UM2712

User manual

Getting started with the FP-ATR-LTE1 software package for asset tracking with LTE connectivity, GNSS and MEMS sensors

Introduction

The FP-ATR-LTE1is an STM32Cube function pack that allows you to connect an IoT node to Amazon Web Services using LTE cellular connectivity and view GNSS and MEMS sensor data using standard application layer protocols.

It fully supports security and protocol requirements to interface with AWS cloud thanks to the integrated Amazon AWS IoT SDK.

This software, together with the suggested combination of STM32 and ST devices, can be used to develop asset tracking and other related applications. ST also provides an asset tracking dashboard for sensor data visualization.

The package contains examples that implements sensor data reading from a local or a remote device on the Internet.

The software runs on the STM32 microcontroller and includes drivers for the Teseo-LIV3F-based GNSS module, and for the MEMS motion and environmental sensors.

RELATED LINKS

Visit the STM32Cube ecosystem web page on www.st.com for further information

1 Acronyms and abbreviations

Table 1. List of acronyms

Acronym	Description
AP	Access point
BSP	Base support package
FOTA	Firmware update over-the-air
GNSS	Global navigation satellite system
GPIO	General purpose input/output
GPS	Global positioning system
HAL	Hardware abstraction layer
HTML	Hypertext markup language
HTTP	Hypertext transfer protocol
IDE	Integrated development environment
IoT	Internet of things
I2C	Inter-integrated circuit
LTE	Long-term evolution
MCU	Microcontroller unit
MEMS	Micro electro-mechanical systems
ODE	Open development environment
REST API	Representational state transfer API
SIM	Subscriber identity module
SDK	Software development kit
SMD	Surface mount device
SSID	Service set identifier
UART	Universal asynchronous receiver/transmitter
URL	Uniform resource locator

2 FP-ATR-LTE1 software expansion for STM32Cube

2.1 Overview

The package features:

- Complete firmware to connect a node with GNSS and MEMS sensors to a cellular network using LTE communication technology
- Middleware library with FreeRTOS, mbedTLS and Amazon Web Services SDK
- Software interface to access GNSS module (Teseo-LIV3F), temperature and humidity sensor (HTS221), pressure sensor (LPS22HB), and motion sensors (LSM6DSO, LIS2MDL and LIS2DW12)
- Sample implementations available for X-NUCLEO-GNSS1A1 and X-NUCLEO-IKS01A3 expansion boards when connected to a P-L496G-CELL02 LTE cellular to cloud pack
- Easy portability across different MCU families, thanks to STM32Cube
- Free, user-friendly license terms

The package integrates the AWS IoT device SDK middleware with APIs to simplify interaction between the 32L496GDISCOVERY and the Amazon AWS IoT services.

You can use it to prototype end-to-end sensors-to-cloud IoT applications, by registering your board to the ST Asset Tracking web dashboard and begin exchanging real-time sensor data and commands.

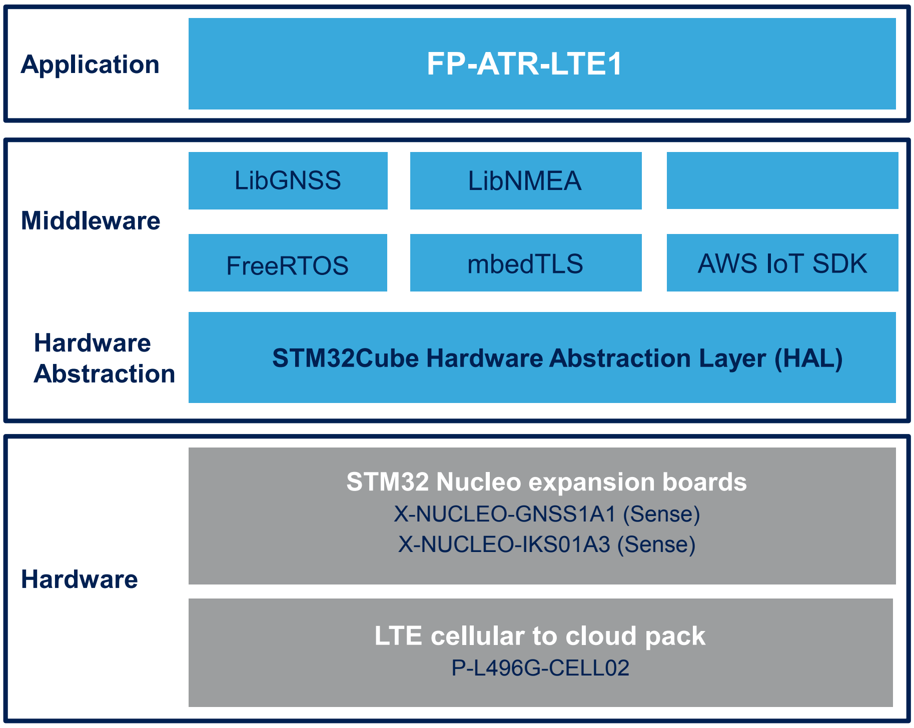
For AWS license terms, visit https://aws.amazon.com.

2.2 Architecture

The software layers used by the application software to access and use the STM32 microcontroller, cellular (LTE-IoT) modem and the sensor expansion boards are:

- **STM32Cube HAL layer**: a simple, generic and multi-instance set of APIs (application programming interfaces) to interact with the upper layer applications, libraries and stack layers. The APIs are based on a common framework so that overlying software can implement functions and routines without specific microcontroller unit (MCU) hardware configurations. This structure improves library code reusability and guarantees easy portability across other devices.
- **Board support package (BSP) layer**: drives the STM32 Nucleo board peripherals such as LEDs, user button, etc. (except the MCU), with a specific set of APIs. This interface also helps to identify the specific board version.
- **Middleware layers**: contains mbedTLS and the Amazon AWS IoT device SDK (https://aws.amazon.com/tools/) to facilitate the connection of the 32L496GDISCOVERY discovery kit to AWS IoT services.

Figure 1. **FP-ATR-LTE1 software architecture**



2.3 Folder structure

The folders in the software package are:

- **Documentation**: a compiled HTML file generated from the source code detailing the software components and APIs (one for each project).
- **Drivers**: HAL drivers and the board-specific drivers for each supported board or hardware platform, including those for the on-board components, and the CMSIS vendor-independent hardware abstraction layer for the ARM Cortex-M processor series.
- **Middlewares**: middleware interface for cellular and mbedTLS, together with the porting of Amazon AWS IoT device SDK.
- **Utilities**: Tera Term script parameters
- **Projects**: sample applications for 32L496GDISCOVERY that can be used to register a device and transmit sensor data to Amazon AWS IoT. Together with the ST Asset Tracking web dashboard, these sample applications enable easy sensor data visualization and device control through AWS IoT. Sample applications can be compiled under the IAR Embedded Workbench for ARM, RealView Microcontroller Development Kit or STM32CubeIDE development environments. For each sample application, the folder also contains a corresponding pre-compiled binary.

3 How to run the firmware application

Before you begin, you should set up the hardware configuration compatible with the FP-ATR-LTE1 function pack.

To set up your remote IoT application with cloud monitoring and control capability, follow the steps below.

Step 1. Configure and launch a serial terminal application.

Step 2. Download the FP-ATR-LTE1 firmware package.

Step 3. Recompile the code or flash pre-compiled binaries to your board.

Step 4. Complete the cellular setup and read the SIM ID from serial terminal.

Step 5. Open the web dashboard and register your board using the SIM ID as Device ID.

Step 6. Download the certificate and key file from the dashboard.

Step 7. Provision credentials and certificates to your board.

Step 8. Login to the dashboard.

Step 9. Visualize sensor data.

3.1 Serial terminal setup

To update the provisioning data (cellular and certificates) to the board, you have to set up a serial terminal on a PC.

A serial terminal is required to:

- configure the board (cellular parameters, device credentials)
- read SIM ID (used as device ID to register the device in the web dashboard)
- display the published and received AWS IoT messages locally

Tera Term is used in the example below, but you can use any other similar tool.

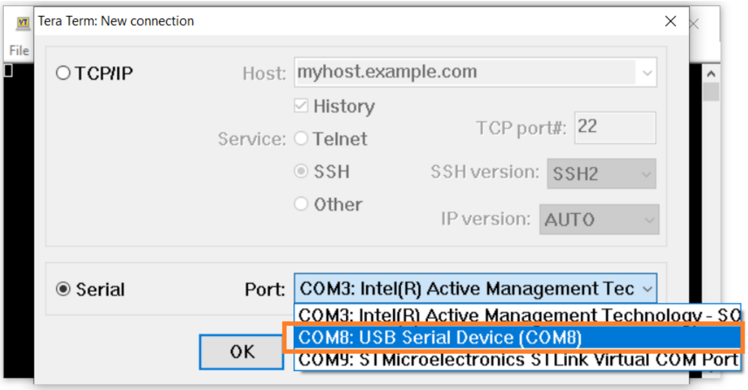
Step 1. When using the board for the first time, connect it to your PC USB port.

Step 2. Determine the STM32 ST-LINK Virtual COM port used for the STM32 Nucleo board.

