

FMB130 Getting Started with AWS IoT Core

[Main Page](#) > [Advanced Trackers](#) > [FMB130](#) > [FMB130 Manual](#) > **FMB130 Getting Started with AWS IoT Core**

□

Contents

- **[1 Document Information](#)**
 - [1.1 Glossary](#)
 - [1.2 Revision History \(Version, Date, Description of change\)](#)
- **[2 Overview](#)**
- **[3 Hardware Description](#)**
 - [3.1 DataSheet](#)
 - [3.2 Standard Kit Contents](#)
 - [3.3 User Provided Items](#)
- **[4 Set up your Development Environment](#)**
 - [4.1 Tools Installation \(IDEs, Toolchains, SDKs\)](#)
 - [4.2 Other software required to develop and debug applications for the device](#)
- **[5 Set up your hardware](#)**
- **[6 Document Information](#)**
 - [6.1 Glossary](#)
- **[7 Other software required to develop and debug applications for the device](#)**
- **[8 Setup your AWS account and Permissions](#)**
- **[9 Create Resources in AWS IoT](#)**
- **[10 Provide Device with credentials](#)**
 - **[10.1 AWS IoT Core Configuration](#)**
 - [10.1.1 Setting up AWS IoT Core](#)
 - [10.1.2 Finding device data endpoint \(server domain\)](#)
 - **[10.2 Configuring the device](#)**
 - **[10.2.1 Security and certificates](#)**
 - [10.2.1.1 Using certificate, private key and root certificate. \(Via Cable\)](#)
 - [10.2.2 Device GPRS configuration for AWS IoT Custom MQTT settings](#)
- **[11 Checking received data and sending commands in the AWS IoT core](#)**
- **[12 Debugging](#)**
- **[13 Troubleshooting](#)**
- **[14 Troubleshooting](#)**
- **[15 Debugging](#)**

Document Information

Glossary

- FMB130 (tracker) - GNSS tracking device manufactured by Teltonika Telematics.
- Wiki - Teltonika IoT knowledge base - <https://wiki.teltonika-iot-group.com/>.
- FOTA - Firmware Over The Air.

- Configurator - Tool to configure Teltonika Telematics devices.
- Crowd support forum - knowledge base dedicated for Troubleshooting.

Revision History (Version, Date, Description of change)

Version	Date	Description
v1.5	2023.02.14	Links updated
v1.4	2022.12.19	Minor information update
v1.3	2022.11.29	Page created

Overview

FMB130 is ADVANCED tracker with internal GNSS, GSM antennas, configurable digital/analogue inputs/negative input/impulse inputs, three DOUT outputs, Bluetooth connectivity and backup battery.

Teltonika keeps its promise to make IoT Easy. FMB130 device is the first device with negative input possibility. It allows you to save your precious integration time while connecting accessories like vehicle door sensor, alarm button, seatbelt detection and similar sensors or accessories.

Additionally, we have integrated possibility to enable Impulse Input for fuel flow meter data reading. Impulse based fuel usage control is much more accurate than different kind of fuel level sensors and enables possibility to be precise if needed. It is perfect solution for high fuel usage machines like cranes, construction vehicles, mining machines, agriculture equipment.

We have added one additional Digital Output. Now it is 3 of them in total! It is not surprising that DOUT is the most popular connection type on FMB1 family devices. It` s allows you to use simultaneously 3 different DOUT scenarios like: immobilizer, buzzer, alarm lights, door lock and others.

New FMB130 is the first advanced vehicle tracking system with programmable DIN/AIN/Negative/Impulse input. First input has following choices: Digital, Analog, Negative, Impulse. Second input may be Digital or Analog.

With these new features you can have one tracking system for much wider range of use cases.

Currently for MQTT solution evaluation firmware is required to be used - 03.27.10.Rev.520. For firmware supporting MQTT please contact your sales manager or contact directly via Teltonika Helpdesk.

Changes in firmware versions and update information can be found in device wiki page: [FMB130 firmware errata](#)

Hardware Description

DataSheet

FMB130 device data sheet can be downloaded here: [DataSheet](#)

Standard Kit Contents

STANDARD PACKAGE CONTAINS

- 10 pcs. of FMB130 trackers
- 10 pcs. of Input/output power supply cables (0.9 m)
- Packaging box with Teltonika branding

Teltonika suggest standard order codes for the device purchase, by contacting us, we can create special order code which would fulfill user needs.

More ordering information at: [Ordering](#)

User Provided Items

- Power supply (10-30V).
- MicroUSB to USB A cable.

Set up your Development Environment

Tools Installation (IDEs, Toolchains, SDKs)

FMB130 comes with our created firmware, therefore no additional development or scripting is required for this unit to support AWS IoT. Only by using Teltonika Configurator [FM Configurator versions](#), connection point of AWS IoT server is required.

Other software required to develop and debug applications for the device

For debugging situations, device internal logs can be downloaded OTA by using our [FotaWEB](#) platform or by using Teltonika Configurator.

