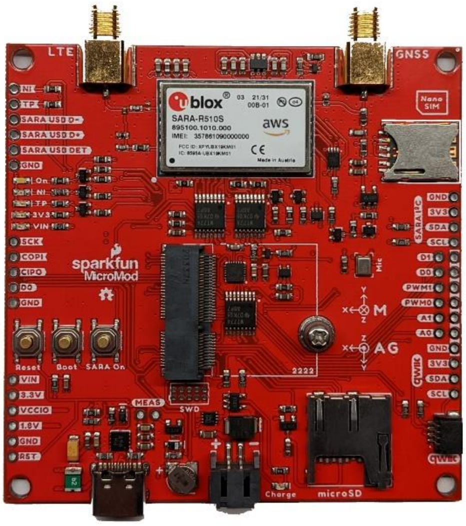
# AWS IoT ExpressLink SARA-R5 Starter Kit

## Getting started guide for u-blox AWS IoT ExpressLink SARA-R5 Starter Kit

Getting started



## Abstract

This document describes how to begin working with the u-blox AWS IoT ExpressLink SARA-R5 Starter Kit, which consists of a u-blox AWS IoT ExpressLink SARA-R5 module mounted on a SparkFun carrier board such that it can be used with a variety of microcontrollers or an attached PC. The u-blox AWS IoT ExpressLink SARA-R5 module provides communications over a cellular network anywhere that LTE-M (also known as LTE Cat-M1) service is available and presents a simple-to-use AT interface designed to allow even resource-constrained microcontrollers to connect to AWS IoT Core.

# Document information

| | | |
|---|---|---|
| **Title** | **AWS IoT ExpressLink SARA-R5 Starter Kit** | |
| **Subtitle** | Getting started guide for u-blox AWS IoT ExpressLink SARA-R5 Starter Kit | |
| **Document type** | Getting started | |
| **Document number** | UBX-21042016 | |
| **Revision and date** | R02 | 1-Dec-2021 |
| **Disclosure restriction** | C1-Public | |

This document applies to the following products:

| Product name |
|---|
| u-blox AWS IoT ExpressLink SARA-R5 Starter Kit |

u-blox or third parties may hold intellectual property rights in the products, names, logos and designs included in this document. Copying, reproduction, modification or disclosure to third parties of this document or any part thereof is only permitted with the express written permission of u-blox.

The information contained herein is provided “as is” and u-blox assumes no liability for its use. No warranty, either express or implied, is given, including but not limited to, with respect to the accuracy, correctness, reliability and fitness for a particular purpose of the information. This document may be revised by u-blox at any time without notice. For the most recent documents, visit www.u-blox.com.

# Contents

# 1 Overview

The u-blox AWS IoT ExpressLink SARA-R5 Starter Kit provides access to AWS IoT Core and AWS IoT services over a cellular LTE-M (also known as LTE Cat M1) connection. The kit includes a u-blox AWS IoT ExpressLink SARA-R5 module mounted on a SparkFun MicroMod Asset Tracker Carrier Board into which a Sparkfun MicroMod processor board can be inserted, allowing the module to be used with many different microprocessors.

⚠ **VERY IMPORTANT**: you MUST complete the setup procedures of sections 3, 4, and 5 to configure the endpoint connecting to your development account. The process used in this prototype version is for the evaluation phase only and will be simplified in the commercial release.

⚠ IMPORTANT: this starter kit is for evaluation purposes only; it is NOT suitable for production use. The module included in the kit has not been provisioned for commercial use and supports limited features. Please see section 8 for the supported features.

⚠ PLEASE BE PATIENT: cellular is not like Wi-Fi, many radio bands must be searched, more than one network might need to be tried, all before a connection to the AWS service can even be attempted. Please be patient, response times can be long, especially for commands such as **AT+CONNECT**, **AT+SEND** and **AT+DISCONNECT**.

# 2 Hardware description

This evaluation kit is based upon the standard SparkFun MicroMod Asset Tracker Carrier Board but has the following critical differences:

- the SARA-R5 module mounted on the board is a SARA-R510S, not a SARA-R510M8S, and hence the board (a) does NOT include GNSS functionality (and so is not provided with a GNSS antenna) and (b) does NOT power on automatically (see section 8 for the module power-on procedure),
- the SARA-R5 module mounted on the board is loaded with **AWS IoT ExpressLink firmware** and hence offers the AWS IoT ExpressLink AT interface and NOT the usual u-blox AT interface,
- because of the above, the example code provided by SparkFun does not apply; instead see the AWS IoT ExpressLink example code here: https://github.com/aws/iot-expresslink,
- the kit includes a Thingstream SIM card,
- the `SIG_INT` (G5/bus 5/pin 73) signal of the MicroMod connector is connected to the `GPIO3` pin of the SARA-R5 module.

## 2.1 Data sheet

The product description for the u-blox AWS IoT ExpressLink SARA-R5 Starter Kit can be found on the product page of the Sparkfun website.

Documentation for the u-blox SARA-R5 cellular module can be found on the SARA-R5 pages of the u-blox web site. All the information found there applies to the u-blox AWS IoT ExpressLink SARA-R5 module **except** the AT command manual; the AWS IoT ExpressLink programmers manual replaces the SARA-R5 AT command manual.

The hardware (pins, voltages, timings, power-on procedure etc.) of the u-blox SARA-R5 module and the u-blox AWS IoT ExpressLink SARA-R5 module is the same, with the following exceptions:

- the UART flow control lines, `RTS` and `CTS`, are NOT used,
- the `EVENT` pin, as defined by AWS IoT ExpressLink, is pin `GPIO3` of the AWS IoT ExpressLink SARA-R5 module and is available on the `SARA_INT` (G5/bus 5/pin 73) signal of the MicroMod connector on the board,
- the baud-rate of the UART is fixed at 115200.

⚠ This means that the power-on procedure for the u-blox AWS IoT ExpressLink SARA-R5 module is NOT the AWS one, it is the original u-blox SARA-R5 one and there is no `WAKE` pin, see section 8.

## 2.2 Standard kit contents

This kit includes:

- the board with the SARA-R5 module,
- a cellular antenna,
- a Thingstream SIM card, which will have already been inserted into the board,
- an Asset Tracker Update Tool M.2 board.

This is sufficient to allow you to use the u-blox AWS IoT ExpressLink SARA-R5 module from an attached PC via the USB-C interface.

☞ The SIM card includes 250 Mbytes of data transfer with the cellular network and after that must be discarded.

Not covered in this getting started guide, but also included in this kit, is an ESP32 MicroMod board that can be plugged into the M.2 socket on the board, allowing you to experiment with driving the module from a microcontroller; please refer to the ESP32 documentation for how to develop software using the ESP32 chip.

## 2.3 User provided items

To use the board via an attached PC, and to power it from USB, a USB-C cable is required.

## 2.4 Third party purchasable items

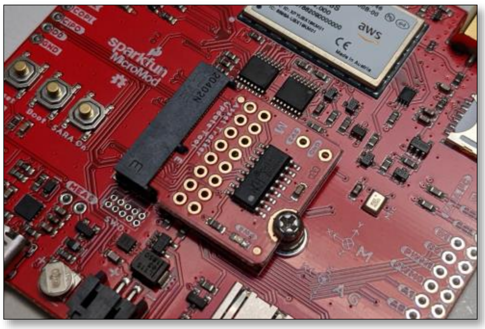
A variety of other SparkFun MicroMod processor boards, each carrying a different microcontroller (e.g. nRF5, STM32F4, Raspberry Pi Pico, etc.) are available to plug into the M.2 socket on the carrier board to use the module directly.

# 3 Set up your hardware

Set up your hardware following the hook-up guide for the standard SparkFun MicroMod Asset Tracker Carrier Board, noting the differences described at the start of section 2.

## 3.1 Set up host machine

To drive the u-blox AWS IoT ExpressLink SARA-R5 Starter Kit from an attached host machine, which is ESSENTIAL during setup, you must plug the provided Asset Tracker Update Tool into the M.2 socket on the carrier board.



