Myriota

Myriota HyperPulse™ Developer Kit User Guide

MYRIOTA-TEC-525_v1.2
January 2026

Revision History

Rev	Date	Description of Change
1.0	September 2025	Initial version.
1.1	December 2025	- Added details regarding SIM card registration and included information on two spare SIM cards. - Introduced the DeployAssist mobile application. - Updated example firmware prints, configuration details, and the GitHub link for the Myriota HyperPulse library. - Revised Sensor Interface Support chapter. - Removed the deployment/debug mode section. - Replaced the Message Format section with a GitHub link.
1.2	January 2026	Added steps to update nRF9151 SiP NTN modem firmware.

Related Documentation

Find the latest versions of all Myriota documentation at support.myriota.com

How to Contact Us

Technical Support

support.myriota.com

Sales Support

sales@myriota.com

Myriota Online

myriota.com

Headquarters

Myriota Pty Ltd
Level 1, McEwin Building, Lot Fourteen
North Terrace, SA 5000 Australia

Disclaimer

The information contained in this document (collectively, the “Information”) is provided to you (both the individual receiving this document and any legal entity on behalf of which such individual is acting) (“You” and “Your”) by Myriota Pty Ltd for information purposes only.

- Myriota Pty Ltd reserves the right to make changes without further notice to any products herein.
- You are responsible for making Your own assessments concerning the Information and Myriota recommends that You assess the accuracy, completeness and relevance of the Information for Your purposes before using or relying on any of the Information.
- Myriota is providing the Information to you “AS IS” and without regard to Your specific requirements.
- Myriota has exercised reasonable care in preparing the Information, however Myriota does not warrant the accuracy, completeness or relevance of the Information and accepts no liability for any errors or omissions in the Information.
- You acknowledge and agree that Your use of the Information is at your sole risk and that to the extent permitted by law Myriota is not liable for any loss or damage of whatever nature (direct, indirect, consequential or other) that arises in any way from Your use of or reliance on the Information.
- For further information, see myriota.com or contact your Myriota sales representative.

Table of Contents


Myriota
Myriota Developer Kit
Myriota
HyperPulse
HyperPulse
5G NTN
Developer Kit

Introduction

Overview

Myriota's HyperPulse™ Developer Kit (DK or Devkit) allows customers to evaluate the Myriota 5G Non-Terrestrial Network (NTN). It comes pre-programmed with location, temperature, and battery voltage data reporting by default, and can be easily re-programmed via the USB.

The device can be powered by either USB or a LiPo rechargeable battery, offering flexible trials in the field.

Key Features

- Built-in Myriota secure and private satellite connectivity
- Integrated rechargeable battery for reliable and convenient operation
- Rugged IP67-rated, UV-resistant enclosure with integrated NTN and GNSS antenna
- Single USB port for both debugging and programming via CMSIS-DAP
- Flexible sensor and interface integration for rapid proof-of-concept development