Step 3. Open the [**Device Manager**].

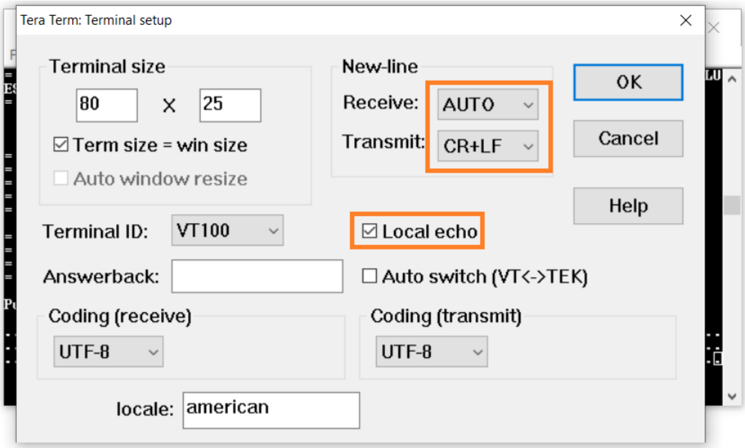
Step 4. Open a serial terminal on the PC and connect it to the above virtual COM port.

Figure 2. Opening the COM port for the STM32 Nucleo board



Step 5. Configure the terminal with the recommended parameters shown below.

Figure 3. Terminal setup



Note: The virtual terminal new-line transmit configuration must be set to LineFeed (\n or LF) to allow copying and pasting from the UNIX type text files. The “Local echo” option makes copy and paste visible on the console.

Step 6. Flash the board with the pre-compiled binaries provided in the firmware package.FP-ATR-LTE1

Step 7. Reset the board and follow the indications on the serial terminal to configure cellular parameters and device credentials (see the following sections).

Data entered via serial terminal is copied and stored in the Flash memory and can be reused without editing after each board reset.

3.2 Configuration of cellular credentials

The P-L496G-CELL02 Discovery Kit features ready-to-run firmware samples using the 2G, 3G, LTE Cat M1, or NB-IoT protocols to support quick evaluation and development of AWS IoT cloud applications.

The FP-ATR-LTE1 package offers out-of-the-box connectivity for communication to the AWS IoT through the MQTT protocol. It implements a complete middleware and application-level stack in C language, which allows the connection of the P-L496G-CELL02 Discovery kit to a AWS IoT.

The Discovery Kit includes an STM32-based main board and a cellular add-on modem board.

To use the BG96 modem in Cat M1 or NB1 mode, you must insert a SIM card compliant with this network technology from an MNO that has already deployed it.

After board reset, you can initialize the cellular module.

Figure 4. **Cellular module initialization and modem connection to the network**

```
*************************************************************
***   STM32 IoT Discovery kit for
***      STM32L475/STM32L496 MCU
***   FP-ATR-LTE1 Cloud Connectivity Demonstration
***   FW version 1.0.0 - 18-March-2020 11:30:00 AM
*************************************************************
Initializing sensors... OK.

*** Board personalization ***

- Network Interface initialized:
- Network Interface starting:

*** C2C connection ***


Initializing the cellular module

- Network Interface connecting:
Trying to connect with the external SIM

Waiting for BG96 modem running
Signal not known or not detectable yet (be patient)
```

Then, you have to connect to the dashboard and register the new device (Thing).

Figure 5. **Dashboard link and Thing registration**

```
- To register your device and display data go to ST Dashboard at https://dsh-assetracking.st.com/
  - First 16 Bytes of SIM Id to use for Board registration: 8939010000212391
```

The first sixteen bytes of the platform SIM ID are used as default Thing Name/Device ID in device registration. When connection to MNO equipment is established, the board SIM ID (that is, the sixteen bytes) is shown on the serial terminal.

3.2.1 How to register the board to the ST IoT web dashboard

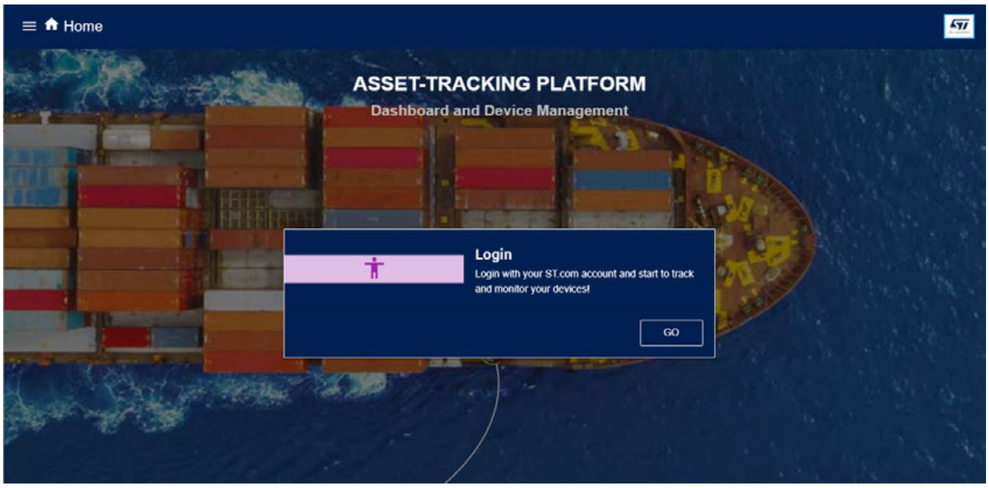
To run the test application included in FP-ATR-LTE1 package and connect it to the ST Asset Tracking web dashboard, it is necessary to register the board and download the device certificate and key.

The communication between ST development boards and AWS IoT is protected by X.509 device certificate, a digital certificate that uses public key infrastructure (PKI) standard.

The dashboard automates the generation of these certificates in the ST AWS Account used to deploy and run the web application.

Step 1. To register a board and generate certificates, open the ST Asset Tracking web dashboard

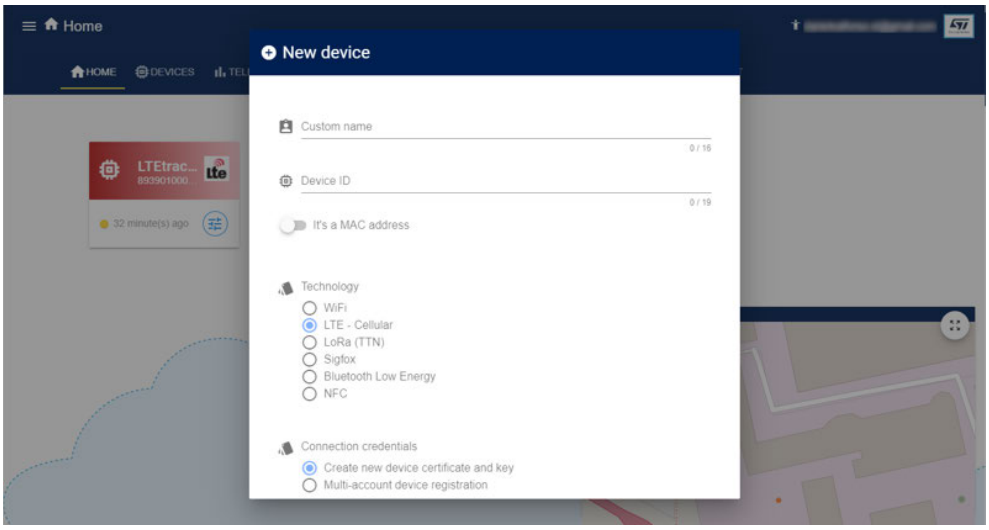
Figure 6. ST Asset Tracking web dashboard



Step 2. Click on [**Register and configure your devices**] and complete the login procedure.

Step 3. After successful login, in the Dashboard view, click on the empty icon with the '+' sign to register your device.
The [**New Device**] popup appears.

Figure 7. Enroll a new device



Step 4. Select [**LTE Cellular**], enter the SIM ID as [**Device ID**] and a descriptive name for the device in the [**Custom Name**], and click on [**Submit**].

Step 5. After successful device registration, a .pem file will be automatically downloaded by the web browser.

Step 6. You can check the file in a text editor.