Set up your hardware

All details about FMB130 can be located in our dedicated wiki page [FMB130 Wiki](#)

- Basic device startup instructions provided in [FMB130 First Start](#).
- Device characteristics, power supply information: [FMB130 General description](#)
- FMB130 firmware change can be performed via [FotaWEB](#) (direct buyer gets access to this platform) or via device [Configurator](#)
- Device LED information: [FMB130 LED Status](#)
- USB driver download, datasheet and quick start guide downloads: [FMB130 Downloads](#)

Document Information

Glossary

- Wiki - Teltonika IoT knowledge base - <https://wiki.teltonika-iot-group.com/>.
- FOTA - Firmware Over The Air.
- Configurator - Tool to configure Teltonika Telematics devices.
- Crowd support forum - knowledge base dedicated for Troubleshooting.

For firmware supporting MQTT please contact your sales manager or contact directly via Teltonika Helpdesk.

Other software required to develop and debug applications for the device

For debugging situations, device internal logs can be downloaded OTA by using our [FotaWEB](#) platform or by using Teltonika Configurator.

Setup your AWS account and Permissions

Refer to the online AWS documentation at Set up your AWS Account. Follow the steps outlined in the sections below to create your account and a user and get started:

- [Sign up for an AWS account](#)
- [Create a user and grant permissions](#)
- [Open the AWS IoT console](#)

Pay special attention to the Notes.

Create Resources in AWS IoT

Refer to the online AWS documentation at Create AWS IoT Resources. Follow the steps outlined in these sections to provision resources for your device:

- [Create an AWS IoT Policy](#)
- [Create a thing object](#)

Pay special attention to the Notes.

Provide Device with credentials

Whole device, AWS IoT and testing information can be downloaded in PDF format [here](#).

NOTE: MQTT will not work without uploaded TLS certificates.

AWS IoT Core Configuration

Setting up AWS IoT Core

When logged in the AWS console, click on Services on the top left hand side screen, to access IoT core.

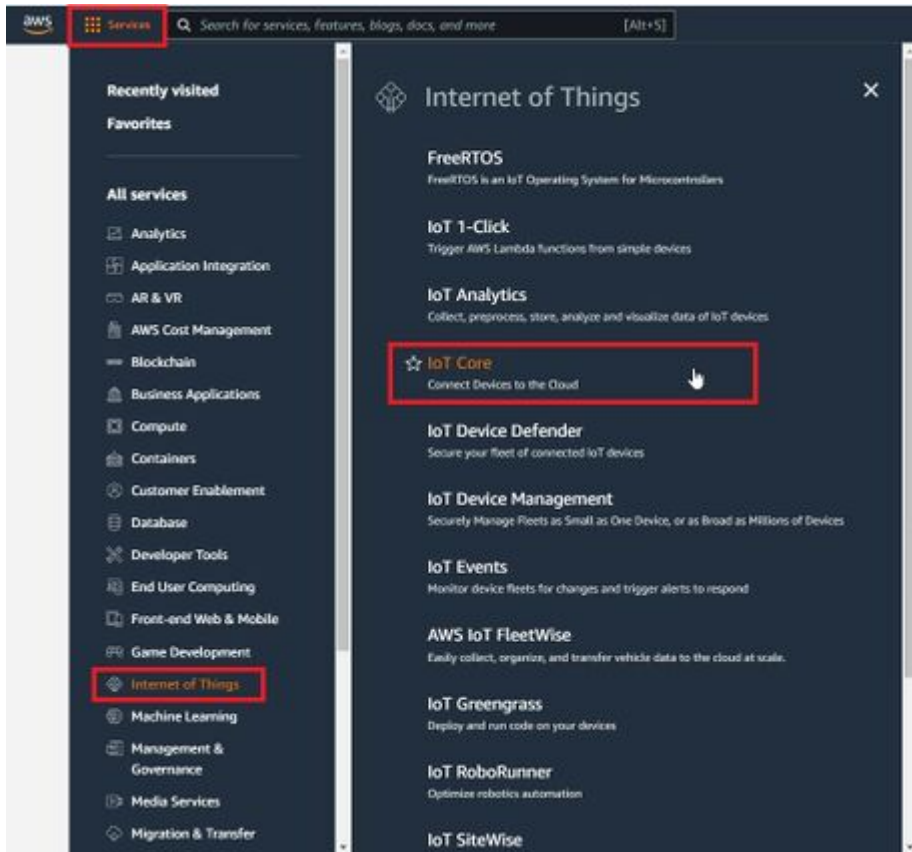


Figure 1. Accessing AWS IoT core from AWS console

NOTE: If you can't see "Services" in the top left, click on "My account" in the top right and "AWS Management Console"


Select Manage, Security, Policies (Manage > Security > Policies) and press Create policy or Create buttons. 

