# Contents

1 DXM700-Bx System Overview ................................................................................................................................. 4  
1.1 DXM Hardware Configuration Overview ............................................................................................................. 5  
1.2 DXM Automation Protocols .................................................................................................................................... 6  
1.3 DXM Modbus Overview .......................................................................................................................................... 7  
1.3.1 DXM Modbus Registers ...................................................................................................................................... 8  
1.4 DXM Configuration Software .................................................................................................................................. 9  
1.5 Dimensions ............................................................................................................................................................ 10  
2 Quick Start Guide ......................................................................................................................................................... 11  
2.1 Device Setup .......................................................................................................................................................... 11  
2.1.1 Apply Power to the Controller ............................................................................................................................... 11  
2.1.2 Configuring the Controller: Read Registers on a Modbus Slave Device .......................................................... 13  
2.2 Configuration Instructions ....................................................................................................................................... 13  
2.2.1 Configuring the Controller: Reading Registers on a Modbus Slave Device .................................................... 13  
3 ISM Radio Board (Slave ID 1) .................................................................................................................................. 16  
3.1 MultiHop Radio DIP Switches .................................................................................................................................. 16  
3.1.1 Application Mode .................................................................................................................................................... 17  
3.1.2 Baud Rate and Parity .............................................................................................................................................. 17  
3.1.3 Disable Serial ......................................................................................................................................................... 17  
3.1.4 Transmit Power Levels/Frame Size ....................................................................................................................... 18  
3.2 Modbus Registers for the MultiHop Radio Board Module ...................................................................................... 18  
3.3 DIP Switch Settings for the Gateway Radio Board Module ..................................................................................... 18  
3.4 Modbus Registers for the Gateway Radio Board Module ..................................................................................... 18  
3.4.1 Alternative Modbus Register Organization ....................................................................................................... 19  
4 Processor/Base Board Connections ........................................................................................................................ 23  
4.1 DIP Switch Settings for the Base Board .................................................................................................................. 24  
4.2 Ethernet ................................................................................................................................................................. 24  
4.3 USB ....................................................................................................................................................................... 24  
4.4 Internal Local Registers (Slave ID 199) for the DXM700, DXM1000, and DXM1500 ............................................. 25  
4.5 Applying Power to the DXM700-Bx Wireless Controller ..................................................................................... 28  
4.6 Connecting the Communication Pins .................................................................................................................... 28  
4.7 Modbus RTU Master and Slave Ports ..................................................................................................................... 28  
4.8 Modbus I/O Registers for the Base Board ............................................................................................................... 29  
4.8.1 Modbus I/O Registers for the Base Board ............................................................................................................... 29  
5 Cellular Modem Boards .......................................................................................................................................... 31  
5.1 Cellular Modem Board for GSM and LTE ............................................................................................................ 31  
5.2 Cellular Power Requirements ................................................................................................................................ 31  
5.3 Using the DXM Cellular Modem ............................................................................................................................ 31  
5.3.1 Activating a Cellular Modem ................................................................................................................................. 31  
6 LCD and Menu System ............................................................................................................................................. 35  
6.1 Registers ............................................................................................................................................................... 35  
6.2 Push ...................................................................................................................................................................... 35  
6.3 ISM Radio ............................................................................................................................................................ 36  
6.4 I/O Board ............................................................................................................................................................ 36  
6.5 System Config ....................................................................................................................................................... 37  
6.5.1 ISM Radio ............................................................................................................................................................ 37  
6.5.2 Ethernet ............................................................................................................................................................. 38  
6.5.3 DXM Modbus ID .................................................................................................................................................. 38  
6.5.4 LCD Contrast ..................................................................................................................................................... 38  
6.5.5 Reset ................................................................................................................................................................. 38  
6.6 System Info .......................................................................................................................................................... 39  
6.7 Display Lock ........................................................................................................................................................ 40  
6.8 Modbus Registers for the LCD Board (Modbus Slave ID 201) ........................................................................... 40  
7 Working with Modbus Devices ................................................................................................................................ 41  
7.1 Assigning Modbus Slave IDs ................................................................................................................................ 41  
7.2 Modbus Operation .................................................................................................................................................. 42  
7.3 Wireless and Wired Devices .................................................................................................................................. 42  
7.4 Modbus Communication Timeouts ....................................................................................................................... 42  
7.4.1 MultiHop Networks vs DX80 Star Networks ..................................................................................................... 43  
7.4.2 Calculating the Communications Timeout for Battery-Powered MultiHop Radios ..................................... 43
1 DXM700-Bx System Overview

Banner’s DXM Logic Controller integrates Banner’s wireless radio, cellular connectivity, and local I/O to provide a platform for the Industrial Internet of Things (IIoT).

![Diagram of DXM700-Bx System Overview](image)

### Connectivity

**Cellular**
- Sure Cross® Radios
- Ethernet
- USB
- RS-485 Master
- RS-485 Slave

**Ethernet**
- Modbus/TCP (master/slave) or Ethernet/IP

**USB**
- Modbus/USB

**RS-485**
- Modbus RS-485 Master
- Modbus RS-485 Slave

### Logic Controller

- Action Rules
- Programming Language
- Scheduler
- Push to the Cloud
- Data Logging
- Email

### User Interface

- LCD Screen
- LED Indicators

### I/O

- Discrete Outputs

---

| Modbus Registers for Internal Local Registers (Modbus Slave ID 199) |
|-----------------|-----------------|-----------------|
| **Local Registers** | **Type** | **Description** |
| 1–845 | 32-bit integer | Local data registers |
| 846–849 | 32-bit integer | Reset, Constant, Timer |
| 851–900 | 32-bit non-volatile integer | Data flash, non-volatile |
| 901–1000 | | Reserved for internal use |
| 1001–5000 | Floating point | Floating point registers, local data registers |
| 5001–7000 | 32-bit integer | Same as 1–845 |
| 7001–8000 | 32-bit non-volatile integer | Same as 851–900 |
| > 10000 | | Read only virtual registers, system-level data |

### Outputs

- Four discrete PNP outputs (supply voltage minus 2 V, 100mA maximum at 30 V dc)

### Connectivity

- The DXM700’s wired and wireless connectivity options make it easy to share data between local and remote equipment. The cellular modem option eliminates the need for IT infrastructures to connect remote equipment for sensing and control to IIoT cloud services. The integrated Sure Cross® wireless radio enables Modbus connectivity to remote sensors, indicators, and control equipment.

**Wired Connectivity**
- Ethernet: Modbus/TCP (master/slave) or Ethernet/IP
- Field Bus: Modbus RS-485 Master/Slave

**Wireless Connectivity**
- Sure Cross Wireless Radio: DX80 900 MHz, DX80 2.4 GHz, MultiHop 900 MHz, or MultiHop 2.4 GHz
- Cellular modem: LTE (United States only) or GSM (Outside the United States)

### Logic Controller

- Program the DXM700’s logic controller using action rules and/or ScriptBasic language, 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 and LCD password protection is an option.
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 wireless devices or local wired Modbus devices with scaling
- Modbus/TCP Master Read or Write rules for external devices on the network

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

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

Optional Text 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

Data Logging
- Cyclic data/event logging
- Email log files

User Interface — A simple user interface consists of an LCD screen and four LED indicators.

User programmable LCD
- Bind Sure Cross radios
- Conduct a site survey to evaluate the radio signal integrity of radios within the network
- View register and output information
- View system status and configuration

API Interface
- Host Initiated control
- Web service integration

User Defined LED indicators
- Indicates the status of the DXM700, processes, or equipment

1.1 DXM Hardware Configuration Overview
The DXM700-Bx Wireless Controller can have multiple configurations. The DXM700 will have a model number label on the housing. Use the model number and model table above to identify which boards are included in the controller.

When opening the DXM700, follow proper ESD grounding procedures.

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.

The top housing contains the LCD display board. The display board is connected to the base board using a ribbon cable with a 20 pin connector.
The DXM700 base board provides connections for all communications connections, outputs and power/ground. The optional cellular modem is installed in the bottom base board. Attach the antenna cable from the cellular modem to the U.FL connection on the base board.

The optional ISM radio fits on the LCD display board in the top housing assembly. The ISM radio boards are available with either a 900 MHz radio or a 2.4 GHz radio. The ISM radio module installs into the 12-pin parallel sockets strips. To install,

1. Orient the mounting through hole in the ISM radio to the mounting hole next to the 12-pin sockets on the display PCB.
2. Connect the antenna cable from the ISM radio U.FL to the U.FL connector on the display PCB.

1.2 DXM Automation Protocols

The DXM700 supports the following automation protocols.
Modbus RTU

The DXM700 manages two separate physical ports running the Modbus RTU protocol. The DXM700 is the Modbus Master when operating the Modbus master RTU port. The DXM700 uses the master Modbus RTU bus to communicate with locally connected Modbus devices or uses the Banner wireless radio to communicate with remote Modbus devices.

The other Modbus RTU port is used by a host system to access the DXM700 as a slave device. The slave Modbus RTU port allows access all the internal registers concurrently with the master RTU port. Set the slave Modbus ID using the LCD menu: **SYSTEM CONFIG > DXM Modbus ID**.

By default, the Modbus RTU ports are active. Configure the port parameters using the DXM Configuration Software.

Modbus TCP/IP

A host system acting as a Modbus master can access the DXM700 using the Modbus TCP/IP protocol over Ethernet. Standard Modbus port 502 is used by the DXM700 for all Modbus TCP/IP requests.

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

By default, Modbus TCP/IP is active. Configure the DXM700 using Modbus TCP rules in the DXM Configuration Software.

EtherNet/IP™

The Ethernet port is actively running EtherNet/IP. From the factory the DXM700 is configured to read and write registers on DX80 wireless devices 1 through 16. Custom configurations can be set using the DXM Configuration Software.

By default, EtherNet/IP is active.

1.3 DXM Modbus Overview

The DXM700 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 DXM700, as a Modbus master device or slave device, exchanges data using the Local Registers. Modbus over Ethernet (Modbus/TCP) uses the Local Registers as the accessible register data.

Using Action, Read/Write, and Threshold Rules allows you to manipulate the processor’s Local Registers. The 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.

Accessing the I/O Base and the LCD follows the same communication as an external Modbus device. Each device has an ID number to uniquely identify itself. The I/O base is Modbus ID 203 and the LCD is Modbus ID 201.
1.3.1 DXM Modbus Registers

The DXM700-Bx Wireless Controller may have up to four internal Modbus slave devices:

<table>
<thead>
<tr>
<th>Modbus Slave ID</th>
<th>Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DX80 Performance Gateway or MultiHop ISM Radio—MultiHop wireless devices connected to the internal MultiHop radio should be assigned Modbus Slave addresses starting at 11.</td>
</tr>
<tr>
<td>199</td>
<td>Local Registers—Internal storage registers of the DXM700</td>
</tr>
<tr>
<td>203</td>
<td>Base Board Outputs—Outputs of the DXM700.</td>
</tr>
<tr>
<td>201</td>
<td>LCD Board—The user has access to the LED indicators on the DXM700.</td>
</tr>
</tbody>
</table>

All Modbus registers are defined as 16-bit Modbus Holding Registers. When connecting external Modbus slave devices, only use Modbus slave IDs 2 through 198. The local registers, the I/O base, and the LCD slave IDs are fixed, but the internal radio slave ID can be changed if needed.

<table>
<thead>
<tr>
<th>Modbus Registers for Internal Local Registers (Modbus Slave ID 199)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Local Registers</strong></td>
</tr>
<tr>
<td>1–845</td>
</tr>
<tr>
<td>846–849</td>
</tr>
<tr>
<td>851–900</td>
</tr>
<tr>
<td>901–1000</td>
</tr>
<tr>
<td>1001–5000</td>
</tr>
<tr>
<td>5001–7000</td>
</tr>
<tr>
<td>7001–8000</td>
</tr>
<tr>
<td>&gt; 10000</td>
</tr>
</tbody>
</table>
### Modbus Registers for the LCD Board (Modbus Slave ID 201)

<table>
<thead>
<tr>
<th>Modbus Register</th>
<th>LED</th>
<th>Color</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1102 : bit 0</td>
<td>LED 1</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>1103 : bit 0</td>
<td>LED 2</td>
<td>Amber</td>
<td></td>
</tr>
<tr>
<td>1104 : bit 0</td>
<td>LED 3</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>1105: bit 0</td>
<td>LED 4</td>
<td>Amber</td>
<td></td>
</tr>
<tr>
<td>1107: bit 0</td>
<td>LED 1</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>1108 : bit 0</td>
<td>LED 2</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>1109 : bit 0</td>
<td>LED 3</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>1110 : bit 0</td>
<td>LED 4</td>
<td>Green</td>
<td></td>
</tr>
</tbody>
</table>

1 = On
0 = Off

### Modbus Registers for the Base Board Outputs (Modbus Slave ID 203)

<table>
<thead>
<tr>
<th>Modbus Register</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2101</td>
<td>0–1</td>
<td>PNP Output 1</td>
</tr>
<tr>
<td>2102</td>
<td>0–1</td>
<td>PNP Output 2</td>
</tr>
<tr>
<td>2103</td>
<td>0–1</td>
<td>PNP Output 3</td>
</tr>
<tr>
<td>2104</td>
<td>0–1</td>
<td>PNP Output 4</td>
</tr>
</tbody>
</table>

### Modbus Registers for the ISM Radio (Modbus Slave ID 1)—See Modbus Registers for the MultiHop Radio Board Module (p. 18) and Modbus Registers for the Gateway Radio Board Module (p. 18).

### 1.4 DXM Configuration Software

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

The DXM Configuration Software configures the DXM700 by creating an XML file that is transferred to the DXM700 using a USB or Ethernet connection. The DXM700 can also receive the XML configuration file from a Web server using a cellular or Ethernet connection. This configuration file governs all aspects of the DXM700 operation.

The wireless network devices are a separate configurable system. Use the DX80 User Configuration Software to configure the internal DX80 wireless Gateway and the attached wireless Nodes. Use the MultiHop Configuration Software if the internal radio is a MultiHop device.

All tools can be connected to the DXM700 using a USB cable or an Ethernet connection.

