

LoRa®-End Nodes-LR1110 LoRa Edge Tracker Getting Started Guide for AWS IoT Core Device Location

Table of Contents

1	Document Information	1
2	About AWS IoT Core Device Location.....	1
3	Overview.....	1
4	Hardware Details.....	2
5	Setup your AWS account and Permissions.....	2
6	Set up and onboard your gateway to AWS.....	2
7	Set up your location tracker device.....	3
8	Provision the location tracker device in AWS.....	7
9	Verify Operation with AWS IoT Core Device Location	10
10	Troubleshooting.....	12
11	Additional Software References.....	12

1 Document Information

1.1 Naming Conventions

The term “downlink device” or “endpoint device” is used in this document to refer to a LoRaWAN device that connects to a LoRaWAN “Gateway”. The “Gateway” in turn, connects to AWS IoT Core for LoRaWAN.

1.2 Revision History (Version, Date, Description of change)

Version	Date	Description
V0.1	2023.11.02	Page created

2 About AWS IoT Core Device Location

Refer to the AWS online documentation for information about [AWS IoT Core Device Location](#) and how it works.

3 Overview

LR1110 Ultra-low power platform that integrates a LoRa® transceiver, long range FHSS modulator, multi-constellation scanner, and passive Wi-Fi AP MAC address scanner targeting asset management applications. Must be based on the Semtech LR1110 chipset and use the LoRa Basics™ Modem-E firmware stack. In this guide, the LoRa®-End Nodes-LR1110 LoRa Edge Tracker can connect seamlessly with AWS IoT Core for LoRaWAN and you can easily integrate LR1110 into your IoT solution stack.

4 Hardware Details

4.1 DataSheet

LoRa®-End Nodes-LR1110 LoRa Edge Tracker data sheet can be downloaded here: [DataSheet](#)

4.2 Standard Kit Contents

The Kit for the LoRa Edge Tracker Reference Design is delivered with the following contents:

- 1x LR1110 Tracker Reference Design for 868 MHz or 915 MHz frequency band
- 2x ER14250 3.6VDC/1200mAh Batteries (located inside the Tracker enclosure)
- 1x Magnet for Tracker activation

4.3 User Provided items

You will need to provide a LoRaWAN gateway. We recommend using an AWS IoT Core for LoRaWAN qualified device (you can find one in the [AWS Partner Device Catalog](#)).

4.4 Additional Hardware References

- <https://www.semtech.com/>
- <https://www.semtech.com/products/wireless-rf/lora-edge/lr1110>

5 Setup your AWS account and Permissions

If you don't have an AWS account, refer to the instructions in the guide [here](#). The relevant sections are [Sign up for an AWS account](#) and [Create an administrative user](#).

If you have an AWS account and administrative user already set up, proceed with the steps below.

6 Set up and onboard your gateway to AWS

Set up your [qualified gateway](#) following documentation provided for the gateway. Once the gateway has been set up, onboard it to AWS IoT Core for LoRaWAN following the [online instructions](#).