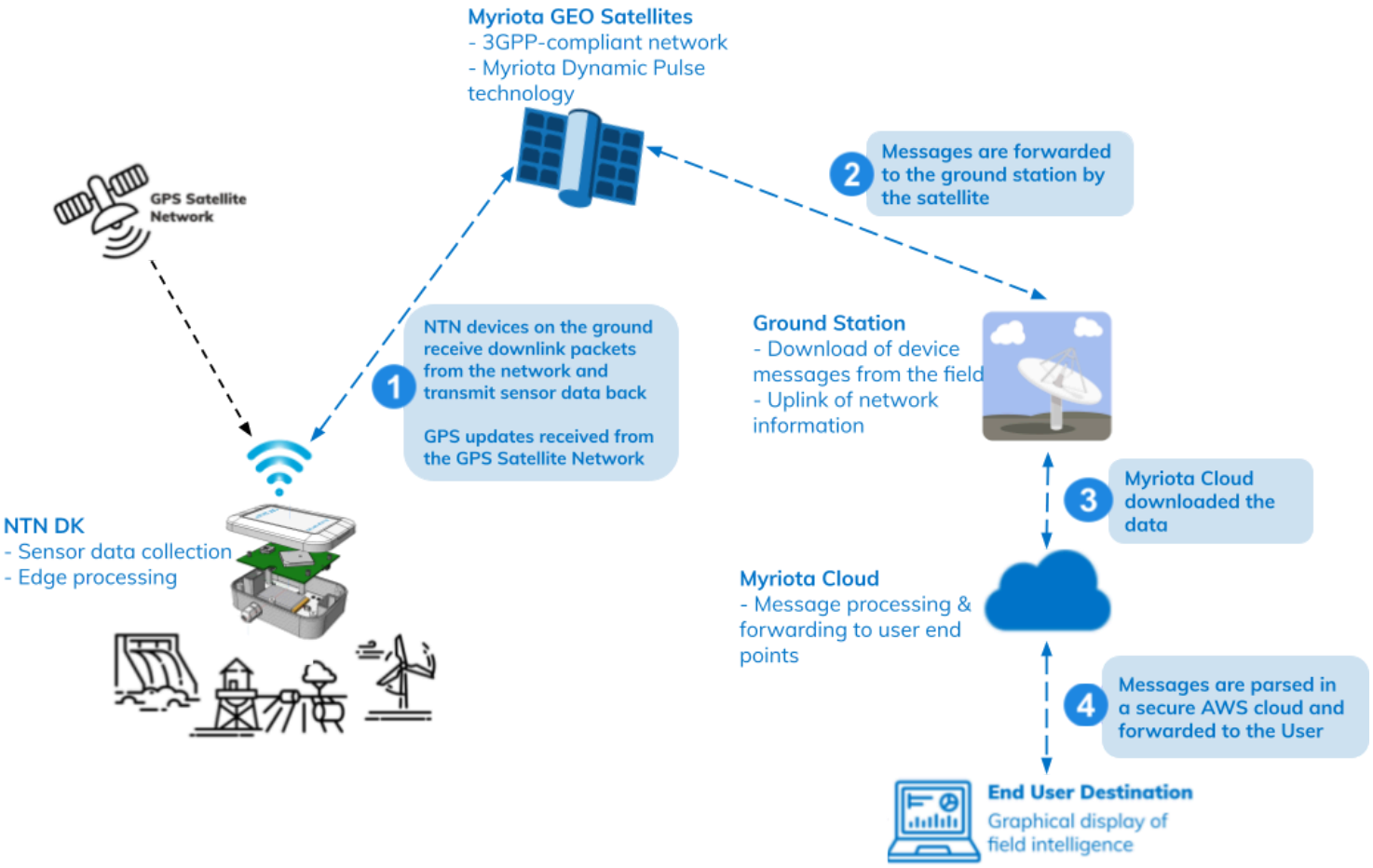
Technical Specifications

Mechanical	Operating temperature	-30°C to +70°C (-22°F to 158°F)
	IP Rating	IP67
	Dimensions	130mm L x 110mm W (with gland) x 37mm H
Onboard MCU	Flash	1 MB
	RAM	256 KB
HyperPulse Transmit & Receive	Transmit frequency	1626.5 MHz - 1660.5 MHz (3GPP n255)
	Receive frequency	1525 MHz - 1559 MHz (3GPP n255)
	Max TX output power (EIRP)	23dBm
Application Firmware	Reference applications	- Demo application with location, temperature, and battery voltage - AT Command Modem application, ready to interact with external host controllers via USB
	Custom firmware	Custom firmware development is supported
	Firmware re-programming	USB-C
Interfaces	Headers	1x Expansion header 1x Gravity 1x Qwiic 1x MikroBUS 1x JST PH 2-pin battery input
	Interfaces on the headers	1x Digital I/O 1x I2C 1x UART 1x SPI
Power Supply	Battery	Type: LiPo Operating range: 2.8-5.5V Current/power rating: Able to support at least 2W of power Battery capacity: > 300mAh - A LiPo rechargeable battery is included in the DK
	USB-C	Operating range: 5V ± 0.2V Current requirements: >= 500mA
SIM	Physical SIM card slot	1x Nano SIM (4FF)
Onboard Temperature Sensor	Measurement range	-30°C to +70°C (-22°F to 158°F)
	Accuracy	± 2°C

Message Communications

The DK communicates data to and from the field via the Myriota HyperPulse Network, as shown in the diagram below.

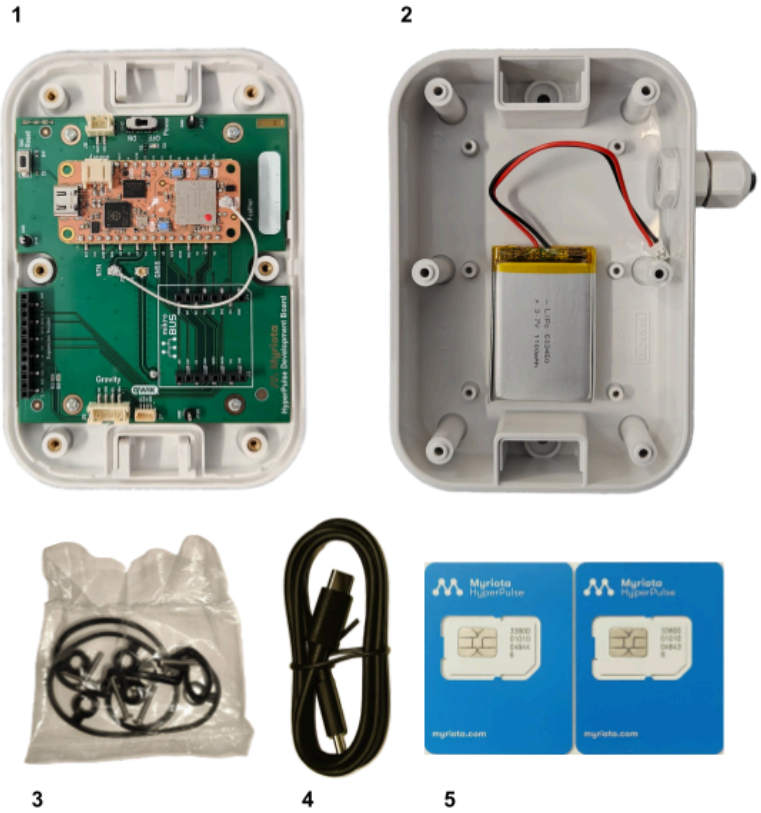
Detailed information can be found in the Myriota HyperPulse Network introduction article from Myriota Developer Site.



