XLR PRO

Long Range Radio Solution

User Guide

XLR PRO User Guide

(Part number 90002202 A)

Revision	Date	Description
A	September 2014	Initial release.

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International Inc.. Digi provides this document "as is," without warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of fitness or merchantability for a particular purpose. Digi may make improvements and/or changes in this manual or in the product(s) and/or the program(s) described in this manual at any time.

Trademarks and copyright

Digi, Digi International Inc., and the Digi logo are trademarks or registered trademarks in the United States and other countries worldwide. All other trademarks mentioned in this document are the property of their respective owners.

Customer support

If you need assistance, contact technical support:

Telephone (8:00 am - 5:00 pm—U.S. Central Time):

866.765.9885 toll-free U.S.A. & Canada

801.765.9885 Worldwide

Fax: 952.912.4952

Online: www.digi.com/support/eservice

Mail:

Digi International Inc.

11001 Bren Road East

Minnetonka, MN 55343

USA

Contents

XLR PRO networking methods

XLR PRO AT command reference tables

XLR PRO API operation

XLR PRO advanced application features

Configuring the XLR PRO using XCTU

Configuring XLR PRO using the web configuration interface

Configuring XLR PRO using Device Cloud

Troubleshooting

Safety notices and certifications

XLR PRO approved antennas

About the XLR PRO

XLR PRO is a high performance, industrial grade long-range radio solution that offers serial, Ethernet socket, and Ethernet bridging connectivity to ensure reliable wireless data communications over long distances.

Packaged in a sturdy, ruggedized enclosure and using Chirp Spread Spectrum technology (patent pending) to maximize range and significantly increase immunity to interference, the XLR PRO 900 MHz radio modem can connect a variety of devices across many industrial applications.

Flexible configuration and management options allow you to quickly set up and deploy one or more XLR PROs, as well as apply firmware updates, get device status information, and more.

Technical specifications

The following table summarizes XLR PRO technical specifications.

XLR PRO	Specifications	
General		
Dimensions	18 x 13 x 3.8 cm (7.1 x 5 x 1.5 inches)	
Weight	0.68kg (1.5 lbs)	
Ethernet protocols	UDP/TCP, DHCP client	
Ethernet physical layer	10/100BASET-T/TX	
RoHS	Compliant	
Performance		
Frequency range	ISM 902 to 928 MHz	
RF data rate	9.4 kbps to 3.2 Mbps	
Maximum transmit power (software selectable)	+30 dBm (1 watt)	
Rural range line of sight	1.2 Mbps	up to 150 miles[1]
Urban range line of sight	9.4 kbps	up to 25 miles[2]
	141 kbps	up to 8 miles[2]
	591 kbps	up to 3.6 miles
	3.2 Mbps	up to 1 mile[2]
Receiver sensitivity	9.4 kbps	-120 dBm
	141 kbps	-112 dBm
	3.2 Mbps	-98 dBm
Receiver selectivity	141 kbps	70 dB (below 908 MHz, above 922 MHz) 40 dB (908 MHz to 922 MHz)
Interface data rate (software selectable)	460.8 kbps (TCP/UDP socket)	
	921.6 kbps (serial RS-485)	
	460.8 kbps (serial RS-232)	
Networking and security		
Modulation	Chirp Spread Spectrum (patent pending)	
Supported network topologies	Point-to-point/point-to-multi-point	

XLR PRO	Specifications	
Encryption	128-bit AES	
Power requirements		
Supply voltage	9VDC to 26VDC	
Receive current	@ 9 VDC	300 mA
	@ 12 VDC	230 mA
	@ 26 VDC	120 mA
Transmit current	@ 9 VDC	950 mA
	@ 12 VDC	840 mA
	@ 26 VDC	400 mA
Environmental		
Operating temperature	-40° C to 70° C	
Regulatory approvals		
Emissions/immunity	FCC Part 15B	
Hazardous locations	Class I, Division 2, Groups A, B, C, and D	
Connectors		
Antenna	RPTNC	
Power supply	Phoenix	
Ethernet	RJ45	
Serial	RJ45	
Configuration port	Mini USB (currently not implemented)	

1. Based on 150-mile range results. Other data rates scale based on sensitivity levels. Results will vary based on noise levels and line or sight quality.
2. Estimated based on 3.6-mile range results

Data rates and sensitivity

The following table lists XLR PRO available data rates along with the corresponding receiver sensitivity.

RF data rate setting (ATBR)	Data rate	Receiver sensitivity (dBm, 25°C)
0	9.4 kbps	-120
1	28 kbps	-118
2	66 kbps	-116
3	141 kbps	-112
4	291 kbps	-109
5	591 kbps	-106
6	1.2 Mbps	-103
7	2.4 Mbps	-100
8	3.2 Mbps	-98

Power supply

The XLR PRO must be powered by a UL-listed power supply rated between 9 and 26V DC. Refer to the following table for the required input current settings.

Input voltage DC	Minimum current rating
9 to 15	3 Amps
15 to 20	2 Amps
21 to 26	1.5 Amps

Operational modes

XLR PRO offers four operational modes

- Serial (RS-232/RS-485)
- Ethernet (IP socket)
- Ethernet RF Bridging

Serial

When operating in serial mode, the front panel serial port provides connectivity to the XLR PRO via RS-232 or RS-485/422.

Ethernet (IP socket)

In IP socket mode, the XLR PRO is able to transmit and receive serial data via a TCP or UDP connection from either of the front panel Ethernet ports to the XLR PRO.

Ethernet RF Bridging

In Ethernet RF bridging mode, the XLR PRO functions as an Ethernet cable replacement, supporting point-to-multi-point transmission for a maximum of 16 XLR PROs. By default, bridging mode is disabled. Serial data from serial or IP socket mode operates concurrently with Ethernet RF Bridging. If serial and Ethernet traffic are sent at the same time, latency will be encountered.

Hardware interfaces

The XLR PRO offers the following hardware interfaces.

Serial (RJ45) port

The XLR PRO serial port (RJ45) supports RS-232 or RS-485/422 protocols.

USB (mini USB)

At present, the mini USB port is for internal manufacturing use only.

Ethernet

The Ethernet interface is 10/100 Base-T with a two-port managed Ethernet switch. The XLR PRO does not support Power over Ethernet (PoE) and must be externally powered through the DC power jack.

LEDs

The following table describes XLR PRO LED behavior.

Action	LED	Status	Description
Startup	Power	Solid	Power LED glows solid red.
	Link margin indicators (RSSI)	Solid	All of the link margin indicator LEDs show green for one second.
	Serial Data In (green)	Solid	On initial power on, if the green serial data in LED shows solid for three seconds, the XLR PRO is in serial mode.
	Serial Data Out (yellow)	Solid	On initial power on, if the yellow serial data out LED shows solid for three seconds, the XLR PRO is in socket mode.
Data transmissions	Power	Solid	While power is on, the power LED shows solid red.
	Link margin indicators (RSSI)	Solid	The link margin indicator LEDs illuminate for four seconds to show the signal strength of the last valid RF packet received. 3 LEDs = Very strong signal (> 30 dB fade margin) 2 LEDs = Strong signal (>20 dB fade margin) 1 LED = Moderate signal (>10 dB fade margin) 0 LED = Weak signal (<10 dB fade margin)
	Serial Data In (green)	Flashing	Serial data in LED shows flashing when serial data is being transmitted.
	Serial Data Out (yellow)	Flashing	Serial data out LED shows flashing when serial data is being received.

Reset button

You can use the Reset button to reset the XLR PRO and to restore factory default settings.

To reset the XLR PRO:

1. Hold down the **Reset** button for up to five seconds. The serial data in and serial data out LEDs flash three times to indicate that five seconds have passed.
2. Release the **Reset** button. The XLR PRO resets.

To restore factory default settings:

1. Hold down the **Reset** button for about eight seconds. While you're holding down the reset button, the serial data in and serial data out LEDs flash three times to indicate that five seconds have passed. Continue to hold down the **Reset** button until the serial data in and serial data out LEDs flash six times.
2. Release the **Reset** button. The XLR PRO is restored to factory default settings.

Antenna port

The antenna port is a 50 ohm RF signal connector for connecting to an external antenna. The connector type is RPTNC (Reverse Polarity TNC) female. The connector has threads on the outside of a barrel and a male center conductor.

XLR PRO documentation

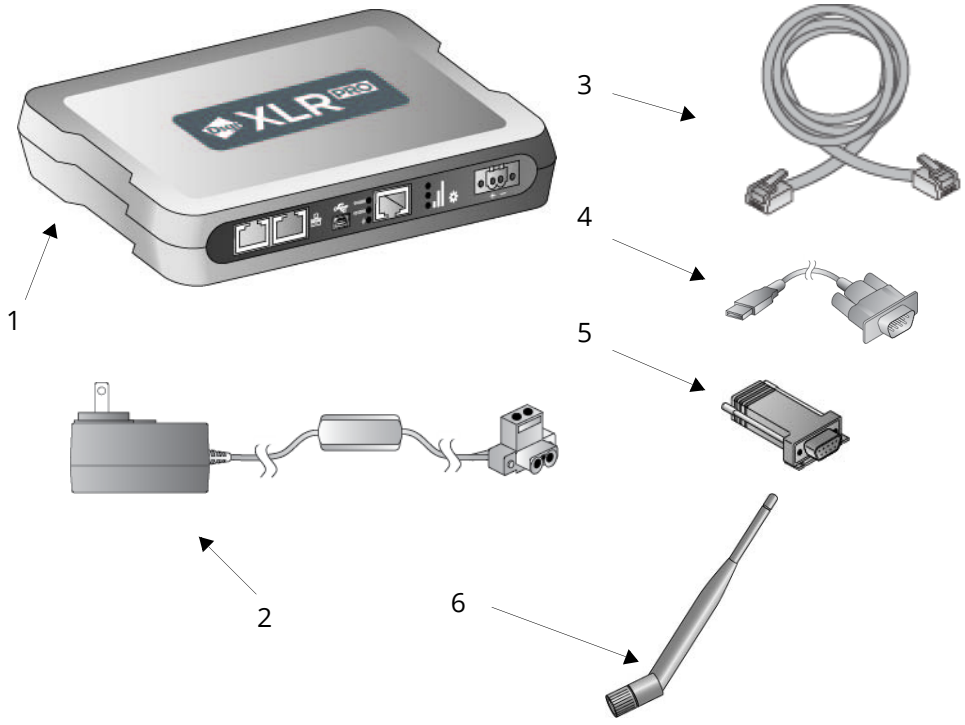
XLR PRO documentation includes the following publications:

Title	Part Number	Description
XLR PRO Quick Start Guide	90002204	Provides a brief summary of the XLR PRO kit.
XLR PRO Getting Started Guide	90002203	Provides step-by-step instructions for setting up a pair of XLR PROs to test over-the-air communications between the radios.
XLR PRO User Guide	90002202	Provides complete information on all XLR PRO features; describes how to configure XLR PROs using XCTU, the Web configuration interface, and Device Cloud; provides reference information on all supported AT commands.

XLR PRO hardware

XLR PRO kit contents

The following figure shows the XLR PRO accessories kit.

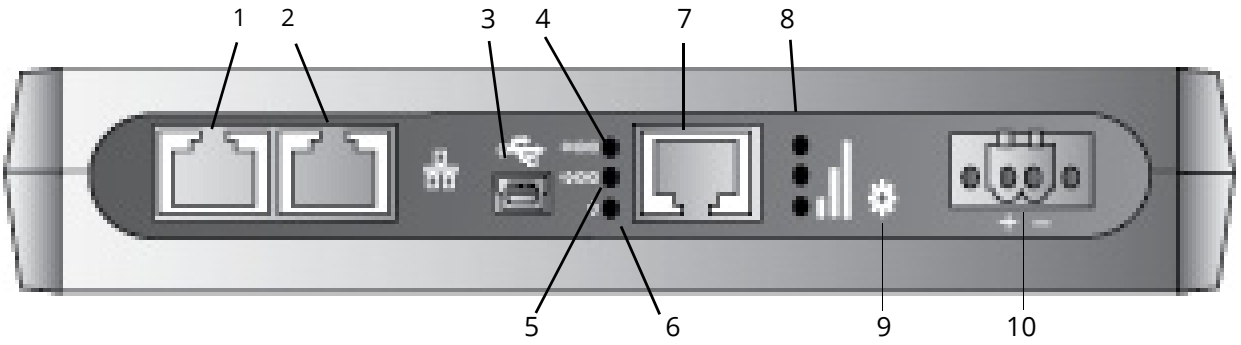


Item	Description
1	XLR PRO
2	Power supply
3	Ethernet cable
4	USB to RS-232 serial converter

Item	Description
5	RJ45/DB9F adapter
6	Antenna

Front panel

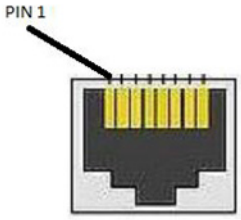
The following figure shows XLR PRO front panel connectors and LEDs.



WARNING! Use the serial port for serial connections only. Do not connect the serial RJ45 port to any PoE (power over Ethernet) device. Doing so can permanently damage the XLR PRO or PoE device and void the XLR PRO warranty.

Item	Description
1	Ethernet port 1
2	Ethernet port 2
3	Mini USB
4	Serial data out LED
5	Serial data in LED
6	Power LED
7	Serial port
8	Link margin indicator LEDs
9	Reset button
10	DC power jack

RJ45 serial port pinout



Pin	RS-232	RS-485 (4-wire)	RS-485 (2-wire)
1	RXD	TX+	TX/RX+
2	CTS	TX-	TX/RX-
3	TXD	RX+	Unused
4	GND	GND	GND
5	GND	GND	GND
6	RTS	RX-	Unused
7	DSR/DCD	Unused	Unused
8	DTR	Unused	Unused

WARNING! Use the serial port for serial connections only. Do not connect the serial RJ45 port to any PoE (power over Ethernet) device. Doing so could permanently damage the XLR PRO or PoE device and void the XLR PRO warranty.

Hardware setup

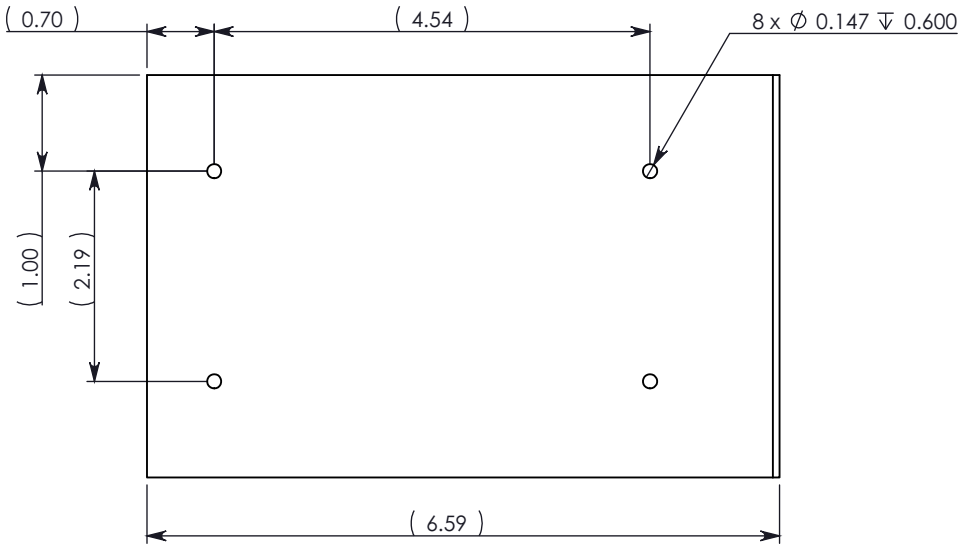
Connect the hardware

The following figure shows how to connect the XLR PRO cables and antenna.

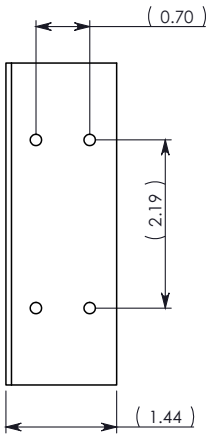
Mount the XLR PRO

CAUTION! You must install the XLR PRO in a Restricted Access Location. In addition, you must restrict access to personnel who have been instructed on potential hazards, as well as physically restrict access using a tool, or lock and key, or other means controlled by a responsible authority.

The XLR PRO provides mounting holes in the bottom of the unit by which you can mount the unit directly on a wall or attach mounting brackets. The following illustration shows XLR PRO mounting-hole locations and dimensions.



The following illustration shows a suggested mounting bracket for the XLR PRO.



Mounting guidelines

Follow these general guidelines when mounting the XLR PRO:

- Use the pre-drilled mounting holes located on the bottom of the XLR PRO unit to attach the XLR PRO to the wall of an enclosure or DIN Rail bracket. Do not alter or move the mounting holes.
- To attach brackets to the XLR PRO, use four (4) 6-32x3/8” screws. Do not use screws that are longer than 3/8”.

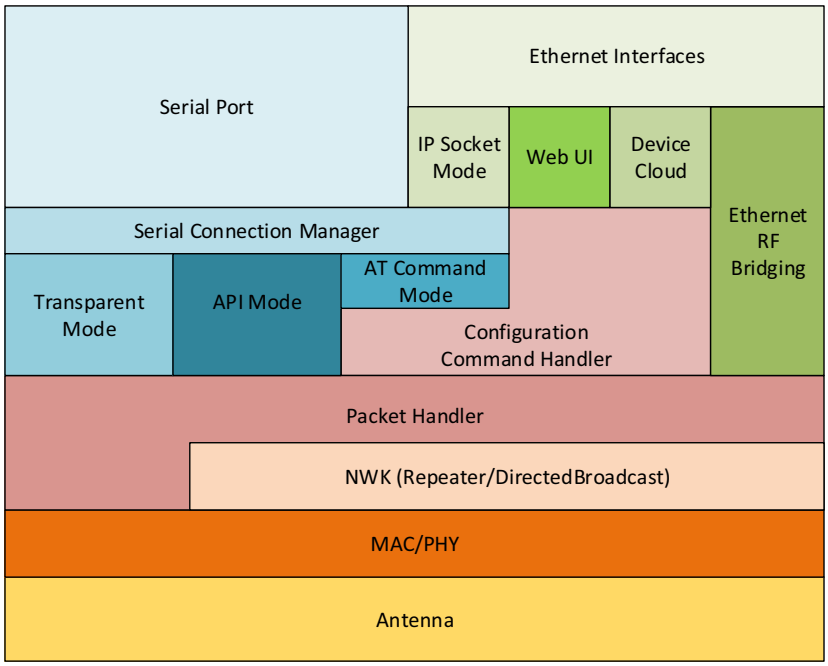
Antenna placement

- Mount the XLR PRO antenna vertically—that is, pointed directly up or down.
- If the XLR PRO is mounted within a metal enclosure, use an antenna external to the enclosure connected to the XLR PRO using a a 50 ohm coaxial cable, suitable for 900MHz UHF radio transmission.

XLR PRO operation

XLR PRO operational design

The XLR PRO® uses a multi-layered firmware base to order the flow of data, and the flow of data is dependent on the hardware and software configuration chosen by the user. The configuration block diagram below shows the host serial interface as the physical starting point and the antenna as the physical endpoint for transferred data. As long as a block is able to touch another block, the two interfaces can interact. For example, if the XLR PRO is using API mode, Transparent Mode is not available. See below:



Ethernet RF bridging

In Ethernet RF bridging mode, the XLR PRO functions as an Ethernet cable replacement, supporting point-to-multipoint transmissions for up to 16 XLR PROs. It is configured by these parameters:

- **BE** - Bridge Enable. This parameter must be set to 1 to enable bridging, but it is disabled by default.
- **BA** – Destination RF MAC address for Ethernet bridging. Default value is 0xFFFF which is the broadcast address. If pairing XLR PRO devices is desired, then this should be set to the RF MAC address of the opposing XLR PRO. This can be identified by querying the SH and SL parameters on the opposite XLR PRO (example: BA=0013A20012345678).

With bridging enabled, the XLR PRO radios on the RF network should be treated as if they were a single Ethernet cable. A qualified network administrator should evaluate the radio deployment if multiple XLR PRO radios will be used on the same LAN or if bridging multiple large networks together.

Serial mode selection

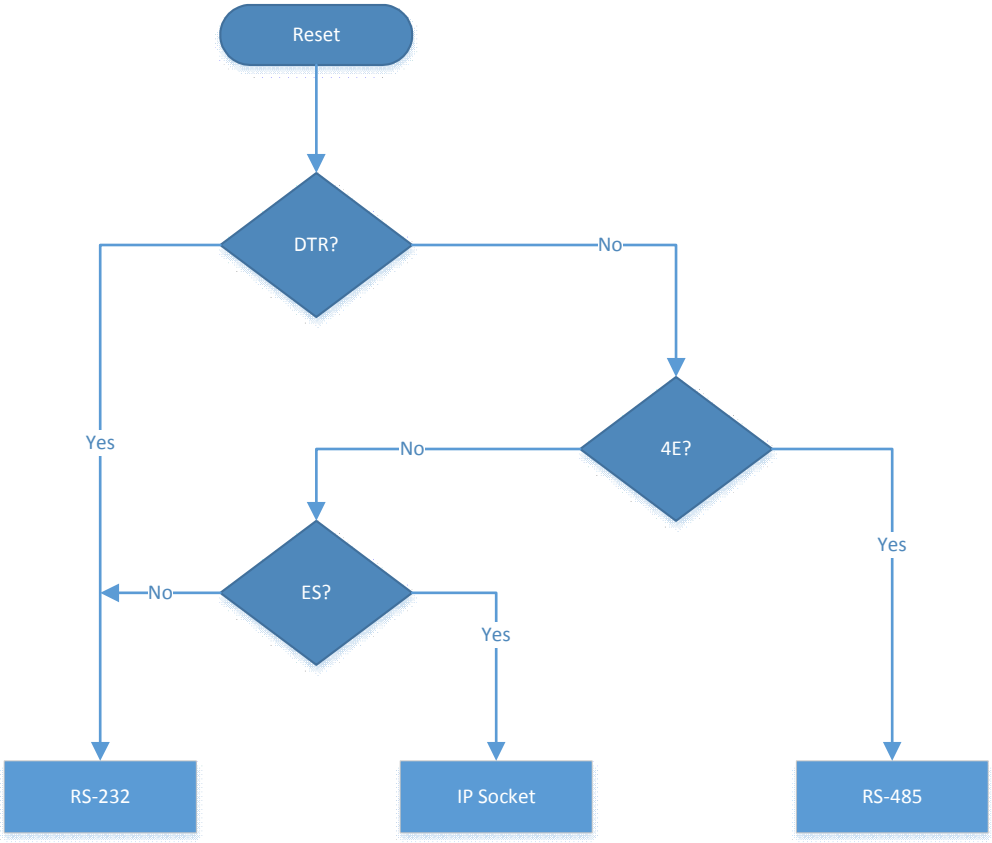
The XLR PRO provides three serial modes:

- RS-232
- RS-485/422
- IP socket

At any given time, only one serial mode can be can be selected based on the detection of cable connections and configuration options. The following factors determine serial mode selection:

- **DTR:** XLR PRO detects DTR on the serial port.
- **4E (Enable RS-485 mode):** RS-485 configuration option is enabled or disabled.
- **ES (Enable IP socket mode):** IP Socket configuration option is enabled or disabled.