Figure 2. Accessing policy creation


In the Create Policy window, enter Policy name. In the Policy document tab for Policy Action (1) select "*" and for Policy resource (2) enter "*" and press create. 

Figure 3. Creating a policy


Now, that you have created a policy, select Manage on the sidebar on the left side, then select All devices, Things (Manage>All devices>Things). And click on Create things. 

Figure 4. Accessing Things

Afterwards select Create single thing and click Next.

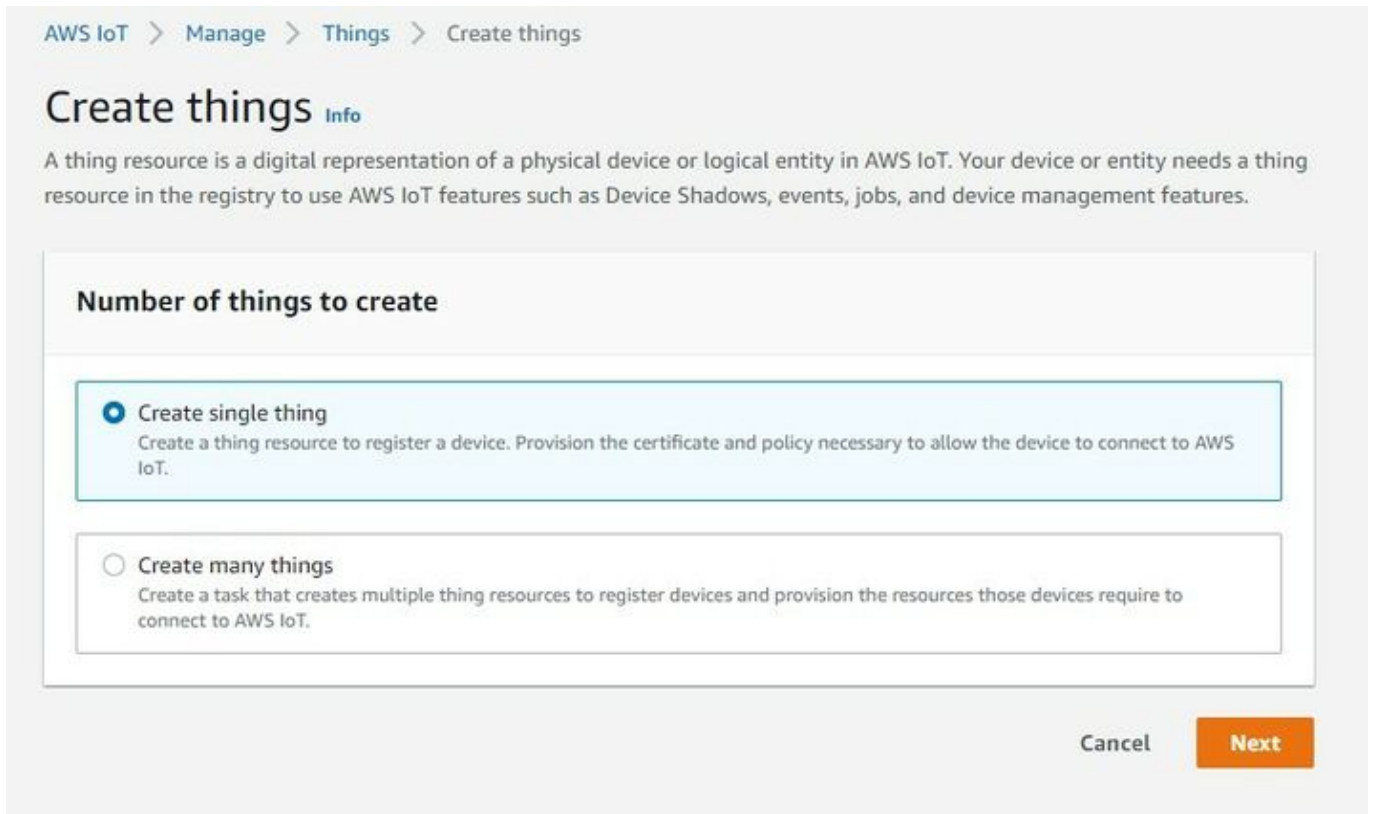


Figure 5. Creating single thing

After creating a single thing, enter Thing's name and in the Device Shadow tab select Unnamed shadow (classic). Then click Next.