The text file named *xxxx.pem* (where xxx is the device ID) is divided into two sections:

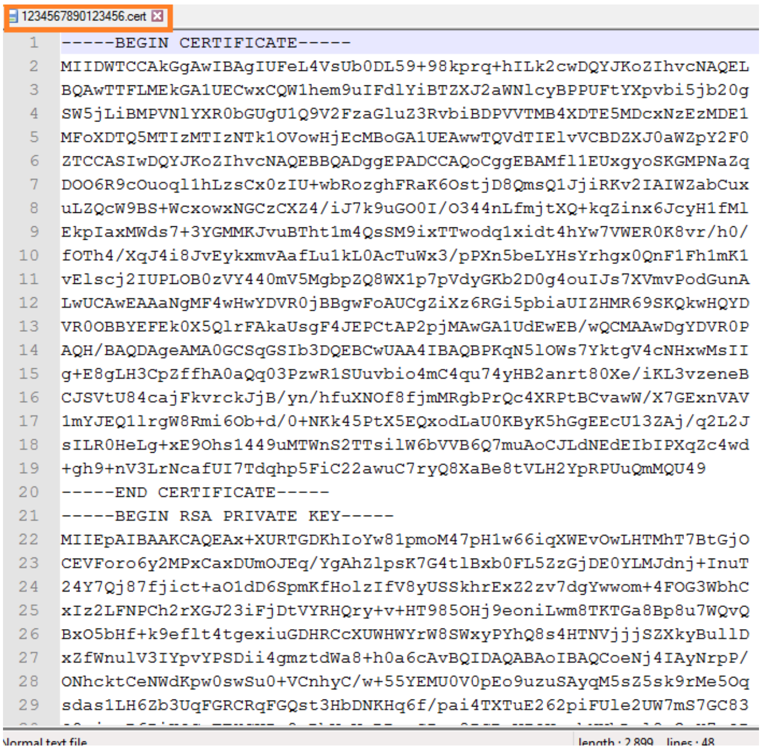
a. The first section is the certificate of the AWS Thing created in AWS IoT core:

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

b. The second section is the private key of the AWS Thing created in AWS IoT:

```
-----BEGIN RSA PRIVATE KEY-----
...
-----END RSA PRIVATE KEY-----
```

Figure 8. Content of downloaded device_id.pem file



3.3 Board security credential provisioning

To set up your board with security credentials downloaded from ST IoT web dashboard after board registration, you can enter or update credentials as follows:

- if the credentials are not available in the Flash memory (blank board), the console prompts to enter device certificates and key at startup;
- if the credentials are available in the Flash (pre-provisioned board), the console prompts to press the user blue button within 5 seconds after the board startup to update them.

3.3.1 Blank board

When the board is flashed with pre-compiled files, enter or update the following parameters:

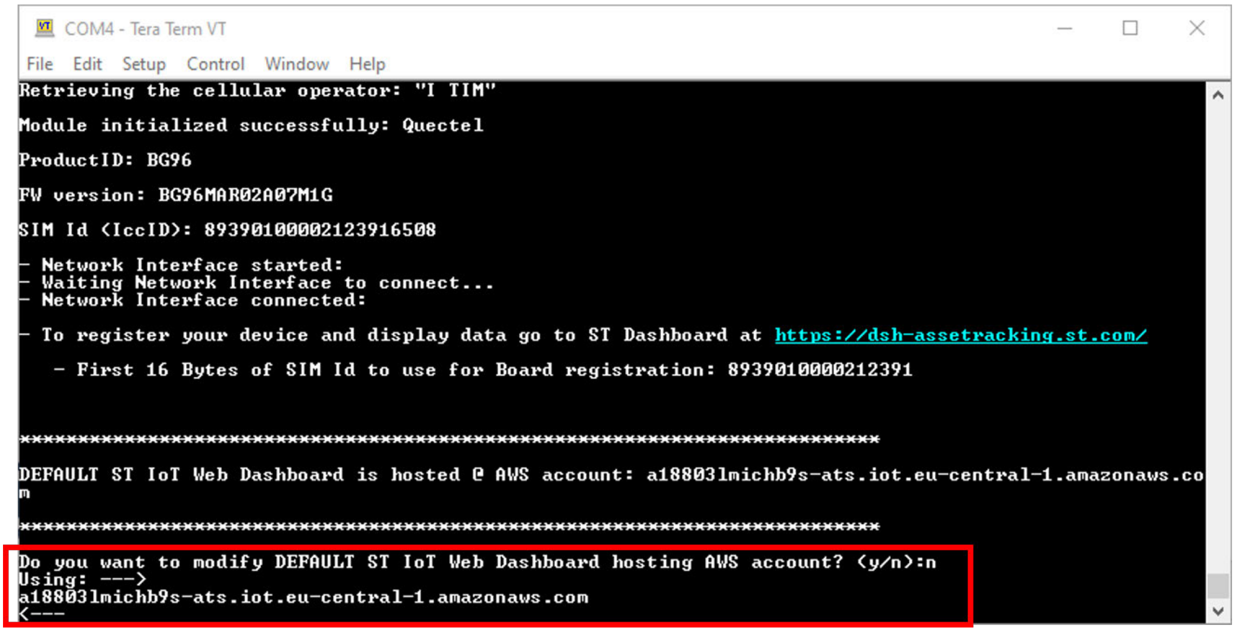
- AWS IoT endpoint
- Thing Name
- AWS Root Certificate
- Thing Device Certificate

- Thing Private Key

Note: *The firmware contains default values for the Root Certificate and AWS IoT endpoint related to the ST account used to deploy web dashboard and must not be modified. By default, the device ID used to register the Thing to the AWS IoT core is the SIM ID (first sixteen bytes).*
You must enter Private Key and Device Certificate copied and pasted from the device_id.pem file downloaded from the web page (as described in Section 3.3 Board security credential provisioning).

Step 1. When prompted to change Default ST IoT Dashboard hosting AWS account, press ‘n’ and [**ENTER**] to proceed.
The default endpoint must not be changed when using the firmware with the web dashboard.

Figure 9. Default ST IoT Dashboard hosting AWS account

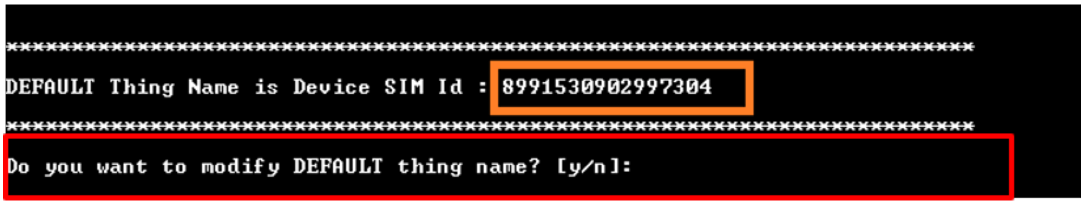


The firmware reads the SIM ID as default device ID, the same used to register the device and create certificates through the web dashboard.

Figure 10. Default Thing name

Step 2. Press ‘n’ and [**ENTER**] when asked to update default Thing name.

Figure 11. Default Thing name change



Step 3. When prompted to change Default Root Certificate, press ‘n’ and [**ENTER**] to proceed.

Figure 12. Default Root CA certificate

```
Enter the x509 certificates or keys as per the following format:
-----BEGIN CERTIFICATE-----
YMPGn8u67GB9t+aEMr5P+1gmIgNb1LTV+/Xjli5wwOQuvfwu7uJBVcA0Ln0kcmnL
R7EUQIN9Z/SG9jGr8XmksrUuEvmEF/Bibyc+E1ixVA0hmnM3oTDPb5Lc9un8rNsu
.......
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
YMPGn8u67GB9t+aEMr5P+1gmIgNb1LTV+/Xjli5wwOQuvfwu7uJBVcA0Ln0kcmnL
.......
-----END CERTIFICATE-----

.......
Do you want to change DEFAULT ROOT CA for ST IoT Web Dashboard? [y/n]
```

Step 4. Enter device certificate.

Step 5. Open your device_id.pem file downloaded from the web dashboard in a text editor.

Step 6. Copy the section highlighted below.

