

Myriota

Myriota Modem Application Note

MYRIOTA-TEC-318-1.0

November 2024

©Myriota Pty Ltd

Revision History

Rev	Date	Description of Change
1.0	November 2024	Initial version

Related Documentation

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

How to Contact Us

Technical Support

developer.myriota.com

support@myriota.com

Sales Support

sales@myriota.com

Myriota Online

myriota.com

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.

- Information in this document is provided solely to enable system and software implementers to use Myriota Pty Ltd products.
- 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

Summary	5
Interface Considerations	5
AT-Modem Interface	5
Hardware Considerations	5
GNSS Module	5
Host Communication Interface	6
Modem Busy Signal	7
Hardware Options	7
Myriota Development Board	7
Myriota FlexSense	9
Building Custom Hardware	10
Firmware Options	10
AT Modem Reference Application	10
Pre-built Reference Binaries	11
Device Deployment	12

Summary

The Myriota Module is a powerful edge computing device that includes an onboard processor which can be configured to different applications. This allows the Myriota Module to be used in standalone configurations where it will interface directly to sensors to collect data and then securely transmit the data to the Myriota Network. The Myriota Module can also be used as a modem in solutions where the solution includes other processors to perform the data collection and processing. This document outlines options and consideration when using the Myriota module in these modem configurations.

Interface Considerations

AT-Modem Interface

The AT Command set is a universal instruction standard for all communication modems. Myriota has an [at_modem](#) reference application which supports essential modem commands making it an ideal starting point for developing your own application. Further information on this firmware is outlined below and in the [documentation](#) on github.

Hardware Considerations

There are several hardware configuration options to use the Myriota module as a modem. The main considerations are around the GNSS and Communications interfaces. These considerations are outlined below.

GNSS Module

The Myriota Module relies on time and location to predict satellite passes. Inaccurate time and location can lead to satellite communication issues and a sub-optimal user experience.

The Myriota Module SDK has support for interfacing directly to GNSS modules. This is typically the easiest and most reliable method as it will reduce latency and firmware complexity. The Myriota SDK currently utilizes uBlox's proprietary UBX protocol and supports any modules from the uBlox M8 family.

Alternatively, the time and location can be set via an external processor using AT-commands. Careful consideration around the accuracy of the time data (factoring potential latency in communications). The Myriota module requires its internal time to be accurate within +/- 1 second to reliably connect to the network.