On the host machine you must install the drivers for the CH340 USB to UART bridge.

Open a terminal application on your host machine (e.g., TeraTerm or Docklight for Windows or CoolTerm for Mac) and select the port corresponding to the evaluation kit. Configure the terminal application as follows – and take special note of the first three items:

| | |
|---|---|
| • End of line: | line feed (i.e. 0x0a, *not* 0x0d) |
| • Local echo: | yes |
| • Flow control: | none |
| • Baud rate: | 115200 |
| • Bits: | 8 |
| • Parity: | none |
| • Stop: | 1 |

⚠ u-blox m-center and tools such as PuTTY will NOT work with the u-blox AWS IoT ExpressLink SARA-R5 module since those tools use carriage return as a line ending when sending to the module, not line feed.

For a quick check, turn the cellular module on by pressing the “SARA On” button on the carrier board for around one second. Your terminal window should show the welcome message **Welcome to u-blox AWS IoT ExpressLink SARA-R5** from the cellular module[1]. In the terminal window type **AT** and press <return>. If you receive the answer **OK**, congratulations! You have successfully connected the evaluation kit to your host machine.

Keep the terminal application open and connected, as it is needed for the subsequent steps.

# 4 Setup your AWS account and permissions for IoT development

Refer to the instructions at Set up your AWS Account. Follow the steps outlined at that link to:

- sign up for an AWS account,
- create a user and grant permissions,
- open the AWS IoT console.

Pay special attention to the notes.

[1] One or two unprintable characters may appear before the welcome message; these are part of the module boot sequence and can be ignored.

# 5 Registering AWS IoT ExpressLink SARA-R5 with your development account

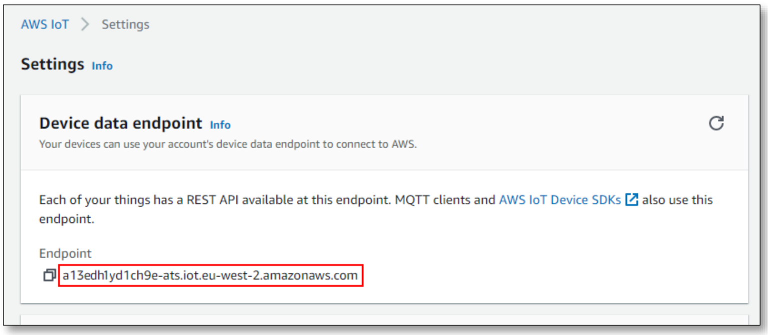
## 5.1 Setting up the module

Before a cellular module can be registered with your development account for the first time, it must have registered with the cellular network for the first time. Ensure that:

- the provided Thingstream SIM is inserted in the board,
- a cellular antenna is connected to the antenna connector marked "LTE",
- you are in range of a cellular network that supports LTE-M,
- the board is connected to a host machine using the USB-C connection,
- a serial terminal application configured as specified in section 3.1 is running on the host machine.

Then a "dummy" CONNECT is performed as in the following steps:

1. In the AWS IoT Console, choose **Settings** (bottom left of the navigation bar) and copy your account "Endpoint" string from "Device data endpoint".



AWS IoT > Settings

Settings Info

Device data endpoint Info

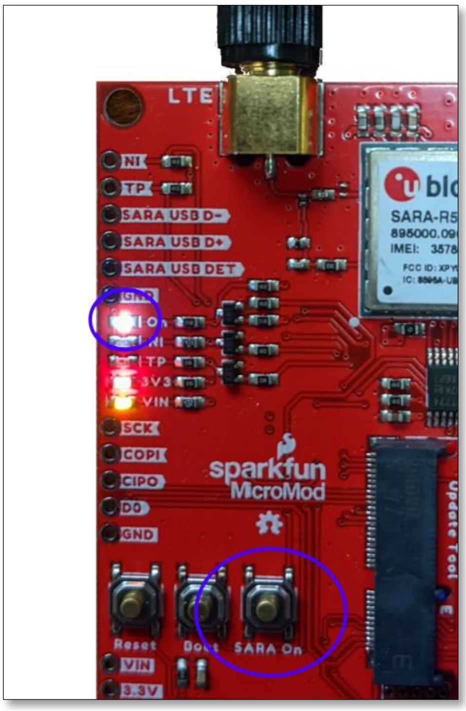
Your devices can use your account's device data endpoint to connect to AWS.

Each of your things has a REST API available at this endpoint. MQTT clients and AWS IoT Device SDKs also use this endpoint.

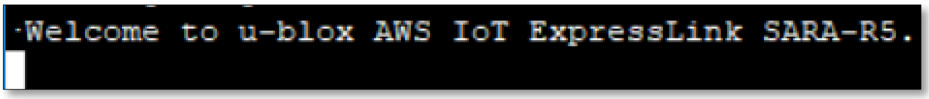
Endpoint

a13edh1yd1ch9e-ats.iot.eu-west-2.amazonaws.com

2. Press the "SARA On" button on the carrier board for around one second to power the cellular module on; the white "On" LED should light.



After a short while, in your terminal application the text **Welcome to u-blox AWS IoT ExpressLink SARA-R5** should appear.

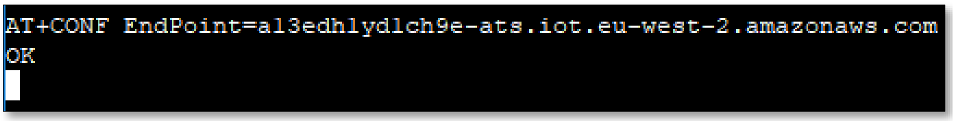


```
Welcome to u-blox AWS IoT ExpressLink SARA-R5.
```

3. In the terminal application type the command AT+CONF EndPoint=<your endpoint string from step 1 here> and press <return>.

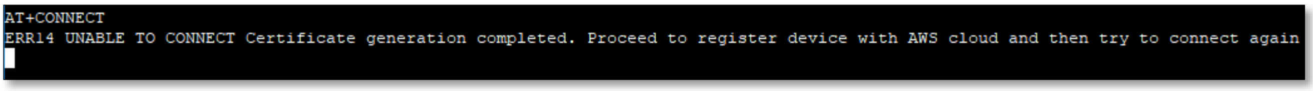
⚠ The configuration parameter name **EndPoint** is case sensitive.

You should get **OK** back.



```
AT+CONF EndPoint=a13edhlydlch9e-ats.iot.eu-west-2.amazonaws.com
OK
```

4. In the terminal application, type the command **AT+CONNECT** and press <return>. This will take a long time and should fail. The ****exact expected**** response is **ERR14 UNABLE TO CONNECT Certificate generation completed. Proceed to register device with AWS cloud and then try to connect again**.



```
AT+CONNECT
ERR14 UNABLE TO CONNECT Certificate generation completed. Proceed to register device with AWS cloud and then try to connect again
```

☞ On this very first cellular connection the module may take some considerable time to find LTE-M network service; many minutes, up to 15 minutes. This is because the device does not yet know the radio environment and must search all of the global LTE-M RF bands on each visible cellular network. Connections should be quicker on subsequent attempts as the module will remember how it gained service previously.

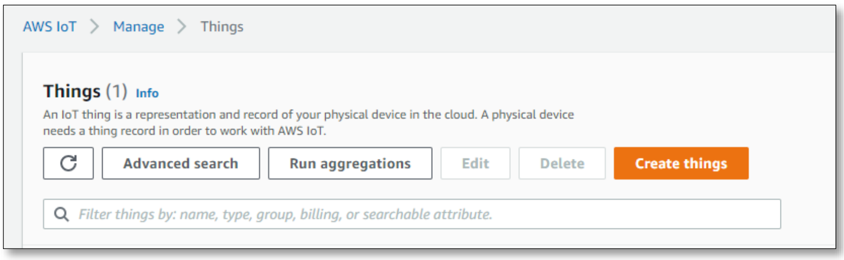
⚠ If you get an error response that is different than the one shown above (e.g., just **ERR14 UNABLE TO CONNECT**), then the cellular connection has not been successful, and you should try again. See section 7 for troubleshooting advice.

