

STREAM+

USER MANUAL



metOcean
telematics

 **iridium**[®]

Document No. CD-200266-00

Version No.: 2.1

Release Date: December 2024

Technical Support Direct: +1 902 468 2505 ext. 333
Toll Free: +1 800 565 1830 | Email: support@metocean.com

metocean.com

Table of Contents

COPYRIGHT INFORMATION	8
1. INTRODUCTION TO STREAM+	9
1.1. STREAM+ OVERVIEW	9
1.2. ABOUT THIS MANUAL	9
1.3. IRIDIUM SATELLITE NETWORK.....	9
1.4. SAFETY INSTRUCTIONS.....	11
2. PREREQUISITE	13
2.1. SETTING UP IRIDIUM COMMUNICATIONS	13
3. STREAM+ DEVELOPMENT KIT CONTENTS.....	14
4. DEVICE IDENTIFICATION.....	21
5. STREAM+ CONNECTORS.....	22
5.1. POWER	22
5.2. CONNECTING THE UNIT	23
5.3. LED INDICATOR.....	23
6. INTERFACING WITH STREAM+.....	25
6.1. GETTING STARTED	25
6.2. FIRST STREAM+ COMMAND	26
7. FIRMWARE UPGRADE	27
7.1. UPDATING STREAM+ FIRMWARE	27
8. STREAM+ API.....	30
8.1. FIRMWARE VERSIONING	30
8.1.1. <i>Migrating from v1.x.x to v2.x.x</i>	30
8.1.1.1. <i>Command Updates</i>	30
8.1.1.2. <i>General Command Formatting</i>	30
8.2. COMMAND AND RESPONSE FORMAT	30
8.2.1. <i>Commands</i>	30

8.2.2.	<i>Response</i>	31
8.2.3.	<i>Cyclic Redundancy Checks</i>	31
8.3.	COMMAND TABLE	32
8.3.1.	<i>Table format</i>	32
8.4.	GENERAL COMMANDS	33
8.4.1.	<i>reset</i>	33
8.4.2.	<i>serialConfig</i>	33
8.4.3.	<i>sysInfo</i>	34
8.4.4.	<i>sysTime</i>	35
8.4.5.	<i>temperature</i>	36
8.4.6.	<i>update</i>	36
8.4.7.	<i>voltage</i>	37
8.5.	FILESYSTEM COMMANDS	37
8.5.1.	<i>file</i>	37
8.5.2.	<i>fileDelete</i>	40
8.5.3.	<i>fileInfo</i>	41
8.5.4.	<i>fileList</i>	42
8.5.5.	<i>storageErase</i>	43
8.5.6.	<i>storageInfo</i>	44
8.6.	NETWORKING COMMANDS	45
8.6.1.	<i>retrieveFile</i>	45
8.6.2.	<i>sendFile</i>	45
8.6.3.	<i>sendFileCancel</i>	47
8.6.4.	<i>serverConfig</i>	47
8.6.5.	<i>serverFileList</i>	50
8.6.6.	<i>serverSyncFiles</i>	51
8.7.	IMT COMMANDS	52
8.7.1.	<i>imt</i>	52

8.7.2.	<i>imtConfig</i>	55
8.7.3.	<i>imtMoList</i>	56
8.7.4.	<i>imtTopicList</i>	56
8.8.	GNSS COMMANDS.....	57
8.8.1.	<i>gnss</i>	57
8.8.2.	<i>gnssConfig</i>	58
8.9.	OTHER COMMANDS.....	59
8.9.1.	<i>ppp</i>	59
8.9.2.	<i>iridium</i>	59
8.10.	API RESPONSES TO COMMANDS	61
9.	FTP IMPLEMENTATION	62
9.1.	FTPS OPTION	62
10.	IMT IMPLEMENTATION	63
10.1.	MT FILE NAMING	63
10.2.	FIRMWARE V1.X FILE NAMING.....	64
10.3.	IMT TOPIC LIST.....	64
11.	INTERNET PASS-THROUGH SETUP.....	65
11.1.	SERIAL PORT	65
11.2.	PPP COMMAND.....	65
11.3.	CHAT SCRIPT.....	66
11.4.	START PPPD	67
12.	CERTUS.....	69
12.1.	CERTUS SERVICE	69
12.2.	CERTUS IP DATA.....	69
13.	MOUNTING INSTRUCTIONS.....	70
13.1.	ANTENNA MOUNTING INSTRUCTIONS	72
13.1.1.	<i>Tools Required</i>	72
14.	MAINTENANCE	75

14.1.	O-RING MAINTENANCE	75
14.2.	O-RING FACTS	75
14.3.	O-RING CARE AND MAINTENANCE	76
15.	SPECIFICATIONS.....	77
15.1.	ELECTRICAL SPECIFICATIONS	77
15.2.	MECHANICAL SPECIFICATIONS	77
15.3.	PHYSICAL PRODUCT	78
15.4.	ENVIRONMENTAL SPECIFICATIONS	78
16.	ACRONYMS.....	79
17.	CERTIFICATIONS.....	80
18.	WARRANTY AND LIABILITY STATEMENTS.....	81
18.1.	LIABILITY STATEMENT	81
18.2.	WARRANTY	81
18.2.1.	<i>Warranty Terms</i>	81
18.2.2.	<i>Technical Support</i>	81
19.	APPENDIX A.....	82
19.1.	STREAM+ BLOCK DESCRIPTION	82
19.1.1.	<i>Power Control</i>	82
19.1.2.	<i>LED Control Jumpers</i>	83
19.1.3.	<i>STREAM+ Connector</i>	83
19.1.4.	<i>nShutdown Control</i>	83
19.1.5.	<i>STREAM+ I/O</i>	83
19.1.6.	<i>RS-485</i>	84
19.1.7.	<i>RS-232</i>	84
20.	APPENDIX B.....	85
20.1.	ANTENNA CONSIDERATIONS	85

Table of Figures

FIGURE 1: STREAM+ DEVICE	9
FIGURE 2: IRIDIUM SATELLITE NETWORK	10
FIGURE 3: STREAM+ DEVELOPMENT KIT	14
FIGURE 4: STREAM+ DEVELOPMENT KIT CONTENTS	15
FIGURE 5: STREAM+ DEVICE DIMENSIONS	16
FIGURE 6: TEST INTERFACE CARD	16
FIGURE 7: ANTENNA (IRIDIUM)	16
FIGURE 8: ANTENNA (GNSS)	17
FIGURE 9: CIGARETTE LIGHTER POWER CABLE	17
FIGURE 10: PIGTAIL POWER CONNECTOR	17
FIGURE 11: RS-232 CABLE	18
FIGURE 12: RS-232 TO USB CONVERTER CABLE	18
FIGURE 13: IRIDIUM ANTENNA CABLE	18
FIGURE 14: GNSS ANTENNA CABLE	19
FIGURE 15: DC WALL WART	19
FIGURE 16: PIGTAIL DATA CONNECTOR CABLE	19
FIGURE 17: USB TO RS485 SERIAL CONVERTER	20
FIGURE 18: DEVICE IDENTIFICATION	21
FIGURE 19: STREAM+ CONNECTORS	22
FIGURE 20: PANEL MOUNT FRONT VIEW	22
FIGURE 21: STREAM+ LED INDICATOR	23
FIGURE 22: STREAM+ TEST INTERFACE CARD	25
FIGURE 23: STREAM+ TEST INTERFACE CARD BLOCKS	25
FIGURE 24: ARCHITECTURE OF STREAM+ CERTUS SERVICE	69
FIGURE 25: MOUNTING HOLES LOCATION	70
FIGURE 26: HORIZONTAL MOUNTING	71
FIGURE 27: VERTICAL MOUNTING EXAMPLES	72
FIGURE 28: 65020-011 SCAN ANTENNA (IRIDIUM)	73
FIGURE 29: 2J ANTENNA (GPS)	74
FIGURE 30: LOCATION OF THE O-RING – ANTENNA (IRIDIUM)	75
FIGURE 31: STREAM+ PHYSICAL PRODUCT	78
FIGURE 32: BLOCK DESCRIPTION	82


List of Tables

TABLE 1: STREAM+ DEVELOPMENT KIT CONTENTS	14
TABLE 2: LED INDICATOR	24
TABLE 3: ELECTRICAL SPECIFICATIONS.....	77
TABLE 4: MECHANICAL SPECIFICATIONS.....	78
TABLE 5: ENVIRONMENTAL SPECIFICATIONS	78
TABLE 6: ANTENNA CONSIDERATIONS	85

Copyright Information

© Copyright 2024 – MetOcean Telematics

All Rights Reserved. STREAM+ User Manual must not be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of MetOcean Telematics.

 Information contained in this guide is subject to regular updates and changes.

MetOcean Telematics has patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from MetOcean Telematics; the furnishing of this document does not give the user any license to these patents, trademarks, copyrights, or other intellectual property.

Unauthorized duplication of any recordings whether downloaded from internet or made from audio CDs is a violation of copyright laws. The making of unauthorized copies of copy-protected material, including computer programmes, files, broadcasts, and sound recordings, may be an infringement of copyrights and constitute criminal offence. This equipment should not be used for such purposes.

1. Introduction to STREAM+

1.1. STREAM+ Overview

The STREAM+ is a versatile device that allows the users to send and receive data via the Iridium Satellite network anywhere on the planet. It is a physical device that utilizes the 9770 transceiver and is designed to be integrated into an overall maritime, land, defence, aerospace and IoT solution.

The STREAM+ offers a range of industry standard protocols, features, and inputs enabling end-users with flexible integration and installation options. STREAM+ simplifies data access by transferring user data to an Amazon Web Service (AWS) file management service.

Powered by the Iridium Certus 100 mid-band service and Iridium Messaging Transport (IMT), the STREAM+ allows users to send and receive files and messages reaching a few megabytes securely, removing the need to worry about intermittent connectivity disruptions.



Figure 1: STREAM+ Device

Use the STREAM+ development kit to start building and testing your custom made IOT/M2M solutions with the STREAM+ using its industry-standard interfaces, real-time GNSS positioning and reporting, and Iridium mid-band and messaging services.

1.2. About this Manual

The purpose of this User Manual is to provide information on the setup, powering the device, connecting the device, establishing serial communications, firmware upgrade, Certus services and technical specifications.

1.3. Iridium Satellite Network

Iridium operates the largest constellation of any satellite network; it consists of a fully meshed network of 66 low-earth orbiting (LEO) cross-linked satellites, with 9 in-orbit spares. This ensures coverage over the entire globe in a constellation of six polar planes.

Each plane has 11 mission satellites performing as nodes in the telephony network. The 9 additional satellites orbit as spares ready to replace any unserviceable satellite. This constellation ensures that every region on the globe is covered by at least one satellite at-all-times.

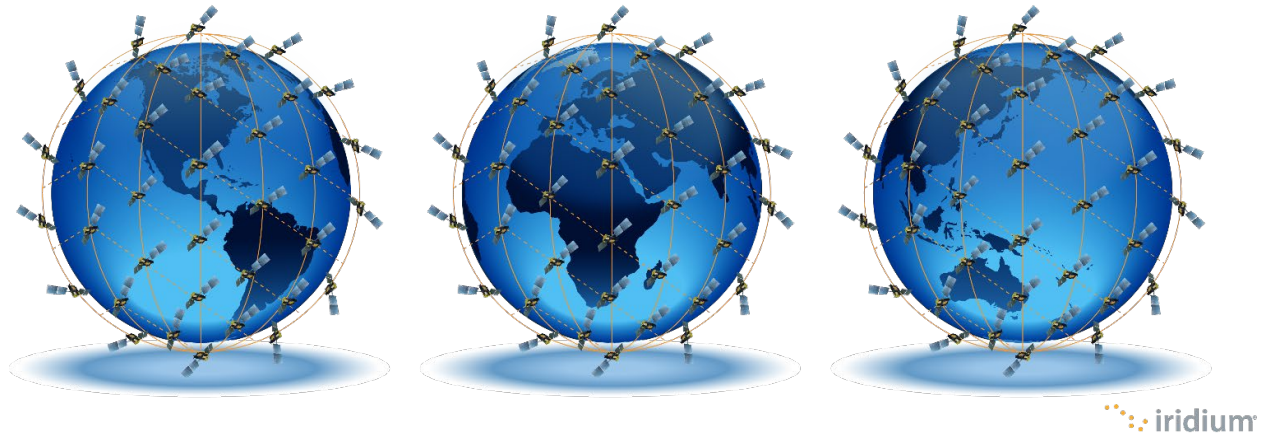


Figure 2: Iridium Satellite Network

The satellites are in a near-polar orbit at an altitude of 485 miles (780 km). They circle the earth once every 100 minutes traveling at a rate of 16,832 miles per hour. Each satellite is cross-linked to four other satellites: two satellites in the same orbital plane and two in an adjacent plane. These links create a dynamic network in space - calls are routed among Iridium satellites without touching the ground, creating a highly secure and reliable connection. Cross-links make the Iridium network particularly impervious to natural disasters - such as hurricanes, tsunamis, and earthquakes - that can damage ground-based wireless towers.

Iridium Ground Network

The ground network consists of the System Control Segment and gateways to the terrestrial networks. The System Control Segment is what commands and controls the satellites for the Iridium system. It provides global operational support and control services for the satellite constellation. It also delivers satellite tracking data to the gateways. There are three main components:


- Four Telemetry Tracking and Control Sites
- The Operational Support Network
- The Satellite Network Operation Center


The primary linkage between the System Control Segment, the satellites, and the gateways is through a satellite-to-satellite communications system called K-Band feeder links and cross-links throughout the satellite constellation.


1.4. Safety Instructions

Prior to using the STREAM+ devices, read and understand the user manual including the safety warnings and information. Failure to do so could result in serious injury or death.


Throughout the manual, important safety considerations and information are highlighted as follows:


	WARNING
	The warning symbol denotes a serious hazard. It calls attention to a procedure, practice, condition, which, if not correctly performed or adhered to, could result in injury or death of personnel.

	CAUTION
	The caution symbol denotes a hazard. It calls attention to a procedure, practice, condition, or the like, which, if not correctly performed or adhered to, could result in damage to part or all of the product.


	NOTE
	The note symbol highlights important information. It calls attention to an essential procedure, practice, condition, which, if not correctly performed or adhered to, could result in unsuccessful missions or diminished system accuracy.


Installation

	WARNING
	Always secure power cables. Failure to secure cables could result in fraying or failure of the power cable, resulting in fire and/or failure of the device. These hazards could cause serious injury or death.


	CAUTION
	Install the STREAM+ device safely and securely. Failure to do so could result in dislodging of the device especially when operating in areas of high winds or when mounted on moving vehicles, which could result in injury or damage.

Explosive Atmosphere / Blasting Areas


	WARNING
	Power down or turn off the STREAM+ device when in a “blasting area” or in areas posted “Turn off two-way radio.” Obey all signs and instructions. Failure to do so could result in serious injury or death.

	WARNING
	Power down or turn off the STREAM+ device in any area with a potentially explosive atmosphere. Obey all signs and instructions. Sparks from your device in such areas could cause an explosion or fire resulting in serious injury or death.


Modifications to Device

	WARNING
	Changes or modifications to this equipment not expressly approved by MetOcean Telematics could result in violation of FCC and Industry Canada rules and void the user's authorization to operate this equipment.

Possible Hazards

	WARNING
	There may be a hazard associated with the operation of your STREAM+ close to inadequately protected personal medical devices such as hearing aids and pacemakers. Consult the manufacturer of the medical device to determine if it is adequately protected

Possible Interference

	WARNING
	Operation of your STREAM+ device close to other electronic equipment may also cause interference if the equipment is inadequately protected. Observe any warning signs and manufacturers' recommendations.

Radio Frequency Exposure Safety

The STREAM+ device contains a radio transmitter. Certifications in most regions, including the United States and Canada, require that the STREAM+ antenna only be operated when it is at least 35 cm (13.77 inches) away from persons.

Operating Frequency: 1616 MHz to 1626.5 MHz

Maximum Average EIRP: 9dBW

MetOcean Telematics declares that the radio transmitter is in compliance with Directive 2014/53/EU. The full text of the EU declaration of conformity is available at the following internet address: <https://metocean.com/>

Painting: Never paint the STREAM+ device. Paint could negatively affect the performance of the device. Further, painting voids all warranties.

Cleaning: Only use mild soaps or detergents to clean the STREAM+ device.

2. Prerequisite

2.1. Setting up Iridium Communications

The STREAM+ product line requires an Iridium service (airtime) plan before the device can send and receive messages.

To enable Iridium communications, please contact sales@metocean.com.

Failure to activate the device with an Iridium service (airtime) plan will result in **NO DATA TRANSMISSION** from the device.

3. STREAM+ Development Kit Contents

This section provides information on the contents of the STREAM+ development kits and how to use them. The STREAM+ development kit contains all the necessary items for you to utilize the full functionality of the device. Start building and testing your custom made IOT/M2M solutions with the STREAM+ using its industry-standard interfaces, real-time GNSS positioning and reporting, and Iridium mid-band and messaging services.

Equipped with antennas, cabling and an external development board, the kit gives individual access to all the STREAM+ communication and power functions.



Figure 3: STREAM+ Development Kit

The items listed below are "off the shelf" components. Please talk to your sales representative to reorder if the items are lost or technical support for more information about the items listed below.

