (199) is fixed for access via ScriptBasic or MicroPython. When accessing the Local Registers through an external Modbus RTU Master, the Slave Port (Port 0) slave ID can be changed using the DXM Configuration Software. Connected devices can use any Modbus Slave ID. For a complete list of registers, see Internal Local Registers (Slave ID 199) on page 16.

Table 3: Modbus registers for internal local registers (Modbus slave ID 199)

Local Registers	Type	Description
1–845	32-bit integer	Local data registers
846–849	32-bit integer	Reset
851–900	32-bit non-volatile integer	Data flash, non-volatile
901–1000		Reserved for internal use
1001–5000	Floating point	Floating point registers, local data registers
5001–7000	32-bit integer	Local data registers
7001–8000	32-bit non-volatile integer	Data flash, non-volatile
> 10000		Read only virtual registers, system-level data

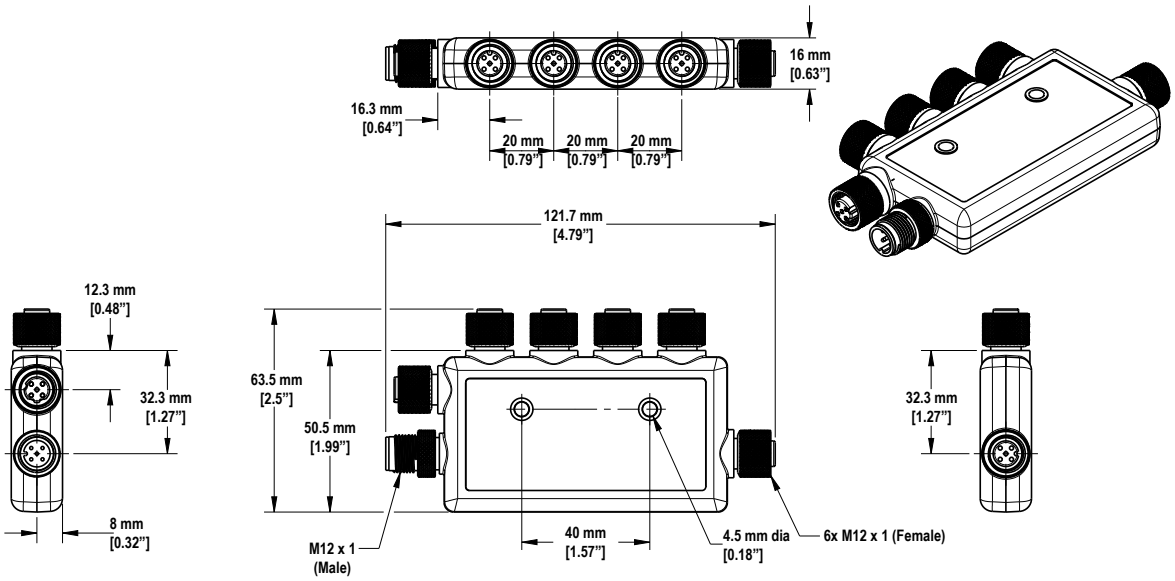
Table 4: Modbus registers for the LCD board (Modbus slave ID 203)

Modbus Register	LED	Color	State
2101: bit 0	LED 1	Green	1 = On 0 = Off
2102 : bit 0	LED 2	Red	
2103 : bit 0	LED 3	Amber	
2104 : bit 0	LED 4	Amber	
2105: bit 0	LED 5	Red	
2106: bit 0	LED 6	Green	

1.6 Dimensions

All measurements are listed in millimeters, unless noted otherwise.

Figure 3. R90x1 dimensions



2 Quick Start Guide

2.1 Apply Power to the Controller

Follow these instructions to apply 12–30 V DC power to the DXMR90 using a wall plug.

Required equipment:

- DXMR90 Series Controller
- **PSW-24-1** Wall plug power supply; 24 V DC, 1 A

1. Connect the **PSW-24-1** power supply to the male M12 connector on the DXMR90, Port 0.
2. Plug in the **PSW-24-1** wall plug power supply.

2.1.1 Wiring

Table 5: Ports 0-4 female connector

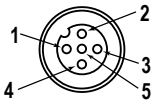
Port 0–4 5-pin M12 Connector (female)	Pin	Wire Color	Description
	1	Brown (bn)	12 V DC to 30 V DC
	2	White (wh)	RS485 / D1 / B / +
	3	Blue (bu)	DC common (GND)
	4	Black (bk)	RS485 / D0 / A / -
	5	Gray (gy)	Not used/reserved

Table 6: Port 0 male connector

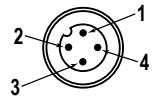
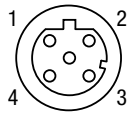
Port 0 4-pin M12 Connector (male)	Pin	Wire Color	Description
	1	Brown (bn)	12 V DC to 30 V DC
	2	White (wh)	RS485 / D1 / B / +
	3	Blue (bu)	DC common (GND)
	4	Black (bk)	RS485 / D0 / A / -

Table 7: D-coded industrial Ethernet connector

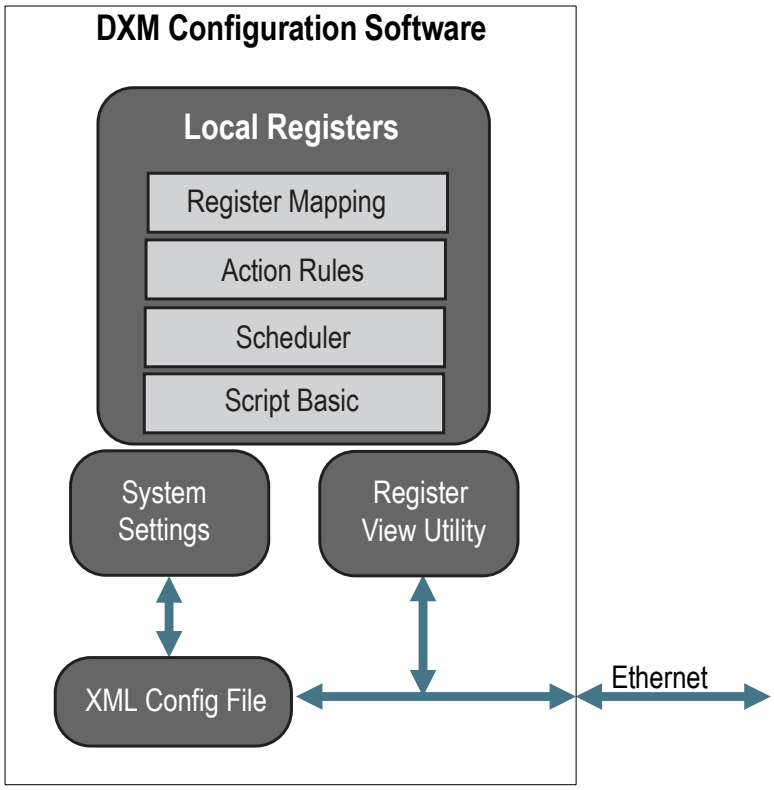
4-pin Industrial Ethernet Connector (female)	Pin	Wire Color	Description
	1	Black (bk)	+Tx
	2	Red (rd)	+Rx
	3	Green (gn)	–Tx
	4	White (wh)	–Rx

2.2 Configuration Instructions

2.2.1 DXM Configuration Software

Configure the DXMR90 using the configuration software. Download the latest version of all configuration software from http://www.bannerengineering.com. For more information on using the DXM Configuration Software, refer to the instruction manual (p/n 209933).

Figure 4. Overview of the configuration software features



The configuration software creates an XML file that is transferred to the DXM using an Ethernet connection. The DXM can also receive the XML configuration file from a Web server using an Ethernet connection. This configuration file governs all aspects of the DXM operation. The DXM Configuration Software allows the user to define parameters for the DXMR90, then saves the configuration in an XML file on the PC.

After the configuration file is saved, upload the XML configuration file to the DXMR90 for operation.

This quick start guide outlines the basic operations to set up a DXMR90 using the configuration software. For a more comprehensive explanation of features, refer to the DXM Configuration Software Instruction Manual (p/n 209933).

2.2.2 Configuring the DXMR90 Controller

This section will walk you through the method of setting up the DXM Configuration Software and communicating with a connected DXM device. Version 4 of the DXM Configuration Software supports multiple DXM device models, each of which incorporates different features.

As soon as a DXM model is connected to your computer, the software automatically detects the correct model and loads the appropriate screens. You may also manually select which model of DXM you are configuring if you intend to create a configuration file without connecting a device. This ensures that the interface and the configuration file use the correct features.

Not all screens are available for all models. To change to another model of DXM, go to the Select Mode screen and use the drop-down list to select another model. If the active configuration is incompatible with the selected model, you will be prompted to either proceed and wipe out the active configuration or cancel the model change and preserve the configuration.

Figure 5. Opening splash screen for Traditional Setup mode with an R90X1 selected

When the **Select DXM Model** drop-down is set to DXMR90, a new network discovery table is displayed. Click **Scan Network for DXMs** to detect DXM devices on the host computer's network. Discovered DXMs are listed in the network discovery table. Double-click any row entry to connect to that DXM. If the DXM's IP address is already known, the standard TCP connection option is available below the network discovery table.

Important: Any model of DXM may connect to the configuration software regardless of which device model is selected in the tool. Compatibility is checked before configuration files are uploaded to the device.

Configuration Example: Reading Registers on a Modbus Slave Device

The local registers are the main global pool of registers that are defined by the user to store data within the DXM. The local registers are listed on the **Local Registers** > **Local Registers in Use** screen.

The bottom status bar displays the communications status, application status, and the DXM Configuration Software version.

In this short example, we will configure the DXM to read six registers on an external Modbus Slave device and save the data into the local registers.

Important: The software only loads a file to the DXM. Internal parameter settings that are changed in the tool but not saved to the file will not be sent to the device.

Modify Multiple Registers

Modify a range of registers from the **Local Registers** > **Local Registers in Use** > **Modify Multiple Registers** screen.

Select which parameter fields to modify. Most parameters have three selections.

- Unchanged—no changes
- Default—change to default settings
- Set—modify parameter. Other selections will appear based on the parameter.

Figure 6. Modify Multiple Registers screen

1. Enter the **Starting register** and **Ending register**.
2. Select the value to change using the drop-down list next to each value.
3. Enter the new value in the field provided.
4. To push register values to the web server, set **Cloud Permissions** to read.

   If the **Cloud Permissions** are set to Read, the web server only views data from the device and cannot write data to the device. If the permissions are set to Write, the web server only writes to the device and cannot read the data. If the permissions are set to Read/Write, the web server can read the data from the device and write to the device from the web.

5. Click **Modify Registers** to save and apply the changes.

Define an RTU Read Rule

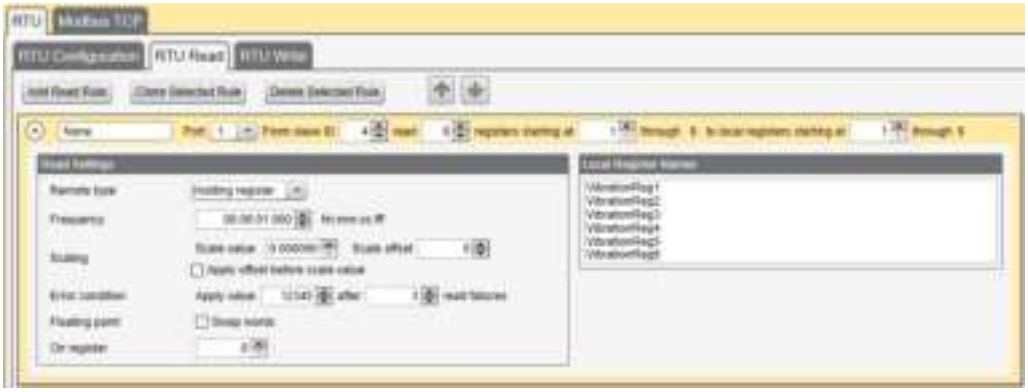
Follow these steps to create a new read rule. This example creates a read rule to read six registers (address 1 through 6), from Port 1 Modbus Slave 4. The results are stored in the Local Registers 1 through 6.