With these steps completed, the cellular module has in fact successfully connected to the cellular network for the first time and registered with the necessary u-blox servers to create the certificates you will need to register it with your AWS account in the next steps (after which you should get **OK 1 CONNECTED** back to your **AT+CONNECT** attempts). Please proceed to section 5.2.

## 5.2 Registering with your AWS account

To create an IoT Thing and add it to your account, you will need to retrieve the u-blox AWS IoT ExpressLink SARA-R5 module **ThingName** and its corresponding certificate. Please follow the steps below.

1. Open the AWS IoT Console. Select **Manage** then select **Things**.



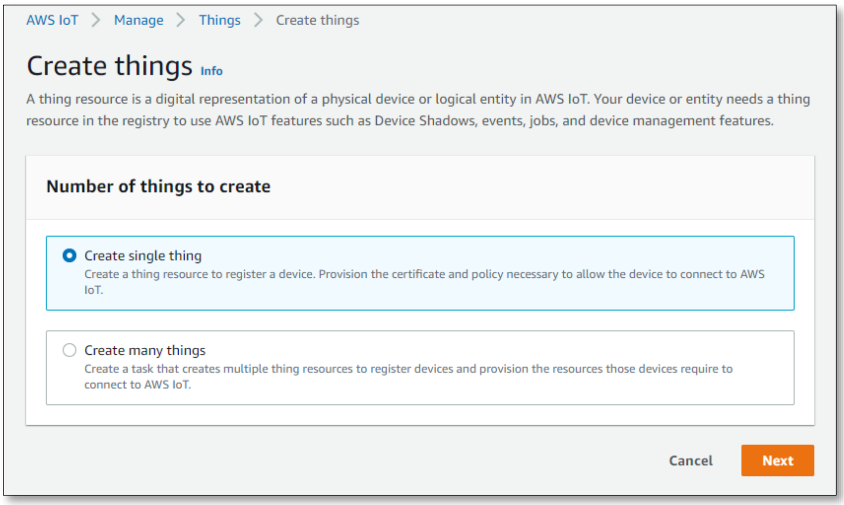
AWS IoT > Manage > Things

Things (1) Info

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

Advanced search
Run aggregations
Edit
Delete
Create things
Filter things by: name, type, group, billing, or searchable attribute.

2. Click Create things, select Create single thing and click Next.



Create things Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

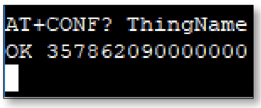
**Number of things to create**

● Create single thing
Create a thing resource to register a device. Provision the certificate and policy necessary to allow the device to connect to AWS IoT.

○ Create many things
Create a task that creates multiple thing resources to register devices and provision the resources those devices require to connect to AWS IoT.

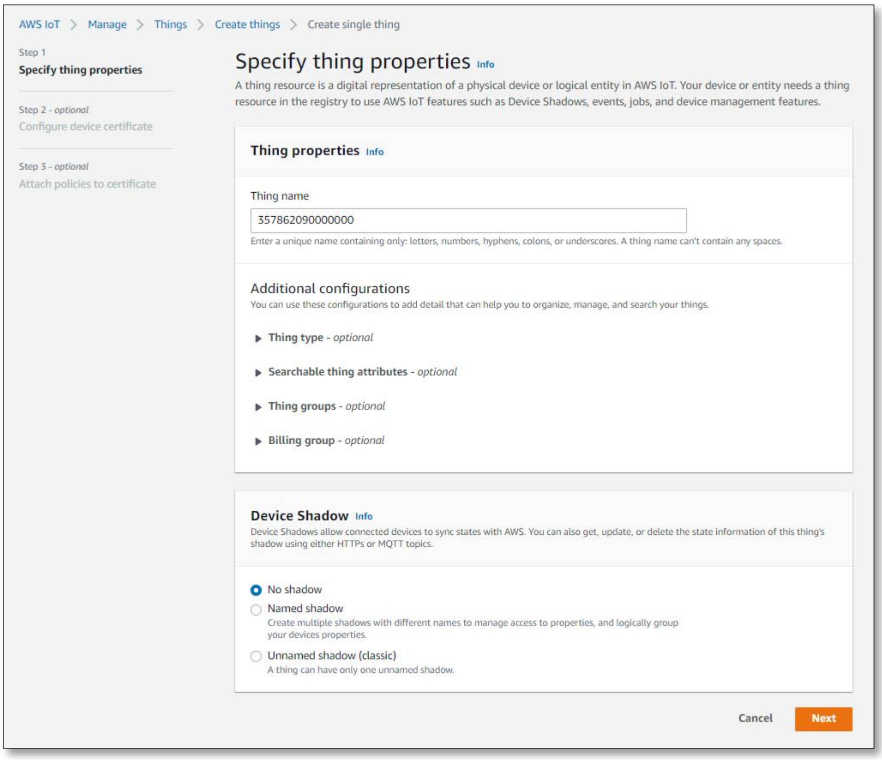
3. With the u-blox AWS IoT ExpressLink SARA-R5 module still switched on, In the terminal application type the command **AT+CONF? ThingName** and press <return>.

⚠ The configuration parameter name **ThingName** is case sensitive.



```
AT+CONF? ThingName
OK 357862090000000
```

4. Copy the sequence of numeric characters (i.e., the equivalent of "357862090000000" in the image above) from the terminal. On the **Specify thing properties** page, paste the copied string from the terminal into the **Thing name** under **Thing properties** in the console. Leave the other fields as default, then click **Next**.



AWS IoT > Manage > Things > Create things > Create single thing
Step 1
Specify thing properties
Step 2 - optional
Configure device certificate
Step 3 - optional
Attach policies to certificate

Specify thing properties Info

A thing resource is a digital representation of a physical device or logical entity in AWS IoT. Your device or entity needs a thing resource in the registry to use AWS IoT features such as Device Shadows, events, jobs, and device management features.

**Thing properties** Info

Thing name: 357862090000000

Enter a unique name containing only: letters, numbers, hyphens, colons, or underscores. A thing name can't contain any spaces.

Additional configurations

You can use these configurations to add detail that can help you to organize, manage, and search your things.

▶ **Thing type** - *optional*

▶ **Searchable thing attributes** - *optional*

▶ **Thing groups** - *optional*

▶ **Billing group** - *optional*

**Device Shadow** Info

Device Shadows allow connected devices to sync states with AWS. You can also get, update, or delete the state information of this thing's shadow using either HTTPs or MQTT topics.

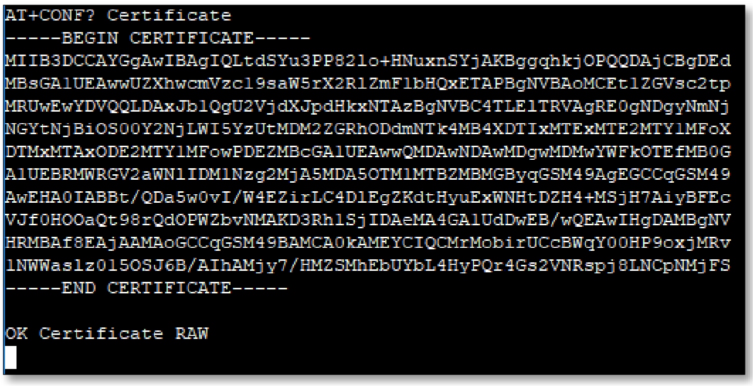
● No shadow

○ Named shadow
Create multiple shadows with different names to manage access to properties, and logically group your devices properties.

○ Unnamed shadow (classic)
A thing can have only one unnamed shadow.