Figure 13. Data.cert file: device certificate and private key

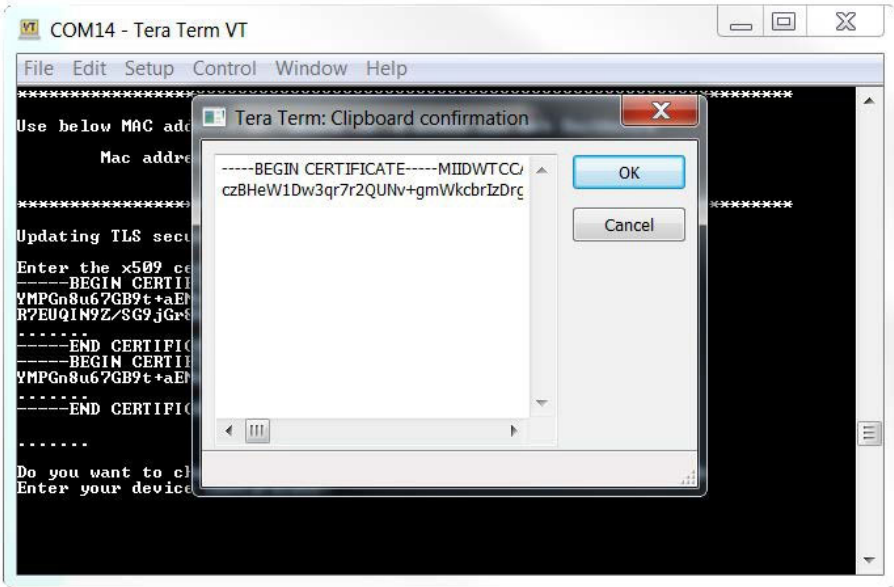
C4_7F_51_03_EE_A2.cert


```
-----BEGIN CERTIFICATE-----
MIIDWjCCAkKgAwIBAgIVAL2XHIBrzqHFzO2/745pM6PqPrMQMA0GCSqGSIb3DQEB
CwUAME0xSzBJBgNVBAsMQkFtYXpvbiBXZWIgU2VydmljZXMgTz1BbWF6b24uY29t
IEluYy4gTD1TZWF0dGxlIFNUPVdhc2hpbmd0b24gQz1VUzAeFw0xOTA3MjQwNjA2
MjFaFw00OTEyMzEyMzU5NTlaMB4xHDAaBgNVBAMME0FXUyBJb1QgQ2VydGlmaWNh
dGUwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQC4kyAMc2uBmsy8homh
e8dFT0aIbLfOJT3WDhp73SJ5mTECY/JYEcTROawjmAVwAR5DiExXN+7jk40GcukO
udyzA8ckh00OYZ7g05WFxMXidvHx391lQYPHZm+zi9jf5jFylhe+o5BqUmIbou/k
6QXg/xSMUkXalJU/iCKORonWPmL5eUA55hLEdumhjemmAp2irneJQkiO7jO9FbUN
yT/HY9b6U/x4al3EoT3uukdwXCU7O0Z/jcpP+OAXyiCrZU26CgbTJYmAmX+oDQxU
WYzF7eHwD0Bk4VX/Z6hI0ti0ygRw8oosQSLKmdwSYJ9ZbMsy1KWP2dtRre4boX7K
IxPBAgMBAAGjYDBeMB8GA1UdIwQYMBaAFAoGYl8+kRouaW4mlCGRzEevUikJMB0G
A1UdDgQWBBRKjuDoYm2FMklPvbcZU7wSmuSVpDAMBgNVHRMBAf8EAjAAMA4GA1Ud
DwEB/wQEAwIHgDANBgkqhkiG9w0BAQsFAAOCAQEAWldJBu2kzv3GybQciCtm5/cF
6cn2qILiq4kmYrhi9ixwO06Kr8wSRxfpeGr8Zt46pFiPrwVODHHqLQYs0qP///vM
JMNLAAftJaxmXnLAvfYudStplfHO17/cDxTJQwaTGY7X3FCf6id6ZQCUuuUMQ3KP
zEaL7oSFdJOoS4R56LTeW6VggfRe+XM3Z/JAwrkc5SR1AQe4uup9qAmuMi3q3QH1
KQbdkmmB9XEoRuv34mUnL5UlOoK1bLx50eL2Mduz2vWkdslu0yFn+nPxYqrJaWBO
7htxWCRh4ZxjYl1ZGBmkEPwyxF8IFV7tw9+9EhM0yOdwfw7cNJdEL6g4cSysZw==
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAuJMgDHNrgZrMvIaJoXvHRU9GiGy3ziU91g4ae90ieZkxAmPy
WBHE0TmsI5gFcAEeQ4hMVzfu45ONBnLpDrncswPHJIdNDmGe4NOVhcTF4nbx8d/d
dUGDx2Zvs4vY3+YxcpYXvqOQalJiG6Lv5OkF4P8UjFJF2pSVP4gijkaJ1j5i+XlA
OeYSxHbpoY3ppgKdoq53iUJIju4zvRW1Dck/x2PW+lP8eGtdxKE97rpHcFwlOztG
f43KT/jgF8ogq2VNugoG0yWJgJl/qA0MVFmMxe3h8A9AZOFV/2eoSNLYtMoEcPKK
LEEiypncEmCfWWzLMtSlj9nbUa3uG6F+yiMTwQIDAQABAoIBAQCoWkH0OV7Iw2qV
Vt24dk9HL8z06L0qgZuKYCjV69jjMtCaDpHgtKxh7onNmX70L4DsNijzsOfrmg0a
4bskHHFqszp0Xhjx0ib5iOQsNFQOgVGjvf6gUhahzRR7n0Y8yxk1HCQiVcyGvggQ
jyjjEYy0MdYLO4vNmJMddM2YGkzlJUsQOEjtBXYlmWKZsZeJOvrTy3TszUElkdcP
Ca620SyEvgPY67Lh76zjsYk+j0+PHTwGnJIv3ZPIJjwsdcrPT87/Yh2S9cGtHgqD
Hk9/o1XbrepbP4LvqNqzn+B6NvNvHuxf/18c5kv8P8QAJ3++1jSjYIEVaOfztsCN
zbYQ5UtVAoGBAOdSlnjTn/75R2m5iO9vw1wpoDsPGVC2bK4DcFGU7fVKJ1V/E/u+
OtjA85befAU7pgq3DJyr0tcNn0nlv3wFhjNg6ErAH7XCOXMogUJmhUSku4/RE+T2
VtCVYfPdccvj/fC11fJ7Ex5WXX19SW+j3pVHzfONp1Io7ExdyZdkBj17AoGBAMxD
2pk6SAijJlEhsTI7rws1j5WVc8FP4yZUxnJnQFqumFoRsMf/Wur8Q1Kfkxmx87zo
hUm+1RAKEFpnQdGdV5v1sajd2UoxH7rz70DgxJ9cjvUK6yR7MnCQgeQEBGF4GVvD
qSr6QYm0TNRCduyH+ecYOVOMShNZ6tvtOYjWOqjzAoGBANdFD00gPAvvddrY06tn
SdnxzvBHK9rl+ZuzDYqOWyhjOtmgqjsjm+XtIoMrRyUjjnD4Sl9es5Z4Ud1WVm2X
Qn172JD/3EKsngQiFLBzceAgoX/EJYx1BvOToRVt5iHFaaPhxFmk2XTpEJIn2vhu
```


Step 7. Go to the terminal application.

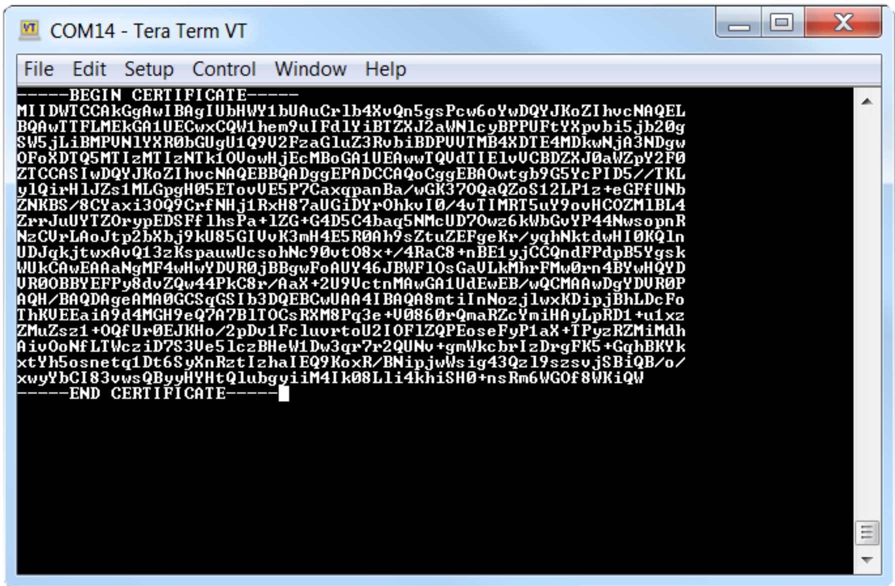
Step 8. Right-click on the terminal.
A clipboard window pops up with the copied certificate data.

Figure 14. Clipboard confirmation



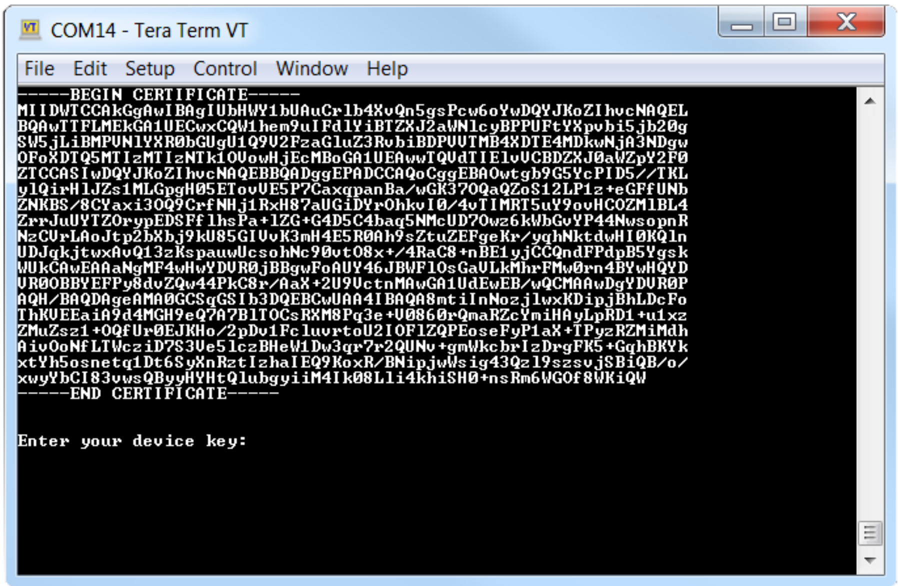
Step 9. Click [**OK**] to copy the data to the terminal.

Figure 15. Device certificate copied to the terminal



Step 10. Press enter twice to send data to the device.