1. Define the **Port** settings to be compatible with the connected devices.
   a) Go to the **Register Mapping** > **RTU** > **RTU Configuration** screen.

Figure 7. RTU Configuration screen

   b) Go to the **Register Mapping** > **RTU** > **RTU Configuration** screen.
   c) Modify the **Port** settings as needed.
   - Verify the **Baud Rate** and **Parity** match that of the connected Modbus slave devices.
   - The **Timeout** controls how long the DXMR90 waits before determining a command failed to send. Set based on the specific application requirements.
   - The **Delay between messages** defines the minimum wait time between resending another command. Set based on the specific application requirements.
2. From the **Register Mapping** > **RTU** > **RTU Read** screen, click **Add Read Rule**.
3. Click the arrow next to the name to display the parameters.
4. Name your rule.
5. Select the Port number to which the device is connected.
6. Select the slave ID of the device.
7. Select how many registers to read, and the beginning register.
8. Define the register type, how often to read the register, and any other appropriate parameters.
9. If necessary, select the error condition. For this example, if the read function fails after three attempts, the read rule writes 12345 to the DXM local registers. Notice the list of local register names this read rule is using.

Figure 8. Read Rules - Configuration Example



Baud Rate

Defined for both the Modbus master and slave

Settings include: 19200 (default), 1200, 2400, 9600, 38400, 57600, and 115200.

Delay between messages

Applies to the Modbus master port

Sets the minimum wait time from the end of a Modbus transaction to the beginning of the next Modbus transaction.

Parity

Defined for both the Modbus master and slave

Settings include: None (default), odd, even, space, and mark

Timeout

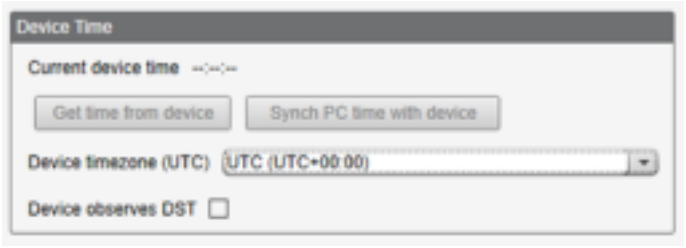
Applies to the Modbus master port

Covers the expected time for messages to be sent throughout the wireless network. For the DXM, the **Timeout** parameter is the maximum amount of time the DXM should wait after a request is sent until the response message is received from the Modbus slave device.

Set the Time

Use the **Settings** > **System** screen to define the time zone and daylight saving option. The time zone and DST options are saved into the configuration file.

Figure 9. Settings > System > Device Time



1. Go to the **Settings** > **System** screen.
2. If you connect the DXM to a computer, click **Sync PC Time with Device** to set the time on the DXM to match the time of the computer.
3. Set your time zone and select whether or not your device observes daylight saving time (DST).

Set the IP Address

Follow these instructions to change the DXMR90's IP address.

By default, the DXMR90 is set to a static IP address of 192.168.0.1. The IP address can be changed by using the DXM Configuration Software and updating the XML.

1. Launch the DXM Configuration Software.
2. Go to the **Settings** > **Ethernet** screen.
3. In the **IP Address** section, select **Static IP** or **DHCP** from the drop-down list.
   - If **DHCP** is selected, the **IP address**, **Subnet**, and **Gateway address** are grayed out and not configurable.
   - If **Static IP** is selected, enter the **IP address**, **Subnet**, and **Gateway address** as desired.
4. Save your changes to the configuration file (**File** > **Save**).
5. Upload the configuration file to your controller (**DXM** > **Send Configuration to DXM**).

Save and Upload the Configuration File

After making any changes to the configuration, you must save the configuration files to your computer, then upload it to the device.

Changes to the XML file are not automatically saved. Save your configuration file before exiting the tool and before sending the XML file to the device to avoid losing data. If you select **DXM** > **Send XML Configuration to DXM** before saving the configuration file, the software will prompt you to choose between saving the file or continuing without saving the file.

1. Save the XML configuration file to your hard drive by going to the **File** > **Save As** menu.
2. Go to the **DXM** > **Send XML Configuration to DXM** menu.

Figure 10. Status indicator bar

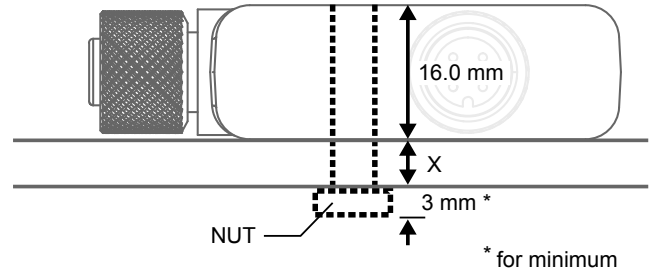
- If the Application Status indicator is red, close and restart the DXM Configuration Tool, unplug and re-plug in the cable and reconnect the DXM to the software.
- If the Application Status indicator is green, the file upload is complete.
- If the Application Status indicator is yellow, the file transfer is in progress.

The device reboots and begins running the new configuration.

2.3 Mechanical Installation

Install the DXMR90 to allow access for functional checks, maintenance, and service or replacement.

All mounting hardware is supplied by the user. Fasteners must be of sufficient strength to guard against breakage. Use of permanent fasteners or locking hardware is recommended to prevent the loosening or displacement of the device. The mounting hole (4.5 mm) in the DXMR90 accepts M4 (#8) hardware. See the figure below to help in determining the minimum screw length.

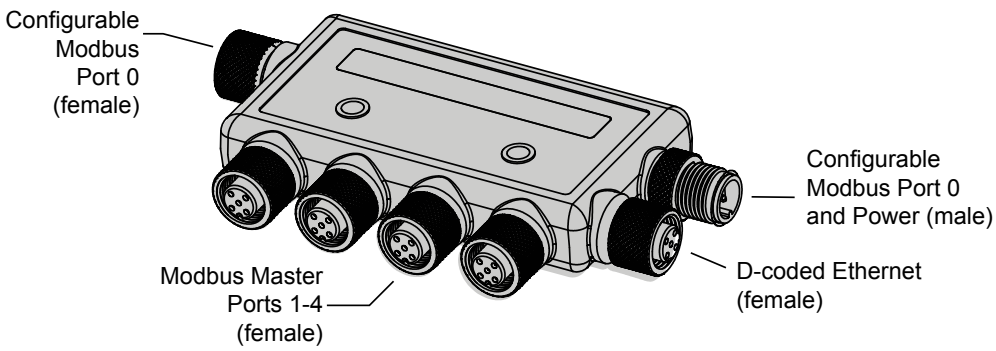


Screw Length = 16.0 mm + “X” mm + 3 mm

CAUTION: Do not overtighten the DXMR90's mounting screw during installation. Overtightening can affect the performance of the DXMR90.

3 Controller Connections

Figure 11. Connections for the R90X1



3.1 Ethernet

Before applying power to the DXMR90, verify the Ethernet cable is connected.

The Ethernet connection supports the DXM Configuration Software, Modbus/TCP, and EtherNet/IP. ScriptBasic also has access to Ethernet for custom programming. Use the software or LCD menu system to configure the characteristics of the Ethernet connection, including the IP address. Any parameters not changeable from the menu system are configurable from the configuration software.

3.2 Internal Local Registers (Slave ID 199)

The main storage elements for the DXMR90 are its Local Registers, which can store 4-byte values that result from register mapping, action rules, MicroPython, or ScriptBasic commands.

Local Registers updated from Modbus transactions are restricted to a16-bit data value to follow standard Modbus Holding Register definition.

The Local Registers defined in Action Rules must all be within the same register group. For example, an Action Rule cannot have inputs from an integer group with the result register defined as a floating point register. To move between integers and floats, use the Register Copy Rule.

Table 8: Modbus registers for internal local registers (Modbus slave ID 199)

Local Registers	Type	Description
1–845	32-bit integer	Local data registers
846–849	32-bit integer	Reset
851–900	32-bit non-volatile integer	Data flash, non-volatile
901–1000		Reserved for internal use
1001–5000	Floating point	Floating point registers, local data registers
5001–7000	32-bit integer	Local data registers
7001–8000	32-bit non-volatile integer	Data flash, non-volatile
> 10000		Read only virtual registers, system-level data

Local Registers 1–845 and 5001–7000 (Internal Processor Memory, 32-bit, Unsigned)—The Local Registers are the main global pool of registers. Local Registers are used as basic storage registers and as the common data exchange mechanism. External Modbus device registers can be read into the Local Registers or written from the Local Registers. The DXMR90, as a Modbus master device or a Modbus slave device, exchanges data using the Local Registers. Modbus over Ethernet (Modbus/TCP) uses the Local Registers as the accessible register data.

Local Registers 846–849 (Reset, Unsigned)—These Local registers are reserved for use as Reset registers. A time interval can be specified in the configuration software for the DXM to reset. If the data in the register does not change within the user-specified time interval, the DXM resets.

Local Registers 851–900 and 7001–8000 (Data Flash, Non-volatile, 32-bit, Unsigned)—The top 50 Local Registers are special non-volatile registers. The registers can store constants or calibration type data that must be maintained when power is turned off. This register data is stored in a data flash component that has a limited write capability of 100,000 cycles, so these registers should not be used as common memory registers that change frequently.

Local Registers 1001–5000—These Local Registers are paired together to store a 32-bit IEEE floating point format number in big endian format. Registers 1001 [31:16], 1002 [15:0] store the first floating point value; registers 1003, 1004 store the second floating point number. There are a total of 2000 floating point values; they are addressed as two 16-bit pieces to accommodate the Modbus protocol. Use these registers when reading/writing external devices that require Modbus registers in floating point format. Since Modbus transactions are 16-bits, the protocol requires two registers to form a 32-bit floating point number.

Virtual Registers—The DXMR90 has a small pool of virtual registers that show internal variables of the main processor. Some register values will be dependent upon the configuration settings of the DXMR90. Do not use Read Rules to move Virtual Local Registers data into Local Registers. Use the Action Rule > Register Copy function to move Virtual Local Registers into Local Registers space (1-850).