![Figure 5. Overview of the DXM Configuration Software features](image-url)
1.5 Dimensions

All measurements are listed in millimeters [inches], unless noted otherwise.
2 Quick Start Guide

2.1 Device Setup

2.1.1 Apply Power to the Controller

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

Equipment used:
- DXM700-Bx Wireless Controller
- MQDMC-501 0.3 m (1 ft) cordset with a 5-pin M12/Euro-style quick disconnect fitting
- PS24W Wall plug power supply; 24 V dc, 1 A

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

1. Connect the brown wire from the MQDMC-501 cordset to the DXM700's PW (+ power) terminal.
2. Connect the blue wire from the MQDMC-501 cordset to the DXM700's GD (- ground) terminal.
3. Connect the PS24W power supply to the MQDMC-501 cordset.
4. Plug in the PS24W wall plug power supply.

2.1.2 Binding and Conducting a Site Survey with the ISM Radio

The DXM700 internal ISM radio will either be a MultiHop master radio or a DX80 Gateway radio. Before the ISM radio can communicate, the ISM radios within the DXM700 must be bound to the other radios in the wireless network.

Use the DXM700 LCD menu to bind radios to the internal ISM radio.

Some configurations of the DXM Controller will make it difficult to run binding or site survey. If you are having trouble accessing these features, disable the XML configuration file using DIP switch 4 on the processor board or load in a blank XML file.

**Bind a DX80 Node to a DXM Gateway and Assign the Node Address**

Before beginning the binding procedure, apply power to all the devices. Separate radios by 2 meters when running the binding procedure. Put only one DXM Gateway into binding mode at a time to prevent binding to the wrong Gateway.

Binding Nodes to a Gateway ensures the Nodes only exchange data with the Gateway they are bound to. After a Gateway enters binding mode, the Gateway automatically generates and transmits a unique extended addressing (XADR), or binding, code to all Nodes within range that are also in binding mode. The extended addressing (binding) code defines the network, and all radios within a network must use the same code.

1. Enter binding mode on the DXM radio:
   a) Use the arrow keys to select the ISM Radio menu on the LCD and press ENTER.
   b) Highlight the Binding menu and press ENTER.
2. Assign the Node address to the Node.
   - For Nodes without rotary dials: Use the DXM arrow keys to select the Node address to assign to the DX80 Node about to enter binding mode. The DXM assigns this Node address to the next Node that enters binding mode. Only bind one Node at a time.
   - For Nodes with rotary dials: Use the Node’s rotary dials to assign a valid decimal Node Address (between 01 and 47). The left rotary dial represents the tens digit (0 through 4) and the right dial represents the ones digit (0 through 9) of the Node Address.
3. Start binding mode on the DXM radio by pressing ENTER on the DXM radio.
4. Enter binding mode on the DX80 Node.
   - For housed radios, triple-click button 2.
   - For board-level radios, triple-click the button.
   - For Nodes without buttons, refer to the Node’s datasheet for instructions on entering binding mode.
The left and right red LEDs flash alternately and the Node searches for a Gateway in binding mode. After the Node binds, the LEDs stay solid momentarily, then they flash together four times. The Node automatically exits binding mode.

5. Label the Node with the assigned address number for future reference.
6. Press BACK on the DXM to exit binding mode for that specific Node address.
7. Repeat steps 2 through 5, for as many DX80 Nodes as are needed for your network.
8. When you are finished binding, press BACK on the DXM until you return to the main menu.

Bind a MultiHop Radio to a DXM and Assign the Device ID

Before beginning the binding procedure, apply power to all the devices. Separate radios by 2 meters when running binding procedure. Put only one DXM MultiHop master radio into binding mode at a time to prevent binding the slave radios to the wrong master radio.

Binding MultiHop radios ensures all MultiHop radios within a network communicate only with other radios within the same network. The MultiHop radio master automatically generates a unique binding code when the radio master enters binding mode. This code is then transmitted to all radios within range that are also in binding mode. After a repeater/slave is bound, the repeater/slave radio accepts data only from the master to which it is bound. The binding code defines the network, and all radios within a network must use the same binding code.

1. Enter binding mode on the DXM radio:
   a) Use the arrow keys select the ISM Radio menu on the LCD and press ENTER.
   b) Highlight the Binding menu and press ENTER.
2. Assign the device address to the repeater or slave radios.
   • For MultiHop radios without rotary dials: Use the DXM arrow keys to select the device ID to assign to the MultiHop radio about to enter binding mode. The DXM assigns this device ID to the next radio that enters binding mode. Only bind one slave radio at a time.
   • For MultiHop radios with rotary dials: Use the repeater or slave’s rotary dials to assign a valid decimal device ID (11 through 60). The left rotary dial represents the tens digit (1 through 6) and the right dial represents the ones digit (0 through 9) of the device ID.
3. Start binding mode on the DXM radio by pressing ENTER on the DXM radio.
4. After entering binding mode on the DXM700, put the MultiHop repeater or slave radio into binding mode.
   • For housed radios, triple-click button 2.
   • For board-level radios, triple-click the button.
   • For radios without buttons, refer to the radio’s datasheet for instructions on entering binding mode.
   After binding is completed, the MultiHop slave automatically exits binding mode and begins operation.
5. Press BACK on the DXM to exit binding mode for that specific device address.
6. Label the MultiHop slave radio with the assigned address number for future reference.
7. Repeat steps 2 through 6, changing the device address for as many MultiHop slaves as are needed for your network.
8. When you are finished binding, press BACK on the DXM until you return to the main menu.

All radio devices begin to form the network after the master data radio exits binding mode.

Conduct a Site Survey

Although the MultiHop and DX80 devices are architecturally different, the site survey process is similar when conducted from the DXM LCD menu.

Conducting a site survey, also known as a radio signal strength indication (RSSI), analyzes the radio communications link between the Gateway (or master radio) and any Node (or slave radio) within the network by analyzing the radio signal strength of received data packets and reporting the number of missed packets that required a retry.

For a DX80 network, the Gateway controls the site survey and the results display on the LCD. Running a site survey on a DX80 network does not affect the throughput of the DX80 network. The DX80 Gateway-Node system can run a site survey analysis while the network is operational.

For a MultiHop network, the master device passes the site survey request to the intended Modbus slave device. The Site Survey runs and the results display on the LCD. Running a site survey on a MultiHop network stops all network traffic to that device.

1. From the ISM Radio menu, use the down arrow to highlight the Site Survey menu. Press ENTER.
2. Use the Up or Down arrows to select the Node number (DX80 network) or Modbus Slave ID (MultiHop network). Press ENTER to run the site survey with that Node or slave.

The site survey results display as green, yellow, red, and missed packets. Green indicates the highest signal strength, then yellow, and red. Missed packets were not received.
2.1.3 Set the IP Address

Change the IP address of the DXM700-B1R1 to connect to a Modbus TCP/IP or Ethernet/IP host controller.

Equipment needed:
- DXM700-B1R1 Wireless Controller

There are two ways to set the IP address: using the DXM’s LCD menu or using the DXM Configuration Software to change the XML file.

IP addresses entered into the LCD menu system override the IP addresses in the XML configuration files. To use the IP addresses set in the XML configuration file, clear the IP addresses from the menu system.

1. On the DXM, use the arrows and move to the System Config menu. Press Enter.
2. Use the arrow keys to select the Ethernet menu. Press Enter.
3. Use the arrow keys to select IP. Press Enter.
   The octet of the IP address displays (for example, 192.168.10.1).
4. Use the up and down arrows to change the IP address. Press Enter to move to the next octet.
5. Press Enter on the final octet to accept the changes.
6. Cycle power to the DXM700.
   The changes are saved on the DXM700 and the new IP address will be used.

Use this same procedures to set the subnet mask (SN) and default gateway (GW) to match your network requirements.

2.2 Configuration Instructions

2.2.1 Configuring the Controller

Configure the DXM700 using the configuration software.

To configure the DXM700, connect the DXM700’s USB or Ethernet port to a computer.

The DXM Configuration Software allows the user to define parameters for the DXM700, then saves the configuration in an XML file on the PC.

After the configuration file is saved, upload the XML configuration file to the DXM700 for operation.

This quick start guide outlines the basic operations to set up a DXM700 using the configuration software. For a more comprehensive explanation of features, refer to the DXM Configuration Software Instruction Manual (p/n 209933).

For a complete list of all associated product documentation, go to Product Support and Maintenance (p. 57).

2.2.2 Configuration Example: Reading Registers on a Modbus Slave Device

The opening page of the DXM Configuration Software displays the Local Registers tab.

The local registers are the main global pool of registers that are defined by the user to store data within the DXM700.

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

In this short example, we will configure the DXM700 to read six registers on an external Modbus Slave device and save the data into the local registers.
Step 1: Define the Local Registers

Change the name and parameter settings for each Local Register under the Local Registers in Use screen. You may change them individually (Edit Register) or in groups (Modify Multiple Registers).

![Modify Multiple Registers - Configuration Example](image)

1. Click on the Modify Multiple Registers section of the Local Registers in Use screen. Use this screen to quickly modify multiple local registers at a time.
2. Select the range of registers to change.
3. Select the fields to change in each local register. In our example, registers one through six will be changed and the names will be GPS Reg followed by an auto-incremented number. This example will also change the LCD permissions flag to Set, then Read to allow the values of the local registers to display on the LCD.
4. Click Modify Registers to apply your changes.

Step 2: Read the Registers

Under Register Mapping, the Read Rules or Write Rules interact with the Local Registers to exchange data with external Modbus devices.

This example screen shows a read rule created to read six registers (address 1 through 6), from Modbus Slave 4. The results are stored in the Local Registers 1 through 6.

![Read Rules - Configuration Example](image)

1. Go to the Register Mapping > RTU or Modbus TCP > Read Rules tab to define a Modbus read rule.
2. Click Add Read Rule.
3. Click the arrow next to the new rule to display all parameters.
4. Type in a name into the name field.
5. Select the slave address. In this example, we will read from Slave ID 4.
6. Select the starting register and ending register. In this example, we will read from register 1 through register 6.
7. Select the beginning local register on the DXM700.
8. Enter a polling frequency. In this example we have entered five seconds.
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 DXM700 local registers. Notice the list of local register names this read rule is using.
Step 3: Define the Time Zone and Set the Time Clock

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.

If you connect the DXM700 to a computer, click Sync PC Time with Device to set the time on the DXM700 to match the time of the computer.

Step 4: Save the Configuration File

To save your configuration file:
1. Go to File > Save As.
2. Enter a file name and save the file. The file name cannot contain spaces or special characters.

Step 5: Connect the DXM700

1. Connect the DXM700 to the computer using the USB port.
2. From the Device menu, select Connection Settings.
3. From the dialog box, select the appropriate com port for the DXM communications.
4. Click Connect to connect to the DXM700.

Step 6: Send the Configuration File to the DXM700

1. From the Device menu, select Send XML Configuration to Device.
2. Select the configuration file to load. The software will have pre-selected the file name you have previously saved.

**Important:** The software only loads a file to the DXM700. Internal parameter settings that are changed in the tool but not saved to the file will not be sent to the device.

After the file is selected, the software uploads the file to the DXM700. The DXM Configuration Software reboots the controller after the configuration file finishes uploading. The new configuration is only read at startup and always requires a reboot or power cycle to take effect. It will take a few seconds for the DXM700 to reboot.

**Important:** If the power cycles to the DXM700 while the DXM Configuration Software is connected, close the USB port from the software and unplug the USB cable. Reconnect the DXM700 by plugging the USB cable into the DXM700, then select Device > Connection Settings.

The DXM700 is now running the new configuration. On the DXM700’s LCD screen, select the Registers menu by clicking the Enter button with the Registers menu highlighted. The local registers defined in the configuration tool display.
3 ISM Radio Board (Slave ID 1)

The ISM embedded radio boards are available in either DX80 MultiHop or DX80 Performance.

![Diagram of ISM radio board]

Plug the ISM radio into the I/O base board with the U.FL antenna connector closest to the SMA connectors.

- **A** - Antenna connector
- **B** - Button
- **C** - LED
- **D1** - DIP switches
- **D2** - DIP Switches

**Figure 11. ISM radio board**

**Button Operation**

For DXM models without a LCD display, use the button (B) to bind the ISM radio. For models with a LCD display, use the ISM menu to bind the radio.

**LED Operation**

The LED located on the ISM radio module indicates power and communications traffic.

- Solid green DX80 ISM radio LED: Indicates power.
- Flashing green MultiHop ISM radio LED indicates operation.
- Red and green combined: Communications traffic and binding.

ISM board LED operations also display on the LED on the right side of the I/O base board.

### 3.1 MultiHop Radio DIP Switches

The DX80 MultiHop architecture creates a tree network with a Master radio and one or more Repeater/Slave devices. The MultiHop architecture is suited for networks requiring repeater devices to provide extended range or obstacle avoidance. MultiHop ISM radio devices are defined with R2, R4, and R5 in the model number.

- DXMxxxx-xxR2 - MultiHop 900 MHz
- DXMxxxx-xxR4 - MultiHop 2.4 GHz
- DXMxxxx-xxR5 - MultiHop 900 MHz, 100 mW
- DXMxxxx-xxR9 - MultiHop 900 MHz, (Australia)

Making changes to the baud or parity settings requires that you make the same settings to the Modbus Master Communications section within the DXM Configuration Software (Settings > General).

**Important:** Disabling the serial port disables the ISM radio in the DXM700. Selecting Transparent mode causes radio communications to be slower and denies access to device I/O register data.

<table>
<thead>
<tr>
<th>Device Settings</th>
<th>D1 Switches</th>
<th>D2 Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial line baud rate 19200 OR User defined receiver slots</td>
<td>OFF*</td>
<td>OFF*</td>
</tr>
</tbody>
</table>
### Device Settings

| Serial line baud rate 38400 OR 32 receiver slots | OFF | ON |
| Serial line baud rate 9600 OR 128 receiver slots | ON | OFF |
| Serial line baud rate Custom OR 4 receiver slots | ON | ON |
| Parity: None | OFF* | OFF* |
| Parity: Even | OFF | ON |
| Parity: Odd | ON | OFF |
| Disable serial (low power mode) and enable the receiver slots select for switches 1-2 | ON | ON |
| Transmit power | OFF |
| 900 MHz radios: 1.00 Watt (30 dBm) | ON * |
| 2.4 GHz radios: 0.065 Watts (18 dBm) and 60 ms frame | ON * |
| Transmit power | OFF |
| 900 MHz radios: 0.25 Watts (24 dBm) | ON * |
| 2.4 GHz radios: 0.065 Watts (18 dBm) and 40 ms frame | ON * |
| Application mode: Modbus | OFF* |
| Application mode: Transparent | ON |
| MultiHop radio setting: Repeater | OFF | OFF |
| MultiHop radio setting: Master | OFF | ON |
| MultiHop radio setting: Slave | ON | OFF |
| MultiHop radio setting: DXM LCD Menu Control | ON * | ON * |

* Default configuration

The default settings for D2 DIP switches 1, 3, and 4 are ON. This allows for forcing the device into Master mode and DXM menu control for the radio power settings.

