XLR PRO™

Radio Frequency (RF) Modem

User Guide

Revision history—90002202

Revision	Date	Description
D	December 2015	Added certifications for Mexico and Australia. Added USB support available with XLR PRO firmware version 1007.
E	February 2016	Noted that user guide applies to all XLR PRO models (XLR PRO and XLR PRO INTL). Moved source content and made miscellaneous editorial corrections.
F	January 2017	Added a serial cable warning. Added Brazil and Peru certification information. Updated AT commands and API frames.
G	December 2017	Added RF power level saturation information.
H	May 2018	Added note on range estimation. Changed IC to ISED.

Trademarks and copyright

Disclaimers

Information in this document is subject to change without notice and does not represent a commitment on the part of Digi International. 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.

Warranty

To view product warranty information, go to the following website:

www.digi.com/howtobuy/terms

Send comments

Documentation feedback: To provide feedback on this document, send your comments to techcomm@digi.com.

Customer support

Digi Technical Support: Digi offers multiple technical support plans and service packages to help our customers get the most out of their Digi product. For information on Technical Support plans and pricing, contact us at +1 952.912.3444 or visit us at www.digi.com/support.

Contents

Operations

Networking methods

AT commands

Operate in API mode

Advanced application features

Configure the XLR PRO using the web configuration interface

Configure the XLR PRO using XCTU

Configure the XLR PRO using Digi Remote Manager

Troubleshooting

Safety notices and certifications

Antennas

XLR PRO Radio Frequency (RF) Modem User Guide

The 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. There are two models:

- The XLR PRO RF Modem for North American distribution
- The XLR PRO INTL RF Modem for international distribution

In this documentation, the term XLR PRO refers to both models.

Packaged in a sturdy, rugged enclosure and using patent-pending Punch2™ Technology to maximize range and significantly increase immunity to interference, the XLR PRO 900 MHz radio can connect a variety of devices across many industrial applications.

Punch2 Technology leverages chirp spread spectrum (CSS) modulation to provide better receiver sensitivity, multipath performance, and interference rejection than is available through commonly used frequency hopping spread spectrum (FHSS) or direct sequence spread spectrum (DSSS) systems. The advantages of Punch2 Technology arise from the characteristics of the chirp signal as well as several digital-signal-processing techniques that enhance performance and reliability. With Punch2 Technology, data is spread to a higher bandwidth by multiplying each transmit modulation symbol with a chirp signal. Operating at an expanded bandwidth provides several benefits:

- Greater receiver sensitivity
- Interference immunity
- Improved multipath performance
- Adjustable data rates

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

Operational modes

The XLR PRO offers the following operational modes:

- **Serial (RS-232/RS-485/USB):** In serial mode, the front panel serial port provides connectivity to the XLR PRO via RS-232 or RS-485/422 as well as a USB serial virtual COM port.
- **Ethernet (IP socket):** In IP socket mode, an XLR PRO can transmit and receive serial data via a TCP or UDP connection from either of the front panel Ethernet ports.
- **Ethernet RF Bridging:** In Ethernet RF bridging mode, an XLR PRO functions as an Ethernet cable replacement, supporting point-to-multi-point transmission for a maximum of 16 XLR PRO RF modems. 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, there will be some latency.

For more information

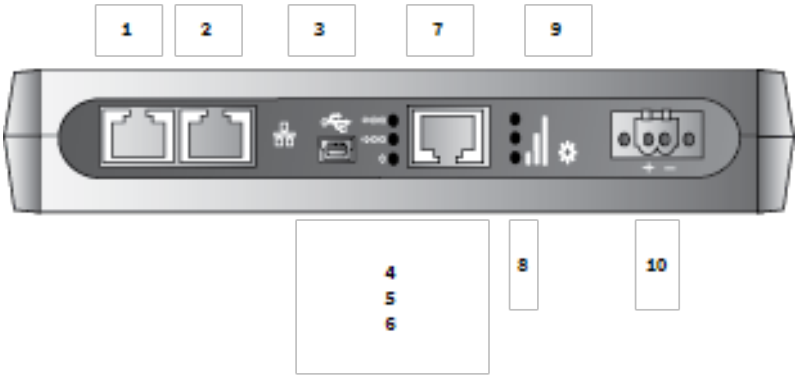
The XLR PRO Radio Frequency (RF) family of products includes the following publications:

Title	Part number	Description
XLR PRO Radio Frequency (RF) Module User Guide	90001407	Provides complete information on all XLR PRO Radio Frequency (RF) Module features; describes how to configure XLR PROs using XCTU; provides reference information on all supported AT commands and API frames.
XLR PRO Radio Frequency (RF) Modem Quick Start Guide	90002204	Provides a brief summary of the XLR PRO and XLR PRO INTL Radio Frequency (RF) Modem kit.
XLR PRO Radio Frequency (RF) Modem Getting Started Guide	90002203	Provides step-by-step instructions for setting up a pair of XLR PRO (or XLR PRO INTL) modems to test over-the-air communications between the radios.

Hardware

XLR PRO front panel

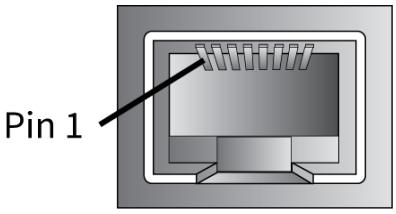
The following figure shows XLR PRO front panel connectors and LEDs.



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

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 your XLR PRO warranty.

RJ45 serial port pinout



Pin	RS-232	RS-485 (4-wire)	RS-485 (2-wire)
1	RXD	TX+	TX/RX+
2	$\overline{\text{CTS}}$	TX-	TX/RX-
3	TXD	RX+	Unused
4	GND	GND	GND
5	GND	GND	GND
6	$\overline{\text{RTS}}$	RX-	Unused
7	DRS/DCD	Unused	Unused
8	DTR	Unused	Unused

Hardware interfaces

The XLR PRO front panel offers the following interfaces:

- **Serial (RJ45) port:** The serial port (RJ45) supports RS-232 or RS-485/422 protocols.
- **USB (mini USB):** The mini USB port is used for configuration and basic serial communication with a host PC. XLR PRO can act as a USB client only and requires drivers (Windows only) to operate. The XLR PRO USB driver is available here: www.digi.com/support/productdetail?pid=5603&type=drivers
- **Ethernet:** The Ethernet interface is 10/100 Base-T with a two-port Ethernet switch. The XLR PRO does not support Power over Ethernet (PoE) and must be externally powered through the DC power jack.

Power supply

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

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

LEDs

The following tables describe XLR PRO LED behavior during startup and data transmissions. For a diagram of the XLR PRO LEDS, see XLR PRO front panel.

Startup

LED	Status	Description
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 IP socket mode.

Data transmission

LED	Status	Description
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)	Solid	Serial data in LED shows flashing when serial data is being transmitted.
Serial Data Out (yellow)	Solid	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 are 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.

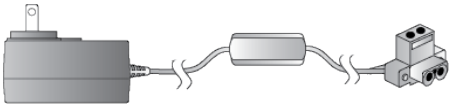
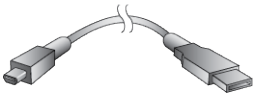
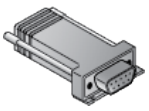
Antenna port

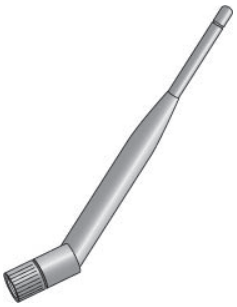
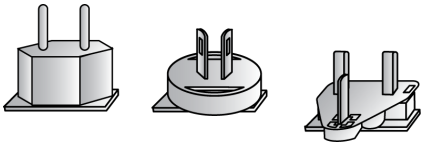
The antenna port is a 50 ohm RF signal connector for connecting to an external antenna. The connector type is Reverse Polarity TNC (RPTNC) female. The connector has threads on the outside of a barrel and a male center conductor.

Set up the hardware

XLR PRO kit contents

The following table shows the XLR PRO accessories kit.

Item	Description
	XLR PRO modem
	Power supply
	Network cable **Note** If you replace the Network cable, the replacement cable must be shielded.
	Mini USB cable
	RJ45/DB9F adapter

Item	Description
	Antenna
	**Note** Australian kit only: Power plug adapter kit (UK, EU, AS)

Connect the hardware

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

Mount the XLR PRO

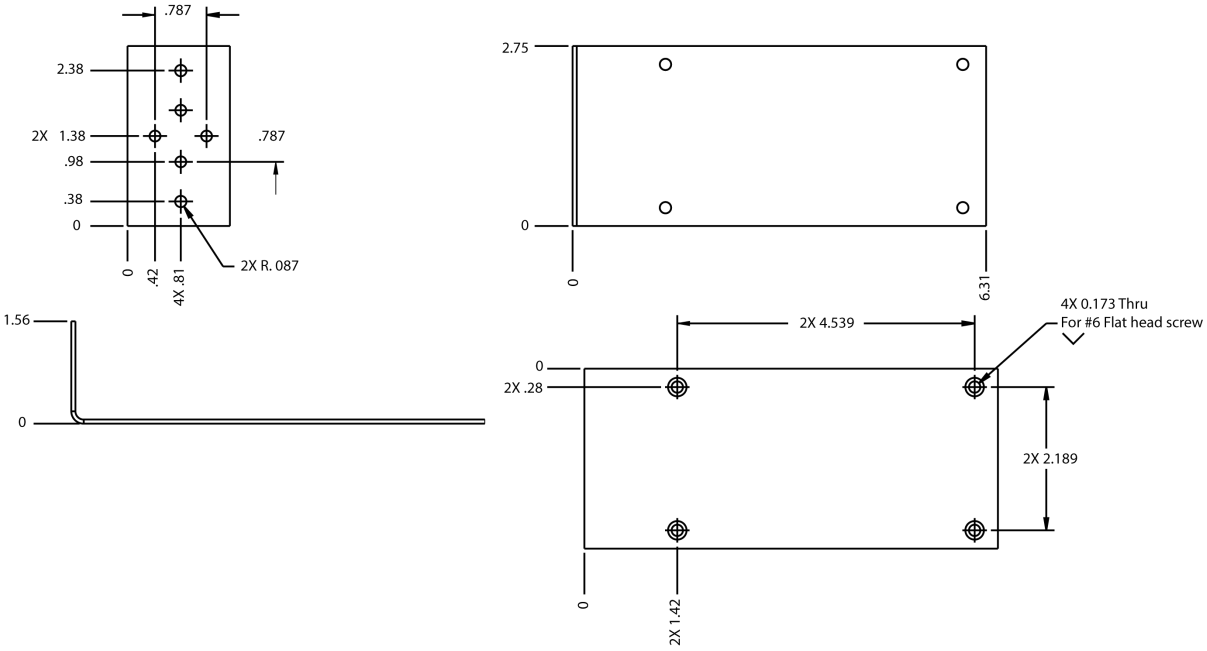
CAUTION! You must install the XLR PRO in a Restricted Access Location. In addition, you must restrict access to personnel who have been instructed on potential hazards, as well as physically restrict access using a tool, or lock and key, or other means controlled by a responsible authority.

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. There are two mounting brackets for the XLR PRO:

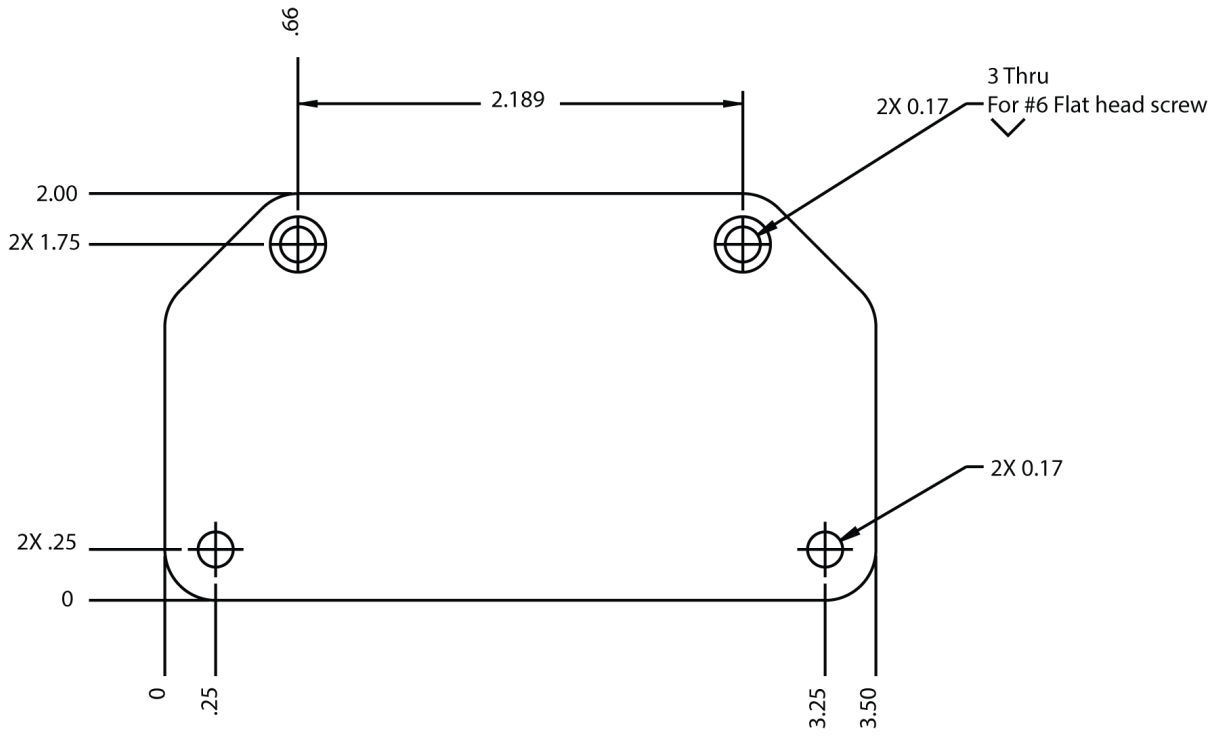
- Right-angle mount
- Flush mount

The following illustrations show the dimensions of each option.

Right-angle mount DIN rail (not to scale)



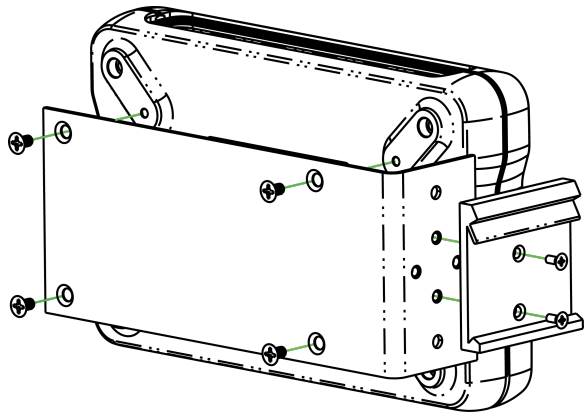
Flush-mount bracket (not to scale)



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 Ω coaxial cable, suitable for 900 MHz UHF radio transmission.

Technical specifications

The following tables provide the device's technical specifications.

General specifications

The following table describes the general specifications for the devices.

Specification	Value
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

RF characteristics

The following table provides the RF characteristics for the device.

Specification	Value
Frequency range: USA, Canada and Mexico	ISM 902 to 928 MHz
Frequency range: Australia and Brazil	ISM 915 to 928 MHz
RF data rate	9.4 kb/s to 3.2 Mb/s
Receiver selectivity at 141 kb/s RF data rate	70 dB (below 908 MHz, above 922 MHz) 40 dB (908 MHz to 922 MHz)
Maximum transmit power (software selectable)	+30 dBm (1 W)
Modulation	Chirp Spread Spectrum
RF power level saturation	Receiver input begins to saturate at 0 dBm. Input damage level is 4 dBm. We recommend placing the modems at least 2 meters apart from each other when transmitting.