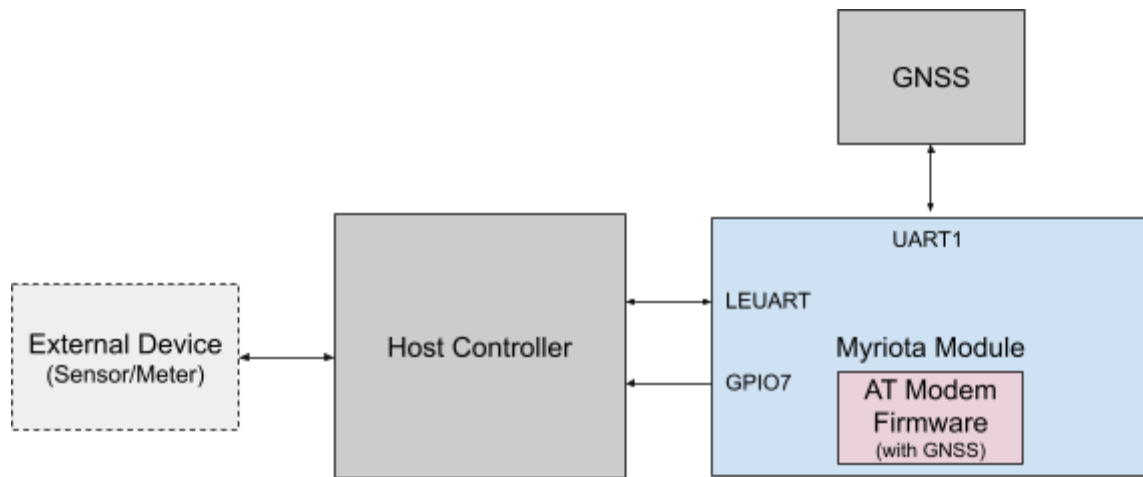


Figure 1: Myriota Module with direct GNSS module interface

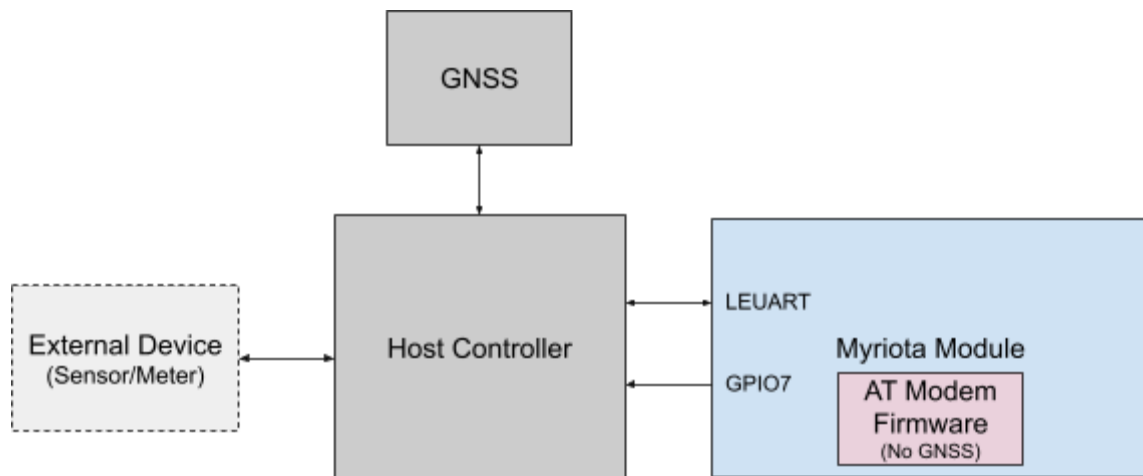


Figure 2: Myriota Module with external GNSS module

Host Communication Interface

The Myriota module includes several UART interfaces. It is recommended to use LEUART (default) as the modem communication port as it supports waking the module from sleep mode, eliminating the need for an additional wake-up pin to be connected to the host. The LEUART interface is restricted to 9600 baud rate and defaults to 8 data bits, no parity, and 1 stop bit (9600, 8, N, 1). Note that I2C and SPI interfaces are not recommended as the SDK does not support slave mode on these interfaces.

UART0 and UART1 can also be used, but they do not support waking up the module directly. An additional GPIO is required to enable the wakeup function. Please note that UART0 serves as the debug port and UART1 is also used for GNSS communication.

Modem Busy Signal

There may be times when the Myriota module is not able to accept new commands (e.g. when it is performing a GNSS fix). The host controller should determine if the Myriota module is ready to accept commands before sending data to the module. The module will be able to buffer some data during this time for later processing, however sending excessive data may result in data loss.

In the [at_modem_application](#), GPIO7 acts as the modem busy indicator. It outputs a high signal when the module is busy and a low signal when it is idle. This indicator helps prevent data loss by allowing the host controller to wait until the module is ready to receive commands. The “[AT+STATE=?](#)” command can also be used to determine the readiness of the module.

Hardware Options

There are several hardware options for developing Prototyping and Production solutions:

1. Use the Myriota Development Board
2. Use the Myriota FlexSense
3. Design custom hardware using the Myriota Module

Myriota Development Board

The Myriota Development Board, also known as the Devboard or Devkit, is an excellent platform for firmware development and evaluating modem applications. It includes an RF circuitry and antenna required to communicate to the satellite as well as a built-in GNSS module. The I/O expansion header allows for easy access to the module interfaces (UART & GPIO) to connect to other hardware and evaluate modem solutions.