#### 3.1.1 Application Mode

The MultiHop radio operates in either Modbus mode or transparent mode. Use the internal DIP switches to select the mode of operation. All MultiHop radios within a wireless network must be in the same mode.

**Modbus** mode uses the Modbus protocol for routing packets. In Modbus mode, a routing table is stored in each parent device to optimize the radio traffic. This allows for point to point communication in a multiple data radio network and acknowledgement/retry of radio packets. To access a radio’s I/O, the radios must be running in Modbus mode.

In **transparent** application mode, all incoming packets are stored, then broadcast to all connected data radios. The data communication is packet based and not specific to any protocol. The application layer is responsible for data integrity. For one to one data radios it is possible to enable broadcast acknowledgement of the data packets to provide better throughput. In transparent mode, there is no access to the radio’s I/O.

#### 3.1.2 Baud Rate and Parity

The baud rate (bits per second) is the data transmission rate between the device and whatever it is physically wired to. Set the parity to match the parity of the device you are wired to.

#### 3.1.3 Disable Serial

If the local serial connection is not needed, disable it to reduce the power consumption of a data radio powered from the solar assembly or from batteries. All radio communications remain operational.
3.1.4 Transmit Power Levels/Frame Size

The 900 MHz data radios can be operated at 1 watt (30 dBm) or 0.250 watt (24 dBm). For most models, the default transmit power is 1 watt.

For 2.4 GHz radios, the transmit power is fixed at 0.065 watt (18 dBm) and DIP switch 5 is used to set the frame timing. The default position (OFF) sets the frame timing to 60 milliseconds. To increase throughput, set the frame timing to 40 milliseconds.

Prior to date code 15341 and radio firmware version 3.6, the frame timing was 40 ms (OFF) or 20 ms (ON).

3.2 Modbus Registers for the MultiHop Radio Board Module

The DX80 MultiHop master radio is a tree-based architecture device that allows for repeater radios to extend the wireless network. Each device in a MultiHop network is a Modbus device with a unique Modbus ID.

Modbus registers in a MultiHop network are contained within each individual radio device. To obtain Modbus register data from a MultiHop device, configure the DXM700 to access each device across the wireless network as an individual Modbus slave device.

Example: MultiHop Modbus Register Table

Example MultiHop Modbus registers with generic devices.

<table>
<thead>
<tr>
<th>MultiHop Device</th>
<th>Slave ID</th>
<th>Modbus Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXM Master radio</td>
<td>1</td>
<td>none</td>
</tr>
<tr>
<td>Slave radio</td>
<td>11</td>
<td>Modbus register 1–16 are inputs, 501–516 are outputs</td>
</tr>
<tr>
<td>Repeater radio</td>
<td>12</td>
<td>Modbus register 1–16 are inputs, 501–516 are outputs</td>
</tr>
<tr>
<td>Slave radio</td>
<td>15</td>
<td>Modbus register 1–16 are inputs, 501–516 are outputs</td>
</tr>
</tbody>
</table>

3.3 DIP Switch Settings for the Gateway Radio Board Module

The 900 MHz radios transmit at 1 Watt (30 dBm) or 250 mW (24 dBm). While the Performance radios operate in 1 Watt mode, they cannot communicate with the older 150 mW radios. To communicate with 150 mW radios, operate this radio in 250 mW mode. For 2.4 GHz models, this DIP switch is disabled. The transmit power for 2.4 GHz is fixed at about 65 mW EIRP (18 dBm), making the 2.4 GHz Performance models automatically compatible with older 2.4 GHz models.

<table>
<thead>
<tr>
<th>DIP Switch 1</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OFF *</td>
<td>1 Watt (30 dBm, 900 MHz models only) (default configuration)</td>
</tr>
<tr>
<td>ON</td>
<td>250 mW (24 dBm, 900 MHz models only), DX80 compatibility mode</td>
</tr>
</tbody>
</table>

3.4 Modbus Registers for the Gateway Radio Board Module

The DX80 Performance Gateway is a star-based architecture device that contains all the Modbus registers for the wireless network within the Gateway. To access any input or output values within the entire wireless network, read the appropriate Modbus register from Gateway.

There are 16 Modbus registers allocated for each device in the wireless network. The first 16 registers (1–16) are allocated for the Gateway, the next 16 (17–32) are allocated for Node 1, the next 16 (33–48) are allocated for Node 2 and so forth. There are no inputs or outputs on the DXM embedded Gateway but the Modbus registers are still allocated for them.

Although only seven Nodes are listed in the table, the Modbus register numbering continues for as many Nodes as are in the network. For example, the register number for Node 10, I/O point 15 , is 175. Calculate the Modbus register number for each device using the equation:

\[ \text{Register Number} = \text{I/O#} + (\text{Node#} \times 16) \]
### Table 1: Modbus Holding Registers

<table>
<thead>
<tr>
<th>I/O Point</th>
<th>Gateway</th>
<th>Node 1</th>
<th>Node 2</th>
<th>Node 3</th>
<th>Node 4</th>
<th>Node 5</th>
<th>Node 6</th>
<th>Node 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
<td>33</td>
<td>49</td>
<td>65</td>
<td>81</td>
<td>97</td>
<td>113</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>18</td>
<td>34</td>
<td>50</td>
<td>66</td>
<td>82</td>
<td>98</td>
<td>114</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>19</td>
<td>35</td>
<td>51</td>
<td>67</td>
<td>83</td>
<td>99</td>
<td>115</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>20</td>
<td>36</td>
<td>52</td>
<td>68</td>
<td>84</td>
<td>100</td>
<td>116</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>21</td>
<td>37</td>
<td>53</td>
<td>69</td>
<td>85</td>
<td>101</td>
<td>117</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>22</td>
<td>38</td>
<td>54</td>
<td>70</td>
<td>86</td>
<td>102</td>
<td>118</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>23</td>
<td>39</td>
<td>55</td>
<td>71</td>
<td>87</td>
<td>103</td>
<td>119</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>24</td>
<td>40</td>
<td>56</td>
<td>72</td>
<td>88</td>
<td>104</td>
<td>120</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>25</td>
<td>41</td>
<td>57</td>
<td>73</td>
<td>89</td>
<td>105</td>
<td>121</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>26</td>
<td>42</td>
<td>58</td>
<td>74</td>
<td>90</td>
<td>106</td>
<td>122</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>27</td>
<td>43</td>
<td>59</td>
<td>75</td>
<td>91</td>
<td>107</td>
<td>123</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>28</td>
<td>44</td>
<td>60</td>
<td>76</td>
<td>92</td>
<td>108</td>
<td>124</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>29</td>
<td>45</td>
<td>61</td>
<td>77</td>
<td>93</td>
<td>109</td>
<td>125</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>30</td>
<td>46</td>
<td>62</td>
<td>78</td>
<td>94</td>
<td>110</td>
<td>126</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>31</td>
<td>47</td>
<td>63</td>
<td>79</td>
<td>95</td>
<td>111</td>
<td>127</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>32</td>
<td>48</td>
<td>64</td>
<td>80</td>
<td>96</td>
<td>112</td>
<td>128</td>
</tr>
</tbody>
</table>

Example: Gateway Modbus Register Table
Access all wireless network registers by reading Modbus slave ID 1.

<table>
<thead>
<tr>
<th>DX80 Device</th>
<th>Slave ID</th>
<th>Modbus Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>DXM Gateway radio</td>
<td>1</td>
<td>Modbus registers 1–8 are inputs, 9–16 are outputs</td>
</tr>
<tr>
<td>Node 1</td>
<td>-</td>
<td>Modbus registers 17–24 are inputs, 25–32 are outputs</td>
</tr>
<tr>
<td>Node 2</td>
<td>-</td>
<td>Modbus registers 33–40 are inputs, 41–48 are outputs</td>
</tr>
<tr>
<td>Node 3</td>
<td>-</td>
<td>Modbus registers 49–56 are inputs, 57–64 are outputs</td>
</tr>
</tbody>
</table>

### 3.4.1 Alternative Modbus Register Organization

The Sure Cross DX80 Alternative Modbus Register Organization registers are used for reordering data registers to allow host systems to efficiently access all inputs or outputs using a single Modbus command. The register groups include the input/output registers, bit-packed registers, and analog registers. This feature is only available with the Performance models using version 3 or newer of the LCD firmware code.

<table>
<thead>
<tr>
<th>Name</th>
<th>Modbus Register Address (Dec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs and Outputs, in order by device</td>
<td>2201 through 4784</td>
</tr>
<tr>
<td>Discrete Bit Packed (Status, Discrete Inputs, Discrete Outputs)</td>
<td>6601 through 6753</td>
</tr>
<tr>
<td>Analog Inputs (1-8) and Analog Outputs (1-8)</td>
<td>6801 through 9098</td>
</tr>
</tbody>
</table>

### Input Registers and Outputs Registers

Modbus registers 2201 through 2584 are used to organize all inputs together. In this format, users can sequentially read all input registers using one Modbus message. Modbus registers 4401 through 4784 organize all outputs together to allow users to sequentially write to all outputs registers using one Modbus message.
Discrete bit-packed registers include the discrete status registers, discrete inputs, and discrete outputs. When networks use similar Nodes to gather data using the same I/O registers for each Node, discrete data from multiple Nodes can be bit packed into a single register on the Gateway. The bit-packed data is arranged by I/O point starting at the most registers.

Refer to your device's datasheet for a list of the active inputs and outputs. Not all inputs or outputs listed in this table may be active for your system.

Discrete Bit-Packed Registers

Discrete bit-packed registers include the discrete status registers, discrete inputs, and discrete outputs.

Bit packing involves using a single register, or range of contiguous registers, to represent I/O values. When networks use similar Nodes to gather data using the same I/O registers for each Node, discrete data from multiple Nodes can be bit packed into a single register on the Gateway. The bit-packed data is arranged by I/O point starting at Modbus register 6601. For example, Discrete IN 1 for all the Nodes in the network is stored in three contiguous 16-bit registers.

The most efficient way to read (or write) discrete data from a SureCross® DX80 Gateway is by using these bit-packed registers because users can read or write registers for all devices using one Modbus message. The following registers contain discrete bit-packed I/O values for the Gateway and all Nodes. Values are stored first for the Gateway, then for each Node in order of Node address.

### Bit-Packed Device Status Registers

<table>
<thead>
<tr>
<th>Register Address (Dec)</th>
<th>16-bit Register Value</th>
<th>Modbus Register Address (Dec)</th>
<th>16-bit Register Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6601</td>
<td>Node 13</td>
<td>4401–4408</td>
<td>Gateway Outputs 1 through 8</td>
</tr>
<tr>
<td>6602</td>
<td>Node 30</td>
<td>4409–4416</td>
<td>Node 1 Outputs 1 through 8</td>
</tr>
<tr>
<td>6603</td>
<td>Node 47</td>
<td>4417–4424</td>
<td>Node 2 Outputs 1 through 8</td>
</tr>
</tbody>
</table>

### Bit-Packed Discrete Input 1

<table>
<thead>
<tr>
<th>Register Address (Dec)</th>
<th>16-bit Register Value</th>
<th>Modbus Register Address (Dec)</th>
<th>16-bit Register Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6611</td>
<td>Node 25</td>
<td>6601–6603</td>
<td>Status for all devices</td>
</tr>
<tr>
<td>6612</td>
<td>Node 25</td>
<td>6611–6613</td>
<td>Input 1 from all devices</td>
</tr>
<tr>
<td>6613</td>
<td>Node 25</td>
<td>6621–6623</td>
<td>Input 2 from all devices</td>
</tr>
</tbody>
</table>

### Bit-Packed Discrete Output 1

<table>
<thead>
<tr>
<th>Register Address (Dec)</th>
<th>16-bit Register Value</th>
<th>Modbus Register Address (Dec)</th>
<th>16-bit Register Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>6661</td>
<td>Node 19</td>
<td>6661–6663</td>
<td>Output 1 from all devices</td>
</tr>
<tr>
<td>6662</td>
<td>Node 19</td>
<td>6661–6663</td>
<td>Output 2 from all devices</td>
</tr>
<tr>
<td>6663</td>
<td>Node 19</td>
<td>6661–6663</td>
<td>Output 3 from all devices</td>
</tr>
</tbody>
</table>

### Inputs

<table>
<thead>
<tr>
<th>Modbus Register Address (Decimal)</th>
<th>Description (Inputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6601-6603</td>
<td>Status for all devices</td>
</tr>
<tr>
<td>6611-6613</td>
<td>Input 1 from all devices</td>
</tr>
<tr>
<td>6621-6623</td>
<td>Input 2 from all devices</td>
</tr>
<tr>
<td>6631-6633</td>
<td>Input 3 from all devices</td>
</tr>
<tr>
<td>6641-6643</td>
<td>Input 4 from all devices</td>
</tr>
<tr>
<td>6651-6653</td>
<td>Input 5 from all devices</td>
</tr>
<tr>
<td>6661-6663</td>
<td>Input 6 from all devices</td>
</tr>
</tbody>
</table>

### Outputs

<table>
<thead>
<tr>
<th>Modbus Register Address (Decimal)</th>
<th>Description (Outputs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6691–6693</td>
<td>Output 1 from all devices</td>
</tr>
<tr>
<td>6701–6703</td>
<td>Output 2 from all devices</td>
</tr>
<tr>
<td>6711–6713</td>
<td>Output 3 from all devices</td>
</tr>
<tr>
<td>6721–6723</td>
<td>Output 4 from all devices</td>
</tr>
<tr>
<td>6731–6733</td>
<td>Output 5 from all devices</td>
</tr>
<tr>
<td>6741–6743</td>
<td>Output 6 from all devices</td>
</tr>
<tr>
<td>Inputs</td>
<td>Outputs</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Modbus Register Address (Decimal)</td>
<td>Description (Inputs)</td>
</tr>
<tr>
<td>6671-6673</td>
<td>Input 7 from all devices</td>
</tr>
<tr>
<td>6681-6683</td>
<td>Input 8 from all devices</td>
</tr>
</tbody>
</table>

**Status registers** (6601-6603) contain a bit-packed representation defining the devices that are operational in the wireless system.

A one (1) written to the Discrete Status Register area indicates the device is active within the wireless system. A zero (0) indicates the device is not active within the wireless network.

