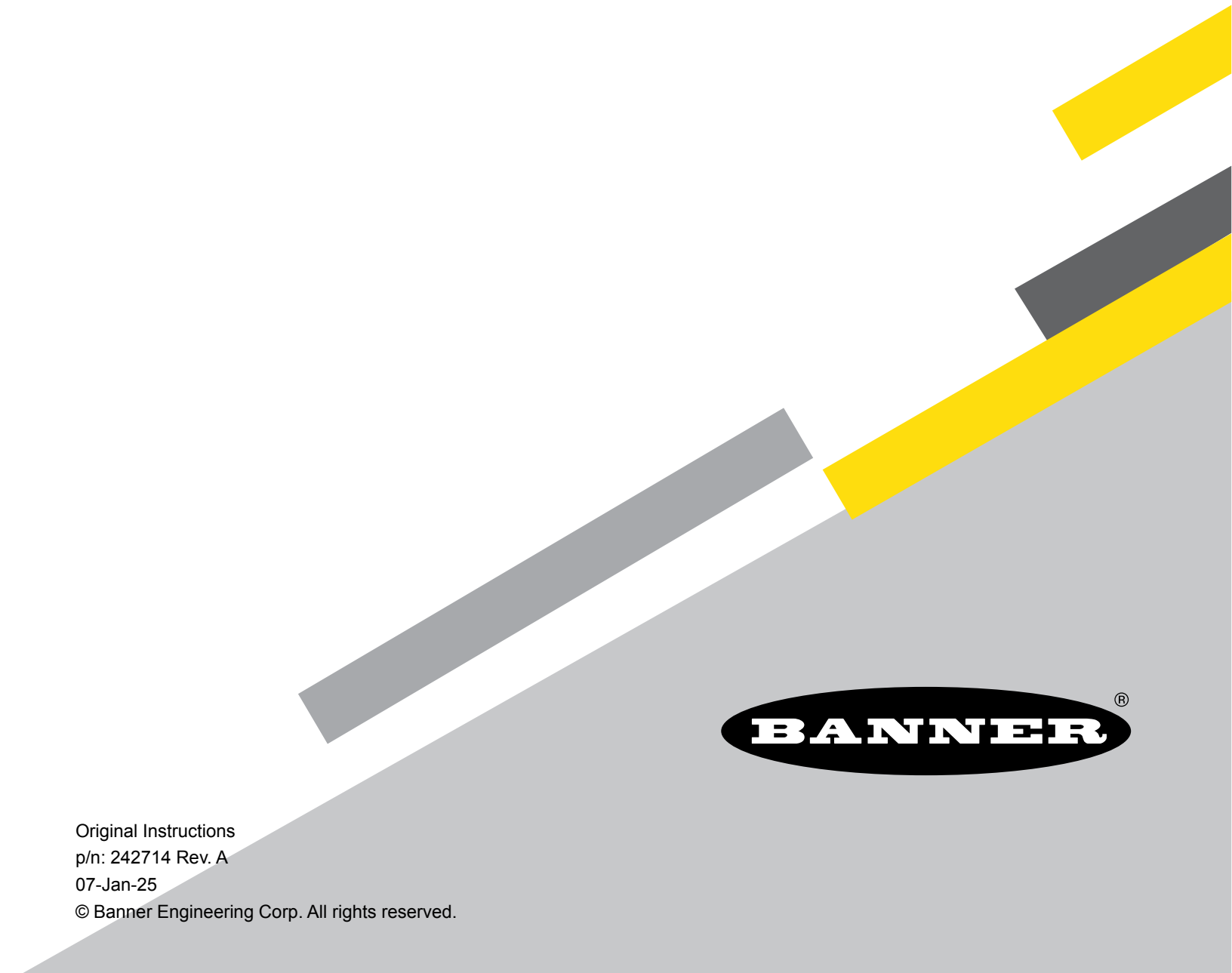
DXMR90-X1E Series Industrial Controller Product Manual



Original Instructions
p/n: 242714 Rev. A
07-Jan-25

Contents

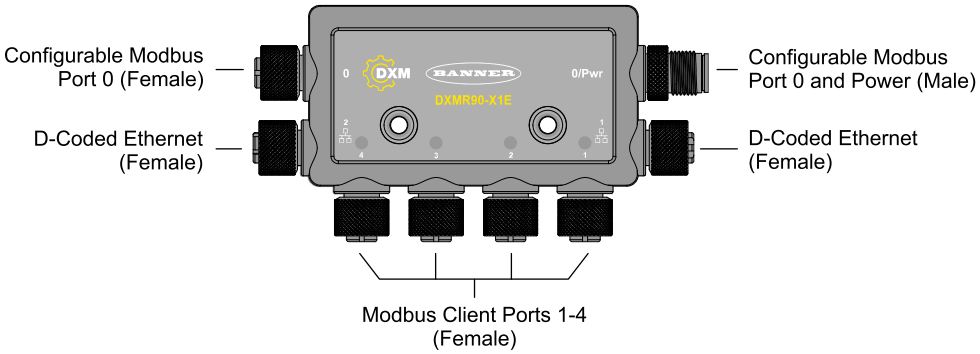
Chapter Contents

Chapter 1 DXMR90-X1E Overview

Banner's DXMR90-X1E Series Industrial 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-X1E contains four individual Modbus clients 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.

Overview of the DXMR90-X1E Series Industrial 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 and to pass power to other DXM Controllers in series. Inline power and Ethernet switches enable multiple DXMR90-X1E Controllers to be connected in series. Two 100 Mbps Ethernet ports (female) use an M12 D-coded Ethernet connection.

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

Four Modbus client connections using female M12 connectors.

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

The DXMR90-X1E is compatible with Internet protocols including RESTful API, MQTT with web services from AWS, and MQTT Sparkplug B.

Logic Controller

Program the DXMR90-X1E'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
- 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

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 Client Read or Write rules for external devices on the network

Wired Connectivity

Ethernet: Modbus/TCP (client/server) or Ethernet/IP

Field Bus: Modbus RS-485 Client/Server

User Interface

API Interface—Host Initiated control and Web service integration

Modbus registers for internal local registers (Modbus ID 199)

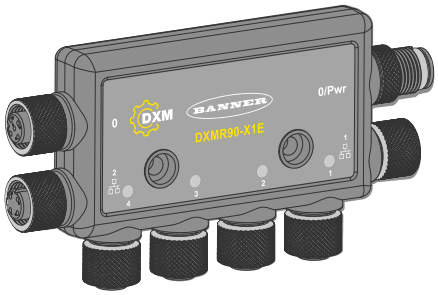
Local Registers	Type	Description
1–845	32-bit integer	Local data registers
846–849	32-bit integer	Reset, Constant, Timer
851–900	32-bit non-volatile integer	Data flash, non-volatile
901–1000		Reserved for internal use
1001–5000	Floating point	Floating point registers, local data registers
5001–7000	32-bit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

DXMR90-X1E Models

Model	Ethernet Connection	Modbus Client Connections	Other Connections
DXMR90-X1E	Two female M12 D-Code Ethernet Connectors	Four female M12 connections for Modbus client connections	One male M12 (Port 0) for incoming power and Modbus RS-485 and one female M12 for outgoing power and serially connecting Port 0 signals.

Hardware Overview

The DXMR90-X1E Series Industrial Controller can have multiple configurations. The DXMR90-X1E will have a model number label on the housing. Use the model number to identify which boards are included in the controller.



Automation Protocols

The DXMR90-X1E Series Industrial Controller supports the following automation protocols.

EtherNet/IP™

By default, EtherNet/IP is disabled. Configure the DXMR90-X1E 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-X1E manages five separate physical ports running the Modbus RTU protocol. The DXMR90-X1E is the Modbus Client when operating the Modbus client RTU port (port 1–4). The DXMR90-X1E uses the client Modbus RTU bus to communicate with locally connected Modbus server devices.

The other Modbus RTU port (port 0) is used by a host system to access the DXMR90-X1E as a server device. The server Modbus RTU port allows access all the internal local registers concurrently with the client RTU port. Port 0 can be configured as a Modbus Client Port using the DXM Configuration Software but is defined as a server port by default.

Configure the port parameters using the DXM Configuration Software.

Modbus TCP/IP

A host system acting as a Modbus client can access the DXMR90-X1E using the Modbus TCP/IP protocol over Ethernet. Standard Modbus TCP port 502 is used by the DXMR90-X1E for all Modbus TCP/IP requests.

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

By default, the DXMR90-X1E is configured as a Modbus TCP/IP Server. To configure the DXMR90-X1E as a Modbus TCP Client, Modbus TCP must be enabled in the DXM Configuration Software and sockets must be defined to point the DXMR90-X1E to up to 5 Servers.

PROFINET®

By default, PROFINET is disabled on the DXMR90-X1E. To configure the DXMR90-X1E for PROFINET communications, PROFINET must be enabled using the DXM Configuration Software. The DXMR90-X1E 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-X1E.

Modbus Overview

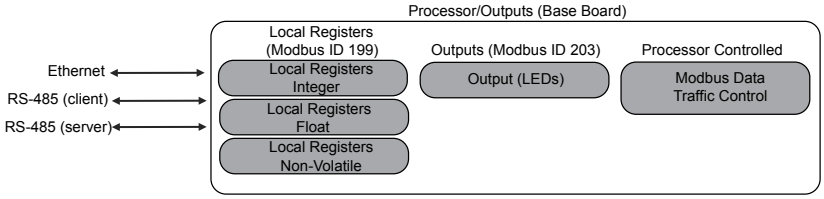
The DXMR90-X1E Series Industrial Controller uses internal 32-bit registers to store information. The processor's internal Local Registers serve as the main global pool of registers and are used as the common data exchange mechanism. External Modbus device registers can be read into the Local Registers or written from the local data registers.

The DXMR90-X1E, as a Modbus client or server 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.

DXMR90-X1E Modbus overview



DXMR90-X1 Modbus Registers

The DXMR90-X1E Series Industrial Controller may have up to two internal Modbus server addresses:

Internal Modbus IDs (factory default)

Modbus 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 ID (199) is fixed for access via ScriptBasic or MicroPython. When accessing the Local Registers through an external Modbus RTU Client, the Server Port (Port 0) ID can be changed using the DXM Configuration Software. Connected devices can use any Modbus ID. For a complete list of registers, see "Internal Local Registers (Modbus ID 199)" on page 15.

Modbus registers for internal local registers (Modbus 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

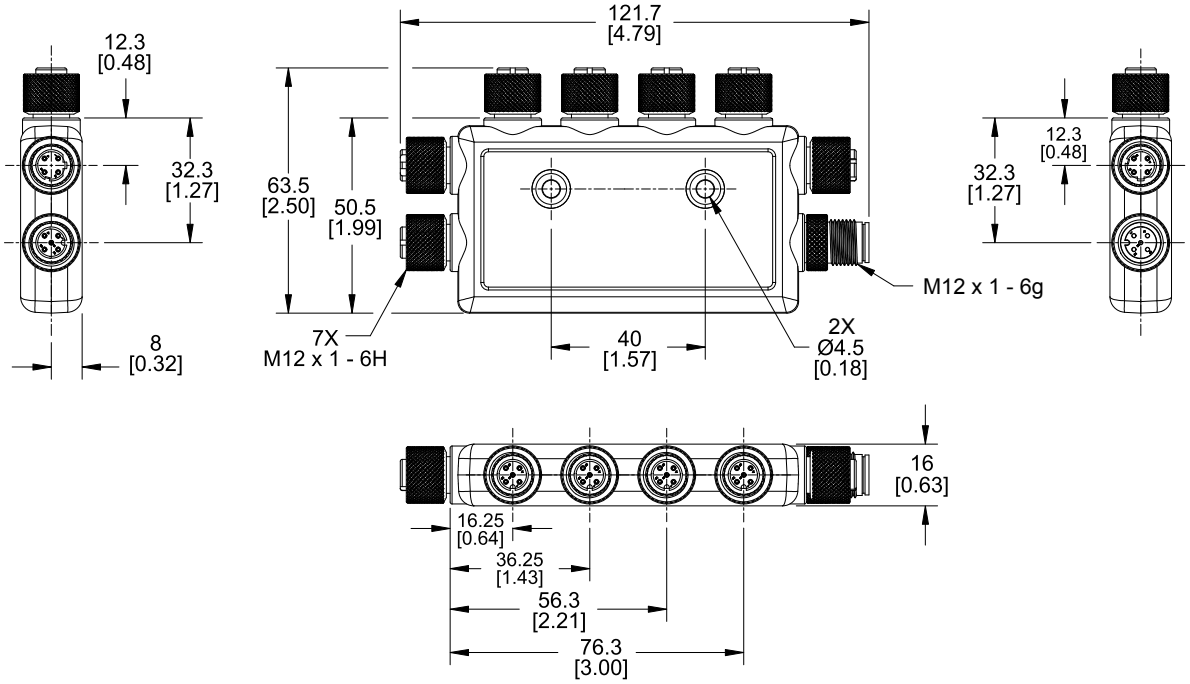
Modbus registers for the LCD board (Modbus ID 203)

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	

DXMR90-X1E Dimensions

All measurements are listed in millimeters, unless noted otherwise. The measurements provided are subject to change.