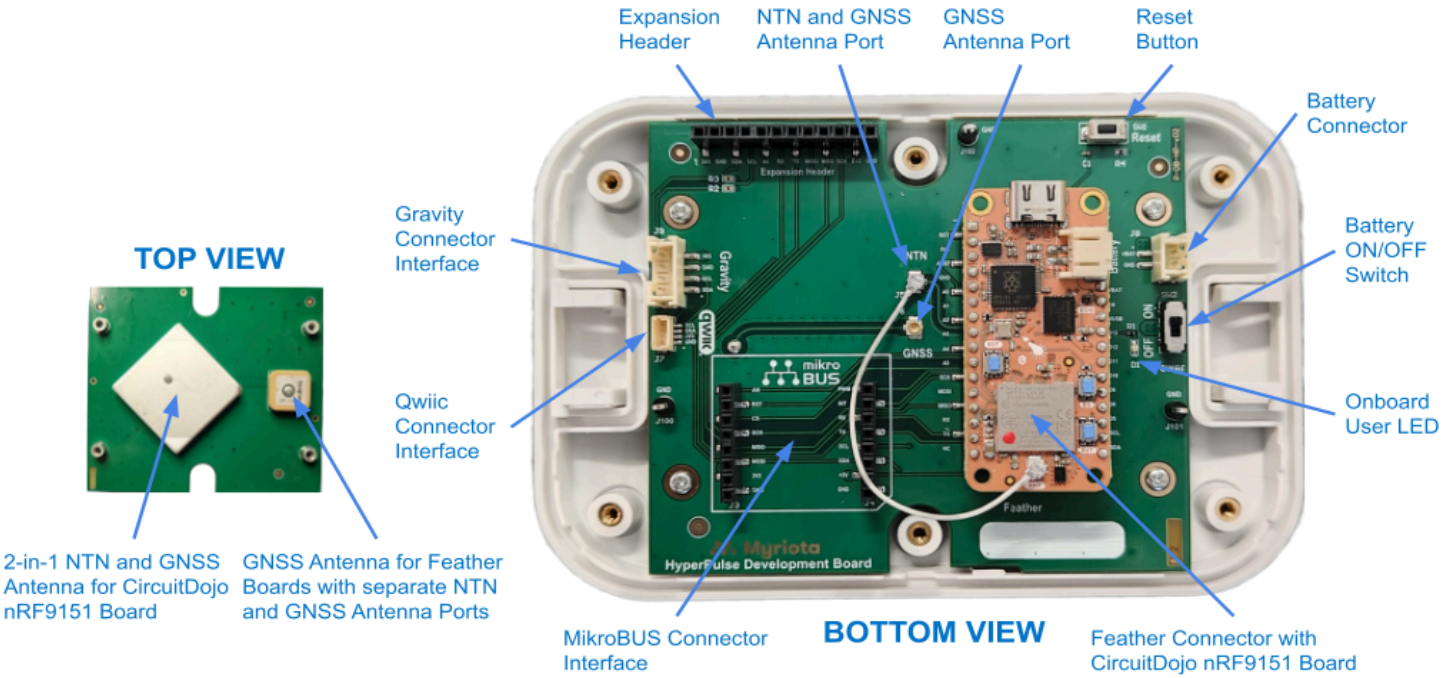
Getting Started

In the Box

1. HyperPulse Developer Board
2. LiPo rechargeable battery with JST PH 2-pin header, attached to the enclosure
3. Rubber gasket and screws
4. USB Type-C to Type-C communication cable
5. 3x Myriota 5G NTN SIM cards (1 pre-assembled SIM plus 2 spare SIMs for future use)



Board Diagram



Tools Required

Laptop or Desktop Computer

A laptop or desktop computer running Windows, Linux, or macOS, with a USB-C port capable of supplying at least 500 mA current (USB 3.0 or higher recommended).

Phillips-head Screwdriver

A #1 Phillips-head screwdriver is recommended (sizes #0-#2 are also suitable) for opening the case and accessing the battery and USB-C port.

Setting up the Developer Kit

The Myriota HyperPulse DK comes pre-programmed with a reference application that is configured to report temperature, battery voltage, and location data hourly (24 messages per day). If this behavior suits your needs, insert the batteries and deploy to start sending messages.

Alongside the Application firmware, Myriota releases a separate Network Information file containing the configurations required for satellite communication. By default, devices are pre-programmed with the latest Network Information prior to shipment.

Please refer to the Programming section when there is a need to reprogram either the User Application or Network Information.

Powering the Developer Kit

The Myriota HyperPulse DK can be powered by:

- One LiPo rechargeable battery or
- USB-C

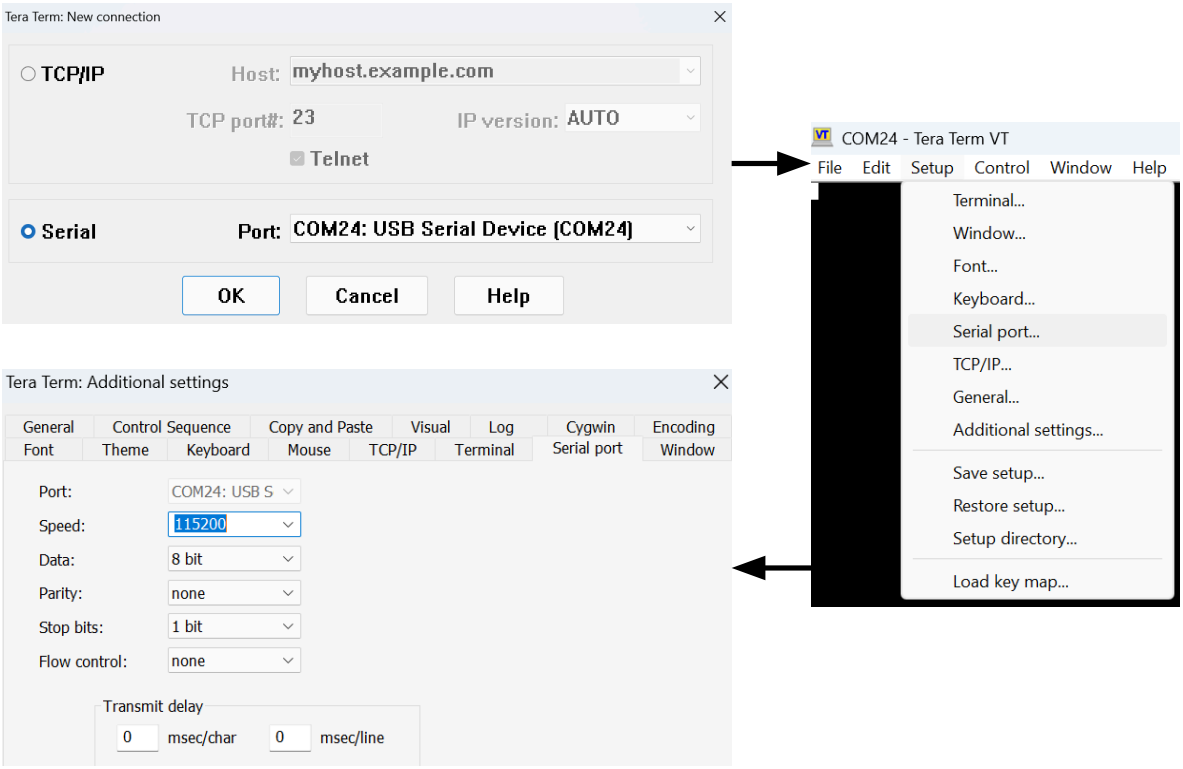
When both are connected and the battery switch is ON, the LiPo battery will be charged. It takes a few hours to fully charge the battery. It is suggested to use a multimeter to check that the battery voltage has reached the desired level before deployment.

Note: When powered via USB, the nRF9151 Feather board incurs slight additional consumption due to voltage regulation overhead.

Viewing Logs

When connected to a Windows PC via a USB-C cable, the HyperPulse DK presents a COM port. This port functions as a console for observing real-time log messages from the application. You can use any serial port emulator to monitor the device output. The default serial configuration is ***115200 baud, 8 data bits, no parity, and 1 stop bit.***

For example, Tera Term is a free Windows terminal emulator commonly used. Configure the tool following the screenshots below, your terminal will be ready to interact with the DK.