ITEM	DESCRIPTION	QUANTITY
1	STREAM+ Device	1
2	Test Interface Card	1
3	Iridium Antenna (External)	1
4	GNSS Antenna (External)	1
5	Cigarette Lighter Power Cable	1
6	Pigtail Power Connector	1
7	RS-232 Cable	1
8	RS-232 to USB Converter Cable	1
9	Iridium Antenna Cable	1
10	GNSS Antenna Cable	1
11	DC Wall Wart	1
12	Pigtail Data Connector Cable	1
13	USB to RS485 Serial Converter Cable	1

Table 1: STREAM+ Development Kit Contents

1. STREAM+ Device 	2. Test Interface Card 	3. Antenna (Iridium) 
4. Antenna (GNSS) 	5. Cigarette Lighter Power Cable 	6. Pigtail Power Connector 
7. RS-232 Cable 	8. RS-232 to USB Converter Cable 	9. Iridium Antenna Cable 
10. GNSS Antenna Cable 	11. DC Wall Wart 	12. Pigtail Data Connector Cable 
13. USB to RS485 Serial Converter 		

Figure 4: STREAM+ Development Kit Contents

1. STREAM+ Device

Overall Dimensions: 7.7" (195.5 mm) L x 4.1" (104 mm) W x 1.9" (48.2 mm) H



Figure 5: STREAM+ Device Dimensions

2. Test Interface Card

The Test Interface Card is used to break out the 10 conductors from the STREAM+ connector giving access to all of the STREAM+ communications and power functions.



Figure 6: Test Interface Card

3. Antenna (Iridium)

The Iridium antenna can be used to connect the STREAM+ device to the Iridium satellite constellation. The antenna can be pole mounted.



Figure 7: Antenna (Iridium)

4. Antenna (GNSS)

The GNSS antenna can be used to receive GNSS signals. The antenna can be pole mounted.



Figure 8: Antenna (GNSS)

5. Cigarette Lighter Power Cable

The Cigarette Lighter Power Cable can be used to provide power to the TIC board from a standard cigarette lighter.



Figure 9: Cigarette Lighter Power Cable

6. Pigtail Power Connector

The Pigtail Power Connector can be used to provide power from a power supply to the TIC board.



Figure 10: Pigtail Power Connector

7. RS-232 Cable

The RS-232 cable can be used to connect the TIC board to an external interface or to a USB connection via the provided RS-232 to USB converter cable.



Figure 11: RS-232 Cable

8. RS-232 to USB Converter Cable

The RS-232 to USB Converter Cable can be used to connect the RS-232 cable to a USB port.



Figure 12: RS-232 to USB Converter Cable

9. Iridium Antenna Cable

The Iridium Antenna Cable can be used to connect the STREAM+ device to the Iridium antenna.



Figure 13: Iridium Antenna Cable

10. GNSS Antenna Cable

The GNSS Antenna Cable can be used to connect the STREAM+ device to the GNSS antenna.



Figure 14: GNSS Antenna Cable

11. DC Wall Wart

The Wall Wart can be used to provide power to the TIC board.



Figure 15: DC Wall Wart

12. Pigtail Data Connector Cable

The Pigtail Data Connector Cable can be used to connect the STREAM+ device to the TIC board.



Figure 16: Pigtail Data Connector Cable

13. USB to RS485 Serial Converter

The USB to RS485 Serial Converter can be used to connect the TIC board to an RS-485 source.



Figure 17: USB to RS485 Serial Converter

4. Device Identification

The following labels are found on the bottom of the STREAM+ device and contains the following product identifiers:

1. **Host Manufacturer Name:** STREAM+
2. **Model:** The STREAM+ device purchased.
3. **IMEI:** The unique 15-digit number to identify the device in the Iridium network.
4. **SN:** The serial number of the device.

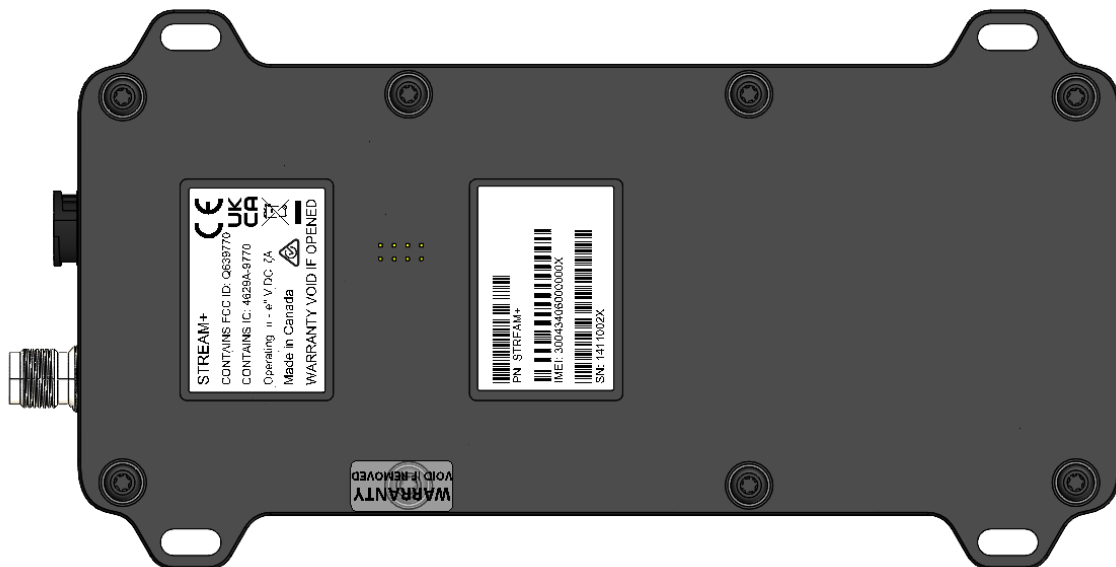
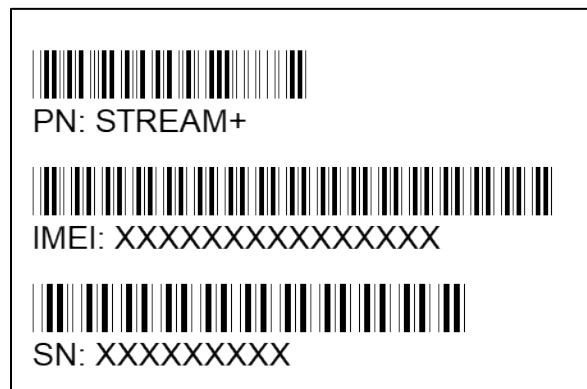
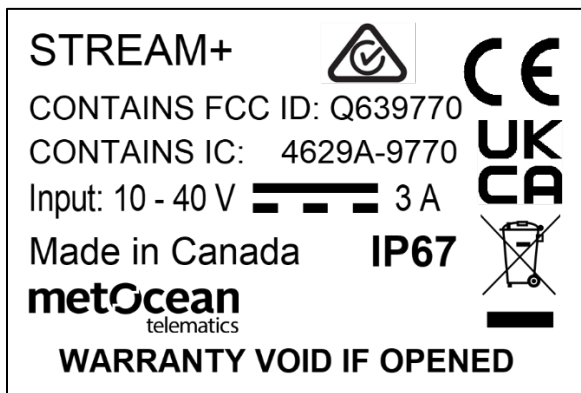


Figure 18: Device Identification

5. STREAM+ Connectors

The standard connectors on STREAM+ are outlined below.



Figure 19: STREAM+ Connectors

5.1. Power

Power and serial communications are enabled through a cable connection to the 10-pin connector on the side of the device. The cable connection is secured with a bayonet locking mechanism.

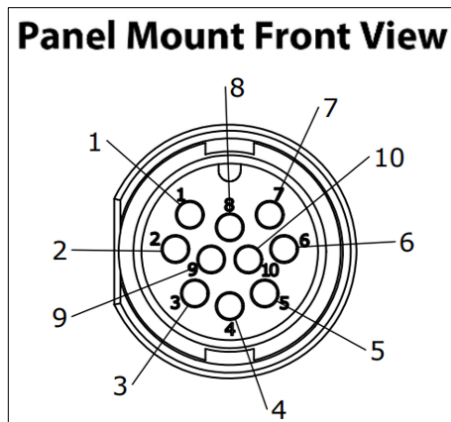


Figure 20: Panel Mount Front View

Connector Part Number: W7882-10xx-xxx family

Mating Part number: W6882-10xx-xxx Family

Name	PIN	Input/ Output
RS232_TXD	1	RS-232 TX (from STREAM+)
RS232_RXD	2	RS-232 RX (to STREAM+)
RS485_P	3	RS-485 A (unterminated)
RS485_N	4	RS-485 B (unterminated)
Inbox state	5	Active low (Open Drain), active when a message is in the inbox (to be downloaded from the server)
Outbox state	6	Active low (Open Drain), active when a message is in the outbox (queued to upload to the server)
nShutdown*	7	Active low shutdown control. Connect to ground to disable STREAM+.
PPS	8	Open Drain GNSS Pulse Per Second signal
Power_IN	9	Power Supply, 10VDC to 40VDC
Power ground	10	Ground

* STREAM+ may be enabled by driving this pin to any voltage above 1V or by leaving it floating. Floating state is defined as having >20MΩ between the pin and ground. This pin is not compatible with open-drain control methods since leakage current through transistors is enough to keep STREAM+ powered off.

5.2. Connecting the Unit

Power input and serial communications are established using the 5m **data cable** provided in the kit. With the Test Interface Board (TIC) board power switch **SW2** in the OFF position:

1. Connect the data cable plug to the power and data connector on the device and tighten the locking nut.
2. Connect the leads from the data cable to the TIC board STREAM+ connector, matching the conductor color to the labels.
3. Connect the DB9 cable to the TIC board and, to the RS232 to USB adaptor cable.
4. Connect the provided coaxial cables (SMA-TNC and TNC-N TYPE) to the applicable connectors on the device. The STREAM+ TNC connector must be connected to the Iridium Antenna. If not using GNSS, the STREAM+ SMA connector may be left open.
5. Connect the provided AD/DC Wall adaptor to a power source (100-240 VAC, 50-60Hz) and connect the barrel plug to the TIC board J3 connector. Ensure SW1 is in the lower position.
6. The TIC board switch (SW2) can now be moved to the ON position.

These instructions are intended as a general overview. For more detailed instructions please read the [Getting Started](#) section.

5.3. LED Indicator

The LED provides the user with visual feedback of the system's operational state.



Figure 21: STREAM+ LED Indicator

When STREAM+ is disabled or unpowered the LED is not illuminated.






Color	State	Description
 Green	Iridium Inactive	STREAM+ is ON but not performing any network operations.
 Cyan	Iridium Active (Excellent Signal)	STREAM+ is ON and accessing the Iridium Satellite Network. Signal strength: 3 to 5 bars
 Yellow	Iridium Active (Marginal Signal)	STREAM+ is ON and accessing the Iridium Satellite Network. Signal strength: 1 or 2 bars
 Magenta	Iridium Active (No Signal)	STREAM+ is ON and accessing the Iridium Satellite Network. Signal strength: 0 bars
 Red	Bootloader Running	STREAM+ is ON but is running the bootloader. Only basic system info, filesystem, and firmware updating commands are available.

Table 2: LED Indicator

6. Interfacing with STREAM+

This section describes how to interface with the STREAM+ Devices. This includes making physical connections, sending commands, etc.

6.1. Getting Started

STREAM+ communicates over RS-232 and/or RS-485 via the 10-pin data connector. To interface with this connector, use the Test Interface Card (TIC). The TIC breaks out all the 10 conductors from the STREAM+ connector, giving individual access to all of the STREAM+ communication and power functions.

📌 **Note:** The TIC is a development tool and should only be used in a desktop/lab setting.



Figure 22: STREAM+ Test Interface Card

The blocks of the TIC are shown in the figure below and are described in [Appendix A](#).

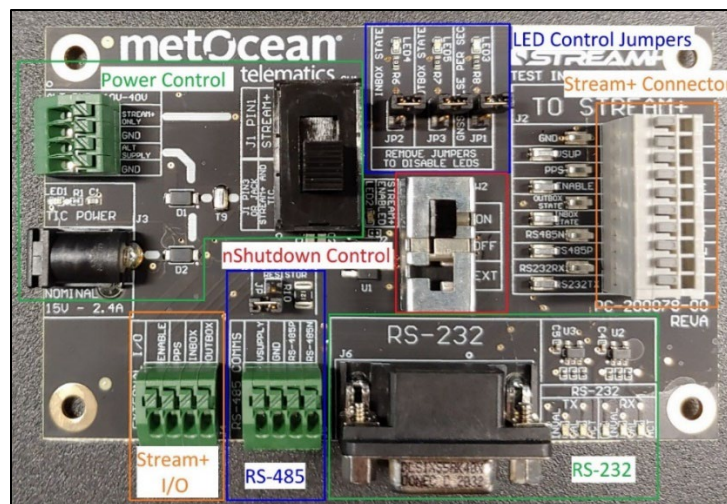


Figure 23: STREAM+ Test Interface Card Blocks

Push each of the tinned ends of the 5m data cable into the J2 connections, matching the color of the wire to the label on the connection. Ensure the connection is secure by pulling back on the wire – it should remain in place and not pull out of the connector.

Use the J3 barrel jack with the provided wall wart power supply or use connections from J1 for input voltage from external power supply. Use “STREAM+ Only” to bypass the TIC and power only the STREAM+ (does not power the LEDs, RS-232 traffic indicators etc.), and use “ALT SUPPLY” to power both the TIC and STREAM+ with a 10-40V. “Alt” in this case indicates an alternate voltage to the supplied 15V barrel connector. Use SW1 to select the power source. It is possible to use “STREAM+ Only” with one of the other power sources provided SW1 is in the upper position, which will power the TIC from the barrel jack or “ALT Supply”, and the STREAM+ will be powered by “STREAM+ Only”. This configuration can be used to evaluate STREAM+ power usage in isolation of the TIC, while still benefiting from the simplified interface.

Connect DB9/RS232 cable to J6, and the other end of the cable to the RS-232 to USB converter. The USB end of this converter then plugs into the control PC which will send commands to the STREAM+.

If using RS-485, the provided RS-485 to USB converter cable may be used with the RS-485 COMMS connector J5 by plugging the black wire into GND, the orange wire to RS-485P, and the yellow wire to RS-485N. It is strongly recommended that the JP4 jumper be populated if RS-232 and RS-485 are both connected to the TIC.

Once all the connections are made, enable the STREAM+ device by flipping SW2 into either the ON or EXT position.

6.2. First STREAM+ Command

From the control PC, download putty or some other equivalent client and configure a serial connection on the appropriate COM port. Use 19200 baud, and force local echoes for keyboard strokes. Once the client is active, turning on the device will not display anything in the client. To confirm the device is properly powered, connected, and loaded with the correct firmware, paste the following line into the terminal:

```
get sysInfo {}
```

Depending on the firmware/hardware version and serial number, the response should be something like

```
200 sysInfo
{"firmware":"1.3.3","bootloader":"1.3.2","serial_number":"Y002GN","board_revision":1,"model":1,"mode
m_version":null,"imei":null,"iccid":null}
```

If this response is received (values for each field will vary), you are now ready to interface with the STREAM+.

7. Firmware Upgrade

7.1. Updating STREAM+ Firmware

This section describes how to update the STREAM+ firmware.

- ☑ **Note:** A few additional steps must be taken for modem updates. Verify the new firmware version prior to removing power from the device. Ensure the Iridium modem has no active services running before the update can be initiated (any server sync file, IMT, or PPP passthrough operations must be stopped). Active services may be checked using the command: `get iridium {}`

Step 1: Connect the STREAM+ to Your PC

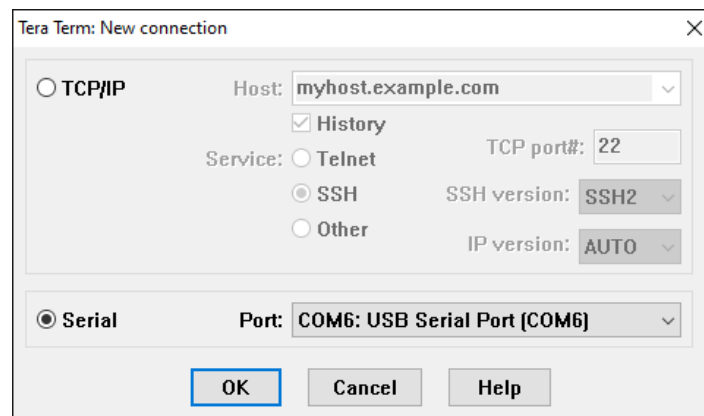
1. Connect the STREAM+ Unit as described in [Getting Started](#) section.
 - Identify the COM port on which your STREAM+ is connected. Windows Device Manager is ideal for this. The COM port should be COM6 or COM13, or something similar.