**Input registers** from all devices use Modbus registers 6611 through 6683 to organize the least significant bit into a sequential array of registers. The first register contains the least significant bit from the input values for the Gateway through Node 15. The second register contains the input values for Node 16 through Node 31, and the third register contains the input values for Nodes 32 through 47.

For discrete inputs, only the least significant bit is used. For analog inputs, the least significant bit indicates if the analog value is above or below the selected threshold value (when using the threshold parameter). For example, a least significant bit of one (1) indicates the analog value is above the selected threshold value. A least significant bit of zero (0) indicates the analog value is below the threshold value.

**Output registers** from all devices use Modbus registers 6691 through 6753 to organize the least significant bit into a sequential array of registers. Output 8 (I/O point 16) cannot be written using the discrete format.

**Analog 16-Bit Registers (Registers 6801 through 9098)**

The most efficient way to read (or write) analog data from a Gateway is by using these 16-bit analog registers. Most networks consist of similar Nodes reporting data using the same I/O registers for each Node. For this reason, the analog data is arranged by I/O point using Modbus registers 6801 through 9098. For example, Input 1 for Gateway and all Nodes is stored in the first 48 contiguous blocks of 16-bit analog registers, beginning with register 6801.

In this format, users can read a 16-bit holding register for all devices or write to a register for all devices using one Modbus message. Using these registers is the most efficient way to read all status registers, read all analog inputs, or write all analog outputs.

The following registers contain analog I/O values for the Gateway and all Nodes. Values are stored first for the Gateway, then for each Node in order of Node address.

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modbus Register Address (Decimal)</td>
<td>Description (Inputs)</td>
</tr>
<tr>
<td>6801</td>
<td>Input 1 for Gateway</td>
</tr>
<tr>
<td>6802</td>
<td>Input 1 for Node 1</td>
</tr>
<tr>
<td>6803</td>
<td>Input 1 for Node 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>6951</td>
<td>Input 2 for Gateway</td>
</tr>
<tr>
<td>6952</td>
<td>Input 2 for Node 1</td>
</tr>
<tr>
<td>6953</td>
<td>Input 2 for Node 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>7101</td>
<td>Input 3 for Gateway</td>
</tr>
<tr>
<td>7102</td>
<td>Input 3 for Node 1</td>
</tr>
<tr>
<td>7103</td>
<td>Input 3 for Node 2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>7851</td>
<td>Input 8 (Status Register) for Gateway</td>
</tr>
<tr>
<td>7852</td>
<td>Input 8 (Status Register) for Node 1</td>
</tr>
<tr>
<td>7853</td>
<td>Input 8 (Status Register) for Node 2</td>
</tr>
<tr>
<td>Modbus Register Address (Decimal)</td>
<td>Description (Inputs)</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

For example, 6801 contains the input 1 value for the Gateway, 6802 contains the input 1 value for Node 1, and 6848 contains the input 1 value for Node 47.
4 Processor/Base Board Connections

**Figure 12. DXM700 base board**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th></th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PW. Power in at 12 to 30 V dc</td>
<td>7</td>
<td>O3. Sourcing Output 3</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>GD. Ground</td>
<td>8</td>
<td>O4. Sourcing Output 4</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>M-. Master RS-485 -</td>
<td>9</td>
<td>PW. Power in at 12 to 30 V dc</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>M+. Master RS-485 +</td>
<td>10</td>
<td>GD. Ground</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>O1. Sourcing Output 1</td>
<td>11</td>
<td>S-. Slave RS-485 -</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>O2. Sourcing Output 2</td>
<td>12</td>
<td>S+. Slave RS-485 +</td>
<td></td>
</tr>
</tbody>
</table>

**Button Operation**
Pressing the button for 5 seconds forces a Push to the webserver. This assumes a proper configuration for the webserver.

**LED Operation**
The PCB LED flashes to indicate the processor board is running.
The LED starts flashing about 10 seconds after power is applied and a network connection is present. Without an ethernet network connection the LED starts to flash after about 40 seconds.

**Cellular Modem Connection**
Install the cellular modem onto the board with the cellular modem's U.FL connector on the right. The antenna cable will go between the cellular U.FL connector and the board U.FL connector. Only install/remove a cellular modem when the power to the device is disconnected.

**Force Cloud Push Button**
Press and hold this button for five seconds to send an immediate push message from the device (if properly configured).
4.1 DIP Switch Settings for the Base Board

After making changes to the DIP switch settings, cycle power to the device.

<table>
<thead>
<tr>
<th>Settings</th>
<th>DIP Switches</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Disable Ethernet Port</td>
<td>OFF *</td>
</tr>
<tr>
<td></td>
<td>ON</td>
</tr>
<tr>
<td>Disable LCD Display</td>
<td>OFF *</td>
</tr>
<tr>
<td></td>
<td>ON</td>
</tr>
<tr>
<td>Not used</td>
<td>OFF *</td>
</tr>
<tr>
<td>Bypass XML</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Bypass XML**

Turn on to have the XML file ignored at boot time. This is useful for ignoring a corrupt or questionable XML configuration file. After the device is running, a new XML file can be loaded using the DXM configuration tool.

Turn on to stop the processor from executing defined configuration. This is useful if the loaded configuration is using all the processing time and not allowing DXM Configuration Tool operations.

The factory default position is OFF.

**Disable Ethernet Port**

Set to on to power down the Ethernet interface. Disabling the unused Ethernet port reduces power consumption.

The factory default position is OFF.

**Disable LCD Display**

Set to on to disable the LCD. This DIP switch should be on when the LCD display board is not connected.

The factory default position is OFF.

4.2 Ethernet

Before applying power to the DXM700, verify the Ethernet cable is connected.

The number of times the processor attempts to connect to the Ethernet network is configured in the DXM Configuration Software (Settings > Network Ethernet Connection Acquisition). The default setting is two retries one minute after the device boots up another retry two minutes later.

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

Ethernet parameter changes entered through the LCD menu override the XML configuration parameters. To return to using the network settings in the XML configuration file, remove the Ethernet parameters defined by the LCD menu using the System Config > Ethernet > Reset menu.

4.3 USB

The USB port is used with the DXM Configuration Software to program the DXM700-Bx Wireless Controller. The USB port is also used as the console output for the processor and ScriptBasic.

Turn on debug messages to the serial console by selecting Print push debug messages to serial console in the DXM Configuration Software Settings > Cloud Services screen.
The main storage elements for the DXM700 are its Local Registers, which can store 4-byte values that result from register mapping, action rules, or ScriptBasic commands.

Local Registers updated from Modbus transactions are restricted to a 16-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.

- Local Registers 1–850 and 5001–7000 are 32-bit integer registers
- Local Registers 851–900 and 7001–8000 are non-volatile 32-bit integer registers
- Local Registers 901–1000 are reserved for internal use
- Local Registers 1001–5000 are floating point format numbers, each address stores half of a floating point number; for example, registers 1001 and 1002 store the first full 32-bit floating point number
- Local Registers 10000 and higher are read only virtual registers; virtual registers collect various system-level data

| Modbus Registers for Internal Local Registers (Modbus Slave ID 199) |
|---------------|-----------------|-----------------|
| Local Registers | Type            | Description            |
| 1–845          | 32-bit integer  | Local data registers  |
| 846–849        | 32-bit integer  | Reset, Constant, Timer|
| 851–900        | 32-bit non-volatile integer | Data flash, non-volatile |
| 901–1000       | Reserved for internal use |
| 1001–5000      | Floating point  | Floating point registers, local data registers |
| 5001–7000      | 32-bit integer  | Same as 1–845         |
| 7001–8000      | 32-bit non-volatile integer | Same as 851–900 |
| > 10000        | Read only virtual registers, system-level data |

Local Registers 1–850 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 DXM700, as a Modbus master device or a Modbus slave device, exchanges data using the Local Registers. Modbus over Ethernet (Modbus/TCP) uses the Local Registers as the accessible register data.

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

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

Virtual Registers—The DXM700 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 DXM700. Do not use Read Rules to move Virtual Local Registers data into Local Registers. Use the Action Rule > Register Copy function to move Virtual Local Registers into Local Registers space (1-850).
<table>
<thead>
<tr>
<th>Registers</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>10003</td>
<td>GPS longitude direction (N, S, E, W)</td>
</tr>
<tr>
<td>10004</td>
<td>GPS longitude</td>
</tr>
<tr>
<td>10011–10012</td>
<td>Resync timer</td>
</tr>
<tr>
<td>10013–10014</td>
<td>Resync timer rollover</td>
</tr>
<tr>
<td>10015–10016</td>
<td>Reboot cause (Restart Codes above)</td>
</tr>
<tr>
<td>10017–10018</td>
<td>Watchdog reset count</td>
</tr>
<tr>
<td>10021</td>
<td>IO Board Battery Voltage (mV)</td>
</tr>
<tr>
<td>10022</td>
<td>IO Board - Incoming Supply Voltage (mV)</td>
</tr>
<tr>
<td>10023</td>
<td>IO Board Voltage Cut-off Feature</td>
</tr>
<tr>
<td>10024</td>
<td>IO Board - Battery Charging Current (mA)</td>
</tr>
<tr>
<td>10025–10026</td>
<td>Http Push SSL Acquires</td>
</tr>
<tr>
<td>10027–10028</td>
<td>Http Push SSL Releases</td>
</tr>
<tr>
<td>10029–10030</td>
<td>Http Push SSL Forced Releases</td>
</tr>
<tr>
<td>10031–10032</td>
<td>Http Push Attempts</td>
</tr>
<tr>
<td>10033–10034</td>
<td>Http Push Successes</td>
</tr>
<tr>
<td>10035–10036</td>
<td>Http Push Failures</td>
</tr>
<tr>
<td>10037–10038</td>
<td>Http Push Last Status</td>
</tr>
<tr>
<td>10039–10040</td>
<td>Cellular Strength, BER</td>
</tr>
<tr>
<td>10055–10056</td>
<td>Alarms, smtp, attempts</td>
</tr>
<tr>
<td>10057–10058</td>
<td>Alarms, smtp, fails</td>
</tr>
<tr>
<td>10100</td>
<td>Number of read maps in default</td>
</tr>
<tr>
<td>10101</td>
<td>Number of read map successes</td>
</tr>
<tr>
<td>10102</td>
<td>Number of read map timeouts</td>
</tr>
<tr>
<td>10103</td>
<td>Number of read map errors</td>
</tr>
<tr>
<td>10104</td>
<td>Read map success streak</td>
</tr>
<tr>
<td>10105</td>
<td>Number of write map successes</td>
</tr>
<tr>
<td>10106</td>
<td>Number of write map timeouts</td>
</tr>
<tr>
<td>10107</td>
<td>Number of write map errors</td>
</tr>
<tr>
<td>10108</td>
<td>Write map success streak</td>
</tr>
<tr>
<td>10109</td>
<td>Number of passthrough successes</td>
</tr>
<tr>
<td>10110</td>
<td>Number of passthrough timeouts</td>
</tr>
<tr>
<td>10111</td>
<td>Number of passthrough errors</td>
</tr>
</tbody>
</table>
### Modbus Registers for Virtual Registers

<table>
<thead>
<tr>
<th>Registers</th>
<th>Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>10112</td>
<td>Passthrough success streak</td>
<td>DX80 Gateway automatic messaging buffer statistics</td>
</tr>
<tr>
<td>10113</td>
<td>Number of 43 buffer successes</td>
<td></td>
</tr>
<tr>
<td>10114</td>
<td>Number of 43 buffer timeouts</td>
<td></td>
</tr>
<tr>
<td>10115</td>
<td>Number of 43 buffer errors</td>
<td></td>
</tr>
<tr>
<td>10116</td>
<td>43 buffer success streak</td>
<td></td>
</tr>
<tr>
<td>11000</td>
<td>Read map success count</td>
<td></td>
</tr>
<tr>
<td>12000</td>
<td>Write map success count</td>
<td></td>
</tr>
<tr>
<td>13000</td>
<td>Read map timeout count</td>
<td></td>
</tr>
<tr>
<td>14000</td>
<td>Write map timeout count</td>
<td></td>
</tr>
<tr>
<td>15000</td>
<td>Read map error count</td>
<td>Read/Write maps statistics</td>
</tr>
<tr>
<td>16000</td>
<td>Write map error count</td>
<td></td>
</tr>
<tr>
<td>17000</td>
<td>Read map success streak</td>
<td></td>
</tr>
<tr>
<td>18000</td>
<td>Write map success streak</td>
<td></td>
</tr>
<tr>
<td>19000</td>
<td>Read map is in default</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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

<table>
<thead>
<tr>
<th>Reset Code</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Undefined</td>
</tr>
<tr>
<td>1</td>
<td>Unknown</td>
</tr>
<tr>
<td>2</td>
<td>General</td>
</tr>
<tr>
<td>3</td>
<td>Brownout</td>
</tr>
<tr>
<td>4</td>
<td>Watchdog</td>
</tr>
<tr>
<td>5</td>
<td>User</td>
</tr>
<tr>
<td>6</td>
<td>Software</td>
</tr>
<tr>
<td>7</td>
<td>Return from backup mode</td>
</tr>
</tbody>
</table>

### 4.5 Applying Power to the DXM700-Bx Wireless Controller

Apply power to the DXM700-Bx Wireless Controller using 12 to 30 V DC.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 9</td>
<td>Power in at 12 to 30 V dc</td>
</tr>
<tr>
<td>2, 10, 13, 14</td>
<td>Ground</td>
</tr>
</tbody>
</table>

### 4.6 Connecting the Communication Pins

The base board communications connections to the DXM700 Wireless Controller are RS-485 (primary) and RS-485 (secondary).

**RS-485** — The DXM700 is defined as the Modbus Master on this bus. Other internal Modbus slaves include the local processor registers (Modbus Slave ID 199) and the display board (Modbus Slave ID 201). When assigning Modbus Slave IDs to externally connected devices, only use IDs 2 through 190.

<table>
<thead>
<tr>
<th>Pin</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Primary RS-485 –</td>
<td>Running Modbus protocol at 19.2k baud, use this bus to connect to other Modbus Slave devices.</td>
</tr>
<tr>
<td>7</td>
<td>Primary RS-485 +</td>
<td>The DXM700 is a Modbus Master device on this RS-485 port.</td>
</tr>
<tr>
<td>13</td>
<td>Secondary RS-485 –</td>
<td>The DXM700 is a Modbus slave on this bus (see Processor/Base Board Connections (p. 23).)</td>
</tr>
<tr>
<td>14</td>
<td>Secondary RS-485 +</td>
<td></td>
</tr>
</tbody>
</table>

### 4.7 Modbus RTU Master and Slave Ports

The DXM700 can be a Modbus RTU master device to other slave devices and can be a Modbus slave device to another Modbus RTU master. The DXM700 uses the primary RS-485 port (M+/M-) as a Modbus RTU master to control external slave devices. All wired devices connected to the master RS-485 port must be slave devices. The secondary port (S+/S-) is the Modbus RTU slave connection.