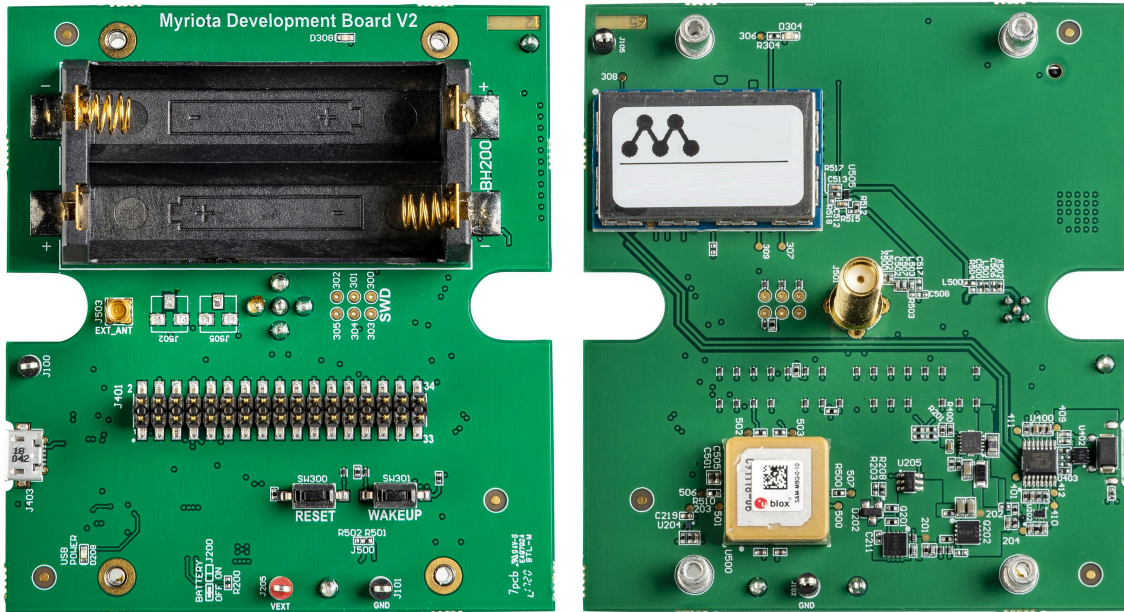
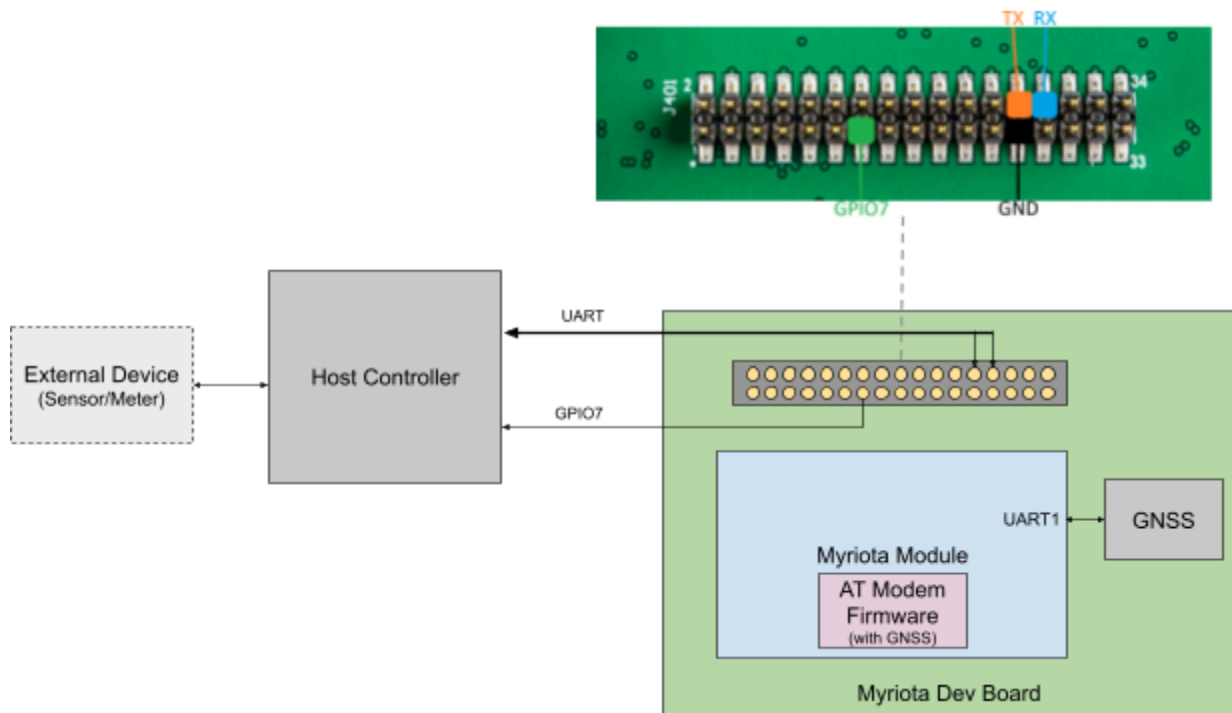


Figure 3: Myriota Development Board

The table below captures the relevant Myriota Development Board I/O expansion header pins used for the AT modem firmware.

Name	Pin	Description
GND	Multiple	
LEUART_TX	26	Default modem communication port. Capable of waking up the module from sleep mode. LEUART is operated at 3.3V TTL, configured to 9600 baud rate, 8 data bits, no parity, and 1 stop bit (9600, 8, N, 1). UART0 and UART1 are also available for use, but they require an additional GPIO to wake up the module before communication, as well as extra firmware processing.
LEUART_RX	28	
GPIO7	13	Modem busy signal. Outputs high when a job is running, low when idle

Below is an overview of how the I/O expansion header pins would be connected to an external host controller.