Example bootup prints:

```
*** Booting Zephyr OS build f791c49f492c ***
[00:00:00.412,353] <inf> hyperpulse: Running Myriota HyperPulse library: v1.1.0
[00:00:00.412,384] <inf> hal_sys: Reset reason: PIN reset
*** Setting antenna configuration: AT%XANTCFG=1 ***
[00:00:00.718,261] <inf> hal_at_modem: Modem restarted
[00:00:00.718,963] <inf> at_comms_host: Modem firmware: mfw_nrf9151-ntn_1.0.0-1.alpha
[00:00:00.719,635] <inf> at_comms_host: IMEI: 359404235411676
[00:00:02.753,387] <inf> at_comms_host: IMSI: 901820101001029
[00:00:02.758,148] <inf> hal_at_modem: UE not registered
[00:00:02.820,007] <inf> diagnostics: Storage metadata: 7f7213c9, config-clean,
2026-01-13T04:09:44Z, 1.0.3
[00:00:02.821,075] <inf> app_startup: *** Vsys Current Limit: 1000 mA ***

myriota:~$ Demo Application: v2.0.1
[00:00:04.822,357] <inf> configuration: Configuration initialized: period=600 s,
GNSS=disabled
[00:00:04.822,662] <inf> hw_ctrl: Console is already enabled
[00:00:04.823,638] <inf> hw_ctrl: Accelerometer disabled
[00:00:04.823,669] <inf> hw_ctrl: SW0 disabled
[00:00:04.823,730] <inf> hw_ctrl: External flash disabled
[00:00:04.828,094] <inf> hw_ctrl: Console is already enabled
[00:00:04.828,125] <inf> hw_ctrl: USB is connected
```

```
[00:00:05.846,282] <inf> gnss: Fix attempt started
[00:00:38.416,961] <inf> gnss: Fix valid!
[00:00:38.425,598] <inf> modem: #MLOCATION: acc=0 lat=-349205217, lon=1386087251,
elev=39532, timestamp=1768376748
```

The IMSI of the SIM card is only printed once at the start. If you missed the print, you can press the **reset button** on the board to reboot the device.

Upon viewing logs in real time, users will also be able to view diagnostic and message scheduling details being printed in regular intervals.

Example diagnostic and message scheduling prints:

```
[00:00:38.968,872] <inf> gnss: Fix attempt started
[00:00:39.089,416] <inf> gnss: Fix valid!
[00:00:39.097,839] <inf> modem: #MLOCATION: acc=0 lat=-349205213, lon=1386087246,
elev=39493, timestamp=1768376748
[00:00:40.982,574] <inf> periodic_uplink: Scheduled uplink message: 0 1768376750
-349205213 1386087246 39 40 4668

Scheduled uplink message (hex): 0x00000000AE496769238D2FEB4E039E522700283C12
[00:00:41.578,918] <inf> diagnostics: Diagnostics | info     | version: 808f2f47
storage: 209ea8c6
[00:00:41.578,918] <inf> diagnostics: Diagnostics | uplink   | scheduled: 1
[00:00:41.578,948] <inf> diagnostics: Diagnostics | downlink | received: 0 last: 0
```

SIM Card Registration

A Myriota HyperPulse SIM card must be registered before any data can be transmitted via the Myriota HyperPulse Network. The SIM card that comes assembled with Hyperpulse DK is already registered with your account on Myriota Device Manager and therefore ready for deployment. If you cannot see the SIM registered in your Device Manager account, please reach out to Myriota Support to get your SIM card registered.

Configuring the Firmware

This section is only relevant when the default firmware configuration does not meet your specific requirements.

While the default firmware allows for basic customisation, such as adjusting the message interval and enabling/disabling GNSS fix for each message, some use cases may require functionality beyond what can be achieved through reconfiguration. In such cases, developing custom firmware may be necessary.

Configuring the Default Firmware

You can configure the default firmware using a serial port emulator (i.e., Tera Term) with the USB cable connected to the DK. The currently supported commands can be acquired by sending **app** and **cfg** commands to the DK.

```
myriota:~$ app
app - Application commands
Subcommands:
  version  : print application version

myriota:~$ cfg
cfg - Configuration commands
Subcommands:
  period    : Get/set message schedule period in seconds (default: 3600)
  gnss_fix  : Get/set GNSS fix enable state (0=disable, 1=enable)
```

Example of reading message interval and reconfiguring it to 10 minutes:

```
myriota:~$ cfg period
3600

myriota:~$ cfg period 600
OK

myriota:~$ cfg period
600
```

Reprogramming the Developer Kit

The latest release of the pre-compiled binary files for the HyperPulse Developer Kit are available on Myriota Device Manager for download.

If you would like to program these files or any other custom applications that you've built, please follow the steps below to complete the process.

The DK can be easily reprogrammed via the USB-C port through the onboard CMSIS-DAP using the Python tool **pyocd**, which is compatible with most operating systems that support Python.

Installing pyocd

Run the following command in a command line to install the ***pyocd*** tool:

```
Shell
python3 -m pip install -U pyocd
```

Note: Python 3.x should be installed before installing pyocd

Programming Application

```
Shell
pyocd load --target nrf91 --format hex application.hex
```

Note: replace ***application.hex*** with the actual application file's directory and name

Programming Network Information

The Network Information file contains the network configurations required for the devices in the field to establish a satellite connection. The DK comes preloaded with up-to-date network information before shipment, and this file will also automatically be updated Over-The-Air (OTA) via satellite communication after deployment. However, it is always recommended to program the latest network information (*HyperPulseNetworkConfig.hex)* downloaded from the Myriota Device Manager portal before deployment for the best network experience.

Please use the following command to program the device:

```shell
pyocd load --target nrf91 --format hex HyperPulseNetworkConfig.hex
```

Note: replace ***HyperPulseNetworkConfig.hex*** with the actual network information file's directory and name.

Resetting Device

To validate the programming, a reset is required, which can be done through the following command or by physically pressing the reset button on the board.

```shell
pyocd cmd -t nrf91 -c 'reset'
```

Extracting Logs (optional)

In rare cases where a deployed device fails, the Myriota support team may request users to extract logs from the device's flash memory. These logs can help the team identify the underlying issue more efficiently.

The command below is used to extract logs from the DK to a file ***log_dump.txt***, to be shared with Myriota:

```shell
pyocd cmd --target nrf91 -c "savemem 0xC0000 0x40000 log_dump.txt"
```

Upgrading nRF9151 NTN Modem Firmware

The nRF9151 SiP contains a dedicated modem firmware image (mfw) that is separate from the application firmware. Upgrading the modem firmware ensures the modem has the required feature set, reliability, and performance for HyperPulse NTN operation.

When This Is Required

- DKs shipped before February 2026 may have an older modem firmware and must be upgraded to the officially released nRF9151 NTN modem firmware from Nordic.
- DKs shipped from February 2026 onward are expected to arrive with upgraded modem firmware preloaded (no action required unless otherwise advised).

This is a one-off step per DK. A further upgrade may be required in the future if Nordic releases an updated version of NTN modem firmware.

Check the Current Version

The modem firmware version is printed during boot. Refer to **Viewing Logs** section to locate the modem firmware revision string. If the version is **earlier than v1.0.0**, you will need to upgrade it.