- As a Modbus RTU master device, the DXM700 controls external slaves connected to the primary RS-485 port, the local ISM radio, local I/O base board, and the local display board.
- As a Modbus RTU slave device, the DXM700 local registers can be read from or written to by another Modbus RTU master device.

The secondary (slave) Modbus RS-485 port (S+/S-) is controlled by another Modbus master device, not the DXM700. The slave port is used by an external Modbus master device that will access the DXM700 as a Modbus slave device.

Use the DXM Configuration Software to define operational settings for both the Modbus RTU master port and the Modbus RTU slave port.

Set the Modbus Slave ID for the secondary RS-485 port using the LCD display menu: **System > DXM Slave ID**.
4.7.1 Modbus Master and Slave Port Settings

The basic communications parameters for the RS-485 ports are set in the DXM Configuration Software and are saved in the XML configuration file. All basic settings are available under Settings > System screen.

Master port parameters include:
- **Baud rate** and **parity**
- Set the **Communications Timeout** parameter to cover the expected time for messages to be sent throughout the wireless network. For the DXM700, the **Communications Timeout** parameter is the maximum amount of time the DXM700 should wait after a request is sent until the response message is received from the Modbus slave device.
- **Maximum Polling Rate** sets the minimum wait time from the end of a Modbus transaction to the beginning of the next Modbus transaction.

![Screenshot of DXM700-Bx Wireless Controller settings](image)

The Modbus Slave port settings include:
- **Baud rate** and **parity** (also set on this screen)
- Set the Modbus Slave port ID using the DXM700 LCD
- Set the **Wireless Modbus Backbone** parameter when there is an ISM radio plugged into the processor board. When this is done, the Modbus slave port uses the MultiHop radio as the slave port instead of the terminal block connection on the I/O base board.

4.7.2 DXM Modbus Slave Port ID

Set the DXM Modbus slave port ID using the LCD menu system.

1. On the LCD, use the down arrow to highlight System Config. Click the Enter button.
2. Highlight **DXM Modbus ID** and click Enter.
3. Use the up and down arrow buttons to change the DXM Modbus Slave Port ID.
4. Press Enter to accept the ID change.
5. Use the DXM Configuration Software to cycle power to the device.

After cycling power to the device, the updated DXM Modbus ID is listed under the **System Config** menu.

4.8 Outputs

The base board is a Modbus slave device (Slave ID 203) that communicates to the processor using Modbus commands. Use the DXM Configuration Software to create write maps that will access outputs on the board.

Communication with the board runs at a maximum rate of 10 ms per transaction. The parameter setting for the bus with the board and the processor board are fixed. External Modbus communication runs at a maximum rate of 50 ms per transaction. The parameter settings for the external RS-485 buses are controlled by the DXM Configuration Software.

Refer to the Modbus Registers section for more descriptions of each Modbus register on the DXM700-Bx Wireless Controller.
4.8.1 Modbus I/O Registers for the Base Board

<table>
<thead>
<tr>
<th>Modbus Register</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2101</td>
<td>0–1</td>
<td>PNP Output 1</td>
</tr>
<tr>
<td>2102</td>
<td>0–1</td>
<td>PNP Output 2</td>
</tr>
<tr>
<td>2103</td>
<td>0–1</td>
<td>PNP Output 3</td>
</tr>
<tr>
<td>2104</td>
<td>0–1</td>
<td>PNP Output 4</td>
</tr>
</tbody>
</table>
5 Cellular Modem Boards

5.1 Cellular Modem Board for GSM and LTE

The GSM or LTE cellular modem is an optional accessory that is installed on the base board on the two 12-pin sockets.

The U.FL connector should be to the right, with the antenna cable going to the base board antenna U.FL connector.

A - U.FL antenna connection
The SIM card slides into the socket on the back of this board.

Figure 14. Cellular modem board (optional)

5.2 Cellular Power Requirements

If the incoming voltage drops below 11.2 V dc, the cellular modem does not turn on and will not turn on until the voltage is above 11.8 V dc. A text file (CmVMon.txt) on the internal micro SD card saves the periodic sampling of the incoming voltage. If cellular operation stops because of voltage, it is logged in this file.

5.3 Using the DXM Cellular Modem

The DXM cellular modem provides a remote network connectivity solution for the DXM700.

To use the cellular modem:

1. Verify the cellular modem is installed and the correct antenna (BWA-CELLA-002) is connected to the cellular antenna port.
2. Activate the cellular service.
3. Configure the DXM700 to use the cellular network as the network interface.

5.3.1 Activating a Cellular Modem

Activating the DXM700 cellular capabilities requires these basic steps:

1. Purchase a cellular modem kit from Banner Engineering Corp.
2. Activate a cellular plan to the SIM card, then insert the SIM card into the cellular modem.
3. Install the cellular modem, connect the antenna cable, and connect the cellular antenna.

### Cellular Kit Model Number

<table>
<thead>
<tr>
<th>Cellular Kit Model Number</th>
<th>Kit Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXI-GSM-001</td>
<td>3G GSM worldwide cellular modem using Telit HE910-D modem kit. The kit includes cellular modem, antenna, and antenna cable.</td>
<td>Requires a GSM cellular wireless plan attached to the SIM card, IMEI (International Mobile Equipment Identity) number.</td>
</tr>
</tbody>
</table>
Install the Cellular Modem

Follow these steps to install the cellular modem and antenna cable.

**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.

Before you activate your cellular plan, verify you have one of the following cellular modem kits.

<table>
<thead>
<tr>
<th>Cell Modem Kit</th>
<th>Cell Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>SXI-GSM-001</td>
<td>3G GSM (HSPA+ networks; GSM/EGPRS fallback)</td>
</tr>
<tr>
<td>SXI-LTE-001</td>
<td>Verizon 4G LTE</td>
</tr>
</tbody>
</table>

1. Insert the SIM card into the socket on the underside of the cellular modem. Verizon LTE SIM cards come in a credit card sized carrier. Snap it out and insert the SIM card into the holder on the cellular modem.
   The SIM card number is on the SIM card and on the credit card sized carrier. You will need the SIM number to associate a wireless plan to this SIM card.

2. Orient the cellular modem as shown and verify the pins are properly aligned.
3. Firmly press the modem into the 24-pin socket.
4. Attach the antenna cable as shown.
5. Install the external cellular antenna on the DXM700’s SMA connector located next to the antenna cable.
Cellular modem and antenna installed; note that the antenna cable uses the top antenna connection. The LTE modem is shown; the HE910 GSM installation is the same.

Activate a 3G GSM Cellular Plan
The GSM cellular modem is operational world-wide and requires an activated card to operate. GSM wireless cellular plans cannot be purchased through the Banner Connected Data Solutions by Sensonix website.

1. Work with the local Banner technical support person to identify and purchase machine-to-machine (M2M) (data plan only) SIM cards in 3FF ‘micro’ form factor.
2. When activating the SIM:
   - Note the Access Point Name, or APN, that the SIM provider says to use with their SIM
   - The IMEI is the 15-digit number on top of the cell module PCB, below the words “Telit HE910-D” and above the 2-d bar code
   - The ICCID is the 20-digit number printed on the SIM card itself

Activate a Verizon 4G LTE Cellular Plan
Activate a cellular plan for your DXM700 using the Banner Connected Data Solutions by Sensonix website.

1. Go to https://bannercds.com to purchase DXM700 cellular data plans.
2. If you have previously created an account, enter your username and password to continue.
3. If you are creating a login for the first time:
   a) Select the subscription type and subscription plan.
   b) Create a username and password (use an email address for the username).
   c) Enter your payment information and mailing address.
4. Go to the Activate a New Verizon 4G Device Here section.
5. Enter the SIM Number (ICCID) and the Module Number (IMEI).
   - The ICCID is the 20-digit number of the SIM, the bottom barcode number on the SIM card carrier. If the carrier card is not available, the ICCID is also printed on the SIM card, but must be removed from its socket to be read.
   - The IMEI is the 15-digit number on top of the 4G LTE device.
6. Click Activate.

   Note: Although new activations are typically functional in 20 minutes or less, it may take up to 24 hours for the cellular plan to become active on the wireless network.

Configure the DXM Controller for a Cellular Modem
Use the DXM Configuration Software to create a configuration using a cellular connection.

1. Go to the Settings > Cloud Services screen.
2. Set the Network Interface to Cell
   - All push data or email alerts will be sent using the cellular modem.
3. Under the Cell Configuration section, select the Cell module from the drop-down list.
   - For LTE, select LE910 4G VZW and set APN to vzwinternet. Requires a SIM module to be purchased from a wireless carrier based on the IMEI number of the cellular modem. The wireless carrier will provide the APN parameters. Not all parameters may be required.
• For GSM, select HE910 3G GSM and set the APN based on your provider’s settings. Requires a SIM module to be purchased with a cellular carrier to activate the DXM700 cellular capability. The wireless carrier will provide the APN parameters. Not all parameters may be required.
• For CDMA, select CE910 2G CDMA. Requires a wireless data plan put in place based on the MEID number of the cellular module. Once activated, the DXM700 must be provisioned on the wireless network by using the LCD display menu System Config > Provision Cell. The APN parameters are not required for this cellular modem.

4. To send data to the webserver, set the Cloud Push settings and the Webserver settings.

When the DXM700 is configured to use the cellular modem, the information on the cellular modem is found on the LCD menu under System Info > Cell. The menu does not display values until a transaction with the wireless cell tower is complete. If there are no webserver parameters defined, the user must force a push to retrieve the data from the cellular network. On the LCD menu, select Push > Trigger Data Push.

**Obtaining LTE service outside of Banner CDS**—Customers have the option of securing a data plan for the Verizon network themselves without using the Banner Connected Data Solutions by Sensonix platform. Suitable plans would include those available from Verizon directly or from a Mobile Virtual Network Operator (MVNO) licensed to resell Verizon network data plans. (The SXI-LTE-001 will not function on AT&T, T-Mobile, or Sprint networks.) When purchasing a data plan, it is important to refer to the modem by its official Verizon network name, SENSX002 and give the IMEI number (found on the cellular modem) to the plan provider. To use the SIM card that comes with the cellular modem kit, give the SIM card number to the provider. The required SIM card form factor is 3FF - Micro.
6 LCD and Menu System

The LCD has four user-defined LED indicators, four control buttons, and an LCD display. The four buttons control the menu system on the LCD menu.

The top-level menu always displays the time in a 24-hour format.
- The up and down arrows scroll through display items.
- The enter button selects the highlighted items on the display.
- The back button returns to a previous menu option.

The left display column shows an arrow at the beginning of the line if the menu has submenus. The right column shows a vertical line with an arrow at the bottom if the user can scroll down to see more menu items.

The DXM700 can be configured to require a passcode be entered before the LCD and Menu system will operate. The passcode configuration is defined in the DXM Configuration Software.

6.1 Registers

The Registers submenu displays the processor’s local registers that can be configured using the DXM Configuration Software.

To configure these local registers, launch the DXM Configuration Software. Go to Local Registers and expand the view for a local register by clicking on the down arrow next to the register number. In the LCD Permissions field, select None, Read, Write, or Read/Write.

- Read allows the register to be displayed, and Write or Read/Write allows the register value to be changed using the LCD.
- The Units and Scaling parameters are optional and affect the LCD.

6.2 Push

The Push menu displays information about the last data sent to the Webserver.

The user can force an immediate push to the webserver using Trigger Push. If a current push is in process it may take several minutes to complete over cellular.
- The Trigger Push submenu forces an immediate push to the web server.
- The status and time fields indicate success or failure of the last attempted push and time of the last attempted push.
6.3 ISM Radio

The ISM Radio menu allows the user to view the Modbus ID of the internal ISM radio, enter binding mode, or run a site survey. This top-level ISM Radio menu is different from the System Config > ISM Radio submenu.

The first option under the ISM Radio menu only displays the type of radio in the DXM700 (MultiHop or DX80 Star) and the Modbus ID of the radio. To change the ISM Radio Modbus ID refer to the System menu.

Select Binding to enter binding mode or select Site Survey to run a site survey.

Binding — All ISM radio devices must be bound to the internal Gateway/master device before the DXM700 can access the wireless devices. The first submenu under binding allows the user to set the wireless address of the device to bind with. This is required to bind with wireless devices that do not have rotary dials (for example: M-GAGEs, ultrasonic sensors, and Q45 devices). See Binding and Conducting a Site Survey with the ISM Radio (p. 11). For more information on binding a particular device, refer to the individual datasheet.

Site Survey — After creating a wireless network using the binding process, run a site survey on each device to see the link quality. See Conduct a Site Survey (p. 12).

6.4 I/O Board

The DXM700 does not have an I/O board. DXM700 has four outputs accessed using Modbus registers.
6.5 System Config

Use the **System Config** menu to set DXM700 system parameters.

**System Config** menu:
- ISM Radio
- DX80 ID: x
  - Auto Detect Radio
  - New ISM Modbus ID: x
  - Radio Detected Type: DX80
  - ID: 1
  - Advanced Options
  - Ref Type: DX80
  - Ref Modbus ID: x
  - Max Node Count: xx
  - New ISM Max Nodes: xx
  - Binding #: xxxxxx
  - New ISM Binding Code: xxxxxx
- Ethernet
  - DHCP
  - Update DHCP Mode
  - IP:
  - Update IP Address
  - SN:
  - Update SN
  - GW:
  - Update GW Address
- Provision Cell
  - DXM Modbus ID: xxx
  - LCD Contrast: xx
- Reset
  - Resets Ethernet parameters to xml defaults.

The **System Config** submenus are:
- ISM Radio
- Ethernet
- Provision Cell
- DXM Modbus ID
- LCD Contrast
- Restart

**6.5.1 ISM Radio**

**DX80/MultiHop ID** — The ISM radio is set at the factory to be Modbus device address 1 (Modbus ID 1). For some applications, you may need to change the Modbus ID. Adjust the Modbus device address using the LCD menu system. Any other method may cause issues with the DXM700 not knowing which Modbus device address is assigned to the radio, which causes issues with running Binding or Site Survey from the LCD menu.