Table 9: Modbus registers for virtual registers

Registers	Definition	
10001	GPS latitude direction (N, S, E, W)	GPS Coordinate Data if the DXM is configured to read an external GPS unit.
10002	GPS latitude	
10003	GPS longitude direction (N, S, E, W)	
10004	GPS longitude	
10011–10012	Resync timer	Engineering use
10013–10014	Resync timer rollover	Engineering use
10015–10016	Reboot cause (Restart Codes above)	Reboot Type
10017–10018	Watchdog reset count	Counter to track how many resets have been caused by the Watchdog
10025–10026	Http Push SSL Acquires	Statistical counts of connections, disconnections and forced disconnects when the DXMR90 creates a connection using SSL/TLS (Encrypted connections)
10027–10028	Http Push SSL Releases	
10029–10030	Http Push SSL Forced Releases	
10031–10032	Http Push Attempts	Statistical counts of connections, disconnections and forced disconnects when the DXM controller creates a connection using HTTP non-encrypted
10033–10034	Http Push Successes	
10035–10036	Http Push Failures	
10037–10038	Http Push Last Status	Last DXMR90 push status 0 = Initial state, no push attempt as finished yet 1 = Attempt complete 2 = Attempt aborted
10055–10056	Alarms, smtp, attempts	Email attempts
10057–10058	Alarms, smtp, fails	Email failures
10100	Number of read maps in default	Read Map statistics
10101	Number of read map successes	
10102	Number of read map timeouts	
10103	Number of read map errors	
10104	Read map success streak	
10105	Number of write map successes	Write Map statistics
10106	Number of write map timeouts	
10107	Number of write map errors	
10108	Write map success streak	
10109	Number of passthrough successes	API message passing statistics
10110	Number of passthrough timeouts	

Registers	Definition	
10111	Number of passthrough errors	
10112	Passthrough success streak	
11000	Read map success count	Read/Write maps statistics
12000	Write map success count	
13000	Read map timeout count	
14000	Write map timeout count	
15000	Read map error count	
16000	Write map error count	
17000	Read map success streak	
18000	Write map success streak	
19000	Read map is in default	

TCP Client Stats—The "x" represents the socket 0 through 4. The flex socket is not used. This range repeats for the next socket.

Table 10: TCP client statistics

Register	Definition
2x001	Socket x connection attempts (20001 is the first socket, 21001 is the second socket...)
2x003	Socket x connections
2x005	Socket x disconnections
2x007	Socket x transmits
2x009	Socket x receives
2x011	Socket x resolver attempts (reserved)
2x013	Socket x resolvers (reserved)
2x015–2x020	Reserved
2x021	Socket x Rule 0 transmits
2x023	Socket x Rule 0 receives
2x025	Socket x Rule 0 timeouts
2x027	Socket x Rule 0 broadcasts
2x029	Reserved
2x031	Socket x Rule 1 transmits
2x033	Socket x Rule 1 receives
2x035	Socket x Rule 1 timeouts
2x037	Socket x Rule 1 broadcasts
2x039	Reserved

Reset Codes—The reset codes are in virtual register 11015 and define the condition of the last restart operation.

Table 11: Reset codes

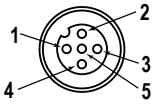
Reset Code	Definition
0	Undefined
1	Unknown
2	General
3	Brownout

Reset Code	Definition
4	Watchdog
5	User
6	Software
7	Return from backup mode

3.3 Connecting to Remote Modbus Devices

The DXMR90 is configured with four independent Modbus Master Ports, all ports use a 4-pin M12 female connector to connect to remote devices. No additional wiring is required if the sensors use compatible wiring.

Table 12: Ports 0-4 female connector

Port 0–4 5-pin M12 Connector (female)	Pin	Wire Color	Description
	1	Brown (bn)	12 V DC to 30 V DC
	2	White (wh)	RS485 / D1 / B / +
	3	Blue (bu)	DC common (GND)
	4	Black (bk)	RS485 / D0 / A / -
	5	Gray (gy)	Not used/reserved

3.3.1 Modbus Master and Slave Ports

The DXMR90 can be a Modbus RTU master device to other slave devices and can be a Modbus slave device to another Modbus RTU master. The DXM uses the ports 1–4 as Modbus RTU master ports to control external slave devices. All wired devices connected to the master RS-485 port must be slave devices.

- As a Modbus RTU master device, the DXMR90 controls external slaves connected to ports 1–4
- As a Modbus RTU slave device, the DXMR90 local registers can be read from or written to by another Modbus RTU master device via port 0.

The Modbus RTU slave connection, port 0, is controlled by another Modbus master device that is not the DXMR90. The slave port is used by an external Modbus master device that will access the DXMR90 as a Modbus slave Device. Use the DXM Configuration Software to define the operational settings for both the Modbus RTU master ports 1–4 and the Modbus RTU slave port 0.

3.3.2 Set the Master and Slave Port Parameters

The basic communications parameters for the RS-485 ports are set in the DXM Configuration Software and are saved in the XML configuration file. Each port can have unique settings such as a unique baud rate, parity, timeout, and delays between messages.

Figure 12. RTU Configuration screen for ports 0-4

1. Define the **Port** settings to be compatible with the connected devices.
   a) Go to the **Register Mapping** > **RTU** > **RTU Configuration** screen.

Figure 13. RTU Configuration screen

   b) Go to the **Register Mapping** > **RTU** > **RTU Configuration** screen.
   c) Modify the **Port** settings as needed.
   - Verify the **Baud Rate** and **Parity** match that of the connected Modbus slave devices.
   - The **Timeout** controls how long the DXMR90 waits before determining a command failed to send. Set based on the specific application requirements.
   - The **Delay between messages** defines the minimum wait time between resending another command. Set based on the specific application requirements.
2. To set the Modbus Slave parameters for Port 0, go to **Settings** > **System** > **Slave Port 0 Settings**.
3. Modify the Baud Rate, Parity, and change the Internal Slave ID. (For a description of the parameters, see Define an RTU Read Rule on page 13.)

   The Internal Slave ID is the Modbus ID that an external Modbus Master will access to read/write to the local registers on the DXMR90.

Figure 14. Slave Port 0 Settings

4 Working with Modbus Devices

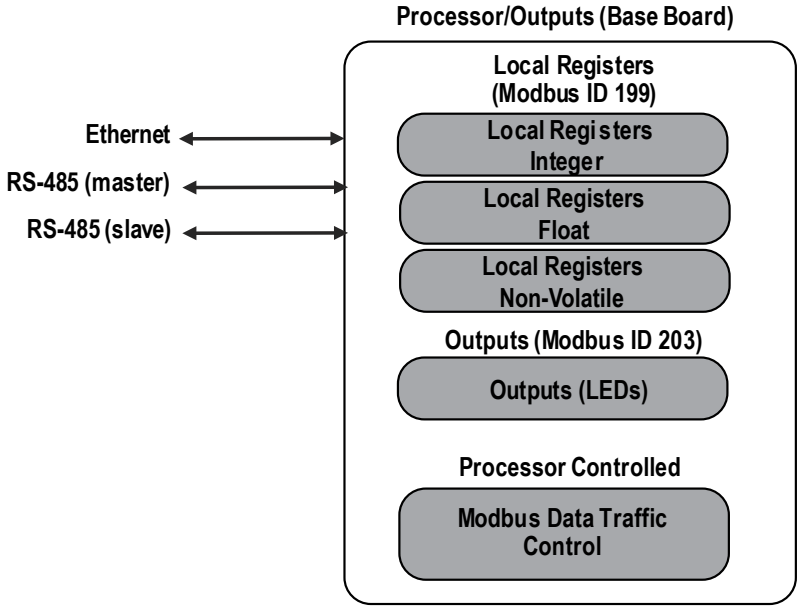
The DXMR90 has five physical RS-485 connections using Modbus RTU protocol.

The master Modbus RS-485 ports are for the DXMR90 to act as a Modbus master device to control external Modbus slave devices.

The Modbus master RS-485 ports are labeled Port 1–4. The Modbus slave port is used when another Modbus master device wants to communicate with the DXMR90 when the DXMR90 is a Modbus slave device.

The Modbus slave RS-485 port is labeled Port 0.

Figure 15. DXMR90 system overview



The DXMR90 has dual Modbus roles: a Modbus slave device and a Modbus master device. These run as separate processes.

The Modbus slave port can only access the DXMR90 local registers. To operate as a Modbus slave device, the DXMR90 needs to be assigned a unique Modbus slave ID as it pertains to the host Modbus network. This slave ID is separate from the internal Modbus slave IDs the DXMR90 uses for its own Modbus network. The DXM Modbus slave ID and other Modbus slave port parameters are defined by using the configuration software.

The DXMR90 operates the Modbus master ports. Each device on a master port must be assigned a unique slave ID. There are slave IDs that are reserved for internal devices in the DXMR90. Each device that shares a master port must have a unique ID. Devices on separate ports may have the same ID.

Table 13: Internal Modbus slave IDs (factory default)

Modbus Slave ID	Device
199	Local Registers—Internal storage registers
203	LED indicators

4.1 Assigning Modbus Slave IDs

Assign the DXM Modbus Slave ID only if a Modbus master device is reading or writing the DXM Local Register data through the Modbus RS-485 slave port 0.

To set the Modbus Slave parameters for Port 0, go to **Settings** > **System** > **Slave Port 0** settings. Here you can modify the **Baud Rate**, **Parity**, and change the **Internal Slave ID**. The **Internal Slave ID** is the Modbus ID that an external Modbus Master accesses to read/write to the local registers on the DXMR90.

DXM Master Configuration—When the DXM operates as a Modbus master device, use the configuration software to configure read or write operations of the DXM Modbus network. The DXM communicates with all internal and external peripheral devices using the external Modbus bus RS-485 ports 1–4.

4.2 Modbus Operation

All Modbus transactions are managed by a central Modbus engine.

If there are Modbus messages intended for a Modbus slave that doesn't exist, the Modbus engine waits for a response until the timeout period is expired. This slows down the Modbus polling loop for read and write operations. Each Master port is running its own modbus engine; timeouts on one port will not affect the other ports.

Verify all Modbus read and write operations are intended for Modbus slave devices that are in the network.

4.3 Modbus Communication Timeouts

A Modbus timeout is the amount of time a Modbus slave is given to return an acknowledgment of a message sent by the Modbus master. If the Modbus master waits for the timeout period and no response is seen, the Modbus master considers it a lost message and continues on to the next operation.

The timeout parameter is simple to set for Modbus devices directly connected to the DXMR90. Special considerations need to be made to set the timeout parameter when the DXMR90 is communicating to an external Modbus device through a serial data radio. In general, longer timeouts may be required to ensure the data is sent and received.

Configure controllers operating wireless networks to allow for enough time for hardware transmission retries. Set the **Communications Timeout** parameter to cover the expected time for messages to be sent throughout the wireless network. For the DXMR90, the **Communications Timeout** parameter is the maximum amount of time the DXMR90 should wait after a request is sent until the response message is received from the Modbus slave device. Use the DXM Configuration Software to set the timeout parameter on the **Register Mapping** > **RTU** > **RTU Configuration** screen.

The default setting for the timeout parameter is 5 seconds.

4.4 Modbus TCP Client

The DXMR90 can operate as a Modbus TCP client on Ethernet. Users may define up to five socket connections for Modbus TCP server devices to read Modbus register data over Ethernet. Use the DXM Configuration Software to define and configure Modbus TCP client communications with other Modbus TCP servers.

5 Configuration Instructions

5.1 Scheduler

Use the **Scheduler** screens to create a calendar schedule for local register changes, including defining the days of the week, start time, stop time, and register values.

Schedules are stored in the XML configuration file, which is loaded to the DXMR90. Reboot the DXMR90 to activate a new schedule.

If power is cycled to the DXMR90 in the middle of a schedule, the DXMR90 looks at all events scheduled that day and processes the last event before the current time.

For screens that contain tables with rows, click on any row to select it. Then click **Clone** or **Delete** to copy/paste or remove that row.

5.1.1 Create a Weekly Event

Use the **Scheduler** > **Weekly Events** screen to define weekly events.

Figure 16. Scheduler > Weekly Events screen

1. Click **Add Weekly Event**.
   A new schedule rule is created.
2. Click on the arrow to the left of the new rule to expand the parameters into view.
   The user-defined parameters are displayed.
3. Name your new rule.
4. Enter the local register.
5. Select the days of the week this rule applies to.
6. Enter the starting value for the local register.
7. Use the drop-down list to select the type of Start at time: a specific time or a relative time.
8. Enter the starting time.
9. Enter the end time and end value for the local register.