The following flow chart indicates how the three factors (DTR, 4E, and ES) affect which mode of operation is active:



By default, 4E (Enable RS-485 mode) is 0 (disabled) and ES (Enable IP socket mode) is 1 (enabled). In this default case, RS-232 mode is selected if DTR is present. Otherwise, IP socket mode is selected. The 4E and ES configuration options allow other modes to be selected, independent of DTR, which may not be present on a serial port connection.

To automatically detect a serial connection requires an active DTR signal. If your serial cable or end device does not include an active DTR signal, disable socket mode (ES = 0) using XCTU, the web configuration, or Device Cloud.

Serial mode

When serial mode is the primary interface, the Serial Data In (green) LED will light for about three seconds. Serial mode can be either RS-232 or RS-485/422, depending on serial mode selection. The following parameters must be configured to match the host device, regardless of whether RS-232 or RS-485 is selected:

- **BD** - Baud rate (See AT command table for limits)
- **NB** - Parity (None, Even, or Odd)
- **SB** - Stop bits (1 or 2)

RS-232

RS-232 connections support hardware flow control using CTS and RTS and requires matching parameters on the XLR PRO and the host device. This includes the following:

- **D6** - RTS flow control. If enabled, then XLR PRO will not output data unless RTS is asserted. The host device should not de-assert RTS for long periods of time to avoid filling the serial transmit buffer. If an RF data packet is received, and the serial transmit buffer does not have enough space for all of the data bytes, the entire RF data packet will be discarded.
- **D7** - CTS flow control. If enabled, then XLR PRO will not assert CTS low unless it can handle more data from the host.
- **FT** - Flow control threshold. If CTS flow control is enabled (with D7 parameter), the XLR PRO deasserts CTS when the serial receive buffer reaches the threshold defined by the FT parameter. Once CTS is de-asserted, it will not be asserted again until the receive buffer has 17 bytes less than the threshold defined by FT. By default, FT is 65 bytes less than the maximum space available for receive data.

RS-485/422

An RS-485 connection requires that 4E=1 and it also requires matching parameters on the XLR PRO and the host device. This includes the following:

- **4E** - Enable RS-485/422. If 4E is set to 0, then the XLR PRO will use RS-232. This parameter needs to be set to 1 in order to use RS-485/422 on the serial port.
- **4D** - Full duplex (4-wire) or half duplex (2-wire) operation. Default value of 0 selects half duplex operation and 1 selects full duplex.
- **4T** - RS-485/422 termination. Enable or disable line termination on the RS-485/422 interface. The default value of 0 indicates that there is no line termination on the XLR PRO. If 4T is set to 1, then a 120 Ohm termination resistor will be present on the RS-485/422 connection. This parameter will have no effect on the XLR PRO if it is configured for RS-232.

IP socket mode

IP socket mode provides serial communication for a single TCP or UDP port on the XLR PRO (multiple simultaneous connections are not supported). This would normally happen over the Ethernet connection, but it may also occur over the bridge if Ethernet RF bridging is enabled and another XLR PRO (which also has Ethernet RF bridging enabled) provides the Ethernet connection to an IP host. With the factory default settings, the XLR PRO listens on port 9750 for incoming TCP traffic. A telnet session can be initiated to the XLR PRO's IP address as a simple IP socket connection.

The same operations that can occur in serial mode can also occur in IP socket mode. Those operations are based on the payload of the IP frames. In other words, serial data coming to and from the XLR PRO is equivalent to the payload of the IP socket mode data.

Controlling parameters

IP socket mode is configured by these parameters:

- **IB** - IP socket mode baud rate. This is set to the maximum rate of 460800bps by default, but it may be set to a lower rate for throttling, if desired.
- **IP** - IP protocol. Default value of 1 selects TCP and 0 selects UDP. This parameter must match the protocol used by the IP host.
- **C0** - TCP or UDP port on which the XLR PRO will listen. The IP host must send data to this port in order for the XLR PRO to accept the incoming data. If configuring the XLR PRO using XCTU, the port number is displayed in hexadecimal.
- **DY** - Destination IP port. (See DX above.)

- **DX** - Destination IP address. This tells the XLR PRO where to send data if it initiates the conversation. If operating in TCP mode and a TCP connection doesn't currently exist, then the XLR PRO will attempt to make a connection to this IP address (and the IP port given by DY) to send the data to the selected IP host and port. However, if a TCP connection already exists, then the data will be sent to that connection, ignoring the DX and DY parameters. If operating in UDP mode, this rule changes slightly because it is a connectionless protocol. If the first IP socket mode data comes from the XLR PRO, then DX/DY will be used. If not, then all UDP data will be sent to the IP address and port from which the original data arrived.
- **TM** - TCP client connection timeout. A client connection is one which was initiated by the XLR PRO. This parameter tells how many seconds a TCP client connection will remain connected when no data is being sent or received on the connection.
- **TS** - TCP server connection timeout. A server connection is one which was initiated by an external IP host. This parameter tells how many seconds a TCP server connection will remain connected when no data is being sent or received on the connection.

In addition to the above IP socket mode parameters, the MY parameter is also used in IP socket mode:

- **MY** - IP address of the XLR PRO. By default, this address is learned from a DHCP server, but it may be set to any value if static mode is used (MA=1).

Operational description

IP socket mode may start up in the following cases:

1. It may start at reset, based on the mode selection rules previously described.
2. It may start when parameters that affect the mode are changed and applied.
3. It may start when the RS-232 cable is unplugged.

When IP socket mode is the primary interface, the Serial Data Out (yellow) LED will light for about three seconds.

Upon starting or restarting the XLR PRO, either a TCP or a UDP listener is set up depending on the IP parameter. If UDP data is received or if a TCP connection gets established before the XLR PRO attempts to send data, then the DX and DY parameters are unused. In this case, the XLR PRO takes the role of a TCP or UDP server. But if the XLR PRO has data to send before an IP host sends data to the XLR PRO, then DX and DY determine the destination of that data until the TCP connection times out or until IP socket mode is restarted, whichever comes first.

Serial communications

Whether the XLR PRO is configured for RS-232, RS-485/422, or IP socket mode, the XLR PRO will handle the traffic as serial data. All serial traffic will be handled by the XLR PRO identically regardless of which interface is being used.

XLR PRO serial buffers

Serial receive buffer

When serial data enters the XLR PRO, the data is stored in the serial receive buffer until it can be processed. Under certain conditions, the XLR PRO may not be able to process data in the serial receive buffer immediately. If large amounts of serial data are sent to the XLR PRO such that the serial receive buffer would overflow, then the new data will be discarded. If using RS-232, this can be

avoided by utilizing hardware flow control. Software flow control can be utilized regardless of which serial interface is being used.

Serial transmit buffer

When serial RF data is received, the data is moved into the serial transmit buffer and sent out of the active serial interface of the XLR PRO. If the serial transmit buffer becomes full and system buffers are also full, then the entire RF data packet is dropped. Whenever data is received faster than it can be processed and transmitted out the serial port, there is a potential of dropping data.

XLR PRO serial interface protocols

The XLR PRO supports both transparent and API (Application Programming Interface) serial interfaces.

Transparent operation

When operating in transparent mode, the XLR PRO acts as a serial line replacement. All serial data received through the serial interface is queued up for RF transmission. When RF data is received, the data is sent out through the active serial interface. The XLR PRO configuration parameters are configured using the AT command mode interface.

Data is buffered in the serial receive buffer until one of the following causes the data to be packetized and transmitted:

- No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, no packetization will occur and characters will be transmitted immediately.
- The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the serial receive buffer before the sequence is transmitted.
- The maximum number of characters that will fit in an RF packet is received. See the NP parameter.

API operation

API operation is an alternative to transparent operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the XLR PRO. When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the XLR PRO.

Transmit Data Frames (received through the serial port) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the serial port) include:

- RF-received data frame
- Command response
- Event notifications such as reset, and so on.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the XLR PRO that contain address and payload information instead of using command mode to modify addresses. The XLR PRO will send data frames to the application containing status packets; as well as source, and payload information from received data packets.

The API operation option facilitates many operations such as the examples cited below:

- Transmitting data to multiple destinations without entering Command Mode
- Receive success/failure status of each transmitted RF packet
- Identify the source address of each received packet

Comparison of transparent and API operation

The following table compares the advantages of transparent and API modes of operation:

Transparent operation features	
Simple Interface	All received serial data is transmitted unless the XLR PRO is in command mode. The XLR PRO will act as a serial cable replacement.
Easy to support	It is easier for an application to support transparent operation and command mode. Integrates easily with existing serial equipment and will work “out of the box” in many applications.
API operation features	
Easy to manage data transmissions to multiple destinations	Transmitting RF data to multiple remotes only requires changing the address in the API frame. This process is much faster than in transparent operation where the application must enter AT command mode, change the address, exit command mode, and then transmit data. Each API transmission can return a transmit status frame indicating the success or reason for failure.
Received data frames indicate the sender's address	All received RF data API frames indicate the source address. This makes receiving data from multiple sources easy to identify and manage.
Advanced networking diagnostics	This allows for diagnosis of RF network problems that may be a result of poor RF performance or identifying nodes that have lost network connectivity.
Remote configuration	Set/read configuration commands can be sent to remote devices to configure them as needed using the API. This also includes over-the-air firmware updates.

As a general rule of thumb, API mode is recommended when a device:

- Sends RF data to multiple destinations.
- Sends remote configuration commands to manage devices in the network.
- Receives RF data packets from multiple devices, and the application needs to know which device sent which packet.

If the above conditions do not apply, then transparent operation may be preferred due to the simplicity of operation. It is acceptable to use a mixture of devices running API mode and transparent mode in a network.

AT Command mode

To modify or read radio parameters, the XLR PRO must first enter into Command Mode - a state in which incoming serial characters are interpreted as commands. When the XLR PRO is configured for API mode, AT Commands can be issued locally or remotely using the appropriate API frames. For

more information on configuring the XLR PRO using API frames, refer to the API section of this manual. While in API mode, AT command mode can still be entered using the command sequence listed below.

To Enter AT Command Mode:

1. Send the 3-character command sequence +++ (three plus signs).
2. Observe guard times before and after the command characters. [Refer to the “Default AT Command Mode Sequence” below.]

Default AT Command Mode Sequence (for transition to Command Mode):

- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]
- Input three plus characters (“+++”) within one second [CC (Command Sequence Character) parameter = 0x2B.]
- No characters sent for one second [GT (Guard Times) parameter = 0x3E8]

Once the AT command mode sequence has been issued, the XLR PRO sends an "OK\r" out of the active serial interface. The "OK\r" characters can be delayed if the XLR PRO has not finished transmitting received serial data.

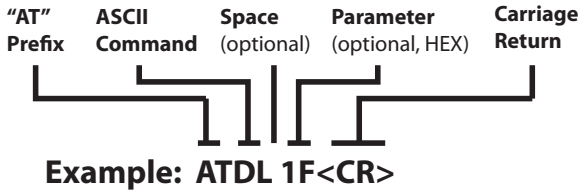
When command mode has been entered, the command mode timer is started (CT command), and the XLR PRO is able to receive AT commands on the active serial interface.

All of the parameter values in the sequence can be modified to reflect user preferences.

Note Failure to enter AT Command Mode is most commonly due to baud rate mismatch. By default, the BD (Baud Rate) parameter = 3 (9600 bps).

To Send AT Commands:

Send AT commands and parameters using the syntax shown below.



To read a parameter value stored in the XLR PRO’s register, omit the parameter field.

The preceding example would change the Destination Address (Low) to “0x1F”. To store the new value to non-volatile (long term) memory, send the WR (Write) command. This allows modified parameter values to persist in the module’s registry after a reset. Otherwise, parameters are restored to previously saved values after a reset.

Command response

When a command is sent to the XLR PRO, it will parse and execute the command. Upon successful execution of a command, an “OK” message is returned. If execution of a command results in an error, an “ERROR” message is returned.

Applying command changes

Any changes made to the configuration command registers through AT commands will not take effect until the changes are applied. For example, sending the BD command to change the baud rate will not change the actual baud rate until changes are applied. Changes can be applied in one of the following ways:

- The AC (Apply Changes) command is issued.
- AT command mode is exited.

To exit AT command mode:

1 Send the ATCN (Exit Command Mode) command (followed by a carriage return).

or

2 If no valid AT Commands are received within the time specified by CT (Command Mode Timeout) Command, the XLR PRO automatically returns to Idle Mode.

For an example of programming the XLR PRO using AT Commands and descriptions of each configurable parameter, please see the AT Command Reference Table chapter.

Serial recovery

In the rare case that the XLR PRO has been configured with unknown serial settings, the radio may be recovered to a known good state by holding a serial break at reset time. Issuing a serial break for 10 seconds temporarily forces a default configuration on the XLR PRO. This will allow access to the serial port using RS-232 at 9600 baud and it will bring up the radio in command mode. Appropriate AT commands can then be sent to the XLR PRO to default the radio (ATRE followed by ATWR), query the current parameters, or change radio parameters so that the XLR PRO can be used normally.

Firmware updates

The firmware on the XLR PRO may be updated using three different methods:

- XCTU
- Web UI
- Device Cloud

The same XLR PRO firmware image is selected for all three firmware update methods. The detailed steps for loading this image are given in the respective sections for Configuring with XCTU, configuring with WEB configuration interface, and configuring with Device Cloud, respectively.

Although there are three different interfaces to update the XLR PRO firmware, the process is the same:

1. Firmware Image is loaded to the XLR PRO with validation on each block as it is being loaded into non-volatile memory.
2. Entire Image is validated in non-volatile memory.
3. XLR PRO resets and places the firmware image into the application area and validates it. Note that this process takes approximately 45 seconds after reset.

XLR PRO networking methods

This chapter explains XLR PRO networking layers and methods, starting from the simplest and building up to the most complex.

Ethernet RF Bridging

PHY is short for physical layer. The PHY layer is responsible for managing the hardware that modulates and demodulates the RF bits.

MAC is short for media access layer. The MAC layer is responsible for sending and receiving RF frames. Each packet includes a MAC layer data header that has addressing information, as well as packet options. This layer implements packet acknowledgments (ACKs) and packet tracking to eliminate duplicates, and so on.

When a radio is transmitting, it cannot receive packets. There are no beacons or master/slave requirements in the design of the MAC/PHY.

Related parameters: ID, PL, RR, MT

The network ID (ID) can be changed to further keep radios from interfering with each other. This ID is matched after the preamble pattern has been matched, and the MAC header has been received. Networks are defined with a unique network identifier. For XLR PRO radios to communicate, they must be configured with the same network identifier. The ID parameter allows multiple networks to co-exist on the same physical channel.

The power level (PL) sets the transmit (TX) power level. The power level can be reduced from the maximum to reduce current consumption or for testing at short distances. This comes at the expense of reduced radio range.

The RR parameter specifies the number of times a sending radio will attempt to get an ACK from a destination radio when sending a packet. RR is only used for serial traffic, Ethernet RF bridging uses a fixed retry amount.

The MT parameter specifies the number of times that a broadcast packet is repeatedly transmitted. This adds redundancy that improves reliability. MT is only used for serial traffic, Ethernet RF bridging does not utilize retransmissions.

Ethernet RF Bridging

Related parameters: BE, BA

The purpose of Ethernet RF bridging is to act as an Ethernet cable replacement. All Ethernet traffic is handled on the MAC/PHY layer of the Ethernet standard. As a result, the XLR PRO does not have to have a valid IP address on the network for bridging to work.

Ethernet packet handling

If the XLR PRO receives an Ethernet packet whose MAC address does not match the MAC address of the XLR PRO, and if Ethernet bridging is enabled, then the entire Ethernet packet will be encapsulated inside of a radio frame and sent over the air (OTA) to another XLR PRO. No fragmentation is supported; therefore, the whole Ethernet packet is sent in one OTA frame. This can cause an issue at lower data rates. If the OTA data rate (BR) is less than 3 (141kbps), then full size Ethernet frames will not work. However, smaller frames will still work at lower data rates. This is not recommended but might be useful in some scenarios.

In the other direction with bridging enabled, the Ethernet MAC address of an OTA bridging packet is inspected. If the address matches Ethernet MAC address of the XLR PRO, then the packet is handled on board. Otherwise, the frame is forwarded over the Ethernet interface.

The MT and RR parameters do not apply to Ethernet bridging packets, but they do apply for all non-bridging traffic (serial or IP socket mode). Serial data from serial or IP socket mode operates concurrently with Ethernet bridging. If serial and Ethernet traffic are sent at the same time, latency will be encountered.

Precautions

The practical application of an Ethernet bridge is to span two networks that are not otherwise connected. If they are connected by some other path, then a bridge loop allows multiple paths to the same node. When multiple paths to the same node occur, then a broadcast storm can result in a saturated network, resulting in denial of service for legitimate traffic. Typically, this is alleviated by a network switch that support the Spanning Tree Protocol (STP) to detect and prevent such a network loop. STP is not implemented by the XLR PRO.

As a precaution, when using bridging, only connect one of the XLR PRO devices to the same Ethernet network to avoid bridging loops. If you do create multiple paths, and enterprise level switches with STP are being connected to the XLR PRO devices then the connected switch ports will be shutdown.

Be aware that when bridging two networks that if each has its own DHCP server many problems may ensue.

Enabling bridging

Bridging is enabled by setting the BE (bridging enable) parameter to 1. This is disabled by default due to the risk of encountering a bridging loop during initial configuration.

64-bit addresses

Each radio has a unique factory-assigned IEEE 64-bit address. The factory-assigned address can be read with the SH and SL commands. At this time, addresses are of the form: 0x0013A2XXXXXXXXXX. The first 6 digits are the Digi OUI. The broadcast address is 0x000000000000FFFF.

Unicast

To transmit to a specific radio:

- For Ethernet RF Bridging, set BA to the SH:SL of the destination radio. BA is the entire 64-bit address and not broken into two 32-bit values. For example: BA=0x0013A20012345678
- If Ethernet bridging is configured for unicasts, up to three retries will occur while waiting for an ACK. As a result, unicasts are slower, but more reliable than broadcasts.

Broadcast

To transmit to all radios:

- For Ethernet RF Bridging, set BA to 0x000000000000FFFF.
- If more than two XLR PRO devices are participating in a bridge, then BA must be set to 0xFFFF.
- By default, Ethernet bridging uses broadcasts and no retransmissions occur. If packets are sent via TCP, then the TCP protocol will provide the retransmissions as needed to provide for reliability. UDP packets may be lost when BA is set to a broadcast address.

XLR PRO Serial addressing basics

Related parameters: SH, SL, DH, DL, TO

64-bit addresses

Each radio has a unique factory-assigned IEEE 64-bit address. The factory-assigned address can be read with the SH and SL commands. This is the source address that is returned in API mode of the radio that sent a packet. At this time, addresses are of the form: 0x0013A2XXXXXXXXXX. The first 6 digits are the Digi OUI. The broadcast address is 0x000000000000FFFF.

Unicast

To transmit to a specific radio:

- When using transparent mode, set DH:DL to the SH:SL of the destination radio.
- For API mode, set the SH:SL address in the 64-bit destination address field.

Broadcast

To transmit to all radios:

- For transparent mode, set DH:DL to 0x000000000000FFFF; for API mode, set the 64-bit destination address field to 0x000000000000FFFF.
- The scope of the broadcast changes based on the delivery method chosen.

Delivery method

There are two delivery methods supported by this radio:

- Point-to-multipoint. (0x40)
- Repeater (directed broadcast). (0x80)

The TO parameter is the default delivery method used by transparent mode. For API transmissions, the TxOptions API field is used to specify the delivery method. When the TxOptions API field is set to 0, the value in the TO parameter will also be used by API transmissions.

The two delivery modes are described below.

Point-to-point/multipoint (P2MP)

P2MP delivery mode does not use a network header, only the MAC header. All messages are always sent directly to the destination. There is no repeating of the packet by other nodes.

A P2MP unicast is only delivered directly to the destination radio, which must be in range of the sending radio. This radio uses patented technology that allows the destination radio to receive transmissions directed to it, even when there is a large amount of traffic. This works best when broadcast transmissions are kept to a minimum. A P2MP broadcast transmission is repeated MT+1 times by the sending node, but is not repeated by nodes which receive it, so like a unicast transmission, the receiving radio must be in range. All radios that receive a P2MP broadcast transmission will output the data through the active serial interface.

Repeater/directed broadcast

Related parameters: CE, NH, NN, BH

Ethernet RF bridging does not support repeater/directed broadcast. All Ethernet frames are transmitted as point-to-point or point-to-multipoint regardless of what the TO parameter is set to.

Directed broadcast transmissions will be received and repeated by all routers in the network. Because ACKs are not used the originating node will send the broadcast multiple times. By default a broadcast transmission is sent four times. Essentially the extra transmissions become automatic retries without acknowledgments. This will result in all nodes repeating the transmission four times as well. Sending frequent broadcast transmissions can quickly reduce the available network bandwidth and as such should be used sparingly.

The MAC layer is the building block that is used to build repeater capability. Repeater mode is implemented with a network layer header that comes after the MAC layer header in each packet. In this network layer there is additional packet tracking to eliminate duplicate broadcasts. In this delivery method, unicasts and broadcast packets are both sent out as broadcasts that are always repeated. All repeated packets are sent to every radio. Broadcast data will be sent out the active serial interface of all radios that receive it.

When a unicast is sent, it specifies a destination address in the network header. Only the radio that has the matching destination address then will send it out the serial port. This is called a directed broadcast. Any node that has a CE parameter set to route will rebroadcast the packet if its broadcast hops (BH) or broadcast radius values have not been depleted. If a repeated broadcast has already been seen, the node will ignore it. The NH parameter sets the maximum number of hops that a broadcast will be repeated. This value is always used, unless a BH value is specified that is smaller.

By default the CE parameter is set to not route broadcasts. Due to the long-range of the XLR PRO, it is advised to evaluate on a per-radio basis which should be configured as repeaters. This will provide a more reliable network by limiting the amount of congestion and RF traffic being generated.