Set the radio Modbus ID to a valid number (1 through 247) that is not being used by the DXM system. Processor Local Registers allocate ID 199, the I/O board is set to ID 200, and the display board is set to ID 201. With a DX80 Gateway (star network), it’s easy to choose a new ID. With a MultiHop network, remember that the master MultiHop radio allocates a range of Modbus IDs for wireless devices, typically 11 through 110.

When setting the new ISM Modbus ID, the system changes the Modbus ID on the internal radio and changes the reference to it on the DXM700. The reference Modbus ID is what the DXM700 uses to access the internal radio when running Binding or Site Survey.

**Auto Detect Radio** — If the internal Modbus ID of the radio was changed or the internal radio was changed, but not recorded, use Auto Detect Radio to determine the radio ID and radio type. The auto-detect routine broadcasts discovery messages and waits for a response. If other devices are connected to the external RS-485 ports, they may need to be disconnected for this process to work properly.

**Advanced Options** — The Advanced Options menu is typically not used unless the Modbus ID is changed without the DXM700 being involved, such as when you write directly to the radio Modbus registers.
- **Reference Type** selects the radio type between DX80 star architecture radios and a MultiHop radio. The DXM700 uses this reference to determine how to communicate to the internal radio. If set incorrectly, the DXM700 may not be able to run Site Survey from the LCD menu. Unless you are changing or adding the internal radio device, there should be no reason to change the radio type.
- **Reference Modbus ID** defines the Modbus ID the DXM700 uses when communicating with the internal radio. If this is set incorrectly, the DXM700 will not be able to run Binding or Site Survey through the LCD menu.
Max Node Count—Defines the maximum number of devices for the DX80 wireless network.

Binding #—This parameter allows the user to define the Binding code within the ISM radio. Typically, you will not have to adjust this number unless you are replacing an existing Gateway or master radio.

RF Ctrl—Displays the status of the ISM radio DIP switch 1 (off or on). The menu doesn’t allow the user to change the DIP switch setting through the display.

6.5.2 Ethernet

Use the Ethernet submenu to sets the IP Address, Gateway Address, and Subnet mask of the DXM700’s Ethernet interface. You may change these settings either from the LCD menu (System Config > Ethernet) or from the XML configuration file created by the DXM Configuration Software.

The network address settings from the LCD menu have the highest priority and override settings in the XML configuration file. To use the parameter settings from the XML configuration file or use DHCP, execute the Reset under System Config > Ethernet or use the LCD display to set the IP Address, Gateway Address, and Subnet Mask to 255.255.255.255. Reboot the DXM700 after changing the Ethernet parameters.

The Ethernet cable should be attached before powering up the DXM700.

6.5.3 DXM Modbus ID

Use the secondary Modbus RS-485 port when the DXM700 is connected to a Modbus RTU network as a Modbus slave device. Set the Modbus ID for the secondary RS-485 port using the LCD display menu System Config > DXM Modbus ID.

6.5.4 LCD Contrast

Use the LCD Contrast option to adjust the LCD contrast. Adjust the starting number lower to decrease the display contrast. The factory default is 28. Do not set a number less than 15 or the display may not be bright enough to see to change back.

6.5.5 Reset

Use the Restart menu to force the main processor to restart. This does not affect the other boards in the system.
6.6 System Info

Various DXM system settings are shown in this menu. The Push, Ethernet, and Cell parameters are helpful for debugging network connections. This is a read only menu.

**Controller**
Displays the date, build, model, and serial number.

**Push**
Shows the current parameters loaded from the XML configuration that applies to pushing data to a webserver, including method (Ethernet or cellular), interval, URL, page, HTTPS, and site ID.

**ISM Radio**
Displays the serial number, model, date, firmware part numbers, and version numbers.

**Ethernet**
Displays the IP address, MAC address, DHCP, Gateway address, and DNS settings.

**Cell**
Shows the cellular MEID number (Mobil Equipment Identifier), MDN (Mobil Device Number), version, signal, firewall setting, and firewall mask. Some of these parameters are not visible until the cellular network is accessed.
6.7 Display Lock

Display Lock protects the DXM LCD menu system from being used until the proper pass code is entered.

Display Lock → Enter password
   ➔ ➔ to change the value
   ENTER to accept
   BACK to return to the previous menu

The display lock feature uses the DXM Configuration Software to set a passcode within the DXM700. A valid passcode is 1 to 9 digits long and uses numbers 0 through 9. For example 1234 or 209384754.

6.8 Modbus Registers for the LCD Board (Modbus Slave ID 201)

Control the four bi-color LEDs using the display board's Modbus registers. Using write maps or ScriptBasic, write the Modbus registers shown below with 0 (off) or 1 (on). The LCD display is Modbus Slave 201.

<table>
<thead>
<tr>
<th>Modbus Register</th>
<th>LED</th>
<th>Color</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1102 : bit 0</td>
<td>LED 1</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>1103 : bit 0</td>
<td>LED 2</td>
<td>Amber</td>
<td></td>
</tr>
<tr>
<td>1104 : bit 0</td>
<td>LED 3</td>
<td>Red</td>
<td></td>
</tr>
<tr>
<td>1105 : bit 0</td>
<td>LED 4</td>
<td>Amber</td>
<td></td>
</tr>
<tr>
<td>1107 : bit 0</td>
<td>LED 1</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>1108 : bit 0</td>
<td>LED 2</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>1109 : bit 0</td>
<td>LED 3</td>
<td>Green</td>
<td></td>
</tr>
<tr>
<td>1110 : bit 0</td>
<td>LED 4</td>
<td>Green</td>
<td></td>
</tr>
</tbody>
</table>

1 = On
0 = Off
7 Working with Modbus Devices

The DXM700 has two physical RS-485 connections using Modbus RTU protocol.

The master Modbus RS-485 port is for the DXM700 to act as a Modbus master device to control internal and external Modbus slave devices.

The Modbus master RS-485 port is labeled M+ and M- on the DXM700. The Modbus slave port is used when another Modbus master device wants to communicate with the DXM700 when the DXM700 is a Modbus slave device.

The Modbus slave RS-485 port is labeled S+ and S1 on the DXM700.

![Figure 18. DXM700 system overview](image)

The DXM700 has dual Modbus roles: a Modbus slave device and a Modbus master device. These run as separate processes.

The Modbus slave port can only access the DXM700 local registers. To operate as a Modbus slave device, the DXM700 needs to be assigned a unique Modbus slave ID as it pertains to the host Modbus network. This slave ID is separate from the internal Modbus slave IDs the DXM700 uses for its own Modbus network. The DXM Modbus slave ID is defined through the LCD menu. Other Modbus slave port parameters are defined by using the DXM Configuration Software.

The DXM700 operates the Modbus master port. Each device on the master port must be assigned a unique slave ID. There are slave IDs that are reserved for internal devices in the DXM700.

<table>
<thead>
<tr>
<th>DXM Internal Modbus Slave IDs (factory default)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Modbus Slave ID</strong></td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>199</td>
</tr>
<tr>
<td>203</td>
</tr>
<tr>
<td>201</td>
</tr>
</tbody>
</table>

7.1 Assigning Modbus Slave IDs

Assign the DXM Modbus Slave ID only if a Modbus master device is reading or writing the DXM700 Local Register data through the Modbus RS-485 slave port (S+, S-).
Set the DXM Slave ID from the LCD menu under **System > DXM Slave ID**. The DXM700 can have any unique slave ID between 1 and 246, depending upon the host Modbus network. Other RS-485 slave port parameters are set in the DXM Configuration Software under the **Settings > General** tab.

**DXM Master Configuration**—When the DXM700 operates as a Modbus master device, use the DXM Configuration Software to configure read or write operations of the DXM Modbus network. The DXM700 communicates with all internal and external peripheral devices using the external Modbus bus RS-485 (M+, M-).

There are four internal Modbus slave devices that are configured from the factory with slave IDs. Assign slave IDs of 2 through 10 to Modbus slave devices that are physically wired to the DXM700. Assign slave IDs of 11 through 60 to wireless slaves within the MultiHop network.

Do not assign a slave ID of greater than 10 to Modbus slave devices that are physically wired using the RS-485 port if there is an internal MultiHop ISM radio in the DXM700. The MultiHop ISM radio attempts to send any Modbus data intended for slaves 11–60 across the radio network, which conflicts with wired slave devices if the slave IDs overlap. The MultiHop master radio can be changed from the factory default of 11–60 Modbus slave IDs if more hardwired slaves are required.

### 7.2 Modbus Operation

All Modbus transactions are managed by a central Modbus engine. If there are Modbus messages intended for a Modbus slave that doesn’t exist, the Modbus engine waits for a response until the timeout period is expired. This slows down the Modbus polling loop for read and write operations. For this reason, verify all Modbus read and write operations are intended for Modbus slave devices that are in the network.

If a Modbus slave is not in the network, either a wired or wireless device, the operation of the LCD menu system can be compromised. Operations like Binding, Site Survey, or accessing the ISM menu may be slower. This is because all internal devices of the DXM700 are also Modbus slaves, ISM radio, I/O base board, LCD, and internal Local registers.

### 7.3 Wireless and Wired Devices

**Wireless DX80 Gateway**—The DX80 Gateway architecture is a star architecture in which all Nodes in the system send their data back to the Gateway. The host can access the entire network data from the Gateway, which is Modbus slave ID 1. Because the DXM700 will not be sending any Modbus messages across the wireless link, the timeout parameter can be set low (less than 1 second) and the device is treated like a directly connected device.

**MultiHop Master**—The MultiHop master radio forms a wireless tree network using repeaters and slave devices. Each device in a MultiHop network must be assigned a unique Modbus Slave ID and is accessed as a separate device.

For the DXM700 to talk with a MultiHop device in the wireless network, the master MultiHop device interrogates every message on the RS-485 bus. If they are within the wireless devices range (slave IDs 11 though 60), the message is sent across the wireless network. To change this range, the user must adjust the offset and range setting in the MultiHop master radio (Modbus Slave ID 1). Modbus register 6502 holds the Modbus offset, default 11. Modbus register 6503 holds the number of Modbus slaves allowed (maximum of 100).

<table>
<thead>
<tr>
<th>Modbus Slave ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Allocated for the internal ISM radio device, either a DX80 Gateway or MultiHop Master</td>
</tr>
<tr>
<td>2–10</td>
<td>Slave addresses available for direct connected Modbus slave devices to the master RS485 port (M+, M-)</td>
</tr>
<tr>
<td>11–60</td>
<td>Allocated for wireless MultiHop radio network devices. If there is not an internal MultiHop in the DXM700, these slave addresses are available to use for directly connected devices.</td>
</tr>
<tr>
<td>61–198</td>
<td>Available to user for direct connected Modbus slave devices or the expansion of the wireless network slave IDs to go past 50 wireless devices.</td>
</tr>
<tr>
<td>199</td>
<td>Allocated for internal Local Register</td>
</tr>
<tr>
<td>200</td>
<td>Allocated for the I/O base board, will be different for special DXM slave only models.</td>
</tr>
<tr>
<td>201</td>
<td>Allocated for the LCD display board, the user can read/write LEDs.</td>
</tr>
</tbody>
</table>

### 7.4 Modbus Communication Timeouts

A Modbus timeout is the amount of time a Modbus slave is given to return an acknowledgment of a message sent by the Modbus master. If the Modbus master waits for the timeout period and no response is seen, the Modbus master considers it a lost message and continues on to the next operation.
The timeout parameter is simple to set for Modbus devices directly connected to the DXM700, if there are no MultiHop
wireless devices. Special considerations need to be made to set the timeout parameter when a MultiHop network uses the
DXM700 as the master radio.

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 DXM700, the Communications Timeout parameter is the maximum amount of time the DXM700 should wait after a
request is sent until the response message is received from the Modbus slave device. Use the DXM Configuration Software
to set the timeout parameter on the Settings > System screen (select Show advanced settings).
The default setting for the timeout parameter is 5 seconds.

7.4.1 MultiHop Networks vs DX80 Star Networks

The DX80 star Gateway collects all the data from the Nodes, which allows the host system to directly read the data from the
Gateway without sending messages across the wireless network. This allows for DX80 Gateway to be treated like any other
wired Modbus device.

In a MultiHop network, the data resides at each device, forcing the controller to send messages across the wireless network
to access the data. For this reason, carefully consider the value of the wireless timeout parameter.

7.4.2 Calculating the Communications Timeout for Battery-Powered
MultiHop Radios

Battery-powered MultiHop radios are configured to run efficiently to maximize battery life. By optimizing battery life, the
allowed communications window for receive messages is slow (once per 1.3 seconds) and sending message rates are
standard (once per 0.04 seconds).

A MultiHop device is set from the factory with the retry parameter of 8. This means that under worst-case conditions, a
message is sent from the DXM700 to an end device a total of nine times (one initial message and eight retry messages). The
end device sends the acknowledgment message back to the DXM700 a maximum of nine times (one initial message and
eight retries). A single Modbus transaction may send up to two messages + 16 retry messages before the transaction is
complete. In addition, the radios randomly wait up to one time period before retransmitting a retry message. So to allow for
the random wait time, add one extra time period for each in-between time of retries.

To calculate the communication timeout parameter for a Master radio to a slave radio (no repeaters):

- Master to Slave Send time = (9 × 1.3 sec) + (8 retry wait × 1.3 sec) = 22 seconds
- Slave to Master Send time = (9 × 0.04 sec) + (8 retry wait × 0.04 sec) = 1 second
- Total Send/Receive time = 23 seconds
- Minimum Timeout period = 23 seconds

If the link quality of the network is poor, the maximum transfer times may happen. Set the timeout parameter to
accommodate the maximum number of retries that may happen in your application.

When MultiHop repeaters are added into the wireless network, each additional level of hierarchical network increases the
required timeout period. Since MultiHop repeaters are running at the highest communications rate, the overall affect is not
as great.

- Master to Repeater Send time = (9 × 0.04 sec) + (8 retry wait × 0.04 sec) = 1 second
- Repeater to Master Send time = (9 × 0.04 sec) + (8 retry wait × 0.04 sec) = 1 second
- Additional Timeout period for a repeater = 2 seconds

Using the timeout calculation above of 23 seconds, if a repeater is added to the network the timeout should be set to 25
seconds. For each additional MultiHop repeater device creating another level of network hierarchy, add an additional two
seconds to the timeout period.

7.4.3 Calculating the Communication Timeout for 10–30 VDC
MultiHop Radios

Line-powered (10–30 V dc) MultiHop devices operate at the maximum communication rate, resulting in a much lower
timeout parameter setting. For each repeater added to the network, increase the timeout parameter 2 seconds.