DXMR90-X1E dimensions



Chapter Contents

Chapter 2 Quick Start Guide

Apply Power to the Controller

Follow these instructions to apply 12–30 V DC power to the DXMR90-X1E using a wall plug.

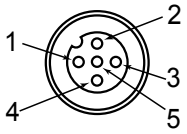
Required equipment:

- DXMR90-X1E Series Industrial Controller
- **PSW-24-1** Wall plug power supply; 24 V DC, 1 A (or equivalent 24 V DC M12 power supply)

1. Connect the **PSW-24-1** power supply to the male M12 connector on the DXMR90-X1E, Port 0.
2. Plug in the **PSW-24-1** wall plug power supply.

DXMR90-X1E Wiring

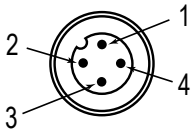
Ports 0-4 female connector

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

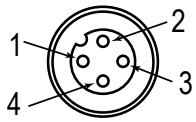
CAUTION:

- Wiring devices incorrectly will cause electrical damage.
- Do not apply more than 12 volts on pins 2 or 4 for ports 1 through 4.

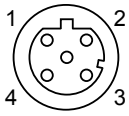
Port 0 male connector

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 / -

Port 0 female connector

Port 0 4-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 / -

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

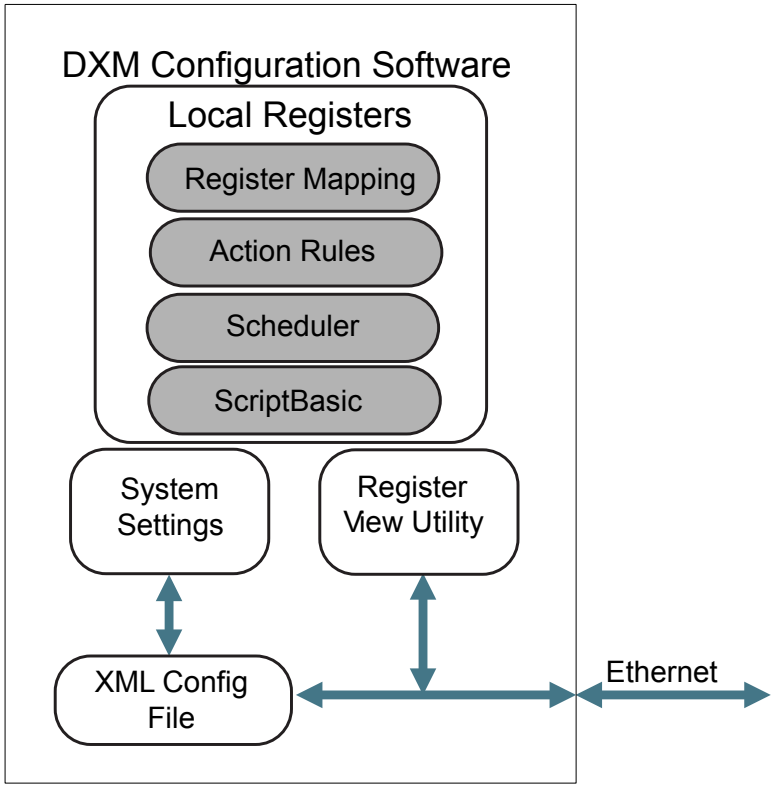
Configuration Instructions

DXM Configuration Software

Configure the DXMR90-X1E using the configuration software. Use this software to customize your configuration and to process data from the Controller.

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).

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-X1E, 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-X1E for operation.

> **IMPORTANT:** The DXMR90-X1E Series Industrial Controller comes preloaded with a default configuration XML file. You can download the default XML on the product page for the DXMR90-X1E.

This quick start guide outlines the basic operations to set up a DXMR90-X1E using the configuration software. For a more comprehensive explanation of features, refer to the DXM Configuration Software Instruction Manual (p/n 209933).

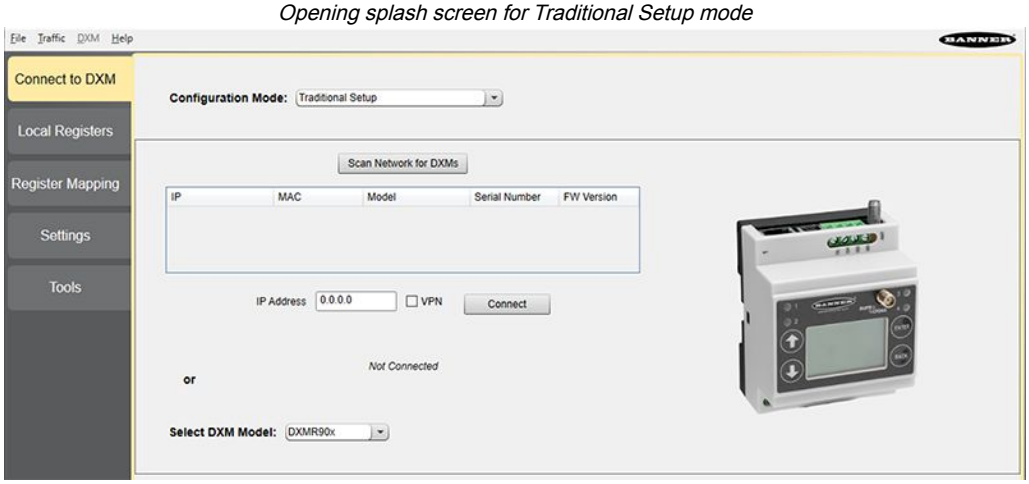
Configuring the DXMR90-X1E Controller

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.

Opening splash screen for Traditional Setup mode



When the **Select DXM Model** drop-down is set to DXMR90-X1E, 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 configuration software. Compatibility is checked before configuration files are uploaded to the device.

Configuration Example: Reading Registers on a Modbus Server 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 server device and save the data into the local registers.

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.

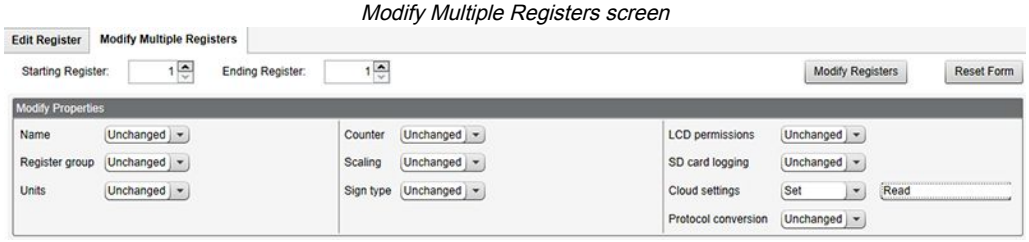
Modify Multiple Registers

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 the parameter. Other selections will appear based on the parameter.

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 Settings** to **Read**.

   If the **Cloud Settings** 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.

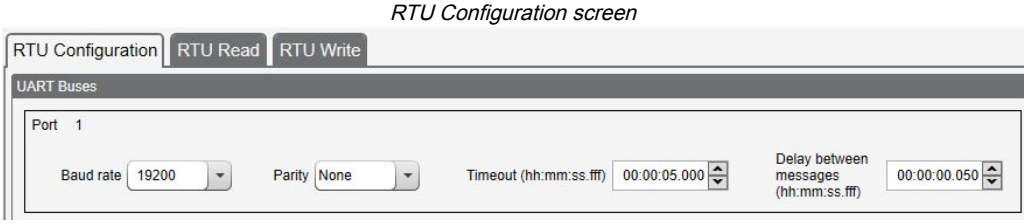
Create an RTU Read Rule Using Controllers with Multiple Client Serial Ports

Follow these steps to create a new read rule.

This example creates a read rule to read six registers (1 through 6), from Port 1 Modbus ID 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.

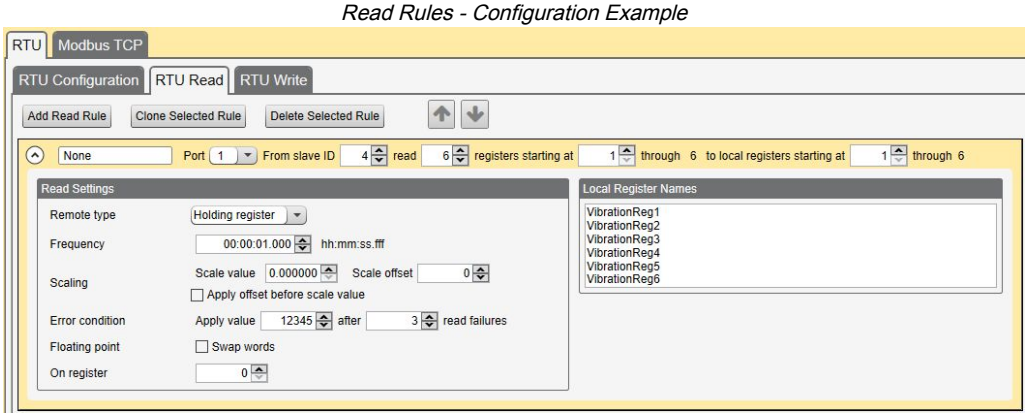
RTU Configuration screen



   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 server devices.
   - The **Timeout** controls how long the DXMR90-X1E 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 Modbus 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.

Read Rules - Configuration Example



Baud Rate

Defined for both the Modbus client and server

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

Delay between messages

Applies to the Modbus client 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 client and server

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

Timeout

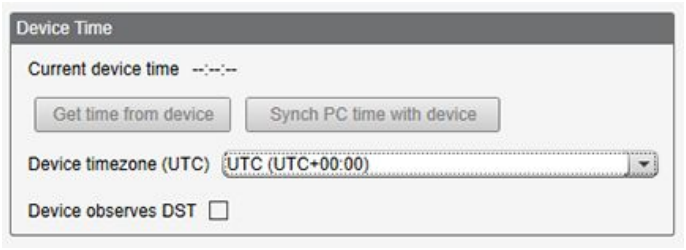
Applies to the Modbus client port

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 server 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.

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-X1E's IP address.

By default, the DXMR90-X1E 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 **Static IP** is selected, enter the **IP address**, **Subnet**, and **Gateway address** as desired.
- If **DHCP** is selected, the **IP address**, **Subnet**, and **Gateway address** are grayed out and not configurable. Changing the IP Address to **DHCP** can make it so the DXM cannot be reached. Before changing this to **DHCP**, you MUST have a server that is going to assign an IP Address to the DXMR90-X1E.

4. Save your changes to the configuration file (**File › Save**).
5. Upload the configuration file to your controller (**DXM › Send Configuration to DXM**).

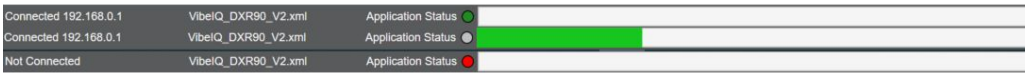
Save and Upload the Configuration File

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 software 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.