Programming the Modem Firmware

1. Download the latest **nRF9151 SiP NTN firmware** zip file from Nordic's official downloads.
2. Program the modem firmware using pyOCD.

```
Shell
pyocd cmd -t nrf91 -c 'nrf91-update-modem-fw mfw_nrf9151-ntn_1.0.0-1.alpha.zip'
```

Notes:

- Replace ***mfw_nrf9151-ntn_1.0.0-1.alpha.zip*** with the actual modem firmware filename.
- On Windows, the command must be run either from the same directory as the modem firmware zip file (e.g., *C:\Users\USER_NAME\Downloads*), or you must provide the absolute path to the file. Relative paths (e.g., *.\Downloads\...*) may not work.

3. Verification

   Reboot the DK and check the modem firmware version in the logs again to confirm the upgrade was successful.

Building Your Firmware

The Myriota HyperPulse Library provides the software solution for enabling IoT connectivity over Myriota's Non-Terrestrial Network(NTN). It provides users with the necessary source files and artifacts to integrate into their applications. The library comes bundled with sample applications and reference documentation for accelerated product development. You can follow the instructions in the README.md file to set up the environment and build your own application.

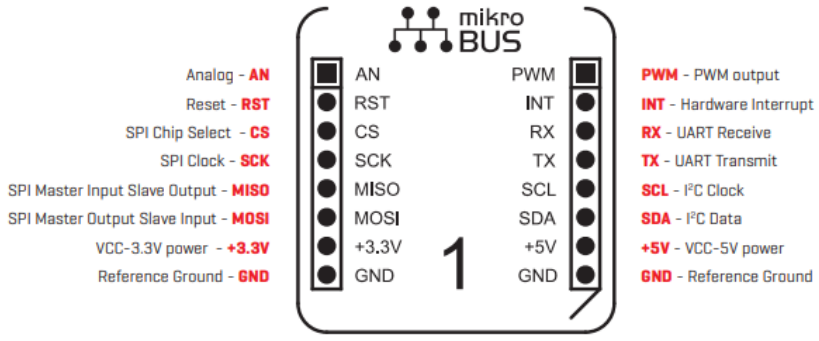
Sensor Interface Support

HyperPulse Developer Kit includes on-board hardware support for the following standard sensor interfaces:

mikroBUS™

The mikroBUS™ standard defines mainboard sockets and add-on boards used for interfacing microcontrollers or microprocessors (mainboards) with integrated circuits and modules (add-on boards)

mikroBUS™ host connector consists of two 1x8 female headers containing pins to be most likely used in the target add-on board. There are three groups of communication pins: SPI, UART and I2C. There are also single pins for PWM, Interrupt, Analog input, Reset and Chip Select. Pinout contains two power groups: +5V and GND on one header and +3.3V and GND on the other 1x8 header.



Click™ boards are popular add-on boards in mikroBUS™ form factor. Each one carries a specific functionality, whether it is a communication layer, sensor, analog input, storage module, etc. Click™ boards are compact in size and are easily integrated into target devices, making them a great choice for fast prototyping. Each comes with a user manual and a set of examples.

For a full list of supported click boards please refer to the mikroe website

Gravity

DFRobot Gravity interfaces is a colour-coded 4-pin (I2C) JST PH 2.0mm connector system for plug-and-play sensors,

Standard Gravity Pinout (4-Pin I2C)

- Red: VCC (Power)
- Black: GND (Ground)
- Blue: SDA (I2C Data)
- Yellow/White (or other colour): SCL (I2C Clock)

For a full list of supported Gravity modules/sensors refer to DFRobot website

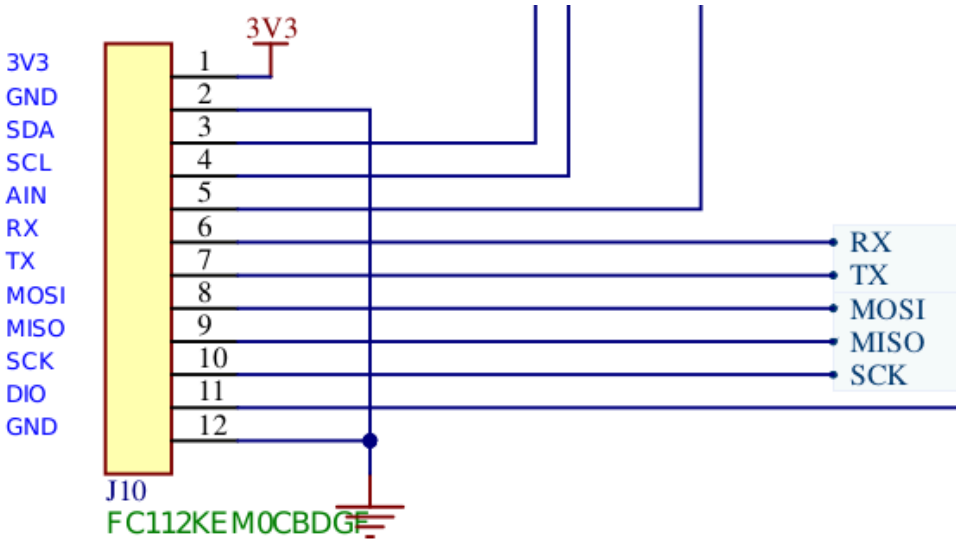
Qwiic

Qwiic is a standardised, plug-and-play connectivity system developed by SparkFun electronics. It uses a series of 4-pin JST SH connectors and cables to quickly and easily connect various sensors, displays, and other components using the I2C protocol without the need for soldering or wiring.

Please refer to sparkfun website for a collection of the compatible sensor boards.

Expansion Header

In addition to the standard hardware interfaces listed above, the Developer Kit base board also exposes some of the commonly used GPIO pins of the MCU for external interfacing.



Please note that some of these GPIOs on the expansion header are multiplexed with the sensor interfaces (Qwiic, Gravity, mikroBUS). If you wish to use the GPIOs on the expansion header in addition to the standard sensor interfaces please review the hardware schematics of the Developer Kit. Reach out to Myriota support to request the full hardware schematics of the Developer Kit.