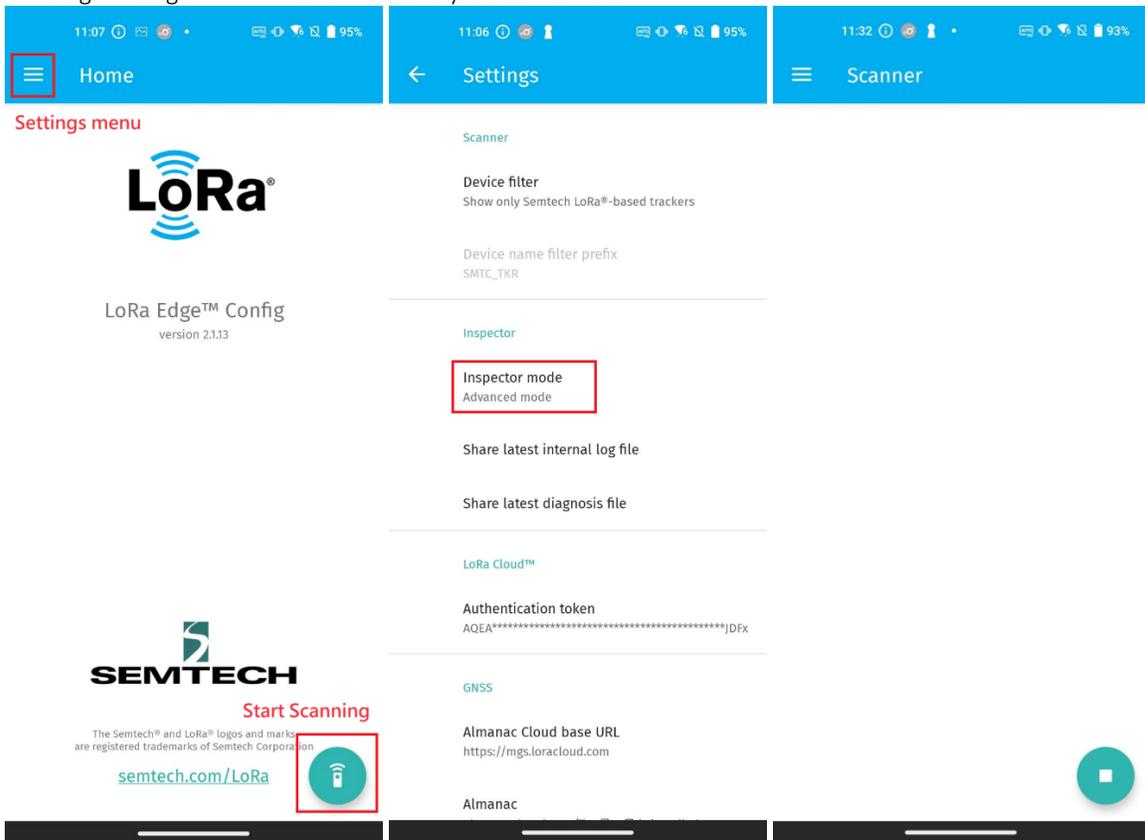
7 Set up your location tracker device

- Order a tracker from a [distributor](#) of compatible trackers
- Set up the tracker
 1. Install the **LoRa Edge Config** mobile app from Google Play or App Store on your phone.



Search: LoRa Edge Config

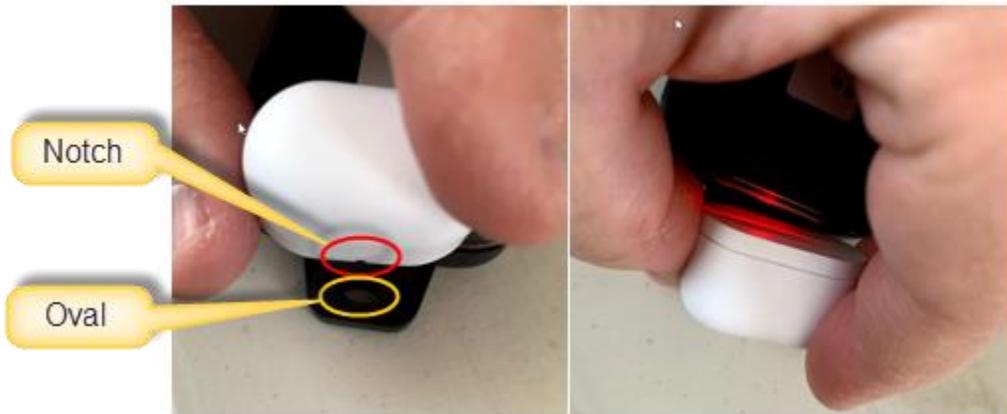
2. Go to **Settings** in the mobile app and select **Inspector mode**. Once there, enable **Advance mode** (this will give you access to additional settings for the tracker's configuration). Click **Start Scan** icon and allow LoRa Edge Config to access the location of your mobile device.