Register updates can be changed up to two times per day for each rule. Each rule can be set for any number of days in the week by clicking the buttons M, T, W, Th, F, S, or Su.

If two register changes are defined for a day, define the start time to be before the end time. Select **End Value** to enable the second event in a 24 hour period. To span across two days (crossing the midnight boundary), set the start value in the first day, without selecting **End Value**. Use the next day to create the final register state.

Start and end times can be specified relative to sunrise and sunset, or set to a specific time within a 24 hour period. When using sunrise or sunset times, set the GPS coordinates on the device so it can calculate sunrise and sunset.

5.1.2 Create a One-Time Event

Define one-time events to update registers at any time within a calendar year.

Similar to Weekly events, the times can be specific or relative to sunrise or sunset. Define one-time events using the **Scheduler** > **One Time Events** screen.

Figure 17. Scheduler > One Time Events screen

1. Click on **Add One Time Event**.
   A new one-time event is created.
2. Click on the arrow to expand the parameters into view.
   The user-defined parameters are displayed.
3. Name your one-time event by clicking on the name link and entering a name.
4. Enter the local register.
5. Enter the starting time, date, and starting value for the local register.
6. Enter the ending time, date, and ending value for the local register.

5.1.3 Create a Holiday Event

Use the **Scheduler** > **Holidays** screen to create date and/or time ranges that interrupt weekly events.

Figure 18. Scheduler > Holidays screen

1. Click on **Add Holiday**.
   A new rule is created.
2. Enter a name your new holiday rule.
3. Select the start date and time for the new holiday.
4. Select the stop date and time for the new holiday.

5.2 Authentication Setup

The DXMR90 has three different areas that can be configured to require a login and password authentication.

- Webserver/ Cloud Services Authentication
- Mail Server Authentication
- DXM Configuration Authentication

The webserver and mail server authentication depends upon the service provider.

5.2.1 Set the Controller to use Authentication

The DXMR90 can be configured to send login and password credentials for every HTTP packet sent to the webserver. This provides another layer of security for the webserver data.

Configuration requires both the webserver and the DXMR90 to be given the same credentials for the login and password. The webserver authentication username and password are not stored in the XML configuration file and must be stored in the DXMR90.

1. From within the DXM Configuration Software, go to the **Settings** > **Cloud Services** screen.
2. In the upper right, select **Show advanced settings**.
3. Define the username and password in the **Web Server Authentication** section of the screen.

Figure 19. Web Server Authentication screen

   The first time you select **Require Authentication**, a pop-up box appears with additional instructions. Since the data is not stored in the XML configuration file, it is hidden from view of the DXM Configuration Software.
4. Click on **Send Authentication**.

   The controller must be connected to the PC for this operation to succeed.

   The data transmits directly to the DXMR90's non-volatile memory. If successful, a pop-up window appears, asking to reboot the device.
5. Select **Yes** to reboot the device.

5.2.2 Set the Web Services to Use Authentication

1. At the Banner Cloud Data Services website, go to **Settings** > **Sites**.
2. To edit the site settings, click **Edit** on the line of the site name.

Figure 20. Settings > Sites screen of the Banner CDS website

At the bottom of the pop-up window is a checkbox to enable authentication/validation.

3. Enter the same username and password as used in the DXM Configuration Software. The username and password do not need to be a defined user within the Banner Cloud Data Services website.

5.2.3 Controller Configuration Authentication

The DXMR90 can be programmed to allow changes to the configuration files only with proper authentication by setting up a password on the **Settings** > **Administration** screen in the DXM Configuration Software.

With the DXMR90 connected to the PC, click **Get Device Status**. The DXMR90 status displays next to the button.

Figure 21. Settings > Administration screen

Use the DXM Configuration Software to:

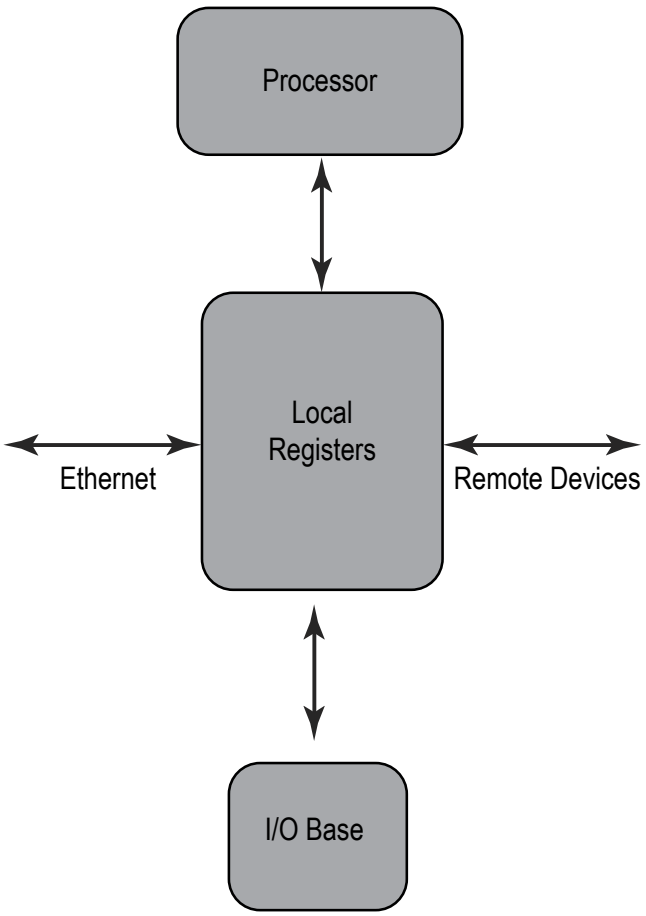
- Set the Admin Password
- Change the Admin Password
- Remove the Admin Password

To change or remove an admin password, the current password must be supplied. The DXMR90 must be connected to the PC to change the administration password.

5.3 Register Flow and Configuration

The DXMR90 register data flow goes through the Local Registers, which are data storage elements that reside within the processor. Using the configuration software, the controller can be programmed to move register data from the Local Register pool to remote devices or the I/O base.

Figure 22. Register flow



5.3.1 Basic Approach to Configuration

When programming an application in the DXMR90, first plan the overall data structure of the Local Registers. The Local Registers are the main storage elements in the DXMR90. Everything goes into or out of the Local Registers.

1. In the DXM Configuration Software, name the Local Registers to provide the beginning structure of the application.
2. Configure the read/write rules to move the data. The Read/Write rules are simple rules that move data between devices (Nodes, Modbus slaves, sensors, etc) and the Local Registers.
3. Most applications require the ability to manipulate the Local Register data, not just move data around. Use the **Action rules** to make decisions or transform the data after the data is in the Local Registers. Action rules can apply many different functions to the Local Register data, including conditional statements, math operations, copy operations, or trending.
4. To perform scheduled events in Local Registers, go to the **Scheduler** screen in the DXM Configuration Software. These rules provide the ability to create register events by days of the week. The scheduler can also create events based on sunrise or sunset.

5.3.2 Troubleshooting a Configuration

View Local Registers using the **Local Registers** > **Local Registers in Use** screen of the configuration software.

When a configuration is running on the DXMR90, viewing the Local Registers can help you to understand the application's operation. This utility can also access data from remote devices and the LED registers.

5.3.3 Saving and Loading Configuration Files

The DXM Configuration Software saves its configuration information in a XML file. Use the **File** menu to Save or Load configuration files.

Save the configuration file before attempting to upload the configuration to the DXMR90. The DXM Configuration Software uploads the configuration file saved on the PC to the DXMR90; it will not send the configuration loaded in the tool.

5.3.4 Uploading or Downloading Configuration Files

The DXMR90 requires a XML configuration file to become operational. To upload or download configuration files, connect a computer to the DXMR90 using the Ethernet port. Then use the **Upload Configuration to Device** or **Download Configuration from Device** under the **Device** menu.

5.4 EtherNet/IP™ Configuration

The DXMR90 can be configured to send/receive local register data to and from an EtherNet/IP™ [1] host. EDS (Electronic Data Sheet) files allow users of the EtherNet/IP protocol to easily add a Banner DXM device to the PLC. Download the EDS files from the Banner website.

- DXM EDS Configuration File (for PLCs) (p/n b_4205242)
- DXM EIP Config File for DXM Controller with Internal Gateway (Models: DXM1xx-BxR1, DXM1xx-BxR3, and DXM1xx-BxCxR1) (p/n 194730)

5.4.1 Configuring the Host PLC

On the host PLC, install the DXMR90 using an EDS file or by using the following parameters:

- Assembly1: Originator to DXM = Instance 112, 456 bytes (228 words)
- Assembly2: DXM to Originator = Instance 100, 456 bytes (228 words)

The Originator is the host PLC system, and the DXM is the DXMR90. The host system sees the DXMR90 as a generic device with the product name of Banner DXM (ProdType: 43 - Generic Device, ProdName: Banner DXM, Integer Type - INT).

Important: Do not set the Requested Packet Interval (RPI) any faster than 150 ms.

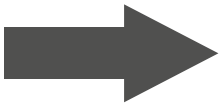
5.4.2 Configuring the Controller

Use the configuration software to define the **Protocol conversion** for each local register to be **EIP Originator** > **DXM** or **EIP DXM** > **Originator** from the **Edit Register** or **Modify Multiple Register** screens.

Define a DXM local register as **EIP Originator** > **DXM** when the host PLC (Originator) will send data to the DXMR90 local register (DXM).

Define a DXM local register as **EIP DXM** > **Originator** when that register data will be sent from the DXMR90 (DXM) to the host PLC (Originator).

Data from an EIP controller in assembly instance 112 is data destined for the DXMR90 local registers. The PLC is normally configured for INT or UINT data transfer. This allows for a seamless transfer of data.

EIP Assembly Instance 112 (16-bit)			DXM Local Registers	
Adrs	Data		Adrs	Data
0	1122		1	1122
1	3344		2	3344
2	5566		3	5566
3	7788		4	7788

[1] EttherNet/IP is a trademark of Rockwell Automation.

EIP Assembly Instance 112 (16-bit)			DXM Local Registers	
4	9900		5	9900

Data from the DXMR90 local registers is sent to the EIP controller using assembly instance 100. Each local register in the DXMR90 defined as **EIP DXM** > **Originator** is collected in numerical order and placed into the data buffer destined for assembly instance 100. DXM local registers are capable of 32-bits, but only the lower 2-bytes (16-bits) for each local register are transferred.

EIP Assembly Instance 100 (16-bit)			DXM Local Registers	
Adrs	Data		Adrs	Data
0	1122		11	1122
1	3344		12	3344
2	5566		13	5566
3	7788		14	7788
4	9900		15	9900

5.5 Set up the Email

The DXMR90 can be configured to send email messages based on threshold conditions.

Ethernet-connected systems can only use email, but can send email to cellular phones as a SMS message depending upon the network carrier. To send email to a Verizon phone, use the phone number followed by `@vtext.com`, for example, 1234567890@vtext.com.

For more information, refer to the DXM Configuration Software Instruction Manual (p/n 209933). Internal log files may be sent using email. Follow these instructions and use the DXM Configuration Software to program the controller for email.

1. On the **Settings** > **System** screen, set the **Device Time** on the DXMR90.
2. On the **Settings** > **Cloud Services** screen, select Ethernet for the **Push Interface**.
3. Configure your Ethernet connection by setting the IP settings on the **Ethernet** screen.
4. Set the email and message parameters on the **Notifications** screen.
5. To send alert messages, define the threshold rule to use email.
6. To send log files, define the log file parameters.