Step 2: Install Tera Term

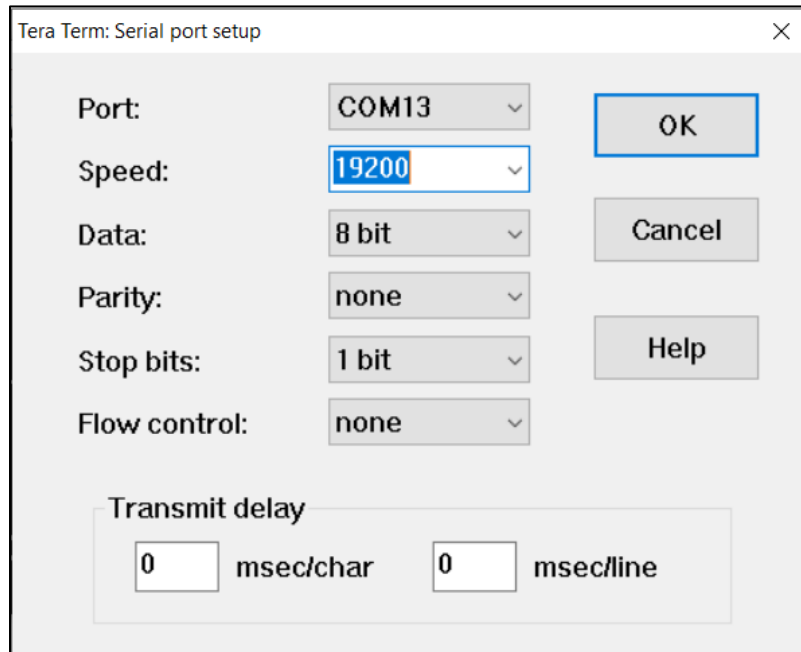
1. Go to the Tera Term home page: <https://teratermproject.github.io/index-en.html>
2. Click the Download link.
3. Under “Download Package list” select the latest release. Choose teraterm-4.xxx.exe.
 - a. teraterm-4.106.exe may be included with these instructions.
4. Download Choose teraterm-4.xxx.exe and install it.
 - a. Choose “Standard installation”

Step 3: Open & Set up Tera Term

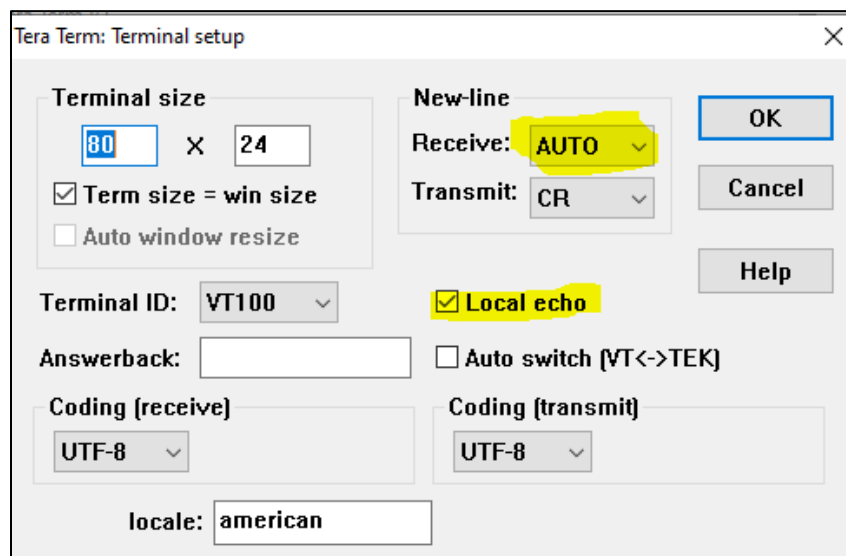
1. Open Tera Term, which you should already have installed.
2. Connect to the STREAM+'s serial port at the correct baud rate.
 1. Go to File > New Connection.
 2. Pick your serial interface's COM number.
 3. Click OK.



3. Verify the serial settings.
 1. Go to Setup > Serial port.
 2. A menu should pop up that looks like this. Make sure all the settings match except the specific port. Make sure to click "OK" when you're done.



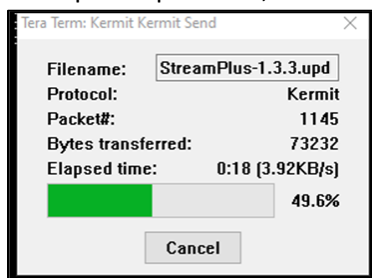
4. Verify the terminal settings.
 1. Go to Setup > Terminal.
 2. Make sure the settings match the following screenshot.



Step 4: STREAM+ Update

Follow these steps to update the STREAM+ firmware:

1. Issue command:
`get sysInfo {}`
Response should be something like:
`200 sysInfo {"firmware":"1.3.2","bootloader":"1.3.2",
"serial_number":"Y000S1","board_revision":1,"model":1,"modem_version":"1.3.1","imei":"
30005801234xxxx","iccid":"89881697712345xxxx"}`
2. Issue command:
`put file {"mode":"kermit"}`
3. In Tera Term, click "file" -> "Transfer" -> "Kermit" -> "Send", and navigate to the file location of the .upd file provided, and select that file.



Response should be

```
200 file {}
```

4. Issue command:
`put update {"file":"*providedfile*.upd"}`
Response should be:
`200 update {}`

Note: For modem firmware updates, this response indicates that the update process has started. It takes roughly 5 minutes for the full update process to complete.

Step 5: Verify STREAM+ Update

Follow these steps to confirm that the update was completed successfully:

1. Issue command (for modem updates only):
`get iridium {}`
The response **should** show an update in progress until the operation is complete.
`200 iridium
{"enabled":true,"signal":5,"provisioned":true,"ip_ready":true,"imt_ready":true,"active_
services":["update"]}`
Repeat this check until the **update** service is no longer shown (takes ~5 minutes to complete).
2. Issue command:
`get sysInfo {}`
Response **should** be something like ("firmware" field should match the provided firmware file):
`200 sysInfo {"firmware":"1.3.3","bootloader":"1.3.2",
"serial_number":"Y000S1","board_revision":1,"model":1,"modem_version":"1.3.4","imei":"
30005801234xxxx","iccid":"89881697712345xxxx"}`

8. STREAM+ API

8.1. Firmware Versioning

The STREAM+ firmware version is in the format: “MAJOR.MINOR.PATCH”.

Changes to the MAJOR version indicate that some backwards compatibility has been lost with the previous major version’s API. It is recommended that the setup of your device and any scripts/programs running on them be carefully reviewed in case they need to be adjusted.

Changes to the minor version indicate that new commands have been created, new keys have been added to existing command, or error codes returned by failed commands may have been updated. Overall, the firmware should remain backwards compatible with previous major versions.

Changes to the patch version indicate either minor improvements to efficiency and/or responsiveness, or bugfixes have been implemented.

📌 **Note:** Currently the only MAJOR version released for STREAM+ is v1. When a new MAJOR version is released migration notes for older version will be included in this document to avoid the need to carefully review the entire document and potentially missing details.

8.1.1. Migrating from v1.x.x to v2.x.x

8.1.1.1. Command Updates

GET imt {}:

“topics” key has been removed (use the GET imtTopicList {} instead for the topic list)

GET iridium {}:

When the FTP service is active, the “active_services” array will now contain “ftp” instead of “mail” as the string value

8.1.1.2. General Command Formatting

In v1.x releases, tabs were usually interchangeable with spaces for whitespace separating the parts of commands. As of v2.0.0 tabs will no longer be accepted and spaces must always be used where whitespace is required.

8.2. Command and Response Format

8.2.1. Commands

Commands consist of a carriage return (CR - ASCII 13) terminated line containing three fields, a method, target, and an object.

Commands can be a maximum of 1024 bytes (including the CR).

The first two fields (method and target) are terminated by spaces (ASCII 32), and the final field (the object) is terminated by a CR.

Example (with the CR omitted):

```
1. GET fileinfo {"file":"test.txt"}
```

The method field can either be `GET` or `PUT`, it is case insensitive. Each target may support either or both methods. `GET` methods will only query information from the STREAM+ (e.g. configurations, operational states, or files). `PUT` methods will tell the STREAM+ to perform some action, or to write configurations and files to the device.

The target field can be one of several commands described in the “General Commands” section of this document. Targets are also case insensitive.

Command arguments are provided as a JSON object of key-value pairs, or an empty object `{}` if there are no arguments. The STREAM+ recognizes standard JSON syntax, however, only supports basic ASCII character encoding.

Note: All non-printable characters received before a command starts will be discarded. Non-printable characters received after printable data has been received will result in error responses (aside from when binary protocols such as kermit are in use).

8.2.2. Response

The STREAM+ responds to every command with a response.

The response consists of a decimal code terminated by a space, followed by the target terminated by a space, followed by a JSON object terminated with a CR.

Example (with the CR omitted):

```
1. 200 fileInfo {"file":"test.txt","size":30}
```

The response code is always three decimal digits. The meaning of these codes is described in the “[API Response to Commands](#)” section of this document.

The response target is always the same as the original command target unless the method or target were unrecognized, in which case the target will be `unknown` and the JSON object will be empty `{}`.

8.2.3. Cyclic Redundancy Checks

If CRC validation has been enabled for a serial port (see the [serialConfig](#) command for more information on enabling CRC checks), then the additional checksum bytes must be appended to every command immediately before the carriage return. The CRC bytes are read as a 16-bit ASCII-encoded-hexadecimal value.

The CRC algorithm used is CRC-16/CCITT-FALSE, which uses the following parameters:

- Width: 16 bits
- Polynomial: `0x1021`

- Initial value: 0xFFFF
- Input data is not reflected
- Output CRC is not reflected
- No XOR is performed on the output CRC (i.e. XOR with 0x0000)
- Check word is 0x29B1
- Residue word is 0x0000

The same CRC algorithm is used when transferring files to/from the device using base64 mode.

When this mode is enabled all command responses will also have a trailing CRC.

For example:

```
1. GET sysinfo {}3914
2. 200 sysInfo
{"firmware":"1.6.2","bootloader":"1.3.3","serial_number":"16AC0001","board_revision":3,"model":1,"modem_version":"1.3.4","imei":"300058001230123","iccid":"8988169770123456789"}4369
```

If a CRC is expected but invalid (or missing) then the command will not be executed.


```
1. GET sysinfo {}
2. 500 sysInfo {}4E51
```



8.3. Command Table

The command tables describe the different JSON keys and values that may be used in a STREAM+ command or response.

8.3.1. Table format

The command table consists of the following columns:

- **Name** is a JSON key
- **Type** is the data type of the value corresponding to the key
- **Description** is just a description of what the key does and how its used
- **GET**, **PUT**, and **RSP** denote where the JSON key is relevant (In other words, if it's used in GET or PUT methods, or if it's returned in a response)
 - Each cell in this table has a meaning
 - If the cells are greyed-out, that method is unsupported by the command (should span the entire column of a method)
 - Blank: This key is ignored by this command, or never sent by the response
 -  This key is mandatory for the command, or always present in the response

-  This key is optional for the command, or conditionally present in the response
-  This key must not be provided in the command, otherwise it will return an error

Example table header:

Name	Type	Description	GET	PUT	RSP
------	------	-------------	-----	-----	-----

8.4. General Commands

8.4.1. reset

The **reset** command is used to perform a software restart of the STREAM+.

Name	Type	Description	GET	PUT	RSP








Example:

```
1. PUT reset {}
2. 200 reset {}
```

8.4.2. serialConfig

The **serialConfig** command is used to configure (PUT) or enquire (GET) the RS-232 and RS-485 serial port settings. When changing baud rate or CRC settings, if no valid command is received at the new rate within 30 seconds, the settings will revert to the previously saved rate. If a valid command is received within 30 seconds, the new settings will be maintained on following power cycles/resets.

The port on which this command is received is not authorized to disable itself.

Name	Type	Description	GET	PUT	RSP
port	String	Either "rs232" or "rs485" (Optional) Default: The port the command was received on			
enabled	Boolean	Serial port is enabled if true (Optional)			
Baud	Number	Baud rate for the serial port (Optional) Valid settings: 4800, 9600, 19200, 38400, 57600, 115200, 230400, and 460800 Default: 19200			

Name	Type	Description	GET	PUT	RSP
crc	Boolean	Require commands to be CRC terminated (Optional) Default: false Added: v1.6		?	?

Example:

```

1. GET serialConfig {}
2. 200 serialConfig {"port":"rs232","enabled":true,"baud":19200}
3.
4. PUT serialConfig {"port":"rs485","enabled":false}
5. 200 serialConfig {}

```

8.4.3. sysInfo

The **sysInfo** command provides STREAM+ device information.

Note: After formatting the filesystem, modem information (modem_revision, IMEI, ICCID) will be lost until the modem is turned on again.

Name	Type	Description	GET	PUT	RSP
firmware	String	Firmware version			✓
bootloader	String	Bootloader version			✓
serial_number	String	Identification/serial number			✓
board_revision	Number	Hardware revision number			✓
Model	Number	Device model: 0 = OEM 1 = Sealed Device			✓
modem_version	String	Modem Firmware Version <i>null</i> if the modem version has not been retrieved yet (requires modem to be powered on)			✓
Imei	String	Modem IMEI			✓

Name	Type	Description	GET	PUT	RSP
		<i>null</i> if the imei has not been retrieved yet (requires modem to be powered on)			
Iccid	String	Modem ICCID <i>null</i> if the iccid has not been retrieved yet (requires modem to be powered on)			✓

Example:

```
1. GET sysInfo {}
2. 200 sysInfo
{"firmware":"1.6.2","bootloader":"1.3.3","serial_number":"16AC0001","board_revision":3,"model":1,"modem_version":"1.3.4","imei":"300058001230123","iccid":"8988169770123456789"}
```

8.4.4. sysTime

The **sysTime** command is used to set or retrieve a system time for STREAM+.

System time on STREAM+ does not have a specific reference epoch, so the reference can be chosen by the host based on system requirements.

The system time allows a host system to be powered down and re-acquire its time from the stream+ when starting up again if desired.

☑ **Note:** System time is not retained on device reset. After any power reset the system time start as 0, and therefore represents the monotonic seconds since startup.

Name	Type	Description	GET	PUT	RSP
timestamp	Number	32bit Unsigned timestamp in seconds. Value can range from 0 to 4294967295. Offset to real calendar time is arbitrarily chosen by the host. Some common choices are: <ul style="list-style-type: none"> • Seconds since 1970-01-01 00:00:00 UTC (Unix time) • Seconds since 1980-01-06 00:00:00 UTC (GPS time) • Seconds since 2000 		✓	✓

```
1. PUT sysTime {"timestamp":1708983030}
2. 200 sysTime {}
3.
```

```
4. GET sysTime {}
5. 200 sysInfo {"timestamp":1708983033}
```

8.4.5. temperature

The **temperature** command returns the internal temperature of the device in degrees Celsius.

Name	Type	Description	GET	PUT	RSP
temperature	Number	Internal temperature of device in degrees Celsius			✓


Example:

```
1. GET temperature {}
2. 200 temperature {"temperature":24.37}
```


8.4.6. update

The **update** command, when issued with PUT, is used to apply an update file that is already loaded onto the device via the file commands or FTP. When issued with GET, it displays the type of update, version, and if the update is already installed.

For STREAM+ updates, the device will reset briefly during the update process. The update process takes about 4 minutes for 9770 modem updates and does not reset the STREAM+ device. For 9770 modem updates, the modem must be powered down when the “PUT update {...}” command is issued.

 **Note:** Update files used by the update command must have the file extension .upd.

Name	Type	Description	GET	PUT	RSP
file	String	Name of update file.	✓	✓	
inbox	Boolean	Use file from the INBOX (Optional).	?	?	
imt	Boolean	Use file from the IMT inbox (Optional).	?	?	
type	String	“firmware” for main application firmware. “bootloader” for bootloader firmware. “9770” for 9770 modem firmware. “invalid” for unsupported/corrupt updates.			?
installed	Boolean	true if the firmware is already installed. false if not installed. This key is not reported for 9770 updates.			?

Name	Type	Description	GET	PUT	RSP
version	String	Version of the firmware.			

Example Updating STREAM+ Device:

```
1. PUT update {"file":"StreamPlus-1.3.3.upd"}
2. 200 update {}
3. ...
4. GET update {"file":"StreamPlus-1.3.3.upd"}
5. 200 update {"type":"firmware","installed":true,"version":"1.3.3"}
```

Example Updating 9770 Modem:

```
1. PUT update {"file":"StreamPlusModem-1.3.0.upd"}
2. 200 update {}
3. ...
4. GET update {"file":"StreamPlusModem-1.3.0.upd"}
5. 200 update {"type":"9770","version":"1.3.0"}
```

8.4.7. voltage

The **voltage** command returns the voltage supplied to the device at the input connector.

Name	Type	Description	GET	PUT	RSP
supply	Number	Input supply voltage of device in volts			

Example:

```
1. GET voltage {}
2. 200 voltage {"supply":15.984}
```

8.5. Filesystem Commands

8.5.1. file

The **file** command can be issued with GET to request file data from the STREAM+ internal storage. If the *mode* key is set to “base64” then the *data* key will return the contents of the file as a string of Base64 encoded data. If *mode* is set to “kermit”, the file will be transferred using the Kermit protocol. The transfer begins immediately after the command is received, before the response message confirming the command is returned.

1. GET file {"file":"data.csv","mode":"kermit"}
2. <STREAM+ enters Kermit mode>

The **file** command can be issued with PUT to transfer file data to the STREAM+ internal storage using the same command parameters as the GET request. When the transfer mode is set to “kermit”, the STREAM+ will be ready to receive the file immediately after replying with **READY**.

1. PUT file {"file":"data.csv","mode":"kermit"}
2. READY
3. <STREAM+ enters Kermit mode>

- ✔ **Note 1:** When using base64 encoding remember to account for the fact that base64 will increase the amount of data by ~33% compared to raw binary data.
- ✔ **Note 2:** In "mode":base64, the CRC function used on the "data" key uses the CRC-16/CCITT-FALSE algorithm. See the [Cyclic Redundancy Checks](#) section for more detail.
- ✔ **Note 3:** In "mode":base64, PUT commands may take up to 10 seconds to write to the filesystem in the worst case
- ✔ **Note 4:** In "mode":"kermit" if no Kermit packets are received by stream+ for 30 seconds stream+ will exit Kermit mode and return a 408 error.

By default files are transferred to, or accessed from, the general-purpose file storage directory. This is also the only location that may be used to upload files to the STREAM+ from the serial port. If there are directories in the string passed to the “file” key, the command will return a bad JSON error.

Files that have been received over the Iridium network may be read by setting the “inbox” or “imt” key to true. “inbox” is used to read files downloaded from a remote server via FTP, as specified with the **serverConfig** command. “imt” is used to read files received using IMT.

The “log” key is used to read locally-generated debug logs.

- ✔ **Note 1:** Issuing the **PUT file** command with the “inbox”, or “log” keys set will return an error. The “config” key cannot be true for GET or PUT.
- ✔ **Note 2:** Setting more than one Boolean flag to true when issuing the **GET file** command will return an error.