Transmission timeouts

When a node receives an API TX Request (API configured XLR PRO radios) or an RO timeout occurs (XLR PRO configured for Transparent Mode) the time required to route the data to its destination depends on a number of configured parameters, whether the transmission is a unicast or a broadcast. Timeouts or timing information is provided for the following transmission types:

- Transmitting a broadcast

Note The timeouts in this section are theoretical timeouts and not precisely accurate. The application should pad the calculated maximum timeouts by a few hundred milliseconds. When using API mode, Tx Status API packets should be the primary method of determining if a transmission has completed.

Transmitting a broadcast

A broadcast transmission must be relayed by all routers in the network. The maximum delay would be when the sender and receiver are on the opposite ends of the network. The NH and %H parameters define the maximum broadcast delay as follows:

BroadcastTxTime=NH*%8

XLR PRO AT command reference tables

Special commands

Command	Name and description	Parameter range	Default
AC	**Apply Changes:** Immediately applies new settings without exiting command mode.	—	—
FR	**Software Reset:** Reset module. Responds immediately with an “OK” then performs a reset 100ms later.	—	—
RE	**Restore Defaults**: Restore module parameters to factory defaults.	—	—
WR	**Write:** Write parameter values to non-volatile memory so that parameter modifications persist through subsequent resets. Note: Once WR is issued, no additional characters should be sent to the module until after the "OK\r" response is received.	—	—

MAC/PHY level commands

Command	Name and description	Parameter range	Default
ID	**Network ID**. The user network identifier. Nodes must have the same network identifier to communicate. Changes to ID can be written to non-volatile memory using the WR command. Only modules with matching IDs can communicate with each other. When receiving a packet this is checked after the preamble ID. If using OEM network IDs, 0xFFFF will use the factory value.	0-0x7FFF	0x7FFF

Command	Name and description	Parameter range	Default
BR	**RF Data Rate**. The rate in which RF data is transmitted. All modes of operation will use this rate for RF data transmission. XLR PRO radios will receive data regardless of the BR value set. BR does not need to match on every radio on the network.	0 = 9.38 kbps 1 = 28.14 kbps 2 = 65.66 kbps 3 = 140.7 kbps 4 = 290.8 kbps 5 = 590.9 kbps 6 = 1.191 Mbps 7 = 2.392 Mbps 8 = 3.189 Mbps	4
PL	**Power Level**. Set/Read the power level at which the RF module transmits conducted power. Power levels are approximate.	0 = 0 dBm, (1 mW) 1 = +10 dBm, (10 mW) 2 = +20 dBm, (100 mW) 3 = +27 dBm, (500 mW) 4 = +30 dBm, (1 Watt)	4
RR	**Unicast Retries**. The maximum number of MAC level packet delivery attempts for unicasts. If RR is non-zero, packets sent from the radio will request an acknowledgment, and can be resent up to RR times if no acknowledgments are received.	0-0xF	0xA
MT	**Broadcast Multi-Transmit**. The number of additional MAC-level broadcast transmissions. All broadcast packets are transmitted MT+1 times to ensure it is received.	0-8	3

Diagnostics commands (MAC statistics and timeouts)

Command	Name and description	Parameter range	Default
DB	**Received Signal Strength**. This command reports the received signal strength of the last received RF data packet. The DB command only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link. The DB command value is measured in -dBm. For example if DB returns 0x60, then the RSSI of the last packet received was -96dBm.	0-0xFF [read-only]	0
EA	**MAC ACK Timeouts**. This count is incremented whenever a MAC ACK timeout occurs on a MAC level unicast. Once the number reaches 0xFFFF further events will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
ER	**Received Error Count**. This count is incremented whenever a packet is received which contained integrity errors of some sort. Once the number reaches 0xFFFF, further events will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0

Command	Name and description	Parameter range	Default
GD	**Good Packets Received**. This count is incremented whenever a good frame with a valid MAC header is received on the RF interface. Once the number reaches 0xFFFF, further events will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
TR	**Transmission Failure Count**. This count is incremented whenever a MAC transmission attempt exhausts all MAC retries without ever receiving a MAC acknowledgment message from the destination node. Once the number reaches 0xFFFF, further events will not be counted. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
UA	**Unicasts Attempted Count**. This count is incremented whenever a MAC unicast transmission occurs for which an ACK is requested. Once the number reaches 0xFFFF further events will not be counted. This value will not increment if RR = 0. The counter can be reset to any 16-bit value by appending a hexadecimal parameter to the command.	0-0xFFFF	0
%H	**MAC Unicast One Hop Time**. The MAC unicast one hop timeout in milliseconds. Changing MAC parameters can change this value.	[read-only]	0x267
%8	**MAC Broadcast One Hop Time**. The MAC broadcast one hop timeout in milliseconds. Changing MAC parameters can change this value.	[read-only]	0x23D
?N	**Network Discovery Timeout**. The maximum response time, in milliseconds, for network discovery responses (ND) and discover node (DN) responses. The timeout is based on the network discovery back-off time (NT) and the network propagation time.	[read-only]	0x3C41

RF network commands

Command	Name and description	Parameter range	Default
CE	**Node Messaging Options.** The module's routing and messaging mode bit field. Nodes can be configured to route, or not route, multi-hop packets when TO is configured for Directed Broadcast. The values from 0 to 2 are interpreted as follows: **0** Standard Router A routing module will repeat broadcasts. **1** NA **2** Non-Routing Node Disable routing on this node. When set, this node will not propagate broadcasts and will not act as a repeater.	0-2	2
BH	**Broadcast Hops**. The transmission hops for broadcast data transmissions. Set to 0 for maximum radius. If BH is set greater than NH then the value of NH is used.	0-4	0
NH	**Network Hops**. The maximum number of hops expected to be seen in a Directed Broadcast network.	0-4	4
NN	**Network Delay Slots**. Set or read the maximum random number of network delay slots before rebroadcasting a Directed Broadcast packet.	0 to 8	3

RF addressing commands

Command	Name and description	Parameter Range	Default
SH	**Serial Number High**. The upper 32 bits of the module's unique IEEE 64-bit RF MAC address.	0-0xFFFFFFFF [read-only]	Factory
SL	**Serial Number Low**. The lower 32 bits of the module's unique IEEE 64-bit RF MAC address.	0-0xFFFFFFFF [read-only]	Factory
DH	**Destination Address High**. The upper 32 bits of the 64-bit RF destination address. When combined with DL, it defines the destination address used for transmission of transparent data in either serial mode or IP socket mode.	0-0xFFFFFFFF	0
DL	**Destination Address Low**. The lower 32 bits of the 64-bit destination address. When combined with DH, DL defines the destination address used for transmission of transparent data in either serial mode or IP socket mode.	0-0xFFFFFFFF	0x0000FFFF

Command	Name and description	Parameter Range	Default
TO	**Transmit Options**. This command defines transmission options for all packets originating from this radio. These options can be overridden on a packet-by-packet basis by using the TxOptions field of the API TxRequest frames. **Bit** **Meaning** **Description** 6, 7 Delivery method b'00 - <invalid option> b'01 - Point-Multipoint b'10 - Repeater mode (directed broadcast of packets) 5 Reserved <set this bit to 0> 4 Reserved <set this bit to 0> 0 Disable ACK Disable acknowledgments on all unicasts Example #1: Setting TO to 0x80 would cause all transmissions to be sent using repeater mode. Example #2: Setting TO to 0x41 would cause all transmissions to be sent using point-to-multipoint, with network acknowledgments disabled.	0x40 - 0xBF [bitfield] Bits 4 & 5 must be set to 0	0x40 (point-to-multipoint)
NI	**Node Identifier**. A string identifier for this module. The string accepts only printable ASCII data In AT Command Mode, the string can not start with a space. A carriage return or comma ends the command. Command will automatically end when maximum bytes for the string have been entered. This string is returned as part of the ND (Network Discover) command. This identifier is also used with the DN (Destination Node) command.	up to 20 byte ASCII string	a space character
NT	**Node Discover Timeout**. The amount of time a node will spend discovering other nodes when ND or DN is issued. This value is used to randomize the responses to alleviate network congestion. The N? command can be used to determine the maximum response time a Node Discover will need based on NT and network propagation time.	0x20 - 0x2EE0 [x 100 msec]	0x82 (13 seconds)

Command	Name and description	Parameter Range	Default
NO	**Node Discovery Options**. The options value for the network discovery command. This bitfield value can change the behavior of the ND (network discovery) command and/or change what optional values are returned in any received ND responses or API node identification frames. Options include: • 0x01 = Append DD value (to ND responses or API node identification frames) • 0x02 = Local device sends ND or FN response frame when ND is issued • 0x04 = Append RSSI (of the last hop for DigiMesh networks) to ND or FN responses or API node identification frames Bitfield masks can have multiple options combined.	0-7 [bitfield]	0
CI	**Cluster ID**. The application layer cluster ID value. This value will be used as the cluster ID for all data transmissions. Supported Cluster IDs: 0x11 = Transparent data (default) 0x12 = Loopback 0x14 = Link Test 0x23 = Memory Access (GPM)	0-0xFFFF	0x11
DE	**Destination Endpoint**. The application layer destination endpoint value. This value will be used as the destination endpoint for all data transmissions. Supported Endpoints: 0xE8 = Digi data endpoint (default) 0xE6 = Digi device endpoint	0-0xFF	0xE8
SE	**Source Endpoint**. The application layer source endpoint value. This value will be used as the source endpoint for all data transmissions. Supported Endpoints: 0xE8 = Digi data endpoint (default) 0xE6 = Digi device endpoint	0-0xFF	0xE8

Addressing discovery/configuration commands

Command	Name and description	Parameter range	Default
DN	**Discover Node**. Resolves an NI (Node Identifier) string to a physical address (case sensitive). The following events occur after the destination node is discovered: <AT Firmware> **1** DL & DH are set to the extended (64-bit) address of the module with the matching NI (Node Identifier) string. **2** OK (or ERROR) <CR> is returned. **3** Command Mode is exited to allow immediate communication. <API Firmware> 0xFFFE and 64-bit extended addresses are returned in an API Command Response frame. If there is no response from a module within (NT * 100) milliseconds or a parameter is not specified (left blank), the command is terminated and an “ERROR” message is returned. In the case of an ERROR, Command Mode is not exited.	20 byte ASCII string	

Command	Name and description	Parameter range	Default

ND	**Network Discover**. Discovers and reports all RF modules found. The following information is reported for each module discovered.
	`MY<CR> (always 0xFFFE)`
	`SH<CR>`
	`SL<CR>`
	`NI<CR> (Variable length)`
	`PARENT_NETWORK ADDRESS<CR> (2 Bytes) (always 0xFFFE)`
	`DEVICE_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device)`
	`STATUS<CR> (1 Byte: Reserved)`
	`PROFILE_ID<CR> (2 Bytes)`
	`MANUFACTURER_ID<CR> (2 Bytes)`
	`DIGI DEVICE TYPE<CR> (4 Bytes. Optionally included based on NO settings.)`
	`RSSI OF LAST HOP<CR> (1 Byte. Optionally included based on NO settings.)`
	`<CR>`
	After (NT * 100) milliseconds, the command ends by returning a <CR>. ND also accepts a Node Identifier (NI) as a parameter (optional). In this case, only a module that matches the supplied identifier will respond.
	If the ND command is sent through a local API frame, each response is returned as a separate Local or Remote AT Command Response API packet, respectively. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a "0x00" null character
		20 byte ASCII string
		(optional)	

Command	Name and description	Parameter range	Default

FN	**Find Neighbors**. Discovers and reports all RF modules found within immediate RF range. The following information is reported for each module discovered.
	`MY<CR> (always 0xFFFE)`
	`SH<CR>`
	`SL<CR>`
	`NI<CR> (Variable length)`
	`PARENT_NETWORK ADDRESS<CR> (2 Bytes) (always 0xFFFE)`
	`DEVICE_TYPE<CR> (1 Byte: 0=Coord, 1=Router, 2=End Device)`
	`STATUS<CR> (1 Byte: Reserved)`
	`PROFILE_ID<CR> (2 Bytes)`
	`MANUFACTURER_ID<CR> (2 Bytes)`
	`DIGI DEVICE TYPE<CR> (4 Bytes. Optionally included based on NO settings.)`
	`RSSI OF LAST HOP<CR> (1 Byte. Optionally included based on NO settings.)`
	`<CR>`
	If the FN command is issued in command mode, after (NT*100) ms + overhead time, the command ends by returning a <CR>.
	If the FN command is sent through a local API frame, each response is returned as a separate Local or Remote AT Command Response API packet, respectively. The data consists of the above listed bytes without the carriage return delimiters. The NI string will end in a "0x00" null character.
		N/A	N/A

RF security commands

Command	Name and description	Parameter range	Default
KY	**AES Encryption Key**. Sets the 16 byte network security key value. This command is write-only; it cannot be read. Attempts to read KY will return an OK status. This command parameter must be set the same on all devices for communication to work. This value is passed in as hex characters when setting from AT command mode, and as binary bytes when set in ATI mode.	128-bit value	n/a

Serial interfacing commands

Command	Name and description	Parameter range	Default
BD	**Baud Rate**. The serial baud rate of the XLR PRO. This value will only affect the interface data rate for RS-232 and RS-485/422 data through the serial port. Values from 1-9 select preset standard rates. The values from 1 to 9 are interpreted as follows: **0** - n/a **4** - 19,200bps **8** - 230,400bps **1** - 2,400bps **5** - 38,400bps **9** - 480,600bps **2** - 4,800bps **6** - 57,600bps **3** - 9,600bps **7** - 115,200bps If a non-standard baud rate is needed, a value above 0x5B9 can be entered to select a specific baud rate. The BD parameter will adjust to the closest supported baud rate if a non-standard value is entered. After entering the specific baud rate, BD can be queried to read the actual baud rate. Baud rates can go as high as 6Mbps, but the host and serial switching circuitry may not support it.	0 to 9, and 0x5B9 to 0x5B8D80	3 (9600 bps)
NB	**Parity**. Set or read parity settings for serial communications. The values from 0 to 2 are interpreted as follows: **0** No parity **1** Even parity **2** Odd parity	0-2	0 (No parity)
SB	**Stop Bits**. The number of stop bits for the UART. **0** - One stop bit **1** - Two stop bits	0-1	0
RO	**Packetization Timeout**. The number of serial character times of inter-character silence required before packetization in transparent mode. Set (RO=0) to transmit characters as they arrive instead of buffering them into one RF packet.	0 - 0xFF [x character times]	3

Command	Name and description	Parameter range	Default
FT	**Flow Control Threshold**. The UART flow control threshold. De-assert CTS and/or send XOFF when FT bytes are in the UART receive buffer. Re-assert CTS when less than FT - 16 bytes are in the UART receive buffer.	0x11 - 0x94F	0x91F
AP	**API Mode**. The serial API mode. The following settings are allowed: **0** Transparent mode, API mode is off. All serial input and output is raw data and packets are delineated using the RO and RB parameters. **1** API mode without escapes is on. All UART input and output data is packetized in the API format. **2** API mode is on with escaped sequences inserted to allow for control characters (XON, XOFF, escape, and the 0x7e delimiter to be passed as data).	0 - 2	0
AO	API Options. The API data frame output format for received frames. This parameter applies to both the UART and SPI interfaces. **0** API RX Indicator (0x90) **1** API Explicit RX Indicator (0x91)	0 - 1	0
D6	RTS Flow Control. Enable or disable RTS flow control for the serial port. **0** = Disabled **1** = RTS flow control enabled	0 - 1	0
D7	**CTS Flow Control**. Enable or disable CTS flow control for the serial port **0** = Disabled **1** = CTS flow control enabled	0 - 1	1
4E	**Serial Protocol**. The serial protocol used for serial mode operation. **0** - RS-232 **1** - RS-485/422	0 - 1	0
4D	**RS-485 Duplex**. The duplex operation for RS-485. This will have no impact on RS-232 communication. **0** - 2-Wire (half-duplex) **1** - 4-Wire (full-duplex)	0 - 1	0
4T	**RS-485 Termination**. Enable or disable line termination for RS-485/422. Line termination will enable a 120 Ohm termination resistor on the RS-485/422 data lines. This will have no impact on RS-232 communication. **0** - No line termination **1** - Line termination enabled	0-1	0

Hardware diagnostics commands

Command	Name and description	Parameter range	Default
TP	Temperature. The temperature of the XLR RF module in degrees Celsius. The temperature value is displayed in 8-bit two's compliment format; for example 0x1A = 26C, and 0xF6 = -10C. Note: The RF module will naturally produce heat so this reading will usually be above the ambient temperature.	0x00 to 0xFF	n/a
RP	**RSSI Timer**. The amount of time that the RSSI LEDs will be active after a valid RF packet is received. When RP = 0xFF, output will always be on.	0 - 0xFF [x 100 ms]	0x28 (4 seconds)

Ethernet and IP socket mode commands

Command	Name and description	Parameter range	Default
ES	**Socket Mode**. Enable or disable IP socket mode. Enabling socket mode will allow serial traffic to be sent to a TCP or UDP port based on the IP parameter. The XLR PRO will be in a listen-only state unless DX is set to a valid IP address. **0** - IP socket mode disabled **1** - IP socket mode enabled	0 - 1	1
IB	**IP Socket Baud Rate**. The IP Socket baud rate of the XLR PRO. This value will only affect the interface data rate for serial TCP/UDP data through the Ethernet port. Values from 1-9 select preset standard rates. The values from 1 to 9 are interpreted as follows: **0** - n/a **4** - 19,200bp **8** - 230,400bps **1** - 2,400bps **5** - 38,400bps **9** - 480,600bps **2** - 4,800bps **6** - 57,600bps **3** - 9,600bps **7** - 115,200bps If a non-standard baud rate is needed, a value above 0x5B9 can be entered to select a specific baud rate. The IB parameter will adjust to the closest supported baud rate if a non-standard value is entered. After entering the specific baud rate, IB can be queried to read the actual baud rate.	0 - 9, and 0x5B9 - 0x70800	9 (480600 bps)
IP	**IP Protocol**. The IP protocol used for client and server socket connections in IP socket mode. **0** - TCP **1** - UDP	0 - 1	1 (TCP)
DX	**Destination IP Address**. The destination IPv4 address for outgoing IP socket mode data. 255.255.255.255 is a broadcast address.	0.0.0.0 - 255.255.255.255	0

Command	Name and description	Parameter range	Default
C0	**Source Port**. The listening IP port number for TCP and UDP traffic. An incoming socket will only be established if the protocol (TCP or UDP) matches what is set by the IP parameter.	0 - 0xFFFF	0x2616 (port 9750)
DY	**Destination Port**. The outgoing IP port number for TCP and UDP socket connections. A socket to this IP port will be made to the destination IPv4 address defined by the DX parameter using the protocol defined by the IP parameter.	0 - 0xFFFF	0x2616 (port 9750)
TM	TCP Client Connection Timeout. The timeout for outgoing TCP socket connections, this timer is used when the XLR PRO is acting as a TCP client. The connection will be closed if no activity is seen during this timeout period. When set to 0, the connection is closed immediately after data is sent. The maximum timeout is 1 day.	0 - 0x15180 [x 1 sec]	0x3C (60 seconds)
TS	TCP Server Connection Timeout. The timeout for incoming TCP socket connections. This timer is used when the XLR PRO is acting as a TCP server. The connection will be closed if no activity is seen during this timeout period. When set to 0, the connection is closed immediately after data is sent. The maximum timeout is 1 day.	0 - 0x15180 [x 1 sec]	0x3C (60 seconds)
MA	XLR IP Address. The IP address assignment mode. If the XLR PRO is configured for DHCP and no DHCP server is detected, Auto-IP will be used instead. **0** - DHCP **1** - Static IP	0 - 1	0 (DHCP)
MY	XLR IP Address. The IP address of the XLR PRO. If MA is configured for DHCP, this parameter is read-only and an IP address will be requested from an available DHCP server on the network. If no DHCP server is detected on the network after 1 minute, then an Auto-IP address will be assigned. After an Auto-IP address is assigned, the XLR PRO will request a DHCP address assignment every 5 minutes. The format of the Auto-IP address is as follows: 169.254.*xxx*.*yyy* *xxx* is the second to last byte of the Ethernet MAC address. *yyy* is the last byte of the Ethernet MAC address. If the Auto-IP address of the XLR PRO conflicts with another address on the network, then the Auto-IP address will be incremented by one until the conflict is resolved. Auto-IP Example: Ethernet MAC = 0x409D5A329. The last two bytes are 0xA3 and 0x29. When converted from hexadecimal to decimal format, these bytes become 136 and 41. The Auto-IP address that would be assigned to this radio is 169.254.163.41.	0.0.0.0 - 255.255.255.255	

Command	Name and description	Parameter range	Default
MK	Subnet Mask. The network subnet mask of the XLR PRO. If MA is configured for DHCP, this parameter is read-only and the subnet mask will be assigned by a DHCP server on the network. The subnet mask that is assigned in Auto-IP is 255.255.0.0.	0.0.0.0 - 255.255.255.255	
GW	Default Gateway Address. The gateway address of the XLR PRO. If MA is configured for DHCP, this parameter is read-only and the gateway address will be assigned by a DHCP server on the network. The gateway address that is assigned in Auto-IP is 0.0.0.0.	0.0.0.0 - 255.255.255.255	
NS	DNS Address. The IPv4 address of the domain name server for the XLR PRO.	0.0.0.0 - 255.255.255.255	
%M	Ethernet MAC Address. The Ethernet MAC Address assigned to the XLR PRO. This is a read-only parameter.	0x00 to 0xFF	Factory-set

Device Cloud commands

Command	Name and description	Parameter range	Default
KP	**Device Description**. Description of the XLR PRO that is displayed on Device Cloud and in the web configuration. This is a user-defined text field.	Up to 31 ASCII characters	
KC	**Device Contact**. Contact information for the XLR PRO that is displayed on Device Cloud and in the web configuration. This is a user-defined text field.	Up to 31 ASCII characters	0x00
KL	Device Location. The physical location of the XLR PRO that is displayed on Device Cloud and in the web configuration. This is a user-defined text field.	Up to 31 ASCII characters	
DO	**Device Cloud Enable**. Enable or disable Device Cloud support for the XLR PRO. **0** - Disable Device Cloud support **1** - Enable Device Cloud support	0 - 1	1 (Enabled)
EQ	**Device Cloud Server FQDN**. The fully qualified domain name of the Device Cloud server. If a valid DNS server is not set through the NS parameter, an IP address can be entered here.	Up to 63 ASCII characters	login.etherios.com
LX	**Latitude**. GPS latitude coordinates of the XLR PRO that is displayed on Device Cloud and in the web configuration. This is a user-defined text field, the XLR PRO does not have GPS functionality.	Up to 15 ASCII characters	0.000
LY	**Longitude**. GPS longitude coordinates of the XLR PRO that is displayed on Device Cloud and in the web configuration. This is a user-defined text field, the XLR PRO does not have GPS functionality.	Up to 31 ASCII characters	-0.000