Figure 16. Device certificate accepted



Step 11. Open device_id.pem file and copy the device Private Key.

Step 12. Follow the same procedure used for the device certificate to paste the Private Key in the serial terminal and send it to the device.

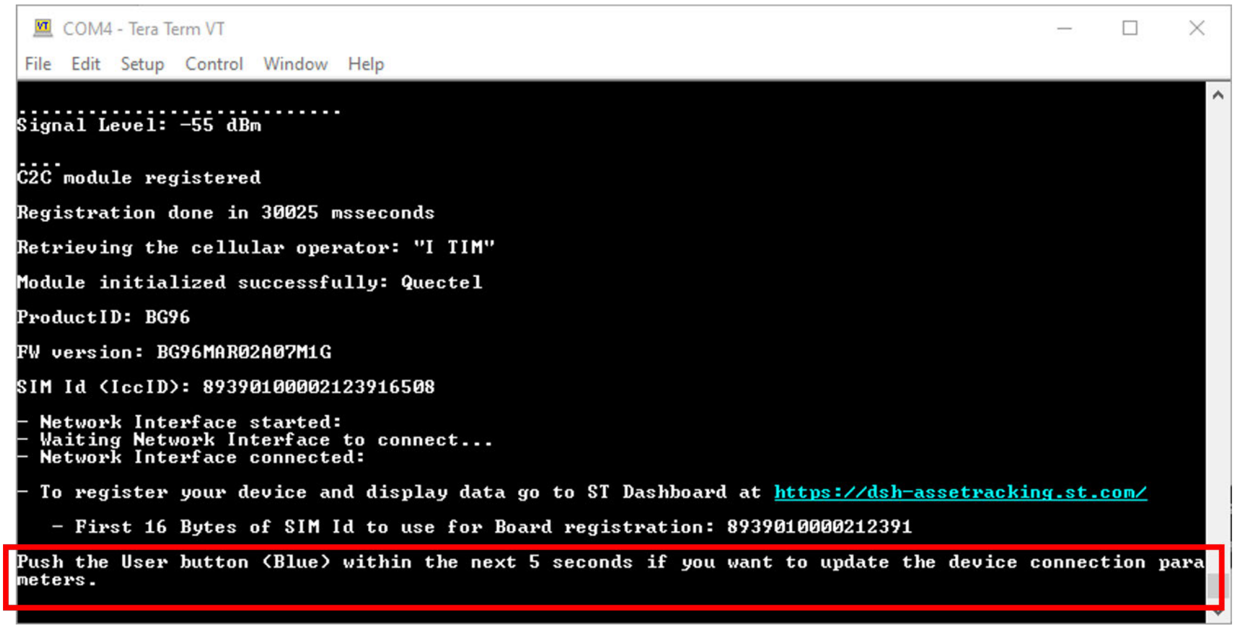
3.3.2 Provisioned board

The entered device certificate and private key and the confirmed default values for Root Certificate, AWS endpoint and Thing name are copied in the Flash memory.

When the board restarts, configuration data are read from the Flash memory.

You can update the pre-provisioned data by pressing the user button within 5 seconds, otherwise the Flash configuration data are used.

Figure 17. Device security parameter update



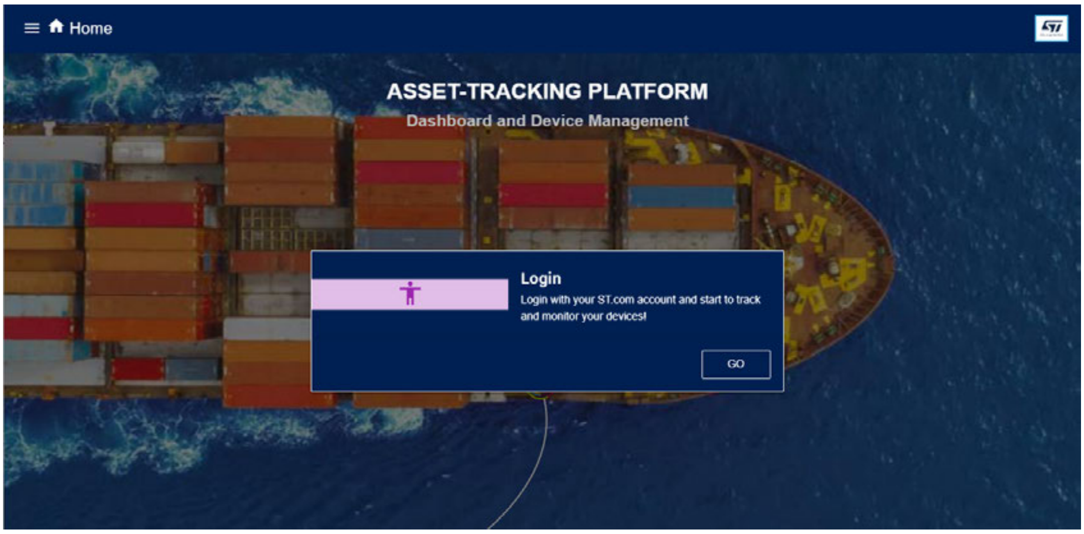
3.4 Data visualization in the web dashboard

Once the device configuration is complete, the board sends sensor data to the AWS IoT Cloud.

To visualize sensor data, follow the procedure below.

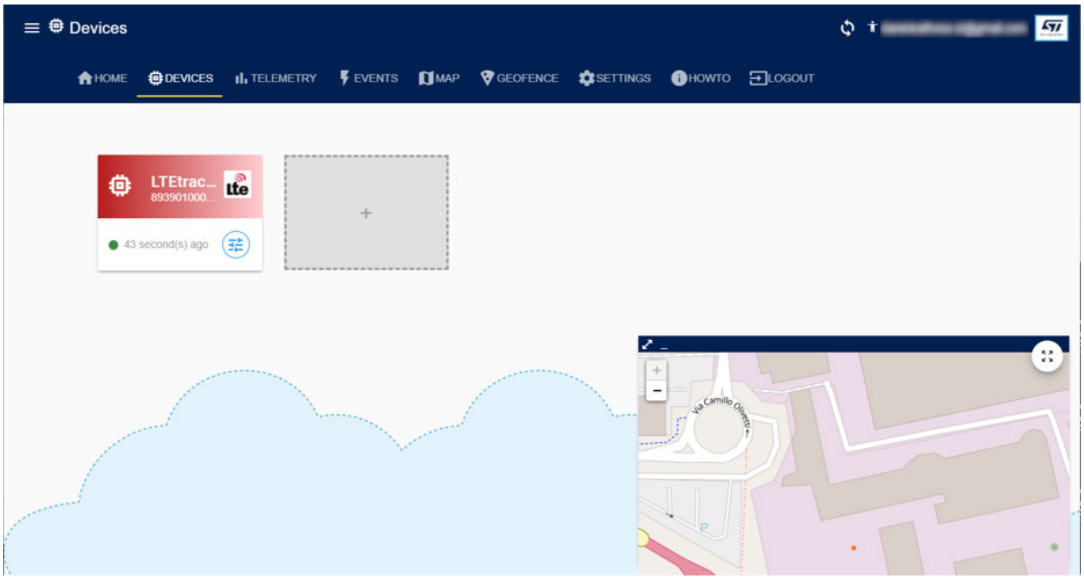
Step 1. Log in to the Web Dashboard.

Figure 18. Dashboard login



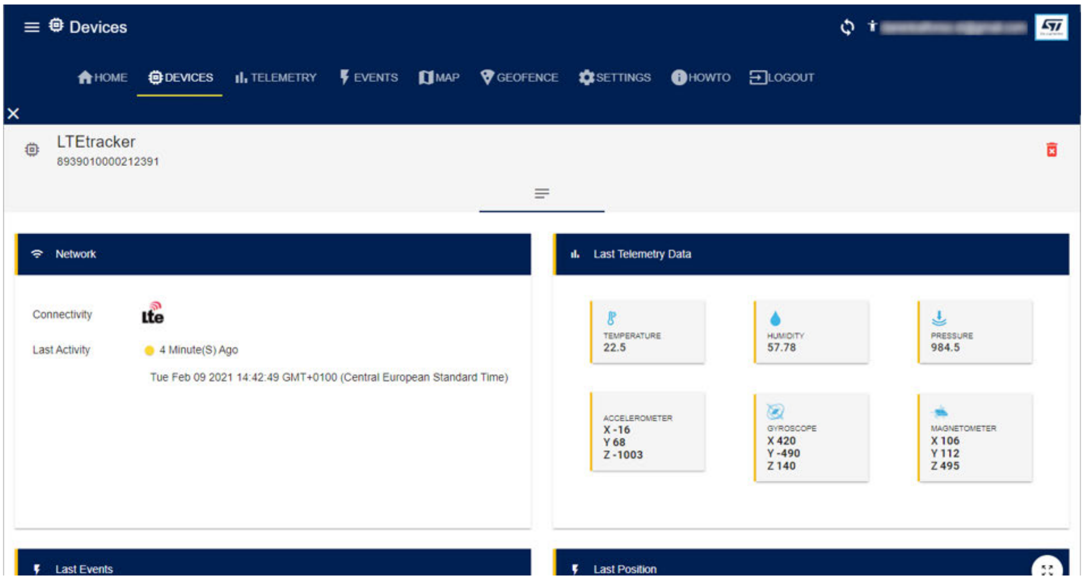
Step 2. After logging in, the list of available devices appears.