Deployment

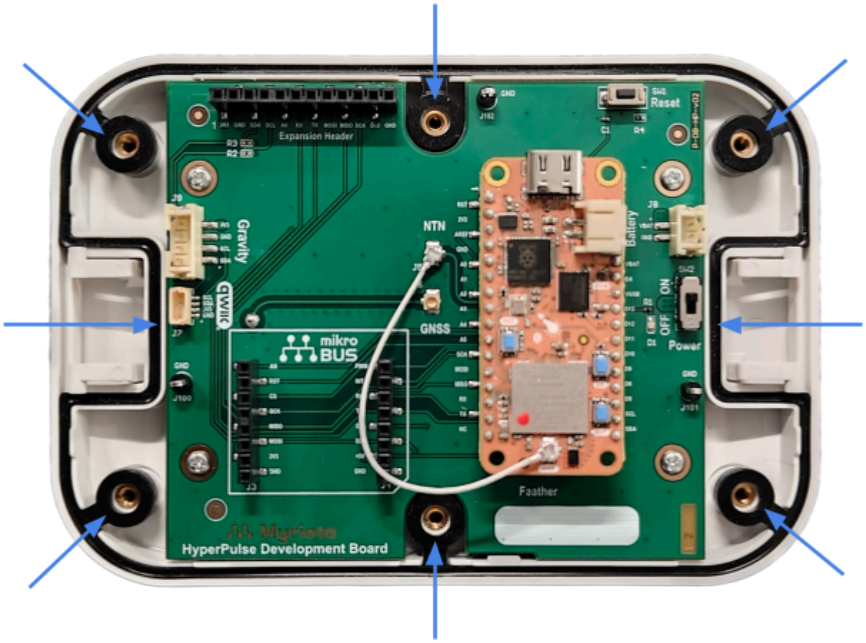
After the DK and the firmware have been properly configured, you can now deploy the DK to transmit data to satellites.

Assembling the Developer Kit

Before deploying the HyperPulse DK, it is important that the DK is properly assembled and the expected start-up sequence is observed to ensure the device operates as expected in the field. Assemble the DK following the instructions below.

Rubber Gasket

Align the rubber gasket with the groove on the top enclosure and press the gasket firmly into the groove. Pay attention to the direction of the gasket - it should be flat after correct installation.



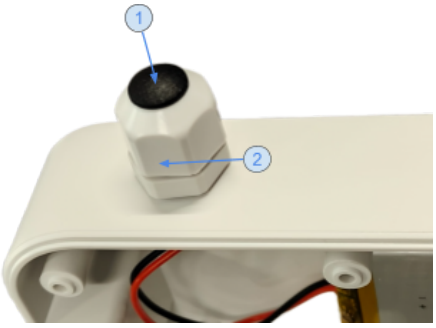
Cable Gland

When connecting external equipment (e.g., sensors) to the HyperPulse DK, use the provided gland to feed the cable through. To maintain a dustproof and waterproof seal during outdoor deployment, it's important to tighten the gland securely. Only cables with a circular

cross-section and a diameter of up to 8 mm are supported. If the gland is not in use, ensure the plastic sealing plug is properly installed.

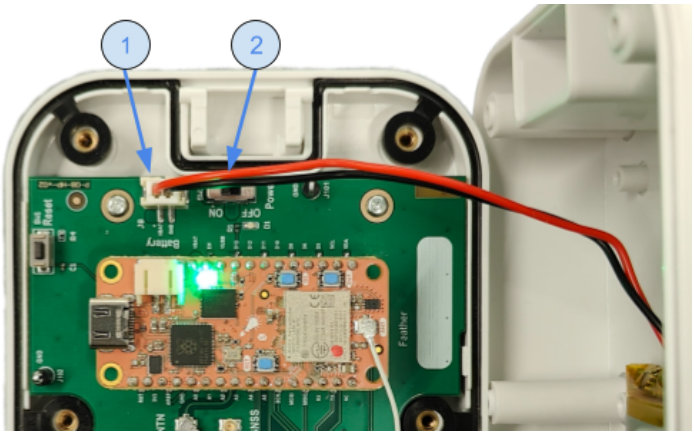
Steps to install the sealing plug:

1. Insert the plug into the gland while the gland is loose
2. Tighten the gland to secure the plug in place

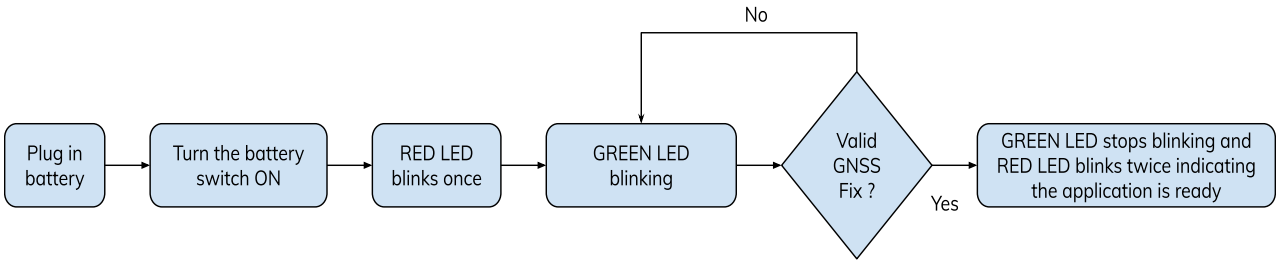


Battery

1. Plug the battery connector into the JST PH 2-pin header
2. Flip the switch to the ON position



If the DK is placed outdoors with GNSS signal availability, you should observe the following sequence. The green LED will blink before a GNSS fix is acquired.



Note: It is recommended to unplug or switch off the battery when the DK is not in use to prevent extra power consumption.

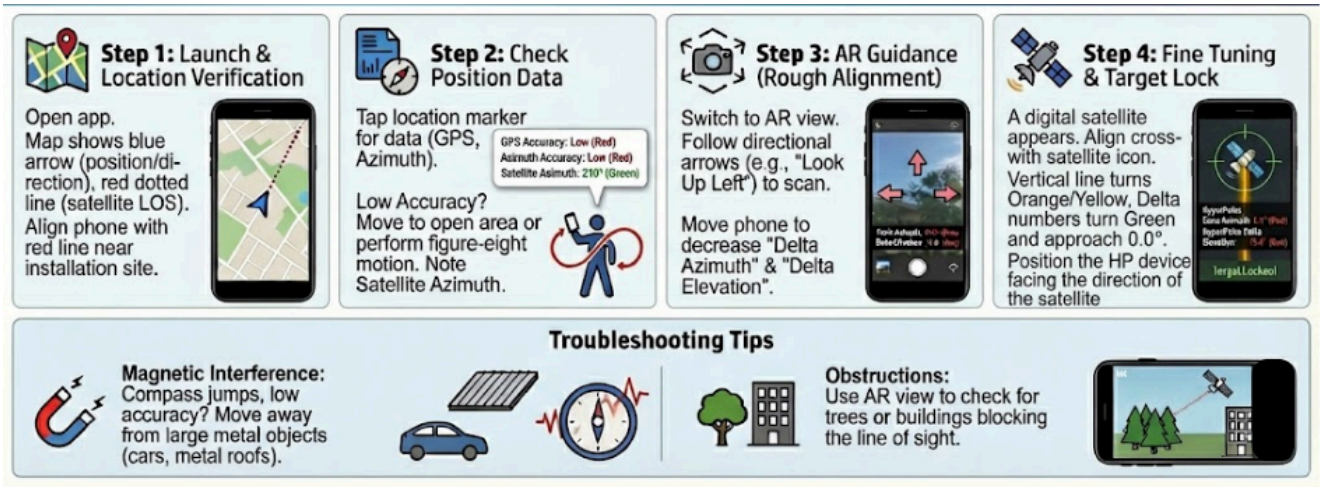
Screws

Secure all six screws using a Phillips-head screwdriver.