Command	Name and description	Parameter range	Default
DO	**Device Cloud Indicator**. Enable or disable Device Cloud support for the XLR PRO. **0** - Connected **1** - Device Cloud not accessible **2** - Device Cloud connection in progress **3** - Disconnecting from Device Cloud **4** - Not configured for Device Cloud **5** - DHCP is enabled, and no DHCP server was found	0 - 5	

Web configuration commands

Command	Name and description	Parameter range	Default
HE	Web Configuration Enable. Enable or disable HTTP server on the XLR PRO for configuration via web browser. **0** - Disable web configuration **1** - Enable web configuration	0 - 1	1 (Enabled)
HU	**Web Configuration User Name**. The user name used to access the web configuration page.	Up to 63 ASCII characters	admin
HW	**Web Configuration Password**. The password used to access the web configuration page. This command is write-only; it cannot be read. Attempts to read HW will return an OK status.	Up to 63 ASCII characters	password

Ethernet RF bridging commands

Command	Name and Description	Parameter Range	Default
BE	Ethernet RF Bridging Enable. Enable or disable Ethernet RF bridging. When enabled, Ethernet traffic can be passed between up to 16 XLR PROs on the same RF network. **0** - Disable Ethernet RF bridging **1** - Enable Ethernet RF bridging **Note:** The XLR PRO does not support STP (Spanning Tree Protocol), if two XLR PROs are connected to the same Ethernet switch, a switching loop may occur.	0 - 1	0 (disabled)
BA	**Bridge Destination MAC**. The destination RF MAC address to use for Ethernet RF Bridging. 0xFFFF is used as a broadcast address.	0 - 0x0013A200FFFFFFFF	0xFFFF

AT command options

Command	Name and description	Parameter range	Default
CC	**Command Sequence Character**. Set or read the character to be used between guard times of the AT Command Mode Sequence. The AT Command Mode Sequence causes the XLR PRO to enter Command Mode (from Idle Mode).	0 - 0xFF Recommended: 0x20 - 0x7F	0x2B (ASCII "+" character)
CN	**Exit Command Mode**. Explicitly exit the module from AT Command Mode. Changes made in command mode will be applied, but not written to flash.	n/a	n/a
CT	**Command Mode Timeout.** Set/Read the period of inactivity (no valid commands received) after which the XLR PRO automatically exits AT Command Mode and returns to Idle Mode.	2 - 0x1770 [x 100ms]	0x64 (10 seconds)
GT	**Guard Times**. Set required period of silence before and after the Command Sequence Characters of the AT Command Mode Sequence (GT + CC + GT). The period of silence is used to prevent inadvertent entrance into AT Command Mode.	0 - 0xFFFF [x 1ms]	0x3E8 (1 second)

Firmware commands

Command	Name and description	Parameter range	Default
VB	**Firmware Version**.The firmware version that is running the XLR PRO.		
VR	**XLR RF Module Firmware Version**. Firmware version that is running on the XLR RF module.		
HV	**XLR RF Module Hardware Version**. Hardware version number of the XLR RF module.		
VH	**XLR Baseboard Hardware Version**. Hardware version number of the XLR Baseboard.		
*C	**Compatibility**. Hardware compatibility number.		
DD	**Device Type Identifier**. The Digi device type identifier value. This can be used to differentiate between multiple Digi products. The XLR PRO product code upper word is 0x000E.	0 - 0xFFFFFFFF	0xE0000
PN	**Part Number**. The manufacturing part number of the XLR PRO.		
CK	**Configuration CRC**. The CRC of the current settings. This command allows detection of unexpected configuration changes on the XLR PRO. After a firmware update, this command may return a different value even if the parameters have not been altered.	0 - 0xFFFF	

XLR PRO API operation

As an alternative to Transparent Operation, API (Application Programming Interface) Operations are available. API operation requires that communication with the XLR PRO be done through a structured interface (data is communicated in frames in a defined order). The API specifies how commands, command responses and XLR PRO status messages are sent and received from the XLR PRO using a serial data frame.

Please note that Digi may add new frame types to future versions of firmware, so please build into your software interface the ability to filter out additional API frames with unknown Frame Types.

API frame format

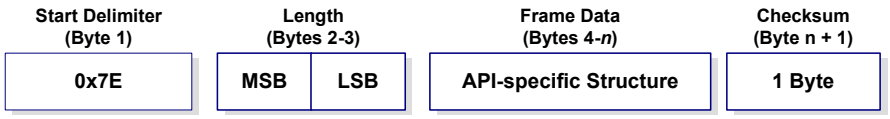
Two API modes are supported and both can be enabled using the AP (API Enable) command. Use the following AP parameter values to configure the XLR PRO to operate in a particular mode:

- AP = 1: API Operation
- AP = 2: API Operation (with escaped characters)

API operation (AP parameter = 1)

When this API mode is enabled (AP = 1), the serial data frame structure is defined as follows:

Serial data frame structure (no escapes):



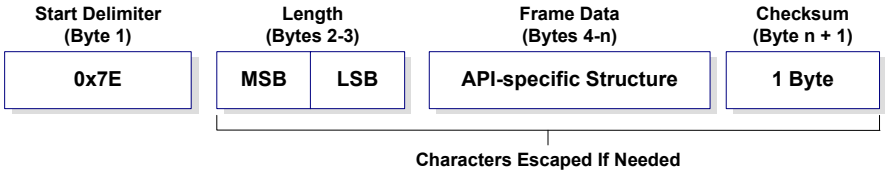
MSB = Most Significant Byte, LSB = Least Significant Byte

Any data received prior to the start delimiter is silently discarded. If the frame is not received correctly or if the checksum fails, the XLR PRO will reply with a radio status frame indicating the nature of the failure.

API operation with escape characters (AP parameter = 2)

When this API mode is enabled (AP = 2), the serial data frame structure is defined as follows:

serial data frame Structure with escape control characters:



MSB = Most Significant Byte, LSB = Least Significant Byte

Escape characters. When sending or receiving a serial data frame, specific data values must be escaped (flagged) so they do not interfere with the data frame sequencing. To escape an interfering data byte, insert 0x7D and follow it with the byte to be escaped XORed with 0x20.

Data bytes that need to be escaped:

- 0x7E – Frame Delimiter
- 0x7D – Escape
- 0x11 – XON
- 0x13 – XOFF

Example - *Raw serial data frame (before escaping interfering bytes):*
0x7E 0x00 0x02 0x23 0x11 0xCB

0x11 needs to be escaped which results in the following frame:
0x7E 0x00 0x02 0x23 0x7D 0x31 0xCB

Note In the above example, the length of the raw data (excluding the checksum) is 0x0002 and the checksum of the non-escaped data (excluding frame delimiter and length) is calculated as: 0xFF - (0x23 + 0x11) = (0xFF - 0x34) = 0xCB.

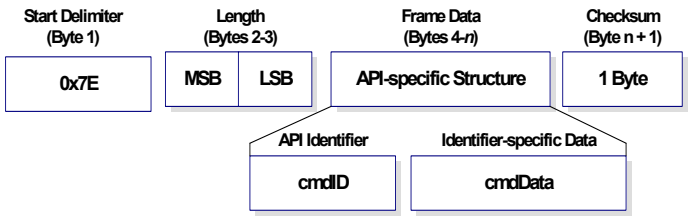
Length

The length field has two-byte value that specifies the number of bytes that will be contained in the frame data field. It does not include the checksum field.

Frame Data

Frame data of the serial data frame forms an API-specific structure as follows:

Serial Data Frame & API-specific Structure:



The cmdID frame (API-identifier) indicates which API messages will be contained in the cmdData frame (Identifier-specific data). Note that multi-byte values are sent big endian.The XLR PRO support the following API frames:

API frame names and values sent from the XLR PRO:

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
TX Request	0x10
Explicit TX Request	0x11
Remote Command Request	0x17

API frame names and values received from the XLR PRO

API Frame Names	API ID
AT Command Response	0x88
Modem Status	0x8A
Transmit Status	0x8B
RX Indicator (AO=0)	0x90
Explicit Rx Indicator (AO=1)	0x91
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97

Note Requests are less than 0x80, and responses are always 0x80 or higher.

Checksum

To test data integrity, a checksum is calculated and verified on non-escaped data.

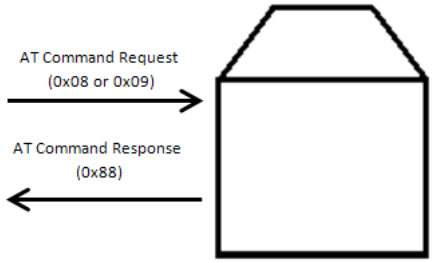
To calculate: Not including frame delimiters and length, add all bytes keeping only the lowest 8 bits of the result and subtract the result from 0xFF.

To verify: Add all bytes (include checksum, but not the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

XLR PRO API serial exchanges

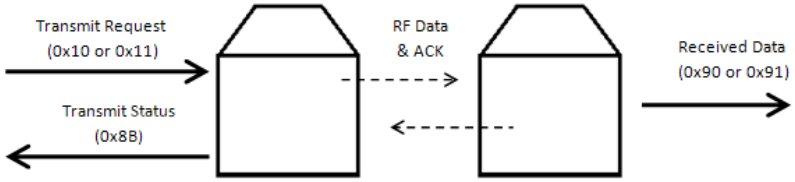
AT commands

The following image shows the API frame exchange that takes place at the serial interface when sending an AT command request to read or set a XLR PRO parameter. The response can be disabled by setting the frame ID to 0 in the request.



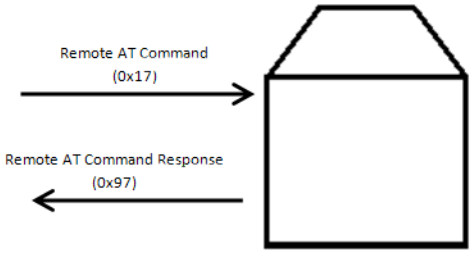
Transmitting and receiving RF data

The following image shows the API exchanges that take place at the serial interface when sending RF data to another device. The transmit status frame is always sent at the end of a data transmission unless the frame ID is set to 0 in the TX request. If the packet cannot be delivered to the destination, the transmit status frame will indicate the cause of failure. The received data frame (0x90 or 0x91) is set by the AP command.



Remote AT commands

The following image shows the API frame exchanges that take place at the serial interface when sending a remote AT command. A remote command response frame is not sent out the serial interface if the remote device does not receive the remote command.



Supporting the API

Applications that support the API should make provisions to deal with new API frames that may be introduced in future releases. For example, a section of code on a host microprocessor that handles received serial API frames (sent out the XLR PRO's active serial interface) might look like this:

```
void XLRPRO_HandleRxAPIFrame(_apiFrameUnion*papiFrame){
  switch(papiFrame->api_id){
    case RX_RF_DATA_FRAME:
      //process received RF data frame
      break;

    case NODE_IDENTIFICATION_FRAME:
      //process node identification frame
      break;

    default:
      //Discard any other API frame types that are not being used
      break;
  }
}
```

XLR PRO frame descriptions

The following sections illustrate the types of frames encountered while using the API.

AT command

Frame type: 0x08
Used to query or set XLR PRO parameters on the local device. This API command applies changes after executing the command. (Changes made to XLR PRO parameters take effect once changes are applied.)

The API example below illustrates an API frame when querying the ID parameter value of the XLR PRO.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x04	
Frame-specific data	Frame type	3	0x08	
	Frame ID	4	0x52	Identifies this command for correlation to a later response frame (0x88) to this command. If set to 0, no response frame will be sent.
	AT command	5	0x49	Command name —Two ASCII characters that identify the AT command **ID**.
		6	0x44	
	Parameter value (optional)			If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
Checksum		8	0x18	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

AT command—queue parameter value

Frame type: 0x09
This API type allows XLR PRO parameters to be queried or set. In contrast to the 0x08, "AT Command"

API type, new parameter values are queued and not applied until either the “AT Command” (0x08) API type or the AC (Apply Changes) command is issued. Register queries (reading parameter values) are returned immediately.

Send a command to change the baud rate (BD) to 115200 baud, but don't apply changes yet. (XLR PRO will continue to operate at the previous baud rate until changes are applied.)

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x05	
Frame-specific data	Frame type	3	0x09	
	Frame ID	4	0x01	Identifies the UART data frame for the host to correlate with a subsequent ACK (acknowledgment). If set to 0, no response is sent.
	AT command	5	0x42	Command Name—Two ASCII characters that identify the AT command **BD**.
		6	0x44	
	Parameter value (ATBD7 = 115200 baud)		0x07	If present, indicates the requested parameter value to set the given register. If no characters present, register is queried.
Checksum		8	0x68	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

Note In this example, the parameter could have been sent as a zero-padded 2-byte or 4-byte value.

TX request

Frame type: 0x10
A TX Request API frame causes the XLR PRO to send data as an RF packet to the specified destination.

The 64-bit destination address should be set to 0x000000000000FFFF for a broadcast transmission (to all devices). For unicast transmissions the 64 bit address field should be set to the address of the desired destination node. The reserved field should be set to 0xFFFE.

This example shows if escaping is disabled (AP=1).

Frame fields		Offset	Example	Description
Start Delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x16	

Frame fields		Offset	Example	Description
Frame-specific data	Frame type	3	0x10	
	Frame ID	4	0x01	Identifies this command for correlation to a later response frame (0x8B) to this command. If set to 0, no response frame will be sent.
	64-bit destination address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF—Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x0A	
		11	0x01	
		LSB 12	0x27	
	Reserved	13	0xFF	Set to 0xFFFE.
		14	0xFE	
	Broadcast radius	15	0x00	Sets maximum number of hops a broadcast transmission can occur. If set to 0, the broadcast radius will be set to the maximum hops value.
	Transmit options	16	0x00	If the Transmit Options Bitfield is 0, then the TO parameter will be used. Bitfield: bit 0: Disable ACK bits 6,7: b'01 - Point-Multipoint b'10 - Repeater mode (directed broadcast) All other bits must be set to 0.
	RF data	17	0x54	Payload data that is sent to the destination device.
		18	0x78	
		19	0x44	
		20	0x61	
		21	0x74	
		22	0x61	
		23	0x30	
		24	0x41	
Checksum		25	0x13	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

Example: The example above shows how to send a transmission to an XLR PRO where escaping is disabled (AP=1) with destination address 0x0013A200 40014011, payload "TxData0A". If escaping is enabled (AP=2), the frame should look like:

0x7E 0x00 0x16 0x10 0x01 0x00 0x7D 0x33 0xA2 0x00 0x40 0x0A 0x01 0x27

0xFF 0xFE 0x00 0x00 0x54 0x78 0x44 0x61 0x74 0x61 0x30 0x41 0x7D 0x33

The checksum is calculated (on all non-escaped bytes) as [0xFF - (sum of all bytes from API frame type through data payload)].

Explicit TX Request

Frame type: 0x11
Allows application layer fields (endpoint and cluster ID) to be specified for a data transmission. Similar to the TX Request, but also requires application layer addressing fields to be specified (endpoints, cluster ID, profile ID). An Explicit TX Request API frame causes the XLR PRO to send data as an RF packet to the specified destination, using the specified source and destination endpoints, cluster ID, and profile ID.

The 64-bit destination address should be set to 0x000000000000FFFF for a broadcast transmission (to all devices). For unicast transmissions the 64 bit address field should be set to the address of the desired destination node. The reserved field should be set to 0xFFFE.

The broadcast radius can be set from 0 up to NH to 0xFF. If the broadcast radius exceeds the value of NH then the value of NH will be used as the radius. This parameter is only used for broadcast transmissions.

The maximum number of payload bytes can be read with the NP command.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x1A	
Frame-specific data	Frame type	3	0x11	
	Frame ID	4	0x01	Identifies this command for correlation to a later response frame (0x8B) to this command. If set to 0, no response frame will be sent.
	64-bit destination address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF—Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x01	
		10	0x23	
		11	0x84	
		LSB12	0x00	
	Reserved	13	0xFF	Set to 0xFFFE.
		14	0xFE	
	Source endpoint	15	0xA0	Source endpoint for the transmission.
	Destination endpoint	16	0xA1	Destination endpoint for the transmission.
	Cluster ID	17	0x15	Cluster ID used in the transmission.
		18	0x54	
	Profile ID	19	0xC1	Profile ID used in the transmission.
		20	0x05	
	Broadcast radius	21	0x00	Sets the maximum number of hops a broadcast transmission can traverse. If set to 0, the transmission radius will be set to the network maximum hops value.
	Transmit options	22	0x00	If the Transmit Options Bitfield is 0, then the TO parameter will be used. Bitfield: bit 0: Disable ACK bits 6,7: b'01 - Point-Multipoint b'10 - Repeater mode (directed broadcast) All other bits must be set to 0.
	Data payload	23	0x54	Payload data that is sent to the destination device.
		24	0x78	
		25	0x44	
		26	0x61	
		27	0x74	
		28	0x61	
Checksum		29	0xDD	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

The above example sends a data transmission to a radio with a 64-bit address of 0x0013A20001238400, using a source endpoint of 0xA0, destination endpoint 0xA1, cluster ID 0x1554, and profile ID 0xC105. Payload will be "TxData".

Remote AT command request

Frame type: 0x17
Used to query or set XLR PRO parameters on a remote device. For parameter changes on the remote device to take effect, changes must be applied, either by setting the apply changes options bit, or by sending an AC command to the remote.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x10	
Frame-specific data	Frame type	3	0x17	
	Frame ID	4	0x01	Identifies this command for correlation to a later response frame (0x97) to this command. If set to 0, no response frame will be sent.
	64-bit destination address	MSB 5	0x00	Set to the 64-bit address of the destination device. The following address is also supported: 0x000000000000FFFF—Broadcast address
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x40	
		11	0x11	
		LSB 12	0x22	
	Reserved	13	0xFF	Set to 0xFFFE.
		14	0xFE	
	Remote command options	15	0x02	0x02—Apply changes on remote. (If not set, AC command must be sent before changes will take effect.) All other bits must be set to 0.
	AT command	16	0x42	Command Name—Two ASCII characters that identify the AT command **BE**.
		17	0x45	
	Command parameter	18	0x01	If present, indicates the requested parameter value to set the given register. If no characters present, the register is queried.
Checksum		18	0xF8	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

The above example sends a remote command to change the bridging enabled register on a remote device to 1 (enable Ethernet RF bridging on the remote device), and apply changes so the new configuration value immediately takes effect. In this example, the 64-bit address of the remote is 0x0013A200 40401122.

AT command response

Frame type: 0x88
In response to an AT Command message, the XLR PRO will send an AT Command Response message. Some commands will send back multiple frames (for example, the ND (Node Discover) command).

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x05	

Frame fields		Offset	Example	Description
Frame-specific data	Frame type	3	0x88	
	Frame ID	4	0x01	Identifies the serial interface data frame being reported. Note: If Frame ID = 0 in the associated request frame then no response frame will be delivered..
	AT command	5	0x42	Command Name—Two ASCII characters that identify the AT command **BD**.
		6	0x44	
	Command status	7	0x00	The least significant nibble indicates the command status: 0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
	Command data			Register data in binary format. If the register was set, then this field is not returned, as in this example.
Checksum		8	0xF0	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

Suppose the BD parameter is changed on the local device with a frame ID of 0x01. If successful (parameter was valid), the above response would be received.

Modem status

Frame type: (0x8A)
Radio status messages are sent from the XLR PRO in response to specific conditions.

The following API frame is returned when an API device powers up.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x02	
Frame-specific data	Frame type	3	0x8A	
	Status	4	0x00	0x00 = Hardware reset 0x01 = Watchdog timer reset
Checksum		5	0x75	0xFF—The 8- bit sum of bytes from offset 3 to this byte.

Transmit status

Frame type: 0x8B
When a TX Request is completed, the XLR PRO sends a TX Status message. This message will indicate if the packet was transmitted successfully or if there was a failure.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x07	

Frame fields		Offset	Example	Description
Frame-specific data	Frame type	3	0x8B	
	Frame ID	4	0x47	Identifies the serial interface data frame being reported. Note: If Frame ID = 0 in the associated request frame then no response frame will be delivered.
	Reserved	5	0xFF	Reserved.
		6	0xFE	
	Transmit retry count	7	0x00	The number of application transmission retries that took place.
	Delivery status	8	0x00	0x00 = Success 0x01 = MAC ACK Failure 0x02 = Collision Avoidance Failure 0x21 = Network ACK Failure 0x31 = Internal Resource Error 0x32 = Internal Error 0x74 = Payload too large
	Discovery status	9	0x02	0x00 = No Discovery Overhead 0x02 = Route Discovery
Checksum		10	0x2E	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

In the above example, a unicast data transmission was sent successfully to a destination device using a frame ID of 0x47.)

RX indicator

Frame type: (0x90)
When the XLR PRO receives an RF data packet, it is sent out the active serial interface using this message type.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x12	

Frame fields		Offset	Example	Description
Frame-specific data	Frame type	3	0x90	
	64-bit source address	MSB 4	0x00	
		5	0x13	64-bit address of sender.
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	Reserved	12	0xFF	Reserved
		13	0xFE	
	Receive options	14	0x01	bit 0: Packet was acknowledged. bit 1: Broadcasted packet. bits 6,7: b'01 - Point-Multipoint b'10 - Repeater mode (directed broadcast) other bits should be ignored.
	Received Data	15	0x52	Received RF data.
		16	0x78	
		17	0x44	
		18	0x61	
		19	0x74	
		20	0x61	
Checksum		21	0x11	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

Example: In the above example, a device with a 64-bit address of 0x0013A200 40522BAA sends a unicast data transmission to a remote device with payload "RxData". If AO=0 on the receiving device, it would send the above frame out its serial interface.

Explicit rx indicator

Frame type:0x91
When the XLR PRO receives an RF packet it is sent out the active serial interface using this message type (when AO=1).

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x18	

Frame fields		Offset	Example	Description
Frame-specific data	Frame type	3	0x91	
	64-bit source address	MSB 4	0x00	64-bit address of sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	Reserved	12	0xFF	Reserved.
		13	0xFE	
	Source endpoint	14	0xE0	Endpoint of the source that initiated the transmission
	Destination endpoint	15	0xE0	Endpoint of the destination the message is addressed to.
	Cluster ID	16	0x22	Cluster ID the packet was addressed to.
		17	0x11	
	Profile ID	18	0xC1	Profile ID the packet was addressed to.
		19	0x05	
	Receive options	20	0x02	bit 0: Packet was acknowledged. bit 1: Broadcasted packet. bits 6,7 b'01 - Point-Multipoint b'10 - Repeater mode (directed broadcast) other bits should be ignored.
	Received data	21	0x52	Received RF data
		22	0x78	
		23	0x44	
		24	0x61	
		25	0x74	
		26	0x61	
Checksum		27	0x56	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

In the above example, a device with a 64-bit address of 0x0013A200 40522BAA sends a broadcast data transmission to a remote device with payload "RxData". Suppose the transmission was sent with source and destination endpoints of 0xE0, cluster ID=0x2211, and profile ID=0xC105. If AO=1 on the receiving device, it would send the above frame out its serial interface.