Name	Type	Description	GET	PUT	RSP
mode	String	File transfer method, may be either “kermit” or “base64” (Optional) Default: “base64”	?	?	

Name	Type	Description	GET	PUT	RSP
file	String	File name <input checked="" type="checkbox"/> Note 1: Optional for PUT requests when mode is set to “kermit”, required for “base64” <input checked="" type="checkbox"/> Note 2: For PUT requests if the mode is set to “kermit” this name will be applied to the first file received if multiple files are being sent to the device in a single kermit session.	?	?	?
config	Boolean	Config file access is not permitted for the file command	✗	✗	
inbox	Boolean	Read file from the INBOX (Optional)	?	✗	?
imt	Boolean	Read file from the IMT inbox (Optional)	?	✗	?
log	Boolean	Read from a file in the debug log (Optional)	?	✗	?
start	Number	Offset into the file of the first byte to be returned or written (Optional) <input checked="" type="checkbox"/> Note: This key is ignored if mode is “kermit” Default: 0	?	?	?
data	String	Base64 encoded file data. The final 16bits of data represent the 16bit CRC16 for the data. <input checked="" type="checkbox"/> Note 1: The total length of encoded data can’t exceed 514 bytes (including the CRC16). The base64 data key may be large enough to fit 514 bytes of encoded data. <input checked="" type="checkbox"/> Note 2: Required if mode is “base64”, otherwise the key is ignored.		?	?
length	Number	Maximum number of bytes to read in a GET request (up to 512) when the transfer mode is “base64”. Default: 512	?		
written	Number	Number of bytes written to file			?

Example Message Flow for Sending a File

This example writes the following file, named `test.txt` on the STREAM+, it will use the `"start"` key to perform the write in two PUT requests.

- ```
1. hello world
2. Add another line!
```

Step 1: Send the first line "hello world\n"

✍ **Note:** "hello world\n" plus a CRC16 encodes to aGVsbG8gd29ybGQKi0Y=

```
1. PUT file {"file":"test.txt","data":"aGVsbG8gd29ybGQKi0Y="}
2. 200 file {"written":12}
```

Step 2: Send the second line "Add another line!\n"

```
1. PUT file {"file":"test.txt","start":12,"data":"QWRkIGFub3RoZXIgbGluZSEN5L8="}
2. 200 file {"written":18}
```

### Example Message Flow for Receiving a File

Trivial case, receiving the test.txt file in one go:

```
1. GET file {"file":"test.txt"}
2. 200 file {"file":"test.txt","start":0,"data":"aGVsbG8gd29ybGQNQRkIGFub3RoZXIgbGluZSENFx0="}
```

Notice how the previously sent base64 "data" strings appear concatenated.

### Example Message for Receiving Part of a File

For example, just receive the word world from test.txt:

```
1. GET file {"file":"test.txt","start":6,"length":5}
2. 200 file {"file":"test.txt","start":6,"data":"d29ybGRvMg=="}
```

"d29ybGRvMg==" decodes to worldc# where world is the data we wanted to receive, and c# is the CRC16 as ASCII characters.

## 8.5.2. fileDelete

The **fileDelete** command will remove the specified file currently stored on STREAM+.

By default files will be deleted from the general-purpose file storage directory. If there are directories in the string passed to the "file" key, the command will return a bad JSON error.

Files that have been received over the Iridium network may be deleted by setting the "inbox" or "imt" key to true. "inbox" is used to delete files downloaded from a remote server via FTP, as specified with the **serverConfig** command. "imt" is used to delete files received using IMT.

The "log" key is used to delete locally-generated debug logs.

The “config” key is used to delete configuration settings, reverting them to the factory defaults. The file name corresponds to the `config` command that creates it. The system must be reset for the default settings to take effect.

✔ **Note 1:** Setting more than one Boolean flag to true will return an error.

✔ **Note 2:** File that are actively queued for transmission over IMT or FTP must first be cancelled before they can be deleted.

| Name   | Type    | Description                                    | GET | PUT | RSP |
|--------|---------|------------------------------------------------|-----|-----|-----|
| file   | String  | File or directory to delete (Required)         |     | ✓   |     |
| inbox  | Boolean | Delete file from the INBOX (Optional)          |     | ?   |     |
| imt    | Boolean | Delete file from the IMT inbox (Optional)      |     | ?   |     |
| log    | Boolean | Delete a debug log (Optional)                  |     | ?   |     |
| config | Boolean | Delete saved configuration settings (Optional) |     | ?   |     |

Example:

```
1. PUT fileDelete {"file":"testfile_0003.txt"}
2. 200 fileDelete {}
```

### 8.5.3. fileInfo

The **fileInfo** command provides details of a file saved in the STREAM+ internal storage.

✔ **Note:** Setting more than one Boolean flag to true will return an error.

| Name   | Type    | Description                                         | GET | PUT | RSP |
|--------|---------|-----------------------------------------------------|-----|-----|-----|
| file   | String  | File name                                           | ✓   |     | ✓   |
| inbox  | Boolean | Get information for an INBOX file (Optional)        | ?   |     | ?   |
| imt    | Boolean | Get information for an IMT inbox file (Optional)    | ?   |     | ?   |
| log    | Boolean | Get information for debug log files (Optional)      | ?   |     | ?   |
| config | Boolean | Get information for a configuration file (Optional) | ?   |     | ?   |

| Name  | Type   | Description                                                                                                                                                                                                                                                 | GET | PUT | RSP |
|-------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| size  | Number | File size in bytes                                                                                                                                                                                                                                          |     |     | ✓   |
| queue | String | String indicating if the file is queued to transmit over iridium: <ul style="list-style-type: none"> <li>• none: Not currently queued to transmit</li> <li>• ftp: Queued to transmit via FTP</li> </ul> imt: Queued to transmit as an IMT MO<br>Added: v2.0 |     |     | ✓   |

Example:

```
1. GET fileInfo {"file":"testfile_033.txt"}
2. 200 fileInfo {"file":"testfile_033.txt","size":1337}
```

#### 8.5.4. fileList

The **fileList** command returns the list of files currently stored on the STREAM+.

By default the files in the general-purpose file storage directory will be listed.

Files that have been received over the Iridium network may be listed by setting the "inbox" or "imt" key to true. "inbox" is used to list files downloaded from a remote server via FTP, as specified with the **serverConfig** command. "imt" is used to list files received using IMT.

The "log" key is used to list locally-generated debug logs.

The "config" key is used to list saved configuration files. The file name corresponds to the **Config** command that creates it.

📌 **Note:** Setting more than one Boolean flag to true, not including "continue", will return an error.

| Name   | Type    | Description                            | GET | PUT | RSP |
|--------|---------|----------------------------------------|-----|-----|-----|
| inbox  | Boolean | List files in the INBOX (Optional)     | ?   |     | ?   |
| imt    | Boolean | List files in the IMT inbox (Optional) | ?   |     | ?   |
| log    | Boolean | List debug log files (Optional)        | ?   |     | ?   |
| config | Boolean | List configuration files (Optional)    | ?   |     | ?   |

| Name         | Type     | Description                                                                                                                                         | GET | PUT | RSP |
|--------------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| continue     | Boolean  | Set to true if the previous file listing had the "files_remain" key set to true, in order to continue the directory listing (Optional)              | ?   |     |     |
| files        | [String] | List of file names                                                                                                                                  |     |     | ✓   |
| files_remain | Boolean  | True if the file array in the response did not contain all of the files remaining in the folder. Not present if the directory listing was complete. |     |     | ?   |

### Example Message Flow with a Continue Flag

For systems with more files than can fit in one response, send two `fileList` invocations to obtain the complete list of files:

```

1. GET fileList {}

2. 200 fileList {"files":["20201001130838!CatLoLCatExample.jpg","Cat_keyboard.jpg","SumatraPDF-3.4.3-64-install.exe","dantest2.txt","ek16.tar","empty_test_1.null","empty_test_2.null","empty_test_3.null","empty_test_4.null","empty_test_5.null","filler1","filler2","filler3","filler4","filler5","filler","foobar.tst","foorandom.bin","output-onlinefiletools - Copy - Copy.bin","output-onlinefiletools - Copy.bin","output-onlinefiletools.bin","ovw_test.bin","ovw_test2.bin","rtest_01d6e7bd-f7d3-11ec-a701-a87eea7d46e5.bin","rtest_1481c6da-f7d3-11ec-a2cf-a87eea7d46e5.bin","rtest_271670ae-f7d3-11ec-951c-a87eea7d46e5.bin","rtest_292623e2-f6fb-11ec-b1a8-a87eea7d46e5.bin","rtest_362ac2ba-f7a3-11ec-b4ce-a87eea7d46e5.bin","rtest_36665893-f6fb-11ec-aefc-a87eea7d46e5.bin"],"files_remain":true}

3.

4. GET fileList {"continue":true}

5. 200 fileList {"files":["rtest_95412ce7-f7d2-11ec-9b81-a87eea7d46e5.bin","rtest_9821ca2f-f722-11ec-92b6-a87eea7d46e5.bin","rtest_a36c1cd3-f7d2-11ec-9a5d-a87eea7d46e5.bin","rtest_a4d4ad1e-f7d3-11ec-a19e-a87eea7d46e5.bin","rtest_ab7398c4-f722-11ec-9ec4-a87eea7d46e5.bin","rtest_bbc6ee82-f7d3-11ec-977e-a87eea7d46e5.bin","rtest_cb54360c-f722-11ec-8a2c-a87eea7d46e5.bin","rtest_f225b501-f7d3-11ec-a64b-a87eea7d46e5.bin","rtest_f37b1871-f7d2-11ec-9deb-a87eea7d46e5.bin","test.txt"],"files_remain":false}

```

Notice how in the first response, `"files_remain":true` is returned, then in the second request `"continue":true` is used to request more files continued from the last reply.

Finally, in the last response `"files_remain":false` indicates we reached the end of the directory.

### 8.5.5. storageErase

The `storageErase` command allows the user to remove all files from storage at once.

📌 **Note:** This command may take several seconds to complete

| Name          | Type    | Description                                                                                                                                                                                                        | GET | PUT | RSP |
|---------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| factory_reset | Boolean | Remove all files from the filesystem and reset configuration back to system default (true).<br><br>Or remove all files and directories individually, but maintain the configuration (false).<br><br>Default: false |     | ?   |     |

Example:

```
1. PUT storageErase {"factory_reset":true}
2. 200 storageErase {}
```

### 8.5.6. storageInfo

The **storageInfo** command provides STREAM+ file system information, indicating the best estimate of free/used space on the device.

 **Note:** This command may take several seconds to complete.

| Name | Type   | Description                                         | GET | PUT | RSP |
|------|--------|-----------------------------------------------------|-----|-----|-----|
| used | Number | Used space, as reported by the file system (in KiB) |     |     | ✓   |
| free | Number | Free space, as reported by the file system (in KiB) |     |     | ✓   |

Example:

```
1. GET storageInfo {}
2. 200 storageInfo {"used":1438,"free":3701346}
```

## 8.6. Networking Commands

### 8.6.1. retrieveFile

The PUT **retrieveFile** command will request a specific file to be downloaded from the server.

| Name         | Type    | Description                                                   | GET | PUT | RSP |
|--------------|---------|---------------------------------------------------------------|-----|-----|-----|
| file         | String  | File name to retrieve from the downloads folder on the server | ✓   |     |     |
| retrieve_now | Boolean | Set to true retrieve immediately (Optional)<br>Default: true  | ?   |     |     |

Example:

```
1. GET serverfilelist {}
2. 200 serverFileList
{"uploads":false,"age":123,"files":["1.txt","2.txt","3.txt"],"files_remain":false}
3.
4. PUT retrievefile {"file":"3.txt","retrieve_now":true}
5. 200 retrieveFile {}
6. ...
7. GET filelist {"inbox":true}
8. 200 fileList {"inbox":true,"files":["3.txt"],"files_remain":false}
```

### 8.6.2. sendFile

The PUT **sendFile** command takes a file already present in the STREAM+ internal storage, and queues it to be transferred to the remote server via FTP, as specified by the **serverConfig** command.

The GET **sendFile** command returns the list of files currently queued for transfer to the remote server via FTP, as specified by the **serverConfig** command. The returned array of file names preserves the order in which the files will be sent.

If the optional “priority” key is specified, then only files of the given priority level will be returned. If the key is missing or negative, then every file queued for transfer will be listed.

If the optional “send\_now” key is set to false the file will not actually be sent to the remote server until either a **serverSyncFiles** command is issued, or another **sendFile** command is issued with “send\_now” set to true.

📌 **Note:** Setting more than one of “inbox”, “imt”, or “log” to true will return an error.

| Name         | Type     | Description                                                                                                                                                                                                                                                             | GET | PUT | RSP |
|--------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| file         | String   | File name on the STREAM+                                                                                                                                                                                                                                                |     | ✓   |     |
| config       | Boolean  | Config file access is not permitted for the <b>sendFile</b> command                                                                                                                                                                                                     | ✗   | ✗   |     |
| inbox        | Boolean  | Send file from the INBOX (Optional)                                                                                                                                                                                                                                     |     | ?   |     |
| imt          | Boolean  | Send file from the IMT inbox (Optional)                                                                                                                                                                                                                                 |     | ?   |     |
| log          | Boolean  | Send a debug log file (Optional)                                                                                                                                                                                                                                        |     | ?   |     |
| delete       | Boolean  | Set true to delete the file after it has successfully been sent (Optional)<br><input checked="" type="checkbox"/> <b>Note:</b> If "remove_after_upload" in <b>serverConfig</b> is true, files will be deleted after upload regardless of this setting<br>Default: false |     | ?   |     |
| server_path  | String   | File path on the destination server (Optional)<br>Default: The file will be placed in the folder provided by the "uploads" key to the serverConfig command. If unspecified, the default FTP login directory will be used.                                               |     | ?   |     |
| priority     | Number   | Send priority, higher priority files are sent first (Optional):<br>Range: 1 (Highest) - 5 (Lowest)<br>Default: 3                                                                                                                                                        | ?   | ?   |     |
| send_first   | Boolean  | Send the file before any others of the same priority.<br>Default: false                                                                                                                                                                                                 |     | ?   |     |
| send_now     | Boolean  | Start server file sync operation now (Optional)<br>Default: true                                                                                                                                                                                                        |     | ?   |     |
| list_from    | Number   | The position in the send queue for the first listed file (Optional)<br>Default: 0                                                                                                                                                                                       | ?   |     | ?   |
| files        | [String] | List of files pending transfer to the remote server                                                                                                                                                                                                                     |     |     | ?   |
| files_remain | Boolean  | True if the file array in the response did not contain all of the files remaining in the outbox.                                                                                                                                                                        |     |     | ?   |

### Example Sending a File Over FTP:

```
1. PUT sendFile {"file":"thisfile.bin"}
2. 200 sendFile {}
```

### Example Retrieving a List of Files Queued for Transfer:

```
1. GET sendFile {}
2. 200 sendFile {"list_from":0,"files":["thisfile.bin","inbox/thatfile.bin"],"files_remain":false}
```

#### 8.6.3. sendFileCancel

The PUT **sendFileCancel** command removes files that were added to the outbox queue using **sendFile**.

🔍 **Note 1:** Setting more than one Boolean flag to true will return an error

🔍 **Note 2:** If file not found (418) is returned it indicates that the file was not found in the queue (it may still exist on the filesystem).

| Name   | Type    | Description                                    | GET | PUT | RSP |
|--------|---------|------------------------------------------------|-----|-----|-----|
| file   | String  | File to cancel sending (Required)              |     | ✓   |     |
| inbox  | Boolean | Cancel files sending from the INBOX (Optional) |     | ?   |     |
| imt    | Boolean | Cancel files sending from IMT inbox (Optional) |     | ?   |     |
| log    | Boolean | Cancel sending log files (Optional)            |     | ?   |     |
| config | Boolean | Cancel sending configuration files (Optional)  |     | ?   |     |

Example:

```
1. PUT sendFileCancel {"file":"thisfile.bin"}
2. 200 sendFileCancel {}
```

#### 8.6.4. serverConfig

When issued as a PUT request, the **serverConfig** command provides the configuration information necessary to connect to a remote FTP server for file transfers.

When issued as a GET request, **serverConfig** returns the active FTP server configuration..