Myriota FlexSense

The Myriota FlexSense is an integrated, ready to deploy, hardware solution. It supports interfacing to a wide range of different sensors using built-in RS232/RS485, analog & digital I/O. The FlexSense is primarily designed to be a master device that interfaces to external devices and sensors. FlexSense can be used for modem type applications with the following considerations:

- The FlexSense supports both RS232 & RS485 serial interfaces. UART (TTL) is not supported.
- The FlexSense requires a wake-up event before it can receive data and/or commands. This is to ensure that the FlexSense device is not in sleep mode and the RS232/RS485 transceivers are active. This wake-up event can be supplied via a GPIO interface.

The [rs485 rs232 example](#) in the Myriota FlexSDK can be used as a starting point. Please note that the [at modem application](#) in the standard Myriota SDK is not supported by FlexSense.

If you are considering using the Myriota FlexSense as a modem, we strongly recommend reaching out to our support team to discuss this further.

Building Custom Hardware

Designing your own hardware solution using the Myriota Module is the most cost effective solution for large-scale applications. Myriota provides several hardware design guides on the support site to assist in design activities. The table below captures the key Module interface pins which are required to implement a modem solution. The Developer Board schematics can also be used as an example. Please reach out to the support team if you have any questions.

Name	Module Pins	Description
VEXT	4	3.0-3.6V power supply to the module
GND	Multiple	
NRST	13	Optional. Reset input with built-in 40kOhm pull-up, active low
UART0_TX	55	Programming and debugging port
UART0_RX	56	
UART1_TX	49	GNSS communication port. Connect to a U-Blox M8 GNSS if using the integrated Myriota SDK support.
UART1_RX	48	
LEUART_TX	58	Default modem communication port. Capable of waking up the module from deep sleep. Restricted to 9600 Baud.
LEUART_RX	57	
RF_PORT	27	RF signal output and input to antenna.
GPIO7	53	Modem busy signal. Outputs high when a job is running while low for idle

Firmware Options

AT Modem Reference Application

The [at_modem](#) reference application is tailored for slave modem applications and includes essential modem operations, making it an ideal starting point for developing your own application. It is compatible with Myriota Development Board and Myriota Module. It is not compatible with FlexSense.

The example operates on LEUART and uses GPIO7 as the modem busy signal indicator. It features two sets of commands: Queries and Controls. Queries are used to retrieve information

from the module, such as state, SDK version, time, and location. Controls include commands such as message scheduling, RF tests, and starting a GNSS fix.

Please note:

- The Myriota Module can become unresponsive during GNSS fixes; this is a normal behavior for the Myriota Module. Monitor the Modem Busy interface to determine when to send data/commands.
- Setting time and location commands are only required when Myriota Module does not interface directly to a GNSS module.
- Reference applications are supplied for evaluation and test purposes only. Customers should develop and test their own firmware to suit their production environments.

You can find detailed information in the [README.md](#).

Pre-built Reference Binaries

Prebuilt AT-modem reference binaries are available to download from [Myriota Device Manager - Binaries](#). There are two options for AT modem:

1. **AT modem (with GNSS fix): `at_modem.bin`** is used when the Myriota module directly interfaces to a GNSS module. On start-up (and at regular intervals), the Myriota module will perform a GNSS fix to obtain time & location. The unit running the application should be deployed outdoors with a clear sky view to ensure the fix is successful.
2. **AT modem (skip GNSS fix): `at_modem_skip_gnssfix.bin`** skips the GNSS fix process and the time and location will be set to initial values. The host will need to feed the accurate time and physical location into the module using the time and location control commands. This application can also be used for AT commands tests indoors when GNSS fixes can not be obtained.

Please note:

- **`system_image.bin`** needs to be programmed into the module corresponding to the SDK version used. After programming this file, the application is erased and will need to be reprogrammed.
- Programming the latest Network Information prior to deployment will allow the module to take advantage of the constellation straightaway. You can use Myriota [DeviceAssist](#) to download and program the latest Network Information. Refer to this article: [NetworkInfo: Warm Start vs Cold Start](#) for more detailed information.

Myriota DeviceAssist application provides a simple way to re-program your device. The latest version for Windows and Linux can be downloaded from [Myriota Device Assist App](#).

Device Deployment

For deployment tests in the field, it's recommended to use lithium AA batteries. USB power or an external power supply can introduce unwanted RF noise, potentially disrupting satellite communications.

You can find further information regarding device deployments on our support site, please refer to [Device Deployment Guide](#).

If you require any further assistance please contact our support team via support@myriota.com.