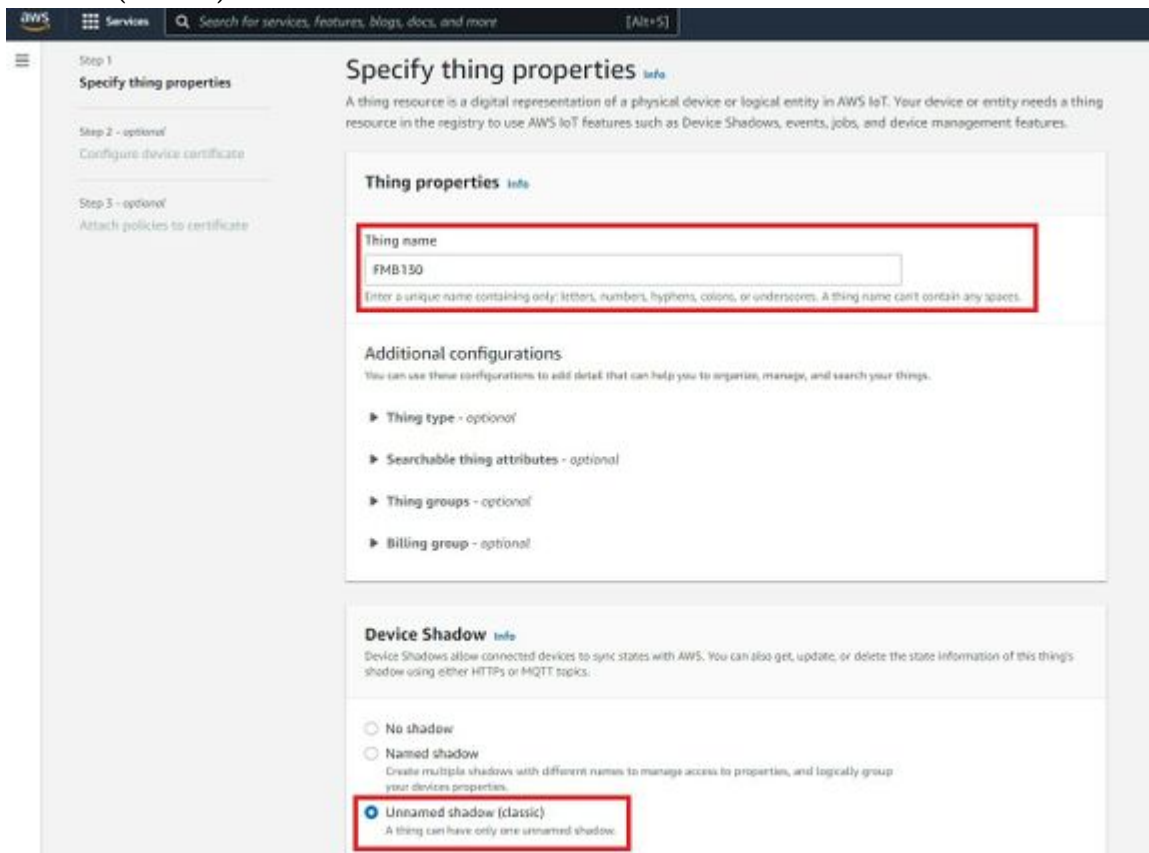


Figure 6. Thing's properties

Then when selecting Device certificate, select Auto-generate a new certificate and click Next.