5.5.1 Mail Server Authentication

Complete the mail server settings to have the DXMR90 send email alert messages or to email the log files.

The SMTP password is stored in the DXMR90, not the XML configuration file. Use the **Settings** > **Notifications** screen to complete this configuration.

Figure 23. Mail server settings

After selecting **Enable SMTP Authentication** for the first time, a pop-up box appears with additional instructions to complete the mail server authentication process.

After entering the user name and password, click on **Send SMTP Password** to save the user name and password to the DXMR90. The DXMR90 must be connected to the PC to complete this operation. If successful, a pop-up window appears, asking to reboot the device. Select **Yes** to reboot the device.

5.5.2 Define the Network Interface Settings

On the **Cloud Services** screen, define the network connection settings by selecting **HTTP Cloud Push** to send data to Banner CDS or **AWS IoT Core Push** to send data to AWS IoT Core.

If you don't require pushing data to a web server, set the **Cloud Push** interval to zero.

Figure 24. Cloud Services screen

5.5.3 Configure your Ethernet Connection

To send email based on a threshold rule or to email log files, first define the network and email servers. When selecting Ethernet, go to the **Settings** > **Ethernet** screen.

1. To define the Ethernet IP address, give the DXMR90 a static IP address. In most cases you may select the device to use DHCP and have the IP address automatically assigned.
2. DNS settings are not typically required. The DXMR90 uses a public service to resolve Domain names, but if the network connection does not have Internet access, the DNS settings may be required.

Figure 25. Settings > Ethernet screen

5.5.4 Set the Email Parameters

From the **Settings** > **Notifications** screen, enter the SMTP definition, login, and password for a mail server.

To send email, you must supply the SMTP Server, Server Port, and login credentials.

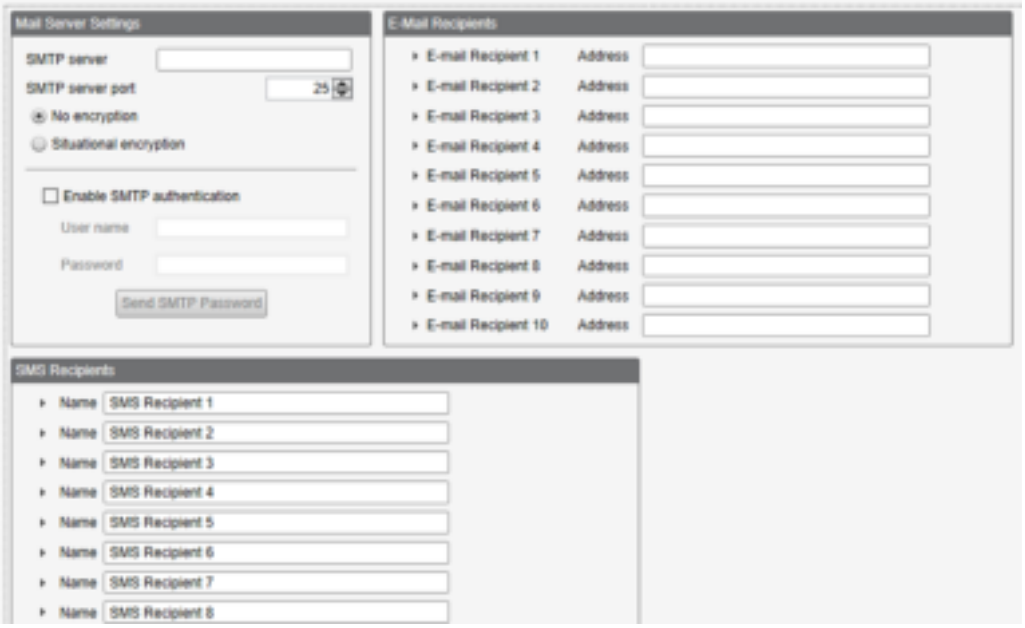
The default SMTP port is 25 but may need to be adjusted for Ethernet-based networks. Note that many facilities block port 25. Port 587 is another common SMTP submission port.

The SMTP password is not stored in the XML configuration file, but on the DXMR90. After the password is entered, click on **Send SMTP Password** to send it to the DXMR90. The password is stored in non-volatile memory, so reboot the DXMR90 to recognize the new password.

When using a GMail server, select **Situational encryption** and **Enable SMTP authentication**. GMail may notify you that you must allow access for less secure apps in your email settings.

For other email servers, the parameters may vary and will require information from the provider.

Figure 26. Email settings



At the bottom of the screen, define the recipient to receive emails. These recipients selected in the threshold definition for sending alert messages.

5.5.5 Define Threshold Rules for Email

To define a threshold, go to **Local Registers** > **Action Rules** > **Thresholds**.

Depending upon which recipients are defined, select the appropriate email or SMS checkbox for the threshold rule (under **Email/SMS on state transition**). When the threshold rules goes active or inactive, an email is generated.

For more information on how to set up threshold rules, refer to the DXM Configuration Software Instruction Manual (p/n 209933).

5.5.6 Define Log File Parameters for Emailing Log Files

The DXMR90 can email log files generated on the device.

Before emailing log files, set the **Mail and Messaging** parameters to provide the login credentials. When using Ethernet, verify the IP address settings are defined on the **Ethernet** screen. Set the DXMR90 time, under **Settings** > **System**, so that all data is properly time stamped.

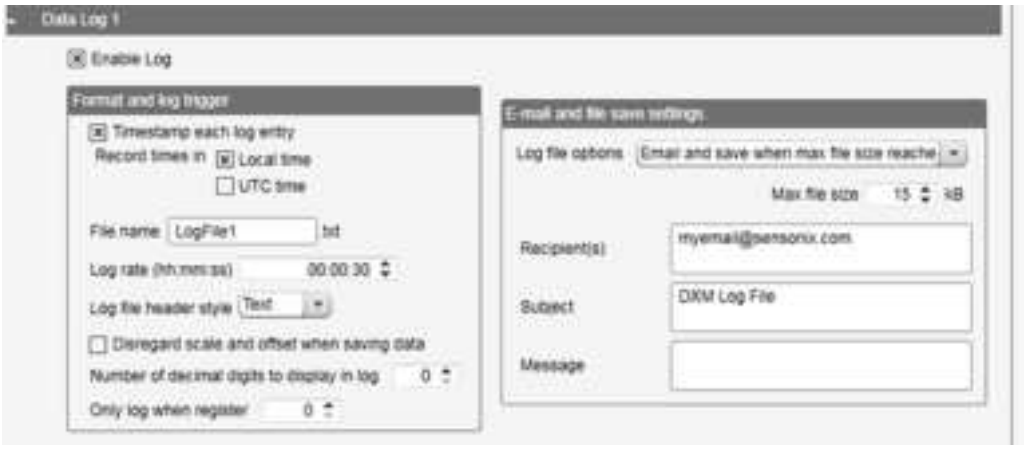
Use the **Local Registers** > **Local Registers in Use** > **Edit Register** screen to select which registers to log which log file (set the **SD card logging** to the log file. Define the setup of the log file using the **Settings** > **Logging** screen. Typical settings are shown.

1. Enable the log and timestamp with every entry.
2. Enter the filename, log rate, and the maximum file size to send via email.

Important: Do not set the log file size larger than 100 kB because this cannot be read through the configuration software.

3. Define the email address.

Figure 27. Defining the data log files



4. Define the local register data put into the log file using the **Local Registers** > **Local Register Configuration** screen, under the **Logging and Protocol Conversion** section. From the **SD Card Logging** drop-down list, select the log file to write to. Log files are written in CSV format.
5. Use the DXM Configuration Software to read back the log files. Under **Settings** > **Logging**, click **Refresh List**, highlight the file to download, then click **Save Selected**.

5.6 Ethernet Push Retries

The DXMR90 can be configured to send register data packets to a webserver. When the Ethernet communications path is not operating, the DXMR90 retries the send procedure. The communications retry process is outlined below for each configuration.

A failed communication attempt results in the register data packet being saved on the local virtual file system (VFS). The number of retries will depend upon the network connection type.

When there is no Ethernet connection, the transmission attempts are not counted as failed attempts to send data. Only when there is a good network connection and there are 10 failed attempts will the controller archive the data on the VFS. Data archived on the VFS must be manually retrieved and will be lost if power to the controller is lost..

5.6.1 Ethernet Push Retries

With an Ethernet-based network connection, the DXMR90 retries a message five times. The five retry attempts immediately follow each other. After all attempts are exhausted, the register data packet is saved in the virtual file system (VFS).

At the next scheduled time, the DXMR90 attempts to send the saved packet as well as the newly created register data packet. If it cannot send the new register data packet, the new register data packet is appended to the saved file on the VFS to be sent later. After 10 rounds of retries, the data set is archived on the VFS under folder **_sxi**. No additional attempts to resend the data are made; the data file must be manually retrieved.

Using SSL on Ethernet will have no retries, but will save each failed attempt to the VFS until 10 failed rounds. At this time, the register data packet is archived.

5.6.2 Event/Action Rule or Log File Push Retries

Event-based pushes caused by Action rules and locally stored log files sent using email follow the same process when failures occur, based on the network connection. The failed Event-based messages are resent with the next cyclical schedule or the next event message that triggers a push message.

5.6.3 Email Message Push Retries

There are no retries for emails that fail to be sent from the DXMR90.

6 PROFINET®

PROFINET is a data communications protocol for industrial automation and processes. PROFINET IO defines how controllers (IO controllers) and peripheral devices (IO devices) exchange data in real time. PROFINET® is a registered trademark of PROFIBUS Nutzerorganisation e.V. and the standard is maintained by PROFIBUS & PROFINET International (PI) , an organization headquartered in Karlsruhe, Germany.

Only the DXMR90, DXM700, DXM1000, and DXM1200 Controller models support PROFINET IO.

6.1 General Station Description Markup Language File

A PROFINET General Station Description (GSD) file is a description of an IO device provided by the device manufacturer in an XML format (GSDML.xml).

The GSD file is a standardized way of describing the device information to engineering tools and the IO controller and can work across a variety of tools as a standard set of device information.

6.2 DXM PROFINET IO Data Model

The PROFINET IO data model is based on the typical, expandable field device that has a backplane with slots. Modules have different functionalities.

Modules are plugged into slots. In the PROFINET IO data model Slot 0, Subslot 1 is reserved for the Device Access Point (DAP) or network interface.

- Inputs Integer Module sends integer data to the PLC
- Inputs Floats Module sends floating point data to the PLC
- Output Integer Module receives integer data from the PLC

6.3 Configure the DXM Controller for a PROFINET IO Connection

To use PROFINET, follow these instructions.

1. Using the DXM Configuration Software, go to the **Settings** > **Ethernet** screen.
2. Select **Enable PROFINET**.
3. Save the configuration file and upload it to the DXM Controller (see Save and Upload the Configuration File on page 14).
   After PROFINET is enabled, the IP address for the DXM Controller is controlled by the PROFINET host. The LCD display and DXM Configuration Software no longer control the setting of the IP address. The PROFINET IP address settings are invisible to the LCD display and the DXM Configuration Software.

The PROFINET data type and data size to/from the DXM Controller is configurable. The PROFINET data is processed from the Local Register of the DXM Controller.

Configure the Local Registers to match the PROFINET module definitions in the host PLC.

6.3.1 Save and Upload the Configuration File

After making any changes to the configuration, you must save the configuration files to your computer, then upload it to the device.

Changes to the XML file are not automatically saved. Save your configuration file before exiting the tool and before sending the XML file to the device to avoid losing data. If you select **DXM** > **Send XML Configuration to DXM** before saving the configuration file, the software will prompt you to choose between saving the file or continuing without saving the file.