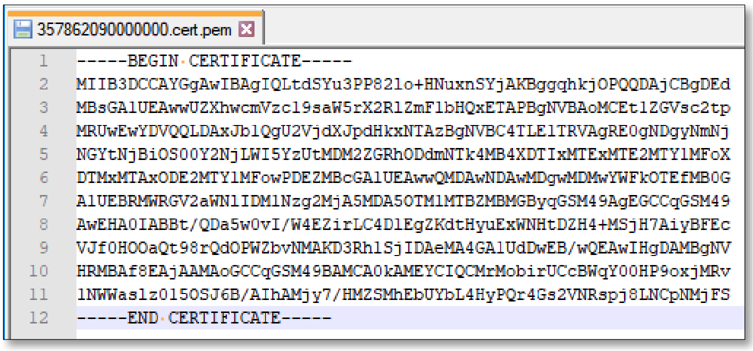
5. In the terminal application type the command **AT+CONF? Certificate** and press <return>.

⚠ You must use the name **Certificate** and NOT the more recently introduced **Certificate pem**, which is not supported/necessary in this release.



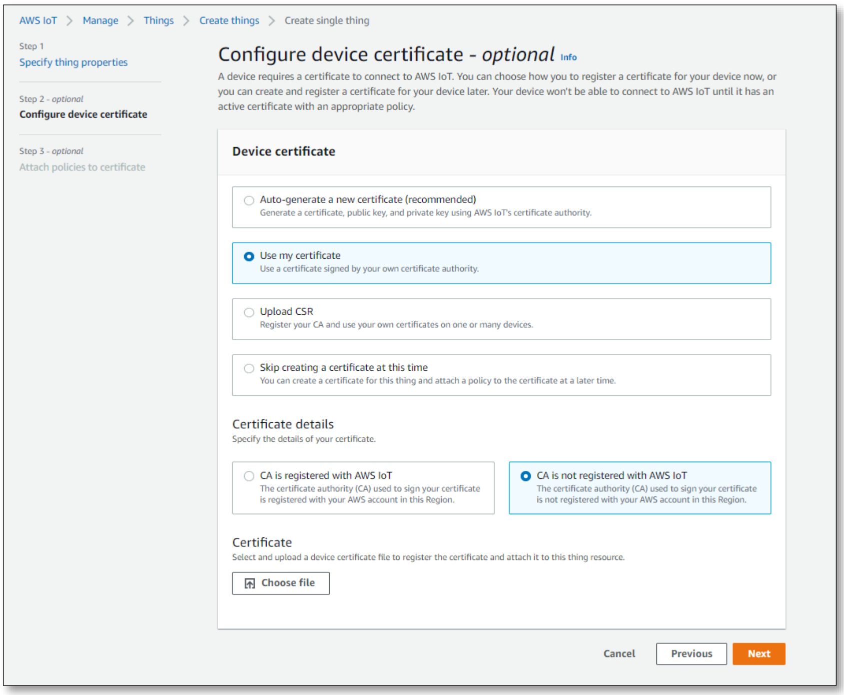
```
AT+CONF? Certificate
-----BEGIN CERTIFICATE-----
MIIB3DCCAYGgAwIBAgIQLtdSYu3PP82lo+HNuxnSYjAKBggqhkjOPQQDAjCBgDEd
MBsGA1UEAwwUZXhwcmVzcl9saW5rX2R1ZmF1bHQxETAPBgNVBAoMCEt1ZGVsc2tp
MRUwEwYDVQQLDAxJb1QgU2VjdXJpdHkxNTAzBgNVBC4TLE1TRVAgRE0gNDgyNmNj
NGYtNjBiOS00Y2NjLWI5YzUtMDM2ZGRhODdmNTk4MB4XDTIxMTExMTE2MTY1MFoX
DTMxMTAxODE2MTY1MFowPDEZMBcGA1UEAwwQMDAwNDAwMDgwMDMwYWFkOTEfMB0G
A1UEBRMWRGV2aWN1IDM1Nzg2MjA5MDA5OTM1MTBZMBMGByqGSM49AgEGCCqGSM49
AwEHA0IABBt/QDa5w0vI/W4EZirLC4D1EgZKdtHyuExWNHtDZH4+MSjH7AiyBFEc
VJf0HOOaQt98rQdOPWZbvNMAKD3Rh1SjIDAeMA4GA1UdDwEB/wQEAwIHgDAMBgNV
HRMBAf8EAjAAMAoGCCqGSM49BAMCA0kAMEYCIQCMrMobirUCcBWqY00HP9oxjMRv
1NWWas1z015OSJ6B/AIhAMjy7/HMZSMhEbUYbL4HyPQr4Gs2VNRspj8LNCpNMjFS
-----END CERTIFICATE-----

OK Certificate RAW
```

6. Copy the returned string, including **-----BEGIN CERTIFICATE-----** and **-----END CERTIFICATE-----** but not including the line beginning with **OK**, into a text file on your host machine named "ThingName.cert.pem".



```
-----BEGIN CERTIFICATE-----
MIIB3DCCAYGgAwIBAgIQLtdSYu3PP82lo+HNuxnSYjAKBggqhkjOPQQDAjCBgDEd
MBsGA1UEAwwUZXhwcmVzcl9saW5rX2R1ZmF1bHQxETAPBgNVBAoMCEt1ZGVsc2tp
MRUwEwYDVQQLDAxJb1QgU2VjdXJpdHkxNTAzBgNVBC4TLE1TRVAgRE0gNDgyNmNj
NGYtNjBiOS00Y2NjLWI5YzUtMDM2ZGRhODdmNTk4MB4XDTIxMTExMTE2MTY1MFoX
DTMxMTAxODE2MTY1MFowPDEZMBcGA1UEAwwQMDAwNDAwMDgwMDMwYWFkOTEfMB0G
A1UEBRMWRGV2aWN1IDM1Nzg2MjA5MDA5OTM1MTBZMBMGByqGSM49AgEGCCqGSM49
AwEHA0IABBt/QDa5w0vI/W4EZirLC4D1EgZKdtHyuExWNHtDZH4+MSjH7AiyBFEc
VJf0HOOaQt98rQdOPWZbvNMAKD3Rh1SjIDAeMA4GA1UdDwEB/wQEAwIHgDAMBgNV
HRMBAf8EAjAAMAoGCCqGSM49BAMCA0kAMEYCIQCMrMobirUCcBWqY00HP9oxjMRv
1NWWas1z015OSJ6B/AIhAMjy7/HMZSMhEbUYbL4HyPQr4Gs2VNRspj8LNCpNMjFS
-----END CERTIFICATE-----
```

7. On the Configure device certificate page, select Use my certificate and choose CA is not registered with AWS IoT.



AWS IoT > Manage > Things > Create things > Create single thing
Step 1
Specify thing properties
Step 2 - optional
Configure device certificate
Step 3 - optional
Attach policies to certificate

Configure device certificate - *optional* Info

A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

○ Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

● Use my certificate
Use a certificate signed by your own certificate authority.

○ Upload CSR
Register your CA and use your own certificates on one or many devices.

○ Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

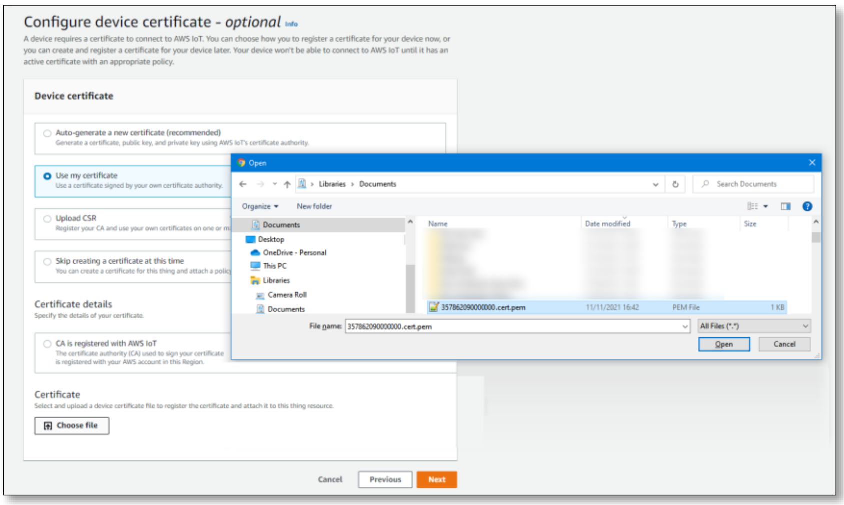
Certificate details
Specify the details of your certificate.