Data sample rx indicator

Frame Type: 0x92
When the XLR PRO receives an RF packet it is sent out the active serial interface using this message type (when AO=1).

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum.
		LSB 2	0x14	

Frame fields		Offset	Example	Description
Frame-specific Data	Frame type	3	0x92	
	64-bit source (remote) address	MSB 4	0x00	64-bit address to sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x52	
		10	0x2B	
		LSB 11	0xAA	
	16-bit source network address	MSB 12	0x7D	16-bit address of sender.
		LSB 13	0x84	
	Receive options	14	0x01	0x01 - Packet Acknowledged. 0x02 - Packet was a broadcast packet. All other bits are reserved and should be ignored.
	Number of samples	15	0x01	Number of sample sets included in the payload. (Always set to 1.)
	Digital channel mask*	16	0x00	Bitmask field that indicates which digital IO lines on the remote have sampling enabled (if any).
		17	0x1C	
	Analog channel mask***	18	0x02	Bitmask field that indicates which analog IO lines on the remote have sampling enabled (if any).
	Digital samples (if included)	19	0x00	If the sample set includes any digital IO lines (Digital Channel Mask > 0), these two bytes contain samples for all enabled digital IO lines. DIO lines that do not have sampling enabled return 0. Bits in these 2 bytes map the same as they do in the Digital Channels Mask field.
		20	0x14	
	Analog sample	21	0x02	If the sample set includes any analog input lines (Analog Channel Mask > 0), each enabled analog input returns a 2-byte value indicating the A/D measurement of that input. Analog samples are ordered sequentially from AD0/DIO0 to AD3/DIO3, to the supply voltage.
		22	0x25	
Checksum		23	0xF5	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

Node identification indicator

Frame Type:0x95
This frame is received when a XLR PRO transmits a node identification message to identify itself (when AO=0). The data portion of this frame is similar to a network discovery response frame (see ND command).

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x25	

Frame fields		Offset	Example	Description
Frame-specific Data	Frame type	3	0x95	
	64-bit source address	MSB 4	0x00	64-bit address of sender.
		5	0x13	
		6	0xA2	
		7	0x00	
		8	0x40	
		9	0x74	
		10	0x02	
		LSB 11	0xAC	
	Reserved	12	0xFF	Reserved.
		13	0xFE	
	Receive options	14	0xC2	0x01 - Packet Acknowledged 0x02 - Packet was a broadcast packet 0x40 - Point-multipoint packet 0x80 - Directed broadcast packet
	Reserved	15	0xFF	Reserved
		16	0xFE	
	64-bit address	MSB 17	0x00	Indicates the 64-bit address of the remote XLR PRO that transmitted the node identification frame.
		18	0x13	
		19	0xA2	
		20	0x00	
		21	0x40	
		22	0x74	
		23	0x02	
		LSB 24	0xAC	
	NI string	25	0x20	Node identifier string on the remote device. The NI string is terminated with a NULL byte (0x00).
		26	0x00	
	Reserved	27	0xFF	Reserved
		28	0xFE	
	Device type	29	0x01	0=Coordinator 1=Normal Mode 2=End Device See the NO command description for more options.
	Source event	30	0x01	1=Frame sent by node identification pushbutton event (see D0 command description).
	Digi profile ID	31	0xC1	Set to Digi application profile ID.
		32	0x05	
	Digi manufacturer ID	33	0x10	Set to Digi manufacturer ID.
		34	0x1E	
	Digi DD value (optional)	35	0x00	Reports the DD value of the responding XLR PRO (this field can be enabled with the NO command).
		36	0x0C	
		37	0x00	
		38	0x00	
	RSSI (optional)	39	0x2E	RSSI (This field can be enabled with the NO command.)
Checksum		40	0x33	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

If the commissioning push button is pressed on a remote router device with 64-bit address 0x0013a200407402ac and default NI string, the following node identification indicator would be received: 0x7e 0025 9500 13a2 0040 7402 acff fec2 fffe 0013 a200 4074 02ac 2000 fffe 0101 c105 101e 000c 0000 2e33

Remote command response

Frame type: 0x97
If an XLR PRO receives a remote command response RF data frame in response to a Remote AT Command Request, the XLR PRO will send a Remote AT Command Response message out the active serial interface. Some commands may send back multiple frames--for example, Node Discover (ND) command.

Frame fields		Offset	Example	Description
Start delimiter		0	0x7E	
Length		MSB 1	0x00	Number of bytes between the length and the checksum
		LSB 2	0x13	
Frame-specific data	Frame type	3	0x97	
	Frame ID	4	0x55	This is the same value passed in to the request. If Frame ID = 0 in the associated request frame then no response frame will be delivered.
	64-bit source (remote) address	MSB 5	0x00	The address of the remote radio returning this response.
		6	0x13	
		7	0xA2	
		8	0x00	
		9	0x40	
		10	0x52	
		11	0x2B	
		LSB 12	0xAA	
	Reserved	13	0xFF	Reserved
		14	0xFE	
	AT commands	15	0x53	Name of the command
		16	0x4C	
	Command status	17	0x00	The least significant nibble indicates the command status: 0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter The most significant nibble is a bitfield as follows: 0x40 = The RSSI field is invalid and should be ignored. 0x80 = Response is a remote command.
	Command data	18	0x40	The value of the required register
		19	0x52	
		20	0x2B	
		21	0xAA	
Checksum		22	0xF4	0xFF—The 8-bit sum of bytes from offset 3 to this byte.

If a remote command is sent to a remote device with 64-bit address 0x0013A200 40522BAA to query the SL command, and if the frame ID=0x55, the response would look like the above example.

XLR PRO advanced application features

Remote configuration commands

An XLR PRO in API mode has provisions to send configuration commands to remote devices using the Remote Command Request API frame (See API Operations chapter.) This API frame can be used to send commands to a remote XLR PRO to read or set command parameters.

Sending a remote command

To send a remote command, the Remote Command Request frame should be populated with the 64-bit address of the remote device, the correct command options value, and the command and parameter data (optional). If a command response is desired, the Frame ID should be set to a non-zero value. Only unicasts of remote commands are supported. Remote commands cannot be broadcast.

Applying changes on remote devices

When remote commands are used to change command parameter settings on a remote device, parameter changes do not take effect until the changes are applied. For example, changing the BD parameter will not change the actual serial interface rate on the remote until the changes are applied. Changes can be applied using remote commands in one of three ways:

- Set the apply changes option bit in the API frame
- Issue an AC command to the remote device
- Issue a WR + FR command to the remote device to save changes and reset the device.

Remote command responses

If the remote device receives a remote command request transmission, and the API frame ID is non-zero, the remote will send a remote command response transmission back to the device that sent the remote command. When a remote command response transmission is received, a device sends a remote command response API frame out its serial interface. The remote command response indicates the status of the command (success, or reason for failure), and in the case of a command query, it will include the register value. The device that sends a remote command will not receive a remote command response frame if:

- The destination device could not be reached
- The frame ID in the remote command request is set to 0.

Network commissioning and diagnostics

Network commissioning is the process whereby devices in a network are discovered and configured for operation. The XLR PRO includes several features to support device discovery and configuration. To accommodate these requirements, the XLR PRO includes various features to aid in device placement, configuration, and network diagnostics.

Device configuration

The XLR PRO can be configured locally through serial commands (AT or API), or remotely through remote API commands. API devices can send configuration commands to set or read the configuration settings of any device in the network.

Device placement

For a repeater network installation to be successful, the installer must be able to determine where to place individual XLR PRO devices to establish reliable RF links throughout the network.

Loopback testing

A good way to measure the performance of a network is to send unicast data through the network from one device to another to determine the success rate of many transmissions. To simplify link testing, the modules support a loopback cluster ID (0x12) on the data endpoint (0xE8). Any data sent to this cluster ID on the data endpoint will be transmitted back to the sender. This is how a range test is accomplished in XCTU.

The configuration steps to send data to the loopback cluster ID depend on the AP setting:

AT Configuration (AP=0)

To send data to the loopback cluster ID on the data endpoint of a remote device, set the CI command value to 0x12. The SE and DE commands should be set to 0xE8 (default value). The DH and DL commands should be set to the address of the remote (the 64-bit address of the remote). After exiting command mode, any received serial characters will be transmitted to the remote device, and returned to the sender.

API Configuration (AP=1 or AP=2)

Send an Explicit Addressing Command API frame (0x11) using 0x12 as the cluster ID and 0xE8 as the source and destination endpoint. Data packets received by the remote will be echoed back to the sender.

RSSI Indicators

It is possible to measure the received signal strength on a device using the DB command. DB returns the RSSI value (measured in -dBm) of the last received packet. However, this number can be misleading in a repeater network. The DB value only indicates the received signal strength of the last hop if using repeater mode. If a transmission spans multiple hops, the DB value provides no indication of the overall transmission path, or the quality of the worst link - it only indicates the quality of the last link and should be used accordingly.

Device discovery

Network discovery

The network discovery command can be used to discover all Digi modules that have joined a network. Issuing the ND command sends a broadcast network discovery command throughout the network. All devices that receive the command will send a response that includes the device's

addressing information, node identifier string (see NI command), and other relevant information. This command is useful for generating a list of all module addresses in a network.

When a device receives the network discovery command, it waits a random time before sending its own response. The maximum time delay is set on the ND sender with the NT command. The ND originator includes its NT setting in the transmission to provide a delay window for all devices in the network. Large networks may need to increase NT to improve network discovery reliability. The default NT value is 0x82 (13 seconds).

Neighbor polling

The neighbor poll command can be used to discover the modules which are immediate neighbors (within RF range) of a particular node. This command is useful in determining network topology. The command is issued using the FN command. The FN command can be initiated locally on a node using AT command mode or by using a local AT command request frame. The command can also be initiated remotely by sending the target node an FN command using a remote AT command request API frame. This is how the network map in XTU is built.

A node which executes an FN command will send a broadcast to all of its immediate neighbors. All radios which receive this broadcast will send an RF packet to the node that initiated the FN command. In the case where the command is initiated remotely this means that the responses are sent directly to the node which sent the FN command to the target node. The response packet is output on the initiating radio in the same format as a network discovery frame.

General purpose flash memory

XLR PRO modules provide 119 512-byte blocks of flash memory which can be read and written by the user application. This memory provides a non-volatile data storage area which can be used for a multitude of purposes. Some common uses of this data storage include: storing logged sensor data, buffering firmware upgrade data for a host microcontroller, or storing and retrieving data tables needed for calculations performed by a host microcontroller. The General Purpose Memory (GPM) is also used to store a firmware upgrade file for over-the-air firmware upgrades of the XLR PRO itself.

Accessing general purpose flash memory

The GPM of a target node can be accessed locally or over-the-air by sending commands to the MEMORY_ACCESS cluster ID (0x23) on the DIGI_DEVICE endpoint (0xE6) of the target node using explicit API frames. (Explicit API frames are described in the API Operation section.

To issue a GPM command, the payload of an explicit API frame should be formatted as follows:

Payload byte offset	Number of bytes	Field name	General field description
0	1	GPM_CMD_ID	Specific GPM commands are described below.
1	1	GPM_OPTIONS	Command-specific options.
2	2[1]	GPM_BLOCK_NUM	Block number addressed in the GPM.
4	2*	GPM_START_INDEX	Byte index within the addressed GPM block.
6	2*	GPM_NUM_BYTES	Number of bytes in the GPM_DATA field, or in the case of a READ, the number of bytes requested.
8	varies	GPM_DATA	

1. Specify multi-byte parameters using big-endian byte ordering.

When a GPM command is sent to a radio via a unicast, the receiving radio will unicast a response back to the requesting radio's source endpoint specified in the request packet. No response is sent for broadcast requests. If the source endpoint is set to the DIGI_DEVICE endpoint (0xE6) or explicit API mode is enabled on the requesting radio, then a GPM response will be output as an explicit API RX indicator frame on the requesting node (assuming API mode is enabled).

The format of the response is similar to the request packet:

Payload byte offset	Number of bytes	Field name	General field description
0	1	GPM_CMD_ID	This field will be the same as the request field
1	1	GPM_STATUS	Status indicating whether the command was successful
2	2[1]	GPM_BLOCK_NUM	The block number addressed in the GPM
4	2*	GPM_START_INDEX	The byte index within the addressed GPM block
6	2*	GPM_NUM_BYTES	The number of bytes in the GPM_DATA field
8	varies	GPM_DATA	

1. Specify multi-byte parameters using big-endian byte ordering.

The following commands exist for interacting with GPM:

- **PLATFORM_INFO_REQUEST (0x00):** A PLATFORM_INFO_REQUEST frame can be sent to query details of the GPM structure.

Field name	Command-specific description
GPM_CMD_ID	Set to PLATFORM_INFO_REQUEST (0x00).
GPM_OPTIONS	Set to 0. This field is unused for this command.
GPM_BLOCK_NUM	Set to 0. This field is unused for this command.
GPM_START_INDEX	Set to 0. This field is unused for this command.
GPM_NUM_BYTES	Set to 0. This field is unused for this command.
GPM_DATA	No data bytes should be specified for this command.

- **PLATFORM_INFO (0x80):** When a PLATFORM_INFO_REQUEST command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Set to PLATFORM_INFO (0x80).
GPM_STATUS	A one (1) in the least-significant bit indicates an error occurred. All other bits are reserved.
GPM_BLOCK_NUM	Indicates the number of available GPM blocks.
GPM_START_INDEX	Indicates the size, in bytes, of a GPM block.

Field name	Command-specific description
GPM_NUM_BYTES	Number of bytes in the GPM_DATA field. For this command, this field is set to 0.
GPM_DATA	No data bytes should be specified for this command.

Example:

A PLATFORM_INFO_REQUEST sent to a radio with a serial number of 0x0013a200407402AC should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 00 00 00 0000 0000 0000 24

Assuming all transmissions were successful, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76

7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 80 00 0077 0200 0000 EB

- E**RASE (0x01):** The ERASE command erases (writes all bits to binary 1) one or all of the GPM flash blocks. The ERASE command can also be used to erase all GPM blocks by setting the GPM_NUM_BYTES field to 0.

Field name	Command-specific description
GPM_CMD_ID	Set to ERASE (0x01).
GPM_OPTIONS	At present, there are no defined options for the ERASE command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block to erase. When erasing all GPM blocks, this field is ignored (set to 0).
GPM_START_INDEX	The ERASE command works on entire GPM blocks only—the command cannot be used to erase part of a GPM block. For this reason, set GPM_START_INDEX to 0 (unused).
GPM_NUM_BYTES	Set the GPM_NUM_BYTES to 0 to erase all GPM flash blocks or set GPM_NUM_BYTES to the GPM flash block size.
GPM_DATA	No data bytes should be specified for this command.

- **ERASE_RESPONSE (0x81):** When an ERASE command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Set to ERASE_RESPONSE (0x81).
GPM_STATUS	A one (1) in the least-significant bit indicates an error occurred. All other bits are reserved.
GPM_BLOCK_NUM	Matches the parameter passed in the request frame.
GPM_START_INDEX	Matches the parameter passed in the request frame.

Field name	Command-specific description
GPM_NUM_BYTES	Number of bytes in the GPM_DATA field. For this command, set to 0.
GPM_DATA	No data bytes should be specified for this command.

Example:

To erase flash block 42 of a target radio with serial number of 0x0013a200407402ac an ERASE packet should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 C0 01 00 002A 0000 0200 37

Assuming all transmissions were successful, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76

7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 81 00 002A 0000 0000 39

- **WRITE (0x02) and ERASE_THEN_WRITE (0x03):** The WRITE command writes the specified bytes to the specified GPM location. Before writing bytes to a GPM block, make sure all the bytes have first been erased. The ERASE_THEN_WRITE command performs an ERASE of the entire GPM block specified with the GPM_BLOCK_NUM field prior to doing a WRITE.

Field name	Command-specific description
GPM_CMD_ID	Set to WRITE (0x02) or ERASE_THEN_WRITE (0x03).
GPM_OPTIONS	At present, there are no defined options for this command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block to be written.
GPM_START_INDEX	Set to the byte index within the GPM block where the data should be written.
GPM_NUM_BYTES	Set to the number of bytes specified in the GPM_DATA field. Only one GPM block can be operated on per command. For this reason, the GPM_START_INDEX plus the GPM_NUM_BYTES cannot be greater than the GPM block size. **Note** The number of bytes sent in an explicit API frame (including the GPM command fields) cannot exceed the maximum payload size of the radio. Use the ATNP command to query the maximum payload size.
GPM_DATA	Data to be written.

- **WRITE _RESPONSE (0x82) and ERASE_THEN_WRITE_RESPONSE(0x83):** When a WRITE or ERASE_THEN_WRITE command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Set to WRITE_RESPONSE (0x82) or ERASE_THEN_WRITE_RESPONSE (0x83).
GPM_STATUS	A one (1) in the least-significant bit indicates an error occurred. All other bits are reserved.
GPM_BLOCK_NUM	Matches the parameter passed in the request frame.

Field name	Command-specific description
GPM_START_INDEX	Matches the parameter passed in the request frame.
GPM_NUM_BYTES	Number of bytes in the GPM_DATA field. For this command, set to 0.
GPM_DATA	No data bytes should be specified for these commands.

Example:

To write 15 bytes of incrementing data to flash block 22 of a target radio with serial number of 0x0013a200407402ac a WRITE packet should be formatted as follows (spaces added to delineate fields):

7E 002B 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 C0 02 00 0016 0000 000F 0102030405060708090A0B0C0D0E0F C5

Assuming all transmissions were successful and that flash block 22 was previously erased, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76

7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 82 00 0016 0000 0000 4C

- **READ (0x04):** The READ command can be used to read the specified number of bytes from the GPM location specified. Data can be queried from only one GPM block per command.

Field name	Command-specific description
GPM_CMD_ID	Set to READ (0x04).
GPM_OPTIONS	At present, there are no defined options for this command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block to read.
GPM_START_INDEX	Set to the byte index within the GPM block where the data should be read.
GPM_NUM_BYTES	Set to the number of data bytes to read. Only one GPM block can be operated on per command. For this reason, the GPM_START_INDEX plus the GPM_NUM_BYTES cannot be greater than the GPM block size. **Note** The number of bytes sent in an explicit API frame (including the GPM command fields) cannot exceed the maximum payload size of the radio. Use the ATNP command to query the maximum payload size.
GPM_DATA	No data bytes should be specified for this command.

- **READ _RESPONSE (0x84):** When a READ command request has been unicast to a node, that node sends a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Set to READ_RESPONSE (0x84).
GPM_STATUS	A one (1) in the least-significant bit indicates an error occurred. All other bits are reserved.

Field name	Command-specific description
GPM_BLOCK_NUM	Matches the parameter passed in the request frame.
GPM_START_INDEX	Matches the parameter passed in the request frame.
GPM_NUM_BYTES	Number of bytes in the GPM_DATA field.
GPM_DATA	Bytes read from the specified GPM block.

Example:

To read 15 bytes of previously written data from flash block 22 of a target radio with serial number of 0x0013a200407402ac a READ packet should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 C0 04 00 0016 0000 000F 3B

Assuming all transmissions were successful and that flash block 22 was previously written with incrementing data, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76

7E 0029 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 84 00 0016 0000 000F 0102030405060708090A0B0C0D0E0F C3

- **FIRMWARE_VERIFY (0x05) and FIRMWARE_VERIFY_AND_INSTALL(0x06):** The FIRMWARE_VERIFY and FIRMWARE_VERIFY_AND_INSTALL commands are used to update module firmware remotely. Remote firmware upgrades are covered in detail in the next section. These commands check if the General Purpose Memory contains a valid over-the-air update file. For the FIRMWARE_VERIFY_AND_INSTALL command, if the GPM contains a valid firmware image, then the module resets and begins using the new firmware.

Field name	Command-specific description
GPM_CMD_ID	Set to FIRMWARE_VERIFY (0x05) or FIRMWARE_VERIFY_AND_INSTALL (0x06).
GPM_OPTIONS	Set this field to 0. At present, there are no defined options for this command.
GPM_BLOCK_NUM	Set to 0. This field is unused for this command.
GPM_START_INDEX	Set to 0. This field is unused for this command.
GPM_NUM_BYTES	Set to 0. This field is unused for this command.
GPM_DATA	This field is unused for this command.

- **FIRMWARE_VERIFY _RESPONSE (0x85):** When a FIRMWARE_VERIFY command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Set to FIRMWARE_VERIFY_RESPONSE (0x85).
GPM_STATUS	A one (1) in the least-significant bit indicates the GPM does not contain a valid firmware image. A zero (0) in the least-significant bit indicates the GPM does contain a valid firmware image. All other bits are reserved.

Field name	Command-specific description
GPM_BLOCK_NUM	Set to 0. This field is unused for this command.
GPM_START_INDEX	Set to 0. This field is unused for this command.
GPM_NUM_BYTES	Set to 0. This field is unused for this command.
GPM_DATA	This field is unused for this command.

- **FIRMWARE_VERIFY _AND_INSTALL_RESPONSE (0x86):** When a FIRMWARE_VERIFY_AND_INSTALL command request has been unicast to a node, that node will send a response in the following format to the source endpoint specified in the requesting frame only if the GPM memory does not contain a valid image. If the image is valid, the module will reset and begin using the new firmware.

Field name	Command-specific description
GPM_CMD_ID	Set to FIRMWARE_VERIFY_AND_INSTALL_RESPONSE (0x86).
GPM_STATUS	A one (1) in the least-significant bit indicates the GPM does not contain a valid firmware image. All other bits are reserved.
GPM_BLOCK_NUM	Set to 0. This field is unused for this command.
GPM_START_INDEX	Set to 0. This field is unused for this command.
GPM_NUM_BYTES	Set to 0. This field is unused for this command.
GPM_DATA	This field is unused for this command.