Rural range line-of-sight

Note Range figure estimates are based on free-air terrain with limited sources of interference. Actual range will vary based on transmitting power, orientation of transmitter and receiver, height of transmitting antenna, height of receiving antenna, weather conditions, interference sources in the area, and terrain between receiver and transmitter, including indoor and outdoor structures such as walls, trees, buildings, hills, and mountains.

Speed	Range
1.2 Mb/s	up to 100+ miles[1]

Receiver sensitivity

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

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

UART interface data rates (software selectable)

UART interface	Data rate
TCP/UDP socket	460.8 kb/s
Serial RS-485	921.6 kb/s
Serial RS-232	460.8 kb/s
UART	Up to 921.6 kb/s

Networking and security specifications

The following table describes the networking and security specifications for the devices.

Item	Specification
Supported network topologies	Point-to-point/point-to-multipoint
Encryption	128-bit AES

[1]Based on 100-mile range results. Other data rates scale based on sensitivity levels. Results will vary based on noise levels and line of sight quality.

Power requirements

The following table describes the power requirements for the XLR PRO.

Item	Value	
Supply voltage	9 VDC to 26 VDC	
Receive current (typical)	@ 9 VDC	300 mA typical
	@ 12 VDC	230 mA typical
	@ 26 VDC	120 mA typical
Transmit current (typical)	@ 9 VDC	950 mA typical
	@ 12 VDC	840 mA typical
	@ 26 VDC	400 mA typical

Power supply

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

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

Environmental

Specification	Description
Operating temperature	-40° C to 70° C

Regulatory conformity summary

This table describes the agency approvals for the devices.

Regulation	Approval
Emissions/immunity	FCC Part 15B
Hazardous locations	Class I, Division 2, Groups A, B, C, and D
Mexico safety Mexico radio	NOM NYCE safety NOM-121

Regulation	Approval
Australia	RCM
Brazil	ANATEL: 0621-16-1209
Peru	Yes

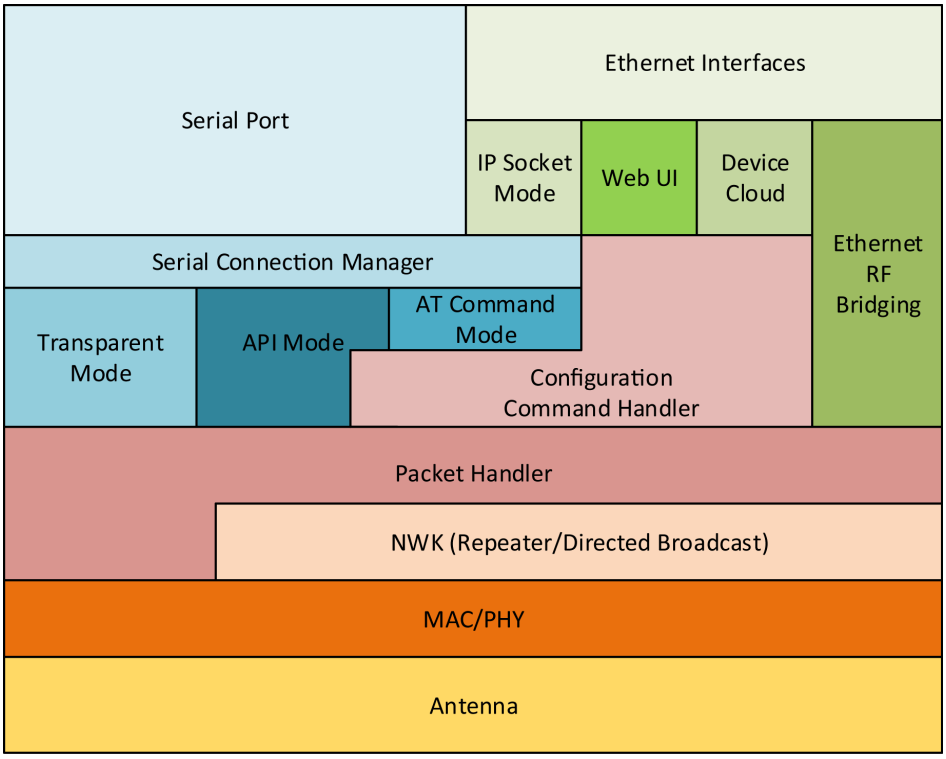
Connectors

Connector	Description
Antenna	RPTNC
Power supply	Phoenix
Ethernet	(2) RJ45
Serial	(1) RJ45
Configuration/Communication port	Mini USB

Operations

XLR PRO operational design

The XLR PRO uses a multi-layered firmware base for data flow. The flow of data depends on the hardware and software configuration you choose. 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 an interface block is able to touch another block above or below, the two interfaces can interact. For example, if the XLR PRO is using API mode, Transparent mode is not available.



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** (serial number high) and **SL** (serial number low) parameters on the opposite XLR PRO (example: **BA**=0x0013A20012345678).

With bridging enabled, the XLR PRO radios on the RF network should be treated as if they were a single Ethernet cable. Consult a qualified network administrator to 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

At any given time, only one serial mode can be selected. The XLR PRO selects the serial mode based on cable connections detection and configuration options.

- **USB**

  XLR PRO detects DTR on the USB virtual COM port.

- **RS-232**

  XLR PRO detects DTR on the RS-232 serial port.

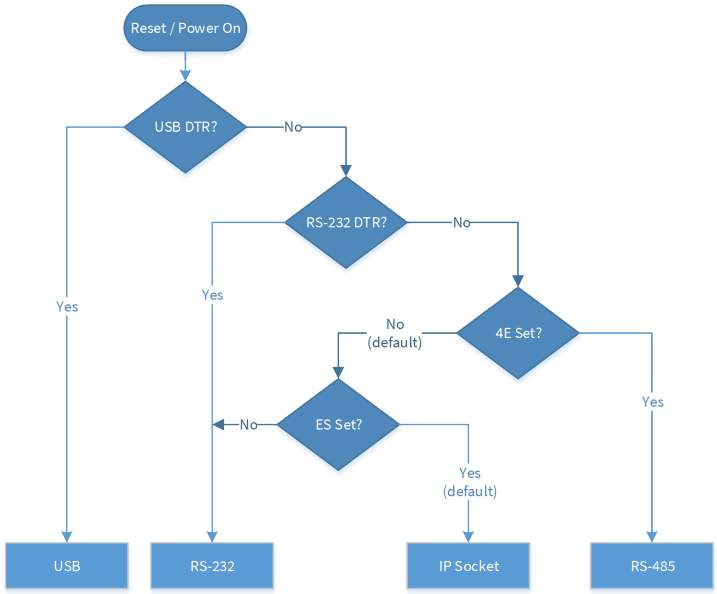
- **RS-485/422**

  RS-485 configuration option is enabled or disabled via the **4E** (Enable RS-485 Mode) parameter.

- **IP socket**

  IP Socket configuration option is enabled or disabled via the **ES** (Enable IP Socket Mode) parameter.

The following flow chart depicts how the XLR PRO selects serial mode:



There must be an active DTR signal to automatically detect an RS-232 serial connection. If the serial cable or end device does not include an active DTR signal, disable IP socket mode (**ES** = 0) using XCTU, the web configuration, or Device Cloud.

By default, RS-485 mode is disabled (**4E** = 0) and IP socket mode is enabled (**ES** = 1). So, by default, RS-232 mode is selected if DTR is present. Otherwise, IP socket mode is selected. Use the **4E** and **ES** configuration parameters to select other modes, independent of DTR, which may not be present on a serial port connection.

USB mode

The mini USB port is used for configuration and basic serial communication. To use the mini USB port, install the XLR PRO USB driver (Windows only) on the host PC. (The driver is available at www.digi.com/support/productdetail?pid=5603&type=drivers.) When connected via the mini USB port, the XLR PRO appears as a virtual COM port on the host. The XLR PRO detects an active virtual DTR signal to determine if a USB connection is active.

When communicating to the XLR PRO using the USB mode, serial settings and flow control are not used. Serial settings (baud rate, parity, and stop bits) can be set to any value on the host PC and do not affect the XLR PRO connection. If a host application requires flow control, use RS-232 serial mode.

Serial mode

When serial mode is the primary interface, the Serial Data In (green) LED lights 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 the 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 require matching parameters on the XLR PRO and the host device. This includes the following:

- **D6:** RTS flow control. If enabled, then the 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 the 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 the **D7** parameter), the XLR PRO de-asserts 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. A 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 W 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 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 460800 b/s 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 listens. The IP host must send data to this port 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**.)
- **DX:** Destination IP address. Tells the XLR PRO where to send data if it initiates the conversation. If operating in TCP mode and a TCP connection does not currently exist, then the XLR PRO attempts 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 is 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** is used. If not, then all UDP data is 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 remains 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 remains 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:

- **Reset:** Based on the mode selection rules previously described.
- **Configuration parameters:** Parameters that affect the mode are changed and applied.
- **Unplugged cable:** An RS-232 cable is unplugged.

When IP socket mode is the primary interface, the Serial Data Out (yellow) LED lights 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 USB, RS-232, RS-485/422, or IP socket mode, the XLR PRO handles the traffic as serial data, and the XLR PRO handles all serial traffic the same, regardless of the interface in use.

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 new data is discarded. If using RS-232, this can be avoided by using hardware flow control. Software flow control can be used regardless of which serial interface is 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.

Networking methods

MAC/PHY layers

PHY stands for “physical layer.” The PHY layer manages the hardware that modulates and demodulates the RF bits.

MAC stands for “media access control.” The MAC layer sends and receives RF frames. Each packet includes a MAC layer data header that contains addressing information, as well as packet options. This layer implements packet acknowledgments (ACKs), 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.

The following table shows the AT commands related to the MAC/PHY layers.

AT command	Description
ID	The **ID** (network identifier) command sets the network identifier. For XLR PRO radios to communicate, you must configure them with the same network identifier.
PL	The **PL** (power level) command sets the transmit (TX) power level. You can reduce the power level from the maximum to reduce current power consumption or to test at short distances. This comes at the expense of reduced radio range.
RR	The **RR** (unicast retries) command specifies the number of times a sending radio attempts to get an ACK from a destination radio when sending a unicast packet.
MT	The **MT** (broadcast multi-transmit) command specifies the number of times a broadcast packet is repeatedly transmitted. This adds redundancy to improve reliability.

Ethernet bridging

The purpose of Ethernet RF bridging is to act as an Ethernet cable replacement. The MAC/PHY layer of the Ethernet standard handles all Ethernet traffic. 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 with a MAC address that does not match its own MAC address, and if you enable Ethernet bridging, then the entire Ethernet packet is encapsulated inside of a radio frame and sent over the air (OTA) to another XLR PRO.

The XLR PRO does not support fragmentation, so the unit sends the entire Ethernet packet in one OTA frame. This can cause an issue at lower data rates. If **BR** (RF data rate) is less than 3 (141 kb/s), then full size Ethernet frames cannot be transmitted. However, smaller frames can be transmitted at lower data rates. We do not recommend this for typical Ethernet applications, but you could use it in some scenarios.

- **BR** = **0** or **1**: 195 bytes maximum frame size
- **BR** = **2**: 451 bytes maximum frame size

When the XLR PRO receives an RF bridging packet, it inspects the Ethernet MAC address of the packet. If the address matches the Ethernet MAC address of the XLR PRO, then the XLR PRO handles the packet on board. Otherwise, it forwards the frame over the Ethernet interface.

The **MT** (broadcast multi-transmit) and **RR** (unicast retries) 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 the XLR PRO sends serial and Ethernet traffic at the same time, the traffic encounters latency.

Bridging precautions

The practical application of an Ethernet bridge is to span two Ethernet networks that are not otherwise connected. If they are connected by another 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. The XLR PRO does not implement STP.

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 create multiple paths and you connect enterprise level switches with STP to the XLR PRO devices, then the connected switch ports are shutdown.

When bridging two networks, if each has its own DHCP server, it can create many problems.

Enable bridging

To enable bridging, set BE (Ethernet RF Bridging Enable) to **1**. This parameter 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. You can read the factory-assigned address with the **SH** (serial number high) and **SL** (serial number low) commands. Addresses use the following form:

0x0013A2XXXXXXXXXX

The first six digits are the Digi Organizationally Unique Identifier (OUI). The broadcast address is 0x000000000000FFFF.

Unicast

To transmit to a specific device:

- For Ethernet RF bridging, set **BA** (bridge destination MAC) to the **SH**:**SL** of the destination radio.

  BA is the entire 64-bit address and is not broken into two 32-bit values. For example:

  BA=0x0013A20012345678

- If you have configured Ethernet bridging for unicasts, up to three retries occur while waiting for an ACK. As a result, unicasts are slower, but more reliable than broadcasts.

Broadcast

To transmit to all devices:

- When using Transparent mode, set **DH** to all 0's and **DL** to 0xFFFF.
- For API mode set 0x000000000000FFFF in the 64-bit destination address field of the API frame.

By default, Ethernet bridging uses broadcasts and no retransmissions occur. If you send packets via TCP, then the TCP protocol provides the retransmissions as needed to provide for reliability. UDP packets may be lost when **BA** is set to a broadcast address.

Serial addressing basics

64-bit addresses

Each radio has a unique factory-assigned IEEE 64-bit address. You can read the factory-assigned address with the **SH** (serial number high) and **SL** (serial number low) commands. Addresses use the following form:

0x0013A2XXXXXXXXXX

The first six digits are the Digi Organizationally Unique Identifier (OUI). The broadcast address is 0x000000000000FFFF.

Unicast

To transmit to a specific device:

- When using Transparent mode, set **DH** and **DL** (destination address high and destination address low) to match the **SH** and **SL** (serial number high and serial number low) of the destination device.
- For API mode, set **SH** and **SL** address in the 64-bit destination address field of the API frame.

Broadcast

To transmit to all devices:

- When using Transparent mode, set **DH** to all 0's and **DL** to 0xFFFF.
- For API mode set 0x000000000000FFFF in the 64-bit destination address field of the API frame.

By default, Ethernet bridging uses broadcasts and no retransmissions occur. If you send packets via TCP, then the TCP protocol provides the retransmissions as needed to provide for reliability. UDP packets may be lost when **BA** is set to a broadcast address.

Delivery methods

The TO (Transmit Options) command sets the default delivery method that the device uses when in Transparent mode. In API mode, the TxOptions field of the API frame overrides the **TO** command, if non-zero.

The XLR PRO supports two delivery methods:

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

Point to Point / Point to Multipoint (P2MP)

This delivery method does not use a network header, only the MAC header.

In P2MP, the sending devices always send all messages directly to the destination. Other nodes do not repeat the packet. The sending device only delivers a P2MP unicast directly to the destination device, which must be in range of the sending device.

The XLR PRO uses patented technology that allows the destination device to receive unicast transmissions directed to it, even when there is a large amount of traffic. This works best if you keep broadcast transmissions to a minimum.

A sending node repeats a P2MP broadcast transmission **MT**+1 times, but the receiving nodes do not repeat it, so like a unicast transmission, the receiving device must be in range.

All devices that receive a P2MP broadcast transmission will output the data through the active serial interface.

Repeater/directed broadcast

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** (transmit option) parameter.