Status indicator bar



- If the Application Status indicator is red, close and restart the DXM Configuration Software, 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 gray and the green status bar is in motion, the file transfer is in progress.

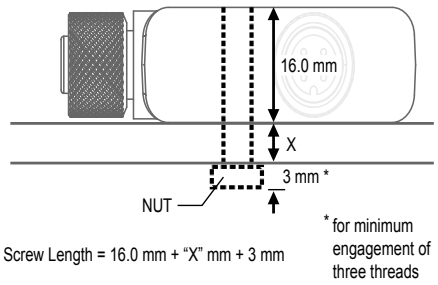
After the file transfer is complete, the device reboots and begins running the new configuration.

Mechanical Installation

Install the DXMR90-X1E to allow access for functional checks, maintenance, and service or replacement. Do not install the DXMR90-X1E in such a way to allow for intentional defeat.

Fasteners must be of sufficient strength to guard against breakage. The 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-X1E accepts M4 (#8) hardware.

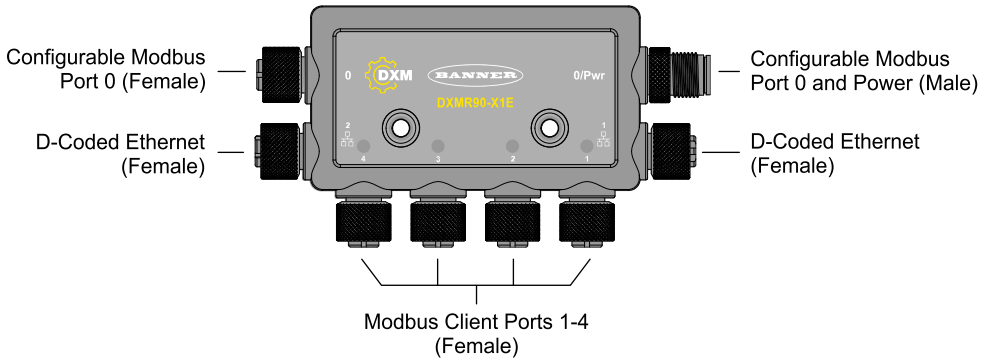
See the figure below to help in determining the minimum screw length.



CAUTION: Do not overtighten the DXMR90-X1E's mounting screw during installation. Overtightening can affect the performance of the DXMR90-X1E.

Chapter Contents

Chapter 3 Controller Connections



Ethernet

Before applying power to the DXMR90-X1E, verify the Ethernet cable is connected.

The Ethernet connection supports the DXM Configuration Software, Modbus/TCP, PROFINET, and EtherNet/IP. ScriptBasic also has access to Ethernet for custom programming. Use the software 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.

Internal Local Registers (Modbus ID 199)

The main storage elements for the DXMR90-X1E 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.

Modbus registers for internal local registers (Modbus 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-X1E, as a Modbus client device or a Modbus server 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 with 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-X1E 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-X1E. 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).

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-X1E 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-X1E 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	

Continued on page 17

Continued from page 16

Registers	Definition	
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	
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.

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.

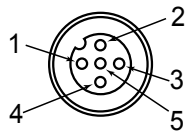
Reset codes

Reset Code	Definition
0	Undefined
1	Unknown
2	General
3	Brownout
4	Watchdog
5	User
6	Software
7	Return from backup mode

Connecting to Remote Modbus Devices

The DXMR90-X1E is configured with four independent Modbus Client 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.

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

Modbus Client and Server Ports

The DXMR90-X1E can be a Modbus RTU client device to other server devices and can be a Modbus server device to another Modbus RTU client. The DXM uses the ports 1–4 as Modbus RTU client ports to control external server devices. All wired devices connected to the client RS-485 port must be server devices.

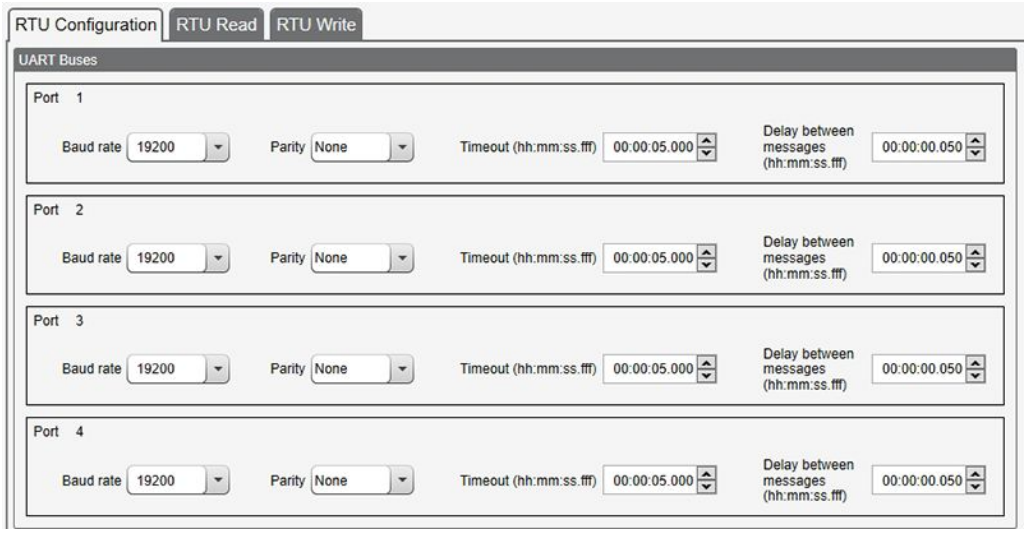
- As a Modbus RTU client device, the DXMR90-X1E controls external servers connected to ports 1–4
- As a Modbus RTU server device, the DXMR90-X1E local registers can be read from or written to by another Modbus RTU client device via port 0.

The Modbus RTU server connection, port 0, is controlled by another Modbus client device that is not the DXMR90-X1E. The server port is used by an external Modbus client device that will access the DXMR90-X1E as a Modbus server Device. Use the DXM Configuration Software to define the operational settings for both the Modbus RTU client ports 1–4 and the Modbus RTU server port 0.

Set the Client and Server 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.

RTU Configuration screen for ports 0-4

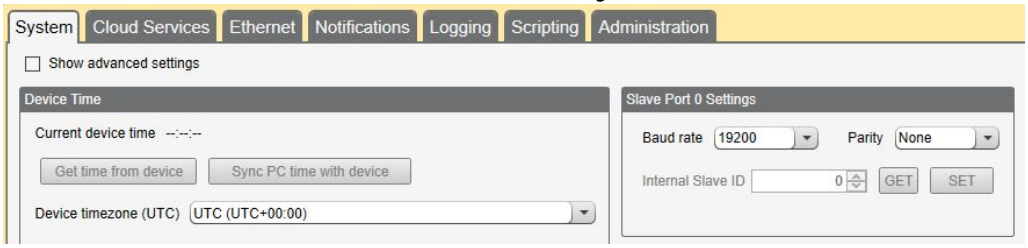


1. Define the **Port** settings to be compatible with the connected devices.
   a. Go to the **Register Mapping › RTU › RTU Configuration** screen.

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 server devices.
   - The **Timeout** controls how long the DXMR90-X1E 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 server parameters for Port 0, go to **Settings › System › Server Port 0 Settings**.
3. Modify the Baud Rate, Parity, and change the Internal Server ID.
   The Internal Server ID is the Modbus ID that an external Modbus Client will access to read/write to the local registers on the DXMR90-X1E.

Server Port 0 Settings



Chapter Contents

Chapter 4 Working with Modbus Devices

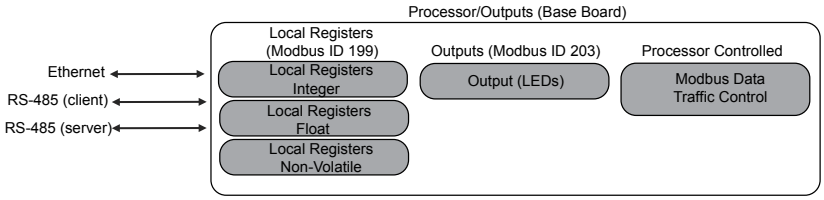
The DXMR90-X1E has five physical RS-485 connections using Modbus RTU protocol.

The client Modbus RS-485 ports are for the DXMR90-X1E to act as a Modbus client device to control external Modbus server devices.

The Modbus client RS-485 ports are labeled Port 1–4. The Modbus server port is used when another Modbus client device wants to communicate with the DXMR90-X1E when the DXMR90-X1E is a Modbus server device.

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

DXMR90-X1E system overview



The DXMR90-X1E has dual Modbus roles: a Modbus server device and a Modbus client device. These run as separate processes.

The Modbus server port can only access the DXMR90-X1E local registers. To operate as a Modbus server device, the DXMR90-X1E needs to be assigned a unique Modbus server ID as it pertains to the host Modbus network. This server ID is separate from the internal Modbus server IDs the DXMR90-X1E uses for its own Modbus network. The DXM Modbus server ID and other Modbus server port parameters are defined by using the configuration software.

The DXMR90-X1E operates the Modbus client ports. Each device on a client port must be assigned a unique server ID. There are server IDs that are reserved for internal devices in the DXMR90-X1E. Each device that shares a client port must have a unique ID. Devices on separate ports may have the same ID.

Internal Modbus IDs (factory default)

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

Assigning Modbus IDs

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

To set the Modbus server parameters for Port 0, go to **Settings › System › Server Port 0** settings. Here you can modify the **Baud Rate**, **Parity**, and change the **Internal Server ID**. The **Internal Server ID** is the Modbus ID that an external Modbus Client accesses to read/write to the local registers on the DXMR90-X1E.

DXM Client Configuration—When the DXM operates as a Modbus client, 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 port(s).

Modbus Operation

All Modbus transactions are managed by a central Modbus engine.

If there are Modbus messages intended for a Modbus server 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 client 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 server devices in the network.

Modbus Communication Timeouts

A Modbus timeout is the amount of time a Modbus server is given to return an acknowledgment of a message sent by the Modbus client. If the Modbus client waits for the timeout period and no response is seen, the Modbus client 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-X1E. Special considerations need to be made to set the timeout parameter when the DXMR90-X1E 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-X1E, the **Communications Timeout** parameter is the maximum amount of time the DXMR90-X1E should wait after a request is sent until the response message is received from the Modbus server 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 five (5) seconds.

Modbus TCP Client

The DXMR90-X1E 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.

Chapter Contents

Chapter 5 Optional Configuration Steps

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-X1E. Reboot the DXMR90-X1E to activate a new schedule.

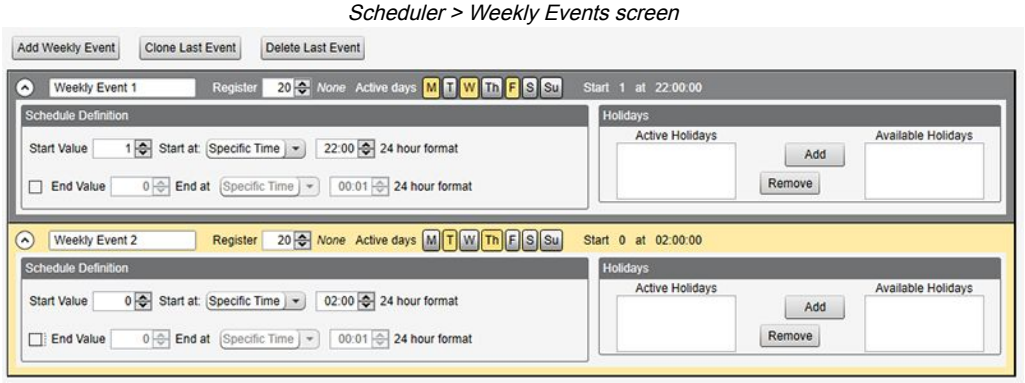
If power is cycled to the DXMR90-X1E in the middle of a schedule, the DXMR90-X1E looks at all events scheduled that day and processes the last event before the current time.

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.

Create a Weekly Event

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

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.