🔍 **Note:** Server configuration updates are not allowed while the device is actively sending data via FTP.

| Name                  | Type    | Description                                                                                                                                                                                                                             | GET | PUT | RSP |
|-----------------------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| host                  | String  | FTP server host name. Accepts IPv4 addresses ("93.184.216.34") and domain names (example.com). IPv4 addresses must be in dot-decimal notation. (Optional)                                                                               |     | ?   | ✓   |
| address               | String  | This is an alias to the host key for backwards compatibility. It is superseded by the host key if both are present. (Optional, Deprecated)                                                                                              |     | ?   |     |
| tls                   | Boolean | Enable FTPS support, see the notes on FTPS server compatibility below.<br>✍ <b>Note:</b> "host" key must be a proper domain name, not an address. (Optional)<br>Default: true                                                           |     | ?   | ✓   |
| port                  | Number  | FTP server port (Optional)                                                                                                                                                                                                              |     | ?   | ✓   |
| user                  | String  | FTP user account name. Maximum 63 bytes long. (Optional)                                                                                                                                                                                |     | ?   | ✓   |
| password              | String  | FTP user account password. Maximum 63 bytes long. (Optional)                                                                                                                                                                            |     | ?   |     |
| uploads               | String  | Remote server directory path for uploads (Optional)                                                                                                                                                                                     |     | ?   | ✓   |
| downloads             | String  | Remote server directory path for downloads (Optional)                                                                                                                                                                                   |     | ?   | ✓   |
| remove_after_download | Boolean | Set true to delete files from the FTP server after downloading (Optional)<br>Default: false<br>Added: v1.7                                                                                                                              |     | ?   | ✓   |
| remove_after_upload   | Boolean | Set true to delete files from the STREAM+ after uploading (Optional)<br>✍ <b>Note:</b> When this key is set to true, this setting overrides the "delete" key value used by the <b>sendFile</b> command<br>Default: false<br>Added: v1.7 |     | ?   | ✓   |

### Example (ordinary FTP):

```
1. PUT serverConfig
{"host":"123.45.67.89","port":21,"tls":false,"user":"FTPtest","password":"PasswordHere","uploads":"/uploads_folder","downloads":"/downloads_folder"}

2. 200 serverConfig {}

3.

4. GET serverconfig {}

5. 200 serverConfig
{"host":"123.45.67.89","port":21,"tls":false,"user":"FTPtest","remove_after_download":false,"remove_after_upload":false,"uploads":"/uploads_folder","downloads":"/downloads_folder"}
```

### Example (FTPS):

```
1. GET serverConfig
{"host":"example.com","port":21,"tls":true,"user":"FTPtest","password":"PasswordHere","uploads":"/uploads_folder","downloads":"/downloads_folder"}

2. 200 serverConfig {}

3.

4. GET serverconfig {}

5. 200 serverConfig
{"host":"example.com","port":21,"tls":true,"user":"FTPtest","remove_after_download":false,"remove_after_upload":false,"uploads":"/uploads_folder","downloads":"/downloads_folder"}
```

The FTPS server must support the following features:

- The server must support TLS 1.2
- The server must support explicit FTPS (implicit FTPS is not supported)


The FTPS server must have a TLS certificate signed against one of the following certificate authorities, note that self-signed certificates are not currently supported.

- Certum Trusted Network CA
- DigiCert Baltimore Root
- DigiCert EV.cer
- DigiCert Global Root G2
- DigiCert Global Root
- DigiCert
- GlobalSign Root CA - R1
- GlobalSign Root CA - R3
- Go Daddy Class 2 Certification Authority
- Go Daddy Root Certificate Authority - G2
- ISRG Root X1
- Sectigo (AAA)
- VeriSign Universal Root Certification Authority
- VeriSign

### 8.6.5. serverFileList

The PUT **serverFileList** command triggers a refresh of the list of files on the FTP server which are available for retrieval.

The GET **serverFileList** command is used to list out files available on the FTP server for download or those already located on the server's upload location.

 **Note:** It may take a few minutes to update the file list from the server after issuing the command with PUT. The "age" key should be used to verify when the list is up to date with new data.

| Name         | Type     | Description                                                                                                                                                                                                                                                         | GET | PUT | RSP |
|--------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| uploads      | Boolean  | Set true to list file in the upload folder on the FTP server. Set false to list files in the download location on the FTP server. If continuing a previous listing, the uploads field must be set to match what was previously listed. (Optional)<br>Default: false | ?   |     | ✓   |
| age          | Number   | This is the time in seconds at which the file list from the server was last updated.                                                                                                                                                                                |     |     | ✓   |
| continue     | Boolean  | Set to true if the previous server file listing had the "files_remain" key set to true, in order to continue the directory listing (Optional)                                                                                                                       | ?   |     |     |
| files        | [String] | List of file names                                                                                                                                                                                                                                                  |     |     | ✓   |
| files_remain | Boolean  | True if the file array in the response did not contain all of the files remaining on the server. Not present if the directory listing was complete.                                                                                                                 |     |     | ✓   |

Example:

```
1. PUT serverfilelist {}
2. 200 serverFileList {}
3. ...
4. GET serverfilelist {}
5. 200 serverFileList {"uploads":false,"age":786,"files":["1.txt","2.txt"],"files_remain":false}
6. ...
7. GET serverfilelist {}
8. 200 serverFileList {"uploads":false,"age":5,"files":["1.txt","2.txt","3.txt"],"files_remain":false}
```

### 8.6.6. serverSyncFiles

The PUT **serverSyncFiles** command triggers a background FTP file sync operation. During this operation, any files in the outbox will be sent to the remote FTP server. If any files that were already queued for download with the retrieveFile command, these files will be retrieved next.

If the “queued\_only” is set to false (default), STREAM+ will then retrieve a listing of the remote server's downloads directory (as specified by the **serverConfig** command), and download any files not already present on the local filesystem.

Setting the optional “stop” flag to true will cause the PUT **serverSyncFiles** command to suspend any FTP send or receive operations in progress.

The “send” and “receive” options control how the sync operation will be performed. Setting “send” to false will cause the server sync operation to skip sending files in the outbox. Setting “receive” to false will skip downloading files in the configured server “downloads” directory.

- ✔ **Note:** If the **serverSyncFiles** command is executed and there are MTs scheduled to be received via IMT, these messages will also be received (any time the Iridium modem is active IMT MT messages will always be received when available). It will however NOT initiate the transmission of queued MO messages.

The GET **serverSyncFiles** provides information about the status and contents of the file sync.

| Name    | Type    | Description                                                                                | GET | PUT | RSP |
|---------|---------|--------------------------------------------------------------------------------------------|-----|-----|-----|
| status  | String  | The status of file sync activity                                                           |     |     | ?   |
| inbox   | Number  | The number of messages in the inbox                                                        |     |     | ?   |
| outbox  | Number  | The number of messages in the outbox                                                       |     |     | ?   |
| stop    | Boolean | Stop the currently running file sync operation (Optional)<br>Default: false<br>Added: v1.4 |     | ?   |     |
| send    | Boolean | Send files in the outbox queue (Optional)<br>Default: true<br>Added: v1.6                  |     | ?   |     |
| receive | Boolean | Retrieve files from the remote server (Optional)<br>Default: true<br>Added: v1.6           |     | ?   |     |

| Name        | Type    | Description                                                                                                                        | GET | PUT | RSP |
|-------------|---------|------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| queued_only | Boolean | Set true to retrieve only those files queued for download via the <b>serverSyncFiles</b> command.<br>Default: false<br>Added: v1.7 |     | ?   |     |

Example:

```

1. GET serverSyncFiles {}
2. 200 serverSyncFiles {"status":"inactive","inbox":2,"outbox":1}
3.
4. PUT serverSyncFiles {}
5. 200 serverSyncFiles {}

```

## 8.7. IMT Commands

### 8.7.1. imt



- ❗ The PUT **imt** command will return a “service unavailable” (406) error if the 9770 modem is offline, or has not been provisioned for IMT when “send\_now” is true.
- ❗ IMT messages sent to the STREAM+ while the service is unavailable may not be received until a message has been sent to or from the STREAM+ while online
- ❗ Files sent over the IMT service must not exceed 100000 bytes and cannot be empty (zero bytes).

The PUT **imt** command takes a file already present in the STREAM+ internal storage, and queues it to be transferred to the specified IMT topic.

The GET **imt** command returns the availability of the IMT service and status of ongoing MO/MT activity.

- ✔ **Note 1:** Setting more than one Boolean file location flag to true will return an error.
- ✔ **Note 2:** PUT imt can not be used with a file that is already queued for transmission (will respond with error 419). In order to change a parameter (ie: topic) the file must be cancelled and then re-queued.
- ✔ **Note 3:** If PUT imt responds with error 408, it is recommended to use GET imtMoList to re-check the the queued files.
- ✔ **Note 4:** When MOs encounter errors during transmission, stream+ will retry 3 times before giving up and reporting the error state.

| Name | Type   | Description              | GET | PUT | RSP |
|------|--------|--------------------------|-----|-----|-----|
| file | String | File name on the STREAM+ |     | ✔   |     |

| Name              | Type    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | GET | PUT | RSP |
|-------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| inbox             | Boolean | Send file from the INBOX (Optional)                                                                                                                                                                                                                                                                                                                                                                                                                                                        |     | ?   |     |
| imt               | Boolean | Send file from the IMT inbox (Optional)                                                                                                                                                                                                                                                                                                                                                                                                                                                    |     | ?   |     |
| log               | Boolean | Send a debug log file (Optional)                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     | ?   |     |
| topic_id          | Number  | IMT topic ID to send the file to.<br>Either this key (preferred) or the topic key must be provided if the “cancel” key is not set to true.                                                                                                                                                                                                                                                                                                                                                 |     | ?   |     |
| topic             | String  | IMT topic to send the file to.<br>This key is required if the “cancel” key is not set to true. It is ignored if the “topic_id” key is set.<br> <b>Note:</b> The topic string associated with a topic is determined by the Iridium network and may change from time to time, even when provisioning details are not changed. As such it is recommended whenever possible to use the “topic_id” key instead |     | ?   |     |
| send_now          | Boolean | Set true to enable the modem for sending MOs now if this has not already been done. Setting to false queues the file for sending the next time IMT is enabled.<br>Default: true<br>Added: v2.0                                                                                                                                                                                                                                                                                             |     | ?   |     |
| cancel            | Boolean | Set true to cancel a file from the MO list instead of queueing a new file to send.<br>Added: v2.0<br> <b>Note:</b> If file not found (418) is returned it indicates that the file was not found in the queue (it may still exist on the filesystem).                                                                                                                                                    |     | ?   |     |
| available         | Boolean | Indicates if the IMT service is available                                                                                                                                                                                                                                                                                                                                                                                                                                                  |     |     | ?   |
| mobile_terminated | Boolean | Indicates if we are busy receiving messages via IMT.<br>Added: v1.7                                                                                                                                                                                                                                                                                                                                                                                                                        |     |     | ?   |
| mobile_originated | Number  | Indicates the number of mobile originated messages queued to send via IMT.<br>Added: v1.7                                                                                                                                                                                                                                                                                                                                                                                                  |     |     | ?   |

| Name           | Type     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | GET | PUT | RSP |
|----------------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| mo_status      | String   | String indicating the current state of MO activity: <ul style="list-style-type: none"> <li>• active: modem is on and trying to send queued MOs</li> <li>• idle: modem is off or no MOs are queued to send</li> <li>• blocked: all available MO slots are blocked by failed messages</li> </ul> Added: v2.0<br>📌 <b>Note:</b> If 3 MOs with errors are detected, further MO activity is blocked until at least one of the failed MOs is cancelled.                                                                                                                                                                                     |     |     | ?   |
| mo_failed_file | [String] | List of files which have failed to transmit over IMT. If the name of the final item is “...” it indicates that more items exist which have not been displayed due to buffer length constraints. This key will not appear if there are no files currently in the failed state.<br>Added: v2.0<br>📌 <b>Note:</b> MOs in the error state must be cancelled to remove them from the queue. In order to retry again the item must first be cancelled then re-added.                                                                                                                                                                        |     |     | ?   |
| mo_error       | [String] | List of error causes for mo_failed_file list. Possible string values: <ul style="list-style-type: none"> <li>• message_discarded_on_overflow</li> <li>• message_expired</li> <li>• message_transfer_timeout</li> <li>• segment_not_supplied</li> <li>• segment_incorrect</li> <li>• network_error</li> <li>• message_cancelled_pre_transit</li> <li>• message_cancelled_in_transit</li> <li>• subscription_invalid</li> <li>• protocol_error</li> <li>• message_dropped_local_crc_error</li> <li>• crc_error_in_transfer</li> <li>• user_supplied_crc_error</li> <li>• io_error</li> <li>• size_invalid</li> <li>• timeout</li> </ul> |     |     | ?   |

| Name | Type | Description                                                                                  | GET | PUT | RSP |
|------|------|----------------------------------------------------------------------------------------------|-----|-----|-----|
|      |      | This key will not appear if there are no files currently in the failed state.<br>Added: v2.0 |     |     |     |

### Example Retrieving IMT Service Information

```
1. GET imt {}
2. 200 imt
{"available":true,"mobile_terminated":false,"mobile_originated":2,"mo_status":"active","mo_failed_file":["Test1.txt","Test2.txt"],"mo_error":["message_discarded_on_overflow","message_expired"]}
```

### Example Sending a File Over IMT


```
1. GET imt {}
2. 200 imt {"available":false,"mobile_terminated":false,"mobile_originated":0,"mo_status":"idle"}
3. PUT imtConfig {"enabled":true}
4. 200 imtConfig {}
5. ...
6. GET imt {}
7. 200 imt {"available":true,"mobile_terminated":false,"mobile_originated":0,"mo_status":"idle"}
8. PUT imt {"file":"thisfile.bin","topic_id":80}
9. 200 imt {}
10. GET imt {}
11. 200 imt {"available":true,"mobile_terminated":false,"mobile_originated":1,"mo_status":"active"}
```

#### 8.7.2. imtConfig

The PUT **imtConfig** command is used to enable or disable the IMT service and the transmission of files over the IMT service.

The GET **imtConfig** command returns the service configuration..

| Name    | Type    | Description                                                                                                                                 | GET | PUT | RSP |
|---------|---------|---------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| enabled | Boolean | Enable or Disable IMT service                                                                                                               |     | ✓   | ✓   |
| send    | Boolean | Set true to enable the sending of files using the IMT service. Set false to disable IMT file transmission.<br>Default: true.<br>Added: v2.0 |     | ?   | ✓   |

| Name | Type | Description                                                                                                                                                                                                                       | GET | PUT | RSP |
|------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
|      |      | <p> <b>Note:</b> Setting the “send_now” field of an <b>imt</b> command will override this setting and will allow files to be sent using IMT.</p> |     |     |     |

Example:


```

1. PUT imtConfig {"enabled":true,"send":false}
2. 200 imtConfig {}
3.
4. GET imtConfig {}
5. 200 imtConfig {"enabled":true,"send":false}

```

### 8.7.3. imtMoList

The GET **imtMoList** command returns the list of files currently queued for transmission over IMT.

| Name     | Type     | Description                                                                                                                                                                                 | GET | PUT | RSP                                                                                  |
|----------|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|--------------------------------------------------------------------------------------|
| mo_queue | [String] | List of file names queued to send as MOs via IMT. If the name of the final item is “...” it indicates that more items exist which have not been displayed due to buffer length constraints. |     |     |  |

Example:


```



1. GET imtMoList {}
2. 200 imtMoList {"mo_queue":["file1","file2","file3"]}

```

### 8.7.4. imtTopicList

The GET **imtTopicList** command returns the list of IMT topics known to STREAM+ currently.

 **Note:** The IMT topic list is cached on the device, and is not guaranteed to be updated when provisioning is adjusted on the server. It is therefore recommended to send an MT to new topics once they have been provisioned in order to push a topic list updated to the STREAM+.

| Name   | Type     | Description                                                  | GET | PUT | RSP                                                                                   |
|--------|----------|--------------------------------------------------------------|-----|-----|---------------------------------------------------------------------------------------|
| topics | [String] | List of available IMT topics                                 |     |     |  |
| ids    | [Number] | List of topic ID numbers corresponding to the “topics” list. |     |     |  |

| Name      | Type   | Description                                                                                                                                                                                                                                                                                                                                                           | GET | PUT | RSP |
|-----------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| topic_age | Number | Number of seconds since the last time the IMT topic list was refreshed.<br>This value does NOT represent the last time the topics were updated in the device provisioning.<br>This key is not contained in the response if the topic list has not been refreshed since powering up the device.<br>The maximum returned value is 604800 (number of seconds in 1 week). |     |     | ✓   |

Example:

```
1. GET imtTopicList {}
2. 200 imtTopicList {"topics":["Topic1","Topic2"],"ids":[80,81],"topic_age":333}
```

## 8.8. GNSS Commands

### 8.8.1. gnss

The **gnss** command is used to retrieve information on the latest GNSS position fix.

| Name      | Type   | Description                    | GET | PUT | RSP |
|-----------|--------|--------------------------------|-----|-----|-----|
| latitude  | Number | Device latitude in degrees     |     |     | ✓   |
| longitude | Number | Device longitude in degrees    |     |     | ✓   |
| accuracy  | Number | Location accuracy in meters    |     |     | ✓   |
| year      | Number | UTC Year                       |     |     | ✓   |
| month     | Number | UTC Month (1-12)               |     |     | ✓   |
| day       | Number | UTC Day (1-31)                 |     |     | ✓   |
| hour      | Number | UTC Hour (0-23)                |     |     | ✓   |
| minute    | Number | Minute (0-59)                  |     |     | ✓   |
| second    | Number | Second (0-60)                  |     |     | ✓   |
| fixtype   | Number | 0: No Fix<br>1: Dead-reckoning |     |     | ✓   |

| Name     | Type   | Description                                                                                      | GET | PUT | RSP |
|----------|--------|--------------------------------------------------------------------------------------------------|-----|-----|-----|
|          |        | 2: 2D Fix<br>3: 3D Fix<br>4: Dead-reckoning + GNSS fix<br>5: Time-only fix<br>255: GNSS disabled |     |     |     |
| num_sat  | Number | Number of visible satellites                                                                     |     |     | ✓   |
| lock_sat | Number | Number of locked satellites                                                                      |     |     | ✓   |

Example:

```
1. GET gnss {}
2. 200 gnss {"latitude":45.3787364,"longitude":-63.2342355,"accuracy":4041,"year":2022,"month":6,"day":30,"hour":17,"minute":3,"second":24,"fixType":3,"num_sat":20,"lock_sat":14}
```

### 8.8.2. gnssConfig

The **gnssConfig** command controls and configurable GNSS settings.