Example:

To verify a firmware image previously loaded into the GPM on a target radio with serial number of 0x0013a200407402ac a FIRMWARE_VERIFY packet should be formatted as follows (spaces added to delineate fields):

7E 001C 11 01 0013A200407402AC FFFE E6 E6 0023 C105 00 00 05 00 0000 0000 0000 1F

Assuming all transmissions were successful and that the firmware image previously loaded into the GPM is valid, the following API packets would be output the source node's serial interface:

7E 0007 8B 01 FFFE 00 00 00 76

7E 001A 91 0013A200407402AC FFFE E6 E6 0023 C105 C1 85 00 0000 0000 0000 5F

Working with flash memory

When working with the General Purpose Memory, observe the following limitations:

- Flash memory write operations are only capable of changing binary 1s to binary 0s. Only the erase operation can change binary 0s to binary 1s. For this reason, you should erase a flash block before performing a write operation.
- When performing an erase operation, you must erase the entire flash memory block—you cannot erase parts of a flash memory block.
- Flash memory has a limited lifetime. The flash memory on which the GPM is based is rated at 20,000 erase cycles before failure. Take care to ensure that the frequency of erase/write operations allows for the desired product lifetime. Digi's warranty does not cover products that have exceeded the allowed number of erase cycles.

- Over-the-air firmware upgrades erase the entire GPM. Any user data stored in the GPM will be lost during an over-the-air upgrade.

Over-the-air firmware upgrades

XLR PRO modules provide two methods of updating the firmware on the module:

- Local firmware update via XCTU using the front-panel serial port interface.
- Over-the-air firmware update using the RF interface.

The over-the-air firmware upgrade method provides a robust and versatile technique which can be tailored to many different networks and applications, with minimum disruption of normal network operations.

There are three phases of the over-the-air upgrade process: distributing the new application, verifying the new application, and installing the new application. In the following section, the node to be upgraded is referred to as the target node. The node providing the update information is referred to as the source node. In most applications, the source node is locally attached to a PC running update software.

Distributing the new application

The first phase of performing an over-the-air upgrade on a module is transferring the new firmware file to the target node. The new firmware image should be loaded in the target node's GPM prior to installation. XLR PRO modules use an encrypted binary (.ebin) file for both serial and over-the-air firmware upgrades. The firmware files are available on the Digi support website.

The contents of the .ebin file should be sent to the target radio using general purpose memory WRITE commands. The entire GPM should be erased prior to beginning an upload of an .ebin file. The contents of the .ebin file should be stored in order in the appropriate GPM memory blocks. The number of bytes that are sent in an individual GPM WRITE frame is flexible and can be catered to the user application.

Example:

XLR PRO firmware version 8060 has an .ebin file of 55,141 bytes in length. Based on network traffic, it was determined that sending a 128-byte packet every 30 seconds minimized network disruption. For this reason, the .ebin should be divided and addressed as follows:

GPM_BLOCK_NUM	GPM_START_INDEX	GPM_NUM_BYTES	.ebin bytes
0	0	128	0 to 127
0	128	128	128 to 255
0	256	128	256 to 383
0	384	128	384 to 511
1	0	128	512 to 639
1	128	128	640 to 767
-	-	-	-
-	-	-	-
-	-	-	-
107	0		54784 to 54911

GPM_BLOCK_NUM	GPM_START_INDEX	GPM_NUM_BYTES	.ebin bytes
107	128		54912 to 55039
107	256	101	55040 to 55140

Verifying the new application

For an uploaded application to function correctly every single byte from the .ebin file must be properly transferred to the GPM. To guarantee that this is the case GPM VERIFY functions exist to ensure that all bytes are properly in place. The FIRMWARE_VERIFY function reports whether or not the uploaded data is valid. The FIRMWARE_VERIFY_AND_INSTALL command will report if the uploaded data is invalid. If the data is valid it will begin installing the application. No installation will take place on invalid data.

Installing the application

When the entire .ebin file has been uploaded to the GPM of the target node a FIRMWARE_VERIFY_AND_INSTALL command can be issued. Once the target receives the command it will verify the .ebin file loaded in the GPM. If it is found to be valid then the module will install the new firmware. This installation process can take up to 8 seconds. During the installation the module will be unresponsive to both serial and RF communication. To complete the installation the target module will reset. AT parameter settings which have not been written to flash (using the WR command) will be lost.

Keep in mind

- The firmware upgrade process requires that the module resets itself. Because of this reset parameters which have not been written to flash will be lost after the reset. To avoid this, write all parameters with the WR command before doing a firmware upgrade.
- Because explicit API Tx frames can be addressed to a local node (accessible via the SPI or UART) or a remote node (accessible over the RF port) the same process can be used to update firmware on a module in either case.

Configuring the XLR PRO using XCTU

Digi International XCTU is a free multi-platform application that enables developers to interact with Digi RF products through a simple-to-use graphical interface. It includes tools that make it easy to set up, configure, and test Digi RF products.

Download and install XCTU

To download and install XCTU:

1. Go to www.digi.com/xctu.
2. Launch the XCTU Next Gen installer and follow the prompts on the installation screens.

Note For XLR PRO support, make sure you install Next Generation XCTU version 6.1.2 or later.

Connect XLR PRO to your PC

To connect an XLR PRO to your PC:

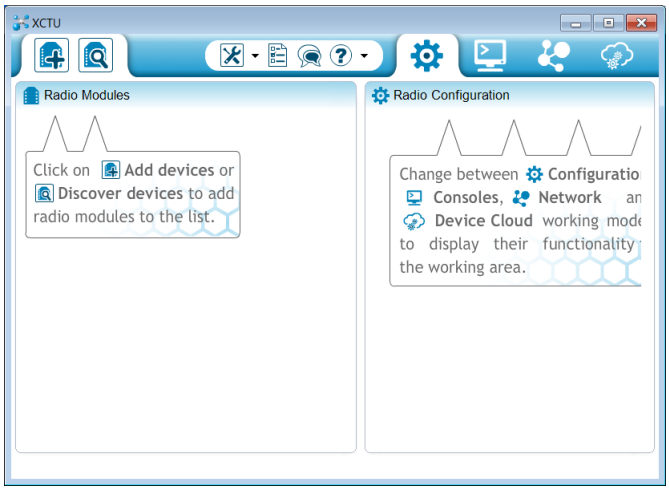
1. Connect the XLR PRO serial port to an available USB port on your PC using a network cable, the RJ45/DB9F adapter, and the USB to RS-232 serial converter:

2. Power on the XLR PRO.

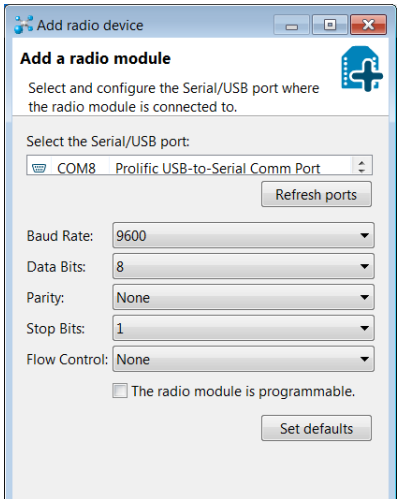
Launch XCTU and add the XLR PRO

To launch XCTU and add the XLR PRO:

1 Double-click on the XCTU program icon. The XCTU main menu appears.



2 Click the Add a radio icon . The Add a radio device dialog appears.



3 Provide connection information for the XLR PRO:

Select the Serial/USB port: Select the USB port connected to the XLR PRO.

Baud Rate: Select the default, **9600**.

Data Bits: Select the default, **8**.

Parity: Select the default, **None**.

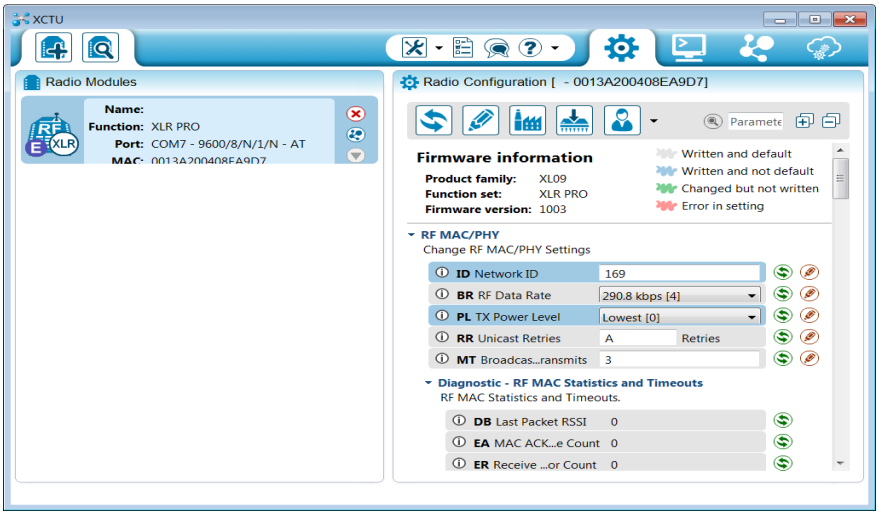
Stop Bits: Select the default **1**.

Flow Control: Select the default, **None**.

The radio module is programmable: Keep the default, unselected.

4 Click **Finish**. XCTU connects to the XLR PRO and displays the device in the list of radios.

5 Click on the XLR PRO to display current properties and configure parameters in the right-hand pane.



Configure parameters using XCTU

All of the XLR PRO parameter values are displayed in the XCTU configuration pane. For a complete list of all parameters, see XLR PRO AT command reference tables on page 35.

To change a configuration parameter:

1 Locate the parameter in the XCTU configuration display.

Note Use the Search function in the upper right corner to quickly locate a parameter.

2 Select a new value for the parameter.

3 If you want to permanently change the parameter value:

- To save an individual parameter value, click on the Write icon to the right of the parameter.
- To save all parameter settings, click on the Write icon at the top of the XCTU configuration pane.

Determine or assign an IP address

Determine the DHCP address

By default, the XLR PRO gets an IP address automatically assigned using DHCP. To have an IP address automatically assigned, connect the XLR PRO to the network via either Ethernet port on the XLR PRO front panel. An IP address should be automatically assigned within one minute. If a DHCP server hasn't assigned an address within that time period, the XLR PRO assigns an address using the following format:

169.254.**a**.**b**

where **a** is the decimal value of the 2nd to last byte of the XLR PRO MAC address and **b** is the last byte of the XLR PRO MAC address. If the assigned address is already in use, the address is incremented until a free address is found.

Once an address is assigned, you can read the address using the **MY** parameter.

Assign a static address

To assign a static address to an XLR PRO, set the following parameters:

- **MA (addressing mode):** Set the addressing mode to 1 (static).
- **MY (XLR IP address):** Enter the IP address for the XLR PRO.
- **MK (XLR subnet mask):** Enter the subnet mask for the XLR PRO.
- **GW (XLR gateway address):** Enter the default gateway address for the XLR PRO.

After setting the parameters, write the new parameter values to save the settings.

Update firmware with XCTU

Note To update XLR PRO firmware, you need XCTU version 6.1.2 (or above).

To update firmware using XCTU:

1. Launch XCTU.
2. Click **Add devices** or **Discover devices** to add the XLR PRO to the list of radios.
   a. Select the COM port to which the XLR PRO serial port is connected.
   b. Select the baud rate (9600 8-N-1 by default)
   c. Switch baud rate to 115200bps to reduce the time required to update the firmware.
   d. Close the COM port and reopen it at the new baud rate.
3. Select the radio configuration tab.
4. Click the icon to download the firmware.
5. Select the desired firmware, function set, and firmware version.
6. Click **Finish** and then the **Yes**.

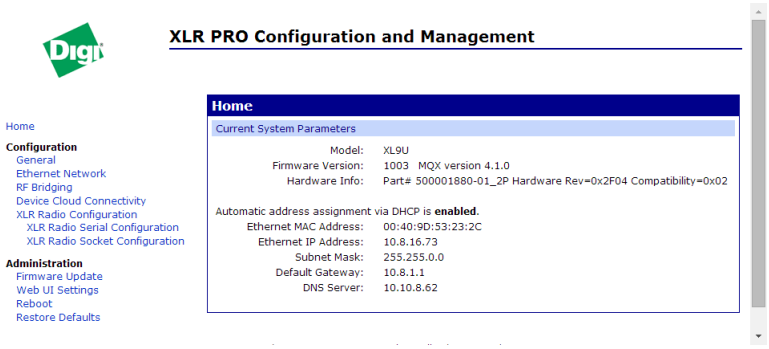
Configuring XLR PRO using the web configuration interface

Access the XLR PRO web configuration interface

Accessing the XLR PRO web configuration interface requires the IP address of the XLR PRO. If your XLR PRO is configured for DHCP addressing (the default), connect the XLR PRO to the network and wait approximately one minute for DHCP to assign an address and then use XCTU to view the assigned address. See Configuring the XLR PRO using XCTU on page 78 for details on using XCTU.

To access the XLR PRO web configuration interface:

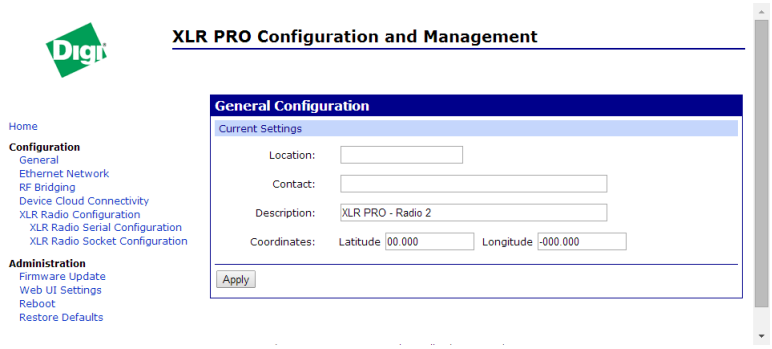
1. Open a browser and go to the XLR PRO IP address.
2. When you are prompted to do so, enter the username and password for the web configuration interface. The default username is **admin** and the default password is **password**. The Device Configuration Home page appears.



Set general options

To set XLR PRO general options:

1 From the XLR PRO Configuration and Management page, select **General**. The General Configuration page appears.



2 Enter the following:

Location: Enter a location for the XLR PRO.

Contact: Enter contact information for the XLR PRO.

Description: Enter a description for the XLR PRO.

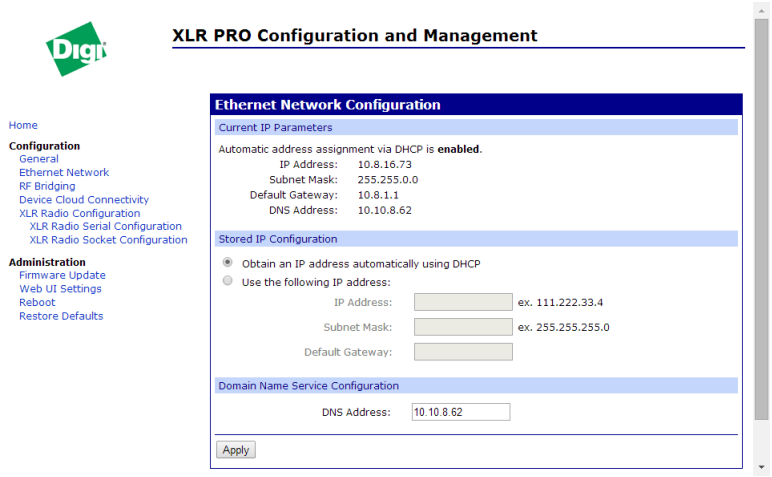
Coordinates: Enter latitude and longitude for the XLR PRO.

3 Click **Apply**.

Set Ethernet network options

Use Ethernet Network options to configure Ethernet options for the XLR PRO. Ethernet Network options are used for Ethernet (socket) mode only.

1 From the XLR PRO Configuration and Management page, select **Ethernet Network**. The Ethernet Network Configuration page appears.



2 Enable or disable DHCP:

- To enable DHCP, select **Obtain an IP address automatically using DHCP**.
- To disable DHCP—that is, use static addressing, select **Use the following IP address**.

3 For static addressing only, enter the following:

IP address: Enter an IP address to assign to the XLR PRO in IPv4 format.

Subnet Mask: Enter the subnet mask to use for the XLR PRO in IPv4 format.

Default Gateway: Enter the default gateway for the XLR PRO in IPv4 format.

4 For DHCP or static addressing, you can specify the Domain Name Server address:

DNS Address: For DHCP addressing, enter a specific DNS address to override the DNS address learned from the DHCP server; for static addressing, enter the DNS address to use. Enter the address in IPv4 format.

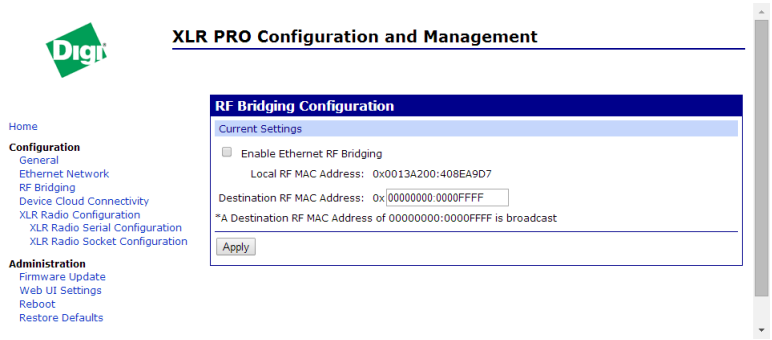
Note The default DNS address is 0.0.0.0. If you use the default DNS address and static mode is configured, **OpenDns** servers are used.

5 Click **Apply**.

Set RF bridging options

By default, bridging is disabled. To enable bridging and set a destination address:

1 From the XLR PRO Configuration and Management page, select **RF Bridging**. The RF Bridging Configuration page appears.



2 Enter the following:

Enable Ethernet RF Bridging: Select this option to enable bridging mode; to disable bridging, deselect this option.

Destination RF MAC Address: If bridging is enabled, enter a destination MAC RF address. A destination address of 00000000:000FFFF is broadcast.

3 Click **Apply**.

Set Device Cloud options

By default, the XLR PRO is enabled for Device Cloud management. To change Device Cloud options:

1 From the XLR PRO Configuration and Management page, select **Device Cloud Connectivity**. The Device Cloud Configuration page appears.

2 Select the following:

Enable Device Cloud Connectivity: Select this option to enable Device Cloud connectivity; deselect this option to disable Device Cloud connectivity.

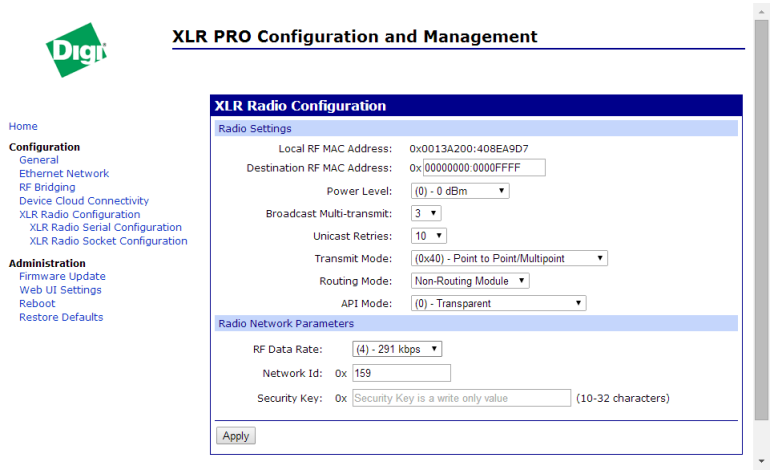
Device Cloud Server: If Device Cloud connectivity is enabled, enter the name of the Device Cloud server. The default is **login.etherios.com**, which is the name of the US server. For the Europe server, enter **login.etheiros.co.uk**.

3 Click **Apply**.

Set XLR radio options

To set XLR radio options:

1 From the XLR PRO Configuration and Management page, select **XLR Radio Configuration**. The XLR Radio Configuration page appears.



2 Set radio options:

Local RF MAC Address: Display only. This address matches the XLR PRO serial number shown on the XLR PRO label.

Destination RF MAC Address: Enter the Destination RF MAC address.

Power Level: Select a power level from the drop-down list. The default is 4.

Broadcast Multi-transmit: Enter the number of broadcast retransmissions, from 0 to 8 retransmissions. If you enter 0, broadcasts are transmitted only once; if you enter 8, a given broadcast can be retransmitted up to 8 times. The default is 3.

Unicast Retries: Enter the maximum number of MAC RF packet delivery attempts of unicast packets. If you enter a non-zero value, then unicast packets sent from the radio request an acknowledgment. The default is 10.

Transmit Mode: Select the transmit mode, either Point to Point/Multi-point or Repeater/Directed Broadcast. The default is Point to Point/Multi-point.

Routing Mode: Select the routing mode, either Standard or Non-Routing. The default is Non-Routing.

API Mode: Select the API mode: 0 for Transparent mode; 1 for API without Escapes; or 2 for API with Escapes. The default is 0 (Transparent).

3 Set radio network parameters:

RF Data Rate: Select an RF Data Rate from the drop-down list. The default is 4.

Network ID: Enter a network ID for the XLR PRO. The default is 7FFF.

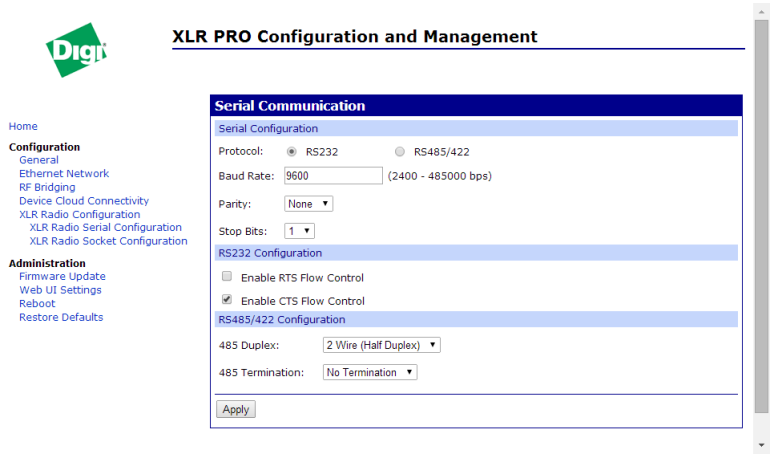
Security Key: Enter an AES encryption key for RF packets, a 128-bit key. By default, encryption is always used, but the key is never displayed. If you enter a new encryption key, the key must match on all devices on the network.

4 Click **Apply**.

Set XLR radio serial options

To set XLR radio serial configuration options:

1 From the XLR PRO Configuration and Management page, select **XLR Radio Serial Configuration**. The Serial Configuration page appears.



2 Set serial configuration options:

Protocol: Select RS-232 or RS-485/RS-422.

Baud Rate: Enter a baud rate:

- For RS-232: Enter a baud rate from 2400 through 460800. The default is 9600.
- For RS-484/RS-422: Enter a baud rate from 2400 to 921000. The default is 9600.

Parity: Select the parity: None, Even, or Odd. The default is None.

Stop Bits: Select the number of stop bits. The default is 1.

3 For the RS-232 protocol only:

Enable RTS Flow Control: Enable this option for RTS Flow Control. By default, RTS Flow Control is disabled.