All of the routers in a network receive and repeat directed broadcast transmissions. Because it does not use ACKs, the originating node sends the broadcast multiple times. By default a broadcast transmission is sent four times—the extra transmissions become automatic retries without acknowledgments. This results in all nodes repeating the transmission four times. Sending frequent broadcast transmissions can quickly reduce the available network bandwidth, so use broadcast transmissions sparingly.

MAC layer

The MAC layer is the building block that is used to build repeater capability. To implement Repeater mode, we use 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 is 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 sends it out the serial port. This is called a directed broadcast. Any node that has a **CE** (node messaging option) set to route will rebroadcast the packet if its **BH** (broadcast hops) or broadcast radius values have not been depleted. If a repeated broadcast has already been seen, the node will ignore it. The **NH** (network hops) parameter sets the maximum number of hops that a broadcast will be repeated. This value is always used, unless a smaller **BH** value is specified.

By default, the **CE** (node messaging option) parameter is set to not route broadcasts. Due to the long-range of the XLR PRO, Digi advises you to evaluate on a per-radio basis which nodes should be configured as repeaters. Limiting the amount of congestion and generated RF traffic provides a more reliable network.

Transmission timeouts

When a node receives an API Tx Request while in API mode or an RO command while in Transparent mode, the time required to route the data to its destination depends on a number of configured parameters and whether the transmission is a unicast or a broadcast.

Note The timeouts in this section are theoretical timeouts. An 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.

Transmit a broadcast

All of the routers in a network must relay a broadcast transmission.

The maximum delay occurs 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 * NN * %8

Unless **BH** < **NH**, in which case the formula is:

BroadcastTxTime = BH * NN * %8

AT commands

Special commands

The following commands are special commands.

AC (Apply Changes)

Immediately applies new settings without exiting Command mode.

Parameter range

N/A

Default

N/A

FR (Software Reset)

Resets the device. The device responds immediately with an **OK** and performs a reset 100 ms later.

If you issue **FR** while the device is in Command Mode, the reset effectively exits Command mode.

Parameter range

N/A

Default

N/A

RE command

Restore device parameters to factory defaults.

In order for the default parameters to persist through subsequent resets, send a separate **WR** command after **RE**.

Parameter range

N/A

Default

N/A

WR command

Writes parameter values to non-volatile memory so that parameter modifications persist through subsequent resets.

When you issue a **WR** command add a 100 millisecond delay or wait for an **OK** response before issuing any subsequent AT commands.

Note Once you issue a **WR** command, do not send any additional characters to the device until after you receive the **OK** response.

Parameter range

N/A

Default

N/A

MAC/PHY commands

The following AT commands are MAC/PHY commands.

ID (Network ID)

Sets or displays the network identifier for the module. To communicate with other modules in a network, the modules must have matching network identifiers. If you are using OEM network identifiers, set **ID** to **FFFF** to use the factory value.

Parameter range

0 - 0x7FFF

Default

0x7FFF

BR (RF Data Rate)

Sets or displays the rate at which RF data is transmitted for all operational modes. Devices within a network do not need to have matching data transmission rates. The **BR** setting does not control the rate at which devices receive data.

Range

An integer from **0** through **8**:

Value	Description
0	9.38 kb/s
1	28.14 kb/s
2	65.66 kb/s
3	140.7 kb/s
4	290.8 kb/s
5	590.9 kb/s
6	1.191 Mb/s
7	2.392 Mb/s
8	3.189 Mb/s

Default

4

PL (Power Level)

Sets or displays the power level at which the device transmits conducted power. Power levels are approximate.

Parameter range

An integer from **0** through **4**:

Value	Description
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)

Default

4

RR (Unicast Retries)

Set or read the maximum number of MAC level packet delivery attempts for unicasts. If **RR** is non-zero, the sent unicast packets request an acknowledgment from the recipient. Unicast packets can be retransmitted up to **RR** times if the transmitting device does not receive a successful acknowledgment.

Parameter range

0 - 0xF

Default

0xA (10 retries)

MT (Broadcast Multi-Transmits)

Set or read the number of additional MAC-level broadcast transmissions. All broadcast packets are transmitted **MT**+1 times to ensure they are received.

Parameter range

0x0 - 0x8

Default

3

Diagnostic commands

The following commands are diagnostic commands.

DB (Received Signal Strength)

Reports the RSSI in -dBm of the last received RF data packet. **DB** returns a hexadecimal value for the -dBm measurement.

For example, if **DB** returns 0x60, then the RSSI of the last packet received was -96 dBm.

DB only indicates the signal strength of the last hop. It does not provide an accurate quality measurement for a multihop link.

Parameter range

0 - 0xFF [read-only]

Default

N/A

EA (MAC ACK Timeouts)

Reports or resets the total number of MAC-level unicast transmissions that timed out waiting for a MAC ACK. The total can be up to **RR** (unicast retries) + 1 timeouts per unicast, up to a maximum of **0xFFFF**. After **0xFFFF**, additional retries are not counted. You can reset the counter to any 16-bit value within the valid range by appending a hexadecimal value to the **EA** command.

EA is a volatile value—that is, the value does not persist across device resets.

Parameter range

0 - 0xFFFF

Default

N/A

ER (Received Error Count)

Reports or resets the total number of received packets that were rejected because of bit errors in the packet, up to a maximum of **0xFFFF** errors. After **0xFFFF**, additional errors are not counted. (Occasionally, random noise can cause a packet to be rejected.) You can reset the counter to any 16-bit value within the valid range by appending a hexadecimal value to the **ER** command.

ER is a volatile value—that is, the value does not persist across device resets.

Parameter range

0 - 0xFFFF

Default

0

GD (Good Packets Received)

Reports or resets the total number of successfully received packets that contain a valid MAC header, up to a maximum of **0xFFFF** packets. After **0xFFFF**, additional successfully received packets are not counted. You can reset the counter to any 16-bit value within the valid range by appending a hexadecimal value to the **GD** command.

GD is a volatile value—that is, the value does not persist across device resets.

Parameter range

0 - 0xFFFF

Default

N/A (0 after reset)

TR (Transmission Failure Count)

Reports or resets the total number of unicast transmissions for which all retries failed with no MAC ACK from the destination node, up to a maximum of **0xFFFF** transmission failures. After **0xFFFF**, failures are no longer counted. You can reset the counter to any 16-bit value within the valid range by appending a hexadecimal value to the **TR** command.

TR is a volatile value—that is, the value does not persist across device resets.

Parameter range

0 - 0xFFFF

Default

N/A

UA (Unicasts Attempted)

Reports or resets the total number of MAC unicast transmissions for which an ACK is requested, up to a maximum of **0xFFFF** transmissions. After **0xFFFF**, additional transmissions are not counted. You can reset the counter to any 16-bit value within the valid range by appending a hexadecimal value to the **UA** command.

UA is a volatile value—that is, the value does not persist across device resets.

Parameter range

0 - 0xFFFF

Default

0

%H (MAC Unicast One Hop Time)

The MAC unicast one hop time timeout in milliseconds. If you change the MAC parameters it can change this value.

Parameter range

[read-only]

Default

0x267

%8 (MAC Broadcast One Hop Time)

The MAC broadcast one hop time timeout in milliseconds. If you change MAC parameters, it can change this value.

Parameter range

[read-only]

Default

0x23D

N? (Network Discovery Timeout)

Reports the maximum response time in milliseconds for **ND** (Network Discover) and **DN** (Discover Node) responses. The timeout is based on **NT** (Node Discovery Timeout) and the network propagation time.

Parameter range

[read-only]

Default

0x3C41

Network commands

The following commands are network commands.

CE (Node Messaging Options)

Sets the routing and messaging mode for the device. A device can be configured to route or not route and configured to multi-hop packets when TO (Transmit Options) is configured for Directed Broadcast (0x80).

Parameter range

An integer from 0 through 2:

Value	Description
0	Standard router node. A standard router repeats directed broadcasts.
1	Not applicable.
2	Non-routing node.

Default

2

BH command

Sets or displays the maximum number of transmission hops for directed broadcast data transmissions when **TO** (Transmit Options) is configured for Directed Broadcast (0x80). For maximum hops, set the value to **0**. If **BH** is set to a value greater than the value for NH (Network Hops), then the **NH** value is used.

Parameter range

An integer from **0** through **4**.

Default

0

NH (Network Hops)

Sets or displays the maximum number of hops expected for a Directed Broadcast network.

Parameter range

An integer from **0** through **4**.

Default

4

NN (Network Delay Slots)

Sets or displays the maximum delay slots before rebroadcasting a Directed Broadcast packet.

Parameter range

An integer from **0** through **8**.

Default

3

Addressing commands

The following AT commands are addressing commands.

SH command

Displays the upper 32 bits of the unique IEEE 64-bit extended address assigned to the XLR PRO in the factory.

Parameter range

0 - 0xFFFFFFFF [read-only]

Default

Set in the factory

SL command

Displays the lower 32 bits of the unique IEEE 64-bit RF extended address assigned to the XLR PRO in the factory.

Parameter range

0 - 0xFFFFFFFF [read-only]

Default

Set in the factory

DH command

Displays the upper 32 bits of the unique IEEE 64-bit RF extended address for the destination module. **DH** and DL command together define the destination address used for transmission of transparent data. For broadcast, use the destination address **0x000000000000FFFF**.

Parameter range

0 - 0xFFFFFFFF

Default

0

DL command

Displays the lower 32 bits of the unique IEEE 64-bit RF extended address for the destination device. **DH** and **DL** together define the destination address used for transmission of transparent data in either serial or IP socket modes.

0x000000000000FFFF is the broadcast address.

Parameter range

0 - 0xFFFFFFFF

Default

0xFFFF

TO (Transmit Options)

The bitfield that configures the transmit options for Transparent mode.

Sets or displays transmit options for all serial transmissions. **TO** options can be overridden packet-by-packet using the **TxOptions** field of an API **TxRequest** frame.

Parameter range

One of the following hexadecimal values:

Value	Description
0x40	Point-to-point/multipoint, ACK enabled
0x41	Point-to-point/mulitpoint, ACK disabled
0x80	Repeater/Directed broadcast, ACK enabled
0x81	Repeater/Directed broadcast, ACK disabled

When you set **BR** to **0** the **TO** option has the DigiMesh and Repeater mode disabled automatically.

Default

0x40

NI command

Sets or displays a string identifier for the XLR PRO. The **NI** string identifier is returned by the ND (Network Discover) command. The **NI** string identifier can also be used by the DN (Discover Node) command to set the destination address—DL command and —to the extended 64-bit address of the XLR PRO with the matching **NI** string identifier.

Parameter range

A string of case-sensitive ASCII printable characters from 0 to 20 bytes in length. A carriage return or a comma automatically ends the command.

Default

One ASCII space character (0x20)

NT (Node Discovery Timeout)

Sets or displays the maximum randomized delay time used for sending network discovery responses—ND (Network Discover), DN (Discover Node), and FN (Find Neighbors) command responses. The random delay time is used to stagger the discovery command responses to alleviate network congestion.

Use N? (Network Discovery Timeout) to determine the maximum response time a ND (Network Discover) response requires based on **NT** and network propagation time.

Parameter range

0x20 - 0x2EE0 (x 100 ms)

Default

0x82 (13 seconds)

NO (Node Discovery Options)

Sets or displays network discovery options. Depending on the selected options, **NO** changes the behavior of the ND (Network Discover) command and determines the values returned for received **ND** responses and API node identification frames.

Parameter range

0x0 - 0x7 (bit field)

Hex value	Bitfield	Description
0x01	0000 0001	Appends DD command value to ND (Network Discover) responses and API node identification frames.
0x02	0000 0010	Sends **ND** or FN (Find Neighbors) response frame when **ND** is issued.
0x03	0000 0011	Selects both 01 and 02 options
0x04	0000 0100	Appends RSSI of the last hop for the repeater networks to **ND** or **FN** responses and API node identification frames.

Hex value	Bitfield	Description
0x05	0000 0101	Selects both 01 and 04 options.
0x06	0000 0110	Both 02 and 04 options.
0x07	0000 0111	Select all options: 01, 02, and 04.

Default

0x0

CI (Cluster ID)

Sets or displays the default application layer cluster identifier used for all data transmissions.

Parameter range

0 - 0xFFFF

Value	Description
0x11	Transparent data
0x12	Loopback (the destination node echoes transmitted packets back to the originator)
0x14	Link test
0x23	Memory Access (GPM)

Default

0x11

DE command

Sets or displays the application layer destination ID value. The value is used as the destination endpoint for all data transmissions. The default value (0xE8) is the Digi data endpoint.

Parameter range

Value	Description
0xE6	Digi device endpoint
0xE8	Digi data endpoint

Default

0xE8

SE command

Sets or displays the application layer source endpoint value. The value is used as the source endpoint for all data transmissions. The default value (0xE8) is the Digi data endpoint.

Sets or displays the application layer source endpoint value used for data transmissions.

This command only affects outgoing transmissions in transparent mode (**AP**=0).

0xE8 is the Digi data endpoint used for outgoing data transmissions.

0xE6 is the Digi device object endpoint used for configuration and commands.

Parameter range

Value	Description
0xE6	Digi device endpoint
0xE8	Digi data endpoint

Default

0xE8

Addressing discovery and configuration commands

DN (Discover Node)

Resolves an **NI** (Node identifier) string to a physical address (case sensitive).

When **DN** is sent in Command mode:

When a destination address is discovered, the device:

1. Sets DL command and **DH** to the extended 64-bit address of the device with the matching string.
2. Returns **OK<CR>**.
3. Exits command mode to allow immediate communications.

For API mode:

When a destination address is discovered:

- Receiving device returns **0xFFFE** and 64-bit extended addresses in an API command response frame.

Errors

If there is no response after the number of milliseconds set by the N? (Network Discovery Timeout) parameter or a parameter is not specified (left blank), the command is terminated and an **ERROR** message is returned. When an **ERROR** is returned, command mode is not exited.

Parameter range

A string of case-sensitive ASCII printable characters from 1 to 20 bytes in length. The string cannot start with the space character. A carriage return or a comma automatically ends the command.

Default

N/A

ND (Network Discover)

Discovers and reports all devices found in the network.

For each discovered device, the following information is returned:

RESERVED<CR> (always **0xFFFE**)

SH<CR> (4 bytes)

SL<CR> (4 bytes)

NI<CR> (Variable length, up to 20 bytes)

PARENT_NETWORK ADDRESS<CR> (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 settings.)

RSSI OF LAST HOP<CR> (1 Byte. Optionally included based on settings.)

<CR>

After the number of milliseconds set by the N? (Network Discovery Timeout) parameter, the command ends by returning a carriage return (CR). Optionally, **ND** also accepts a as a parameter and only a device that matches the identifier is returned.

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 returned is the same without carriage return delimiters. The string ends with a **0x00** (null) character.

Parameter range

N/A

Default

N/A

FN (Find Neighbors)

Discovers and reports all devices found within immediate RF range.

For each discovered device, the following information is reported:

RESERVED<CR> (always **0xFFFE**)

SH<CR> (4 bytes)

SL<CR> (4 bytes)

NI<CR> (Variable length, up to 20 bytes)

PARENT_NETWORK ADDRESS<CR> (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 settings.)

RSSI OF LAST HOP<CR> (1 Byte. Optionally included based on settings.)

<CR>

If the **FN** command is issued in command mode, after the number of milliseconds set by the N? (Network Discovery Timeout) parameter + overhead time, the command ends by returning a carriage return (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 returned is the same without carriage return delimiters. The string ends with a **0x00** (null) character.

Parameter range

N/A

Default

N/A

Security commands

The following AT commands are security commands.

KY (AES Encryption Key)

Sets the 16-byte network security key used for encryption and decryption of transmitted data. This command is write-only. If you attempt to read **KY**, an **OK** status is returned. You must set the encryption key to the same value for all devices for successful communication.