3. Unbox your tracker

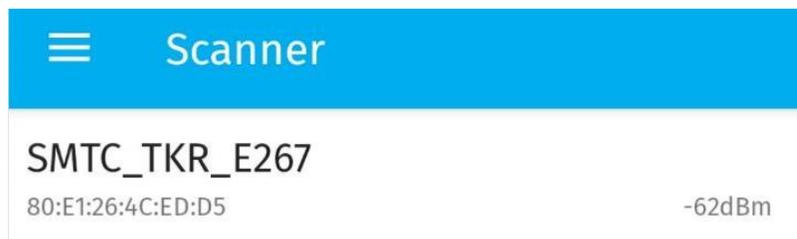


4. Put your LoRa Edge Tracker Reference Design in BLE pairing mode by aligning the provided magnet, with the notch side down, against the oval hole on the device. Then place the flat side of the magnet directly against the device, as illustrated in the figure. The red light will flash on the tracker when it has successfully been put into pairing mode. It remains in this mode for 30 seconds.

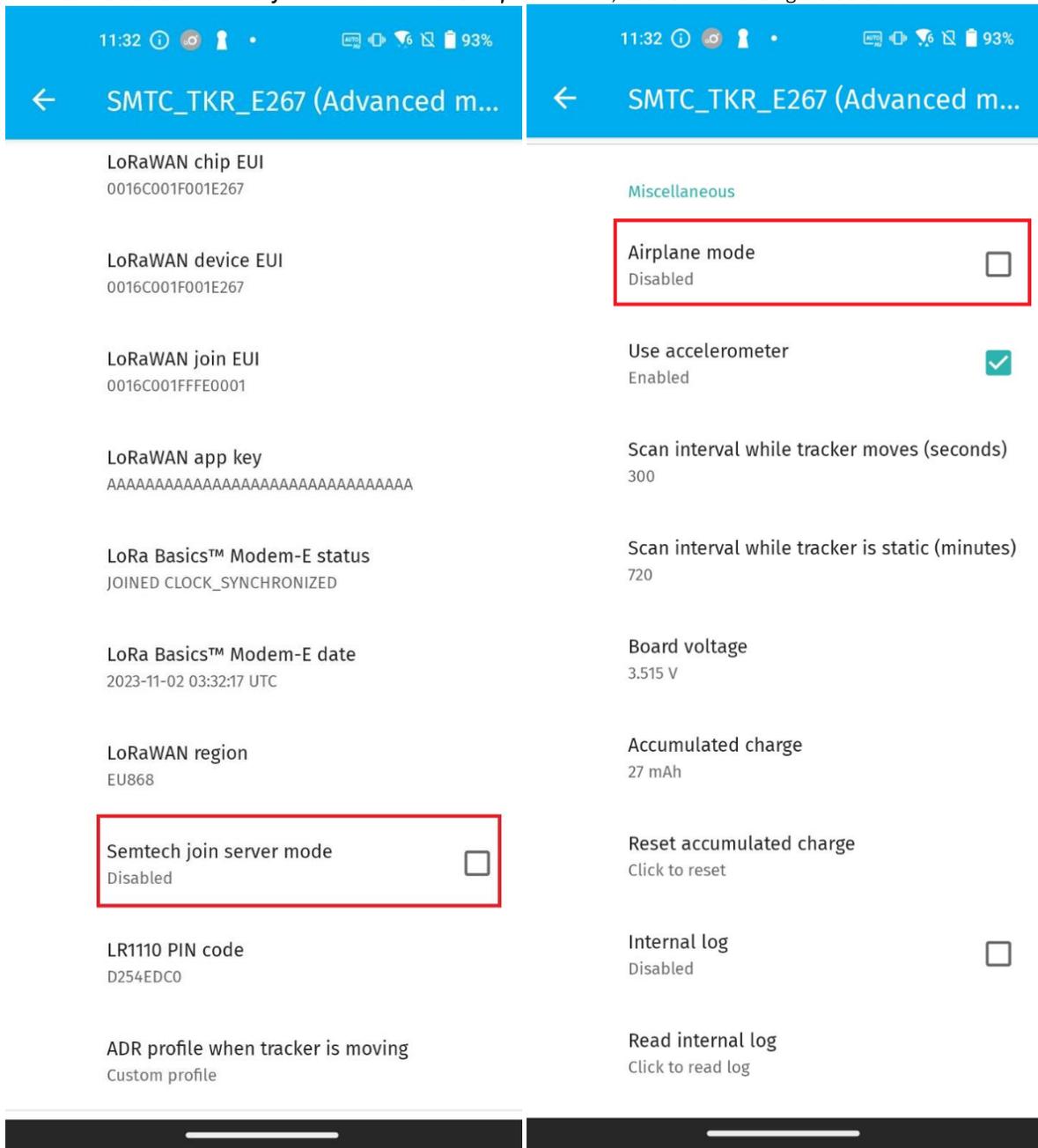


Source of photo credit : [LoRa developers Semtech documentation](#)