☑ **Note:** At least one key must be applied when issuing this command with PUT or and error will be returned.

| Name    | Type    | Description                          | GET | PUT | RSP |
|---------|---------|--------------------------------------|-----|-----|-----|
| enabled | Boolean | GNSS power enabled if true           |     | ?   | ?   |
| pps     | Boolean | PPS signal output is enabled if true |     | ?   | ?   |

Example:

```
1. GET gnssConfig {}
2. 200 gnssConfig {"enabled":false,"pps":false}
3.
4. PUT gnssConfig {"enabled":true}
5. 200 gnssConfig {}
```

## 8.9. Other Commands

### 8.9.1. ppp

The PUT **ppp** command is used to start the Point-to-Point Protocol server running on the Iridium Certus 9770 modem, and provide access through the STREAM+ RS232 serial port. In order for the **ppp** command to succeed, the STREAM+ must not already be connected to the Iridium network. This can be verified using the **iridium** command and confirming the “enabled” key is false:

```
1. GET iridium {}
2. 200 iridium {"enabled":false,"active_services":[]}
```

Once PPP has started the serial port will switch to PPP communications mode until either a stop command is issued, or the escape character (**Esc** on the keyboard) is encountered 3 times in a row.

**i** In order to prevent the PPP connection from disconnecting unintentionally, the Esc character (ASCII 0x1B) should be added to the client’s ACCM configuration. The Linux pppd program supports this using the “**asynmap 80a0000**” command line option.

The GET **ppp** command provides information about the PPP service.

| Name    | Type    | Description                                                                                                                                                           | GET | PUT | RSP |
|---------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| enabled | Boolean | Enable or disable the PPP service                                                                                                                                     |     |     |     |
| port    | String  | Serial port to use for PPP connection (Optional if executed from RS232, or must be set to “rs232” if executed from RS485)<br>Default: same as command was received on |     |     |     |

Example:

```
1. GET ppp {}
2. 200 ppp {"enabled":false}
3.
4. PUT ppp {"enabled":true,"port":"rs232"}
5. 200 ppp {}
6. <STREAM+ enters PPP mode>
```

### 8.9.2. iridium

The GET **iridium** command will return Iridium service information such as active status and signal level.

| Name            | Type     | Description                                                                                                                                                                                                                                                                                                                                                       | GET | PUT | RSP |
|-----------------|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|-----|-----|
| enabled         | Boolean  | Status of the Iridium Certus 9770 modem                                                                                                                                                                                                                                                                                                                           |     |     | ✓   |
| fault           | Boolean  | Indicates that modem was disabled due to a fault condition. This key is only present when the modem has been shut down due to a fault.<br>Added: v2.0                                                                                                                                                                                                             |     |     | ?   |
| signal          | Number   | Iridium signal level in bars. This key is only present when active<br>Values: 0 to 5                                                                                                                                                                                                                                                                              |     |     | ?   |
| provisioned     | Boolean  | Network provisioning status. This key is only present when active                                                                                                                                                                                                                                                                                                 |     |     | ?   |
| ip_ready        | Boolean  | Data network connection availability. This key is only present when active                                                                                                                                                                                                                                                                                        |     |     | ?   |
| imt_ready       | Boolean  | IMT connection availability. This key is only present when active                                                                                                                                                                                                                                                                                                 |     |     | ?   |
| active_services | [String] | String array of STREAM+ services using the network<br>Potential services: <ul style="list-style-type: none"> <li>ftp: FTP outbox/inbox handling is active</li> <li>imt: IMT message handling active</li> <li>ppp: PPP passthrough mode is active</li> <li>update: Modem firmware update in progress (no other services may run during firmware update)</li> </ul> |     |     | ✓   |

Example:

```
1. GET iridium {}
2. 200 iridium {"enabled":false,"active_services":[]}
```

```
1. PUT imtConfig {"enabled":true}
2. 200 imtConfig {}
3. ...
4. GET iridium {}
5. 200 iridium
{"enabled":true,"signal":5,"provisioned":true,"ip_ready":true,"imt_ready":true,"active_services":["i
mt"]}
```

## 8.10. API Responses to Commands

| Response Code | Description                                            |
|---------------|--------------------------------------------------------|
| 200           | Command was executed successfully                      |
| 401           | Method (GET/PUT) not supported for this command target |
| 402           | Invalid command length                                 |
| 404           | Command target unknown                                 |
| 406           | Operation not allowed                                  |
| 407           | Bad JSON syntax or invalid parameter setting           |
| 408           | Request failed                                         |
| 409           | Command/option unauthorized                            |
| 417           | Invalid file size                                      |
| 418           | File not found                                         |
| 419           | File is already queued                                 |
| 420           | File queue is full                                     |
| 500           | CRC validation failed                                  |
| 501           | CRC missing                                            |
| 502           | Resource required was busy                             |

## 9. FTP Implementation

STREAM+ allows for file transfers to/from the device by acting as an FTP client.

FTP initiates a passive FTP connection (PASV) in order to upload files to a specified server and download files from said server.

Files downloaded from the server can either all be retrieved or selectively retrieved based on the STREAM+ configuration (see [API Networking Commands](#) section for more detail).

☑ **Note:** The firewall running on the server must be set up to allow incoming TCP connections from ports 1024-65535 on port 21 and the port assigned to FTP data connections. By default, the data port can be any number between 1024-65535. Check your server software manual to determine if you can set up a more limited data port range if required.

### 9.1. FTPS Option

STREAM+ allows the connection to use FTPS if more security is required.

When using FTPS the server must support the following features:

- The server must support TLS 1.2
- The server must support explicit FTPS (implicit FTPS is not supported)

The FTPS server must have a TLS certificate signed against one of the following certificate authorities (self-signed certificates are not currently supported):

- Certum Trusted Network CA
- DigiCert Baltimore Root
- DigiCert EV.cer
- DigiCert Global Root G2
- DigiCert Global Root
- DigiCert
- GlobalSign Root CA - R1
- GlobalSign Root CA - R3
- Go Daddy Class 2 Certification Authority 46
- Go Daddy Root Certificate Authority - G2
- ISRG Root X1
- Sectigo (AAA)
- VeriSign Universal Root Certification Authority
- VeriSign

## 10. IMT Implementation

STREAM+ allows for file transfers to/from the device via Iridium Certus IMT protocol.

Files sent from STREAM+ are referred to as mobile originated (MO) messages.

Files sent to the STREAM+ are referred to as mobile terminated (MT) messages.

Files transferred over IMT to/from STREAM+ must be at least 1 byte and no more than 100000 bytes in size.

Sending of MO files can be deferred until the host system initiates sending the files, even when the modem itself is turned on by making use of the “send” key in the *imtConfig* API command. While the modem is disabled the “send\_now” key in the *imt* API command can be set “false” to avoid immediately powering on the modem.

MT files may be retrieved if available any time that the modem is enabled (whether enabled specifically for IMT actions or for FTP actions). This cannot be disabled, so if IMT is being used it is recommended that the “mobile\_terminated” key be checked in the *GET imt {}* command response and shut down only when it is set to *false* to avoid loss of MT data.

### 10.1. MT File Naming

MT files can be view using the command: `GET fileList {"imt":true}`.

Files are name according to the following scheme:

`<topic_id>-<count>-<topic_name>-<message_id><extension>`

- **topic\_id:** This is the topic id number that is associated with the received IMT message. Values range from 64 to 65535.
- **count:** The count increments each time a new message is received. Values range from 0 to 4294967295.
- **topic\_name:** This is value is related to the name of the topic provisioned for the device. It will contain the name string supplied when the device was provisioned and in some cases an additional numeric value. If a message was received from an unrecognized topic ID, the `topic_name` will be set to “<unknown>”.
- **message\_id:** A temporarily unique number from 0-255 supplied by the Iridium server’s messaging system.
- **extension:** For successfully received MT messages this extension is not included. If the final CRC check fails when receiving an MT the extension will be “.bcrc”. For other errors (timeouts, file IO errors), the extension will be “.err”.

If an error occurs in receiving an MT (indicated by file extension), the received contents of the MT message (which in theory include a CRC16 as the final 2 bytes) are contained within the file so that the host system can choose whether to discard the contents or attempt to parse some useful data from them first.

If the file was successfully received (no file extension), the CRC bytes are trimmed off the end of the data so that only the original MT sent to the device is retained.

## 10.2. Firmware v1.x File Naming

Firmware v1.x used a different naming scheme for IMT files:

<topic>-<count><extension>

- **topic:** This is the topic string that is associated with the received IMT message. If a message was received from an unrecognized topic ID, the topic string will be “unknown”
- **count:** The count increments each time a new message is received on a topic (value resets when the device is reset, or in some circumstances when an updated topic list is received by the modem).
- **extension:** For successfully received MT messages this extension is not included. Partially received MTs will have a “.part” extension (file however will contain no data during this state). If the final CRC check fails when receiving an MT the extension will be “.bcrc”.

## 10.3. IMT Topic List

The list of topics known to a stream+ can be enquired using the GET imtTopicList {} command. This command will return the most recently cached list of topics provisioned for the device. The topic list is updated on the modem during MO/MT transmissions or on a schedule set of the Iridium network.

If immediate feedback is desired after updating the provisioned topic list, it is recommended that a new MT be sent to new topics after provisioning to ensure the modem is made aware of the new topics in a timely manner.

MT data can be sent either by specifying the “topic” key which includes the human readable string used when provisioning the topic, or by specifying the “topic\_id” which is a number assigned to the topic by the Iridium network. It is recommended whenever possible to use the “topic\_id” key, as from time to time the formatting of the “topic” string may change even after initial provisioning.

## 11. Internet Pass-Through Setup

Setting up the STREAM+ for using the Point to Point Protocol over Serial (PPPoS) starts with the same hardware setup required for any other Iridium-based functionality. Importantly the Iridium antenna must be attached, have a clear view of the sky, and the modem must be provisioned for data services.

- ❗ If any other STREAM+ services are actively using the Iridium network (FTP, IMT, etc.) when the PPP command to start pass-through mode is issued, STREAM+ will report an `API_REQUEST_FAILED` error.

### 11.1. Serial Port

Available network speed through the PPP link will be artificially limited by the STREAM+ serial interface if the baud rate is left at the default 19200. Increasing the baud rate will ensure available bandwidth is only limited by the Iridium network.

```
1. PUT serialConfig {"port":"rs232","baud":230400}
```

- ✔ **Note:** The configuration **must** be confirmed by successfully issuing any command with the new baud settings within 30 seconds of issuing `PUT serialConfig`. Otherwise the setting will be reverted.
- ✔ PPPoS mode is only available on STREAM+'s RS232 serial port.

### 11.2. PPP Command

The STREAM+ will not be able to enter PPP pass-through mode if there is an existing service making use of the Iridium network. This is guaranteed to be the case if no network-accessing commands have been issued since the last time the STREAM+ was enabled or reset with the `PUT reset {}` command.

When in doubt the status of the Iridium network connection, including a list of services currently making use of it, may be queried with the `GET iridium {}` command. A response of `200 iridium {"enabled":false,"active_services":[]}` indicates that it is safe to use PPP.

When not using the `splus_init.chat` script provided in the following section the STREAM+ command to start the PPP server is:

```
1. PUT ppp {"enabled":true,"port":"rs232"}
```

Pass-through may be disabled by re-issuing the `ppp` command with `enabled` set to `false` (over the serial port not being used for PPP).

```
1. PUT ppp {"enabled":false}
```

It may also be canceled directly from the pass-through port by sending 3 consecutive `Esc` characters.

The `GET ppp {}` command queries the status of the PPP connection, and the response will include the active serial port if it is running.

- ✔ **Note:** There will be a delay between the `ppp` command being issued and the PPP interface becoming responsive; the time required to initialize the modem and register on the Iridium network. With good signal conditions 30 seconds is usually sufficient, but with poor signal this may take some time.

It is possible to confirm the PPP server is active by sending the `\r` character; if the string `ERROR` is returned then the pass-through interface is active. At this point a regular PPP client should be able to establish a connection through the serial link.

PPP is ready if it responds:

```
PUT ppp {"enabled":true,"port":"rs232"}
200 ppp {}
ERROR
```

- ❗ Because the `Esc` character is used to generate the escape sequence, it is recommended to add it to the PPP client's ACCM map. See the `asynctmap` argument for `pppd` in the section below.

### 11.3. Chat Script

The `pppd` program requires an external utility to open a connection to the PPP server, normally the `chat` program is used with a small script detailing the “dial-up” procedure. There are 2 options provided for this script: `splus_init.chat` and `splus.chat`

#### `splus_init.chat`

```
1. ABORT 'NO CARRIER'
2. ECHO OFF
3. TIMEOUT 10
4. SAY "Enabling PPP pass-through on STREAM+...\n"
5. '' 'PUT ppp {"enabled":true}'
6. '200 ppp {}' ''
7. SAY "Waiting for Iridium connection...\n"
8. TIMEOUT 60
9. '' ''
10. ERROR-''-ERROR 'ATDT *99#'
11. SAY "Starting PPP session\n"
12. TIMEOUT 10
13. ABORT ERROR
14. CONNECT '\dCLIENT\c'
15. CLIENTSERVER
```

The first script handles sending the `PUT ppp {"enabled":true}` command so it doesn't need to be issued separately. Also note that this feature is only supported on the RS232 serial port.

If `STREAM+` fails to establish a connection to the Iridium network before the timeout period, causing the connection to fail, `STREAM+` will remain in PPP pass-through mode. If this happens then the next script should be used for follow-up connection attempts, as it skips the `PUT ppp {"enabled":true}` command.

## splus.chat

```
1. ABORT ERROR
2. ABORT 'NO CARRIER'
3. ECHO OFF
4. TIMEOUT 10
5. SAY "Starting PPP session...\n"
6. '' 'ATDT *99#'
7. CONNECT '\dCLIENT\c'
8. CLIENTSERVER
```

This second chat script is essentially a trimmed-down version of `splus_init.chat`, skipping the `PUT ppp {"enabled":true}` command. This allows follow-up connection attempts without disrupting an established Iridium connection if STREAM+ managed to connect to the network in the meantime.

### 11.4. Start pppd

At this point everything is ready for the `pppd` program to open a connection to the Iridium modem. The shell command for starting this service on a standard Ubuntu Linux system is:

```
1. sudo pppd /dev/ttyUSB0 230400 192.168.1.2:192.168.1.1 asyncmap 80a0000 noauth xonxoff usepeerdms
defaultroute replacedefaultroute connect "chat -s -S -f /path/to/script.chat"
```

📌 **Note 1:** Replace `/path/to/script.chat` with the actual location of the above `chat` script.

📌 **Note 2:** Replace `/dev/ttyUSB0` with the serial device attached to STREAM+ (and `230400` with the baud).

This will execute `pppd` as a background service connected to STREAM+. Depending on system setup this may be sufficient to start using the Internet normally, however in the latest versions of Ubuntu `pppd` may not be able to set the local DNS nameservers. If this is the case the following shell command may be used after `pppd` to add the DNS server addresses separately:

```
1. sudo resolvectl dns ppp0 8.8.8.8 8.8.4.4
```

If the `resolvectl` command fails, then DNS servers may be manually added to the network configuration by adding the following lines to `/etc/resolv.conf`:

```
1. nameserver 8.8.8.8
2. nameserver 8.8.4.4
```

#### Breakdown of the `pppd` options:

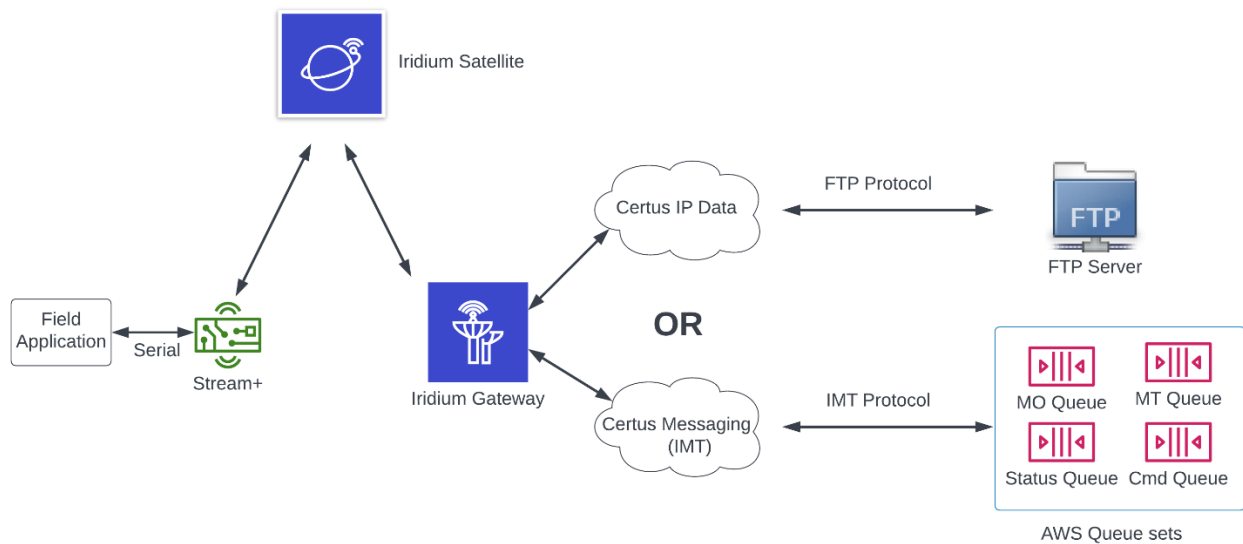
| Option                                            | Value                                | Description                                 |
|---------------------------------------------------|--------------------------------------|---------------------------------------------|
| <i>(positional)</i>                               | <code>/dev/ttyUSB0</code>            | STREAM+ serial port                         |
| <i>(positional)</i>                               | <code>230400</code>                  | STREAM+ serial baud rate                    |
| <code>&lt;local-ip&gt; : &lt;remote-ip&gt;</code> | <code>192.168.1.2:192.168.1.1</code> | Define local and remote interface addresses |

| Option              | Value                                | Description                                                                                                                                                                                                     |
|---------------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| asyncmap            | 80a0000                              | The ACCM configuration defines ASCII control characters that need to be (transparently) escaped over the serial link. The value 80a0000 specifies <code>xON</code> , <code>xOFF</code> , and <code>Esc</code> . |
| noauth              |                                      | No username/password authentication                                                                                                                                                                             |
| xonxoff             |                                      | Software flow control                                                                                                                                                                                           |
| usepeerdns          |                                      | Use the DNS servers suggested by the remote                                                                                                                                                                     |
| defaultroute        |                                      | Make this interface the default route to the internet                                                                                                                                                           |
| replacedefaultroute |                                      | This option is a flag to the defaultroute option. If defaultroute is set and this flag is also set, pppd replaces an existing default route with the new default route.                                         |
| connect             | "chat -s -S -f /path/to/script.chat" | External program used to "dial" the modem; the Linux <i>chat</i> utility in this case.                                                                                                                          |
| debug               |                                      | Output verbose debugging information.                                                                                                                                                                           |
| nodetach            |                                      | Don't detach from the terminal and start running as a background process.                                                                                                                                       |

## 12. Certus

### 12.1. Certus Service

A high-level architecture of the STREAM+ Certus service is presented in the figure below.