Figure 19. Dashboard menu



Step 3. Select the device you want to monitor.
The dashboard shows the last activity of the device.

Figure 20. Dashboard selected device



Step 4. Select the device, the desired telemetry data to display and the window time, then press the blue arrow.

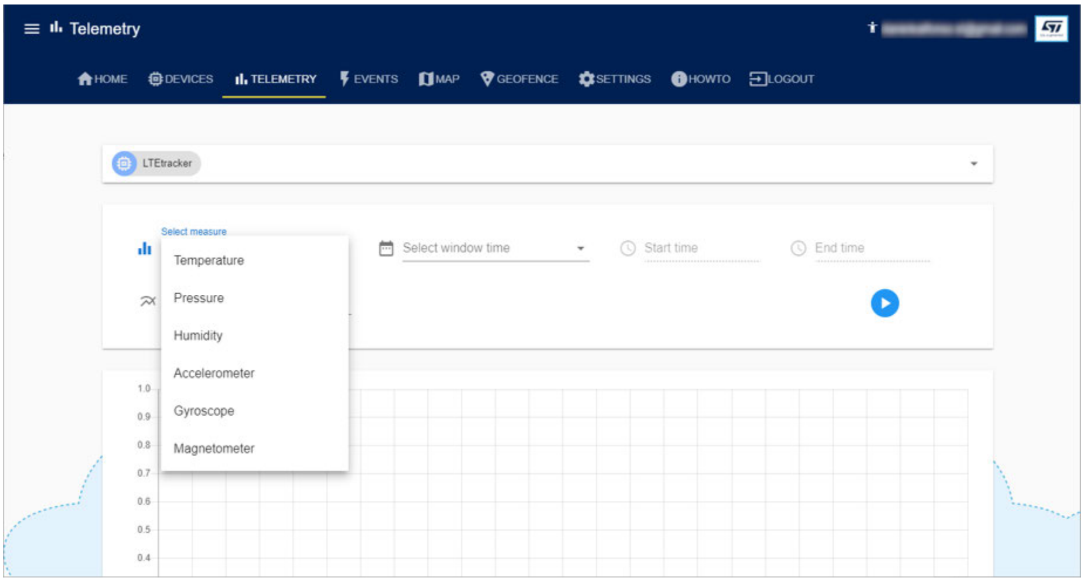


Figure 21. Telemetry data

You can see sensor data related to environmental sensors (temperature, pressure, humidity) and motion sensors (accelerometer, magnetometer, gyroscope) charts and geo tracking information on a map.

