AWS MQTT cloud connection

Contents

Introduction

In this article you will find instructions on how to setup AWS IoT as a MQTT Broker and how to setup a RUT router as a MQTT Publisher and send data over to this AWS Broker.

With that, you will be able to configure any other device as a MQTT Subscriber, and listen to any published info by the router or other devices on this same broker.

Prerequisites

You will need:

- An AWS account
- A router from RUTx, TCRx or TRBx series

Hardware Description

Hardware descriptions can be found in different Quick Start Guides (QSG). There you will find an overview of the various components on the front and back of a device, hardware installation instructions, first login information, device specifications, and general safety information. Link: Quick Start Guides

Set up your Development Environment

Teltonika Networks devices comes with our created firmware, therefore no additional development or scripting is required for this unit to support AWS IoT.

Setting up AWS IoT

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.

Creating a thing

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.

First off, open the AWS Management Console and login with your credentials. After that, you will see a screen similar to this:

On the search bar on the top, search for "IoT Core", and click on the first search result. You will see a screen like the one below, open the "Manage" section and click on "Things".

Then, do the following procedure to create a Thing: click on the "Create Things" button → "Create single thing" → Give it any name → "No shadow" →"Auto-generate a new certificate (recommended)" → "Create thing".

Certificate Handling

Certificates are used by Publishers and Subscribers to connect to your AWS MQTT Broker.

You will be prompted to download the certificates, download the "Device certificate", "Private key file, "Public key file" and "Amazon Root CA 1".

Move all the 4 files to a folder on the C:\ drive, so it's easy to locate them. Then, rename them as following: Keep the "AmazonRootCA1.pem" as it is, the file xxxxx.**pem.crt** as device_certificate.pem.crt, the file xxxxx-**private.pem.key** as private_key.pem.key and the file xxxxx-**public.pem.key** as public_key.pem.key. After that, you will have the following:

Setting up policies

The policies are needed for allowing incoming data into AWS.

Go back to the AWS IoT HuB, open the "Secure" tab and click on "Policies".

Do the following procedure: Click on "Create policy" → Give it a name → Policy effect: allow → Policy action: * → Policy resource: *→ Create.

NOTE: The examples in this document are intended only for dev environments. All devices in your production fleet must have credentials with privileges that authorize only intended actions on specific resources. The specific permission policies can vary for your use case. Identify the permission policies that best meet your business and security requirements. For more information, refer to Example policies and Security Best practices.

Then, on the same "Secure" tab, click on certificates. There you will see one certificate, the one you've created, select it by checking the checkbox, then go to "Actions", and click on "Attach policy".

Then select the policy you've created previously, and click on "Attach policies"

Getting your endpoint

The endpoint is the host address of your MQTT Broker, where Publishers and Subscribers will connect to.

You can find it into the "Settings" tab, copy and save it.

With that, your MQTT Broker is all setup, and you can start setting up your RUT router as a Publisher.

RUT as MQTT Publisher

What data do you want to publish?

Depending on your RUT model, you can have the following options of data sources to send over MQTT:

- Bluetooth
- Wifi scanner
- MODBUS

Publishing Bluetooth/Wifi scanner/Modbus data

Before sending data from those sources to the MQTT Broker, the router needs to know what data from each source to send:

- For Bluetooth, first you have to pair the device you want to get data from
- For WiFi scanner data, you have to enable the wifi scanner functionality first
- For MODBUS data, you need to set what data from what MODBUS slave the router has to get

If you need any help on setting up each functionality, the Teltonika Wiki has topics for each one of them: Bluetooth, WiFi Scanner, MODBUS

Then, you will use the "Data to server" functionality, under "Services" menu.

Click on the "Add" button on the right side of the page, you will see the following menu.

The main fields you have to fill up are pretty straight forward, just pay more attention for the fields needed for AWS MQTT Broker connection:

"Server address" : There you will paste your AWS Endpoint

"Port" : 8883

"Topic" : Any name you want, just write it down so you can subscribe this topic later

"Enable Secure Connection" : On

"TLS type" : "Certificate based"

"CA File, Client certificate, Private key" : There you will select the files you've downloaded from AWS IoT HuB

To check if your setup is working, you can use any MQTT client, and subscribe to the topic you've created, you should be able to see the data of the source you've selected. You can use the AWS IoT MQTT test client to subscribe to the topic that the router was publishing.

Example: Publishing RUT MODBUS to AWS IoT using MQTT

In this example the RUT device will act as MODBUS TCP Master and MODBUS TCP Slave, so the device will make requests (Master) and answer to himself (Slave). The received reply, will be sent over MQTT. You can also send data from another MODBUS Slave devices connected to the router.

Enabling MODBUS TCP Slave

Enabling the MODBUS Slave option on the router allows it to answer any requests coming from a MODBUS Master. To do that, go to the router configuration page→Services→MODBUS→MODBUS TCP Slave. Then clock the "Enable" slider and save.

Enabling MODBUS TCP Master

Enabling the MODBUS Master option on the router allows it to make specific requests to any slave in the MODBUS network. To do that, go to the router configuration page→Services→MODBUS→MODBUS TCP Master. Click on the "Add" button, and do the following configuration on the page:

"Name" : Any name

"Slave ID" : The slave ID you've set on the slave configuration, by default its 1

"IP address" : The LAN IP address of the router, by default its 192.168.1.1

"Port" : 502

"Period" : 10

"Timeout" : 5

Then scroll down the page a bit, on the "Add new request" section, give any name to your request and click on the "Add" button. Then do the following configuration

"Data type" : Data type of the data you are going to receive, in this case, the router reports its device name using ASCII

"Function" : MODBUS Protocol function, in this case, we are going to Read holding registers

"First register number" : Depends on your MODBUS device, in this case, its 8

"Register count / Values" : Depends on your MODBUS device, in this case, its 10

Save and apply settings.

Configuring Data to Server

Go to the router configuration page→Services→Data to server. Click on the "Add" button, and the configuration is basically the same as described on the "Publishing Bluetooth/Wifi scanner/Modbus data" section of this article, just change the data source to "MODBUS data" and format the data as you wish. You should have something similar to this.

Checking if it works

Then, you can use the AWS MQTT test client to check if your setup works, if everything was setup correctly, you should see something like this.

Debugging

In the situation when the issue with services Bluetooth, WiFi Scanner, MODBUS, Data to Server appears, device internal logs can be taken directly from WebUI troubleshoot section. Also, there is a lot of useful information in frequently asked questions page.

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 Support Forum dedicated for troubleshooting, where engineers are actively solving problems.