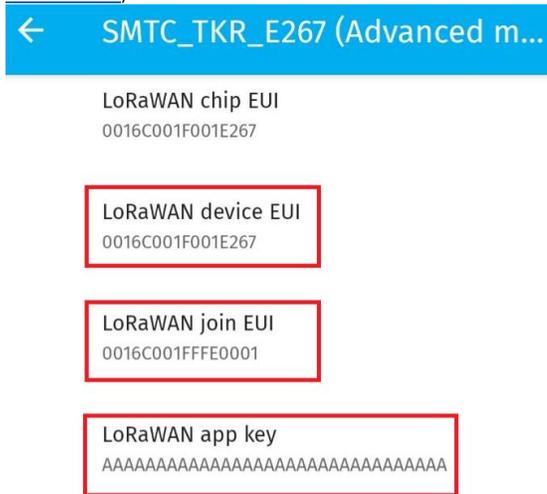
5. The tracker should appear on the Trackers section of the mobile app. The name of your LoRa Edge Tracker Reference Design is displayed. The last four digits are the DevEUI LSB, RSSI and MAC address. The DevEUI LSB must be the same as the DevEUI printed on the tracker label. Select the tracker in the mobile app by tapping on its name as shown in the figure below:



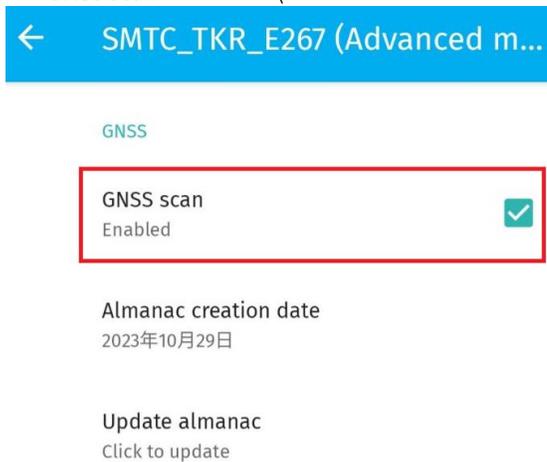
6. Disabled *Semtech join server mode* and *Airplane mode*, as shown in the figure below:



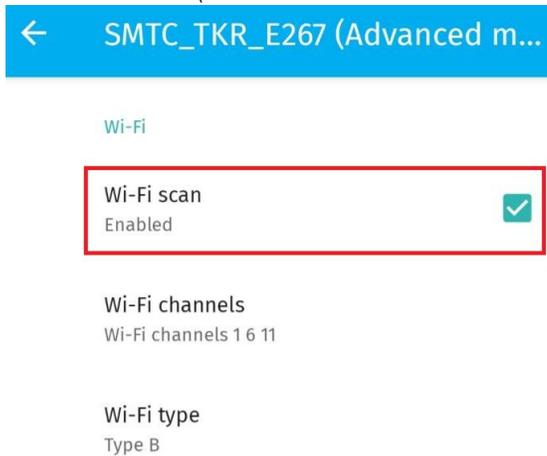
- Retrieve and copy the *LoRaWAN device EUI(DevEui)* 、 *LoRAWAN join EUI(AppEui)* and *LoRaWAN app key(AppKey)* attributes from your tracker. You will use these in the next step (see section [Create the device in AWS IoT](#)).



- To use GNSS estimation for the LoRa Edge Tracker Reference Design, you must enable it by checking the *GNSS scan* checkbox. (Please Choose one of GNSS and Wi-Fi to enable)



- To use Wi-Fi to scan for a LoRa Edge Tracker Reference Design, enable Wi-Fi scanning by checking the *Wi-Fi scan* checkbox. (Please Choose one of GNSS and Wi-Fi to enable)



For additional details, refer to the [Mobile Application: LoRa Edge Config™](#)

8 Provision the location tracker device in AWS

8.1.1 Prerequisites

There should be enough information available about the device to provision it into the AWS IoT Core for LoRaWAN service. Steps to provision the tracker are detailed below. For additional details, refer to the AWS [LoRaWAN developer guide](#).