○ CA is registered with AWS IoT
The certificate authority (CA) used to sign your certificate is registered with your AWS account in this Region.

● CA is not registered with AWS IoT
The certificate authority (CA) used to sign your certificate is not registered with your AWS account in this Region.

Certificate
Select and upload a device certificate file to register the certificate and attach it to this thing resource.

8. Under **Certificate**, select **Choose file** and upload the file “ThingName.cert.pem” from step 6.



Configure device certificate - *optional* Info

A device requires a certificate to connect to AWS IoT. You can choose how you to register a certificate for your device now, or you can create and register a certificate for your device later. Your device won't be able to connect to AWS IoT until it has an active certificate with an appropriate policy.

Device certificate

○ Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

● Use my certificate
Use a certificate signed by your own certificate authority.

○ Upload CSR
Register your CA and use your own certificates on one or m

○ Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy

Certificate details
Specify the details of your certificate.

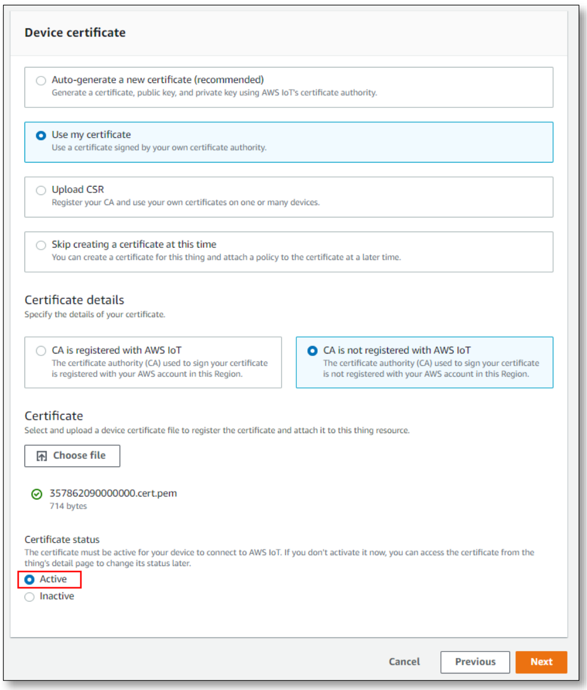
○ CA is registered with AWS IoT
The certificate authority (CA) used to sign your certificate is registered with your AWS account in this Region.

Certificate
Select and upload a device certificate file to register the certificate and attach it to this thing resource.

357862090000000.cert.pem 11/11/2021 16:42 PEM File 1 KB

File name: 357862090000000.cert.pem

9. Under Certificate status, select Active and click Next.



Device certificate

○ Auto-generate a new certificate (recommended)
Generate a certificate, public key, and private key using AWS IoT's certificate authority.

● Use my certificate
Use a certificate signed by your own certificate authority.

○ Upload CSR
Register your CA and use your own certificates on one or many devices.

○ Skip creating a certificate at this time
You can create a certificate for this thing and attach a policy to the certificate at a later time.

Certificate details
Specify the details of your certificate.

○ CA is registered with AWS IoT
The certificate authority (CA) used to sign your certificate is registered with your AWS account in this Region.

● CA is not registered with AWS IoT
The certificate authority (CA) used to sign your certificate is not registered with your AWS account in this Region.

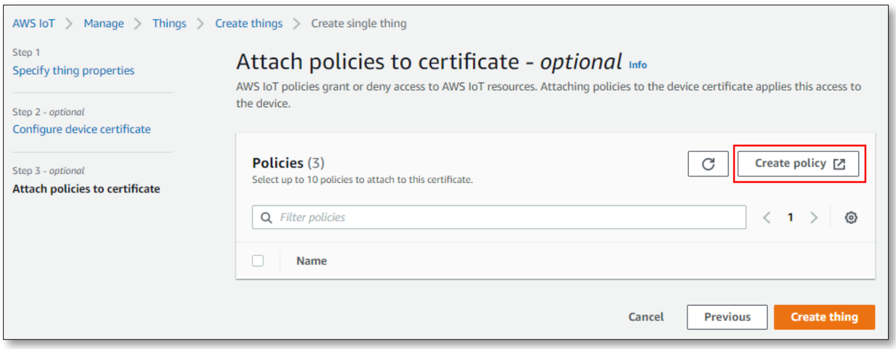
Certificate
Select and upload a device certificate file to register the certificate and attach it to this thing resource.

✓ 357862090000000.cert.pem
714 bytes

Certificate status
The certificate must be active for your device to connect to AWS IoT. If you don't activate it now, you can access the certificate from the thing's detail page to change its status later.

● Active
○ Inactive

10. Click **Create policy** and a new browser tab will open:



AWS IoT > Manage > Things > Create things > Create single thing

Step 1
Specify thing properties

Step 2 - *optional*
Configure device certificate

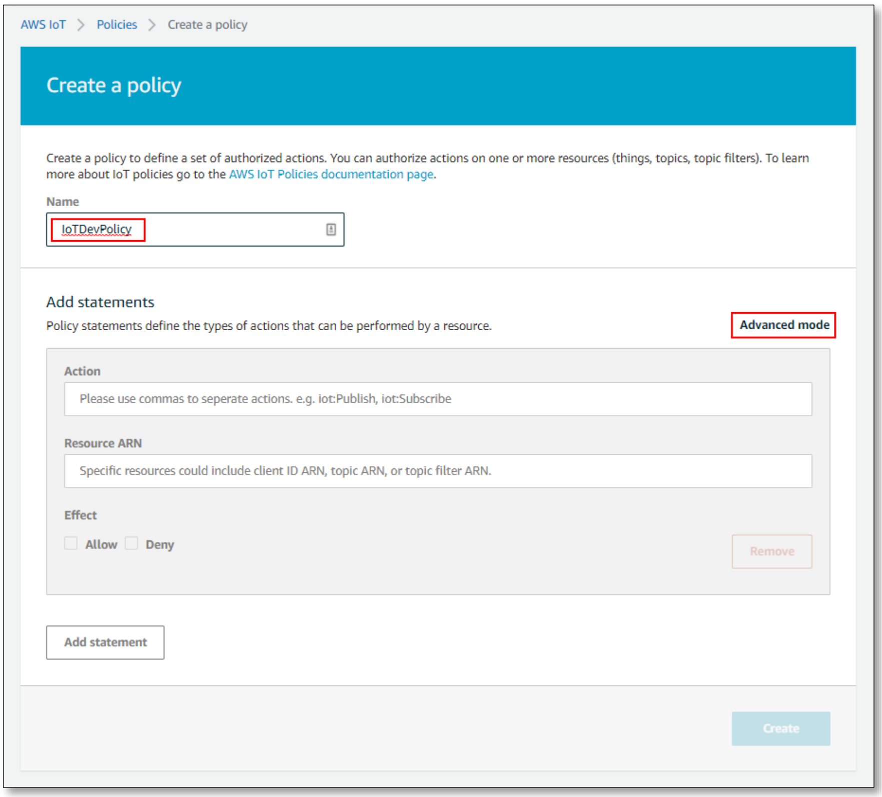
Step 3 - *optional*
Attach policies to certificate

Attach policies to certificate - *optional* Info

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

Policies (3)
Select up to 10 policies to attach to this certificate.