Parameter range

128-bit value

Default

N/A

Serial interfacing commands

The following AT commands are serial interfacing commands.

BD (Baud Rate)

Sets or displays the serial baud rate for the XLR PRO.

BD affects only the interface data rate for RS-232 and RS-485/422 data through the serial port.

To set a non-standard baud rate, enter a value above **0x5B9**. **BD** adjusts the value to the closest supported baud rate. After entering a specific baud rate, query **BD** to read the actual baud rate. Baud rates can be set as high as 6 Mb/s, but the host and serial switching circuitry may not support it.

Parameter range

Standard baud rates: 0x1 - 0xA

Non-standard baud rates: 0x5B9 through 0x5B8D80

Default

0x03 (9600 b/s)

NB (Parity)

Set or display the parity settings for serial communications.

Parameter range

0x00 - 0x02

Parameter	Description
0x00	No parity
0x01	Even parity
0x02	Odd parity

Parameter	Description
0	No parity
1	Even parity
2	Odd parity

Default

0x00

SB (Stop Bits)

Sets or displays the number of stop bits for the UART.

Parameter range

0 - 1

Value	Description
0	One (1) stop bit.
1	Two (2) stop bits.

Default

0

RO command

Set or read the number of character times of inter-character silence required before transmission begins when operating in Transparent mode.

Set or read the number of character times of inter-character silence required before transmission begins when operating in Transparent mode.

Set **RO** to 0 to transmit characters as they arrive instead of buffering them into one RF packet.

Parameter range

0 - 0xFF (x character times)

Default

3

FT (Flow Control Threshold)

Set or display the flow control threshold.

The device de-asserts CTS and/or send XOFF when **FT** bytes are in the UART receive buffer. It re-asserts CTS when less than **FT**-16 bytes are in the UART receive buffer.

Parameter range

0x11 - 0x94F

Default

0x91F

API Mode

When you enable API, you must format the serial data as API frames because Transparent operating mode is disabled.

The device ignores this command when using SPI, where API mode is always enabled.

Parameter range

0 - 2

Value	Description
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. The device packetizes all UART input and output data in API format, without escape sequences.
2	API mode with escapes. The device is in API mode and inserts escaped sequences to allow for control characters.

Default

0

AO command

Sets or displays the API data frame output format for received frames. Applies to both UART and SPI interfaces.

Parameter range

0, 1

Value	Description
0	API RX indicator (0x90)
1	API Explicit RX indicator (0x91)

Default

0

4E (Serial Protocol)

Sets or displays the serial protocol used for serial mode operation.

Range

Value	Description
0	RS-232
1	RS-485/422

Default

0 (RS-232)

4D (RS-485 Duplex)

Sets or displays duplex option for RS-485. This settings has no impact on RS-232 communications.

Range

Value	Description
0	Two-wire (half-duplex)
1	Four-wire (full-duplex)

Default

0 (half-duplex).

4T (RS-485 Termination)

Sets or displays whether line termination is enabled or disabled for RS-485/422. Line termination enables a 120 Ω termination resistor on the RS-485/422 data lines. This setting has no impact on RS-232 communication.

Range

Value	Description
0	Line termination is disabled.
1	Line termination is enabled.

Default

0 (line termination is disabled)

D6 (RTS Flow Control)

Sets or displays the $\overline{\text{RTS}}$ configuration () for the serial port.

Parameter range

0 - 1

Value	Description
0	$\overline{\text{RTS}}$ flow control is disabled
1	$\overline{\text{RTS}}$ flow control is enabled

Default

0

D7 (CTS Flow Control)

Sets or displays whether $\overline{\text{CTS}}$ flow control is enabled or disabled for the serial port.

Parameter range

Parameter	Description
0	$\overline{\text{CTS}}$ flow control is disabled
1	$\overline{\text{CTS}}$ flow control is enabled

Default

0x1

Hardware diagnostics commands

TP command

Displays the temperature of the XLR PRO in degrees Celsius. The temperature value is displayed in 8-bit two's compliment format. For example, **0x1A** = 26 °C, and **0xF6** = -10 °C.

Because the XLR PRO produces heat, this temperature reading is usually above the ambient temperature.

Parameter range

0 - 0xFF which indicates degrees Celsius displayed in 8-bit two's compliment format.

Default

N/A

RP command

Sets or displays the amount of time (in deciseconds) the RSSI LEDs are active after a valid RF packet is received. When **RP** is **FF**, output is always on.

Parameter range

0 - 0xFF (x 100 ms)

Default

0x28 (four seconds)

Ethernet and IP socket mode commands

ES (IP Socket Mode Enable)

Sets or displays whether IP socket mode is enabled. Enabling socket mode allows serial traffic to be sent to a TCP or UDP port based on the parameter. The XLR PRO remains in listen-only state unless DX (Destination IP Address) is set to a valid IP address.

Range

Value	Description
0	IP socket mode is disabled.
1	IP socket mode is enabled.

Default

1 (IP socket mode is enabled)

IB (IP Socket Baud Rate)

Sets or displays the IP socket baud rate for the XLR PRO. This value only affects the interface data rate for serial TCP/UDP data through the Ethernet port.

Range

An integer from 1 through 9 to select a preset baud rate; or a hexadecimal value from 0x5B9 through 0x5B8D80 to specify a specific baud rate.

Preset values include:

Value	Description
1	2,400 b/s
2	4,800 b/s
3	9,600 b/s
4	19,200 b/s
5	38,400 b/s
6	57,600 b/s

Value	Description
7	115,200 b/s
8	230,400 b/s
9	460,800 b/s

To set a non-standard baud rate, enter a value above 0x5B9. **IB** adjusts the value to the closest supported baud rate. After entering a specific baud rate, query **IB** to read the actual baud rate.

Default

Default is 9 (460,800 b/s).

IP (IP Protocol)

Sets or displays the protocol used for the listening port (while the device is in Transparent mode (**AP** = **0**).

Parameter range

0, 1

Value	Description
0x00	UDP
0x01	TCP

Default

0x0

0x00 (UDP)

DX (Destination IP Address)

Sets or displays the destination IP address in IPv4 format for outgoing IP socket mode data. For broadcast, set **DX** to **255.255.255.255**.

Range

A valid address in IPv4 format from **0.0.0.0** to **255.255.255.255**.

Default

Default is **0.0.0.0**.

C0 (Source Port)

Sets or displays the listening IP port number for TCP and UDP traffic. An incoming socket is established only if the protocol (TCP or UDP) matches the protocol set by the IP (IP Protocol) parameter.

Range

A hexadecimal value from 0x0 through 0xFFFF

Default

0x2616 (port 9750)

DY (Destination Port)

Sets or displays the outgoing IP port number for TCP and UDP socket connections. A socket to this IP port is made to the destination IPv4 address defined by the DX (Destination IP Address) parameter using the protocol defined by the IP (IP Protocol) parameter.

Range

A hexadecimal value from **0x0** through **0xFFFF**.

Default

Default is hexadecimal **0x2616** (port 9750).

TM command

Sets or displays the timeout in seconds for outgoing TCP socket connections when the XLR PRO is acting as a TCP client. The connection is closed if no activity is detected during this timeout period. When set to 0x0, the connection is closed immediately after data is sent. The maximum timeout is 1 day.

Range

A hexadecimal value from 0x0 through 0x15180 (86400 seconds or 24 hours).

Default

0x3C (60 seconds).

TS (TCP Server Connection Timeout)

Sets or displays the timeout in seconds for incoming TCP socket connections when the XLR PRO is acting as a TCP server. The connection is closed if no activity is detected during this timeout period. When set to 0, the connection is closed immediately after data is sent. The maximum timeout is 1 day.

Parameter Range

A hexadecimal value from 0x0 through 0x15180 (86400 seconds or 24 hours).

Default

0x3C (60 seconds).

MA (IP Addressing Mode)

Sets or displays the IP addressing mode: DHCP or static.

If you configure DHCP and no DHCP server is detected, **Auto-IP** is used. See MY command for details.

Range

Value	Description
0	DHCP addressing mode.
1	Static addressing mode.

Default

0

MY command

Sets or displays the IP address of the XLR PRO. If MA (IP Addressing Mode) is DHCP, this parameter is read-only and an IP address is 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 is assigned. After an **Auto-IP** address is assigned, the XLR PRO requests a DHCP address assignment every 5 minutes.

The format of the **Auto-IP** address is as follows:

169.254.xxx.yyy

xxx	Second to last byte of the Ethernet MAC address.
yyy	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 is 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, these bytes become 163 and 41. The **Auto-IP** address assigned to this radio is **169.254.163.41**.

Range

A valid address in IPv4 format from **0.0.0.0** to **255.255.255.255**.

Default

N/A

MK command

Sets or displays the network subnet mask of the XLR PRO. If MA (IP Addressing Mode) is DHCP, this parameter is read-only and the subnet mask is assigned by a DHCP server on the network. The subnet mask that is assigned in **Auto-IP** is 255.255.0.0.

Range

A valid address in IPv4 format from 0.0.0.0 to 255.255.255.255.

Default

N/A

GW command

Sets or displays the gateway address of the XLR PRO. If MA (IP Addressing Mode) is DHCP, this parameter is read-only and the gateway address is assigned by a DHCP server on the network. The gateway address that is assigned in **Auto-IP** is 0.0.0.0.

Range

A valid address in IPv4 format from **0.0.0.0** to **255.255.255.255**.

Default

N/A

NS (DNS Address)

Sets or displays the IPv4 address of the domain name server for the XLR PRO.

Range

A valid address in IPv4 format from **0.0.0.0** to **255.255.255.255**

Default

N/A

%M (Ethernet MAC address)

Displays the Ethernet MAC Address assigned to the XLR PRO. This is a read-only parameter.

Range

A hexadecimal value from 0x0 through 0xFFFFFFFFFF.

Default

Default is a factory set value.

Remote Manager commands

DO (Device Cloud Enable)

Sets or displays whether Remote Manager support is enabled.

Range

0x00 - 0x03

Value	Description
0x00	Disable Remote Manager support.
0x01	Enable Remote Manager support.

Default

0x01

KP (Device Description)

Sets or displays a user-defined description for the XLR PRO displayed in Remote Manager and web configuration interfaces.

Range

From 0 through 31 ASCII characters

Default

One ASCII space character (0x20)

KC (Device Contact)

Sets or displays a user-defined contact for the XLR PRO displayed in Remote Manager and web configuration interfaces.

Range

From 0 through 31 ASCII characters.

Default

One ASCII space character (0x20).

KL (Device Location)

Sets or displays a user-defined physical location for the XLR PRO displayed in Remote Manager and web configuration interfaces.

Range

From 0 through 31 ASCII characters.

Default

One ASCII space character (0x20).

LX (Latitude)

Sets or displays a user-defined GPS latitude coordinate for the XLR PRO displayed in Remote Manager and web configuration interfaces. The latitude is a user-defined field—the XLR PRO does not have GPS functionality.

Range

From 0 through 15 ASCII characters.

Default

0.000

LY (Longitude)

Sets or displays a user-defined GPS longitude coordinate for the XLR PRO displayed in Remote Manager and web configuration interfaces. The longitude is a user-defined field—the XLR PRO does not have GPS functionality.

Range

From 0 through 31 ASCII characters.

Default

-0.000

EQ (Device Cloud FQDN)

Sets or display the fully qualified domain name of the Remote Manager server.

If NS (DNS Address) does not define a valid DNS server, enter an IP address for **EQ**.

Range

From 0 through 63 ASCII characters.

Default

devicecloud.digi.com

DI (Device Cloud Indicator)

Displays the current Remote Manager status for the XLR PRO.

Parameter range

Value	Description
0	Connected
1	Remote Manager not accessible
2	Remote Manager connection in progress
3	Disconnecting from Remote Manager
4	Not configured for Remote Manager support
5	DHCP is enabled and no DHCP server was found

Default

N/A

Web configuration commands

HE (Web Configuration Enable)

Sets or displays whether XLR PRO web configuration (HTTP) is enabled.

Range

Value	Description
0	Web configuration is disabled.
1	Web configuration is enabled.

Default

1 (web configuration is enabled).

HU (Web Configuration User Name)

Sets or displays the user name for web configuration (HTTP) access.

Range

From 0 to 63 ASCII characters.

Default

Default is **admin**.

HW (Web Configuration Password)

Sets the password for the web configuration (HTTP) access user name. This is a write-only parameter. If you attempt to display/read the password, **OK** is returned.

Range

From 0 to 63 ASCII characters.

Default

Default is **password**.

Ethernet bridging commands

BE (Ethernet RF Bridging Enable)

Sets or displays whether Ethernet RF bridging is enabled.

Range

Value	Description
0	Ethernet RF bridging is disabled.
1	Ethernet RF bridging is enabled.

Default

0x0 (disabled).

BA (Bridge Destination MAC)

Sets or displays the destination RF MAC address to use for Ethernet RF Bridging. Set **BA** to **0xFFFF** as the broadcast address.

Range

A hexadecimal value from 0x0 to 0013A200FFFFFFFF

Default

0xFFFF (broadcast address)

Command mode options

The following commands are Command mode option commands.

CC (Command Sequence Character)

Sets or displays the ASCII sequence character to use for entering Command mode. Repeating the **CC** character three times causes the XLR PRO to enter Command mode. The device responds with **OK\r** when Command mode is successfully entered. The following commands are related to **CC**:

- Use GT command to define a guard time—the amount of time before and after entering a command sequence—to guard against inadvertently entering Command mode.
- Use CT command to define the timeout for Command mode.
- Use CN command to immediately exit Command mode and return to Idle mode.

Parameter range

0 - 0xFF

Default

0x2B (the ASCII plus character: **+**)

CN command

Exits Command mode and returns the XLR PRO to Idle mode.

Parameter range

N/A

Default

N/A

CT command

Sets or displays the Command mode timeout parameter. If a device does not receive any valid commands within this time period, it returns to Idle mode from Command mode.

Parameter range

2 - 0x1770 (x 100 ms)

Default

0x64 (10 seconds)

GT command

Set the required period of silence before and after the command sequence characters of the Command mode sequence (**GT** + **CC** + **GT**). The period of silence prevents inadvertently entering Command mode.

Parameter range

0x0 - 0xFFFF

Default

0x3E8 (one second)

Firmware commands

The following AT commands are firmware commands.

VB (Firmware Version)

Displays the XLR PRO firmware version.

Range

N/A

Default

N/A

VR (XLR PRO Firmware Version)

Displays the XLR PRO module firmware.

Range

N/A

Default

N/A

HV command

Display the hardware version number of the device.

Parameter range

N/A

Default

N/A

VH command

Displays the XLR PRO RF modem baseboard hardware version.

Range

N/A

Default

N/A

*C (compatibility)

Displays the compatibility setting for the XLR PRO.

Range

N/A

Default

N/A

DD command

Stores the Digi device type identifier value. Use this value to differentiate between multiple XBee devices.

If you change **DD**, RE command will not restore defaults. The only way to get **DD** back to default values is to explicitly set it to defaults.

The XLR PRO product code upper word is **0x000E**.

Parameter range

0 - 0xFFFFFFFF

Default

0xE001C

PN (Part Number)

Displays the manufacturing part number for the XLR PRO.

Range

N/A

Default

N/A

NP (Maximum Packet Payload Bytes)

Reads the maximum number of RF payload bytes that you can send in a transmission.

Parameter range

0 - 0xFFFF (bytes) [read-only]

Default

0x640 (1600d)

CK (Configuration CRC)

Displays the cyclic redundancy check (CRC) of the current AT command configuration settings.
This command allows you to detect an unexpected configuration change on a device.
After a firmware update this command may return a different value.