Enable CTS Flow Control: Enable this option for CTS Flow Control. By default, CTS Flow Control is enabled.

4 For the RS-485/RS-422 protocol only:

485 Duplex: Select 2 Wire (Half Duplex) or 4 Wire (Full Duplex). The default is 2 Wire.

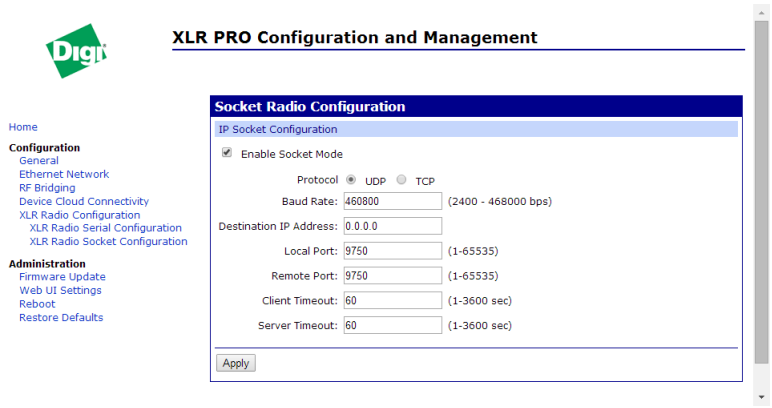
485 Termination: Select Termination or No Termination. The default is No Termination.

5 Click **Apply**.

Set XLR radio socket (Ethernet) options

To set XLR radio socket (Ethernet) configuration options:

1 From the XLR PRO Configuration and Management page, select **XLR Radio Socket Configuration**. The Socket Radio Configuration page appears.



2 Enable or disable socket mode:

- To enable socket mode, select **Enable Socket Mode**.
- To disable socket mode, deselect **Enable Socket Mode**.

3 If socket mode is enabled, enter the following:

Protocol: Select UDP or TCP.

Baud Rate: Enter the baud rate for socket mode, from 2400 through 460800. The default is 460800.

Destination IP Address: Enter the destination IP address in IPv4 format. This address is only used when traffic is initiated from the XLR PRO. When traffic is initiated from the Ethernet interface, the Destination IP address is the address of the external device.

Local Port: Enter the local listening port. The default is 9750.

Remote Port: Enter the remote listening port. The default is 9750.

Client Timeout: Enter the number of seconds the client waits without any traffic before closing the connection. The default is 60 seconds.

Server Timeout: Enter the number of seconds the server waits without traffic before closing the connection. The default is 60 seconds.

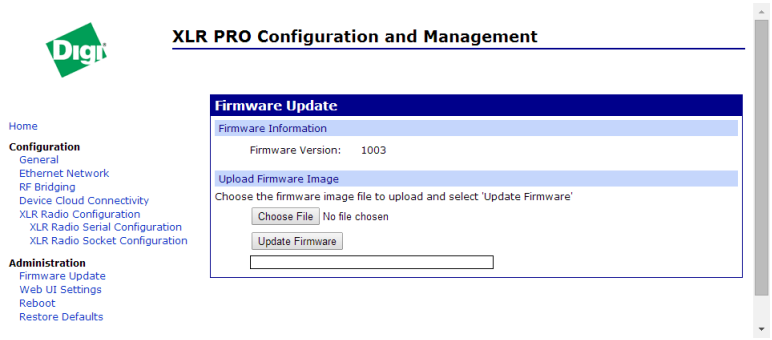
4 Click **Apply**.

Update firmware

Use the Firmware Update option on the XLR PRO Device Configuration Home page to update the XLR PRO firmware. To update firmware:

1 Go to the XLR PRO product page: www.digi.com/xlrpro and download the latest XLR PRO firmware.

2 From the XLR PRO Configuration and Management page, select **Firmware Update**. The Firmware Update page appears.



3 Click **Choose File**.

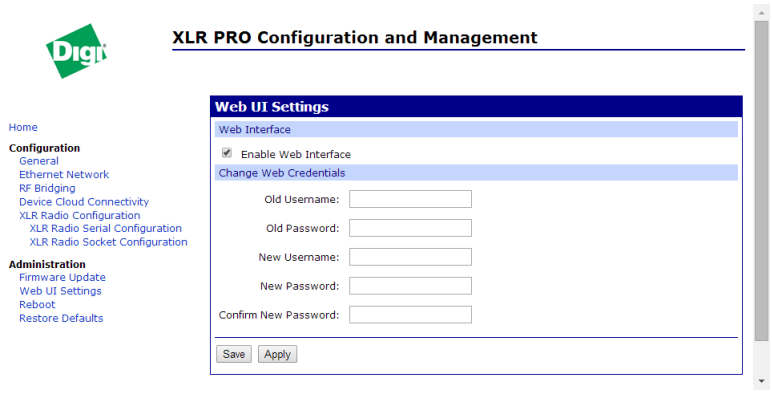
4 Browse to and select an XLR PRO firmware file. XLR PRO firmware files have the **.ebin** extension.

5 After selecting a firmware file, click **Update Firmware**. XLR PRO firmware is updated and the XLR PRO resets. (The amount of time needed to update the XLR PRO depends on the speed of your network.) Approximately 30 seconds after the XLR PRO resets, the XLR PRO is operational.

Set Web UI options

By default, configuration and management via the web configuration interface is enabled. The default username is **admin** and the default password is **password**. To change web configuration options:

1 From the XLR PRO Configuration and Management page, select **Web UI Settings**. The Web UI Settings page appears.



2 Enable or disable configuration and management via the Web UI:

- To enable the Web UI, select **Enable Web Interface.**
- To disable the Web UI, deselect **Enable Web Interface**.

3 If the Web interface is enabled, enter the following:

Old Username: Enter the current username for the Web UI.

Old Password: Enter the current password for the Web UI.

New Username: Enter a new username for the Web UI.

New Password: Enter a new password for the Web UI.

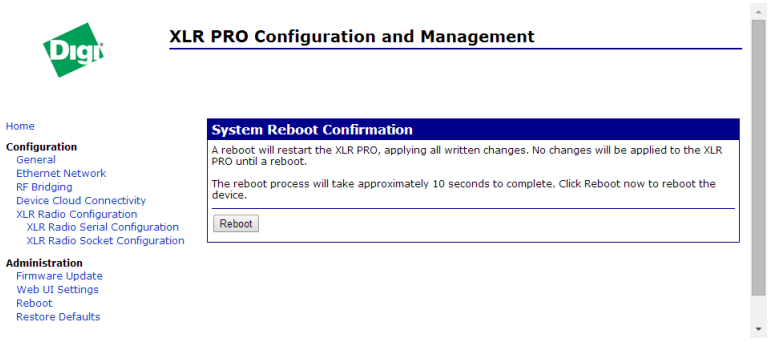
Confirm New Password: Enter the new password for the Web UI again.

4 Click **Apply**.

Reboot an XLR PRO

Use the Reboot option to reboot the XLR PRO. To reboot:

1 From the XLR PRO Configuration and Management page, select **Reboot**. The System Reboot Confirmation page appears.

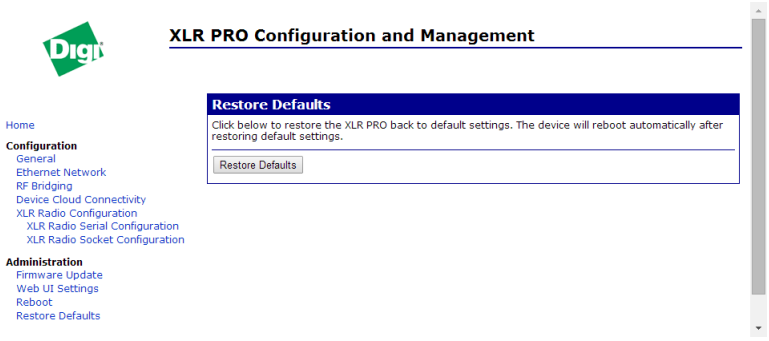


2 Click **Reboot**.

Restore factory default settings

To restore the XLR PRO factory default settings:

1 From the XLR PRO Configuration and Management page, select **Restore Defaults**. The Restore Defaults page appears.



2 Click **Restore Defaults**.

Configuring XLR PRO using Device Cloud

Device Cloud by Etherios™ is an on-demand service with no infrastructure requirements. Remote devices and enterprise business applications connect to Device Cloud through standards-based web services. This section describes how to configure and manage an XLR PRO using Device Cloud.

For detailed information on using Device Cloud, refer to the *Device Cloud User Guide*, available via the **Documentation** tab in Device Cloud.

Create a Device Cloud account

Before you can manage an XLR PRO with Device Cloud, you must create a Device Cloud account. To create a Device Cloud account:

1. Go to www.etherios.com.
2. In the top menu bar, click **Free Trial**.
3. Follow the online instructions to complete account registration. You can upgrade your trial account to a Premier account at any time.

Note For initial XLR PRO deployment, you can use a 30-day free trial account. When you're ready to deploy multiple XLR PROs in the field, upgrade to a Premiere account to access additional Device Cloud features.

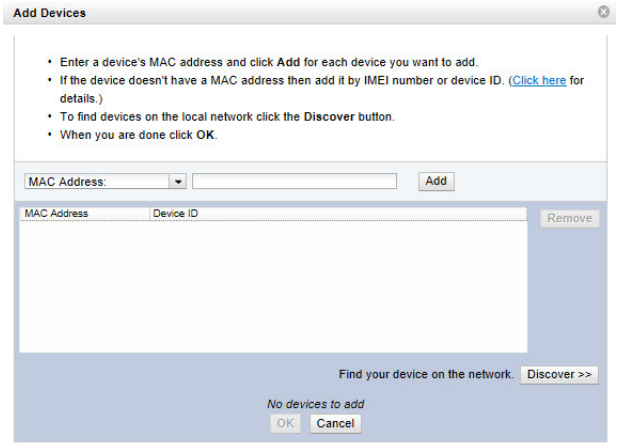
Get the MAC address for the XLR PRO

Before adding an XLR PRO to your Device Cloud account inventory, you need to determine the MAC address for the device. You can view the MAC address using either XCTU or the Web UI. For information on using XCTU, see Configuring the XLR PRO using XCTU on page 78; for information on using the Web UI, see Configuring XLR PRO using the web configuration interface on page 82.

Add a Digi XLR PRO to Device Cloud

To add an XLR PRO to your Device Cloud account inventory, follow these steps:

1. Go to login.etherios.com.
2. Log in to your account
3. Click **Device Management > Devices**.
4. Click **Add Devices**. The Add Devices dialog appears.



Note You can use the **Discover** button to discover an XLR PRO device if the XLR PRO is connected to your local network.

5. Select **Mac Address**, and enter the MAC address of the XLR PRO you want to add. Enter the MAC address in uppercase ASCII hexadecimal.
6. Click **Add** to add the device. The XLR PRO is added to your inventory.
7. Click **OK** to close the Add Devices dialog and return to the Devices view.

Update firmware with Device Cloud

1. After logging on to Device Cloud, select the Device Management tab
2. Right click on the device to update. (Device Type should be XLR PRO and the Device ID will match the MAC address of the XLR PRO to be updated.)
3. In the drop-down box, select Firmware->Update Firmware.
4. Click on the Browse button to select the **.ebin** file that is the firmware image to be loaded.
5. Click on Update Firmware. A progress bar will be displayed. This load takes about 100 seconds
6. Fifteen seconds later, the power LED will turn on.

About 30 seconds after that the new firmware will be fully operational and running.

Configure XLR PRO with Device Cloud

XLR PRO configuration includes the following options:

- Bridging (see Configure bridging options on page 93)

- Device Cloud (see Configure Device Cloud options on page 94)
- HTTP server (see Configure HTTP server options on page 95)
- IP (see Configure IP options on page 96)
- IP to XLR (see Configure IP to XLR options on page 97)
- Serial to XLR (see Configure serial to XLR options on page 98)
- System (see Configure system options on page 99)
- XLR radio (see Configure XLR Module options on page 100)

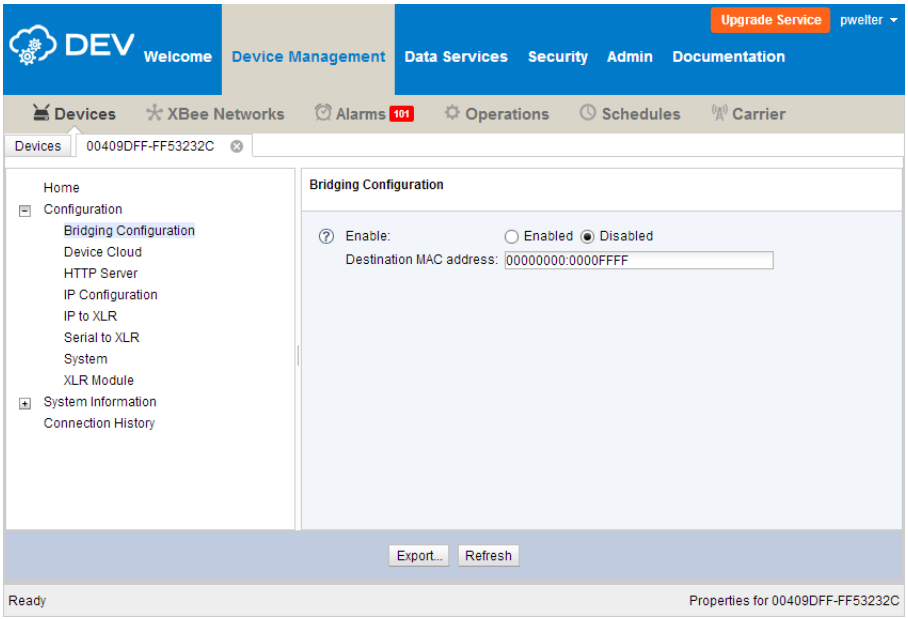
Configure bridging options

By default, bridging is disabled. To change the XLR PRO bridging setting:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you want to configure.

3 Click **Configuration > Bridging Configuration**.



4 Set options for bridging:

Enabled/Disabled: Select **Enabled** to enable bridging or select **Disabled** to disable bridging. The default is disabled.

Destination MAC address: If bridging is enabled, enter the destination RF MAC address for the destination XLR PRO.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

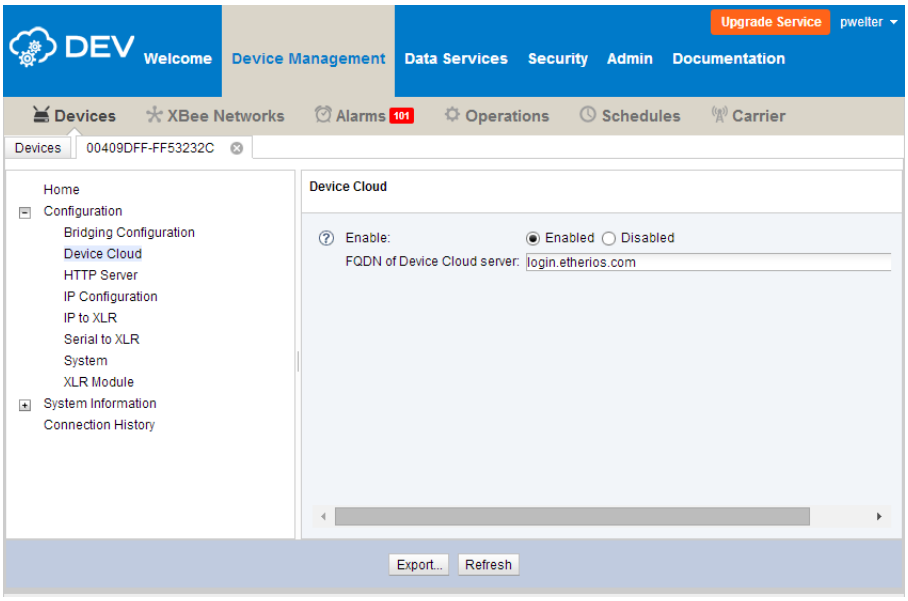
Configure Device Cloud options

By default, the XLR PRO is configured to enabled Device Cloud management. To change Device Cloud options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you need to configure.

3 Click **Configuration > Device Cloud**.



4 Set options for Device Cloud:

Enabled/Disabled: Select **Enabled** to enable Device Cloud management or select **Disabled** to disable Device Cloud management. The default is enabled.

FQDN of Device Cloud server: If Device Cloud management is enabled, enter the FQDN of the Device Cloud server. The default FQDN is the US server: **login.etherios.com**.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

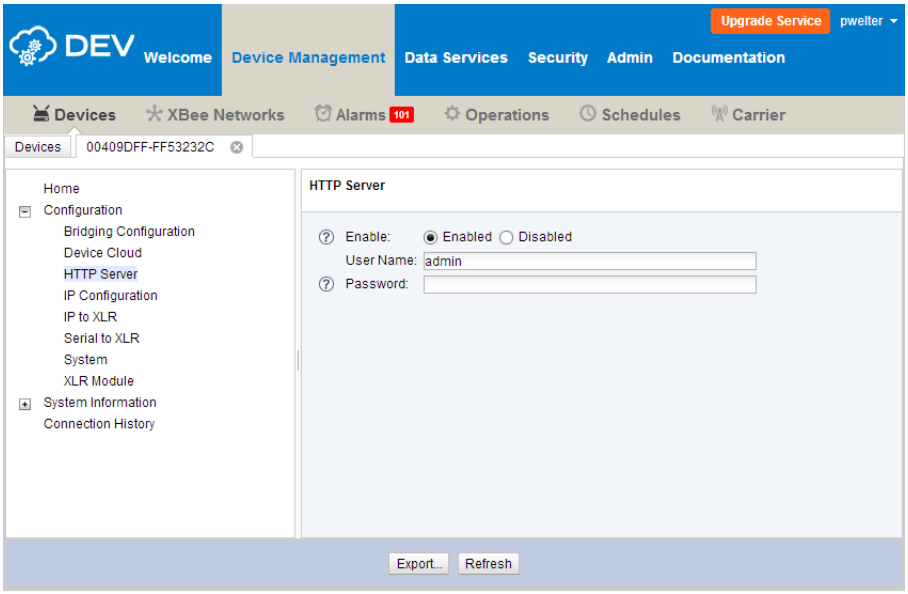
Configure HTTP server options

By default, the XLR PRO HTTP server interface is enabled. To change HTTP server options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you need to configure.

3 Click **Configuration > HTTP Server**.



4 Set HTTP server options:

Enabled/Disabled: Select **Enabled** to enable the HTTP server or select **Disabled** to disable the HTTP server. The default is enabled.

User Name/Password: If the HTTP server is enabled, you can set a user name and password to control access to the server. The default HTTP server user name is **admin**; and the default password is **password**. To remove a user name and password, blank out the input fields.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

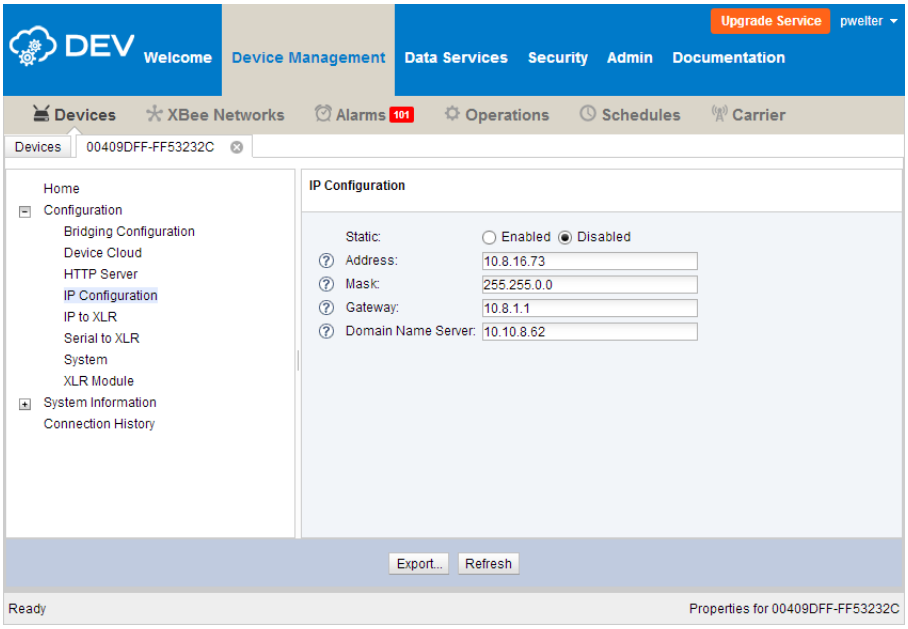
Configure IP options

By default, the XLR PRO is configured to use DHCP addressing. To change the IP addressing options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you need to configure.

3 Click **Configuration > IP Configuration**.



4 Set IP configuration options:

DHCP: Select **Enabled** to enable DHCP or select **Disabled** for static addressing.

Address: If DHCP is disabled, enter the IP address for the XLR PRO in IPv4 format.

Mask: If DHCP is disabled, enter the IP subnet mask for the XLR PRO in IPv4 format.

Gateway: If DHCP is disabled, enter the IP address of the network gateway in IPv4 format.

Domain Name Server: If DHCP is disabled, enter the IP address of the DNS server in IPv4 format to override the DNS address learned from the DHCP server. For static addressing, enter the IP address of the DNS server in IPv4 format.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

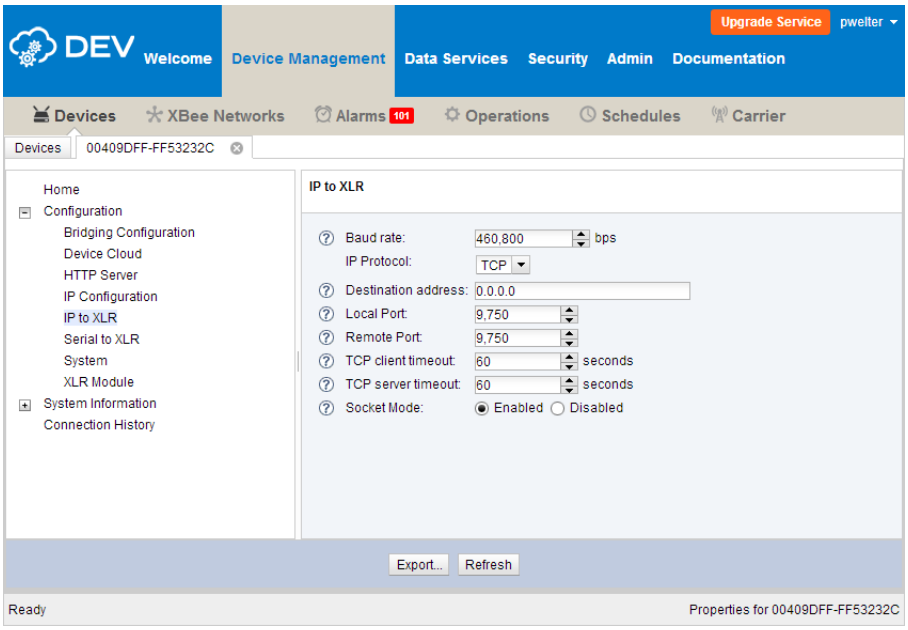
Configure IP to XLR options

The IP to XLR options determine how data is transmitted from the external Ethernet port to the XLR radio. To change IP to XLR options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you need to configure.