11. Type in a policy name of your choosing (e.g., "IoTDevPolicy") and click **Advanced mode**.



AWS IoT > Policies > Create a policy

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: IoTDevPolicy

Add statements

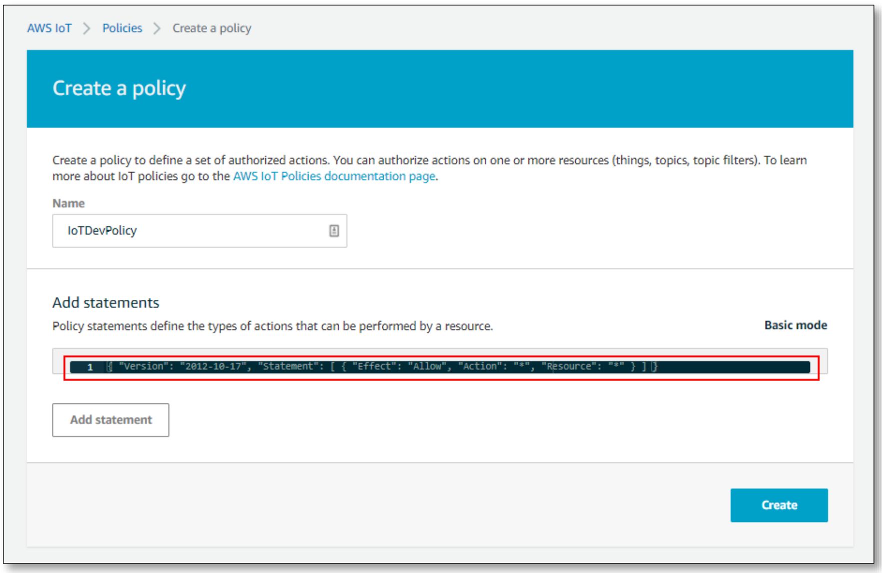
Policy statements define the types of actions that can be performed by a resource.

Advanced mode
Action
Please use commas to seperate actions. e.g. iot:Publish, iot:Subscribe
Resource ARN
Specific resources could include client ID ARN, topic ARN, or topic filter ARN.
Effect
□ Allow □ Deny
Remove
Add statement
Create

12. Replace the contents of the black console area that appears with the single line of text below:

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "*", "Resource": "*" } ] }
```

...then click **Create** to create the policy.



AWS IoT > Policies > Create a policy

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page.

Name: IoTDevPolicy

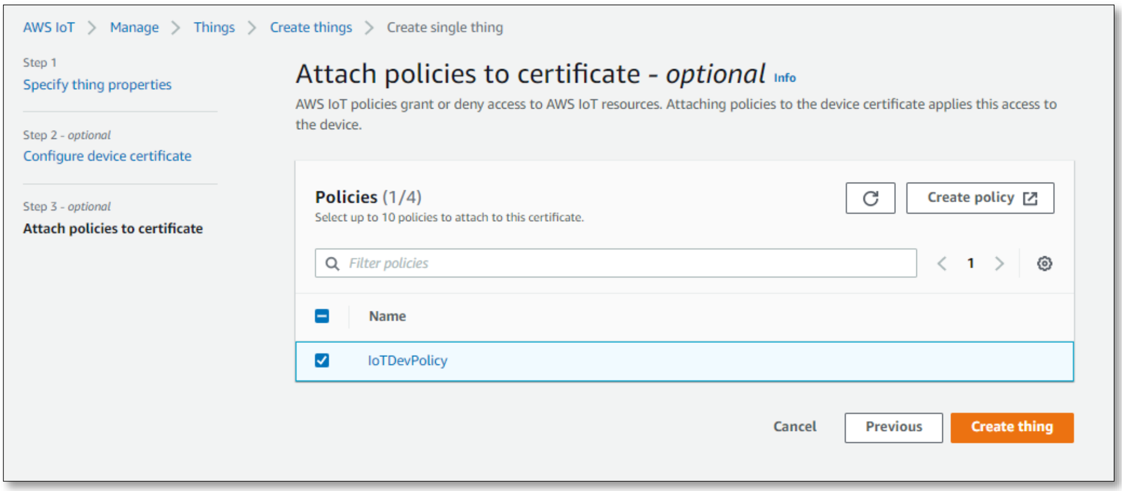
Add statements

Policy statements define the types of actions that can be performed by a resource.

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "*", "Resource": "*" } ] }
```

☞ Note: the examples in this document are intended only for development environments. All devices in your 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 IoT Policies and Security Best Practices.

13. Return to the original browser tab, select the policy you just created (you can close the policy creation tab now) and click **Create thing** to complete your Thing creation.


AWS IoT > Manage > Things > Create things > Create single thing
Step 1
Specify thing properties
Step 2 - optional
Configure device certificate
Step 3 - optional
Attach policies to certificate

Attach policies to certificate - *optional* Info

AWS IoT policies grant or deny access to AWS IoT resources. Attaching policies to the device certificate applies this access to the device.

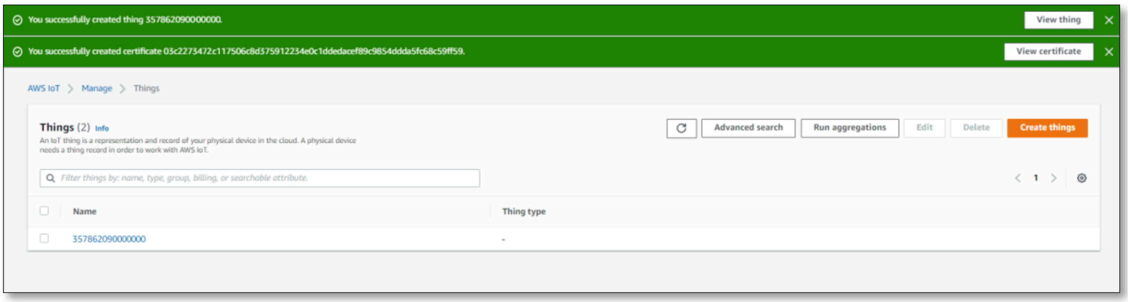
**Policies** (1/4)

Select up to 10 policies to attach to this certificate.

| ☐ | Name |
|---|---|
| ☑ | IoTDevPolicy |

Create policy
Filter policies
Cancel
Previous
Create thing

14. Congratulations! You have successfully created a new Thing in your AWS account.



You successfully created thing 357862090000000.

You successfully created certificate 03c2273472c117506c8d375912234e0c1ddedacef89c9854ddda5fc68c59ff59.

View thing
View certificate
AWS IoT > Manage > Things

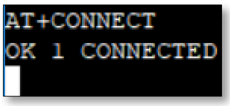
**Things** (2) Info

An IoT thing is a representation and record of your physical device in the cloud. A physical device needs a thing record in order to work with AWS IoT.

Advanced search
Run aggregations
Edit
Delete
Create things
Filter things by: name, type, group, billing, or searchable attribute.

| ☐ | Name | Thing type |
|---|---|---|
| ☐ | 357862090000000 | - |

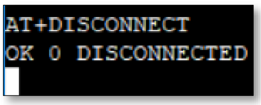
15. Return to the terminal application, type the command **AT+CONNECT** once more and press <return>. After a short while you should receive the response **OK 1 CONNECTED**.



```
AT+CONNECT
OK 1 CONNECTED
```

Congratulations! You have successfully connected to your AWS cloud account.

16. To close the connection, type the command **AT+DISCONNECT** in the terminal application and press <return>. After a short while you should receive the response **OK 0 DISCONNECTED**.



```
AT+DISCONNECT
OK 0 DISCONNECTED
```

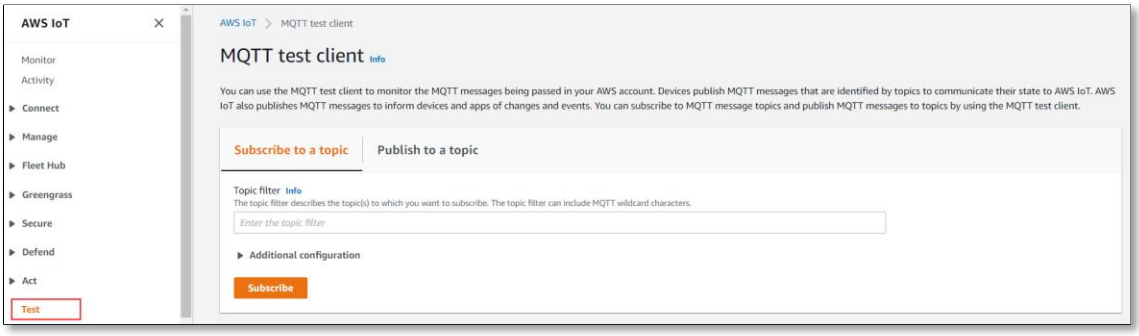
## 5.3 Completion

Congratulations! You have completed the registration of the evaluation kit as a Thing in your IoT account. You will not need to repeat these steps the next time you connect, as the u-blox AWS IoT ExpressLink SARA-R5 module will remember its configuration and will be ready to connect to your AWS account automatically.