Parameter range

0 - 0xFFFF

Default

N/A

Operate in API mode

API mode overview

By default, the XLR PRO acts as a serial line replacement (Transparent operation), it queues all UART data that it receive through the DI pin for RF transmission. When the device receives an RF packet, it sends the data out the DO pin with no additional information.

The following behaviors are inherent to Transparent operation:

- If device parameter registers are to be set or queried, a special operation is required for transitioning the device into Command Mode.

API operating mode is an alternative to transparent mode. API mode is a frame-based protocol that allows you to direct data on a packet basis. It can be particularly useful in large networks where you need to control the destination of individual data packets or when you need to know which node a data packet was sent from. The device communicates UART data in packets, also known as API frames. This mode allows for structured communications with serial devices. It is helpful in managing larger networks and is more appropriate for performing tasks such as collecting data from multiple locations or controlling multiple devices remotely.

Use the AP command to set the operation mode

Use API Mode to specify the operation mode:

AP command setting	Description
AP = 0	Transparent operating mode, UART serial line replacement with API modes disabled. This is the default option.
AP = 1	API operation.
AP = 2	API operation with escaped characters (only possible on UART).

The API data frame structure differs depending on what mode you choose.

API frame format

An API frame consists of the following:

- Start delimeter
- Length
- Frame data
- Checksum

API operation (AP parameter = 1)

This is the recommended API mode for most applications. The following table shows the data frame structure when you enable this mode:

Frame fields	Byte	Description
Start delimiter	1	0x7E
Length	2 - 3	Most Significant Byte, Least Significant Byte
Frame data	4 - number (n)	API-specific structure
Checksum	n + 1	1 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 replies with a radio status frame indicating the nature of the failure.

API operation with escaped characters (AP parameter = 2)

Setting API to 2 allows escaped control characters in the API frame. Due to its increased complexity, we only recommend this API mode in specific circumstances. API 2 may help improve reliability if the serial interface to the device is unstable or malformed frames are frequently being generated.

When operating in API 2, if an unescaped 0x7E byte is observed, it is treated as the start of a new API frame and all data received prior to this delimiter is silently discarded. For more information on using this API mode, see the Escaped Characters and API Mode 2 in the Digi Knowledge base.

API escaped operating mode works similarly to API mode. The only difference is that when working in API escaped mode, the software must escape any payload bytes that match API frame specific data, such as the start-of-frame byte (0x7E). The following table shows the structure of an API frame with escaped characters:

Frame fields	Byte	Description	
Start delimiter	1	0x7E	
Length	2 - 3	Most Significant Byte, Least Significant Byte	Characters escaped if needed
Frame data	4 - n	API-specific structure	
Checksum	n + 1	1 byte	

Start delimiter field

This field indicates the beginning of a frame. It is always 0x7E. This allows the device to easily detect a new incoming frame.

Escaped characters in API frames

If operating in API mode with escaped characters (**AP** parameter = 2), 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).

The following data bytes need to be escaped:

- 0x7E: start delimiter
- 0x7D: escape character
- 0x11: XON
- 0x13: XOFF

To escape a character:

1. Insert 0x7D (escape character).
2. Append it with the byte you want to escape, XORed with 0x20.

In API mode with escaped characters, the length field does not include any escape characters in the frame and the firmware calculates the checksum with non-escaped data.

Example: escape an API frame

To express the following API non-escaped frame in API operating mode with escaped characters:

Start delimiter	Length	Frame type	Frame Data	Checksum
			Data	
7E	00 0F	17	01 00 13 A2 00 40 AD 14 2E FF FE 02 4E 49 6D	

You must escape the 0x13 byte:

1. Insert a 0x7D.
2. XOR byte 0x13 with 0x20: 13 ⊕ 20 = 33

The following figure shows the resulting frame. Note that the length and checksum are the same as the non-escaped frame.

Start delimiter	Length	Frame type	Frame Data	Checksum
			Data	
7E	00 0F	17	01 00 7D 33 A2 00 40 AD 14 2E FF FE 02 4E 49 6D	

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

Length field

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

Frame data

This field contains the information that a device receives or will transmit. The structure of frame data depends on the purpose of the API frame:

Start delimiter	Length		Frame data								Checksum
			Frame type	Data							
1	2	3	4	5	6	7	8	9	...	n	n+1
0x7E	MSB	LSB	API frame type	Data							Single byte

- **Frame type** is the API frame type identifier. It determines the type of API frame and indicates how the Data field organizes the information.
- **Data** contains the data itself. This information and its order depend on the what type of frame that the Frame type field defines.

Multi-byte values are sent big-endian.

Checksum field

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.

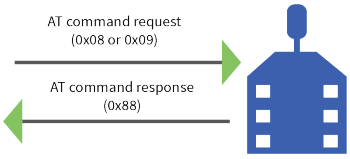
To verify: Add all bytes (include checksum, but exclude the delimiter and length). If the checksum is correct, the sum will equal 0xFF.

API serial exchanges

You can use the Frame ID field to assign an identifier to each outgoing API frame. This Frame ID, if non-zero, can correlate between the outgoing frames and the associated responses.

AT command frames

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 an XLR PRO parameter. To disable the response, set the frame ID to 0 in the request.

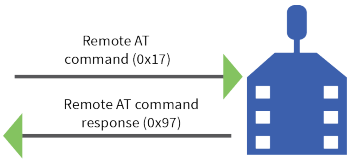


Transmit and receive RF data

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 indicates the cause of failure. The received data frame type (standard 0x90, or explicit 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.



Code to support future API frames

If your software application supports the API, you should make provisions that allow for new API frames in future firmware releases. For example, you can include the following section of code on a host microprocessor that handles serial API frames that are sent out the device's DOUT pin:

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

               case RX_IO_SAMPLE_FRAME:
                       //process IO sample frame
                       break;

               case NODE_IDENTIFICATION_FRAME:
                       //process node identification frame
                       break;


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

API frames

The following sections document API frame types.

AT Command frame - 0x08

Description

Use this frame to query or set device parameters on the local device. This API command applies changes after running the command. You can query parameter values by sending the 0x08 AT Command frame with no parameter value field (the two-byte AT command is immediately followed by the frame checksum).

A 0x88 response frame is populated with the parameter value that is currently set on the device.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x08

Frame data fields	Offset	Description
AT command	5-6	Command name: two ASCII characters that identify the AT command.
Parameter value	7-n	If present, indicates the requested parameter value to set the given register. If no characters are present, it queries the register.

Example

The following example illustrates an AT Command frame when you query the device's **NH** parameter value.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x04
Frame type	3	0x08
Frame ID	4	0x52
AT command	5	0x4E (N)
	6	0x48 (H)
Parameter value (**NH**2 = two network hops)	7	0x02
Checksum	8	0x0D

AT Command - Queue Parameter Value frame - 0x09

Description

This frame allows you to query or set device parameters. In contrast to the AT Command (0x08) frame, this frame queues new parameter values and does not apply them until you issue either:

- The **AT** Command (0x08) frame (for API type)
- The **AC** command

When querying parameter values, the 0x09 frame behaves identically to the 0x08 frame. The device returns register queries immediately and not does not queue them. The response for this command is also an **AT** Command Response frame (0x88).

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x09
Frame ID	4	Identifies the data frame for the host to correlate with a subsequent ACK. If set to **0**, the device does not send a response.
AT command	5-6	Command name: two ASCII characters that identify the AT command.
Parameter value	7-n	If present, indicates the requested parameter value to set the given register. If no characters are present, queries the register.

Transmit Request frame - 0x10

Description

This frame causes the device to send payload data as an RF packet to a specific destination.

- For broadcast transmissions, set the 64-bit destination address to **0x000000000000FFFF** .
- For unicast transmissions, set the 64 bit address field to the address of the desired destination node.
- Set the reserved field to **0xFFFE**.
- Query the **NP** command to read the maximum number of payload bytes.

You can set the broadcast radius from **0** up to **NH**. If set to **0**, the value of **NH** specifies the broadcast radius (recommended). This parameter is only used for broadcast transmissions.

You can read the maximum number of payload bytes with the **NP** command.

Note Using source routing reduces the RF payload by two bytes per intermediate hop in the source route.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x10
Frame ID	4	Identifies the data frame for the host to correlate with a subsequent ACK. If set to **0**, the device does not send a response.
64-bit destination address	5-12	MSB first, LSB last. Set to the 64-bit address of the destination device. Broadcast = **0x000000000000FFFF**
Reserved	13-14	Set to **0xFFFE**.
Broadcast radius	15	Sets the maximum number of hops a broadcast transmission can occur. If set to **0**, the broadcast radius is set to the maximum hops value.
Transmit options	16	
RF data	17-n	Up to **NP** bytes per packet. Sent to the destination device.

Example

The example shows how to send a transmission to a device if you disable escaping (**AP** = 1), with destination address 0x0013A200 400A0127, and payload “TxData0A”.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x16
Frame type	3	0x10
Frame ID	4	0x01
64-bit destination address	MSB 5	0x00
	6	0x13
	7	0xA2
	8	0x00
	9	0x40
	10	0x0A
	11	0x01
	LSB 12	0x27
Broadcast radius	15	0x00
Options	16	0x40
RF data	17	0x54
	18	0x78
	19	0x44
	20	0x61
	21	0x74
	22	0x61
	23	0x30
	24	0x41
Checksum	25	0x13

If you enable escaping (**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 device calculates the checksum (on all non-escaped bytes) as [0xFF - (sum of all bytes from API frame type through data payload)].

Explicit Addressing Command frame - 0x11

Description

This frame is similar to Transmit Request (0x10), but it also requires you to specify the application-layer addressing fields: endpoints, cluster ID, and profile ID.

This frame causes the device to send payload data as an RF packet to a specific destination, using specific source and destination endpoints, cluster ID, and profile ID.

- For broadcast transmissions, set the 64-bit destination address to **0x000000000000FFFF** .
- For unicast transmissions, set the 64 bit address field to the address of the desired destination node.
- Set the reserved field to **0xFFFE**.

Query the **NP** command to read the maximum number of payload bytes. For more information, see Firmware commands.

You can set the broadcast radius from 0 up to **NH** to 0xFF. If set to 0, the value of **NH** specifies the broadcast radius (recommended). This parameter is only used for directed broadcast transmissions (transmit options = 0x80).

Format

The following table provides the contents of the frame. For details on the frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x11
Frame ID	4	Identifies the data frame for the host to correlate with a subsequent ACK (0x8B). If set to **0**, the device does not send a response.
64-bit destination address	5-12	MSB first, LSB last. Set to the 64-bit address of the destination device. Broadcast = 0x000000000000FFFF
Reserved	13-14	Set to 0xFFFE.
Source endpoint	15	Source endpoint for the transmission.
Destination endpoint	16	Destination endpoint for the transmission.
Cluster ID	17-18	The Cluster ID that the host uses in the transmission.
Profile ID	19-20	The Profile ID that the host uses in the transmission.
Broadcast radius	21	Sets the maximum number of hops a broadcast transmission can traverse. If set to 0, the transmission radius set to the network maximum hops value.

Frame data fields	Offset	Description
Transmission options	22	
Data payload	23-n	Up to **NP** bytes per packet. Sent to the destination device.

Transmit Options bit field

Bit	Meaning	Description
0	Disable ACK	Disable acknowledgments on all unicasts
1	Disable RD	Disable Route Discovery on all DigiMesh unicasts
2	NACK	Enable NACK messages on all DigiMesh API packets
3	Trace Route	Enable a Trace Route on all DigiMesh API packets
4	Reserved	<set this bit to 0>
5	Reserved	<set this bit to 0>
6, 7	Delivery method	b’00 = <invalid option> b’01 - Point-multipoint (0x40) b'10 = Directed Broadcast (0x80)

Set all other bits to 0.

Example

The following example sends a data transmission to a device with:

- 64-bit address: 0x0013A200 01238400
- Source endpoint: 0xE8
- Destination endpoint: 0xE8
- Cluster ID: 0x11
- Profile ID: 0xC105
- Payload: TxData

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x1A
Frame type	3	0x11
Frame ID	4	0x01

Frame data fields	Offset	Example
64-bit destination address	MSB 5	0x00
	6	0x13
	7	0xA2
	8	0x00
	9	0x01
	10	0x23
	11	0x84
	LSB12	0x00
Reserved	13	0xFF
	14	0xFE
Source endpoint	15	0xE8
Destination endpoint	16	0xE8
Cluster ID	17	0x00
	18	0x11
Profile ID	19	0xC1
	20	0x05
Broadcast radius	21	0x00
Transmit options	22	0x00
Data payload	23	0x54
	24	0x78
	25	0x44
	26	0x61
	27	0x74
	28	0x61
Checksum	29	0xDD

Remote AT Command Request frame - 0x17

Description

Used to query or set device parameters on a remote device. For parameter changes on the remote device to take effect, you must apply changes, either by setting the Apply Changes options bit, or by sending an **AC** command to the remote.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x17
Frame ID	4	Identifies the data frame for the host to correlate with a subsequent ACK (0x97). If set to **0**, the device does not send a response.
64-bit destination address	5-12	MSB first, LSB last. Set to the 64-bit address of the destination device.
Reserved	13-14	
Remote command options	15	0x02 = Apply changes on remote. If you do not set this, you must send the **AC** command for changes to take effect. Set all other bits to 0.
AT command	16-17	Command name: two ASCII characters that identify the command.
Command parameter	18-n	If present, indicates the parameter value you request for a given register. If no characters are present, it queries the register.

Example

The following example sends a remote command:

In this example, the 64-bit address of the remote device is 0x0013A200 40401122.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x10
Frame type	3	0x17
Frame ID	4	0x01

Frame data fields	Offset	Example
64-bit destination address	MSB 5	0x00
	6	0x13
	7	0xA2
	8	0x00
	9	0x40
	10	0x40
	11	0x11
	LSB 12	0x22
Reserved	13	0xFF
	14	0xFE
Remote command options	15	0x02 (apply changes)
AT command	16	0x42 (B)
	17	0x48 (H)
Command parameter	18	0x01
Checksum	19	0xF5

AT Command Response frame - 0x88

Description

A device sends this frame in response to an AT Command (0x08 or 0x09) frame. Some commands send back multiple frames; for example, the **ND** command.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x88
Frame ID	4	Identifies the data frame for the host to correlate with a subsequent ACK. If set to **0**, the device does not send a response.
AT command	5-6	Command name: two ASCII characters that identify the command.
Command status	7	0 = OK 1 = ERROR 2 = Invalid command 3 = Invalid parameter
Command data	8-n	The register data in binary format. If the host sets the register, the device does not return this field.

Example

If you change the **BD** parameter on a local device with a frame ID of 0x01, and the parameter is valid, the user receives the following response.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x05
Frame type	3	0x88
Frame ID	4	0x01
AT command	5	0x42 (B)
	6	0x44 (D)
Command status	7	0x00

Frame data fields	Offset	Example
Command data		(No command data implies the parameter was set rather than queried)
Checksum	8	0xF0

Modem Status frame - 0x8A

Description

Devices send the status messages in this frame in response to specific conditions.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x8A
Status	4	0x00 Hardware reset 0x01 Watchdog timer reset

Example

When a device powers up, it returns the following API frame.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
LSB 2	LSB 2	0x02
Frame type	3	0x8A
Status	4	0x00
Checksum	5	0x75

Transmit Status frame - 0x8B

Description

When a Transmit Request (0x10, 0x11) completes, the device sends a Transmit Status message out of the serial interface. This message indicates if the Transmit Request was successful or if it failed.