1. Save the XML configuration file to your hard drive by going to the **File** > **Save As** menu.
2. Go to the **DXM** > **Send XML Configuration to DXM** menu.

Figure 28. Status indicator bar

- If the Application Status indicator is red, close and restart the DXM Configuration Tool, unplug and re-plug in the cable and reconnect the DXM to the software.
- If the Application Status indicator is green, the file upload is complete.
- If the Application Status indicator is yellow, the file transfer is in progress.

The device reboots and begins running the new configuration.

6.4 Slots and Modules

There are nine slots to accommodate the DXM Controller data.

Table 14: Slots for input and output values

Values	Slots	Maximum Data Size
Input values	1–6	1440 bytes
Output values	7–9	1440 bytes

Table 15: Listing of slots for input and output values

Slot	PLC		DXM Local Register		Module Size
	Module Definition		Start	End	512
Slot 1	Inputs Integer	<-	1	256	
Slot 2	Inputs Integer	<-	257	512	
Slot 3	Inputs Integer	<-	513	768	
Slot 4	Inputs Float	<-	1001	1256	
Slot 5	Inputs Float	<-	1257	1512	
Slot 6	Inputs Float	<-	1513	1768	
Slot 7	Output Integer	->	5001	5256	
Slot 8	Output Integer	->	5257	5512	
Slot 9	Output Integer	->	5513	5768	

The DXM Local Register association shown uses a Module size of 512 bytes, which equals 256 Local Registers in the DXM. Module sizes supported are 64, 128, 256 and 512 bytes. Input Integers are data from the DXM to the PLC. Output integers are data from the PLC to the DXM.

Table 16: Slots 1 through 3

Module	Notes
Input Integer 512	Allowed in slots 1-3, Module Identifier= 0x30
Input Integer 256	Allowed in slots 1-3, Module Identifier= 0x31
Input Integer 128	Allowed in slots 1-3, Module Identifier= 0x32
Input Integer 64	Allowed in slots 1-3, Module Identifier= 0x33

Table 17: Slots 4 through 6

Module	Notes
Input Float 512	Allowed in slots 4-6, Module Identifier= 0x34
Input Float 256	Allowed in slots 4-6, Module Identifier= 0x35
Input Float 128	Allowed in slots 4-6, Module Identifier= 0x36
Input Float 64	Allowed in slots 4-6, Module Identifier= 0x37

Table 18: Slots 7 through 9

Module	Notes
Output Integer 512	Allowed in slots 7-9, Module Identifier= 0x40
Output Integer 256	Allowed in slots 7-9, Module Identifier= 0x41
Output Integer 128	Allowed in slots 7-9, Module Identifier= 0x42
Output Integer 64	Allowed in slots 7-9, Module Identifier= 0x43

Example Configuration

Table 19: Example configuration for slots and modules

Slot	Module	Description
Slot 1	Input Integer 512	The two input integer modules have a total of 640 bytes (320 Modbus registers) The data will come from DXM Local Registers 1 through 320
Slot 2	Input Integer 128	
Slot 4	Input Float 128	The input Floating register module has a total of 128 byes (64 Modbus registers) Since it takes two Modbus registers to make a 32-bit floating value, there will be 32 floating point values coming from Local Registers 1001-1064
Slot 7	Output Integer 64	The output integer 64 module has a total of 64 bytes (32 Modbus registers). The data will come from the PLC and be put into DXM Local Registers 5001 through 5032

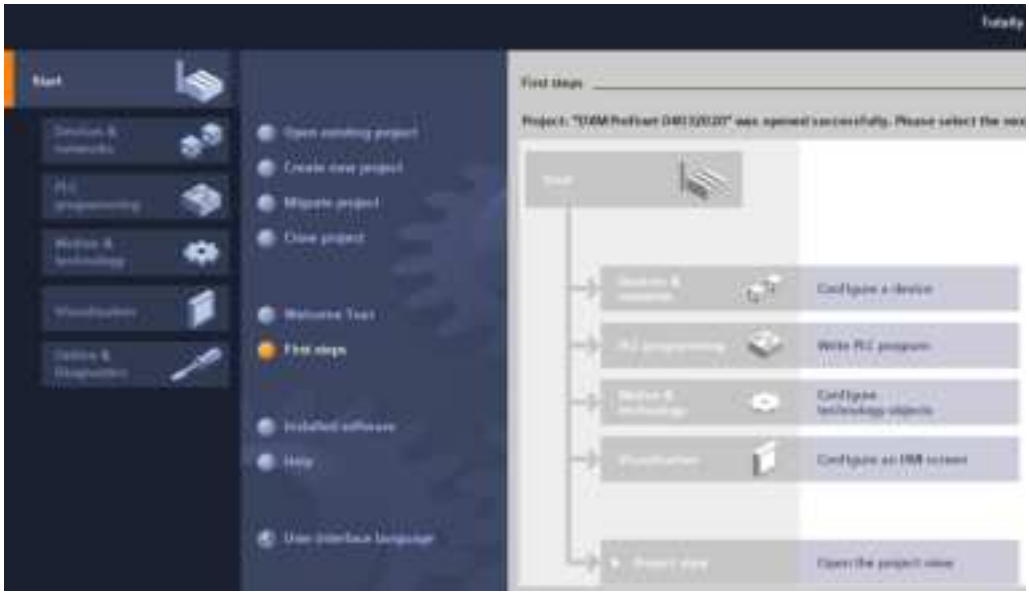
6.5 Configuration Instructions

6.5.1 Install the GSD File

Although these instructions are specific for the Siemens TIA Portal (v14) software, you may use these instructions as a basis for installing the GSD file into another controller.

1. Download the GSD file from www.bannerengineering.com.
2. Launch the Siemens TIA Portal (v14) software.
3. Click **Open existing project**.
4. Select a project and open it.
5. After the project is uploaded, click **Devices & networks**.

Figure 29. Devices and networks screen

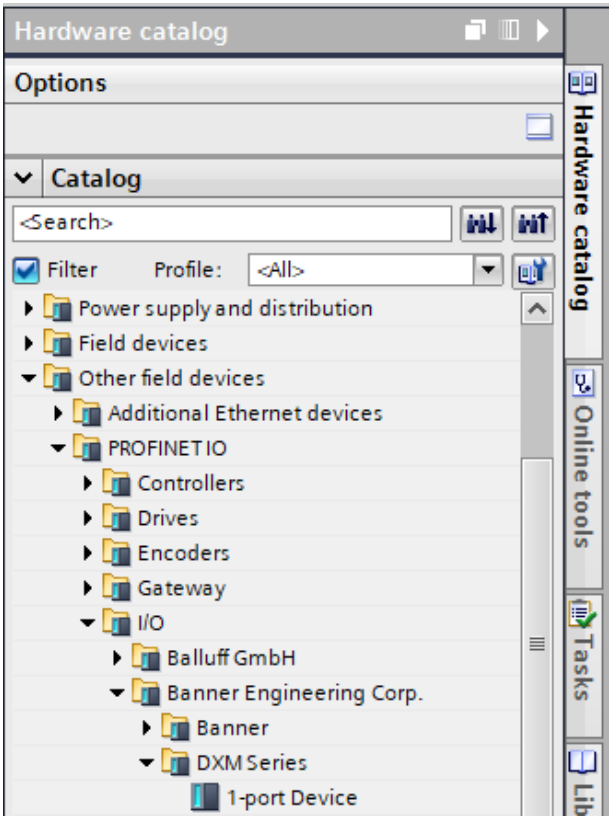


6. Click **Configure networks**.

Figure 30. Configure networks screen

7. Click **Options** and select **Manage general station description file (GSD)**.
   The **Install general station decription file** window opens.
8. Click the **More options (...)** icon to the right of the **Source path** field and browse to the location the DXM GSD file was downloaded to.
9. Select the DXM GSD file.
10. Click **Install**.

Figure 31. Hardware catalog



The system installs the DXM GSD file and places it in the **Hardware catalog**. In the example, the DXM GSD file is located under **Other field devices** > **PROFINET IO** > **Banner Engineering Corp.** > **Banner**.

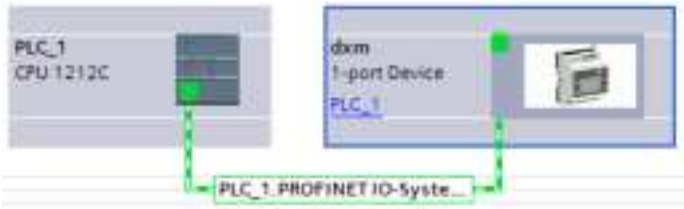
If the DXM GSD file does not install properly, save the log and contact Banner Engineering Corp.

6.5.2 Change the Device IP Address

Follow these instructions to change the IP address of the DXM device using the Siemens TIA Portal (v14) software. Use these instructions as a basis if you are using another controller (PLC).

1. Launch the Siemens TIA Portal (v14) software.
2. Click **Open existing project**.
3. Select a project and open it.
4. Click **Devices & networks**.

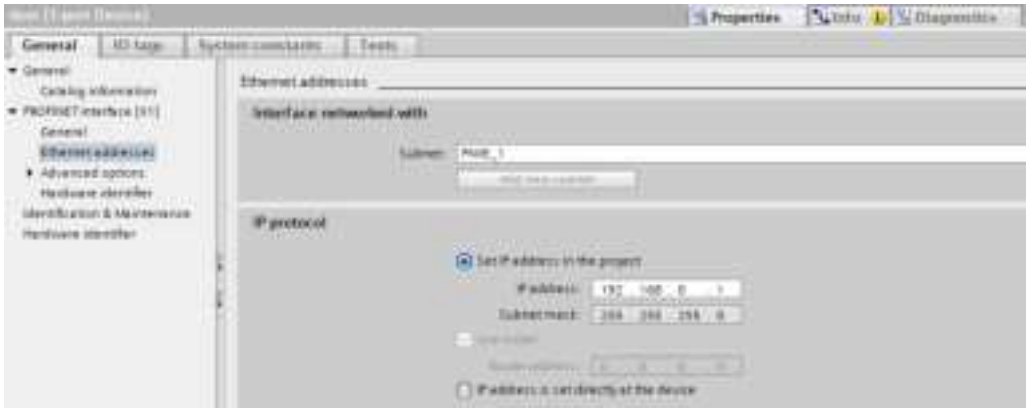
Figure 32. Network view



The **Network view** displays.