# 6 Connecting to and interacting with the AWS cloud

We will use the MQTT client in the AWS IoT console to help us monitor the communication between your evaluation kit and the AWS Cloud.

1. Navigate to the AWS IoT console and select **Test** from the navigation pane to open the **MQTT test client**.



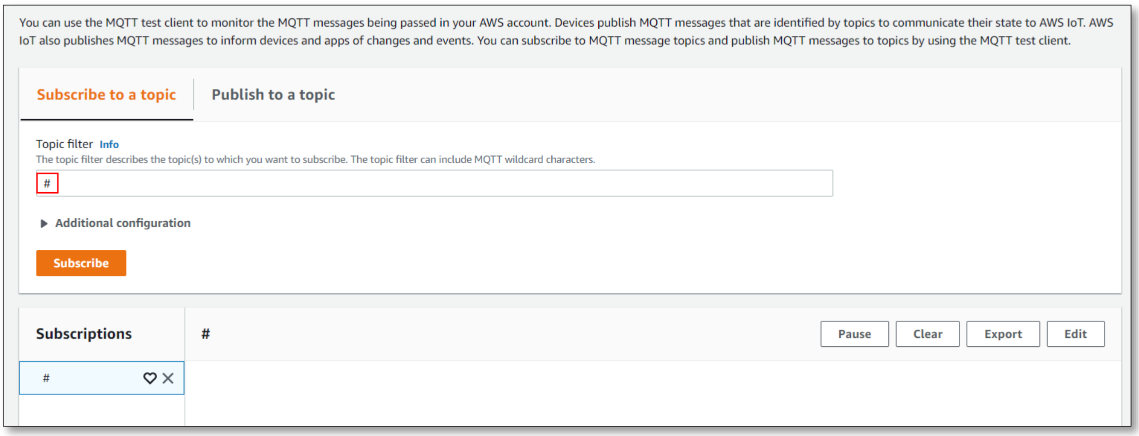
MQTT test client Info

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Topic filter Info

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

2. In Subscribe to a topic, enter # as the Topic filter and then click Subscribe.



You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Topic filter Info

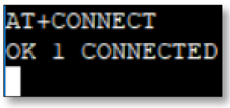
The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

Topic filter: #

## 6.1 Connecting

With the u-blox AWS IoT ExpressLink SARA-R5 module switched on, in the terminal application, establish a secure connection by typing the command **AT+CONNECT** and pressing <return>.

After a while, you should receive the response **OK 1 CONNECTED**.



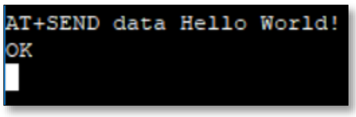
```
AT+CONNECT
OK 1 CONNECTED
```

Some cellular networks will refuse service to a device for a "guard time" (e.g., 1 hour) if the device is seen to connect and disconnect [again] more than a handful of times in quick succession; frequent connections/disconnections should be avoided.

## 6.2 Sending data to the AWS cloud

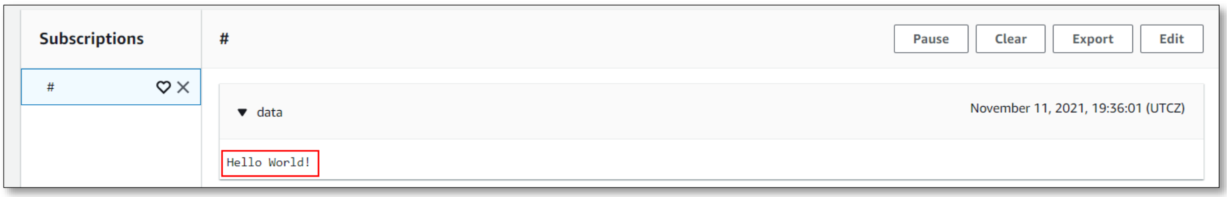
To send a "Hello World!" message, in the terminal application type the command **AT+SEND data Hello World!** and press <return>.

After a short while, you should receive the response **OK**.



```
AT+SEND data Hello World!
OK
```

You should see the **Hello World!** message appear on the AWS IoT console under the topic **data**.



▼ data

November 11, 2021, 19:36:01 (UTCZ)

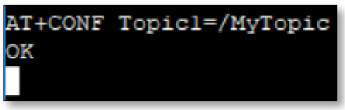
```
Hello World!
```

☞ In poor coverage conditions there may be up to a 30 second delay between the message arriving in the AWS IoT console and **OK** being returned in the terminal application.

## 6.3 Receiving data and commands from the AWS cloud

To receive messages on your device, in the terminal application type the command **AT+CONF Topic1=/MyTopic** and press <return>.

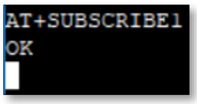
You should receive the response **OK**.



```
AT+CONF Topic1=/MyTopic
OK
```

⚠ The **AT+CONF TopicX** command is NOT persistent across power-cycles of the device.

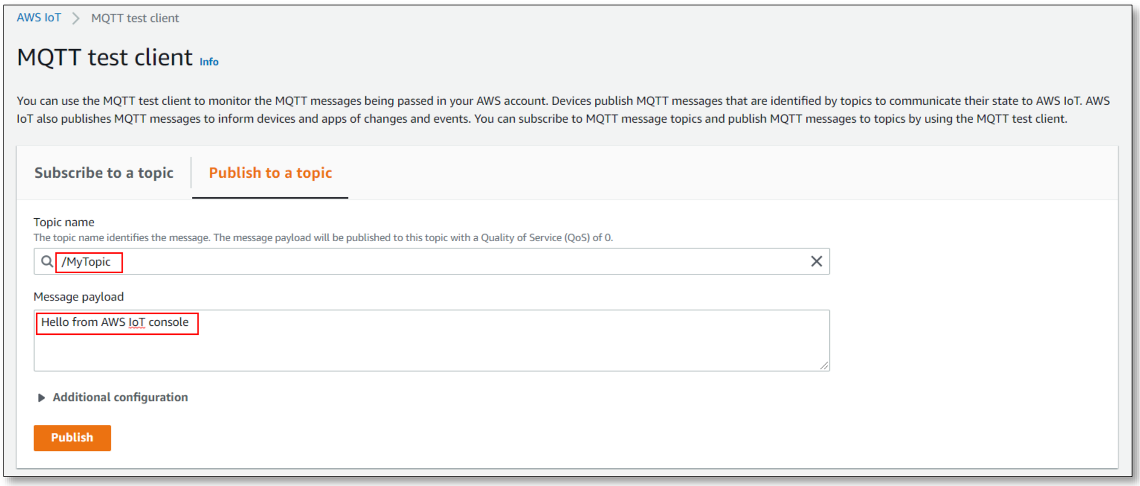
Next, type the command **AT+SUBSCRIBE1** and press <return>; you should receive the response **OK**.



```
AT+SUBSCRIBE1
OK
```

⚠ You must re-subscribe after an **AT+DISCONNECT** or after powering-down the module.

From the AWS IoT console **MQTT test client**, select **Publish to a topic**, type **/MyTopic** into the **Topic name** field (replacing the **#** that was there), under **Message payload** enter **Hello from the AWS IoT console** and then click **Publish**.



MQTT test client

You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

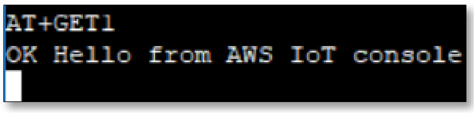
/MyTopic

Message payload

Hello from AWS IoT console

Wait for a handful of seconds for the message to arrive at the device (or monitor the `EVENT` pin) and then, in the terminal application, type the command **AT+GET1** and press <return>.

You should receive the response **OK Hello from AWS IoT console**.