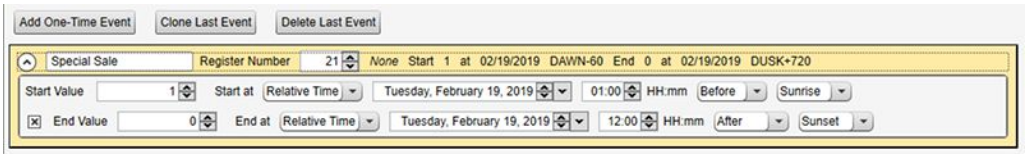
Start and end times can be specified relative to sunrise and sunset, or set to a specific time within a 24-hour period. When using sunrise or sunset times, set the GPS coordinates on the device so it can calculate sunrise and sunset.

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 **Tools › Scheduler › One Time Events** screen.

Scheduler > One Time Events screen

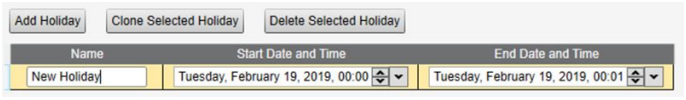


1. Click on **Add One Time Event**.
   A new one-time event is created.
2. Click on the arrow to expand the parameters into view.
   The user-defined parameters are displayed.
3. Name your one-time event by clicking on the name link and entering a name.
4. Enter the local register.
5. Enter the starting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

Create a Holiday Event

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

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.

Authentication Setup

The DXMR90-X1E has three areas that can be configured to require 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.

Set the Controller to use Authentication

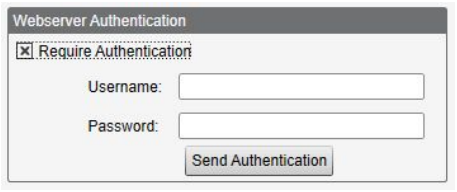
The DXMR90-X1E can be configured to send login and password credentials for every HTTP packet sent to the webserver. This provides another layer of security for the webserver data.

Configuration requires both the webserver and the DXMR90-X1E 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-X1E.

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.

   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.

   Web Server Authentication screen

4. Click on **Send Authentication**.

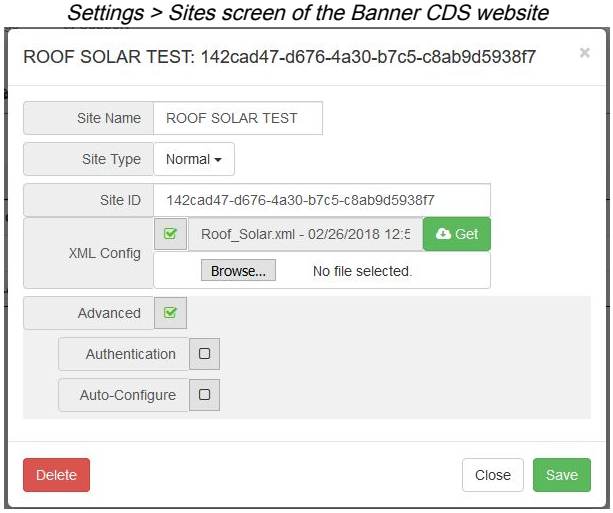
   The controller must be connected to the PC for this operation to succeed.

   The data transmits directly to the DXMR90-X1E's non-volatile memory. If successful, a pop-up window appears, asking to reboot the device.
5. Select **Yes** to reboot the device.

Set the Web Services to Use Authentication

1. At the Key definition for "WEB_SERVICE" not found in the DITA map. website, go to **Settings › Sites**.
2. To edit the site settings, click **Edit** on the line of the site name.

   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 Key definition for "WEB_SERVICE" not found in the DITA map. website.

Controller Configuration Authentication

The DXMR90-X1E 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.

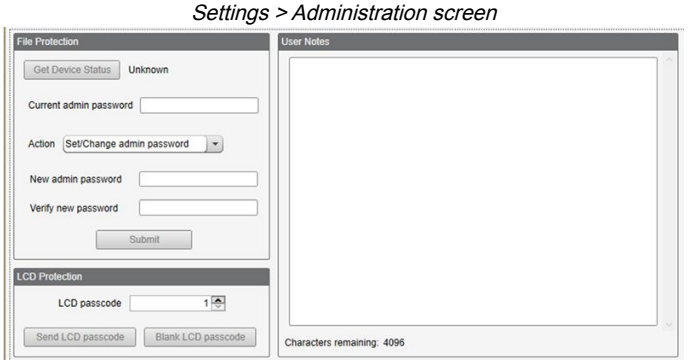
With the DXMR90-X1E connected to the PC, click **Get Device Status**. The DXMR90-X1E status displays next to the button.

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, you are required to enter the current password and the DXMR90-X1E must be connected to the PC.

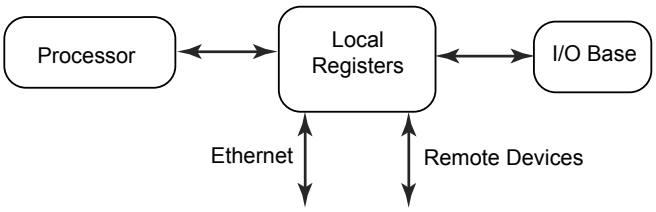
Settings > Administration screen



Register Flow and Configuration

The DXMR90-X1E 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.

Register flow



Basic Approach to Configuration

When programming an application in the DXMR90-X1E, first plan the overall data structure of the Local Registers. The Local Registers are the main storage elements in the DXMR90-X1E. 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 servers, 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 day of the week. The scheduler can also create events based on sunrise or sunset.

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-X1E, viewing the Local Registers can help you to understand the application's operation. This utility can also access data from remote devices and LED registers.

Saving and Loading Configuration Files

The DXM Configuration Software saves its configuration information in an 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-X1E. The DXM Configuration Software uploads the configuration file saved on the PC to the DXMR90-X1E; it will not send the configuration loaded in the tool.

Uploading or Downloading Configuration Files

The DXMR90-X1E requires a XML configuration file to become operational. To upload or download configuration files, connect a computer to the DXMR90-X1E using the Ethernet port. Then use the **Upload Configuration to Device** or **Download Configuration from Device** under the **Device** menu.

EtherNet/IP™ Configuration

The DXMR90-X1E 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)

Configuring the Host PLC

On the host PLC, install the DXMR90-X1E 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-X1E. The host system sees the DXMR90-X1E 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.

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-X1E local register (DXM).

Define a DXM local register as **EIP DXM › Originator** when that register data will be sent from the DXMR90-X1E (DXM) to the host PLC (Originator).

Data from an EIP controller in assembly instance 112 is data destined for the DXMR90-X1E 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
4	9900		5	9900

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

Data from the DXMR90-X1E local registers is sent to the EIP controller using assembly instance 100. Each local register in the DXMR90-X1E 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

Set up the Email

The DXMR90-X1E 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). 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-X1E.
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.

Mail Server Authentication

Complete the mail server settings to have the DXMR90-X1E send email alert messages.

The SMTP password is stored in the DXMR90-X1E, not the XML configuration file. Use the **Settings › Notifications** screen to complete this configuration.

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.

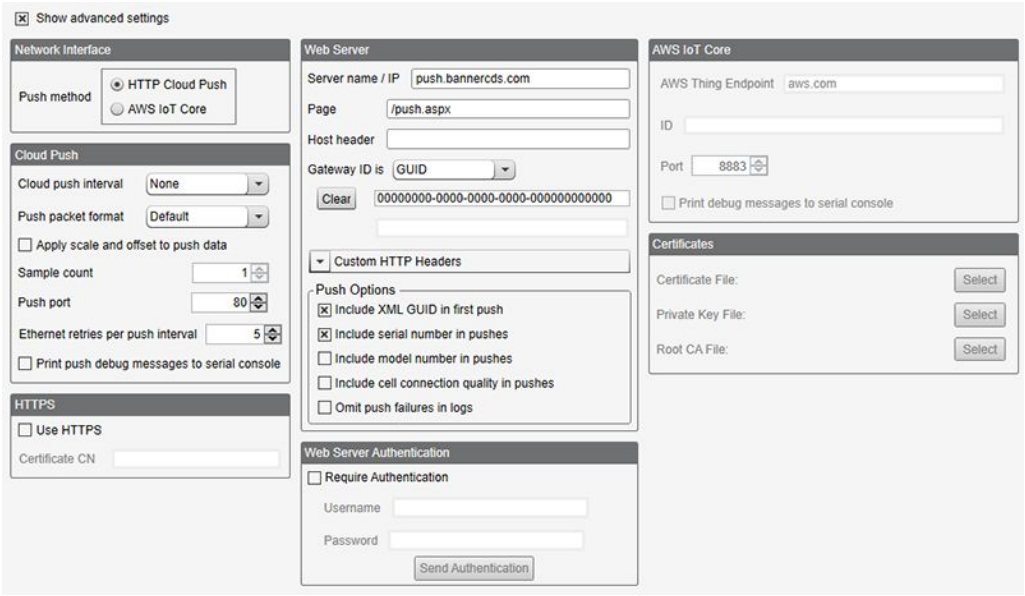
After entering the user name and password, click on **Send SMTP Password** to save the username and password to the DXMR90-X1E. The DXMR90-X1E must be connected to the PC to complete this operation. If successful, a pop-up window appears, asking to reboot the device. Select **Yes** to reboot the device.

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.

Cloud Services screen

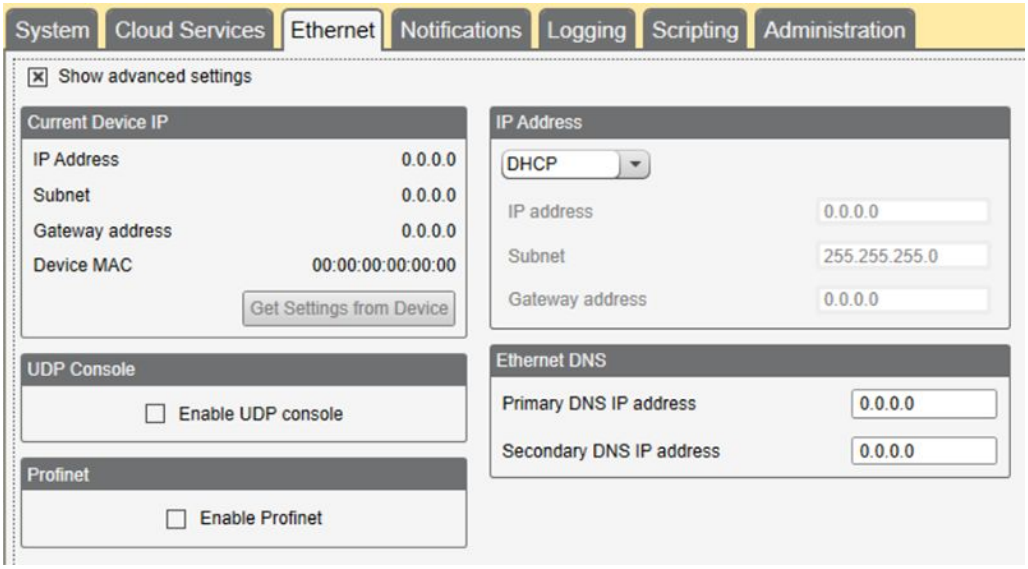


Configure your Ethernet Connection

To send email based on a threshold rule, 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-X1E 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-X1E uses a public service to resolve Domain names, but if the network connection does not have Internet access, the DNS settings may be required.

Settings > Ethernet screen



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.