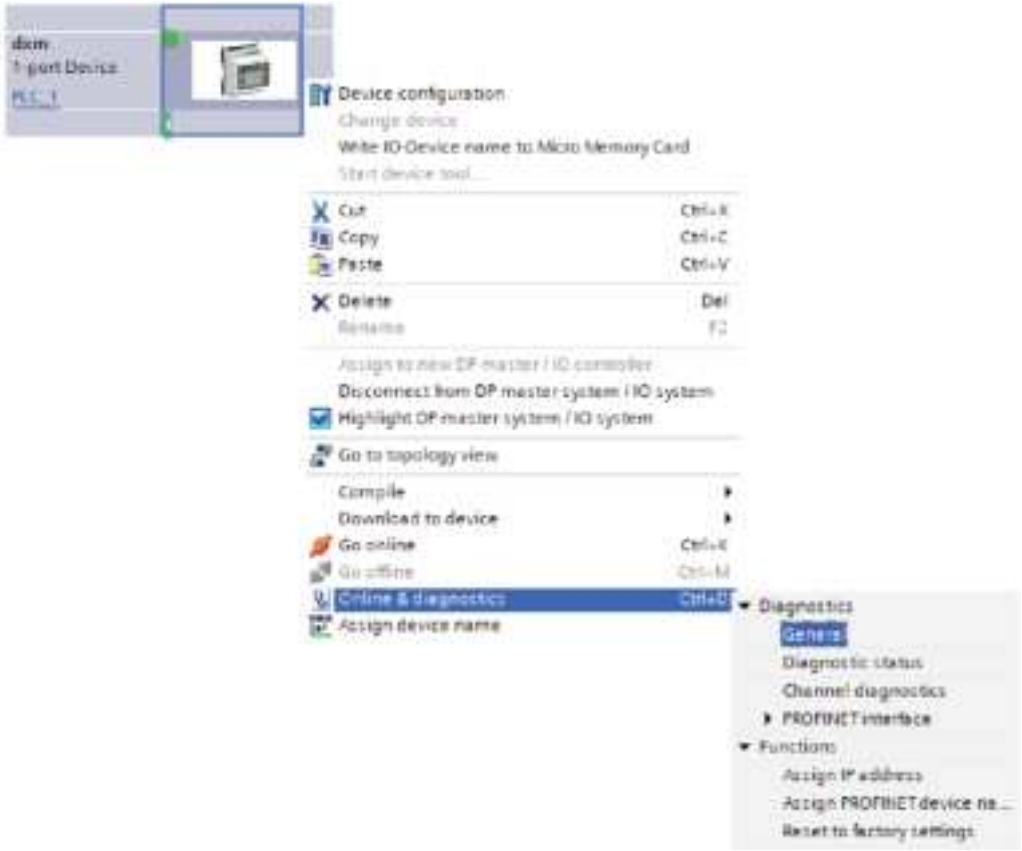
5. Double-click on the DXM icon to open the **Device view** screen.
6. Click on the DXM icon in the graphic area of the **Device view** screen.
   The **Module properties** window displays and the module can now be configured.
7. Click **Properties**.
8. Click **General**.
9. Select **PROFINET Interface** > **Ethernet addresses**.

Figure 33. Ethernet addresses



10. Select **Set IP address in the project**.
11. Enter the IP address.
12. Right-click on the device icon and select **Online & diagnostics**.

Figure 34. Online & diagnostics menu and screen



The **Online & diagnostics** windows displays.

13. Select **Assign IP address** under **Functions**.
14. Click **Accessible devices**.
    The Select device window searches the network for available devices.
15. Determine the device to be adjusted via the MAC address and select it.
16. Click **Apply**.
    The IP address for the device is updated.
17. Click **Assign IP address** to complete the step.
    This step is completed for every device.

By default, each DXM shipped from the factory is assigned the IP address 192.168.0.1.

Immediately after the PROFINET protocol is enabled, the DXM has an IP address of 0.0.0.0. We recommend using the TIA Portal to give the DXM an IP address so that the address is saved in the unit. When the PLC powers up, this IP address is accessible. The PLC can change the IP address if it is configured to do so.

If the PLC assigns the DXM IP address (for example, using the Set IP address in the project option in Siemens TIA Portal), the DXM receives the specified address, but only after the program has been loaded into the PLC and is running. If the DXM is restarted after it was discovered and configured by the PLC, the DXM retains the IP address that was assigned to it using the LCD or software until after the PLC discovers the DXM and assigns it the specified address again. However, if this address is different than what is specified in the PLC, the DXM reverts to the address specified in the PLC after the PLC becomes active again.

These configuration options conform to the PROFINET standard.

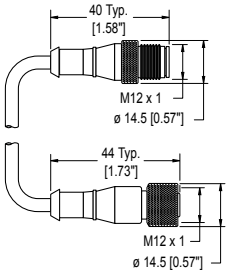
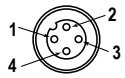
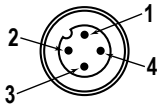
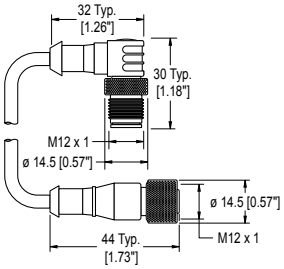
6.5.3 Change the Device Name

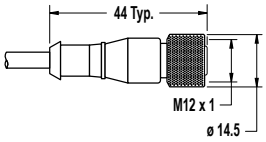
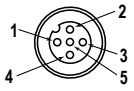
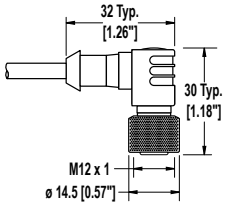
Follow these instructions to change the name of the DXM using the Siemens TIA Portal (v14) software. Use these instructions as a basis if you are using another controller (PLC).

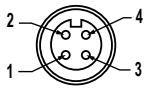
1. Open a project and click on **Devices & networks**.
   The Network view displays.
2. Right-click on the DXM icon and select **Assign device name**.
   The **Assign PROFINET device name** window displays. The software searches for devices of the same type.

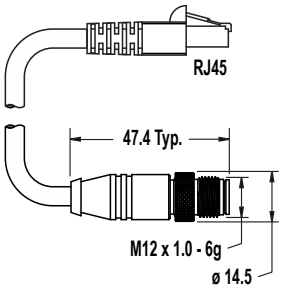
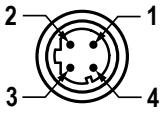
3. Enter the desired name in the **PROFINET device name** field. Note that each name can be used only once.
4. Click **Assign name**.
   The device now has a PROFINET name.

7 Accessories

4-Pin Threaded M12 Cordsets—Double Ended				
Model	**Length**	**Style**	**Dimensions**	**Pinout**
MQDEC-401SS	0.31 m (1 ft)	Male Straight/ Female Straight		Female  Male  1 = Brown 2 = White 3 = Blue 4 = Black
MQDEC-403SS	0.91 m (2.99 ft)			
MQDEC-406SS	1.83 m (6 ft)			
MQDEC-412SS	3.66 m (12 ft)			
MQDEC-420SS	6.10 m (20 ft)			
MQDEC-430SS	9.14 m (30.2 ft)			
MQDEC-450SS	15.2 m (49.9 ft)			
MQDEC-403RS	0.91 m (2.99 ft)	Male Right-Angle/ Female Straight		
MQDEC-406RS	1.83 m (6 ft)			
MQDEC-412RS	3.66 m (12 ft)			
MQDEC-420RS	6.10 m (20 ft)			
MQDEC-430RS	9.14 m (30.2 ft)			
MQDEC-450RS	15.2 m (49.9 ft)			

5-Pin Threaded M12 Cordsets—Single Ended				
Model	**Length**	**Style**	**Dimensions**	**Pinout (Female)**
MQDC1-501.5	0.5 m (1.5 ft)	Straight		 1 = Brown 2 = White 3 = Blue 4 = Black 5 = Gray
MQDC1-506	2 m (6.5 ft)			
MQDC1-515	5 m (16.4 ft)			
MQDC1-530	9 m (29.5 ft)			
MQDC1-506RA	2 m (6.5 ft)	Right-Angle		
MQDC1-515RA	5 m (16.4 ft)			
MQDC1-530RA	9 m (29.5 ft)			

4-Pin Threaded M12 RS-485 to USB Adapter Cordset, with Wall Plug				
Model	**Length**	**Style**	**Dimensions**	**Pinout (Female)**
BWA-UCT-900	1 m (3.28 ft)	Straight		 1 = Brown 2 = White 3 = Blue 4 = Black

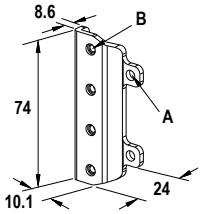
4-pin M12 D-code to RJ45 Shielded Ethernet				
Model	**Length**	**Style**	**Dimensions**	**Pinout (Male)**
STP-M12D-406	1.83 m (6 ft)	Straight		  1 = White/ Orange 2 = Orange 3 = White/ Blue 6 = Blue 1 = White/ Orange 2 = White/ Blue 3 = Orange 4 = Blue
STP-M12D-415	4.57 m (15 ft)			
STP-M12D-430	9.14 m (30 ft)			

SMBR90S

- Stainless steel bracket
- 4x M4-07 pemnuts (B)
- Includes 2x M4 stainless steel hex head screws and flat washers

Hole center spacing: A = 40, B = 20

Hole size: A = ø 5



Power Supplies

PSD-24-4—DC Power Supply, Desktop style, 3.9 A, 24 V DC, Class 2, 4-pin M12/Euro-style quick disconnect (QD)

PSDINP-24-06—DC power supply, 0.63 Amps, 24 V DC, with DIN Rail Mount, Class I Division 2 (Groups A, B, C, D) Rated

PSDINP-24-13 —DC power supply, 1.3 Amps, 24 V DC, with DIN Rail Mount, Class I Division 2 (Groups A, B, C, D) Rated

PSDINP-24-25 — DC power supply, 2.5 Amps, 24 V DC, with DIN Rail Mount, Class I Division 2 (Groups A, B, C, D) Rated

PSW-24-1—DC power supply with multi-blade wall plug, 100–240 V AC 50/60 Hz input, 24 V DC 1 A output, UL Listed Class 2, 4-pin female M12 connector

PSWB-24-1—DC power supply with multi-blade wall plug,100–240 V AC 50/60 Hz input, 24 V DC 1 A output, UL Listed Class 2, barrel jack connector

8 Product Support and Maintenance

8.1 Specifications

Supply Voltage
12 V DC to 30 V DC

Supply Protection Circuitry
Protected against reverse polarity and transient voltages

Power Consumption
120 mA maximum at 12 V DC

Construction
Connector Body: PVC translucent black

Indicators
Amber: Power port 0
Amber: Modbus communications port 0-4
Green/amber: Ethernet communications
Red/amber/green: User configurable LEDs

Connections
Five integral 5-pin M12 female quick disconnect
One integral 4-pin M12 male quick disconnect
One integral 5-pin M12 female D-Code quick disconnect

Application Note
When connecting external devices through the DXMR90, it is important not to exceed maximum current limitations of 3.5 Amps

Communication Protocols
Modbus® RTU, Modbus/TCP, EtherNet/IP™, and PROFINET® [2]

Security Protocols
TLS, SSL, HTTPS

Communication Hardware (RS-485)
Interface: 2-wire half-duplex RS-485
Baud rates: 1.2K, 2.4K, 9.6k, 19.2k (default), 38.4k, 57.6K, or 115.2K
Data format: 8 data bits, no parity, 1 stop bit

Environmental Ratings
For Indoor Use Only
IP65, IP67, NEMA 1, UL Type 1

Operating Conditions
–40 °C to +70 °C (–40 °F to +158 °F)
90% at +70 °C maximum relative humidity (non-condensing)

Storage Temperature
–40 °C to +80 °C (–40 °F to +176 °F)

Vibration and Mechanical Shock
Meets IEC 60068-2-6 requirements (Vibration: 10 Hz to 55 Hz, 1.0 mm amplitude, 5 minutes sweep, 30 minutes dwell)
Meets IEC 60068-2-27 requirements (Shock: 30G 11 ms duration, half sine wave)

Required Overcurrent Protection

WARNING: Electrical connections must be made by qualified personnel in accordance with local and national electrical codes and regulations.

Overcurrent protection is required to be provided by end product application per the supplied table.
Overcurrent protection may be provided with external fusing or via Current Limiting, Class 2 Power Supply.
Supply wiring leads < 24 AWG shall not be spliced.
For additional product support, go to www.bannerengineering.com.

Supply Wiring (AWG)	Required Overcurrent Protection (Amps)
20	5.0
22	3.0
24	2.0
26	1.0
28	0.8
30	0.5

8.2 File System and Archive Process

The DXM file system consists of two physical components: the serial EEPROM that stores non-volatile configuration information and a removable micro SD card that stores file backup data and user created files.