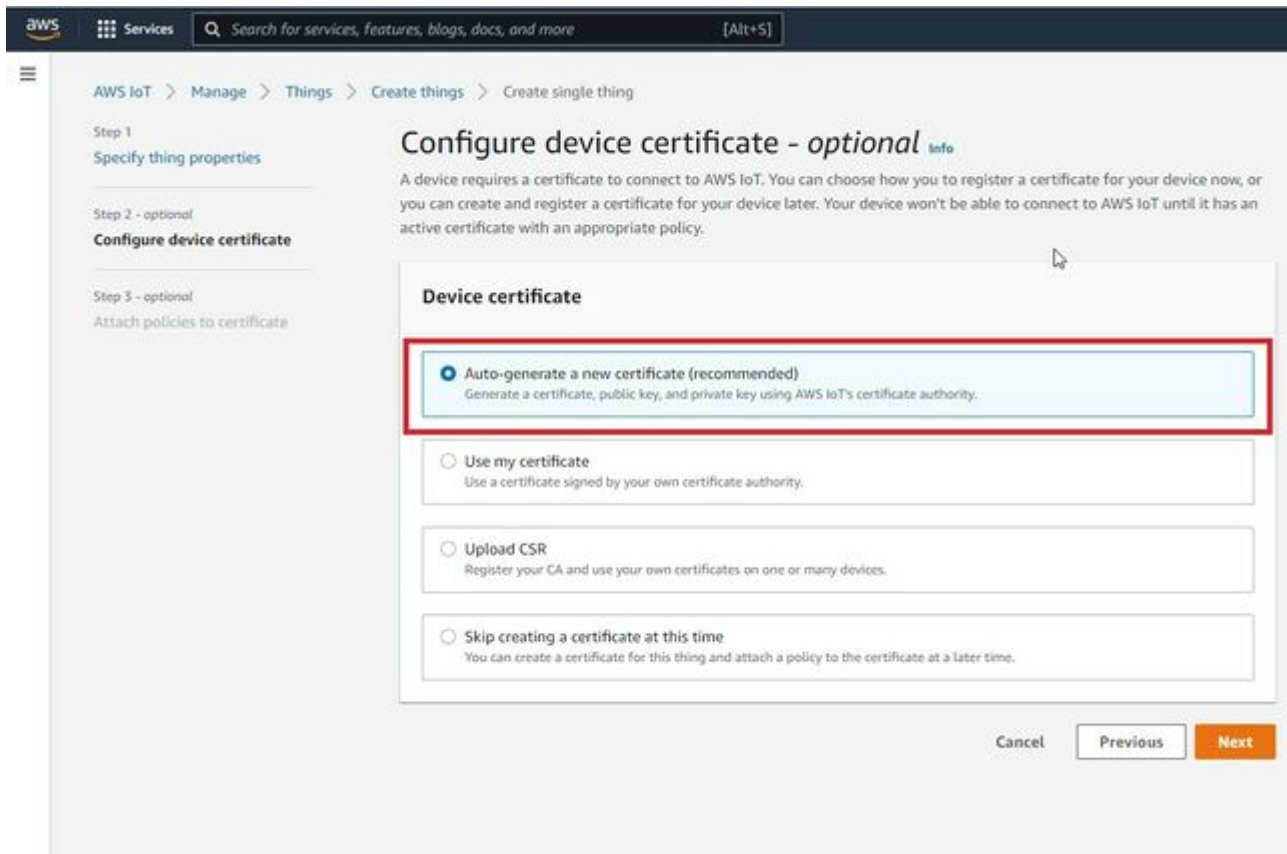


Figure 7. Certificate configuration

Now, select the policy you have created before to attach it to the certificate and thing. After that click Create thing.

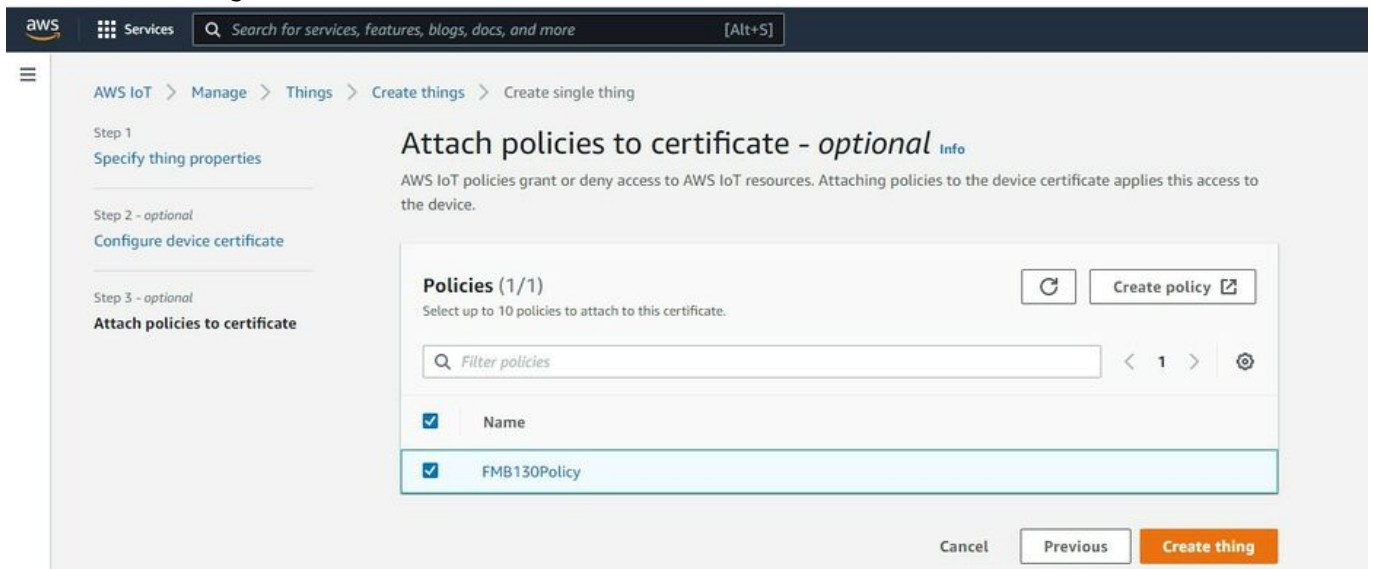


Figure 8. Attaching policy to certificate

Then window with Certificate files and key files download options should pop out. It's recommended to download all files, because later some of them will not be available for download. The files that are required for usage with FMX devices are: Device certificate (1), private key(2), and Amazon Root CA 1 file(3), but it's recommended to download them all and store them in secured place.