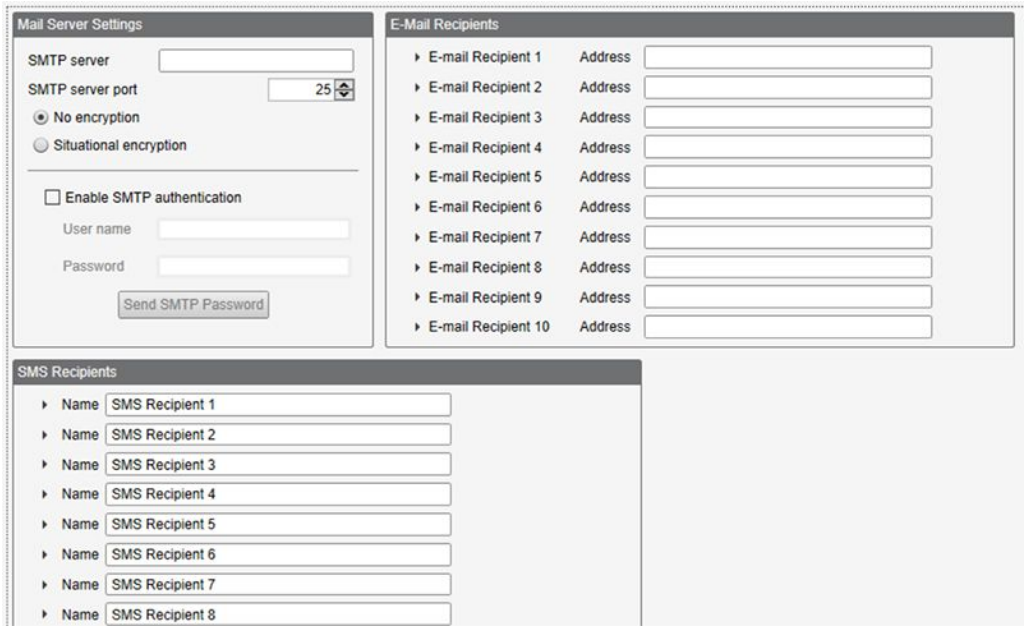
The default SMTP port is 25 but may need to be adjusted for Ethernet-based networks. Note that many facilities block port 25. Port 587 is another common SMTP submission port.

The SMTP password is not stored in the XML configuration file, but on the DXMR90-X1E. After the password is entered, click on **Send SMTP Password** to send it to the DXMR90-X1E. The password is stored in non-volatile memory, so reboot the DXMR90-X1E 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.

Email settings



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

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 become 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).

Push Retries

Ethernet—The DXMR90-X1E can be configured to send register data packets to a web server. When the Ethernet communications path is not operating, the DXMR90-X1E retries the send procedure. With an Ethernet-based network connection, the DXMR90-X1E retries a message five times. The five retry attempts immediately follow each other. After all attempts are exhausted, the register data packet is lost. At the next scheduled time, the DXMR90-X1E attempts to send only the new data. Any past data that the DXMR90 was unable to push is lost and cannot be recovered. Using SSL on Ethernet will have no retries.

Event/Action—Event-based pushes caused by Action rules sent using email follow the same process when failures occur, based on the network connection.

Email—There are no retries for emails that fail to be sent from the DXMR90-X1E.

Chapter Contents

Chapter 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 DXMR90-4K, DXMR90-X1, DXMR110-8K, DXM700, DXM1000, and DXM1200 Controller models support PROFINET IO.

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.

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.

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 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 IO-Link ports in the XML according to the modules selected for each port.

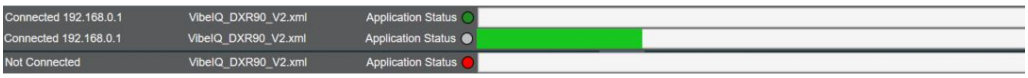
Save and Upload the Configuration File

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 software 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.

Status indicator bar



- If the Application Status indicator is red, close and restart the DXM Configuration Software, 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 gray and the green status bar is in motion, the file transfer is in progress.

After the file transfer is complete, the device reboots and begins running the new configuration.

Slots and Modules for the DXMR90-X1, DXM700, DXM1000, and DXM1200 PROFINET

There are nine slots to accommodate the DXM Controller data.

Slots for input and output values

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

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.

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

Slots 4 through 6

Module	Notes
Input Float 512	Allowed in slots 4-6, Module Identifier= 0x34

Continued on page 33

Continued from page 32

Module	Notes
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

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 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

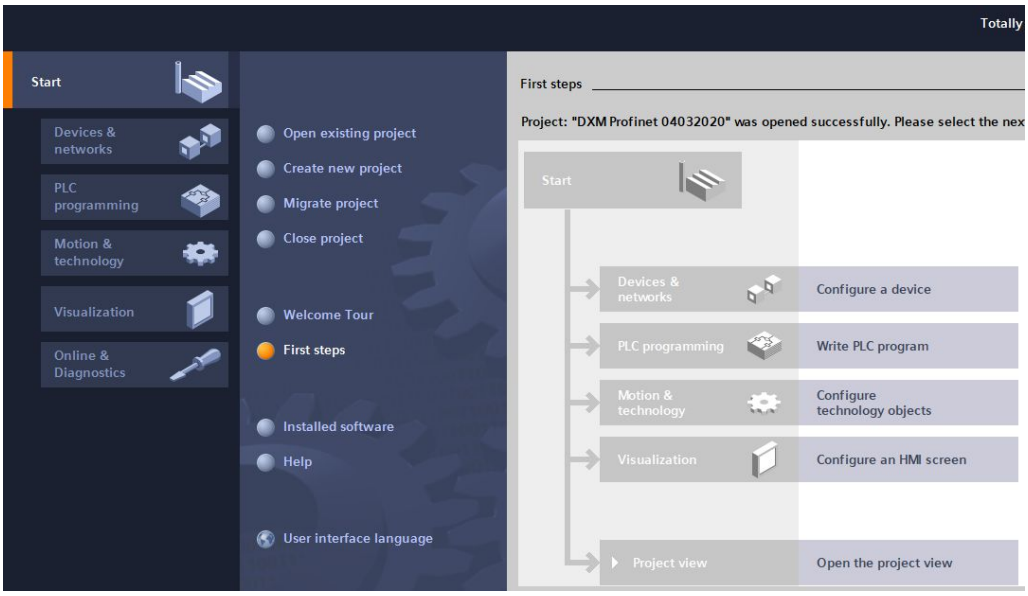
Configuration Instructions

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.

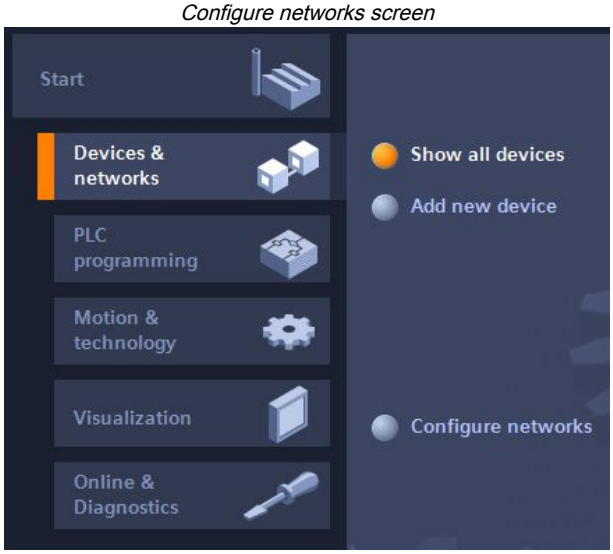
1. Download the GSD file from www.bannerengineering.com.
2. Launch the Siemens TIA Portal (v14) software.
3. Click **Open existing project**.
4. Select a project and open it.
5. After the project is uploaded, click **Devices & networks**.

Devices and networks screen



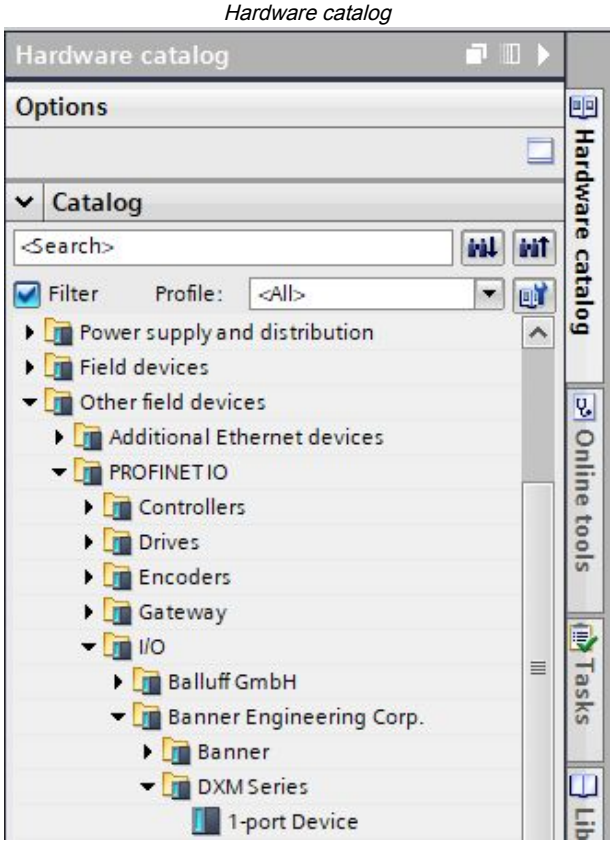
6. Click **Configure networks**.

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**.

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**.

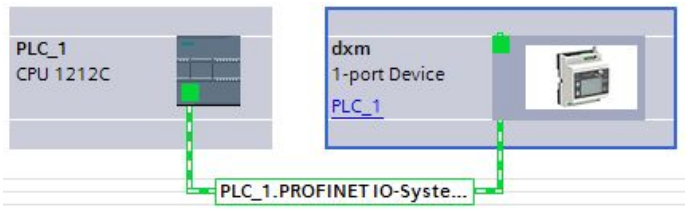
If the DXM GSD file does not install properly, save the log and contact Banner Engineering Corp.

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**.

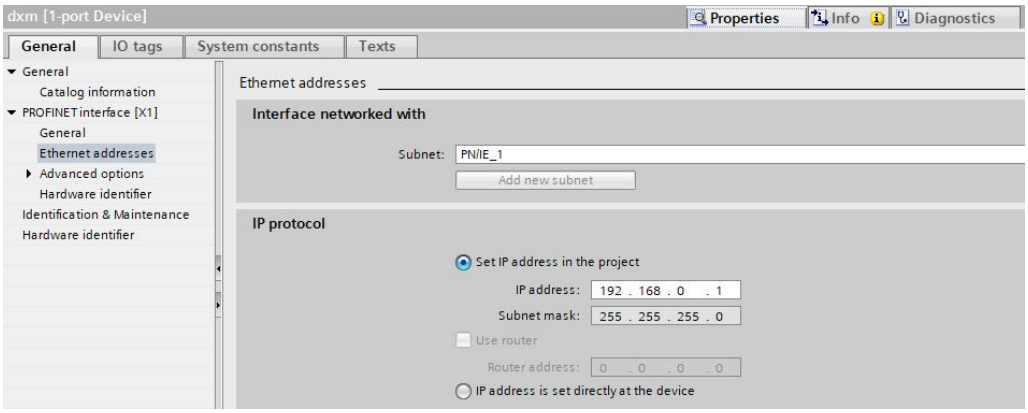
Network view



The **Network view** displays.