EEPROM Files—The serial EEPROM stores basic data that is required to be non-volatile, including network configuration data, IP address, MAC address, network masks, firewall settings, and authentication information. The controller XML configuration file created by the DXM Configuration Software is stored in EEPROM. The small section of non-volatile local registers is also stored in EEPROM.

Micro SD Card Files—The micro SD card contains most files at the root level. The archive directory contains files kept by the system for history backup. Archive files are stored in the directory **_sxi** and are only accessible by removing the SD card.

- Data Log Files
- HTTP Push Files
- User created ScriptBasic file
- ScriptBasic program file
- CmVMon file
- _sxi Archive directory

[2] EtherNet/IP™ is a trademark of ODVA, Inc. Modbus® is a registered trademark of Schneider Electric USA, Inc. PROFINET® is a registered trademark of PROFIBUS Nutzerorganisation e.V.

Data Log files

Users may create up to four data log files using the DXM Configuration Software. The log files are stored in the root directory on the SD card. When the file size limit is reached, the filename is changed to include the date and time and the file is moved into the archive directory **_sxi**. If a finished log file is to be e-mailed, it will be done at this time and then moved into the archive directory. Archived log files are deleted based on the Clear Logs parameter.

HTTP Push File

If the DXMR90 is configured to send data to a webserver or host system, the device creates an HTTP.LOG file on the SD card. The HTTP log is created only if the Logging Interval is non-zero and the HTTP enable log is set. An entry is placed in the HTTP log file at the Logging Interval specified by the user. At the Push Interval time, the HTTP log file is sent to the webserver or host system. If the transmission is successful, the HTTP log file is time stamped and placed into the archive directory (**_sxi**). If the transmission fails, the file remains in the root directory and subsequent Logging Intervals are appended to the file and are sent at the next Push Interval. See GUID-6A15569F-0E0B-4903-949E-01020D372E34#GUID-6A15569F-0E0B-4903-949E-01020D372E34.

User Created ScriptBasic Files

Users may use ScriptBasic to create files on the SD card by using the FILEOUT function. The filenames are fixed and up to five files can be created in the root directory.

ScriptBasic Program File

The main ScriptBasic program that runs at boot time is stored on the SD card in the root directory.

CmVMon File

The CmVMon.txt file (Cellular milli-Volt Monitor) is created by the system and is used to track power events. Every power-up cycle is date/time stamped with the voltage read from the I/O board. The value 24487 is equal to 24.487 volts. If the voltage drops below 11.2 V, another entry is put in the log file indicating the cellular modem will shut down.

CM	2015-09-22 18:52:43	VMon	Power entered normal range 24487
CM	2015-10-13 20:49:47	VMon	Power entered normal range 24004
CM	2015-10-16 15:00:20	VMon	Power entered normal range 24014
CM	2015-10-19 19:12:26	VMon	Power entered normal range 12845

_sxi Archive Directory

Only two types of files are moved into the archive directory: data log files and HTTP log files. Data log files are date/time stamped and placed into the archive directory when the size limit is reached. HTTP log files are date/time stamped then placed into the archive directory when they are successfully sent to the webserver or host system. If the HTTP log files were not successfully sent after the retries have been exhausted, the files are placed into a root directory called **sav**.

8.3 Update Your DXM Processor Firmware Using the Configuration Software

To update your processor firmware using the DXM Configuration Software, follow these instructions.

1. Using the DXM Configuration Software version 4 or later, connect to the DXMR90 via Ethernet.

   File loads to the DXMR90 will take about 15 minutes using USB or approximately 2 minutes using Ethernet.
2. On the DXM Configuration Software, go to **Settings** > **General** > **Device Information** to verify the current firmware version.

   You must load a different version with the same firmware number for the boot loader to operate. Download firmware files from the Banner website.

Figure 35. Device Information

3. Under **Settings** > **Reprogram**, click **Select upgrade file** to select the firmware file to program.

After the file load is completed, the DXMR90 restarts and loads the new firmware file. It takes about 2 minutes to complete the programming process. The device reboots when finished. Verify the firmware has been updated, under **Settings** > **General** > **Device Information**.

8.4 DXMR90 Support Policy

The DXM Wireless Controllers are industrial wireless controllers that facilitate Industrial Internet of Things (IIoT) applications. As a communications gateway, it interfaces local serial ports, local I/O ports, and local ISM radio devices to the Internet using either a cellular connection or a wired Ethernet network connection. In a continuing effort to provide the best operation for the DXMR90, stay connected with Banner Engineering Corp to hear about the latest updates through the Banner website. Create a login today to stay informed of all Banner product releases.

8.4.1 Firmware Updates

The DXMR90 has been designed to be a robust and secure IOT device. To provide the most reliable and secure device possible, periodic firmware updates are released to enhance and expand the capabilities of the DXMR90. Firmware updates and description details are found on the Banner website. Customers with critical update requirements will get access to pre-released firmware from the factory.

8.4.2 Website Information

The Banner website is the main method of disseminating DXMR90 information to customers. The data found on the website include:

- DXM instruction manuals
- Configuration manuals
- Firmware downloads
- Firmware release notes
- Errata data, any known issues with a release of firmware
- Possible work-around solutions for known issues
- DXM Solutions Guides

8.4.3 Feature Requests

Our customer is our most valuable resource to improve our DXMR90. If you have suggestions for improvements to the DXMR90 or configuration tools, please contact Banner Engineering Corp.

8.4.4 Potential DXM Issues

Potential issues with the DXMR90 are collected from Banner's support engineers to provide solutions. Users can get help from the website documentation or by calling Banner Engineering for support help. Solutions are as simple as configuration adjustments, work-around configuration solutions, or potential new firmware updates.

8.4.5 DXM Security

The DXMR90 was designed to collect local wireless sensor data, local sensor data, provide simple control, and send the data to the cloud.

The DXMR90 does not run a Linux or Windows based operating system but an embedded real-time operating system (RTOS) environment. As a proprietary operating system, the security aspects are easier to manage and minimize.

Security updates are released through the Banner Engineering Corp website (www.bannerengineering.com) and New Product Release Announcements (NPRA).

8.5 Contact Us

Banner Engineering Corp. headquarters is located at:

9714 Tenth Avenue North
Minneapolis, MN 55441, USA
Phone: + 1 888 373 6767

For worldwide locations and local representatives, visit www.bannerengineering.com.

8.6 Warnings

Install and properly ground a qualified surge suppressor when installing a remote antenna system. Remote antenna configurations installed without surge suppressors invalidate the manufacturer's warranty. Keep the ground wire as short as possible and make all ground connections to a single-point ground system to ensure no ground loops are created. No surge suppressor can absorb all lightning strikes; do not touch the Sure Cross® device or any equipment connected to the Sure Cross device during a thunderstorm.

Exporting Sure Cross® Radios. It is our intent to fully comply with all national and regional regulations regarding radio frequency emissions. **Customers who want to re-export this product to a country other than that to which it was sold must ensure the device is approved in the destination country.** The Sure Cross wireless products were certified for use in these countries using the antenna that ships with the product. When using other antennas, verify you are not exceeding the transmit power levels allowed by local governing agencies. This device has been designed to operate with the antennas listed on Banner Engineering's website and having a maximum gain of 9 dBm. Antennas not included in this list or having a gain greater that 9 dBm are strictly prohibited for use with this device. The required antenna impedance is 50 ohms. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen such that the equivalent isotropically radiated power (EIRP) is not more than that permitted for successful communication. Consult with Banner Engineering Corp. if the destination country is not on this list.

Important: Please download the complete DXMR90 Series Controller technical documentation, available in multiple languages, from www.bannerengineering.com for details on the proper use, applications, Warnings, and installation instructions of this device.

Important: Por favor descargue desde www.bannerengineering.com toda la documentación técnica de los DXMR90 Series Controller, disponibles en múltiples idiomas, para detalles del uso adecuado, aplicaciones, advertencias, y las instrucciones de instalación de estos dispositivos.

Important: Veuillez télécharger la documentation technique complète des DXMR90 Series Controller sur notre site www.bannerengineering.com pour les détails sur leur utilisation correcte, les applications, les notes de sécurité et les instructions de montage.

WARNING:

- **Do not use this device for personnel protection**
- Using this device for personnel protection could result in serious injury or death.
- This device does not include the self-checking redundant circuitry necessary to allow its use in personnel safety applications. A device failure or malfunction can cause either an energized (on) or de-energized (off) output condition.

Important:

- **Never operate a 1 Watt radio without connecting an antenna**
- Operating 1 Watt radios without an antenna connected will damage the radio circuitry.
- To avoid damaging the radio circuitry, never apply power to a Sure Cross® Performance or Sure Cross MultiHop (1 Watt) radio without an antenna connected.

Important:

- **Electrostatic discharge (ESD) sensitive device**
- ESD can damage the device. Damage from inappropriate handling is not covered by warranty.
- Use proper handling procedures to prevent ESD damage. Proper handling procedures include leaving devices in their anti-static packaging until ready for use; wearing anti-static wrist straps; and assembling units on a grounded, static-dissipative surface.

8.7 Banner Engineering Corp Limited Warranty

Banner Engineering Corp. warrants its products to be free from defects in material and workmanship for one year following the date of shipment. Banner Engineering Corp. will repair or replace, free of charge, any product of its manufacture which, at the time it is returned to the factory, is found to have been defective during the warranty period. This warranty does not cover damage or liability for misuse, abuse, or the improper application or installation of the Banner product.

THIS LIMITED WARRANTY IS EXCLUSIVE AND IN LIEU OF ALL OTHER WARRANTIES WHETHER EXPRESS OR IMPLIED (INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE), AND WHETHER ARISING UNDER COURSE OF PERFORMANCE, COURSE OF DEALING OR TRADE USAGE.

This Warranty is exclusive and limited to repair or, at the discretion of Banner Engineering Corp., replacement. **IN NO EVENT SHALL BANNER ENGINEERING CORP. BE LIABLE TO BUYER OR ANY OTHER PERSON OR ENTITY FOR ANY EXTRA COSTS, EXPENSES, LOSSES, LOSS OF PROFITS, OR ANY INCIDENTAL, CONSEQUENTIAL OR SPECIAL DAMAGES RESULTING FROM ANY PRODUCT DEFECT OR FROM THE USE OR INABILITY TO USE THE PRODUCT, WHETHER ARISING IN CONTRACT OR WARRANTY, STATUTE, TORT, STRICT LIABILITY, NEGLIGENCE, OR OTHERWISE.**

Banner Engineering Corp. reserves the right to change, modify or improve the design of the product without assuming any obligations or liabilities relating to any product previously manufactured by Banner Engineering Corp. Any misuse, abuse, or improper application or installation of this product or use of the product for personal protection applications when the product is identified as not intended for such purposes will void the product warranty. Any modifications to this product without prior express approval by Banner Engineering Corp will void the product warranties. All specifications published in this document are subject to change; Banner reserves the right to modify product specifications or update documentation at any time. Specifications and product information in English supersede that which is provided in any other language. For the most recent version of any documentation, refer to: www.bannerengineering.com.

For patent information, see www.bannerengineering.com/patents.

Index

A

authentication 24, 25

C

configuration file
    save 14, 33
    upload 14, 33
create
    holiday event 24
    one-time event 23
    weekly event 23

H

holiday event 24

M

modify
    multiple registers 12

O

one-time event 23

R

registers
    multiple registers 12

S

save
    configuration file 14, 33
scheduler 23, 24

U

upload
    configuration file 14, 33

W

webserver authentication 24, 25
weekly event 23