For the list of valid device classes and versions supported by AWS IoT Core for LoRaWAN, refer to [LoRaWAN version](#), [Activation modes](#) and [Device classes](#). You can choose any combination that is valid for your device.

8.1.2 Create a Device Profile

Device profiles define the device capabilities and boot parameters that the network server uses to set the LoRaWAN radio access service. It includes selection of parameters such as LoRa frequency band, LoRa regional parameters version, and MAC version of the device.

The instructions below use the AWS CLI. If you prefer, you can use the AWS console to create a device profile. Refer to [Add device profiles](#).

Ensure that the RF region (e.g., US915) of your device matches the gateway's RF region.

First create the file `device-profile.json` with the appropriate settings. An example is shown below for an OTAA 1.0.3 Class A device

```
% cat device-profile.json
{
  "LoRaWAN": {
    "FactoryPresetFreqsList": [],
    "MacVersion": "1.0.3",
    "MaxDutyCycle": 10,
    "MaxEirp": 15,
    "RegParamsRevision": "RP002-1.0.1",
    "RfRegion": "US915",
    "RxDataRate2": 8,
    "RxDelay1": 1,
    "RxDrOffset1": 0,
    "RxFreq2": 9233000,
    "Supports32BitFCnt": true,
    "SupportsClassB": false,
    "SupportsClassC": false,
    "SupportsJoin": true
  },
  "Name": "testDeviceProfile"
}
```

Use the AWS CLI to create the device profile.

```
% aws iotwireless create-device-profile --cli-input-json file://device-profile.json
```

Sample output:

```
{
  "Arn": "arn:aws:iotwireless:your-account-id:your-account-id:DeviceProfile/your-device-profile-id",
  "Id": "your-device-profile-id"
}
```

```
}
```

Save the value of *your-device-profile-id* for later use.

8.1.3 Create a Service Profile

Create a service profile with `AddGwMetadata=true` so that you will receive additional gateway metadata such as RSSI and SNR for each payload.

```
% aws iotwireless create-service-profile --lorawan AddGwMetadata=true --name "testServiceProfile"
```

Sample output:

```
{
  "Arn": "arn:aws:iotwireless:your-region:your-account-id:ServiceProfile/your-service-profile-id",
  "Id": "your-service-profile-id"
}
```

Save the value of *your-service-profile-id* for later use.

8.1.4 Create uplink and location destinations

An AWS IoT Core for LoRaWAN destination allows you to either define an AWS IoT rule that processes a device's messages for use by AWS services, or send the message directly to an MQTT topic. In the steps below, we will send the message directly to an MQTT topic. Refer to the [online guide](#) for more details.

Create the following two destinations:

1. An uplink destination to route the raw messages from the device and publish it to the topic *testUplinkTopic*
2. A location destination to route the solved location data from the service to the topic *testLocationTopic*.

To create the destinations, you first need an IAM role – follow the instructions in the section [Create an IAM role](#) below.

8.1.4.1 Create an IAM role

AWS IoT Core for LoRaWAN destinations require IAM roles (you can use the same role for both destinations) that give AWS IoT Core for LoRaWAN the permissions necessary to send data to the destination topic.

To create this IAM role, follow the instructions in the sections **To create an IAM policy for your AWS IoT Core for LoRaWAN destination role**, and **To create an IAM role for an AWS IoT Core for LoRaWAN destination** as detailed online at [Create an IAM role for your destinations](#).

For **Role name**, use *testDestinationRoleName*. Once the role has been created, save the ARN associated with this role for later use.