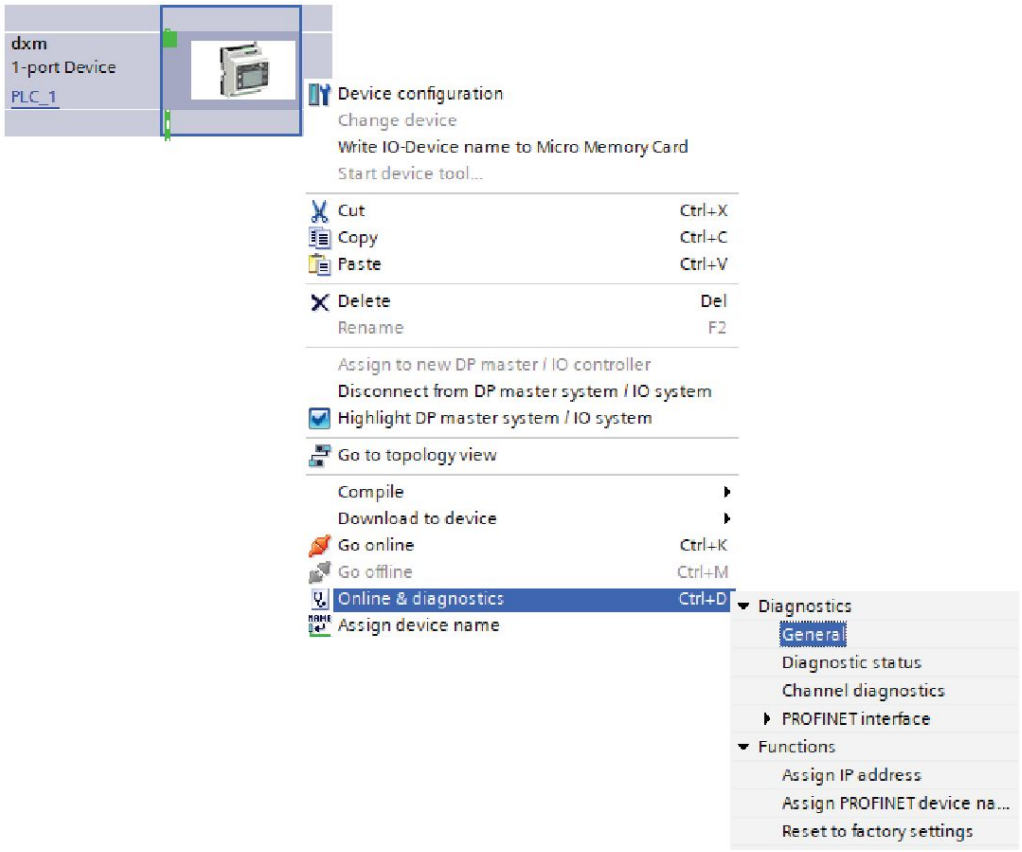
5. Double-click on the DXM icon to open the **Device view** screen.
6. Click on the DXM icon in the graphic area of the **Device view** screen.
   The **Module properties** window displays and the module can now be configured.
7. Click **Properties**.
8. Click **General**.
9. Select **PROFINET Interface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

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**.

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.

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).

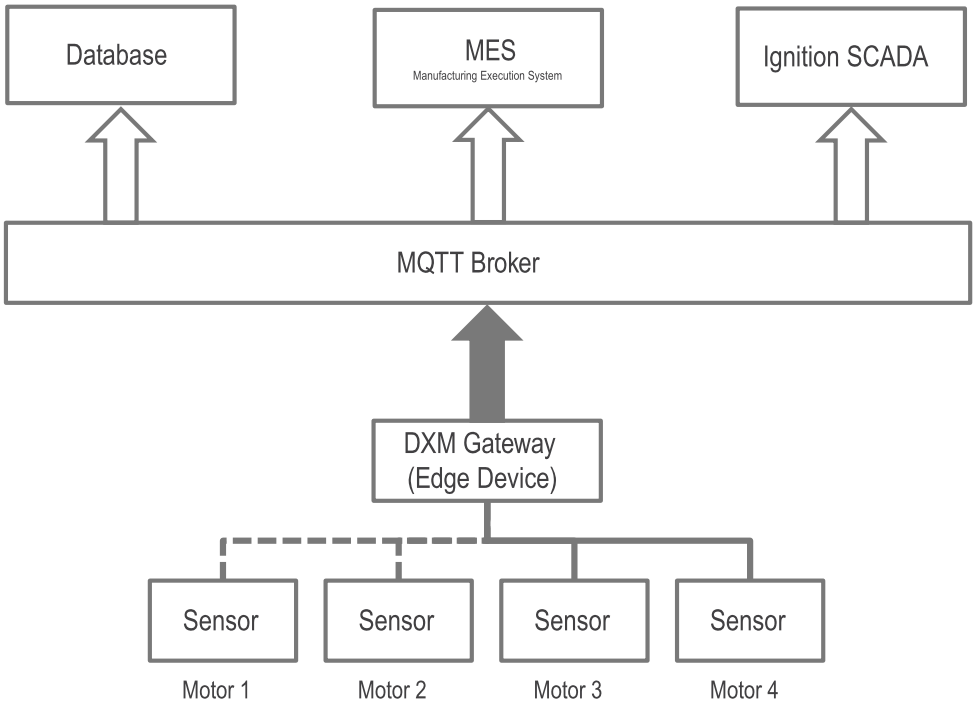
1. Open a project and click on **Devices & networks**.
   The Network view displays.
2. Right-click on the DXM icon and select **Assign device name**.
   The **Assign PROFINET device name** window displays. The software searches for devices of the same type.
3. Enter the desired name in the **PROFINET device name** field. Note that each name can be used only once.
4. Click **Assign name**.
   The device now has a PROFINET name.

Chapter Contents

Chapter 7 MQTT Overview

Message Queuing Telemetry Transport (MQTT) messages are efficient and have real-time capabilities, especially in remote monitoring, predictive maintenance, and machinery and equipment control applications. Banner's DXMR90-X1E, DXM1200-X2, DXM700-B1R#, DXM1200-B2R#, and DXM1200-X2R# models support Flat MQTT and Sparkplug™ B profiles.

MQTT flowchart



A Banner DXM can publish to an MQTT broker and various services can subscribe to the MQTT Broker.

For example, a Banner DXM can publish QM30VT2 sensor vibration data from motor 1 to the MQTT broker. That vibration data will be in the broker until the DXM republishes it. At any given time, only one value for each sensor register is in the broker (the broker does not log data). If a user's ignition dashboard needs the vibration data of Motor 1, they subscribe to it in the broker and obtain the latest value. To log the data into a database, you would configure a database that subscribes to the broker and logs the value over time.

Sparkplug B is a framework for MQTT to pre-define topic structures for industrial data. It provides MQTT clients the framework to integrate data from applications, sensors, devices, and gateways within the MQTT infrastructure in a bi-directional and interoperable way. Sparkplug B organizes data for topic namespace, payload, state management (birth and death certificates), store and forward (data buffering), and compression. Sparkplug B serves as an open-source software specification that provides MQTT clients with an interoperability protocol to seamlessly integrate data from various applications, devices, sensors, and other elements of MQTT infrastructure.

Element	Definition	Source
Group ID	A logical identifier for a group of MQTT Nodes	Defined by the user

Continued on page 39

Continued from page 38

Element	Definition	Source
Message Type	Indicates whether the message contains state information, data, or a command and whether it pertains to a node, device, or the primary application	Predefined by SpB specification; cannot be changed by the user
Edge Node ID	Identifies a specific MQTT node	Defined by the user. The Group ID/Edge Node ID combination must be unique
Device ID	Identifies a device attached physically or logically to a node	Optional Field. Defined by the user, if applicable

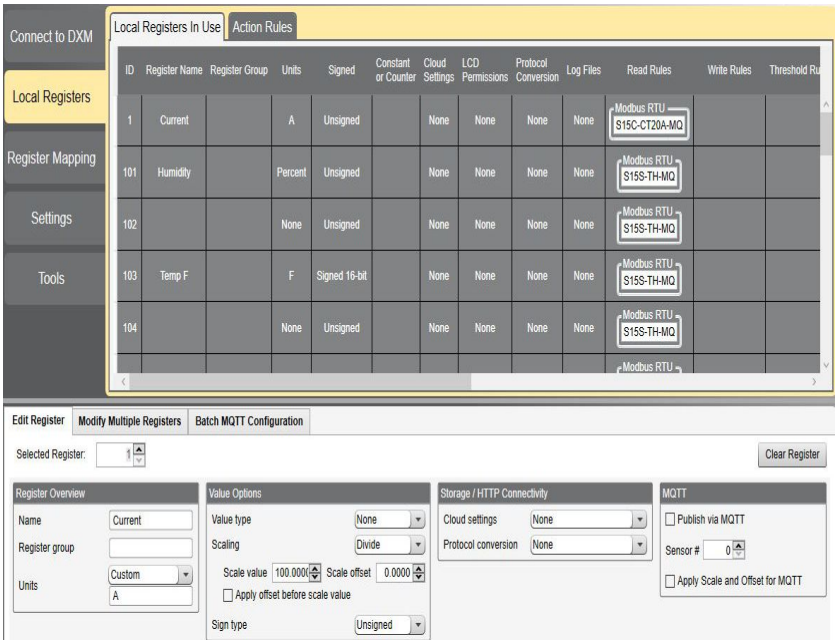
Integrate an R90-X1E to a Broker Using Flat MQTT

Follow these instructions to send data from a Banner DXM Controller to an MQTT Broker using flat MQTT. The examples shown use a QM30VT2 vibration and temperature sensor plugged into port 1 of a DXMR90-X1E Industrial Controller.

These instructions assume familiarity with the DXM Configuration Software.

1. Launch the DXM Configuration Software.
2. On the **Connect to DXM** screen, connect to your DXMR90-X1E Series Industrial Controller by selecting the subnet to scan and the DXMR90-X1E from the download list.

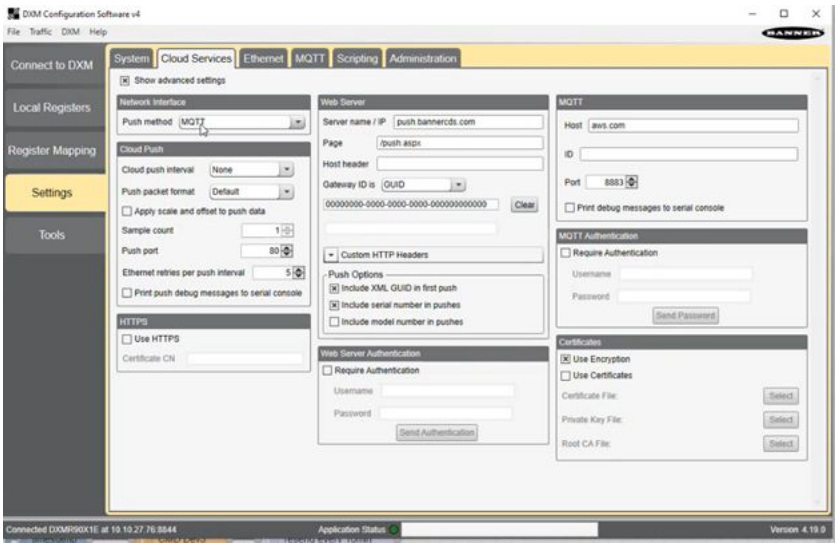
3. Go to the **Local Registers › Local Registers in Use** screen.



4. Configure the local registers required for your application.
    a. Name and configure all the required local registers.

b. Users can configure registers to be published via MQTT individually under the **Edit Register** sub-tab or batched under the **Batch MQTT Configuration** sub-tab.

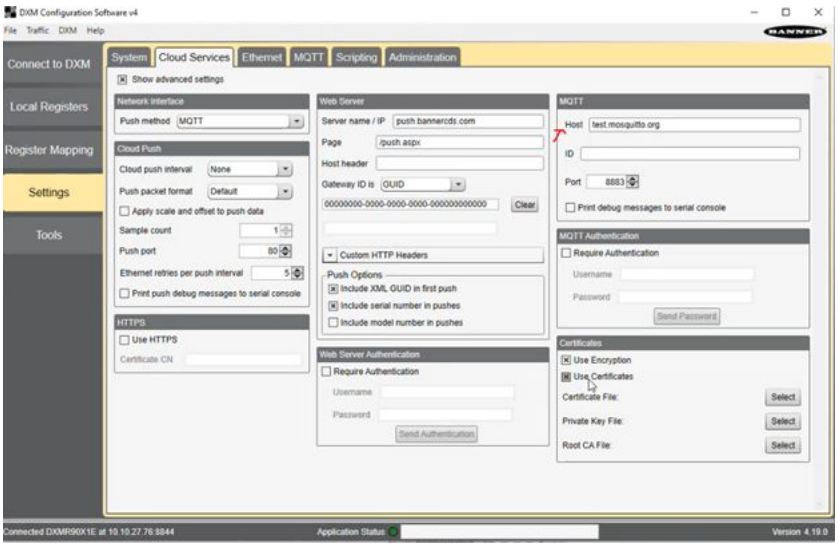
5. Go to the **Settings › Cloud Services** screen.
6. Under the **Network Interface** section, use the drop-down list to set the **Push method** to **MQTT**.