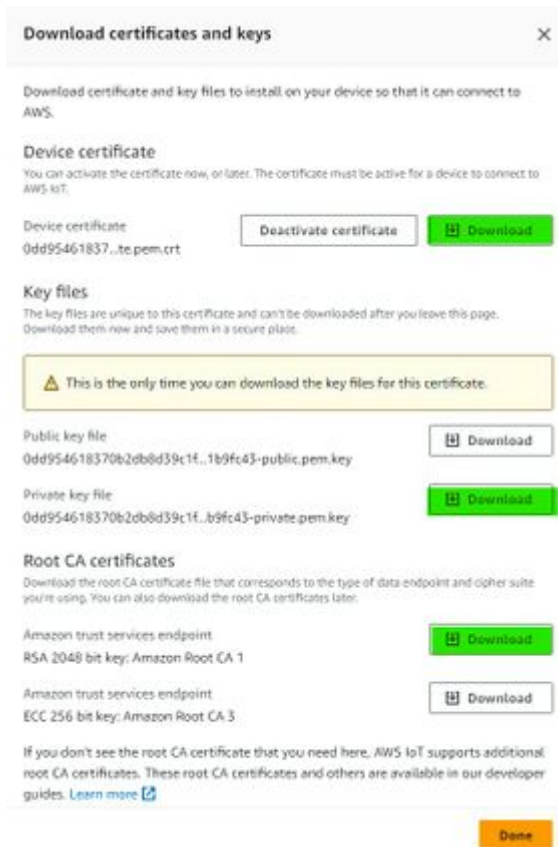


Figure 9. Certificate and key download

Finding device data endpoint (server domain)

To receive server domain (in AWS endpoint) click on the side bar on the left Settings (AWS IoT->Settings). Or click on the side bar on left side Things, select the created thing, after it click Interact->View Settings. Whole path - (Things->*YourThingName*->Interact->ViewSettings). Page containing endpoint will open. Copy the whole endpoint address. Port for accessing this endpoint is 8883.

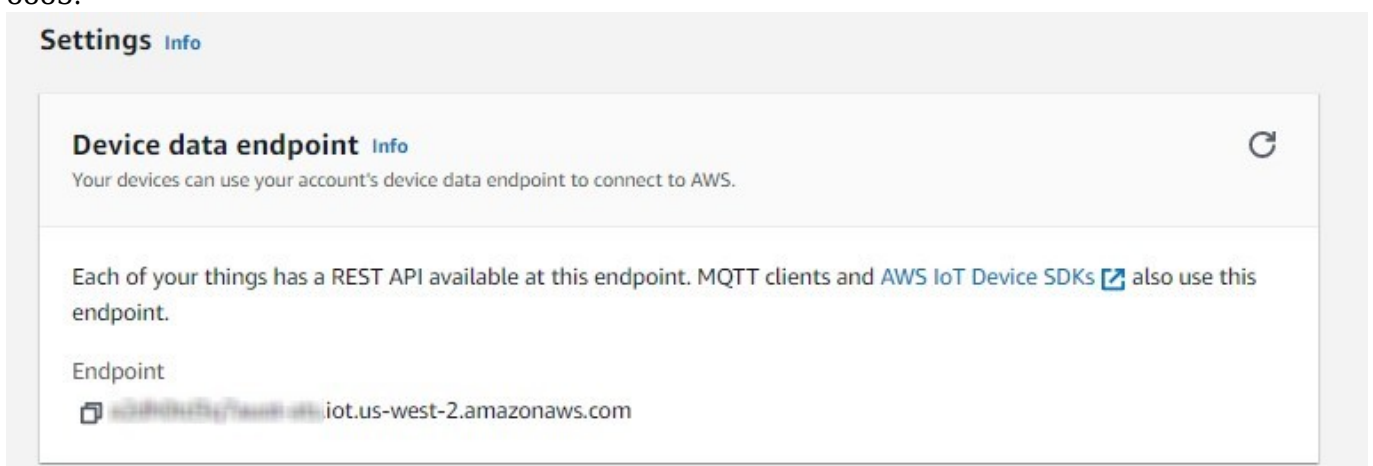


Figure 10. Device data endpoint

Configuring the device

Security and certificates

Using certificate, private key and root certificate. (Via Cable)

Find Certificate file ending with extension pem.crt (ending may be just .pem) Private key file and AmazonRootCA1 file (no need to change filenames). These files should have been downloaded when creating Thing in AWS IoT Core.

Download certificates and keys ✕

Download certificate and key files to install on your device so that it can connect to AWS.

Device certificate
You can activate the certificate now, or later. The certificate must be active for a device to connect to AWS IoT.

Device certificate
0dd954618370b20b0d39c1f..b9fc43-[Deactivate certificate](#) [Download](#)

Key files
The key files are unique to this certificate and can't be downloaded after you leave this page. Download them now and save them in a secure place.

⚠ This is the only time you can download the key files for this certificate.

Public key file
0dd954618370b20b0d39c1f..b9fc43-public.pem.key [Download](#)

Private key file
0dd954618370b20b0d39c1f..b9fc43-private.pem.key [Download](#)

Root CA certificates
Download the root CA certificate file that corresponds to the type of data endpoint and cipher suite you're using. You can also download the root CA certificates later.