Note Broadcast transmissions are not acknowledged and always return a status of 0x00, even if the delivery failed.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x8B
Frame ID	4	Identifies the serial interface data frame being reported. If Frame ID = 0 in the associated request frame, no response frame is delivered.
Reserved	5-6	Set to 0xFFFE.
Transmit retry count	7	The number of application transmission retries that occur.
Delivery status	8	
Discovery status	9	0x00 = No discovery overhead 0x02 = Route discovery

Example

In the following example, the destination device reports a successful unicast data transmission successful and a route discovery occurred. The outgoing Transmit Request that this response frame uses Frame ID of 0x47.

Frame Fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x07
Frame type	3	0x8B
Frame ID	4	0x47

Frame Fields	Offset	Example
Reserved	5	0xFF
	6	0xFE
Transmit retry count	7	0x00
Delivery status	8	0x00
Discovery status	9	0x02
Checksum	10	0x2E

Receive Packet frame - 0x90

Description

When a device configured with a standard API Rx Indicator (**AO** = **0**) receives an RF data packet, it sends it out the serial interface using this message type.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x90
64-bit source address	4-11	The sender's 64-bit address. MSB first, LSB last.
Reserved	12-13	Reserved.
Receive options	14	Bit field: bit 0 = Packet acknowledged bit 1 = Packet was a broadcast packet bits 6 and 7: b'01 = Point-Multipoint b'10 = Repeater mode (directed broadcast) Ignore all other bits.
Received data	15-n	The RF data the device receives.

Example

In the following 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 sends the following frame out its serial interface.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x12
Frame type	3	0x90

Frame data fields	Offset	Example
64-bit source address	MSB 4	0x00
	5	0x13
	6	0xA2
	7	0x00
	8	0x40
	9	0x52
	10	0x2B
	LSB 11	0xAA
Reserved	12	0xFF
	13	0xFE
Receive options	14	0x01
Received data	15	0x52
	16	0x78
	17	0x44
	18	0x61
	19	0x74
	20	0x61
Checksum	21	0x11

Explicit Rx Indicator frame - 0x91

Description

When a device configured with explicit API Rx Indicator (**AO** = **1**) receives an RF packet, it sends it out the serial interface using this message type.

The Cluster ID and endpoints must be used to identify the type of transaction that occurred.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x91
64-bit source address	4-11	MSB first, LSB last. The sender's 64-bit address.
Reserved	12-13	Reserved.
Source endpoint	14	Endpoint of the source that initiates transmission.
Destination endpoint	15	Endpoint of the destination where the message is addressed.
Cluster ID	16-17	The Cluster ID where the frame is addressed.
Profile ID	18-19	The Profile ID where the fame is addressed.
Receive options	14	Bit field: bit 0 = Packet acknowledged bit 1 = Packet was a broadcast packet bits 6 and 7: b'01 = Point-Multipoint b'10 = Repeater mode (directed broadcast) Ignore all other bits.
Received data	21-n	Received RF data.

Example

In the following example, a device with a 64-bit address of 0x0013A200 40522BAA sends a broadcast data transmission to a remote device with payload RxData.

If a device sends the transmission:

- With source and destination endpoints of 0xE0
- Cluster ID = 0x2211
- Profile ID = 0xC105

If **AO** = **1** on the receiving device, it sends the following frame out its serial interface.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x18
Frame type	3	0x91
64-bit source address	MSB 4	0x00
	5	0x13
	6	0xA2
	7	0x00
	8	0x40
	9	0x52
	10	0x2B
	LSB 11	0xAA
Reserved	12	0xFF
	13	0xFE
Source endpoint	14	0xE0
Destination endpoint	15	0xE0
Cluster ID	16	0x22
	17	0x11
Profile ID	18	0xC1
	19	0x05
Receive options	20	0x02
Received data	21	0x52
	22	0x78
	23	0x44
	24	0x61
	25	0x74
	26	0x61

Remote Command Response frame - 0x97

Description

If a device receives this frame in response to a Remote Command Request (0x17) frame, the device sends an AT Command Response (0x97) frame out the serial interface.

Some commands, such as the **ND** command, may send back multiple frames.

Format

The following table provides the contents of the frame. For details on frame structure, see API frame format.

Frame data fields	Offset	Description
Frame type	3	0x97
Frame ID	4	This is the same value passed in to the request. If Frame ID = 0 in the associated request frame the device does not deliver a response frame.
64-bit source (remote) address	5-12	The address of the remote device returning this response.
Reserved	13-14	Reserved.
AT commands	15-16	The name of the command.
Command status	17	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
Command data	18-n	The value of the requested register.

Example

If a device sends a remote command 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 following example.

Frame data fields	Offset	Example
Start delimiter	0	0x7E
Length	MSB 1	0x00
	LSB 2	0x13
Frame type	3	0x97
Frame ID	4	0x55

Frame data fields	Offset	Example
64-bit source (remote) address	MSB 5	0x00
	6	0x13
	7	0xA2
	8	0x00
	9	0x40
	10	0x52
	11	0x2B
	LSB 12	0xAA
Reserved	13	0xFF
	14	0xFE
AT commands	15	0x53
	16	0x4C
Command status	17	0x00
Command data	18	0x40
	19	0x52
	20	0x2B
	21	0xAA
Checksum	22	0xF4

Advanced application features

Network commissioning and diagnostics

We call the process of discovering and configuring devices in a network for operation, "network commissioning." Devices include several device discovery and configuration features. In addition to configuring devices, you must develop a strategy to place devices to ensure reliable routes. To accommodate these requirements, modules include features to aid in placing devices, configuring devices, and network diagnostics.

Local configuration

You can configure devices locally using serial commands in Transparent or API mode, or remotely using remote API commands. Devices that are in API mode can send configuration commands to set or read the configuration settings of any device in the network.

Remote configuration

When you do not have access to the device's serial port, you can use a separate device in API mode to remotely configure it. To remotely configure devices, use the following steps.

Send a remote command

To send a remote command, populate the Remote AT Command Request frame - 0x17 with:

1. The 64-bit address of the remote device.
2. The correct command options value.
3. Optionally, the command and parameter data.
4. If you want a command response, set the Frame ID field to a non-zero value.

The firmware only supports unicasts of remote commands. You cannot broadcast remote commands.

XCTU has a Frames Generator tool that can assist you with building and sending a remote AT frame; see Frames generator tool in the *XCTU User Guide*.

Apply changes on remote devices

When you use remote commands to change the command parameter settings on a remote device, you must apply the parameter changes or they do not take effect. For example, if you change the **BD** parameter, the actual serial interface rate does not change on the remote device until you apply the changes. You can apply the changes using remote commands in one of three ways:

1. Set the apply changes option bit in the API frame.
2. Send an **AC** command to the remote device.
3. Send the **WR** command followed by the **FR** command to the remote device to save the changes and reset the device.

Remote command response

If a local device sends a command request to a remote device, and the API frame ID is non-zero, the remote device sends a remote command response transmission back to the local device.

When the local device receives a remote command response transmission, it sends a remote command response API frame out its UART. The remote command response indicates:

1. The status of the command, which is either success or the reason for failure.
2. In the case of a command query, it includes the register value.

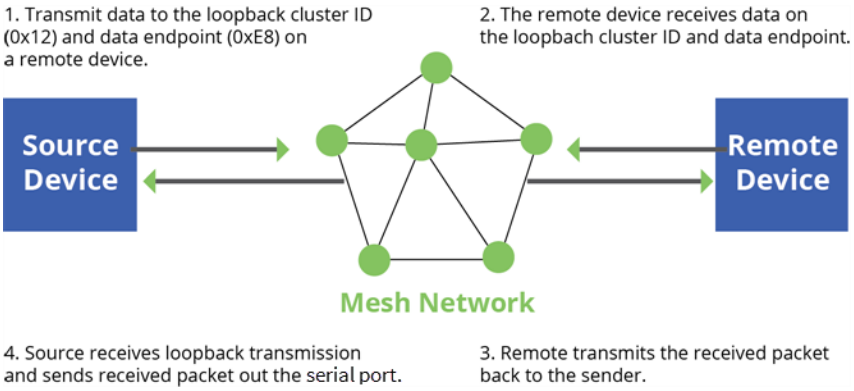
The device that sends a remote command does not receive a remote command response frame if:

1. It could not reach the destination device.
2. You set the frame ID to 0 in the remote command request.

Test links in a network - loopback cluster

To measure the performance of a network, you can send unicast data through the network from one device to another to determine the success rate of several transmissions. To simplify link testing, the devices support a Loopback cluster ID (0x12) on the data endpoint (0xE8). The cluster ID on the data endpoint sends any data transmitted to it back to the sender.

The following figure demonstrates how you can use the Loopback cluster ID and data endpoint to measure the link quality in a mesh network.



The configuration steps for sending data to the loopback cluster ID depend on what mode the device is in. For details on setting the mode, see API Mode. The following sections list the steps based on the device's mode.

Transparent operating mode configuration (AP = 0)

To send data to the loopback cluster ID on the data endpoint of a remote device:

1. Set the **CI** command to **0x12**.
2. Set the **DH** and **DL** commands to the address of the remote device.

After exiting Command mode, the device transmits any serial characters it received to the remote device, which returns those characters to the sending device.

API operating mode configuration (AP = 1 or AP = 2)

Send an Explicit Addressing Command frame - 0x11 using **0x12** as the cluster ID and **0xE8** as both the source and destination endpoint.

The remote device echoes back the data packets it receives to the sending device.

Test links between adjacent devices

It often helps to test the quality of a link between two adjacent modules in a network. You can use the Test Link Request Cluster ID to send a number of test packets between any two devices in a network.

To clarify the example, we refer to "device A" and "device B" in this section.

To request that device B perform a link test against device A:

1. Use device A in API mode (**AP** = **1**) to send an Explicit Addressing Command (0x11) frame to device B.
2. Address the frame to the Test Link Request Cluster ID (0x0014) and destination endpoint: 0xE6.
3. Include a 12-byte payload in the Explicit Addressing Command frame with the following format:

Number of bytes	Field name	Description
8	Destination address	The address the device uses to test its link. For this example, use the device A address.
2	Payload size	The size of the test packet. Use the **NP** command to query the maximum payload size for the device.
2	Iterations	The number of packets to send. This must be a number between 1 and 4000.

4. Device B should transmit test link packets.
5. When device B completes transmitting the test link packets, it sends the following data packet to device A's Test Link Result Cluster (0x0094) on endpoint (0xE6).
6. Device A outputs the following information as an API Explicit RX Indicator (0x91) frame:

Number of bytes	Field name	Description
8	Destination address	The address the device used to test its link.
2	Payload size	The size of the test packet device A sent to test the link.
2	Iterations	The number of packets that device A sent.
2	Success	The number of packets that were successfully acknowledged.
2	Retries	The number of MAC retries used to transfer all the packets.
1	Result	0x00 - the command was successful. 0x03 - invalid parameter used.
1	RR	The maximum number of MAC retries allowed.
1	maxRSSI	The strongest RSSI reading observed during the test.
1	minRSSI	The weakest RSSI reading observed during the test.
1	avgRSSI	The average RSSI reading observed during the test.

Example

Suppose that you want to test the link between device A (**SH**/**SL** = 0x0013A200 40521234) and device B (**SH**/**SL**=0x0013A 200 4052ABCD) by transmitting 1000 40-byte packets:

Send the following API packet to the serial interface of device A.

In the following example packet, whitespace marks fields, bold text is the payload portion of the packet:

7E 0020 11 01 0013A20040521234 FFFE E6 E6 0014 C105 00 00 **0013A2004052ABCD 0028 03E8** EB

When the test is finished, the following API frame may be received:

7E 0027 91 0013A20040521234 FFFE E6 E6 0094 C105 00 **0013A2004052ABCD 0028 03E8 03E7 0064 00 0A 50 53 52** 9F

This means:

- 999 out of 1000 packets were successful.
- The device made 100 retries.
- **RR** = 10.
- maxRSSI = -80 dBm.
- minRSSI = -83 dBm.
- avgRSSI = -82 dBm.

If the Result field does not equal zero, an error has occurred. Ignore the other fields in the packet.

If the Success field equals zero, ignore the RSSI fields.

The device that sends the request for initiating the Test link and outputs the result does not need to be the sender or receiver of the test. It is possible for a third node, "device C", to request device A to perform a test link against device B and send the results back to device C to be output. It is also possible for device B to request device A to perform the previously mentioned test. In other words, the frames can be sent by either device A, device B or device C and in all cases the test is the same: device A sends data to device B and reports the results.

RSSI indicators

The received signal strength indicator (RSSI) measures the amount of power present in a radio signal. It is an approximate value for signal strength received on an antenna.

You can use the **DB** command to measure the RSSI on a device. **DB** returns the RSSI value measured in -dBm of the last packet the device received. This number can be misleading in multi-hop DigiMesh networks. The **DB** value only indicates the received signal strength of the last hop. 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.

To determine the **DB** value in hardware:

1. Use the RSSI module pin (pin 36). When the device receives data, it sets the RSSI PWM duty cycle to a value based on the RSSI of the packet it receives.

This value only indicates the quality of the last hop of a multi-hop transmission. You could connect this pin to an LED to indicate if the link is stable or not.

Discover all the devices on a network

You can use the **ND** (Network Discovery) command to discover all devices on a network. When you send the **ND** command:

1. The device sends a broadcast **ND** command through the network.
2. All devices that receive the command send a response that includes their addressing information, node identifier string and other relevant information. For more information on the node identifier string, see NI command.

ND is useful for generating a list of all device addresses in a network.

When a device receives the network discovery command, it waits a random time before sending its own response. You can use the **NT** command to set the maximum time delay on the device that you use to send the **ND** command.

- The device that sends the **ND** includes its **NT** setting in the transmission to provide a delay window for all devices in the network.
- The default **NT** value is 0x82 (13 seconds).

Discover devices within RF range

- You can use the **FN** (Find Neighbors) command to discover the devices that are immediate neighbors (within RF range) of a particular device.
- **FN** is useful in determining network topology and determining possible routes.

You can send **FN** locally on a device in Command mode or you can use a local AT Command frame - 0x08.

To use **FN** remotely, send the target node a Remote AT Command Request frame - 0x17 using **FN** as the name of the AT command.

The device you use to send **FN** transmits a zero-hop broadcast to all of its immediate neighbors. All of the devices that receive this broadcast send an RF packet to the device that transmitted the **FN** command. If you sent **FN** remotely, the target devices respond directly to the device that sent the **FN** command. The device that sends **FN** outputs a response packet in the same format as an AT Command Response frame - 0x88.

General Purpose Flash Memory

XLR PROs provide 119 512-byte blocks of flash memory that an application can read and write to. This memory provides a non-volatile data storage area that an application uses for many purposes. Some common uses of this data storage include:

- Storing logged sensor data
- Buffering firmware update data for a host microcontroller
- 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 update file for over-the-air firmware updates of the device itself.

Access General Purpose Flash Memory

To access the GPM of a target node locally or over-the-air, send commands to the MEMORY_ACCESS cluster ID (0x23) on the DIGI_DEVICE endpoint (0xE6) of the target node using explicit API frames. For a description of Explicit API frames, see Operate in API mode.

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

Byte offset in payload	Number of bytes	Field name	General field description
0	1	GPM_CMD_ID	Specific GPM commands are described in detail in the topics that follow.
1	1	GPM_OPTIONS	Command-specific options.
2	2*	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, or in the case of a READ, the number of bytes requested.
8	varies	GPM_DATA	
* Specify multi-byte parameters with big-endian byte ordering.			