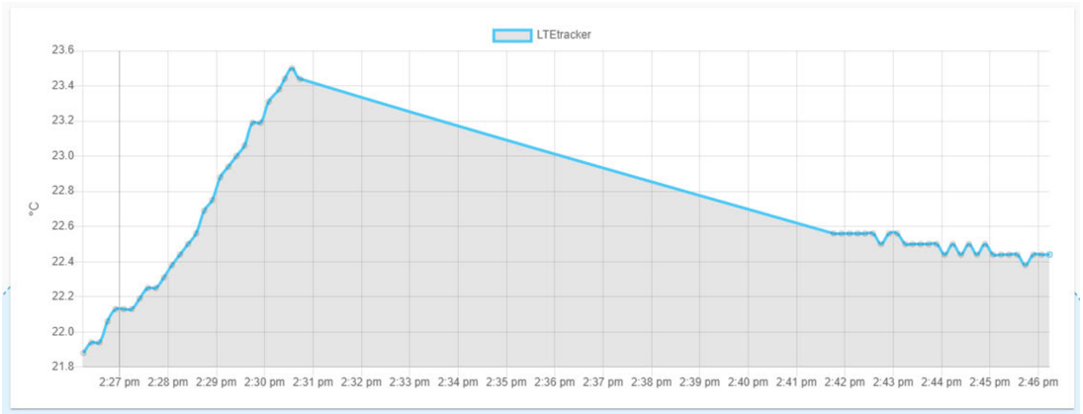


Figure 22. Temperature

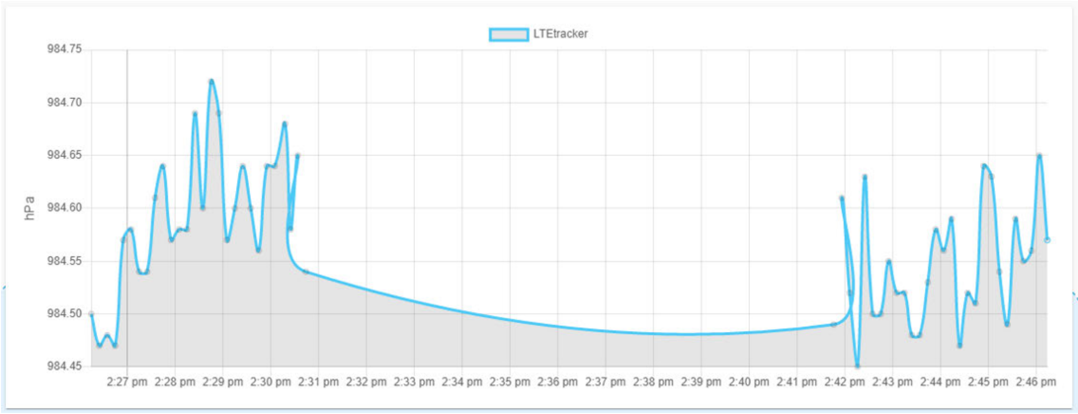


Figure 23. Humidity

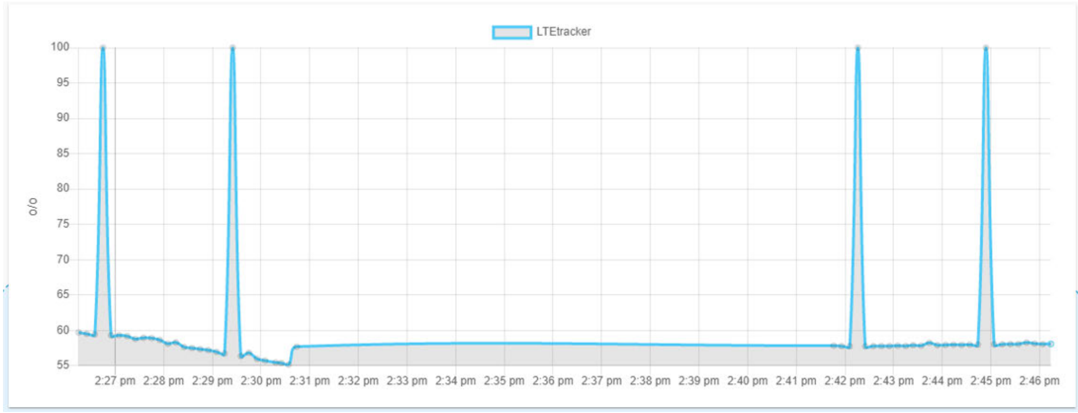


Figure 24. Pressure

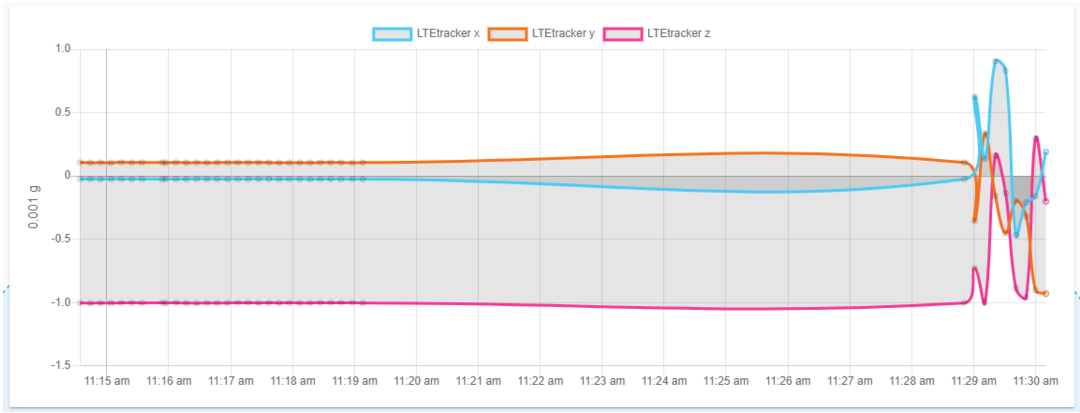


Figure 25. Accelerometer

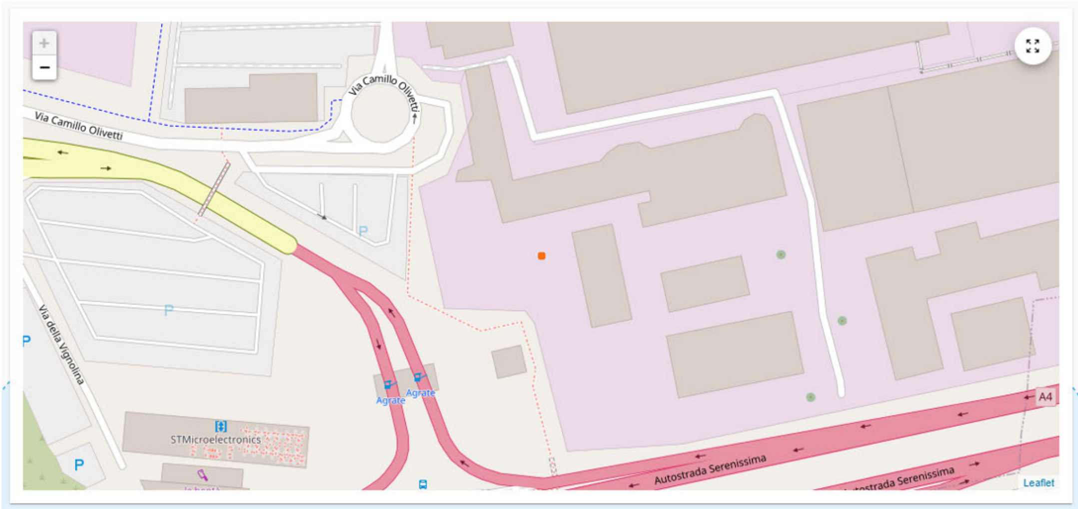


Figure 26. **Geo tracking**

3.5 Delete a registered board

You can remove a registered board and the corresponding Thing created in the AWS IoT Core directly from the web dashboard by simply clicking on the device icon and then on the trash icon to delete the device.

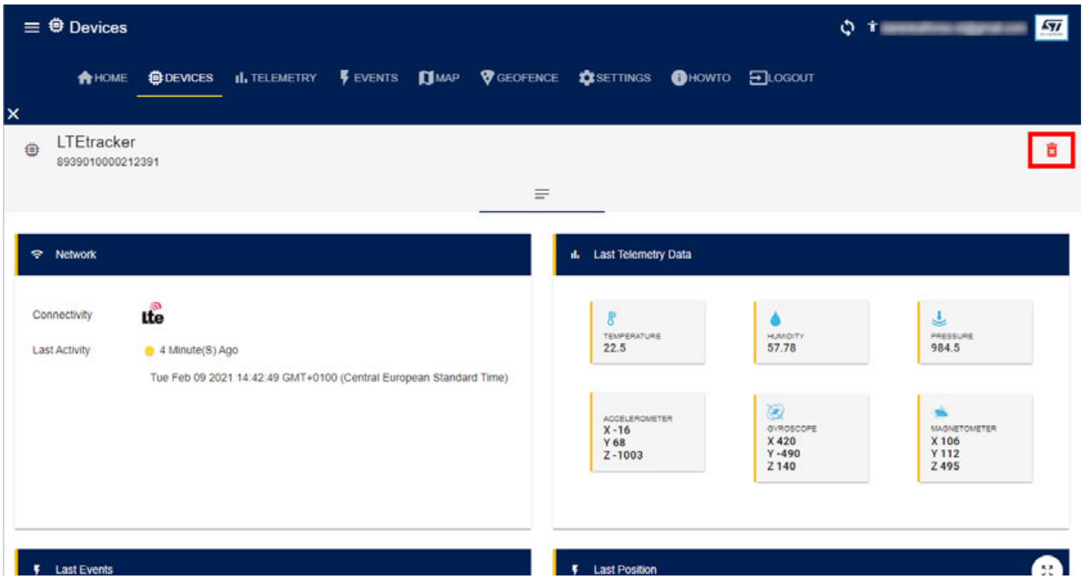


Figure 27. **Dashboard: delete a registered board section**

4 System setup guide

4.1 Hardware description

This section summarizes the hardware components needed for connecting the STM32 based platform to the Amazon AWS IoT console.

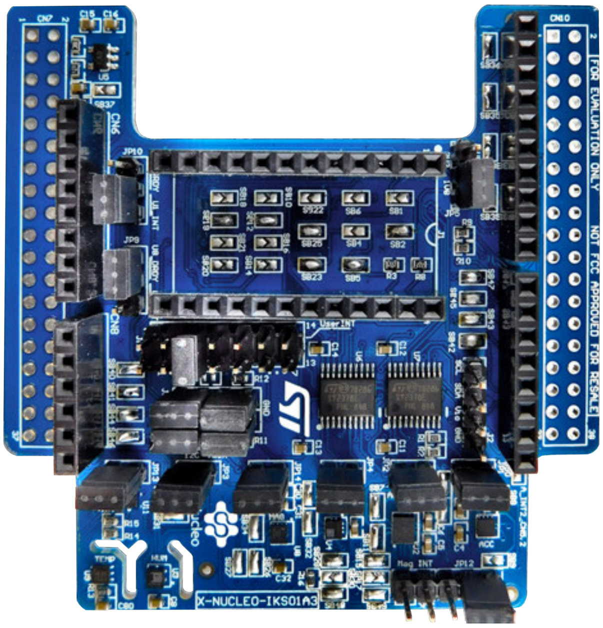
4.1.1 X-NUCLEO-IKS01A3 expansion board

The X-NUCLEO-IKS01A3 is a motion MEMS and environmental sensor evaluation board system.

It is compatible with the Arduino UNO R3 connector layout and features the LSM6DSO 3-axis accelerometer + 3-axis gyroscope, the LIS2MDL 3-axis magnetometer, the LIS2DW12 3-axis accelerometer, the HTS221 humidity and temperature sensor, the LPS22HH pressure sensor, and the STTS751 temperature sensor.

The X-NUCLEO-IKS01A3 interfaces with the STM32 microcontroller via the I²C pin, and it is possible to change the default I²C port.

Figure 28. X-NUCLEO-IKS01A3 MEMS and environmental sensor expansion board



4.1.2 X-NUCLEO-GNSS1A1 expansion board

The X-NUCLEO-GNSS1A1 expansion board is based on the Teseo-LIV3F tiny GNSS module.

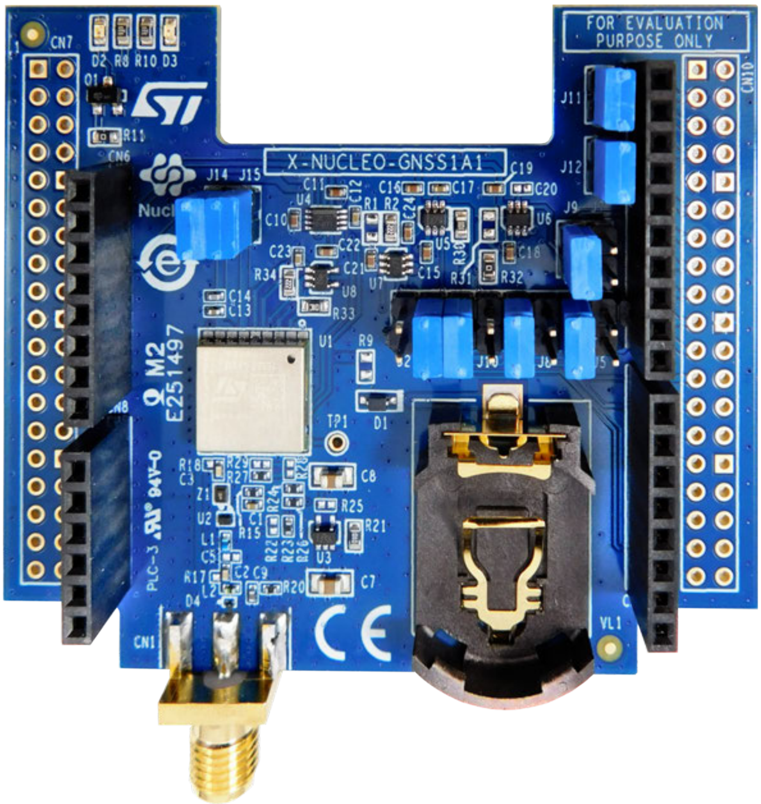
It represents an affordable, easy-to-use, global navigation satellite system (GNSS) module, embedding a TeseoIII single die standalone positioning receiver IC, usable in different configurations in your STM32 Nucleo project.

The Teseo-LIV3F is a compact (9.7x10.1 mm) module that provides superior accuracy thanks to the on-board 26 MHz temperature compensated crystal oscillator (TCXO) and a reduced time-to-first fix (TTFF) with its dedicated 32 KHz real-time clock (RTC) oscillator.

The Teseo-LIV3F module runs the GNSS firmware (X-CUBE-GNSS1) to perform all GNSS operations including acquisition, tracking, navigation and data output without external memory support.

The X-NUCLEO-GNSS1A1 expansion board is compatible with the Arduino™ UNO R3 connector and the ST morpho connector, so it can be plugged to the STM32 Nucleo development board and stacked with additional STM32 Nucleo expansion boards.