3 Click **Configuration > IP to XLR**.



4 Set IP to XLR options:

Baud rate: Select the baud rate for IP to XLR traffic. The default is 460800.

IP Protocol: Select the IP protocol to use: **TCP** or **UDP**. The default is TCP.

Destination address: Enter the IPv4 destination address. The default is 0.0.0.0.

Local Port: Enter the listening IP port number for UDP and TCP operations. The default is 9750.

Remote Port: Enter the destination IP port number for UDB and TCP operations. The default is 9750.

TCP client timeout: Enter the number of seconds the TCP client socket should remain open without any traffic. The default is 60 seconds.

TCP server timeout: Enter the number of seconds the TCP server socket should remain open without any traffic. The default is 60 seconds.

Socket Mode: Socket mode is enabled or disabled depending on what cables are plugged in to the XLR PRO. To prevent socket mode regardless of cables, select **Disabled**.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

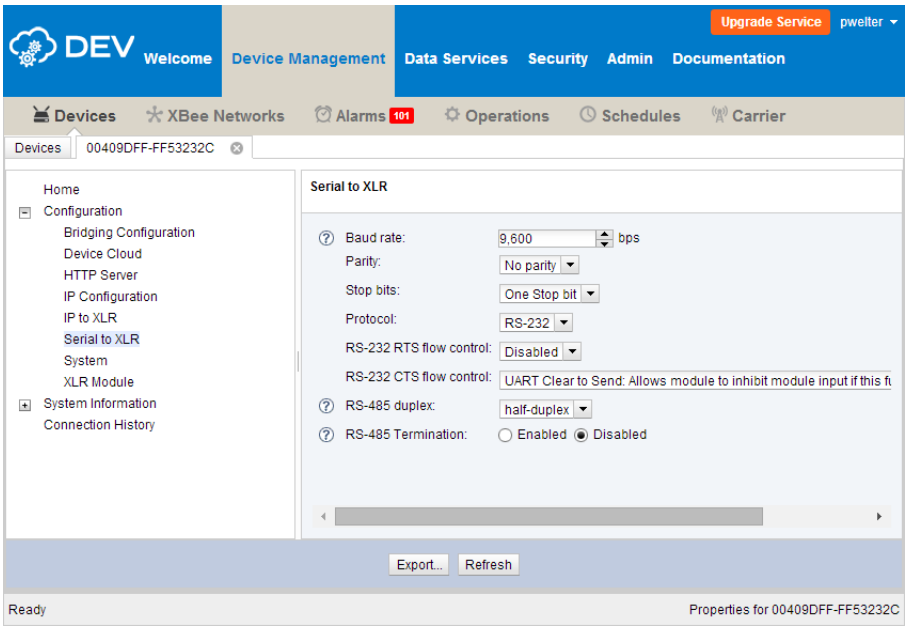
Configure serial to XLR options

The IP to XLR options determine how data is transmitted from the external Ethernet port to the XLR radio. To change IP to XLR options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you want to configure.

3 Click **Configuration > Serial to XLR**.



4 Set Serial to XLR options:

Baud rate: Select the baud rate for Serial to XLR traffic. The default is 9600.

Parity: Select the parity: no parity, even parity, or odd parity. The default is no parity.

Stop bits: Select the stop bits to use: one or two. The default is one stop bit.

Protocol: Select the protocol: RS-232 or RS-485/RS-422. The default is RS-232.

RS-232 RTS flow control: Select RTS flow control for RS-232: disabled or UART clear to send. The default is disabled.

RS-232 CTS flow control: Select CTS flow control for RS-232: disabled or UART clear to send. The default is UART clear to send.

RS-485 duplex: Select the duplex for RS-485: half-duplex or full-duplex. The default is half-duplex.

RS-458 Termination: Enable or disable RS-485 termination. The default is disabled.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

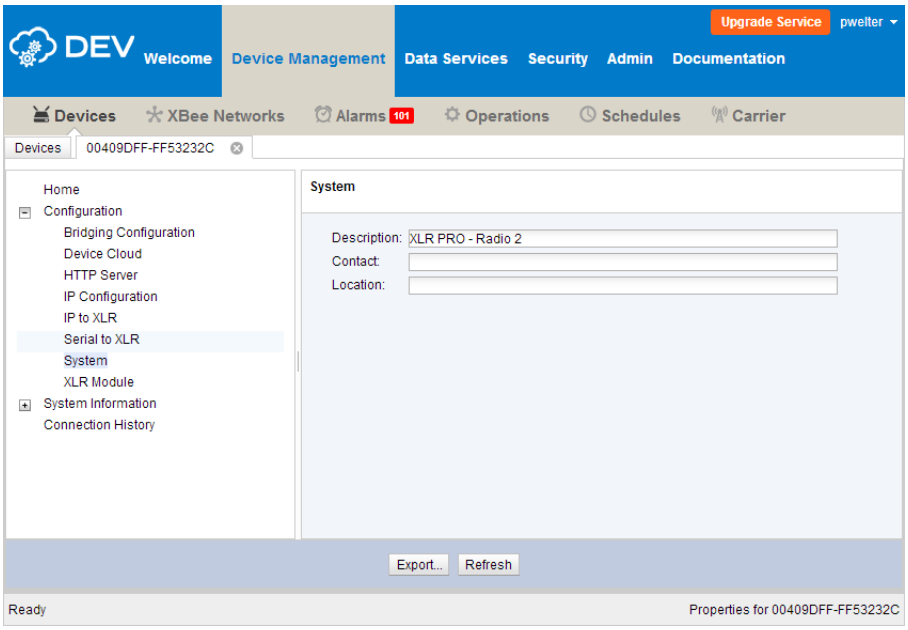
Configure system options

The IP to XLR options determine how data is transmitted from the external Ethernet port to the XLR radio. To change IP to XLR options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you want to configure.

3 Click **Configuration > System**.



4 Set system options:

Description: Enter a description for the XLR PRO.

Contact: Enter contact information for the XLR PRO.

Location: Enter a location for the XLR PRO.

5 Apply the configuration changes:

- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

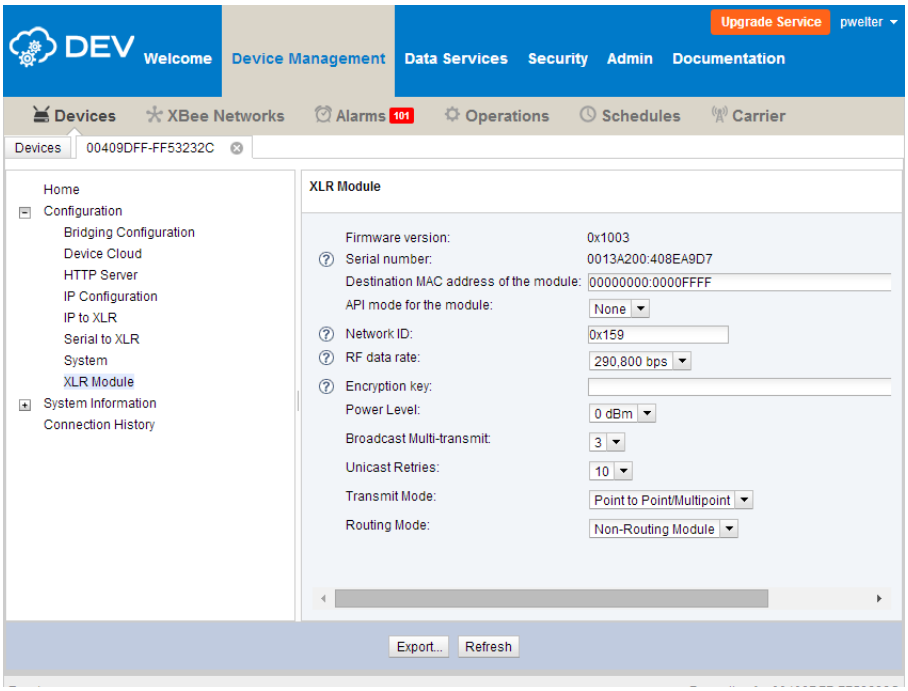
Configure XLR Module options

The IP to XLR options determine how data is transmitted from the external Ethernet port to the XLR radio. To change IP to XLR options:

1 Click **Device Management > Devices**.

2 Double-click on the XLR PRO you want to configure.

3 Click **Configuration > XLR Module**.



4 Set XLR Module options:

Firmware version: Displays the current firmware version. This value cannot be changed.

Serial number: Displays the serial number. This value cannot be changed.

Destination MAC address of the module: Enter the destination RF MAC address for the XLR PRO. The default is 00000000:0000FFFF.

API mode for the module: Select the API mode for the XLR PRO: none (transparent), API, or API with escapes. The default is none (transparent).

Network ID: Enter a network ID for the XLR PRO. The default is 7FFF. Only XLR PROs with matching network IDs can communicate with each other.

RF data rate: Select an RF Data Rate from the drop-down list. The default is 290,800 bps.

Encryption key: Enter an AES encryption key for RF packets, a 128-bit key. By default, encryption is always used, but the key is never displayed. If you enter a new encryption key, the key must match on all devices on the network.

Power Level: Select a power level from the drop-down list. The default is Max Power.

Broadcast Multi-transmit: Enter the number of broadcast retransmissions, from 0 to 8 retransmissions. If you enter 0, broadcasts are transmitted only once; if you enter 8, a given broadcast can be retransmitted up to 8 times. The default is 3.

Unicast Retries: Enter the maximum number of MAC RF packet delivery attempts of unicast packets. If you enter a non-zero value, then unicast packets sent from the radio request an acknowledgment. The default is 10.

Transmit Mode: Select the transmit mode, either Point to Point/Multi-point or Repeater/Directed Broadcast. The default is Point to Point/Multi-point.

Routing Mode: Select the routing mode, either Standard or Non-Routing. The default is Non-Routing.

5 Apply the configuration changes:

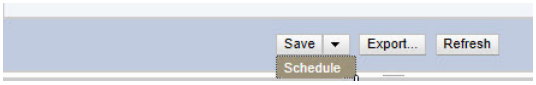
- If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
- If the XLR PRO is currently not connected to Device Cloud, use the Schedule option to schedule when to apply configuration changes to the device. See Schedule configuration changes on page 101.

Schedule configuration changes

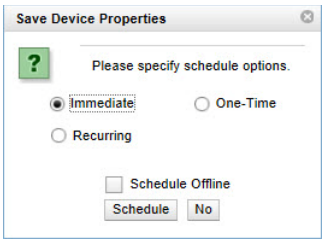
After changing one or more configuration options, you can immediately apply the changes to the XLR PRO device or schedule a time for applying the configuration changes.

To schedule a time for applying configuration changes:

1 On any configuration page, click the down arrow next to the **Save** button.



2 Click **Schedule**. The Save Device Properties dialog appears.



3 Enable the **Schedule Offline** checkbox to create a schedule for a device that is offline.

Note If a device becomes disconnected from Device Cloud for any reason during the execution of this command, Device Cloud retries the command when the device reconnects.

Troubleshooting

Serial interface issues

No data on serial port

To enable the XLR PRO serial port, the DTR line on your RS-232 cable must be active and connected. If the DTR line is not present, the XLR PRO uses socket mode instead of the serial port for serial communication.

Solution: If you do not have a DTR line on your serial cable or end device, explicitly disable socket mode by setting the ES parameter to 0.

RS-485/422 does not work

RS-232 is the default serial protocol used for the front panel serial port. In order to use RS-485/422, the 4E parameter must be set to 1. This will also disable IP socket mode, so it would be advisable to connect the XLR PRO to your Ethernet network if configuration changes need to be made.

Ethernet issues

LAN stopped working when XLR PRO was plugged in

With Ethernet RF bridging enabled, the XLR PRO acts like an Ethernet cable. If two XLR PRO radios are connected to the same LAN, a bridging loop could occur and cause a broadcast storm. Make sure that only one XLR PRO is connected to a LAN to avoid this.

General issues

All of the LEDs are lit on the XLR PRO

If all of the LEDs on the front panel are lit solid, this indicates that the XLR PRO is in a bootloader state and will not respond to any user input. This can happen if the reset button is held down upon startup, verify that the reset button is clear of obstructions and perform a power cycle.

Safety notices and certifications

Before installing and powering on the XLR PRO, read all instructions and keep these instructions in a safe place for future reference.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Do not attempt to repair the product. Any attempt to service or repair the unit by the user will void the product warranty.

The XLR PRO must be maintained by Digi or a Digi qualified technician only.

RF exposure statement

The XLR PRO, when used with approved antennas, complies with the FCC and IC certifications detailed in this section. For a list of antennas approved for use with XLR PRO, see XLR PRO approved antennas on page 106.

To comply with RF exposure limits established in the ANSI C95.1 standards, the distance between the antenna or antennas and the user should not be less than 25 cm (or 10 inches) for USA and 34 cm for Canada.

Class I, Division 2 (CID2) certification—USA and Canada

This equipment is suitable for use in Class I, Division 2, Groups A, B, C, and D hazardous locations or non-hazardous locations only.

This equipment is an open type device and is meant to be installed in an enclosure suitable for the environment, such that the equipment is accessible only with the use of a tool.

Note The USB port on the XLR PRO is not intended for use in Class I, Division 2 environments.

WARNING! Explosion hazard—Do not disconnect equipment while the circuit is live unless the area is known to be free of ignitable concentrations.

WARNING! Explosion hazard—Substitution of any components may impair suitability for Class I, Division 2.

WARNING! Explosion hazard—The area must be known to be non-hazardous before servicing/ replacing the unit and before installing the unit.

Cet équipement est adapté à une utilisation en Classe I, Division 2, Groupes A, B, C et D ou non dangereux uniquement.

Ces dispositifs sont des dispositifs de type ouvert qui doivent être installés dans un boîtier adapté à l'environnement et accessible seulement par l'intermédiaire d'un outil.

AVERTISSEMENT! RISQUE D'EXPLOSION—Ne pas débrancher l'équipement que l'alimentation est coupée ou que la zone est connue pour être non dangereux.

AVERTISSEMENT! RISQUE D'EXPLOSION—Remplacement de tous les composants peut altérer l'aptitude de Classe I, Division 2.

AVERTISSEMENT! RISQUE D'EXPLOSION—La zone doit être reconnu comme non dangereux avant l'entretien / remplacement de l'unité et avant l'installation.

FCC (United States) certification

FCC Part 15 Class B

Radio Frequency Interface (RFI) (FCC 15.105)

This device has been tested and found to comply with the limits for Class B digital devices pursuant to Part 15 Subpart B of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential environment. This equipment generates, uses, and can radiate radio frequency energy, and if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. 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 receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

Labeling requirements (FCC 15.19)

This device complies with Part 15 of 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.

Modifications (FCC 15.21)

Changes or modifications to this equipment not expressly approved by Digi may void the user's authority to operate this equipment.

Industry Canada (IC) certifications

This digital apparatus does not exceed the Class B limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le present appareil numerique n'emet pas de bruits radioelectriques depassant les limites applicables aux appareils numeriques de la class B prescrites dans le Reglement sur le brouillage radioelectrique edicte par le ministere des Communications du Canada.

RF Exposure

CAUTION! This equipment is approved for mobile and base station transmitting devices only. Antenna(s) used for this transmitter must be installed to provide a separation distance of at least 34 cm from all persons and must not be co-located or operating in conjunction with any other antenna or transmitter.

ATTENTION Cet équipement est approuvé pour les périphériques de transmission de mobile et station de base seulement. Antenne (s) utilisée pour ce transmetteur doit être installé pour fournir une distance de séparation d'au moins 34 cm de toutes les personnes et ne doivent pas être co-implantés ou utilisés dans conjointement avec une autre antenne ou émetteur.

XLR PRO approved antennas

Omni-directional antennas

Part Number	Type	Connector	Gain (dBi)	Application	Minimum Cable Loss or TX Power Reduction Required in dB
A09-F0	OMNI	RPN	0.0	Fixed	0
A09-F1	OMNI	RPN	1.0	Fixed	0
A09-F2	OMNI	RPN	2.1	Fixed	0
A09-F3	OMNI	RPN	3.1	Fixed	0
A09-F4	OMNI	RPN	4.1	Fixed	0
A09-F5	OMNI	RPN	5.1	Fixed	0
A09-F6	OMNI	RPN	6.1	Fixed	0.1
A09-F7	OMNI	RPN	7.1	Fixed	1.1
A09-F8	OMNI	RPN	8.1	Fixed	2.1
A09-W7	OMNI	RPN	7.1	Fixed	1.1
A09-F0	OMNI	RPSMA	0.0	Fixed	0
A09-F1	OMNI	RPSMA	1.0	Fixed	0
A09-F2	OMNI	RPSMA	2.1	Fixed	0
A09-F3	OMNI	RPSMA	3.1	Fixed	0
A09-F4	OMNI	RPSMA	4.1	Fixed	0
A09-F5	OMNI	RPSMA	5.1	Fixed	0
A09-F6	OMNI	RPSMA	6.1	Fixed	0.1
A09-F7	OMNI	RPSMA	7.1	Fixed	1.1

Part Number	Type	Connector	Gain (dBi)	Application	Minimum Cable Loss or TX Power Reduction Required in dB
A09-F8	OMNI	RPSMA	8.1	Fixed	2.1
A09-M7	OMNI	RPSMAF	7.2	Fixed	1.2
A09-W7SM	OMNI	RPSMA	7.1	Fixed	1.1
A09-F0TM	OMNI	RPTNC	0.0	Fixed	0
A09-F1TM	OMNI	RPTNC	1.0	Fixed	0
A09-F2TM	OMNI	RPTNC	2.1	Fixed	0
A09-F3TM	OMNI	RPTNC	3.1	Fixed	0
A09-F4TM	OMNI	RPTNC	4.1	Fixed	0
A09-F5TM	OMNI	RPTNC	5.1	Fixed	0
A09-F6TM	OMNI	RPTNC	6.1	Fixed	0.1
A09-F7TM	OMNI	RPTNC	7.1	Fixed	1.1
A09-F8TM	OMNI	RPTNC	8.1	Fixed	2.1
A09-W7TM	OMNI	RPTNC	7.1	Fixed	1.1
A09-HSM-7	OMNI	RPSMA	3.0	Fixed/Mobile	0
A09-HASM-675	OMNI	RPSMA	2.1	Fixed/Mobile	0
A09-HABMM-P61	OMNI	MMCX	2.1	Fixed/Mobile	0
A09-HABMM-6-P61	OMNI	MMCX	2.1	Fixed/Mobile	0
A09-HBMM-P61	OMNI	MMCX	2.1	Fixed/Mobile	0
A09-HRSM	OMNI	RPSMA	2.1	Fixed	0
A09-HASM-7	OMNI	RPSMA	2.1	Fixed	0
A09-HG	OMNI	RPSMA	2.1	Fixed	0
A09-HATM	OMNI	RPTNC	2.1	Fixed	0
A09-HATM-10	OMNI	RPTNC	2.1	Fixed/Mobile	0
A09-H	OMNI	RPSMA	2.1	Fixed	0
A09-HBMMP61	OMNI	MMCX	2.1	Fixed/Mobile	0
A09-QBMMP61	OMNI	MMCX	1.9	Fixed/Mobile	0
A09-QSM-3	OMNI	RPSMA	1.9	Fixed/Mobile	0
A09-QSM-3H	OMNI	RPSMA	1.9	Fixed/Mobile	0

Part Number	Type	Connector	Gain (dBi)	Application	Minimum Cable Loss or TX Power Reduction Required in dB
A09-QBMM-P61	OMNI	MMCX	1.9	Fixed/Mobile	0
		Max Gain	8.1		

Yagi antennas

Part Number	Type	Connector	Gain (dBi)	Application	Minimum Cable Loss or TX Power Reduction Required in dB
A09-Y6	2-Element Yagi	RPN	6.1	Fixed/Mobile	0.1
A09-Y7	3-Element Yagi	RPN	7.1	Fixed/Mobile	1.1
A09-Y8	4-Element Yagi	RPN	8.1	Fixed/Mobile	2.2
A09-Y9	4-Element Yagi	RPN	9.1	Fixed/Mobile	3.1
A09-Y10	5-Element Yagi	RPN	10.1	Fixed/Mobile	4.1
A09-Y11	6-Element Yagi	RPN	11.1	Fixed/Mobile	5.1
A09-Y12	7-Element Yagi	RPN	12.1	Fixed/Mobile	6.1
A09-Y13	9-Element Yagi	RPN	13.1	Fixed/Mobile	7.1
A09-Y14	10-Element Yagi	RPN	14.1	Fixed/Mobile	8.1
A09-Y14	12-Element Yagi	RPN	14.1	Fixed/Mobile	8.1
A09-Y15	13-Element Yagi	RPN	15.1	Fixed/Mobile	9.1
A09-Y15	15-Element Yagi	RPN	15.1	Fixed/Mobile	9.1
A09-Y6TM	2-Element Yagi	RPTNC	6.1	Fixed/Mobile	0.1
A09-Y7TM	3-Element Yagi	RPTNC	7.1	Fixed/Mobile	1.1
A09-Y8TM	4-Element Yagi	RPTNC	8.1	Fixed/Mobile	2.1
A09-Y9TM	4-Element Yagi	RPTNC	9.1	Fixed/Mobile	3.1
A09-Y10TM	5-Element Yagi	RPTNC	10.1	Fixed/Mobile	4.1
A09-Y11TM	6-Element Yagi	RPTNC	11.1	Fixed/Mobile	5.1
A09-Y12TM	7-Element Yagi	RPTNC	12.1	Fixed/Mobile	6.1
A09-Y13TM	9-Element Yagi	RPTNC	13.1	Fixed/Mobile	7.1
A09-Y14TM	10-Element Yagi	RPTNC	14.1	Fixed/Mobile	8.1
A09-Y14TM	12-Element Yagi	RPTNC	14.1	Fixed/Mobile	8.1
A09-Y15TM	13-Element Yagi	RPTNC	15.1	Fixed/Mobile	9.1

Part Number	Type	Connector	Gain (dBi)	Application	Minimum Cable Loss or TX Power Reduction Required in dB
A09-Y15TM	15-Element Yagi	RPTNC	15.1	Fixed/Mobile	9.1
		Max Gain	15.1		