When a device sends a GPM command to another device via a unicast, the receiving device sends a unicast response back to the requesting device's source endpoint specified in the request packet. It does not send a response for broadcast requests. If the source endpoint is set to the DIGI_DEVICE endpoint (0xE6) or Explicit API mode is enabled on the requesting device, then the requesting node outputs a GPM response as an explicit API RX indicator frame (assuming it has API mode enabled).

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

Byte offset in payload	Number of bytes	Field name	General field description
0	1	GPM_CMD_ID	This field is the same as the request field.
1	1	GPM_STATUS	Status indicating whether the command was successful.
2	2*	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	
* Specify multi-byte parameters with big-endian byte ordering.			

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	Should be set to PLATFORM_INFO_REQUEST (0x00).
GPM_OPTIONS	This field is unused for this command. Set to 0.
GPM_BLOCK_NUM	This field is unused for this command. Set to 0.
GPM_START_INDEX	This field is unused for this command. Set to 0.
GPM_NUM_BYTES	This field is unused for this command. Set to 0.
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 sends a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Should be set to PLATFORM_INFO (0x80).
GPM_STATUS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
GPM_BLOCK_NUM	Indicates the number of GPM blocks available.
GPM_START_INDEX	Indicates the size, in bytes, of a GPM block.
GPM_NUM_BYTES	The number of bytes in the GPM_DATA field. For this command, this field will be set to 0.
GPM_DATA	No data bytes are specified for this command.

Example

A PLATFORM_INFO_REQUEST sent to a device 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

ERASE (0x01)

The ERASE command erases (writes all bits to binary 1) one or all of the GPM flash blocks. You can also use the ERASE command to erase all blocks of the GPM by setting the GPM_NUM_BYTES field to 0.

Field name	Command-specific description
GPM_CMD_ID	Should be set to ERASE (0x01).
GPM_OPTIONS	There are currently no options defined for the ERASE command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block that should be erased. When erasing all GPM blocks, this field is ignored (set to 0).
GPM_START_INDEX	The ERASE command only works on complete GPM blocks. The command cannot be used to erase part of a GPM block. For this reason GPM_START_INDEX is unused (set to 0).
GPM_NUM_BYTES	Setting GPM_NUM_BYTES to 0 has a special meaning. It indicates that every flash block in the GPM should be erased (not just the one specified with GPM_BLOCK_NUM). In all other cases, the GPM_NUM_BYTES field should be set to the GPM flash block size.
GPM_DATA	No data bytes are specified for this command.

ERASE_RESPONSE (0x81)

When an ERASE 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	Should be set to ERASE_RESPONSE (0x81).
GPM_STATUS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
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	The number of bytes in the GPM_DATA field. For this command, this field will be set to 0.
GPM_DATA	No data bytes are specified for this command.

Example

To erase flash block 42 of a target radio with serial number of 0x0013a200407402ac format an ERASE packet 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	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	Description
GPM_CMD_ID	Set to WRITE_RESPONSE (0x82) or ERASE_THEN_WRITE_ RESPONSE (0x83).
GPM_OPTIONS	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.
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)

You can use the READ command 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	Should be set to READ (0x04).
GPM_OPTIONS	There are currently no options defined for this command. Set this field to 0.
GPM_BLOCK_NUM	Set to the index of the GPM block that should be read.
GPM_START_INDEX	Set to the byte index within the GPM block where the given data should be read.
GPM_NUM_BYTES	Set to the number of data bytes to be read. Only one GPM block can be operated on per command. For this reason, GPM_START_INDEX + GPM_NUM_BYTES cannot be greater than the GPM block size. The number of bytes sent in an explicit API frame (including the GPM command fields) cannot exceed the maximum payload size of the device. You can query the maximum payload size with the **NP** AT command.
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	Should be set to READ_RESPONSE (0x84).
GPM_STATUS	A 1 in the least significant bit indicates an error occurred. All other bits are reserved at this time.
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	The number of bytes in the GPM_DATA field.
GPM_DATA	The bytes read from the GPM block specified.

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)

Use the FIRMWARE_VERIFY and FIRMWARE_VERIFY_AND_INSTALL commands when remotely updating firmware on a device. For more inoformation about firmware updates. These commands check if the GPM 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 device resets and begins using the new firmware.

Field name	Command-specific description
GPM_CMD_ID	Should be set to FIRMWARE_VERIFY (0x05) or FIRMWARE_ VERIFY_AND_INSTALL (0x06)
GPM_OPTIONS	There are currently no options defined for this command. Set this field to 0.
GPM_BLOCK_NUM	This field is unused for this command. Set to 0.
GPM_START_INDEX	This field is unused for this command. Set to 0.
GPM_NUM_BYTES	This field is unused for this command. Set to 0.
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 sends a response in the following format to the source endpoint specified in the requesting frame.

Field name	Command-specific description
GPM_CMD_ID	Should be set to FIRMWARE_VERIFY_RESPONSE (0x85)
GPM_STATUS	A 1 in the least significant bit indicates the GPM does not contain a valid firmware image. A 0 in the least significant bit indicates the GPM does contain a valid firmware image. All other bits are reserved at this time.
GPM_BLOCK_NUM	This field is unused for this command. Set to 0.
GPM_START_INDEX	This field is unused for this command. Set to 0.
GPM_NUM_BYTES	This field is unused for this command. Set to 0.
GPM_DATA	This field is unused for this command

Work 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 (OTA) firmware updates

XLR PROs provide two methods of firmware update:

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

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

Over-the-air firmware updates can be sent to a remote node using a local node and XCTU, or an external application can be programmed to follow this process. There are three phases of the over-the-air update process: distributing the new application, verifying the new application, and installing

the new application. In the following section, the node to be updated 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.

Distribute the new application

The first phase of performing an over-the-air update on a device is transferring the new firmware file to the target node. Load the new firmware image in the target node's GPM prior to installation. XLR PROs use an encrypted binary (.ebin) file for both serial and over-the-air firmware updates. These firmware files are available on the Digi Support website and via XCTU.

Send the contents of the .ebin file to the target device using general purpose memory WRITE commands. Erase the entire GPM 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 1003 has an .ebin file of 1,048,576 bytes in length. Based on using a recommended packet size of 1024 bytes, sending a packet every 30 seconds minimized network disruption. For this reason, the .ebin should be divided and addressed as follows:

GPM_BLOCK_NUM	BPM_START_INDEX	BPM_NUM_BYTES	.ebin bytes
0	0	102	0 to 1023
0	1024	1024	1024 to 2047
0	2048	1024	2048 to 3071
0	3072	1024	3072 to 4095
1	0	1024	4096 to 5119
1	1024	1024	5120 to 6143
1	2048	1024	6144 to 7167
1	3072	1024	7168 to 8191
-	-	-	-
-	-	-	-
-	-	-	-
255	0	1024	1044480 to 1045503
255	1024	1024	1045504 to 1046527
255	2048	1024	1046528 to 1047551
255	3072	1024	1047552 to 1048575

Verify 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 reports if the uploaded data is invalid. If the data is valid, it begins installing the application. No installation takes place on invalid data.

Install 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, the XLR PRO will install the new firmware. This installation process can take up to 8 seconds. During the installation the XLR PRO will be unresponsive to both serial and RF communication. To complete the installation, the target XLR PRO will reset. Any AT parameter settings which have not been written to flash using the **WR** (write) command will be lost.

Keep in mind

The firmware upgrade process requires that the XLR PRO resets itself and parameters which have not been written to flash will be lost after the reset. To avoid this, write all parameters with the **WR** (write) 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 an XLR PRO in either case.

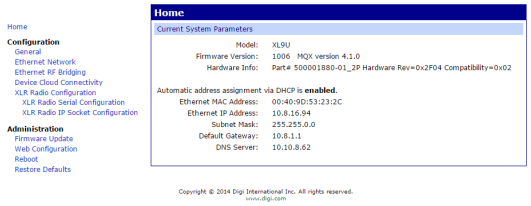
Configure the XLR PRO using the web configuration interface

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. Once an address is assigned, use XCTU to view the assigned address. See Configure the XLR PRO using XCTU for details on using XCTU.

To access the XLR PRO web configuration interface:

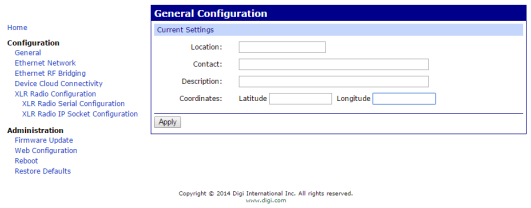
1. Open a browser and go to the XLR PRO IP address.
2. When you are prompted to do so, enter the username and password for the web configuration interface. The default username is **admin** and the default password is **password**. The XLR PRO Configuration and Management Home page appears.



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.

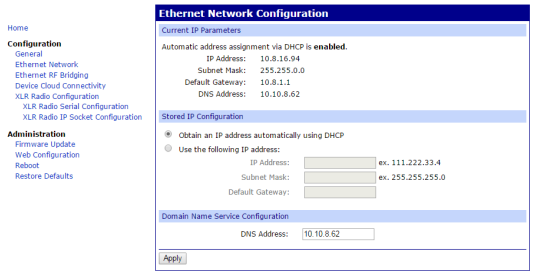


2. Enter the following:
   - **Location:** Enter a location for the XLR PRO.
   - **Contact:** Enter contact information for the XLR PRO.
   - **Description:** Enter a description for the XLR PRO.
   - **Coordinates:** Enter latitude and longitude for the XLR PRO.
3. Click **Apply**.

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.

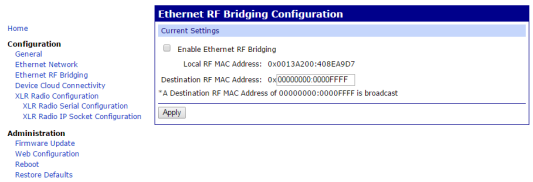
   Note The default DNS address is 0.0.0.0. If you use the default DNS address and static mode is configured, OpenDns servers are used.

5. Click **Apply**.

Set Ethernet 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 **Ethernet RF Bridging**. The Ethernet 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.

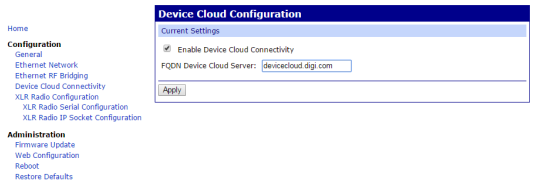
- **Destination RF MAC Address:** If bridging is enabled, enter a destination RF MAC address. A destination address of 00000000:000FFFF is broadcast.

3. Click **Apply**.

Set Device Cloud connectivity 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.

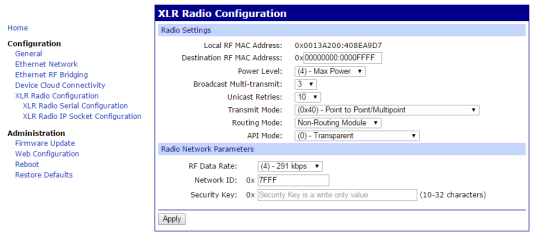


2. Select the following:
   - **Enable Device Cloud Connectivity:** Select this option to enable Device Cloud connectivity; deselect this option to disable Device Cloud connectivity.
   - **FQDN Device Cloud Server:** If Device Cloud connectivity is enabled, enter the fully qualified name of the Device Cloud server. The default is devicecloud.digi.com, which is the default Device Cloud server.
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.

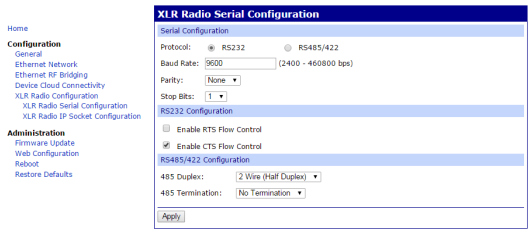
- **Broadcast Multi-transmit:** Enter the number of broadcast retransmissions, from 0 to 8 retransmissions. If you enter 0, broadcasts are transmitted only once; if you enter 8, a given broadcast can be retransmitted up to 8 times. The default is 3.
- **Unicast Retries:** Enter the maximum number of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



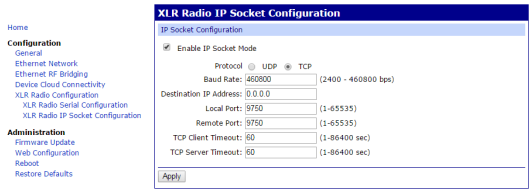
2. Set serial configuration options:
   - **Protocol:** Select RS-232 or RS-485/RS-422.
   - **Baud Rate:** Enter a baud rate:
   - **For RS-232:** Enter a baud rate from 2400 through 460800. The default is 9600.
   - **For RS-485/RS-422:** Enter a baud rate from 2400 to 921000. The default is 9600.
   - **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 IP socket (Ethernet) options

To set XLR radio IP socket (Ethernet) configuration options:

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

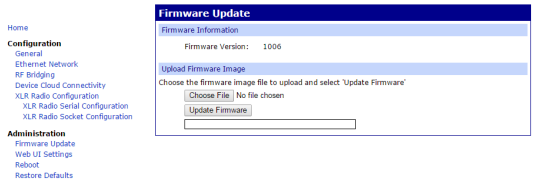


2. Enable or disable IP socket mode:
   - To enable socket mode, select **Enable IP Socket Mode**.
   - To disable socket mode, deselect **Enable Socket Mode**.
3. If socket mode is enabled, enter the following:
   - **Protocol:** Select UDP or TCP.
   - **Baud Rate:** Enter the baud rate for socket 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.
   - **TCP Client Timeout:** Enter the number of seconds the client waits without any traffic before closing the connection. The default is 60 seconds.
   - **TCP 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.

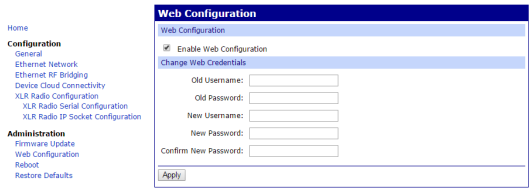


3. Click **Choose File**.
4. Browse to and select an XLR PRO firmware file. XLR PRO firmware files have the **.ebin** extension.
5. After selecting a firmware file, click **Update Firmware**. XLR PRO firmware is updated and the XLR PRO resets. (The amount of time needed to update the XLR PRO depends on the speed of your network.) Approximately 30 seconds after the XLR PRO resets, the XLR PRO is operational.

Set web configuration 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**.

Configure the XLR PRO using XCTU

Download and install XCTU

For XLR PRO support, make sure you install XCTU version 6.2 or later.

To download and install XCTU:

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

Download and install the USB driver

To download and install the XLR PRO USB driver:

1. Go to the Drivers section of the product page.
2. Launch the XLR PRO USB driver installer and follow the prompts on the installation screens.

Connect XLR PRO to your PC

To connect XLR PRO to your PC:

1. Connect the XLR PRO mini USB port to an available USB port on your PC using the mini USB cable.
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 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 AT commands.

To change a configuration parameter:

1. Locate the parameter in the XCTU configuration display.
2. Use the **Search** function in the upper right corner to quickly locate a parameter.
3. Select a new value for the parameter.
4. 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

By default, the XLR PRO gets an IP address automatically assigned using DHCP.

Determine the assigned DHCP address

To determine the assigned DHCP address:

1. Connect the XLR PRO to the network via either Ethernet port on the XLR PRO front panel.
2. Wait approximately one minute. An IP address should be automatically assigned within one minute. If a DHCP server hasn't assigned an IP address within that time period, the XLR PRO assigns an IP address using the following format:

   169.254.xxx.yyy

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

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

Assign a static IP address

To assign a static IP address:

1. Set the following XLR PROparameters:
   - **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.
2. After setting the parameters, write the new parameter values to save the settings.