7. Go to the **Local Registers › Local Registers in Use** screen.
8. To batch multiple registers, go to the **Batch MQTT Configuration** sub-tab and set the **Publish via MQTT** to **Set**.
9. Set the **Publish Group** to **Set** and set it to a value.

10. Configure scaling in the **Apply Scale and Offset for MQTT** field, and configure the registers to be published only when the value changes in the **Publish when value changes** field.
11. After making the necessary changes, click on **Modify Multiple Registers** to modify properties.
12. Still in the **Edit Register** sub-tab, select the **Publish via MQTT** checkbox and select the group to publish to. You can use up to 32 groups.
13. Go to the **Settings › Cloud Services** screen and the **MQTT** section.
14. In the **Host** field, fill in the host address.

    The host is your endpoint, which could be several things. Example hosts include a Node Red Broker, your computer's IP address, or test.mosquito.org, which is another broker that lives on the web. Use the ID field to differentiate this DXM from another DXM that might be used. This information is sent via a JSON packet. In the **ID** field fill in an ID. This must be unique if multiple DXM Controllers are going to the same host address or there will be conflicts.



15. If desired, use select **Encryption**.

    When using encryption, certificates are required. Select **Certificates** and add certificates.

16. Under **MQTT Authentication**, fill in a username and password.

    Most applications use MQTT authentication. This password is stored in the DXM in the non-volatile memory, not the XML configuration file. If you do not know if the DXM already has a password, it is best to send a blank password.

17. Click **Send Password**.

18. Go to the **Settings › MQTT** screen.

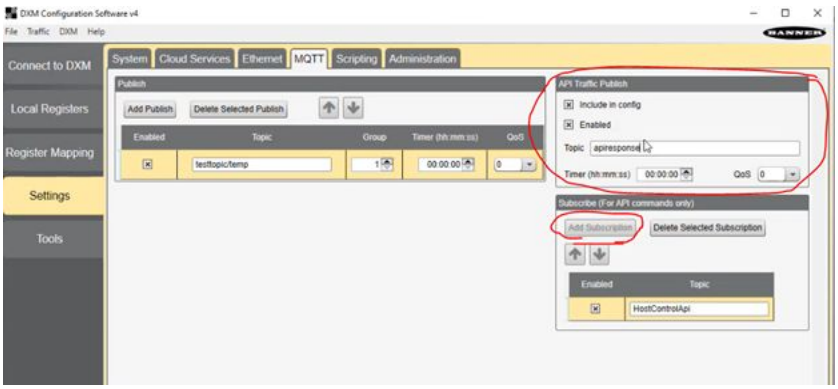
19. Add the desired topics and select **Enabled**.

    Use the **Add Publish** button to add multiple topics. Your local registers will feed information into these topics. If necessary, you may split up your registers into 32 different topics.

20. Enter values into the timers to define how often to send local register values to the topics or enter a 1 in the Quality of Service (QoS) field to only write to the topic upon a local register change of state.

21. To control this device from another broker client, follow these steps:

    The API Traffic Publish is only necessary if you want to see the response from the DXM when a host-controlled message is sent through the subscribe. To control and not monitor the API response, ignore API Traffic Publish area because the DXM still reacts to the changes regardless of the API response.



a. Click **Add Subscription**.
b. Name your subscription.

This allows you to write to a local register from another client.

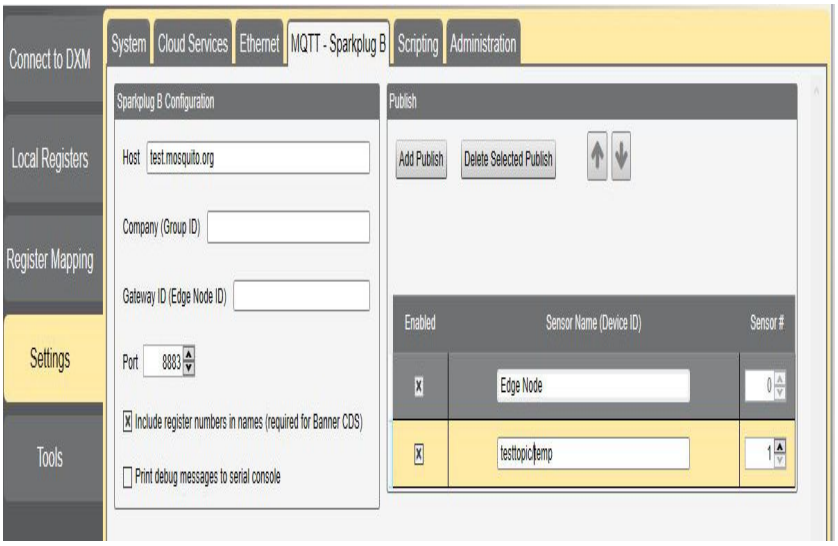
Integrate an R90-X1E to a Broker Sparkplug™ B Profile

Follow these instructions to configure your DXMR90-X1E Series Industrial Controller to use an MQTT broker Sparkplug™ B profile.

These instructions assume you are familiar with the DXM Configuration software.

1. Launch the DXM Configuration Software.
2. On the **Connect to DXM** screen, connect to your DXMR90-X1E Series Industrial Controller by selecting the subnet to scan and the DXMR90-X1E from the download list.

3. Go to the **Settings › Cloud Services** screen.
4. In the **Network Interface** section, select **MQTT - Sparkplug B** from the **Push method** drop-down list.
   An **MQTT - Sparkplug B** tab appears across the top.
5. Go to the **MQTT - Sparkplug B** screen.
   For both flat MQTT and Sparkplug B, port 1883 is used when encryption and certificates are turned off and port 8883 is used when encryption and certificates are turned on.



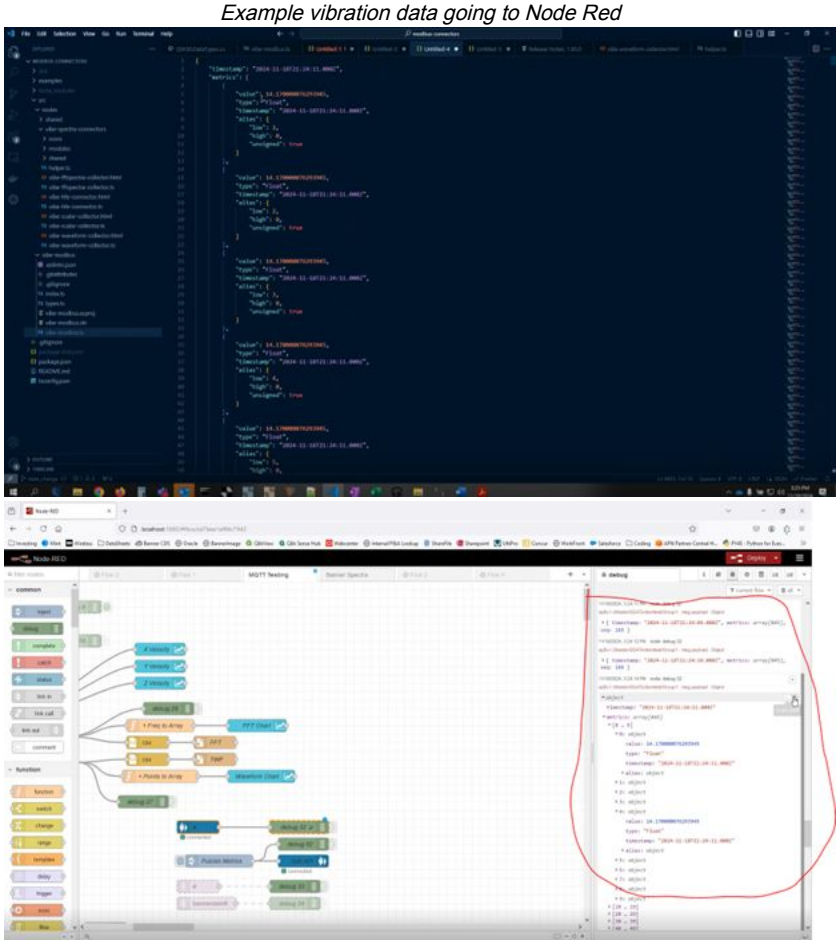
In the **Publish** section, the Edge Node set to Sensor #0 will always be listed. The system will not push sensor data to Sensor 0.

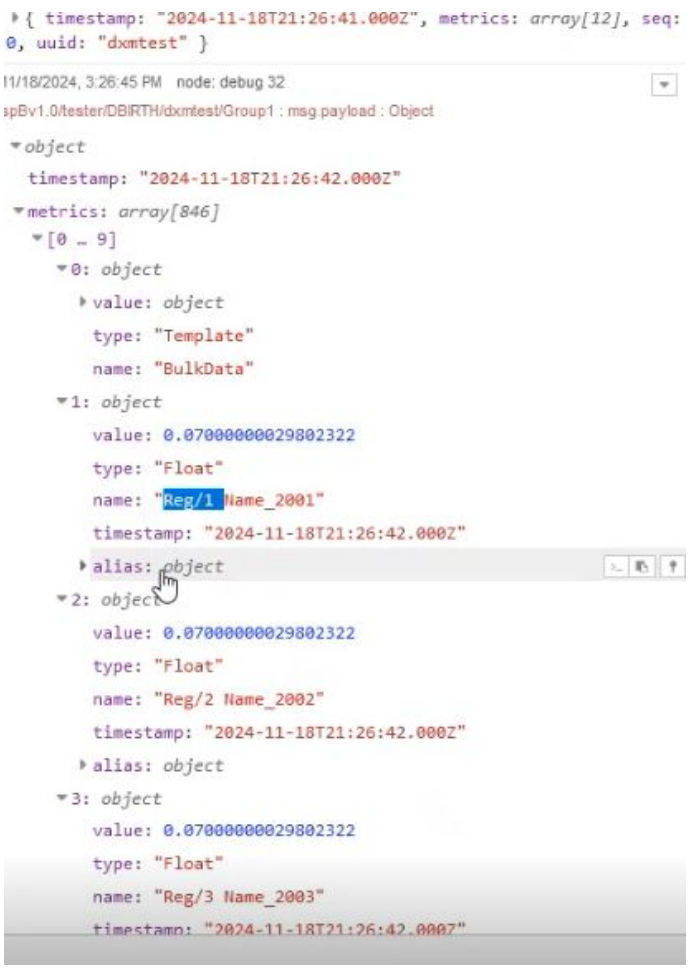
6. Click **Add Publish** to add new publishes to the system and assign a unique sensor name.
7. In the **Sparkplug B Configuration** section, enter the **Company (Group ID)** and **Edge Node ID** to create the full Sparkplug B topic.
   Sparkplug topics following this format: `namespace/group_id/message_type/edge_node_id/[device_id]` with the namespace defined as **spBv1.0**.
   The namespace element defines the structure of the namespace elements and data encoding. The current Sparkplug specification defines two namespaces: one for payload definition A (spAv1.0) and one for payload definition B (spBv1.0).

   The `group_id` in this context is the same as the **Company (Group ID)** field in the DXM Configuration Software's **Setting › MQTT - Sparkplug B** screen.