Amazon trust services endpoint
RSA 2048 bit key: Amazon Root CA 1 [Download](#)

Amazon trust services endpoint
ECC 256 bit key: Amazon Root CA 3 [Download](#)

If you don't see the root CA certificate that you need here, AWS IoT supports additional root CA certificates. These root CA certificates and others are available in our developer guides. [Learn more](#)

[Done](#)

Figure 17. Certificate, private key and root certificate

Upload the mentioned files in the Security tab in the Teltonika Configurator.



Figure 18. Uploading certificates and keys

After uploading certificates, go to System tab and in Data protocol section select - Codec JSON.



Figure 19. Choosing data protocol

Device GPRS configuration for AWS IoT Custom MQTT settings

In the GPRS tab, under Server Settings select:

1. Domain - Endpoint from the AWS, Port: 8883
2. Protocol - MQTT
3. TLS Encryption - TLS/DTLS

In the MQTT Settings section select:

1. MQTT Client Type - AWS IoT Custom
2. Device ID - enter device IMEI (optional)
3. Leave Data and Command Topics unchanged.

Save the configuration to the device.



Figure 27. GPRS Settings for MQTT AWS IoT

Checking received data and sending commands in the AWS IoT core


The data received from the device can be found in the MQTT test client, which can be found above "Manage" in the sidebar on the left. 

Figure 28. MQTT test client location

To see incoming data, subscribe to topic - *DeviceImei*/data . Or subscribe to # to see all incoming outgoing data in the Topics.

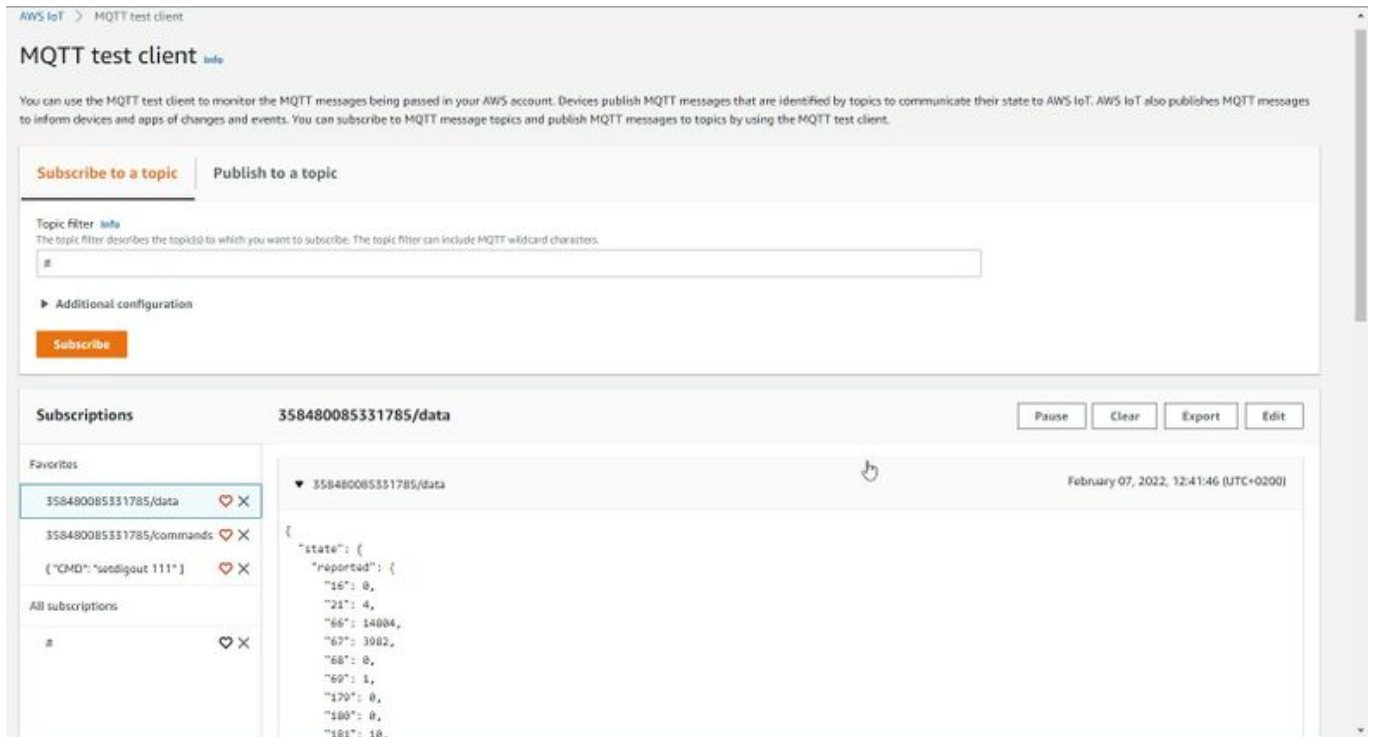


Figure 29. MQTT test client

Incoming data is received in JSON format, for e.g.:



Figure 30. Received data format

To send SMS/GPRS commands to the device subscribe to a topic name - *DeviceIMEI*/commands, and, in the same MQTT test client window select Publish to a topic. Enter topic name - *DeviceIMEI*/commands. In the Message payload enter wanted GPRS/SMS command in following format and press Publish:

{“CMD”: “<Command>”}

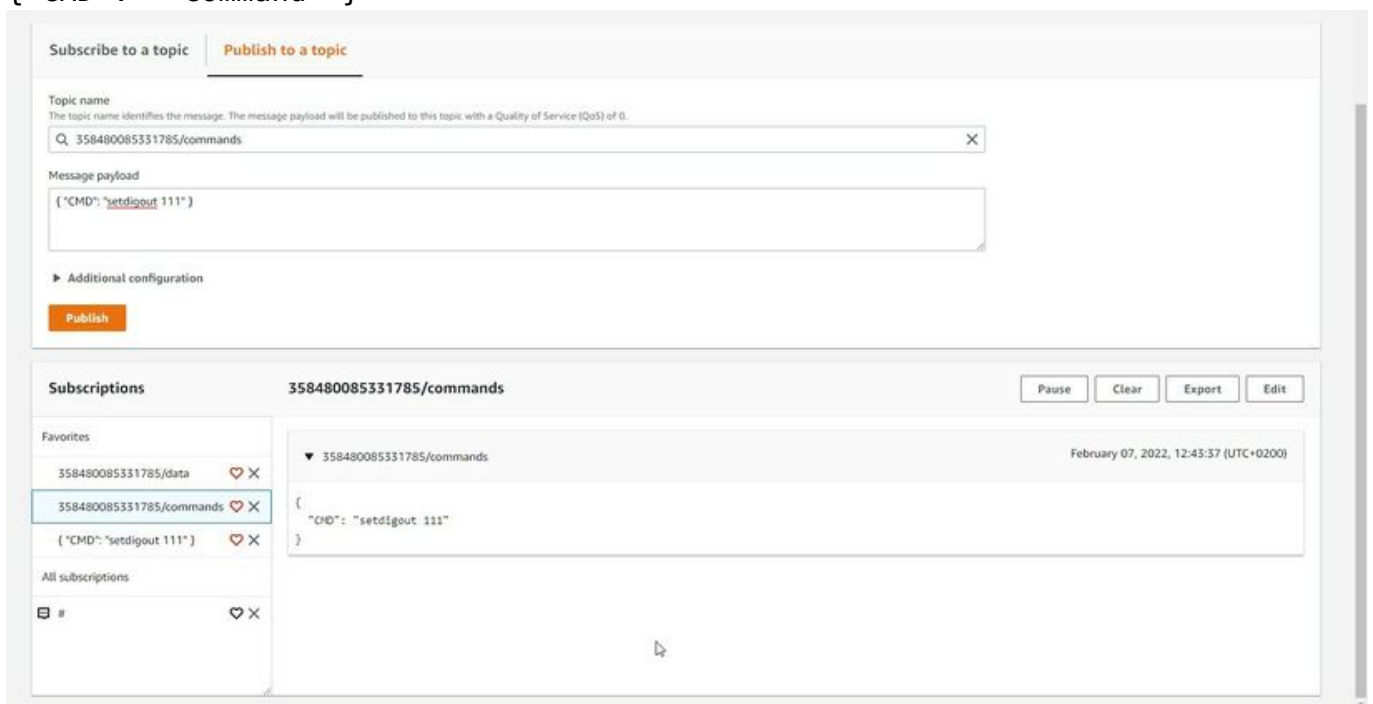


Figure 31. Sending Command in AWS IoT Core

The response to the command will be shown in the Data topic:



Figure 32. Response to a command in the data topic, the command was published in command topic

Debugging

In the situation when the issue with information upload appears, device internal logs can be taken directly from device configuration software ([instructions](#)), via Terminal.exe by connecting selecting device USB connection port, or by receiving internal logs via FotaWEB in [task section](#).

Troubleshooting

The information can be submitted to Teltonika HelpDesk and Teltonika engineers will assist with troubleshooting. For a more detailed information regarding what information should be collected for debugging, please visit the dedicated page on [Teltonika Wiki](#).

Alternatively, Teltonika has a [Crowd Support Forum](#) dedicated for troubleshooting, where engineers are actively solving problems.

Troubleshooting

The information can be submitted to Teltonika HelpDesk and Teltonika engineers will assist with troubleshooting. For a more detailed information regarding what information should be collected for debugging, please visit the dedicated page on [Teltonika Wiki](#).

Alternatively, Teltonika has a [Crowd Support Forum](#) dedicated for troubleshooting, where engineers are actively solving problems.

Debugging

In the situation when the issue with information upload appears, device internal logs can be taken directly from device configuration software ([instructions](#)), via Terminal.exe by connecting selecting device USB connection port, or by receiving internal logs via FotaWEB in [task section](#).