*Figure 24: Architecture of STREAM+ CERTUS Service*

A Field application communicates with the STREAM+ device via the serial port. The data received by the STREAM+ device is transmitted through the Iridium satellite and Gateway using one of the following services Certus IP Data or Certus Messaging.

### 12.2. Certus IP Data

The Certus IP Data service enables the STREAM+ device to establish a connection to the internet with speeds up to 88 kbps upload speeds and 22kbps download speeds (real-life speeds averaging ~20kbps up and down). The STREAM+ implements the FTP protocol, so any data transmitted by the STREAM+ device over the Certus IP Data is downloaded to a file on an FTP server.

### 13. Mounting Instructions

There are several ways to mount the device including bolts, zip ties and hose clamps. The largest recommended width for a strap fastener is  $5/16"$ . STREAM+ can be mounted in any orientation. If mounted vertically in an environment where it will be exposed to water, ensure the connectors are facing downwards.

It is recommended to use #8 fasteners to mount the device through the 4 mounting holes on the baseplate, but the holes can also accommodate #10 fasteners (tight fit). Insulating washers should be used between the fasteners and the baseplate.

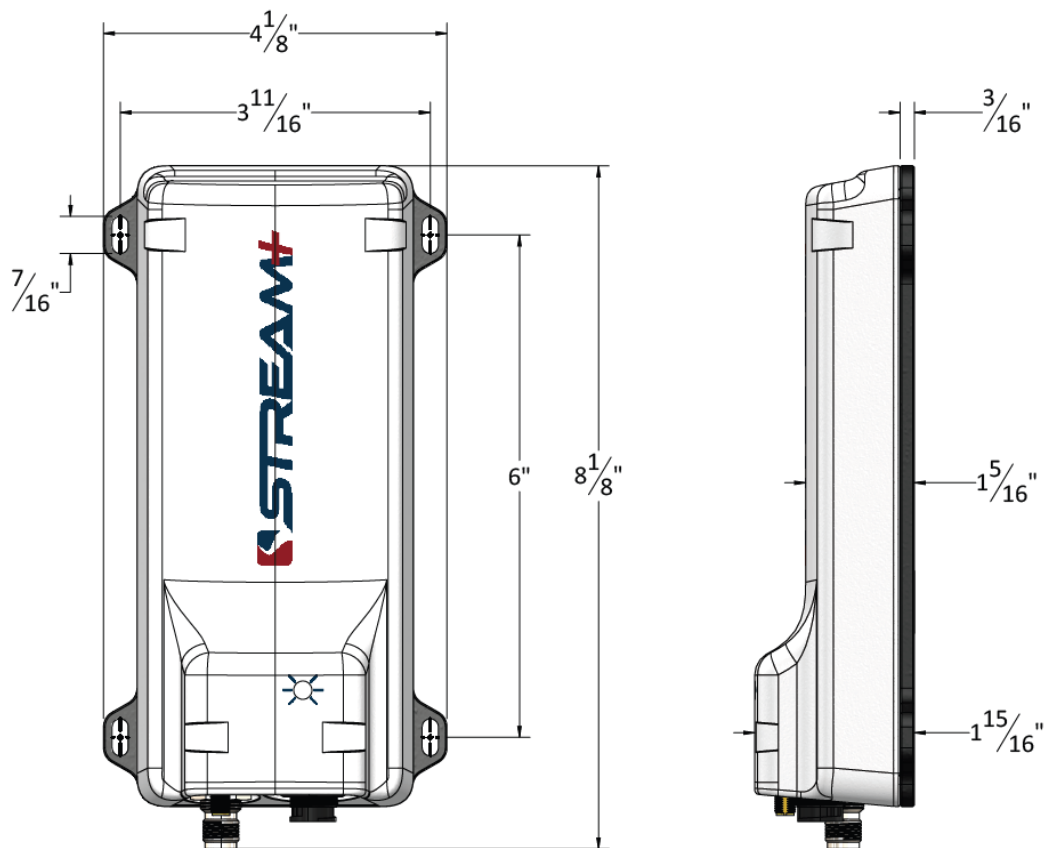
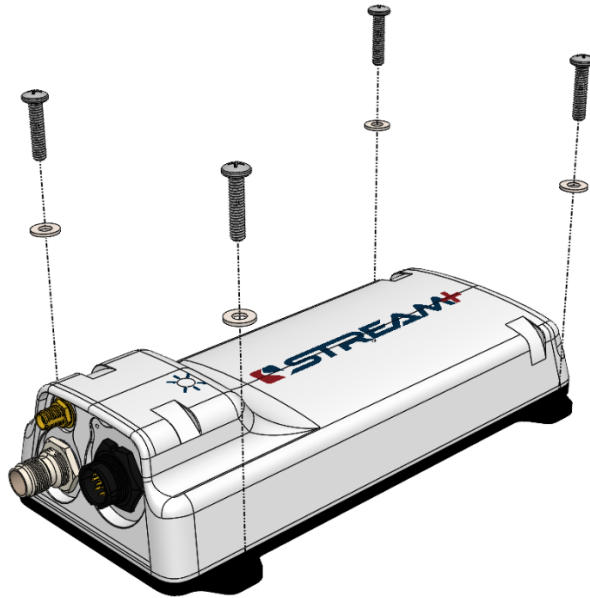


Figure 25: Mounting Holes Location

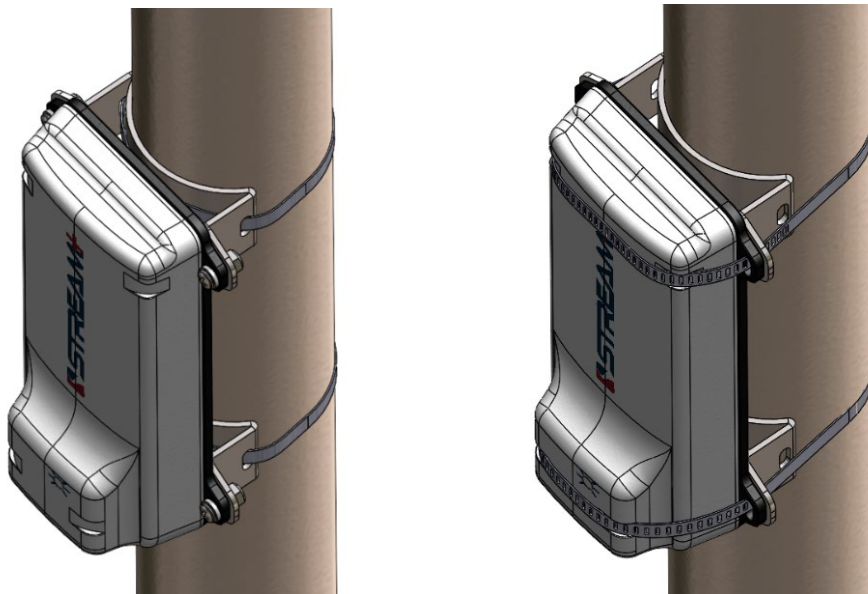
If the device is to be mounted on a flat plate, use the above dimensions to layout the through holes.

**Horizontal Mounting:** To mount the device to a surface use M4 or #8 size screws to secure the unit in place. A minimum screw length of  $3/8"$  is recommended. It is also recommended to use plastic washers between the mounting screws and the device to prevent damage to the anodized finish. The maximum washer OD is  $3/8"$ .



*Figure 26: Horizontal Mounting*

**Vertical Mounting:** The device can be either mounted to the pole using #8 screws, or directly to the pole using zip ties or hose clamps. The device must be mounted with the connectors facing downwards to avoid water pooling in the vent hole and connector panel (see image below).





*Figure 27: Vertical Mounting Examples*

## 13.1. Antenna Mounting Instructions

The two antennas provided in the Developers Kit can be pole mounted. Both antennas can be mounted on standard 1" 14 TPI (1"-14NF) mast mounts. The Iridium and GNSS antennas should be mounted at least 30cm (12") apart.

### 13.1.1. Tools Required

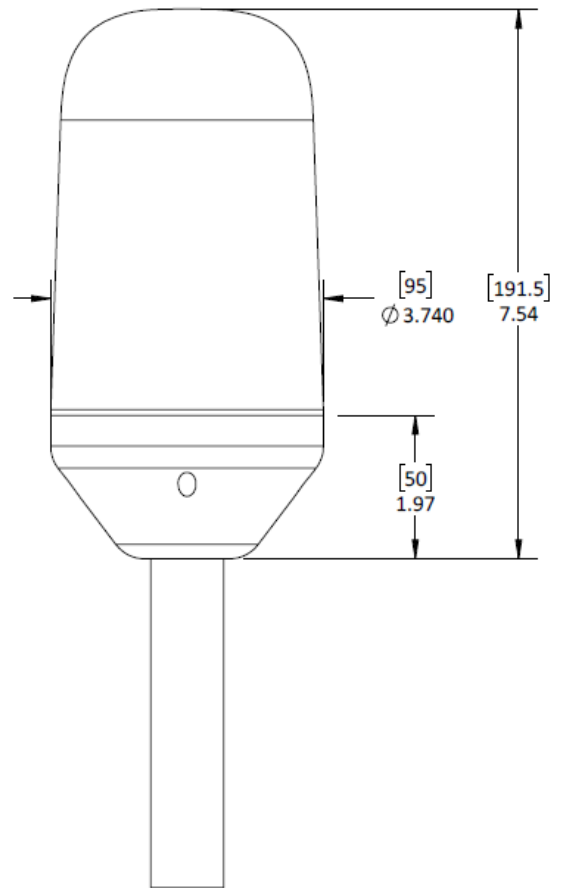
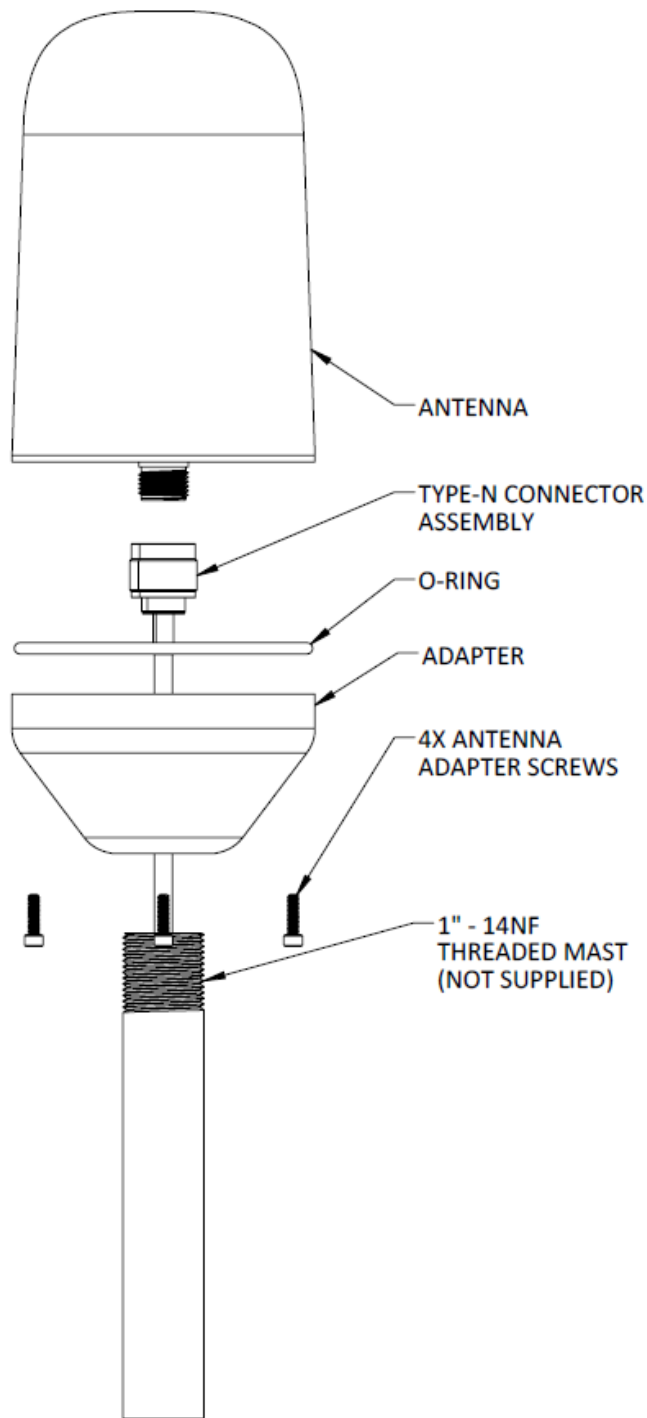
- Hex Key (included)

To mount the Iridium antenna (65020-011 SCAN Antenna), the pole mounting base adapter must first be removed.

The N-TYPE terminated cable can then be passed through the base and connected to the antenna.

The base can then be mounted to the antenna using the 4 screws (hex 4-40) and the provided hex key, and the whole assembly can be mounted to the mast.

The antennas can also be mounted to a free-standing mast using u-bolts.



4 screws (hex 4-40) and hex key (included)

Figure 28: 65020-011 SCAN Antenna (Iridium)

The 2J GPS antenna can be mounted in a similar way but does not need to be disassembled. The RF cable simply need to be connected to the device and then the assembly can be mounted on the mast.

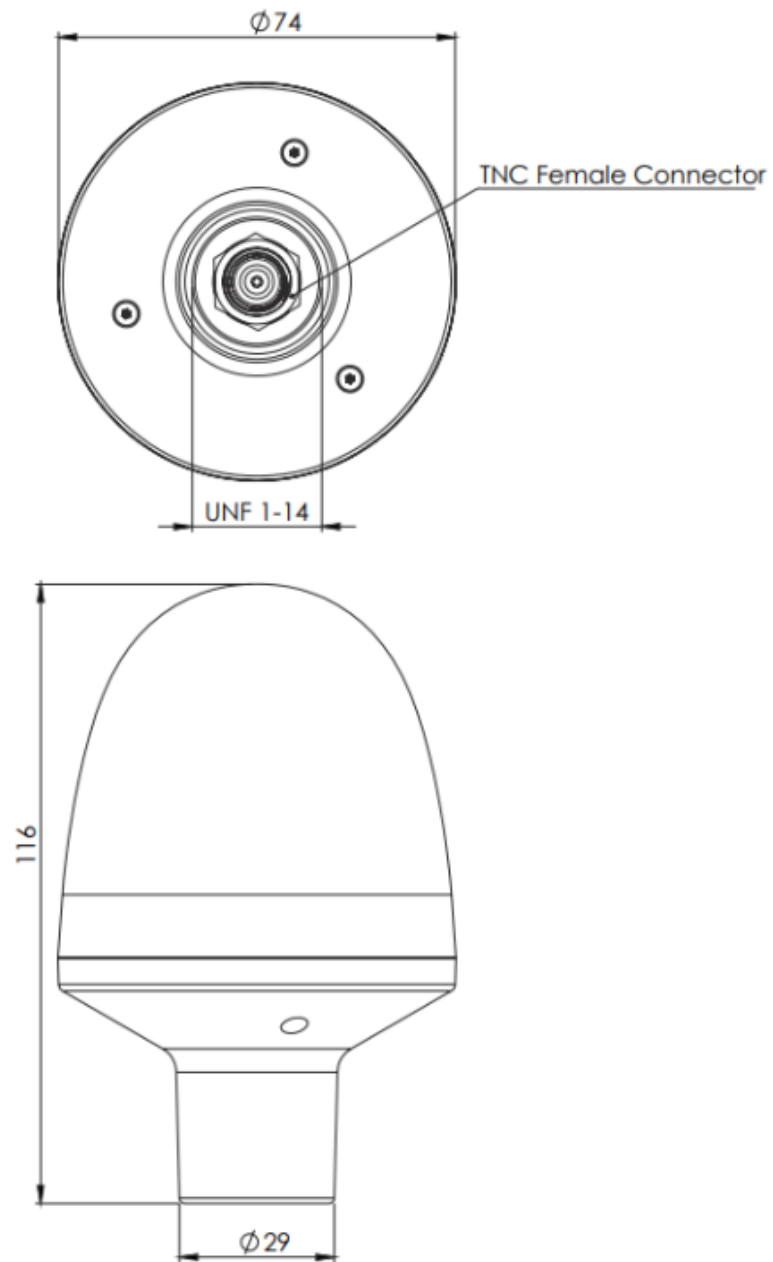


Figure 29: 2J Antenna (GPS)

## 14. Maintenance

There are no user serviceable parts in the STREAM+ development kit. Return the unit to the factory for repairs.

**Power:** Power off the device when not in use.

### O-Ring:

- a. Replace worn or damaged O-ring.
- b. Protect O-ring: We recommend using Parker Super O Lube to protect O-ring surface from damage.