Locating the Satellite

To ensure optimal device performance, it is essential to follow the steps below using Myriota's HyperPulse DeployAssist Mobile App to locate the HyperPulse satellites at the point of install and ensure there is a clear line of sight to the satellite. A suboptimal deployment may significantly impact message delivery.

HyperPulse DeployAssist is the companion mobile application designed to simplify the installation and deployment of your Myriota HyperPulse devices. This application is available on Android and iOS platforms.



Detailed Instructions for using the DeployAssist Mobile app can be found here.

Mounting

When choosing a mounting position, consider the risk of the device being crushed, dislodged, or otherwise damaged.

The HyperPulse DK can be secured using industrial adhesives and should be mounted horizontally on a flat surface with the lid facing upwards. For reliable NTN communication, maintain a clear line of sight to the satellite serving your region. An unobstructed sky view also enhances GNSS performance.

Site Selection Tips

- Do not deploy indoors
- Place outdoors in a secure location with the antennas facing up, and elevate if necessary
- Keep away from strong sources of RF or electrical interference (e.g., radios, high-voltage equipment)
- Ensure antennas are free of obstructions; metal, brick, or concrete structures can fully block signals
- Be mindful that trees may intermittently block signals as they sway or when foliage is dense

Deployment Examples

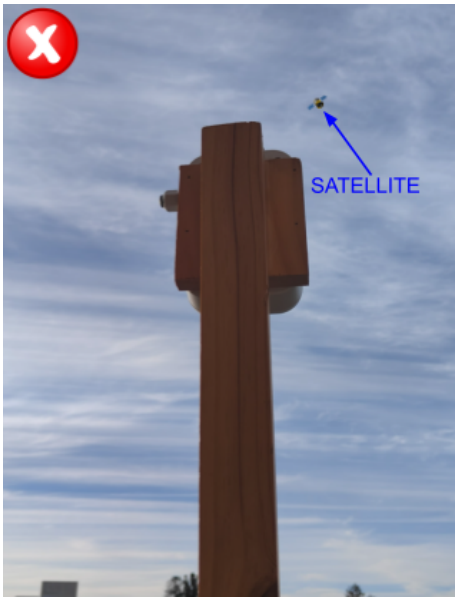
Recommended Deployments

Horizontal mounting is considered best practice and is suitable for all applications.

Problematic Deployments

In general, the DK is not suitable for indoor deployments, including placement by a window.

Below are some problematic outdoor deployment examples that should be avoided.



The above vertical mounting is considered a suboptimal deployment, although it faces the satellite:

1. Reflected satellite signals from buildings and the ground cause interference
2. Electromagnetic noise from the ground can be easily picked up by the device

As a consequence, the performance of this deployment is not guaranteed.

Despite having a clear view of the sky, the antenna is blocked by the mounting bracket and the DK circuit board. Combined with the vertical orientation, this may prevent transmission entirely.

A building is blocking the line of sight to the satellite.

The satellite signal is obstructed by trees.

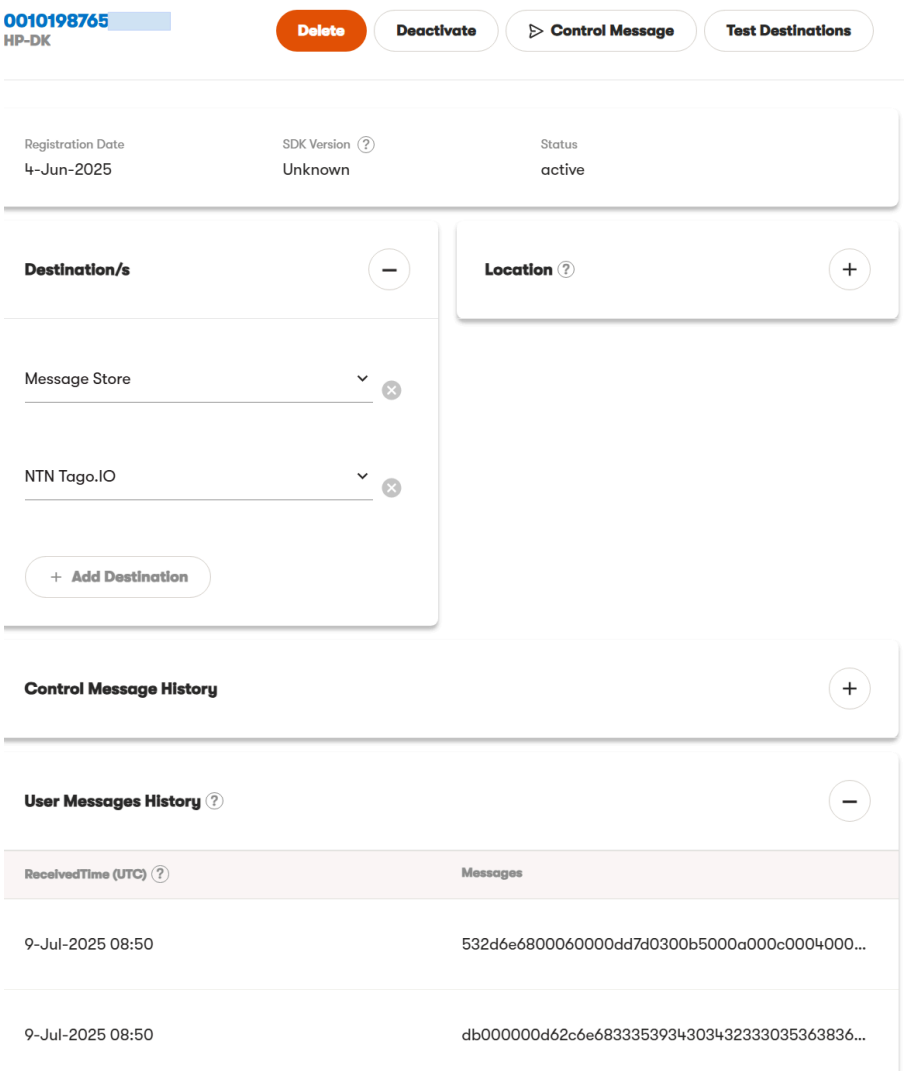
Viewing Uplink Messages

This section outlines three commonly used methods for viewing uplink messages:

- Myriota Device Manager (test only)
- TagoIO
- User-defined Destinations

Viewing Your Messages in Device Manager

To view messages from Myriota Device Manager, ensure your device includes the ***Message Store*** destination. The latest 10 messages will be displayed under ***User Messages History***.