For a Master radio to a 10–30 V dc powered slave radio (no repeaters):

- Master to Slave Send time = (9 × 0.04 sec) + (8 retry wait × 0.04 sec) = 1 second
- Slave to Master Send time = (9 × 0.04 sec) + (8 retry wait × 0.04 sec) = 1 second
- Total Send/Receive time = 2 seconds

www.bannerengineering.com - Tel: + 1 888 373 6767
Minimum Timeout period = 2 seconds

7.4.4 Adjusting the Receive Slots and Retry Count Parameters

The number of receive slots governs how often a MultiHop device can communicate on the wireless network.

Battery-powered devices typically have DIP switches that allow the user to set the number of receive slots, which directly affects the battery life of the radio. Adjusting the receive slots changes how often a message can be received. By default, the receive slots are set to 4 (every 1.3 seconds). When the receive slots are set to 32, the radio listens for an incoming message every 0.16 seconds.

Users may also leave the retry mechanism to the application that is accessing the wireless network, in this case the DXM700. Adjust the number of retries in the MultiHop devices by writing the number of retries desired to Modbus register 6012. The factory default setting is 8.

7.4.5 Calculating the Communication Timeout for a DX80 Star Network

In the DX80 network, all Node data is automatically collected at the Gateway to be read. The DXM700 does not use the wireless network to access the data, which allows for much faster messaging and much lower timeout values.

For a DXM700 with an internal DX80 Gateway, set the timeout value 0.5 seconds. If other Modbus slave devices are connected to the RS-485 lines, the timeout parameter governs all communication transactions and must be set to accommodate all devices on the bus.

7.5 Modbus TCP Client

The DXM700 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.
8 Configuration Instructions

8.1 Scheduler

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

Schedules are stored in the XML configuration file, which is loaded to the DXM700. Reboot the DXM700 to activate a new schedule.

If power is cycled to the DXM700 in the middle of a schedule, the DXM700 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.

### 8.1.1 Create a Weekly Event

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

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.

### 8.1.2 Create a One-Time Event

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

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

### 8.1.3 Create a Holiday Event

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

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.

### 8.2 Authentication Setup

The DXM700 has three different areas that can be configured to require a login and password authentication.

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

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

#### 8.2.1 Set the Controller to use Authentication

The DXM700 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 DXM700 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 DXM700.

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.

   ![Web Server Authentication screen](Figure 22. Web Server Authentication screen)

   The first time you select **Require Authentication**, a pop-up box appears with additional instructions. Since the data is not stored in the XML configuration file, it is hidden from view of the DXM Configuration Software.
4. Click on **Send Authentication**.
   The controller must be connected to the PC for this operation to succeed.
   The data transmits directly to the DXM700’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 Banner Connected Data Solutions by Sensonix 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](image)
   
   At the bottom of the pop-up window is a checkbox to enable authentication/validation.
3. Enter the same username and password as used in the DXM Configuration Software. The username and password do not need to be a defined user within the Banner Connected Data Solutions by Sensonix website.

### 8.2.2 Controller Configuration Authentication

The DXM700 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 DXM700 connected to the PC, click **Get Device Status**. The DXM700 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, the current password must be supplied. The DXM700 must be connected to the PC to change the administration password.
8.3 Register Flow and Configuration

The DXM700 register data flow goes through the Local Registers, which are data storage elements that reside within the processor. Using the DXM Configuration Software, the controller can be programmed to move register data from the Local Register pool to remote devices, the internal radio, the I/O base, or the display.

![Diagram of Register Flow](image)

**8.3.1 Basic Approach to Configuration**

When programming an application in the DXM700, first plan the overall data structure of the Local Registers. The Local Registers are the main storage elements in the DXM700. Everything goes into or out of the Local Registers.

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

**8.3.2 Troubleshooting a Configuration**

View Local Registers using the **Local Registers > Local Registers in Use** screen of the DXM Configuration Software.

When a configuration is running on the DXM700, viewing the Local Registers can help you to understand the application’s operation. This utility can also access data from remote devices.

To configure the Local Register data to display on the LCD menu, go to the **Local Registers** screen, set the **LCD permissions** to read or read/write.

**8.3.3 Saving and Loading Configuration Files**

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

Save the configuration file before attempting to upload the configuration to the DXM700. The DXM Configuration Software uploads the configuration file saved on the PC to the DXM700; it will not send the configuration loaded in the tool.
8.3.4 Uploading or Downloading Configuration Files

The DXM700 requires a XML configuration file to become operational. To upload or download configuration files, connect a computer to the DXM700 using the USB port or Ethernet port. Then use the Upload Configuration to Device or Download Configuration from Device under the Device menu.

8.4 EtherNet/IP™ Configuration

The DXM700 is defined from the factory to send/receive register data from the Gateway and the first 16 Nodes with an EtherNet/IP™ host.

To expand the number devices going to Ethernet/IP, change the Devices in system parameter in the DX80 Gateway (default setting is 8) to 32. To change this value:

1. Launch the DX80 Configuration Software.
2. In the menu bar, go to Device > Connection Settings and select Serial or Ethernet DXM.
3. In the Configuration > Device Configuration screen, click on the arrow next to the Gateway to expand and display the Gateway's parameters.
4. In the System section, use the Devices in system drop-down list to make your selection.

This allows the user to maximize the use of the EtherNet/IP buffer to 28 devices.

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)

8.4.1 Configuring the Host PLC

On the host PLC, install the DXM700 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 DXM700. The host system sees the DXM700 as a generic device with the product name of Banner DXM (ProdType: 43 - Generic Device, ProdName: Banner DXM).

8.4.2 Configuring the Controller

Use the DXM 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 DXM700 local register (DXM).
- Define a DXM local register as EIP DXM -> Originator when that register data will be sent from the DXM700 (DXM) to the host PLC (Originator).

Data from an EIP controller in assembly instance 112 is data destined for the DXM700 local registers. The first two bytes of the assembly instance are stored in the first local register defined as an EIP Originator -> DXM register. For example, if DXM local registers 5, 12, 13, and 15 are configured as EIP Originator -> DXM registers, the assembly instance 112 will have the first eight bytes of data coming from the DXM local registers 5, 12, 13, and 15. The system ignores the rest of the bytes in the assembly instance.

Data from the DXM700 local registers is sent to the EIP controller using assembly instance 100. Each local register in the DXM700 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 for each local register are transferred. For example, if DXM registers 1, 10, 20, and 21 are defined as EIP DXM -> Originator registers, the assembly instance 100 will have the first eight bytes of data coming from the DXM local registers 1, 10, 20, and 21. The rest of the data is in assembly instance 100 is zero.

For example, define register 1 as an EIP Originator -> DXM (target) register. The EIP PLC will write data into register 1. For local registers to be sent to the EIP controller, define registers as EIP DXM -> Originator. Use the Modify Multiple Registers screen to change many registers parameters at one time.

The DXM700 is big endian: the upper bits of a local register (15:8) are stored in the first byte of the assembly instance and the second byte of the assembly instance is stored in the lower bits of a local register (7:0).

---

1 EtherNet/IP is a trademark of Rockwell Automation.

www.bannerengineering.com - Tel: + 1 888 373 6767
The following table shows DXM local registers 1, 5, and 10 being written from the EIP controller using assembly instance 112. Only registers 1, 5, and 10 are defined in the DXM Configuration Software as EIP Originator -> DXM registers.

<table>
<thead>
<tr>
<th>EIP Assembly Instance 112</th>
<th>DXM Local Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adrs</td>
<td>Data</td>
</tr>
<tr>
<td>00</td>
<td>11</td>
</tr>
<tr>
<td>01</td>
<td>22</td>
</tr>
<tr>
<td>02</td>
<td>33</td>
</tr>
<tr>
<td>03</td>
<td>44</td>
</tr>
<tr>
<td>04</td>
<td>66</td>
</tr>
<tr>
<td>05</td>
<td>88</td>
</tr>
</tbody>
</table>

The following table shows DXM local registers 10, 11, and 19 defined as EIP DXM -> Originator. The lower 2-bytes of each register data is placed into assembly instance 100.

<table>
<thead>
<tr>
<th>EIP Assembly Instance 112</th>
<th>DXM Local Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adrs</td>
<td>Data</td>
</tr>
<tr>
<td>00</td>
<td>77</td>
</tr>
<tr>
<td>01</td>
<td>88</td>
</tr>
<tr>
<td>02</td>
<td>99</td>
</tr>
<tr>
<td>03</td>
<td>10</td>
</tr>
<tr>
<td>04</td>
<td>11</td>
</tr>
<tr>
<td>05</td>
<td>12</td>
</tr>
</tbody>
</table>

8.5 Setting up Email

The DXM700 can be configured to send email based on threshold conditions. Internal log files may be sent using email. Cellular-connected systems can use email. 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.

Follow these instructions and use the DXM Configuration Software to program the controller for email.

1. Set the time on the DXM700.
2. Define the Network Interface settings by selecting either Ethernet or Cell on the Cloud Services screen.
3. If you selected Ethernet, configure your Ethernet connection by setting the IP settings on the Network screen.
4. Set the email and message parameters on the Mail and Messaging screen.
5. To send alert messages, define the threshold rule to use email.
6. To send log files, define the log file parameters.

8.5.1 Mail Server Authentication

Complete the mail server settings to have the DXM700 send email alert messages or to email the log files. The SMTP password is stored in the DXM700, not the XML configuration file. Use the Settings > Notifications screen to complete this configuration.
After selecting **Enable SMTP Authentication** for the first time, a pop-up box appears with additional instructions to complete the mail server authentication process.

After entering the user name and password, click on **Send SMTP Password** to save the user name and password to the DXM700. The DXM700 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.

### 8.5.2 Define the Network Interface Settings

On the **Cloud Services** screen (shown with **Show advanced settings** selected), define the network connection settings by selecting Ethernet or Cell from the **Network Interface** drop-down list. This determines how the DXM700 sends data.

If you don’t require pushing data to a web server, set the **Cloud Push** interval to zero.
8.5.3 Configure your Ethernet Connection

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

1. To define the Ethernet IP address, give the DXM700 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 DXM700 uses a public service to resolve Domain names, but if the network connection does not have Internet access, the DNS settings may be required.

8.5.4 Configure your Cellular Connection

To use a cellular connection, select Cell as the network connection on the Settings > Cloud Services screen (see Configure the DXM Controller for a Cellular Modem (p. 33)). The Cellular screen does not display unless the Network interface is set to Cell.

Using a 4G LTE cell module requires a cellular plan; follow the instructions on p/n 205026 to activate a cell modem.

1. On the Settings > Cellular screen, select your cellular modem from the drop-down list.

2. Set the APN.
   - If you are using a Banner 4G LTE Verizon Module (LE910), set the APN to vzwinternet.
   - If you are using an Emnify 3G GSM Cellular Radio (HE910), set the APN to EM. This module does not require an APN username or password.
   - If you are using a third-party SIM card, the APN, APN Username, and Password must be provided by the cellular service provider.

8.5.5 Set the Email and Messaging Parameters

From the Settings > Mail and Messaging screen, enter the SMTP definition, login, and password for a mail server. You must supply the SMTP Server, Server Port, and login credentials to send email.

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 DXM700. After the password is entered, click on Send SMTP Password to send it to the DXM700. The password is stored in non-volatile memory, so reboot the DXM700 to recognize the new password.
At the bottom of the screen, define the recipient to receive emails. These recipients for email can be selected in the threshold definition for sending alert messages.

8.5.6 Define Threshold Rules for Email

To define a threshold, go to Action Rules > Thresholds. Depending upon which recipients are defined, select the appropriate email checkbox for the threshold rule (under Email/SMS on state transition). When the threshold rules go active or inactive, an email is generated.

8.5.7 Define Log File Parameters for Emailing Log Files

The DXM700 can email log files generated on the device.

Before emailing log files, set the Mail and Messaging parameters to provide the login credentials. When using Ethernet, verify the IP address settings are defined on the Ethernet screen. Set the DXM700 time, under Settings > System, so that all data is properly time stamped.
Use the **Local Registers > Local Registers in Use > Edit Register** screen to select which registers to log which log file (set the **SD card logging** to the log file. Define the setup of the log file using the **Settings > Logging** screen. Typical settings are shown.

1. Enable the log and timestamp with every entry.
2. Enter the filename, log rate, and the maximum file size to send via email (5 to 10k is an efficient size for a cellular connection). Banner does not recommend setting the log file size larger than 100 kB as this cannot be read through the configuration software and must be read directly from the SD card.
3. Define the email address.

**Figure 31. Defining the data log files**

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

### 8.6 Ethernet and Cellular Push Retries

The DXM700 can be configured to send register data packets to a webserver. When the Ethernet or cell communications path is not operating, the DXM700 retries the send procedure. The communications retry process is outlined below for each configuration.

Regardless of the communications type (Ethernet or cellular), a failed attempt results in the register data packet being saved on the local micro SD card. The number of retries will depend upon the network connection type.

When there is bad cellular signal strength or there is no Ethernet connection, the transmission attempts are not counted as failed attempts to send data. Only when there is a good network connection and there are 10 failed attempts will the controller archive the data on the SD card. Data archived on the SD card must be manually retrieved.

#### 8.6.1 Ethernet Push Retries

With an Ethernet-based network connection, the DXM700 retries a message five times. The five retry attempts immediately follow each other. After all attempts are exhausted, the register data packet is saved on the micro SD card.

At the next scheduled time, the DXM700 attempts to send the saved packet as well as the newly created register data packet. If it cannot send the new register data packet, the new register data packet is appended to the saved file on the micro SD card to be sent later. After 10 rounds of retries, the data set is archived on the micro SD card under folder _sxi. No additional attempts to resend the data are made; the data file must be manually retrieved.

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

#### 8.6.2 Cellular Push Retries

In a cellular-connected system there are no retries. Failed transmissions are saved on the micro SD card.

After 10 successive failed attempts, the data is archived in the _sxi folder. Send attempts with a low signal quality are not counted against the 10 count limit. For example, if the cellular antenna is disconnected for period that the DXM controller would have sent 20 messages under normal circumstances, all 20 messages would be saved and will be retried when the

---

8. Enable HTTP logging to save data on the SD card; this is the factory default. See SETTINGS -> LOGGING in the DXM Configuration Tool.
If the antenna is reconnected. If the signal quality was good, but the cellular network was not responding, the DXM700 archives the register data packets after 10 failed attempts.