8.1.4.2 Create the uplink and location destinations

Using the AWS CLI, create the uplink destination:

```
% aws iotwireless create-destination --name "testUplinkDestination" --expression-type "MqttTopic" --expression "testUplinkTopic" --role-arn "arn:aws:iam::your-account-id:role/testDestinationRoleName"
```

Sample output:

```
{
  "Arn": "arn:aws:iotwireless:your-region:your-account-id:Destination/testUplinkDestination",
  "Name": "testUplinkDestination"
}
```

Using the AWS CLI, create the location destination, using the same role:

```
% aws iotwireless create-destination --name "testLocationDestination" --expression-type "MqttTopic" --expression "testLocationTopic" --role-arn "arn:aws:iam::your-account-id:role/testDestinationRoleName"
```

Sample output:

```
{  
  "Arn": "arn:aws:iotwireless:your-region:your-account-id:Destination/testLocationDestination",  
  "Name": "testLocationDestination"  
}
```

8.1.5 Create the device in AWS IoT

The device can now be created using the above resources.

While creating the device, you have to specify the configuration for the tracker and specify the geolocation frame port (Fport), which is used by AWS IoT Core for LoRaWAN to pass the Wi-Fi / GNSS scan data to the AWS IoT Core Device Location service.

Create a file with the name *wireless-device.json* and specify the configuration in the file as shown below. Note that “Positioning” must be set to “Enabled”, and FPort must be 199.

```
% cat wireless-device.json  
{  
  "ClientRequestToken": "1234",  
  "Description": "Device qualification testing",  
  "DestinationName": "testUplinkDestination",  
  "LoRaWAN": {  
    "DevEui": "your-DevEui",  
    "DeviceProfileId": "your-device-profile-id",  
    "ServiceProfileId": "your-service-profile-id"  
  },  
  "FPorts": {  
    "Applications": [  
      {  
        "DestinationName": "testLocationDestination",  
        "FPort": 199,  
        "Type": "SemtechGeolocation"  
      }  
    ]  
  },  
  "OtaaV1_0_x": {  
    "AppEui": "...", // received after claiming the device  
    "AppKey": "...", // received after claiming the device  
  }  
},  
"Name": "Qualification Test Device",  
"Positioning": "Enabled",  
"Type": "LoRaWAN"  
}
```

Using the AWS CLI, create the wireless device.

```
% aws iotwireless create-wireless-device --cli-input-json file://wireless-device.json
```

Sample output:

```
{
```

```

"Arn": "arn:aws:iotwireless:your-region:your-account-id:WirelessDevice/your-wireless-device-id",
"Id": "your-wireless-device-id"
}

```

Save the value of *your-wireless-device-id* for later use.

Check that the device has connected successfully using the instructions at [Check device connection status using the console](#).

9 Verify Operation with AWS IoT Core Device Location

9.1 Check decoded data is received and republished by AWS

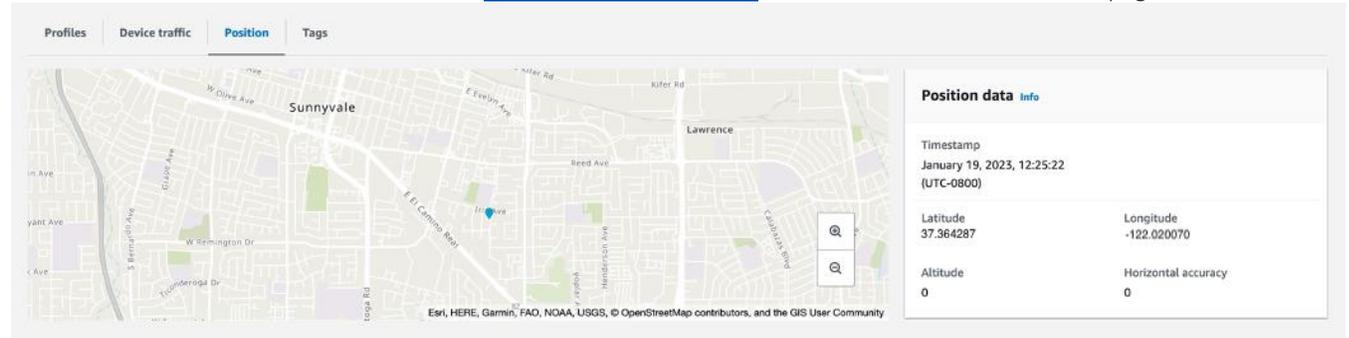
You can now check data is received and published by AWS by triggering a condition or event on the device itself:

- Go to the AWS IoT console. In the navigation pane, select Test, and choose MQTT test client.
- Subscribe to the wildcard topic `#` to receive messages from all topics (Include uplink topic and location topic)
- You should see traffic like that shown below.

testUplinkTopic	testLocationTopic
<pre> { "MessageId": "6c248b79-9e4b-4781-946c-2e497debfb2c", "WirelessDeviceId": "36a13d17-8f8b-447d-82dc-6f9fedb4ca1c", "PayloadData": "FAdPBEdUEw0BD6D0g9HtZFYkZd4zqCo=", "WirelessMetadata": { "LoRaWAN": { "ADR": false, "Bandwidth": 125, "ClassB": false, "CodeRate": "4/5", "DataRate": "2", "DevAddr": "00c2d0b3", "DevEui": "0016c001f001e267", "FCnt": 89, "FOptLen": 0, "FPort": 199, "Frequency": "868300000", "Gateways": [{ "GatewayEui": "5813d3fffe298538", "Rssi": -116, "Snr": 4.75 }], "MIC": "8693508d", "MType": "UnconfirmedDataUp", "Major": "LoRaWANR1", "Modulation": "LORA", "PolarizationInversion": false, "SpreadingFactor": 10, "Timestamp": "2023-09-06T12:18:34Z" } } } </pre>	<pre> { "coordinates": [121.00886, 24.870716, 0], "WirelessDeviceId": "36a13d17-8f8b-447d-82dc-6f9fedb4ca1c", "WirelessMetadata": { "LoRaWAN": { "FPort": 199, "lorawanGateways": [{ "gatewayEui": "5813d3fffe298538", "rssi": -116, "snr": 6, "wirelessGatewayId": "24d80428-a387-4105-9819-ffbaba9a4f40" }], "dataRate": 3, "devEui": "0016c001f001e267", "mType": "UNCONFIRMED_DATA_UP", "frequency": 867900000, "timestamp": "2023-09-06T12:38:18.728575789Z" } }, "type": "Point", "properties": { "verticalAccuracy": 0, "horizontalAccuracy": 0, "timestamp": "2023-09-06T12:38:20.725Z" } } </pre>

9.2 Check the location via AWS Console map

The device location can be visualized on the [AWS IoT Wireless console](#) in the Position tab of the device page:



9.3 Check the location using AWS CLI

Once the device location is solved, the last known position of the device is updated and invoking the [get-resource-position](#) command will yield the coordinates, along with any accuracy value.

The command should specify the resource-identifier of your wireless device. Use the value of *your-wireless-device-id* from the section [Create the device](#) or determine it using the [list-wireless-devices](#) command.

```
% aws iotwireless list-wireless-devices
```

Sample output:

```
{
  "WirelessDeviceList": [
    {
      "Name": "myLoRaWANDevice",
      "DestinationName": "IoTWirelessDestination",
      "Id": "your-wireless-device-id",
      "Type": "LoRaWAN",
      "LoRaWAN": {
        "DevEui": "ac12efc654d23fc2"
      },
      "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/your-wireless-device-id"
    }
  ]
}
```

The output of the [get-resource-position](#) command will be in GeoJSON format and will need to be stored in an output file.

```
% aws iotwireless get-resource-position --resource-identifier your-wireless-device-id --resource-type WirelessDevice
outfile.geojson
```

```
% cat outfile.geojson | jq
```

```
{
  "coordinates": [
    -122.020070,
    37.364287,
    0
  ],
  "type": "Point",
}
```

```
"properties": {  
  "verticalAccuracy": 0,  
  "horizontalAccuracy": 0,  
  "timestamp": "Thu Jan 19 20:25:22 UTC 2023"  
}  
}
```

10 Troubleshooting

- Refer to [Troubleshooting AWS IoT Core](#) for AWS IoT Core troubleshooting tips.
- Refer to [Troubleshooting Firmware Updates Over the Air](#) for common tracker troubleshooting tips.

11 Additional Software References

- <https://www.semtech.com/products/wireless-rf/lora-edge/lr1110>
- <https://www.browan.com/product/LR1110Tracker/detail>