Message Format and Decoding

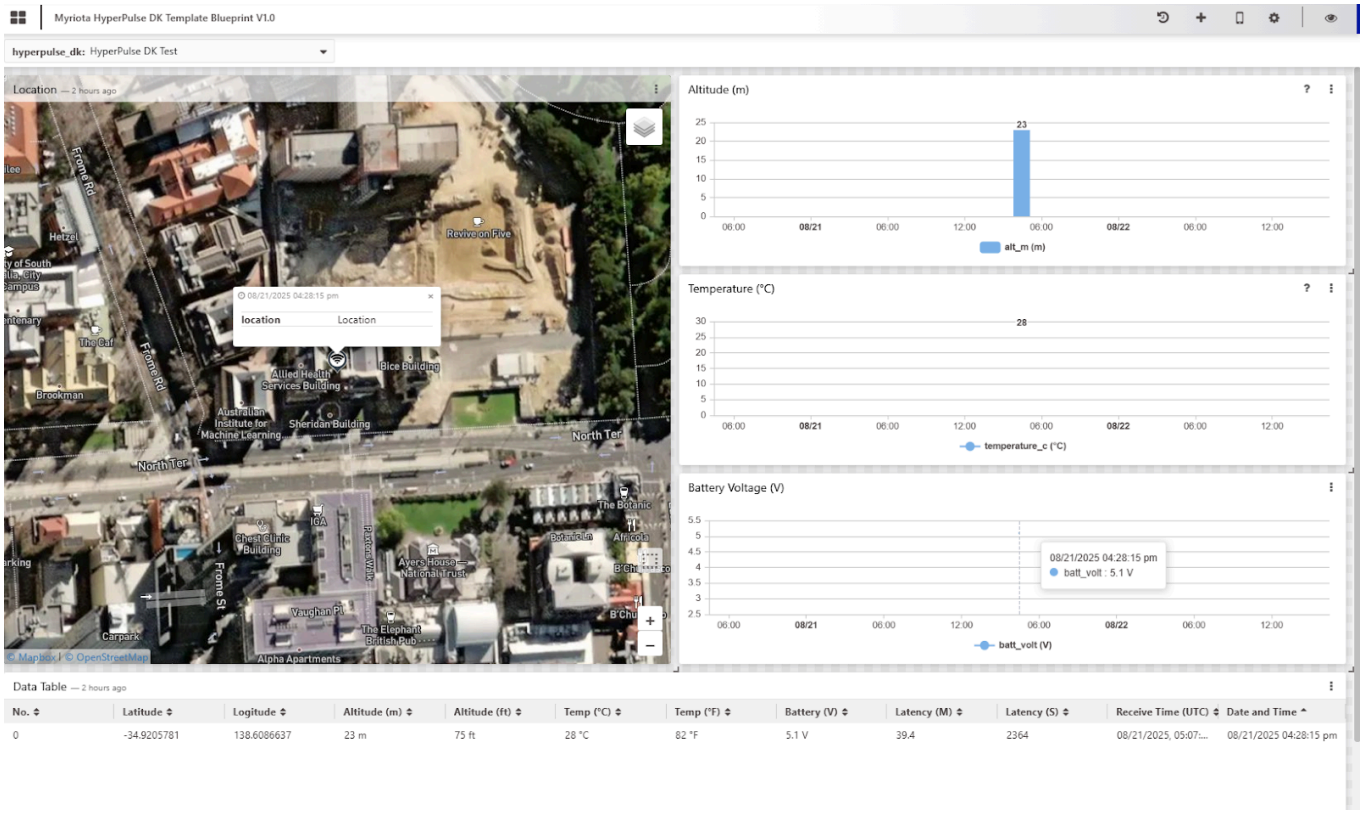
The message format for the default demo firmware can be found in the README file.

Assume the raw message payload is **04000000 20158768 88712FEB 6B099E52 3200 18 3610** (spaces added for clarity). The decoded message content is as follows:

Content	Hex	Decimal	Decoded Value
Sequence Number	00000004	4	The 5th message after reset
Timestamp	68871520	1753683232	GMT: Monday, July 28, 2025 6:13:52 AM
Latitude	EB2F7188	-349212280	-34.9212280
Longitude	529E096B	1386088811	138.6088811
Altitude	0032	50	50 meters
Temperature	18	24	24 ℃
Battery Voltage	1036	4150	4.15 V

Visualising Data on TagoIO

Myriota partners with Tago.io to accelerate your prototyping. TagoIO is an IoT cloud platform that transforms the way businesses create value from connected products and user interactions. It provides tools for your business to manage devices, store data, run analytics, and integrate services, with a high level of customisation and specific functions based on your application.



For details, please refer to the TagoIO Integration Article.

Creating Your Own Visualisation

Data received from Myriota modules can be forwarded to one or more destinations. Data is sent as an HTTP POST request with Content-Type: application/json. All data is accompanied by a reference (EndpointRef), timestamp, a unique identifier (UUID), and a digital signature that can be used to verify that the data originated from Myriota.

Please refer to the Destinations Article for more details.

Warranty

Myriota's Manufacturer's Warranty ("Manufacturer's Warranty")

Myriota offers a Product manufacturer warranty for a period of 12 months from delivery. To the extent permitted by law, this warranty is limited to rectification of manufacturing defects and/or non-compliance with the Product Brief.

Myriota will undertake in-warranty service for Products subject to verification of proof of purchase and purchase date.

To the maximum extent permitted by law, Myriota accepts no liability for any defect or failure not caused by Myriota (including Customer accident, misuse and non-observance of operating instructions), or for Products which have been repaired by a person other than Myriota or used for purposes other than for which they are intended or deterioration due to normal use and exposure, including environmental conditions.