8.6.3 Event/Action Rule or Log File Push Retries

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

8.6.4 Email Message Push Retries

There are no retries for emails that fail to be sent from the DXM700.
9 Accessories

For a complete list of all the accessories for the Sure Cross wireless product line, please download the Accessories List (p/n b_3147091).

Cordsets
- MQDC1-506—5-pin M12/Euro-style, straight, single ended, 6 ft
- MQDC1-530—5-pin M12/Euro-style, straight, single ended, 30 ft
- MQDC1-506RA—5-pin M12/Euro-style, right-angle, single ended, 6 ft
- MQDC1-530RA—5-pin M12/Euro-style, right-angle, single ended, 30 ft

Static and Surge Suppressor
- BWC-LFNBMN-DC—Surge Suppressor, bulkhead, N-Type, dc Blocking, N-Type Female, N-Type Male

Misc Accessories
- BWA-CG.5-3X5.6-10—Cable Gland Pack: 1/2-inch NPT, Cordgrip for 3 holes of 2.8 to 5.6 mm diam, qty 10
- BWA-HW-052—Cable Gland and Vent Plug Pack: includes 1/2-inch NPT gland, 1/2-inch NPT multi-cable gland, and 1/2-inch NPT vent plug, qty 1 each

Antenna Cables
- BWC-1MRSMN05—LMR100 RP-SMA to N-Type Male, 0.5 m
- BWC-2MRSRFS6—LMR200, RP-SMA Male to RP-SMA Female Bulkhead, 6 m
- BWC-4MNFS6—LMR400 N-Type Male to N-Type Female, 6 m

Short-Range Omni Antennas
- BWA-2O2-D—Antenna, Dome, 2.4 GHz, 2 dBi, RP-SMA Box Mount
- BWA-9O2-D—Antenna, Dome, 900 MHz, 2 dBi, RP-SMA Box Mount
- BWA-9O2-RA—Antenna, Rubber Fixed Right Angle, 900 MHz, 2 dBi, RP-SMA Male Connector

Medium-Range Omni Antennas
- BWA-9O5-C—Antenna, Rubber Swivel, 900 MHz 5 dBi, RP-SMA Male Connector
- BWA-2O5-C—Antenna, Rubber Swivel, 2.4 GHz 5 dBi, RP-SMA Male Connector

Long-Range Omni Antennas
- BWA-9O8-AS—Antenna, Fiberglass, 3/4 Wave, 900 MHz, 8 dBi, N-Type Female Connector
- BWA-2O8-A—Antenna, Fiberglass, 2.4 GHz, 8 dBi, N-Type Female Connector

Long-Range Yagi Antennas
- BWA-9Y10-A—Antenna, 900 MHz, 10 dBd, N-Type Female Connector

Cellular Antenna
- BWA-CELLA-002—Cellular multiband, 2 dBi, RP-SMA male connection, 6.3 inch blade style. Datasheet: b_4475176

Enclosures and DIN Rail Kits
- BWA-AH864—Enclosure, Polycarbonate, with Opaque Cover, 8 × 6 × 4
- BWA-AH1084—Enclosure, Polycarbonate, with Opaque Cover, 10 × 8 × 4
- BWA-AH12108—Enclosure, Polycarbonate, with Opaque Cover, 12 × 10 × 6
- BWA-AH8DR—DIN Rail Kit, 8", 2 trilobular/self-threading screws
- BWA-AH10DR—DIN Rail Kit, 10", 2 trilobular/self-threading screws
- BWA-AH12DR—DIN Rail Kit, 12", 2 trilobular/self-threading screws

Power Supplies
- PSD-24-4—DC Power Supply, Desktop style, 3.9 A, 24 V dc, Class 2, 4-pin M12/Euro-style quick disconnect (QD)
- PSDINP-24-06—DC Power Supply, 0.63 Amps, 24 V DC, with DIN Rail Mount, Class I Division 2 (Groups A, B, C, D) Rated
- PSDINP-24-13—DC Power Supply, 1.3 Amps, 24 V DC, with DIN Rail Mount, Class I Division 2 (Groups A, B, C, D) Rated
- PSDINP-24-25—DC Power Supply, 2.5 Amps, 24 V DC, with DIN Rail Mount, Class I Division 2 (Groups A, B, C, D) Rated
- PSW-24-1—DC Power Supply, 100-240 V AC 50/60 Hz input, 24 V DC 1 A output, UL Listed Class 2
10 Product Support and Maintenance

10.1 File System and Archive Process

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

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

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

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

Data Log files

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

HTTP Push File

If the DXM700 is configured to send data to a webserver or host system, the device creates an HTTP.LOG file on the SD card. The HTTP log is created only if the Logging Interval is non-zero and the HTTP enable log is set. An entry is placed in the HTTP log file at the Logging Interval specified by the user. At the Push Interval time, the HTTP log file is sent to the webserver or host system. If the transmission is successful, the HTTP log file is time stamped and placed into the archive directory (_sxi). If the transmission fails, the file remains in the root directory and subsequent Logging Intervals are appended to the file and are sent at the next Push Interval. See Ethernet and Cellular Push Retries (p. 54).

User Created ScriptBasic Files

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

ScriptBasic Program File

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

CmVMon File

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

<table>
<thead>
<tr>
<th>Date/Time</th>
<th>Power Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM 2015-09-22 18:52:43</td>
<td>Power entered normal range 24487</td>
</tr>
<tr>
<td>CM 2015-10-13 20:49:47</td>
<td>Power entered normal range 24004</td>
</tr>
<tr>
<td>CM 2015-10-16 15:00:20</td>
<td>Power entered normal range 24014</td>
</tr>
<tr>
<td>CM 2015-10-19 19:12:26</td>
<td>Power entered normal range 12845</td>
</tr>
</tbody>
</table>

_sxi Archive Directory

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

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

1. Using the DXM Configuration Software version 3 or later, connect to the DXM700 via USB or Ethernet.
   File loads to the DXM700 will take about 15 minutes using USB or approximately 2 minutes using Ethernet.
2. On the DXM Configuration Software, go to **Settings > General > Device Information** to verify the current firmware version.
   You must load a different version with the same firmware number for the boot loader to operate. Download firmware files from the Banner website.

![Device Information](image)

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

After the file load is completed, the DXM700 restarts and loads the new firmware file. It takes about 2 minutes to complete the programming process. The device reboots when finished. Verify the firmware has been updated, under **Settings > General > Device Information**.

10.3 Clear the Password on DXM700-Bx, DXM1000-Bx, or DXM1500-Bx Models

By default, the DXM Controllers do not require a password to load a configuration file. If a password is defined, the password must be entered before uploading a configuration file. To change the password, you must already know the current password. If you do not know the current password, follow these steps to clear the password.

**Important:** Clearing the password erases the current configuration and any program files, log files, or history files.

1. Turn on the power to the DXM Controller.
2. Set DIP switch 4 to the ON position.
3. Press and hold the processor button until processor board LED flashes.
4. Set DIP switch 4 to the OFF position.
5. Cycle power to the DXM Controller.
6. Reload the configuration file before resuming normal operation.

10.4 DXM700 Documentation

For more information about the DXM700 family of products, please see additional documentation and videos on the Banner website: www.bannerengineering.com.

- DXM Wireless Controller Sell Sheet, p/n 194063
- DXM700-B1 Wireless Controller Datasheet, p/n 207893
- DXM700-Bx Wireless Controller Instruction Manual, p/n 207894
- DXM ScriptBasic Instruction Manual, p/n 191745
- DXM Controller Configuration Quick Start, p/n 191247
- DXM Configuration Tool software (p/n b_4447978)
- DXM Configuration Tool Instruction Manual, p/n 158447

---

3 While the file download is in process over a USB connection, do not use other applications on the PC. After the DXM700 reboots for a firmware update, the USB port may be unresponsive. Clear the connection by disconnecting the USB cable and restarting the DXM Configuration Software software.
10.5 DXM700 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 DXM700, 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.

10.5.1 Firmware Updates

The DXM700 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 DXM700. 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.

10.5.2 Website Information

The Banner website is the main method of disseminating DXM700 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

10.5.3 Feature Requests

Our customer is our most valuable resource to improve our DXM700. If you have suggestions for improvements to the DXM700 or configuration tools, please contact Banner Engineering Corp.

10.5.4 Potential DXM Issues

Potential issues with the DXM700 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.

10.5.5 DXM Security

The DXM700 was designed to collect local wireless sensor data, local sensor data, provide simple control, and send the data to the cloud.

The DXM700 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).

10.6 Contact Us

Banner Engineering Corp. headquarters is located at:

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

Banner Engineering de México, S. de R.L. de C.V.
David Alfaro Siqueiros 103 Piso 2 Valle oriente
San Pedro Garza García Nuevo León, C. P. 66269
81 8363.2714

10.8 FCC Certification, 900 MHz, 1 Watt Radios

The DX80 Module complies with Part 15 of the FCC rules and regulations.

FCC ID: UE3RM1809 This device complies with Part 15 of the FCC Rules. 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.

10.8 FCC Notices

IMPORTANT: The radio modules have been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Changes or modifications not expressly approved by the manufacturer could void the user's authority to operate the equipment.

IMPORTANT: The radio modules have been certified for fixed base station and mobile applications. If modules will be used for portable applications, the device must undergo SAR testing.

IMPORTANT: If integrated into another product, the FCC ID label must be visible through a window on the final device or it must be visible when an access panel, door, or cover is easily removed. If not, a second label must be placed on the outside of the final device that contains the following text: Contains FCC ID: UE3RM1809.

10.8 Note

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna,
- Increase the separation between the equipment and receiving module,
- Connect the equipment into an outlet on a circuit different from that to which the receiving module is connected, and/or
- Consult the dealer or an experienced radio/TV technician for help.

Antenna WARNING: This device has been tested with Reverse Polarity SMA connectors with the antennas listed in Table 2 (p. 60). When integrated into OEM products, fixed antennas require installation preventing end-users from replacing them with non-approved antennas. Antennas not listed in the tables must be tested to comply with FCC Section 15.203 (unique antenna connectors) and Section 15.247 (emissions).

10.8 FCC Approved Antennas

WARNING: This equipment is approved only for mobile and base station transmitting devices. Antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be collocated or operating in conjunction with any other antenna or transmitter.

DX80 Module may be used only with Approved Antennas that have been tested with this module.

Table 2: Certified Antennas for 900 MHz 1 Watt

<table>
<thead>
<tr>
<th>Model Number</th>
<th>Antenna Type</th>
<th>Maximum Gain</th>
<th>Minimum Required Cable/Connector Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integral Antenna</td>
<td>Unity gain</td>
<td>0</td>
</tr>
<tr>
<td>BWA-901-x</td>
<td>Omni, 1/4 wave dipole</td>
<td>≤2 dBi</td>
<td>0</td>
</tr>
<tr>
<td>Model Number</td>
<td>Antenna Type</td>
<td>Maximum Gain</td>
<td>Minimum Required Cable/Connector Loss</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
<td>--------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>BWA-902-C</td>
<td>Omni, 1/2 wave dipole, Swivel</td>
<td>≤2 dBi</td>
<td>0</td>
</tr>
<tr>
<td>BWA-906-A</td>
<td>Omni Wideband, Fiberglass Radome</td>
<td>≤8.2 dBi</td>
<td>2.2 dB</td>
</tr>
<tr>
<td>BWA-905-B</td>
<td>Omni Base Whip</td>
<td>≤7.2 dBi</td>
<td>1.2 dB</td>
</tr>
<tr>
<td>BWA-9Y10-A</td>
<td>Yagi</td>
<td>≤10 dBi</td>
<td>4 dB</td>
</tr>
</tbody>
</table>

### 10.9 FCC Certification, 2.4GHz

The DX80 Module complies with Part 15 of the FCC rules and regulations.

FCC ID: UE300DX80-2400 This device complies with Part 15 of the FCC Rules. 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.

### 10.9 FCC Notices

**IMPORTANT:** The DX80 Modules have been certified by the FCC for use with other products without any further certification (as per FCC section 2.1091). Changes or modifications not expressly approved by the manufacturer could void the user’s authority to operate the equipment.

**IMPORTANT:** The DX80 Modules have been certified for fixed base station and mobile applications. If modules will be used for portable applications, the device must undergo SAR testing.

**IMPORTANT:** If integrated into another product, the FCC ID label must be visible through a window on the final device or it must be visible when an access panel, door, or cover is easily removed. If not, a second label must be placed on the outside of the final device that contains the following text: Contains FCC ID: UE300DX80-2400.

### 10.9 Note

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna,
- Increase the separation between the equipment and receiving module,
- Connect the equipment into an outlet on a circuit different from that to which the receiving module is connected, and/or
- Consult the dealer or an experienced radio/TV technician for help.

Antenna Warning: This device has been tested with Reverse Polarity SMA connectors with the antennas listed in Table 3 (p. 61). When integrated into OEM products, fixed antennas require installation preventing end-users from replacing them with non-approved antennas. Antennas not listed in the tables must be tested to comply with FCC Section 15.203 (unique antenna connectors) and Section 15.247 (emissions).

### 10.9 FCC Approved Antennas

**WARNING:** This equipment is approved only for mobile and base station transmitting devices. Antenna(s) used for this transmitter must be installed to provide a separation distance of at least 20 cm from all persons and must not be collocated or operating in conjunction with any other antenna or transmitter.

DX80 Module may be used only with Approved Antennas that have been tested with this module.

**Table 3: Certified Antennas for 2.4 GHz**

<table>
<thead>
<tr>
<th>Model</th>
<th>Antenna Type</th>
<th>Maximum Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integral antenna</td>
<td></td>
<td>Unity gain</td>
</tr>
<tr>
<td>BWA-202-C</td>
<td>Omni, 1/2 wave dipole, Swivel</td>
<td>≤2 dBi</td>
</tr>
<tr>
<td>BWA-205-C</td>
<td>Omni, Collinear, Swivel</td>
<td>≤5 dBi</td>
</tr>
</tbody>
</table>
10.10 Warnings

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

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

10.11 Banner Engineering Corp. Limited Warranty

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

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

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

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

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

A
authentication 46, 47

C
cell modem
activate 3G GSM 33
activate Verizon 4G LTE 33
configure DXM 33
install 32
purchase 32
create
holiday event 46
one-time event 45
weekly event 45

D
DXM
configure for cell 33

H
holiday event 46

M
Modbus registers 8

O
one-time event 45

R
registers 8

S
scheduler 45, 46

W
webserver authentication 46, 47
weekly event 45