Update firmware with XCTU

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.
   - Select the COM port to which the XLR PRO serial port is connected.
   - Select the baud rate (9600 8-N-1 by default).
   - Switch baud rate to 115200b/s to reduce the time required to update the firmware.
   - 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**.

Configure the XLR PRO using Digi Remote Manager

Use Digi Remote Manager (https://remotemanager.digi.com/) to perform the operations in this section. Each operation requires that you enable Remote Manager with the **DO** command and that you connect the XLR PRO to an access point that has an external Internet connection to allow access to Digi Remote Manager.

Note Digi is consolidating our cloud services, Digi Device Cloud and Digi Remote Manager®, under the Remote Manager name. This phased process does not affect device functionality or the functionality of the web services and other features. However, customers will find that some user interface and firmware functionality mention both Device Cloud and Digi Remote Manager.

Create a Remote Manager account

Digi Remote Manager is an on-demand service with no infrastructure requirements. Remote devices and enterprise business applications connect to Remote Manager through standards-based web services. This section describes how to configure and manage an XLR PRO using Remote Manager. For detailed information on using Remote Manager, refer to the Remote Manager User Guide, available via the Documentation tab in Remote Manager.

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

1. Go to https://www.digi.com/products/cloud/digi-remote-manager.
2. Click **30 DAY FREE TRIAL/LOGIN**.
3. Follow the online instructions to complete account registration. You can upgrade your Developer account to a paid account at any time.

When you are ready to deploy multiple XLR PROs in the field, upgrade your account to access additional Remote Manager features.

Get the XLR PRO MAC address

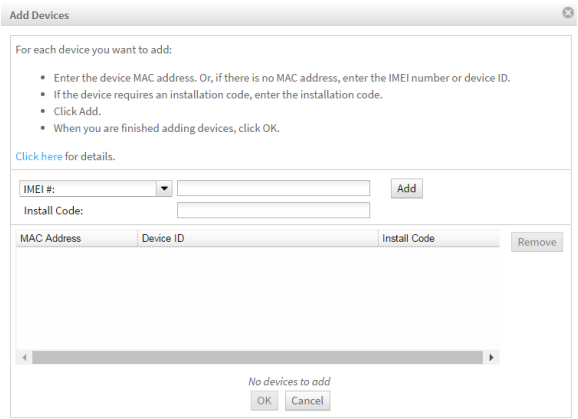
Before adding an XLR PRO to your Device Cloud account inventory, you need to determine the Ethernet MAC address for the device. You can view the Ethernet MAC address by querying the **%M** parameter using either XCTU or the Web UI. For information on using XCTU, see Configure the XLR PRO using XCTU; for information on using the Web UI, see Configure the XLR PRO using the web configuration interface.

Add a XLR PRO to Remote Manager

To add an XLR PRO to your Remote Manager account inventory, follow these steps:

Go to https://remotemanager.digi.com/.

1. Log in to your account
2. Click **Device Management** > **Devices**.
3. Click **Add Devices**. The Add Devices dialog appears.
4. Select **IMEI #**, and type or paste the IMEI number of the XLR PRO you want to add.



5. Click **Add** to add the device. The XLR PRO is added to your inventory.
6. Click **OK** to close the Add Devices dialog and return to the Devices view.

Update firmware with Remote Manager

To update firmware using Remote Manager:

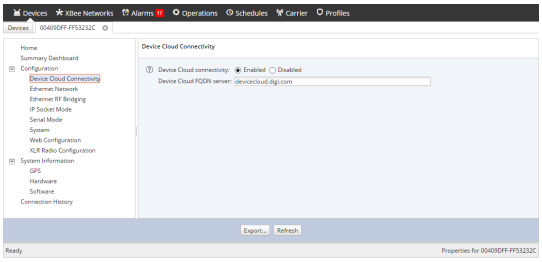
1. After logging on to Remote Manager, select the **Device Management** tab.
2. Right click on the device to update. (Device Type should be XLR PRO and the Device ID matches the MAC address of the XLR PRO to be updated.)
3. In the drop-down box, select **Firmware >Update Firmware**.
4. Click **Browse** to select the **.ebin** firmware image file to be loaded.
5. Click **Update Firmware**. A progress appears. Loading the firmware file takes about 100 seconds

After fifteen seconds, the XLR PRO power LED illuminates. After about 30 seconds, the new firmware is fully operational and running.

Configure Device Cloud connectivity options

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

1. Click Device **Management > Devices**.
2. Double-click the XLR PRO you need to configure.
3. Click **Configuration > Device Cloud Connectivity**.

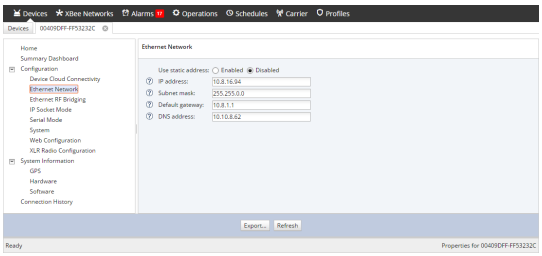


4. Set options for Device Cloud:
   - **Enabled/Disabled:** Select Enabled to enable Device Cloud management or select Disabled to disable Device Cloud management. The default is enabled.
   - **Device Cloud FQDN server:** If Device Cloud management is enabled, enter the FQDN of the Device Cloud server. The default FQDN is the US server: devicecloud.digi.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 Device Cloud configuration changes.

Configure Ethernet network 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 the XLR PRO you need to configure.
3. Click **Configuration > Ethernet Network**.



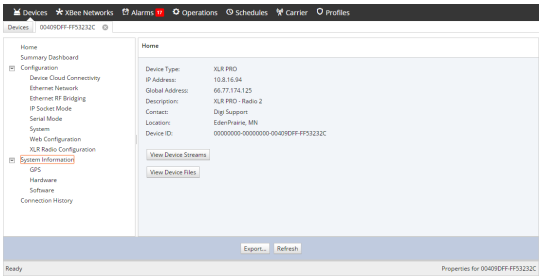
4. Use static or dynamic addressing:
   - To use a static address, enable **Use static address**.
   - To obtain an IP address automatically using DHCP, disable **Use static address**.
5. For static addressing only, enter the following:
   - **IP address:** Enter an IP address to assign to the XLR PRO 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.
6. For static or DHCP 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.

   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.
7. 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 Device Cloud configuration changes.

Configure Ethernet RF bridging options

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

1. Click **Device Management > Devices**.
2. Double-click the XLR PRO you want to configure.
3. Click **Configuration > Ethernet RF Bridging**.

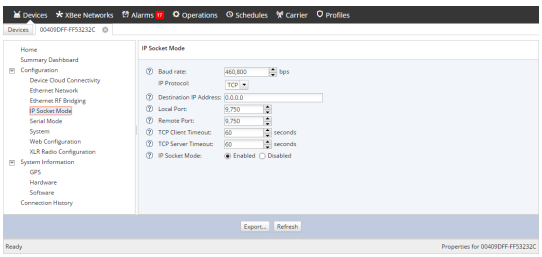


4. Set options for bridging:
   - **Enabled/Disabled:** Select Enabled to enable bridging or select Disabled to disable bridging. The default is disabled.
   - **Destination RF MAC address:** If bridging is enabled, enter the destination RF MAC address for the destination XLR PRO.
5. Apply the configuration 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 Device Cloud configuration changes.

Configure IP socket mode options

The IP Socket Mode 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 the XLR PRO you need to configure.
3. Click **Configuration > IP Socket Mode**.

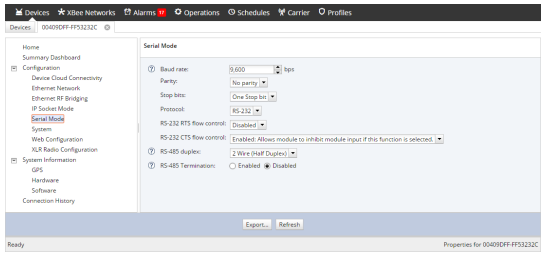


4. Set IP Socket Mode options:
   - **Baud rate:** Select the baud rate for IP to XLR traffic. The default is 460800.
   - **IP Protocol:** Select the IP protocol to use: TCP or UDP. The default is TCP.
   - **Destination IP address:** Enter the IPv4 destination address. The default is 0.0.0.0.
   - **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 UDP 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.
   - **IP 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 Device Cloud configuration changes.

Configure serial mode options

The Serial Mode options determine how data is transmitted from the serial port to the XLR radio. To change Serial to XLR options:

1. Click **Device Management > Devices**.
2. Double-click the XLR PRO you want to configure.
3. Click **Configuration > Serial Mode**.



4. Set Serial Mode 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.

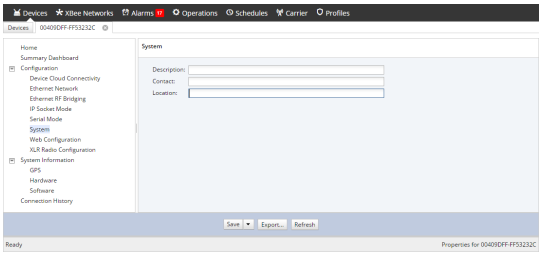
- **Stop bits:** Select the stop bits to use: one or two. The default is one stop bit.
- **Protocol:** Select the protocol: RS-232 or RS-485/RS-422. The default is RS-232.
- **RS-232 RTS flow control:** Select RTS flow control for RS-232: disabled or UART clear to send. The default is disabled.
- **RS-232 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-485 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 Device Cloud configuration changes.

Configure system options

The System options determine the contact information and location of the XLR radio. To change System options:

1. Click **Device Management > Devices**.
2. Double-click the XLR PRO you want to configure.
3. Click **Configuration > System**.



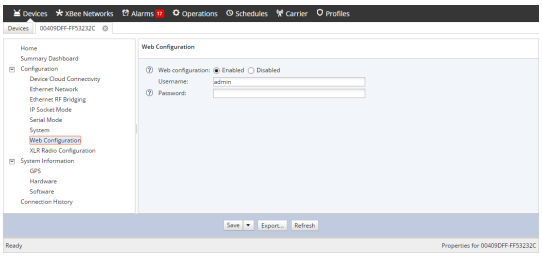
4. Set system options:
   - **Description:** Enter a description for the XLR PRO radio.
   - **Contact:** Enter contact information for the XLR PRO radio.
   - **Location:** Enter the location of the XLR PRO radio.

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 Device Cloud configuration changes.

Configure web configuration options

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

1. Click **Device Management > Devices**.
2. Double-click the XLR PRO you need to configure.
3. Click **Configuration > Web Configuration**.



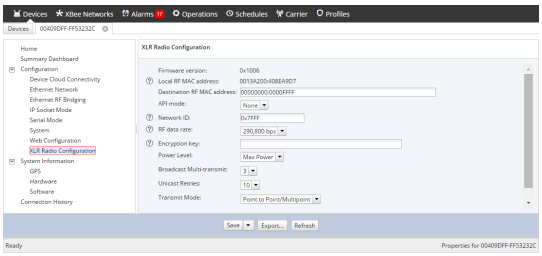
4. Set HTTP server options:
   - **Enabled/Disabled:** Select Enabled to enable the HTTP server or select Disabled to disable the HTTP server. The default is enabled.
   - **User Name/Password:** If the HTTP server is enabled, you can set a user name and password to control access to the server. The 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 Device Cloud configuration changes.

Configure XLR radio configuration options

The XLR radio configuration options determine several important configuration parameters for the XLR radio. To change XLR Radio Configuration options:

1. Click **Device Management > Devices**.
2. Double-click the XLR PRO you want to configure.

3. Click **Configuration > XLR Radio Configuration**.



4. Set XLR radio configuration options:
   - **Firmware version:** Displays the current firmware version. This value cannot be changed.
   - **Local RF MAC address:** Displays the serial number. This value cannot be changed.
   - **Destination RF MAC address:** Enter the destination RF MAC address for the XLR PRO. The default is 00000000:0000FFFF.
   - **API mode:** 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 b/s.
   - **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.

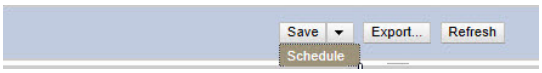
5. Apply the configuration changes:
   - If the XLR PRO is currently connected to Device Cloud, click **Save** to immediately apply configuration settings.
   - If the XLR PRO is currently not connected to Device Cloud, use the **Schedule** option to schedule when to apply configuration changes to the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

Schedule Device Cloud configuration changes

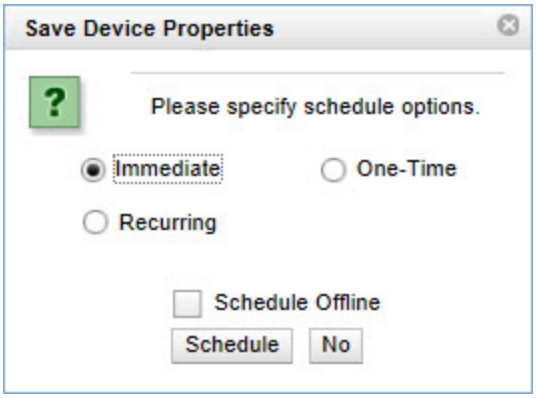
After changing one or more configuration options, you can immediately apply the changes to the XLR PRO device or schedule a time for applying the configuration changes.

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.

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

This section contains troubleshooting steps for the XLR PRO.

Serial interface issues

Condition

No data on serial port.

Solution

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 IP socket mode instead of the serial port for serial communication.

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

Condition

RS-485/422 does not work.

Solution

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

Condition

LAN stopped working when XLR PRO was plugged in.

Solution

With Ethernet RF bridging enabled, the XLR PRO acts like an Ethernet cable. If two XLR PRO 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

Condition

All of the LEDs are lit on the XLR PRO.

Solution

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 Antennas.

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 (10 inches) for USA and 34 cm for Canada.

Class 1, Division 2 (C1D2) 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.

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

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

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.

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

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

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.

ISED (Innovation, Science and Economic Development Canada) certification

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

⚠	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.
⚠	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.

Australia certification

This radio module complies with requirements to be operated in end products in Australia. All products with EMC and radio communications must have a registered RCM mark. Registration to use the compliance mark will only be accepted from Australian manufacturers or importers, or their agent, in Australia.

In order to have a RCM mark on an end product, a company must comply with a or b below.

- Have a company presence in Australia.
- Have a company/distributor/agent in Australia that will sponsor the importing of the end product.

Contact Digi for questions related to locating a contact in Australia.

Mexico IFETEL

Manufacturer: Digi International Inc.
Country: USA
Brand: Digi

The XLR Pro RF Modem for carries the following Mexican approval marks:

Mexico: 1502CE10266

IFT# RCPDIXL15-1494

Note Only the 2.1 dBi dipole antenna (A09-HATM-10) is approved for use with the modem in Mexico.

Brazil (ANATEL)

XLR PRO INTL radio modem complies with Brazil ANATEL standards in Resolution No.506.

ANATEL	0621-16-1209 (01)07899029306117
Este equipamento opera em caráter secundário, isto é, não tem direito a proteção contra interferência prejudicial, mesmo de estações do mesmo tipo, e não pode causar interferência a sistemas operando em caráter primário	

Antennas

Omni-directional antennas

All antenna part numbers followed by an asterisk (*) are not available from Digi. Consult with an antenna manufacturer for an equivalent option.

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
A09-QBMM-P61	OMNI	MMCX	1.9	Fixed/mobile	0

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
A09-Y15TM	15-Element Yagi	RPTNC	15.1	Fixed/Mobile	9.1
		Max gain	15.1		