Figure 29. **X-NUCLEO-GNSS1A1 expansion board**



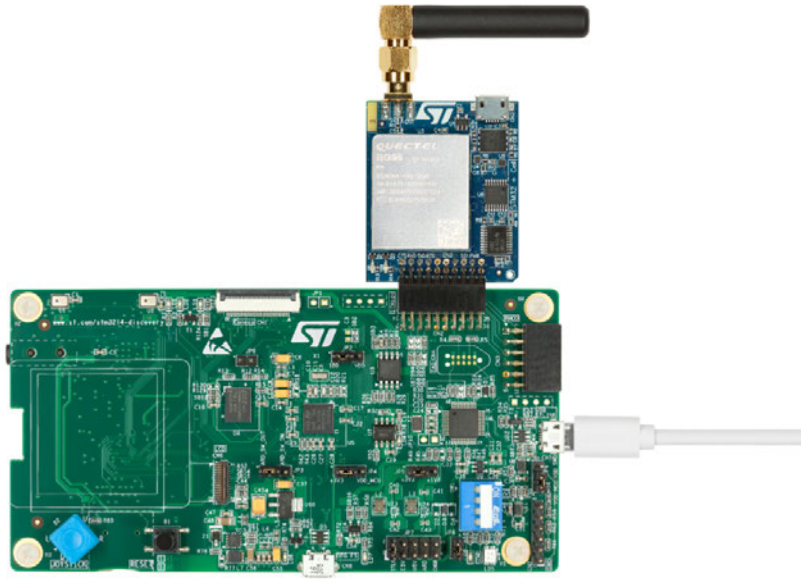
4.1.3 P-L496G-CELL02 discovery pack

The P-L496G-CELL02 STM32 discovery pack for LTE IoT cellular to cloud (STM32-C2C/LTE IoT) is a turnkey development platform for cellular and cloud technology-based solutions.

The discovery pack consists of the following boards:

1. An STM32L496AGI6-based low-power discovery motherboard with preloaded firmware (32L496GDISCOVERY). The microcontroller has 1 Mbyte of Flash memory and 320 Kbytes of RAM in a UFBGA169 package.
2. An STMod+ cellular expansion board with antenna. The expansion board features a Quectel BG96 worldwide cellular modem LTE Cat M1/Cat NB1/EGPRS module 300 kbps downlink, 375 kbps uplink.

Figure 30. **P-L496G-CELL02 LTE cellular to cloud pack**



4.2 Hardware setup

The following hardware components are needed:

1. A P-L496G-CELL02 LTE cellular to cloud pack
2. A sensor expansion board (order code: X-NUCLEO-IKS01A3)
3. A GNSS expansion board (order code: X-NUCLEO-GNSS1A1)
4. A USB type A to Mini-B USB cable to connect the board to the PC

4.2.1 P-L496G-CELL02 LTE cellular to cloud pack and expansion board setup

Step 1. Connect the P-L496G-CELL02 host board to the modem board via the STMod+ connector.

Step 2. When the UICC chip soldered on the add-on board is not used, insert a UICC compliant with the network used into the UICC socket.

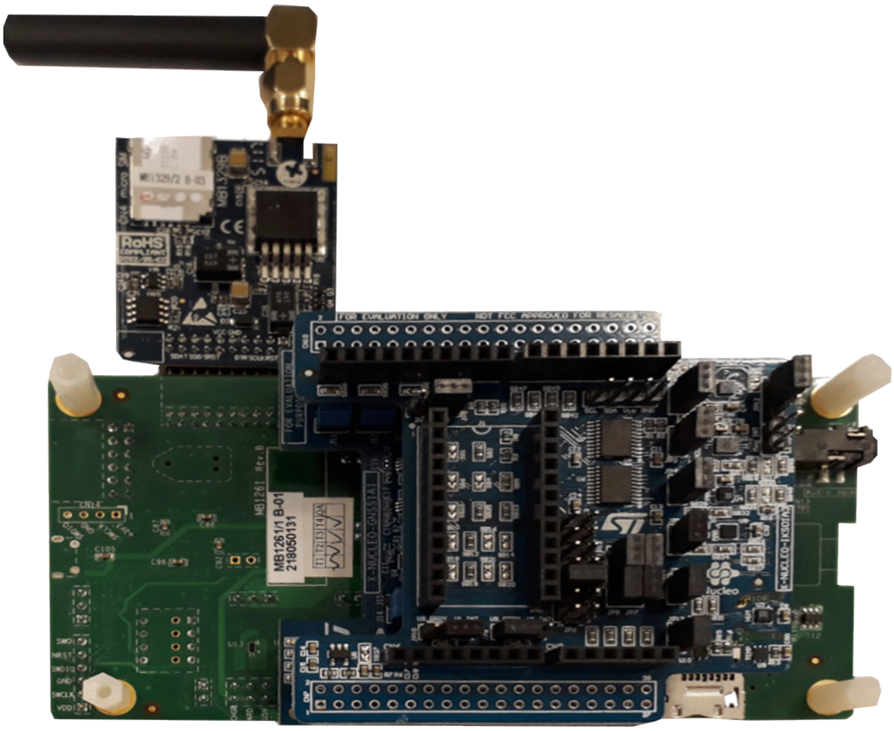
If the soldered UICC chip is used, it must be activated and selected in the boot menu.

Step 3. Power the board on by plugging its USB connector to a PC, USB power supply, or USB power bank.

Note: *The USB connector must be connected to a PC with an open console application. For instance, power bands are needed when 2G is selected under particular radio conditions.*

Step 4. Connect the X-NUCLEO-IKS01A3 and the X-NUCLEO-GNSS1A1 expansion boards to the P-L496G-CELL02 through the Arduino UNO R3 connector as shown below.

Figure 31. X-NUCLEO-IKS01A3 and X-NUCLEO-GNSS1A1 expansion boards connected to P-L496G-CELL02



4.2.2 X-NUCLEO-GNSS1A1 jumper setup

On the X-NUCLEO-GNSS1A1 expansion board, jumpers from J1 to J10 have to remain opened, while jumpers from J11 to J15 have to be closed, as shown in the following picture.

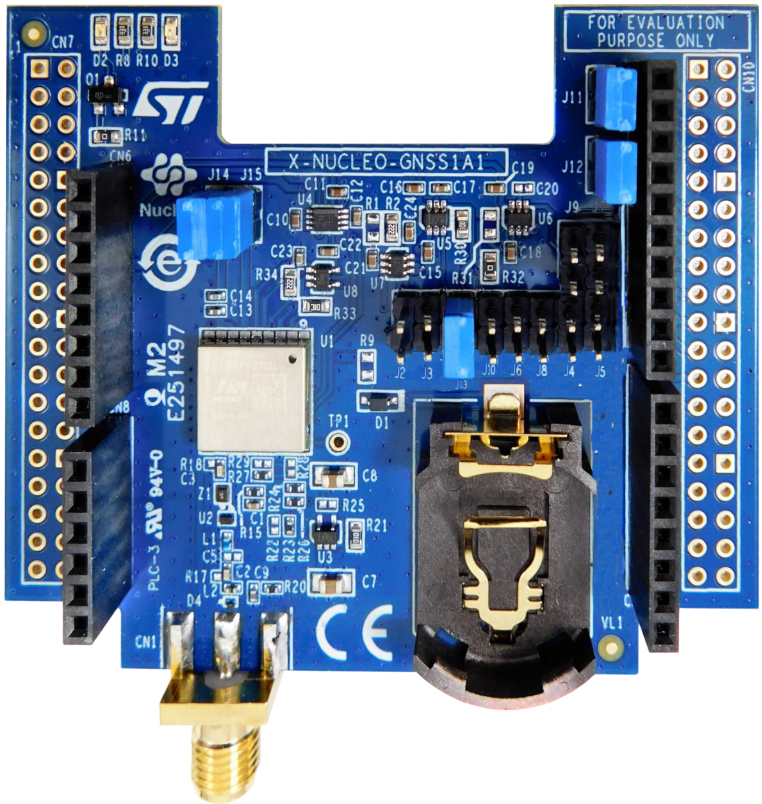


Figure 32. **X-NUCLEO-GNSS1A1 expansion board jumper settings**

4.3 Software setup

The following software components are needed to set up a suitable development environment for compiling and running the FP-ATR-LTE1 package:

- FP-ATR-LTE1 software, available on www.st.com/stm32cube
- One of the following development tool-chain and compilers:
  - IAR Embedded Workbench for ARM® toolchain plus ST-LINK
  - RealView Microcontroller Development Kit toolchain plus ST-LINK
  - STM32CubeIDE plus ST-LINK

Note: Project files for the supported IDEs can be found in FP-ATR-LTE1-Release//Projects/32L496GDISCOVERY/Applications/Cloud/AssetTracker.

- a serial line monitor (e.g. Tera Term)

Revision history

Table 2. **Document revision history**

Date	Version	Changes
15-Apr-2020	1	Initial release.
12-Feb-2021	2	Updated Section 2.1 Overview, Section 2.2 Architecture, Section 3 How to run the firmware application, Section 3.2 Configuration of cellular credentials, Section 3.2.1 How to register the board to the ST IoT web dashboard, Section 3.3.1 Blank board, Section 3.3.2 Provisioned board, Section 3.4 Data visualization in the web dashboard and Section 3.5 Delete a registered board.

Contents

List of figures

List of tables