Contact technical support at [support@metocean.com](mailto:support@metocean.com) or call Toll Free +1 (800) 565-1830 Ext (333) for device repairs and maintenance.

### 14.1. O-Ring Maintenance

The figure below shows the location of the O-ring on the external Iridium Antenna. To access the O-ring, remove the base of the antenna (Iridium). Lubricate the O-ring lightly with silicone grease; the recommended grease is Parker Super O Lube.



*Figure 30: Location of the O-Ring – Antenna (Iridium)*

**Remove the O-ring and lubricate at least once a year, preferably more often, and replace every 2 years or sooner.**

### 14.2. O-Ring Facts

The main problem with O-rings is that over time they can take a “set” shape. They lose their original round shape and they become slightly flattened resulting in less compressibility.

Reduced compressibility can lead to a water leak. The best way to prevent a problem is to service the O-rings before every deployment. Some of the main factors that can cause a “set” are:


- High storage temperature (exceeding 40°C) over time will cause the rubber to harden.
- Exposure to sunlight and ozone will cause the O- ring to deteriorate and stiffen.
- Temperature does not have to be that cold for an O- ring to lose a great deal of its flexibility. At 4°C, an O-ring has stiffened significantly.
- The O-ring becomes less flexible as the rubber ages.
- If you are operating or storing at low temperature, it makes it even more important to service the O-ring regularly.

### 14.3. O-Ring Care and Maintenance

This section provides details on the recommended procedure for care and maintenance of the O-ring:

**O-Ring Removal:** Use a small, plastic blunt-tipped tool to remove the O-ring from its groove. Check the O-ring and the groove for any excess grease or contamination.

If contamination is evident, clean the O-ring and groove with lens cleaning tissue or equivalent lint-free cloth. Alcohol can be used to clean the plastic parts but use only water and mild detergent on the O-ring itself.

|                                                                                     |                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p><b>CAUTION</b></p> <p>DO NOT over-grease the O-Ring. Excess grease can collect grit particles that can compromise the seal. If excess grease is present, remove it using lint-free cloth.</p> |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Also, inspect the O-ring for nicks and imperfections. Using alcohol on O-ring may cause a loss of elasticity and may promote cracking.

DO NOT use a sharp object or metal to remove the O-ring. This can result in damages to the O-rings or the groove.

Before re-installing the O-ring, make sure that you are using a clean workspace and clean hands and are avoiding contact with anything that may leave fibers on the O-ring or groove. Even a very small bit of contamination (hair, grit, etc.) may cause a catastrophic leak.

**O-Ring Re-Installation:** Place a small amount of grease between your thumb and index finger. We recommend using Parker Super O Lube for re-installation of the O-rings. Draw the O-rings through the grease while pressing the fingers together. Use this action to place a very light covering of grease to all sides of the O-ring.

Place the O-ring into its groove making sure that it does not twist or roll. Use the previously grease-coated finger to once again lightly go over the mating surface of the O-ring.

DO NOT use excess grease on the O-ring or the O-ring groove. Carefully mate the two surfaces together being careful to not pinch the O-rings.

## 15. Specifications

### 15.1. Electrical Specifications

|                              |                            |
|------------------------------|----------------------------|
| <b>Supply Voltage</b>        | 10-40V DC Supply           |
| <b>Outputs</b>               | Protected Open Drain       |
| <b>Supply Power</b>          | 30W peak, 10W max avg      |
| <b>Visual Indicator</b>      | LED Status Indicator       |
| <b>Circuit Protection</b>    | Automotive Pulse Protected |
| <b>Standby Mode Current</b>  | 20 mA, at 12 V             |
| <b>Shutdown Mode Current</b> | 4.5 uA, at 10-40 V         |
| <b>Max EIRP</b>              | 10 W                       |

Table 3: Electrical Specifications

### 15.2. Mechanical Specifications

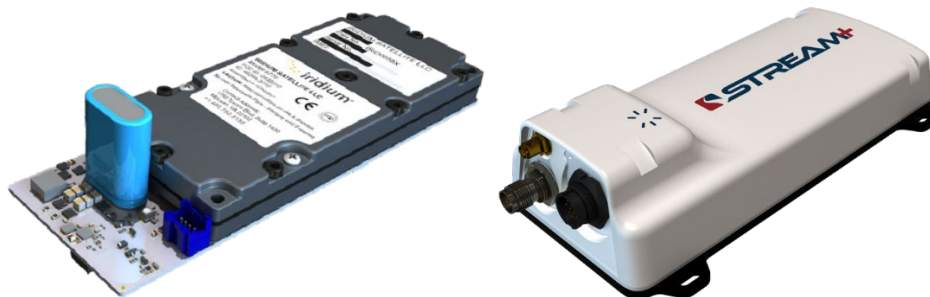
|                           |                                                        |
|---------------------------|--------------------------------------------------------|
| <b>Overall Dimensions</b> | 7.7" (195.5 mm) L x 4.1" (104 mm) W x 1.9" (48.2 mm) H |
|                           |                                                        |
| <b>Material</b>           | Polycarbonate                                          |
| <b>Colour</b>             | White                                                  |
| <b>Weight</b>             | 620 g (21.9 oz.)                                       |

|                               |                                                      |
|-------------------------------|------------------------------------------------------|
| <b>Data / Power Connector</b> | 10 Pin Mini-Con-X, Male Pins, Bayonet Lock Connector |
| <b>GNSS Antenna</b>           | Female SMA Connector                                 |
| <b>Iridium Antenna</b>        | Female TNC Connector                                 |
| <b>Thermal Management</b>     | Integrated Aluminium Heat Sink                       |

*Table 4: Mechanical Specifications*

### 15.3. Physical Product

|                             |                                                                                                                      |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>uController</b>          | Controls file and message transfers<br>Drives the status lines<br>Provides a simple JSON API through the serial port |
| <b>Power Control Line</b>   | Allows an external controller to switch the device on/off                                                            |
| <b>Status Lines</b>         | Message/File Inbox state (open drain)<br>Message/File Outbox state (open drain)<br>GNSS pulse per second             |
| <b>Hardware Interfacing</b> | RS232 interface<br>RS485 interface (half-duplex)                                                                     |
| <b>Flash Memory</b>         | Extra storage for large file transfers                                                                               |
| <b>GNSS</b>                 | GNSS Antenna: 2J 2J9C01JWCF-C392N, external active antenna                                                           |
| <b>Iridium Interface</b>    | External Passive Certus 100 Iridium Antenna, Aero AT1621-142 or SCAN 65020-011                                       |



*Figure 31: STREAM+ Physical Product*

### 15.4. Environmental Specifications

|                                          |                                |
|------------------------------------------|--------------------------------|
| <b>Operating and Storage Temperature</b> | -40°C (-40°F) to +70°C (158°F) |
| <b>Ingress Protection</b>                | IP67                           |

*Table 5: Environmental Specifications*

\*All specifications are subject to change without notice to improve reliability, function, or design or otherwise. Please contact technical support at [support@metocean.com](mailto:support@metocean.com) for more information.

## 16. Acronyms

| Name           | Meaning                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------|
| ACK            | Acknowledgement                                                                                        |
| EIRP           | Effective, or Equivalent, Isotropic Radiated Power                                                     |
| FTP            | File transfer protocol                                                                                 |
| FW             | Firmware                                                                                               |
| GNSS           | Global Navigation Satellite System. GNSS includes all geo-location standards worldwide, including GPS. |
| IMEI           | International mobile equipment identifier                                                              |
| LED            | Light emitting diode                                                                                   |
| MDF            | Message Descriptor File                                                                                |
| MO             | Mobile originated. A message that originates on the STREAM+ device is a MO message.                    |
| MT             | Mobile terminated. A message that is received by the STREAM+ device is a MT message.                   |
| OTA            | Over The Air. OTA specifically refers the messaging over the Iridium satellites.                       |
| PC             | Personal Computer                                                                                      |
| PCB            | Printed circuit board                                                                                  |
| RS-232         | Recommended standard 232                                                                               |
| RS-485         | Recommended standard 485                                                                               |
| SBD            | Short Burst Data. Iridium's store and forward messaging system.                                        |
| SMA            | SubMiniature version A                                                                                 |
| STREAM+ device | The physical device, also referred to as "the Device"                                                  |
| TIC            | Test interface card                                                                                    |
| TNC            | Threaded Neill-Concelman                                                                               |

## 17. Certifications

- FCC:
  - Tested to and conforms with FCC parts 15b and 25.
  - Contains FCC ID Q639770.
- ISED:
  - Contains IC 4629A-9770.
  - Tested to ICES-003 ISSUE 7 – October (2020).
- Iridium:
  - Iridium-Certified Certus 100 device, SLBT4/L1 Class Iridium Compatible Equipment
- RED Directive 2014/53/EU (CE) – pending.
- Australia and New Zealand Radio Type Approval (RCM) – pending.
- UKCA – pending.

## 18. Warranty and Liability Statements

### 18.1. Liability Statement

In no event will MetOcean Telematics or its employees, affiliates, agents, or licensors be liable for any indirect, incidental, special, consequential, or exemplary damages, including but not limited to damages for any loss of revenues, profit, goodwill, use, data, or other intangible losses arising out of or related to the use of this product and or services affiliated with the use of this product.

### 18.2. Warranty

MetOcean Telematics products are designed and manufactured to the highest standards and deliver high-quality performance, ease of use and ease of installation.

If you encounter any difficulties while using your product, we recommend that you first consult the user manual.

#### 18.2.1. Warranty Terms

MetOcean warrants products manufactured by MetOcean to be free from defects in materials and workmanship under normal use and service for twelve (12) months from the date of purchase, unless specified otherwise, subject to the following conditions:

1. This warranty shall not apply to any MetOcean products that have been subjected to tampering, modification, misuse, neglect, accidents of nature, or shipping damage.
2. MetOcean's obligation under this warranty is limited to replacing devices that fail within the first thirty (30) days of purchase. After the first thirty (30) days, MetOcean's obligation is to repair the devices that fail within the first twelve (12) months from the date of purchase.
3. MetOcean will return warranted equipment by surface carrier, prepaid.
4. There will be no out-of-warranty RMAs.

This warranty, and MetOcean's obligation there under, is in lieu of all other warranties, expressed or implied, including warranties of suitability and fitness for a particular purpose. MetOcean is not liable for consequential damages.

#### 18.2.2. Technical Support

For product support, please contact MetOcean's support team at (902) 468 2505 ext. 333 or by email: [support@metocean.com](mailto:support@metocean.com)

## 19. Appendix A

### 19.1. STREAM+ Block Description

The Test interface card can be sectioned into 7 blocks:

1. Power Control
2. LED Control Jumpers
3. STREAM+ Connector
4. nShutdown Control
5. STREAM+ I/O
6. RS-485
7. RS-232

These blocks are shown in the image below:

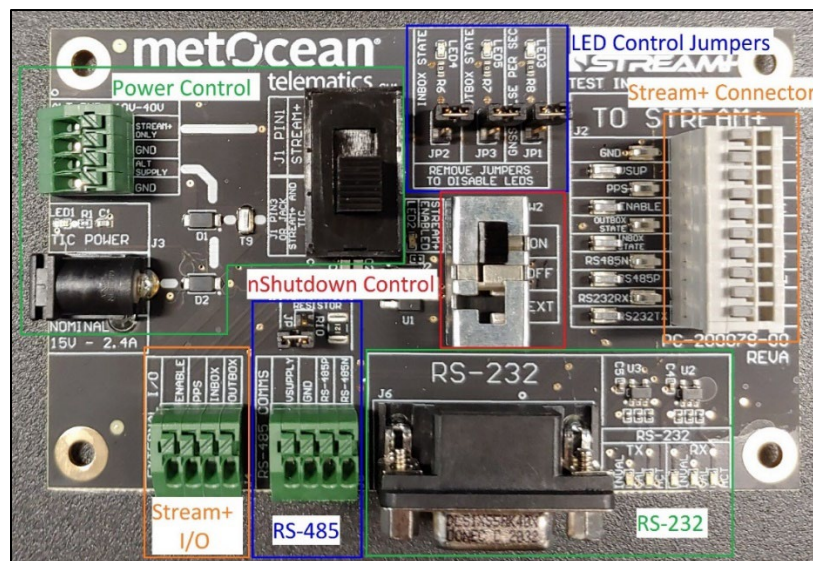


Figure 32: Block Description

#### 19.1.1. Power Control

This section defines what voltage the STREAM+ receives. A 15V wall wart power supply is provided with the development kit, but other power supplies within the prescribed voltage range (10-40V) can be used. The wall wart should be plugged into J3. To use the wall wart option, ensure SW1 is in the lower position. If using a different power supply, use either “STREAM+ Only” or “ALT Supply” on J1.

Use ALT SUPPLY to apply a voltage other than 15V to STREAM+. This will power the TIC as well as the STREAM+ device. To use this option, slide SW1 into the lower position.

Use “STREAM+ Only” (top connection, pin 1) to pass the applied voltage directly to the STREAM+ device, bypassing the TIC. This will allow the user to more accurately assess the current draw of the STREAM+ device. To use this option, slide SW1 into the upper position. Using this option will disable all of the TIC LEDs.

### 19.1.2. LED Control Jumpers

Connect these jumpers to enable the TIC LEDs corresponding to Inbox state, Outbox state, and Pulse Per Second. Leave them disconnected to use offboard pullup resistors or LEDs.

### 19.1.3. STREAM+ Connector

Connect desired conductors from the 10-conductor cable to STREAM+ via this connector. Not all conductors need to be connected for operation, but at minimum, GND (Black), VSUP (Red), and either RS-232 (Grey and white) or RS-485 (green and yellow) are required to operate and communicate with the STREAM+.

The conductor colour corresponding to each connector pin is printed on the PCB for ease of use (this applies to the provided cable only - this may not apply to custom cable assemblies).

**Note:** If not connecting nShutdown (Orange), ensure it is fully floating, not resistively connected to either electrical or earth ground as this will prevent the device from turning on.

### 19.1.4. nShutdown Control

This switch provides control over how the STREAM+ device is enabled. If not using an external controller to pull this line low to disable the STREAM+, toggle between the upper position to enable the device and the middle position to disable the device.

**Note:** Using the lower position (EXT) will also enable the device if nothing is connected to the nShutdown pin of J4.

To use an external controller and use the TIC as a passthrough, keep the slide switch in the lower (EXT) position. As on the STREAM+ device itself, floating this connection or driving it high will enable the STREAM+, and pulling it to ground (even through a weak pulldown resistor) will disable the device.

### 19.1.5. STREAM+ I/O

This connector, J4, breaks out the non-critical connections of STREAM+ to a customer’s controller. With the exception of nShutdown, these are protected open collector outputs on the STREAM+ and can be used with pullup resistors or LEDs to communicate the state of these outputs.

The maximum pullup voltage allowed is 40V, and the recommended pullup resistance is 47kOhms (use at least 20k to ensure the “LOW” voltage is below 1V when applying 40V).

LEDs may be used with or without current-limiting resistors. The maximum current that will be drawn (when 40V is used to supply the LED or resistor) is 10mA. These connections may be used simultaneously with the on-board LEDs.

☑ **Note:** The onboard LED states for these I/Os correspond to when there is current flowing into the collectors, resulting in the LEDs being ON when the voltage at the corresponding pin is LOW.

#### 19.1.6. RS-485

This connector breaks out the half-duplex RS-485 connections from the STREAM+. The Vsupply and pin corresponds to the supply voltage of the TIC and only needs to be connected if the customer RS-485 transceiver requires it. It may be left open circuit.

Ground should be connected to the RS-485 transceiver, and it is strongly recommended that the 120R termination resistor be used via JP4. Connect this jumper across the two pins to use this resistor.

#### 19.1.7. RS-232

Plug in a male DB9 with TX, RX, and GND connected to pins 2, 3, and 5 respectively. The other pins are not used. The LEDs in the corner are present to provide visual feedback of RS-232 connectivity.

If a red LED is lit, there is an error with the connection and two-way communication will not take place.

Two green LEDs indicates proper connection, and the blue LEDs will flash to indicate TX/RX activity.

## 20. Appendix B

### 20.1. Antenna Considerations

The following antenna/cable pairs have been tested with STREAM+:

| Model               | Cable        | Antenna                     |
|---------------------|--------------|-----------------------------|
| STREAM+             | 0.5m LMR-200 | AeroAntenna, PN: AT1621-142 |
| STREAM+             | 2.5m LTR-240 | SCAN Antenna, PN: 65020-011 |
| STREAM+ OEM Dev Kit | 2.5m LTR-240 | SCAN Antenna, PN: 65020-011 |

*Table 6: Antenna Considerations*

Only these Iridium cable/antenna pairs are approved by Iridium and MetOcean Telematics to use with STREAM+ or STREAM+ OEM, since they are the only ones tested for Iridium and Regulatory (e.g. FCC, CE) approval.

The GNSS antenna requirements are much less strict: any active antenna can be used which accepts 3.3V as the DC supply voltage. Cable lengths for GNSS are undefined, but as an example the Development kits use a 4m LMR-100 cable which performs well.

### **CANADA (HEAD OFFICE)**

MetOcean Telematics  
11 Thornhill Drive  
Dartmouth, NS, B3B 1R9  
Fax: +1 902 468 4442

[sales@metocean.com](mailto:sales@metocean.com)

+1 902 468 2505

### **UNITED STATES**

MetOcean Telematics  
4296 Forbes Blvd.  
Lanham, Maryland  
20706, United States

[sales@metocean.com](mailto:sales@metocean.com)

+1 800 565 1830

### **UNITED KINGDOM**

MetOcean Telematics  
Titchfield Lane,  
Wickham, Fareham  
UK PO17 5NZ

[sales@metocean.com](mailto:sales@metocean.com)

+44 1489 888 555

[metocean.com](http://metocean.com)