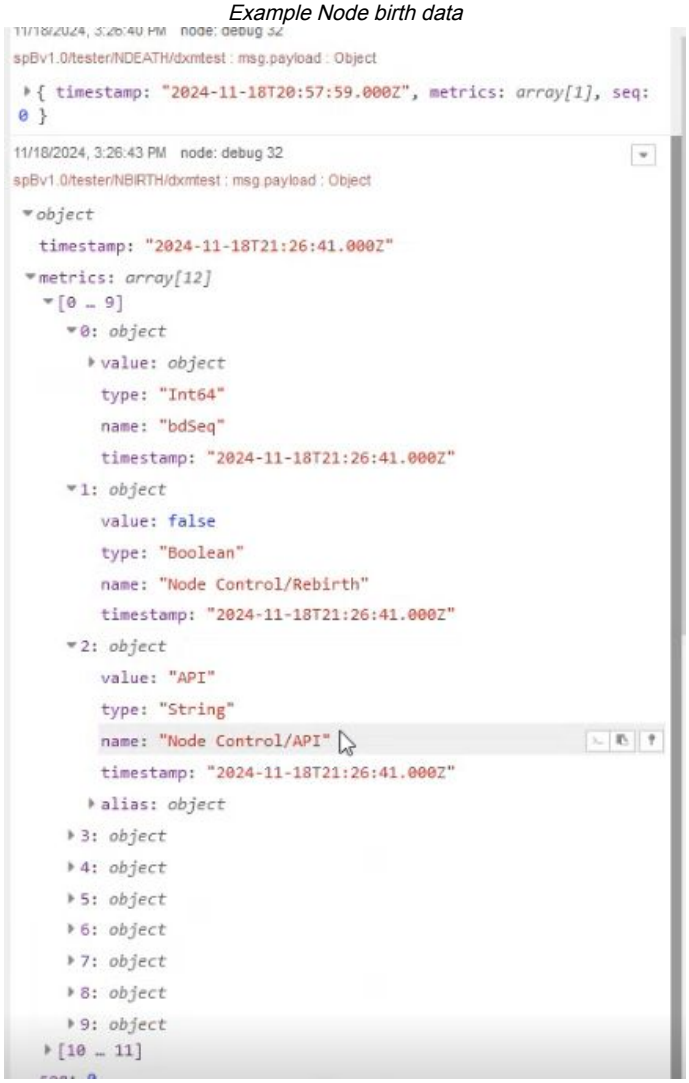
   The `message_type` is delivered here and the DDATA is the device data that is delivered after your topics are published.
8. Enter an `edge_node_ID` into the **Gateway ID (Edge Node ID)** field.
   This value must be unique for each controller to ensure no overlap of data between controllers going to the same host/broker.
9. Enter a `device_ID` into the **Sensor Name (Device ID)** field for each sensor defined. Verify your sensors are enabled.

Example vibration data going to Node Red





Example Node birth data



message_type

NBIRTH - Birth certificate for MQTT EoN nodes
NDEATH - Death certificate for MQTT EoN nodes
DBIRTH - Birth certificate for devices
DDEATH - Death certificate for devices
NDATA - Node data message
DDATA - Device data message
NCMD - Node command message
DCMD - Device command message
STATE - Critical application state message

To publish an API comment, send the comment to NCMD with the group ID and Edge Node ID.

```
Write Local Registers (ex CMD0002, Register 851, 1 register, Value or 2)
{
 "metrics":[
 {
   "name":"Node Control/API",
   "type":"String",
   "value":"CMD0002851,1,1,0,0,0,2"
 }
]
}
```

Chapter Contents

Chapter 8 DXMR90-X1E Accessories

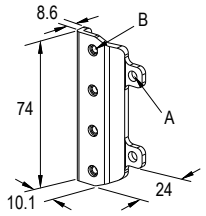
Power Supplies

PSD-24-4—DC Power Supply, Desktop style, 3.9 A, 24 V DC, Class 2, 4-pin M12 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

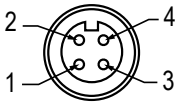
SMBR90S

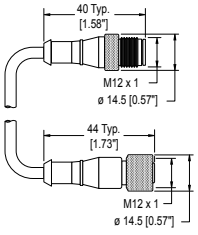
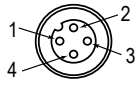
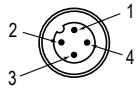
- Stainless steel bracket
- 4x M4-07 pemnuts (B)
- Includes 2x M4 stainless steel hex head screws and flat washers

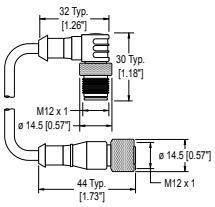
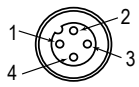
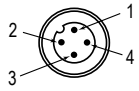
Hole center spacing: A = 40, B = 20
Hole size: A = ø 5

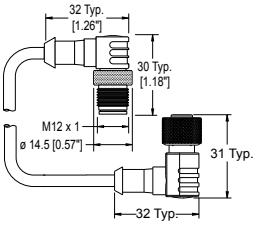
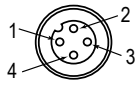
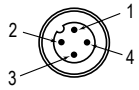


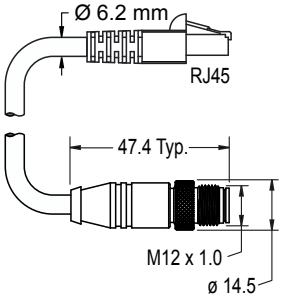
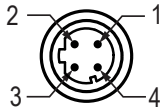
Cordsets

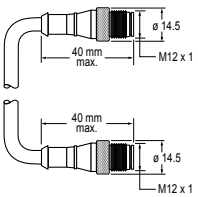
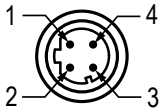
4-Pin M12 Female RS-485 to USB Adapter Cordset, with Wall Plug				
Model	**Length**	**Style**	**Dimensions**	**Pinout (Female)**
BWA-UCT-900	1 m (3.28 ft)	Straight		 1 = Brown 2 = White 3 = Blue 4 = Black

4-pin A-Code Double-Ended M12 Female to M12 Male Cordsets				
Model	**Length**	**Dimensions (mm)**	**Pinouts**	
BC-M12F4-M12M4-22-1	1 m (3.28 ft)		Female  Male 	1 = Brown 2 = White 3 = Blue 4 = Black
BC-M12F4-M12M4-22-2	2 m (6.56 ft)			
BC-M12F4-M12M4-22-5	5 m (16.4 ft)			
BC-M12F4-M12M4-22-8	8 m (26.25 ft)			
BC-M12F4-M12M4-22-10	10 m (30.81 ft)			
BC-M12F4-M12M4-22-15	15 m (49.2 ft)			

4-pin A-Code Double-Ended M12 Female to M12 Male Right-Angle Cordsets				
Model	**Length**	**Dimensions (mm)**	**Pinouts**	
BC-M12F4-M12M4A-22-1	1 m (3.28 ft)		Female  Male 	1 = Brown 2 = White 3 = Blue 4 = Black
BC-M12F4-M12M4A-22-2	2 m (6.56 ft)			
BC-M12F4-M12M4A-22-5	5 m (16.4 ft)			
BC-M12F4-M12M4A-22-8	8 m (26.25 ft)			
BC-M12F4-M12M4A-22-10	10 m (30.81 ft)			
BC-M12F4-M12M4A-22-15	15 m (49.2 ft)			

4-pin A-Code Double-Ended M12 Female Right-Angle to M12 Male Right-Angle Cordsets				
Model	**Length**	**Dimensions (mm)**	**Pinouts**	
BC-M12F4A-M12M4A-22-0.3	0.3 m (1 ft)		Female  Male 	1 = Brown 2 = White 3 = Blue 4 = Black
BC-M12F4A-M12M4A-22-1	1 m (3.28 ft)			
BC-M12F4A-M12M4A-22-2	2 m (6.56 ft)			
BC-M12F4A-M12M4A-22-5	5 m (16.4 ft)			
BC-M12F4A-M12M4A-22-8	8 m (26.25 ft)			
BC-M12F4A-M12M4A-22-10	10 m (30.81 ft)			
BC-M12F4A-M12M4A-22-15	15 m (49.2 ft)			

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)			

4-Pin D-Code Double-Ended M12 Male Ethernet Cordsets				
Model	**Length**	**Style**	**Dimensions**	**Pinout (Male)**
BCD-M12DM-M12DM-0.3M	0.3 m (13 in)	Straight		 1 = White/Orange 2 = White/Green 3 = Orange 4 = Green
BCD-M12DM-M12DM-1M	1 m (39 in)			

Chapter Contents

Chapter 9 Product Support and Maintenance

DXMR90-X1E 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 fixed nylon M12 female quick disconnect connectors

One integral 4-pin nickel-plated brass M12 male quick disconnect connector

Two integral 5-pin fixed nylon M12 female D-Code quick disconnect connectors

Application Note

When connecting external devices through the DXMR90-X1E, it is important not to exceed the maximum current limitations of 3.5 Amps

Certifications

Banner Engineering BV
Park Lane, Culliganlaan 2F bus 3
1831 Diegem, BELGIUM

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.

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

Communication Protocols

Modbus® RTU, Modbus/TCP, EtherNet/IP™, and PROFINET®

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.

Security Protocols

TLS, SSL, HTTPS

Environmental Ratings

For Indoor Use Only

IP65, IP67, NEMA 1, UL Type 1

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)

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)

Turck Banner LTD Blenheim House
Blenheim Court
Wickford, Essex SS11 8YT
GREAT BRITAIN

For additional product support, go to www.bannerengineering.com.

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

FCC Part 15 Class A for Unintentional Radiators

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

(Part 15.21) Any changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate this equipment.

Industry Canada ICES-003(A)

This device complies with CAN ICES-3 (A)/NMB-3(A). Operation is subject to the following two conditions: 1) This device may not cause harmful interference; and 2) This device must accept any interference received, including interference that may cause undesired operation.

Cet appareil est conforme à la norme NMB-3(A). Le fonctionnement est soumis aux deux conditions suivantes : (1) ce dispositif ne peut pas occasionner d'interférences, et (2) il doit tolérer toute interférence, y compris celles susceptibles de provoquer un fonctionnement non souhaité du dispositif.

File System and Archive Process

The DXM file system is in a serial EEPROM that stores non-volatile configuration information. 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.

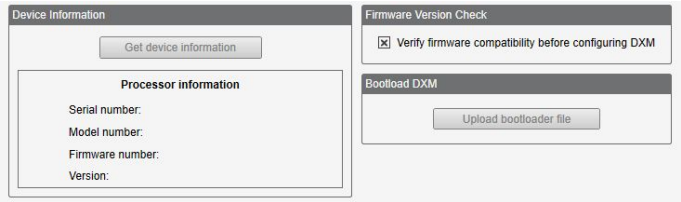
The controller XML configuration file created by the DXM Configuration Software is stored in EEPROM. The small section of non-volatile local registers is also stored in EEPROM.

Update Your DXMR90 and DXMR110 Processor Firmware Using the Configuration Software

Follow these steps to update your DXMR90 and DXMR110 processor firmware using the DXM Configuration Software.

1. Using the DXM Configuration Software version 4 or later, connect to the DXMR90-X1E via Ethernet.
   File loads to the DXMR90-X1E will take several minutes.
2. On the DXM Configuration Software, go to **Tools › Reprogram › Get Device Information** to verify the current firmware version.
   You must load a different version with the same firmware number for the bootloader to operate. Download firmware files from the Banner website.

Example Device Information screen; every device's information will be different



3. Under **Tools › Reprogram**, click **Upload bootloader file** to select the firmware file to program.
4. Select the .HEX file provided to bootload the device.
   This is a large file, so it may take 10-15 minutes to upload.
5. After the file load is completed, reboot the device by selecting **DXM › Reboot DXM**.
   Upon reboot, the device will begin to bootload. There will be a solid green light on for 6 to 7 minutes. Do not be alarmed if the device appears to not do anything. After 6 to 7 minutes, an amber LED nearest the power connector flashes for 2 to 3 minutes. After the boatload process is finished, the device returns to normal operation.
6. DO NOT disconnect the power during the 6 to 7 minutes after the device cycles the power.

To verify the firmware has been updated, go to **Tools › Reprogram › Get Device Information** and verify the new versions are listed.

DXM 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 DXM, 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.

Firmware Updates

The DXM 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 DXM. 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.

Website Information

The Banner website is the main method of disseminating DXM 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

Feature Requests

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

Potential DXM Issues

Potential issues with the DXM 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.

DXM Security

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

The DXM 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).

Warnings

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: Please download the complete DXMR90-X1E Series Industrial 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-X1E Series Industrial 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-X1E Series Industrial 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.

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.

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.

Contact Us

Banner Engineering Corp. headquarters is located at: 9714 Tenth Avenue North | Plymouth, MN 55441, USA | Phone: + 1 888 373 6767

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

LinkedIn

X (formerly Twitter)

Facebook