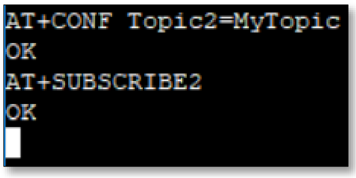


```
AT+GET1
OK Hello from AWS IoT console
```

## 6.4 Exchanging Thing-specific data and commands with the AWS cloud

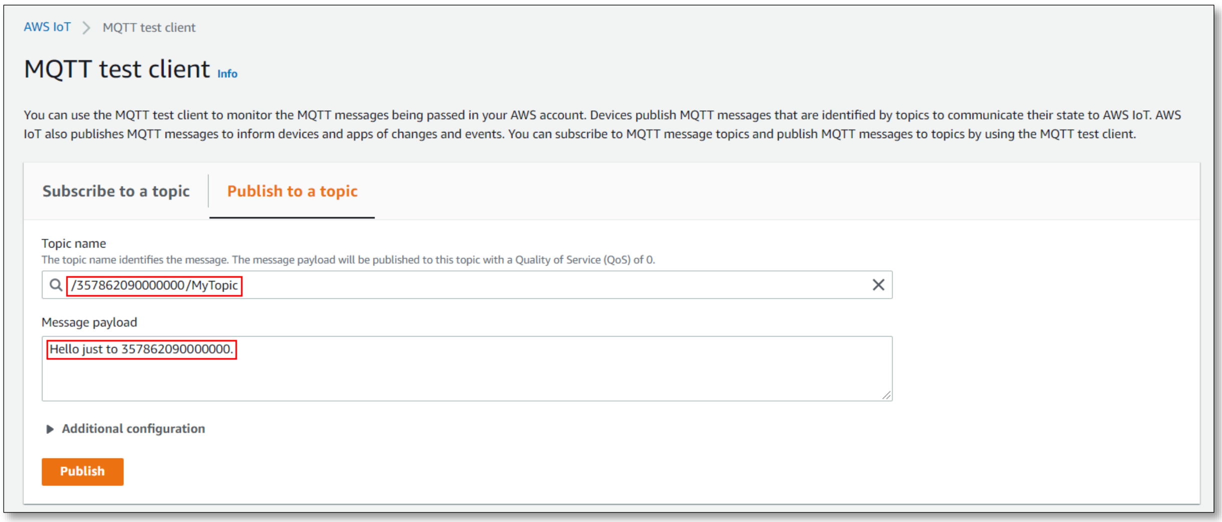
If you leave the leading **/** off the topic name then the **TopicRoot**, which defaults to the **ThingName**, will automatically be prepended to the subscription.

For instance, if you typed the command **AT+CONF Topic2=MyTopic** (i.e., leaving off the leading **/**) and pressed <return>, then subscribed to that topic by typing the command **AT+SUBSCRIBE2** and pressing <return>:



```
AT+CONF Topic2=MyTopic
OK
AT+SUBSCRIBE2
OK
```

…then in the AWS IoT console **MQTT test client**, under **Publish to a topic**, you typed **/ThingName/MyTopic** into the **Topic name** field and, under **Message payload**, entered **Hello just to ThingName** (or any message you like) and then clicked **Publish**:

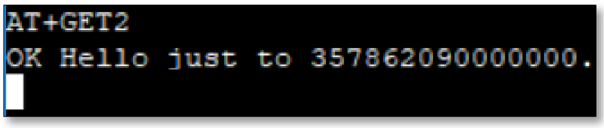


You can use the MQTT test client to monitor the MQTT messages being passed in your AWS account. Devices publish MQTT messages that are identified by topics to communicate their state to AWS IoT. AWS IoT also publishes MQTT messages to inform devices and apps of changes and events. You can subscribe to MQTT message topics and publish MQTT messages to topics by using the MQTT test client.

Topic name
The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.
/357862090000000/MyTopic

Message payload
Hello just to 357862090000000.

…and then a few seconds later, in the terminal application, you typed the command **AT+GET2** and pressed <return>, you should receive the response **OK Hello just to ThingName**.



```
AT+GET2
OK Hello just to 357862090000000.
```

You can also send to a **TopicX** using the command **AT+SENDx <message>**; if that **TopicX** does not start with a leading **/** then your entire send/get exchange will automatically occur beneath the unique **ThingName** of this device.

# 7 Troubleshooting

| Problem | Action |
|---|---|
| No lights come on. | Ensure power is supplied to the board, e.g., via the USB-C connector. |
| No welcome message is displayed when an AT terminal application is attached to the board. | Check that the white LED marked “On” is illuminated; the cellular module does NOT come on automatically when the board is powered, you must press the “SARA On” button for around one second to power the module on.<br>Also make sure that the baud rate of your terminal application is set to 115200. |
| I don’t see anything in the terminal application when I type in it. | The AWS IoT ExpressLink specification does not include echoing of received characters, which is normal for a “machine”-oriented interface; when typing in commands manually, please set “local echo” on in your terminal application so that you can see what you’re doing. |
| I can’t use backspace or delete or the cursor keys to edit my AT commands when typing in the terminal application. | The AWS IoT ExpressLink specification does not support use of editing keys; it is meant to be used by a microcontroller, another machine, which would have no use for editing keys. If you find this irritating when typing in commands manually, then you should find a terminal application that holds the entire command string locally, in the application, allowing you to edit it, and only sends the command string once <return> is pressed. |
| No response is received to any AT commands typed in the terminal application. | Make sure that the baud rate of your terminal application is set to 115200 and that AT commands are formed exactly as defined in the AWS IoT ExpressLink programmers manual: the line ending must be line feed (0x0a); carriage return (0x0d) and carriage return followed by line feed (0x0d 0a) cannot be used. |
| The AT command **CONNECT** does not work. | Make sure that the supplied SIM card is inserted in the board, an antenna is connected to the antenna socket on the board (the one marked “LTE”, *not* the GNSS one) and that you are within range of a cellular network that provides LTE-M coverage.<br>Make sure that you have followed the registration procedure of section 5, including registering your device in your AWS account; the connection is to the endpoint of the device in that account (not just the cellular network) and so it cannot succeed until all of the registration steps have been completed.<br>If **CONNECT** has worked in the past but is now failing in the same location and after using the device for some while, it may be that the cellular network has decided to refuse service for a “guard time” (e.g. 1 hour) because the device has connected and disconnected [again] more than a handful of times in quick succession; unfortunately there is nothing that can be done about this (other than avoiding many connections/disconnections), it is a network “feature”. |
| The AT command **SEND** returns **OK** even when I have disconnected the cellular antenna and so I am plainly not in coverage. | Messages are queued until it is possible to send them; if you reconnect the cellular antenna, you should find that the messages will be successfully published. If you do not do so, or you disconnect or reset the device, the queued messages will be lost. |
| Setting a topic with the AT command **CONF TopicX** (where **X** is 1 to 16) returns an error. | While a topic is subscribed-to it cannot be modified; please use the AT command **UNSUBSCRIBEX** to unsubscribe from topic **X** before modifying it. |
| The AT commands **CONFMODE**, **OTA**, **HOTA** and **WHERE** do not work. | These AT commands are not supported in this preview release. |
| Some parameters to the **CONF** command do not work. | Only parameters that are relevant to a cellular device, and are not required only by OTA or HOTA, are supported; see section 8 for a full list. |
| Issuing the **SLEEP** AT command does not reduce the power consumption of the board. | In this preview release the **SLEEP** command is accepted for compatibility purposes but does not actually put any hardware into a low power state. |
| The AT command **TIME?** does not work. | You must be connected to the cellular network to obtain the time. If the problem happens when you *are* connected, then the cellular network you are using does not support reporting of time to the mobile terminal (the feature is optional in the network). |
| Some downlink messages from the server are going missing. | Make sure that no downlink message exceeds 2048 bytes in size; in this release any downlink message larger than 2048 bytes in size will be silently discarded. |
| Messages larger than around 512 bytes in size cannot be sent. | The maximum AT command length in this release is 512 bytes and hence this limits the maximum length of MQTT message that can be sent. |

# 8 Features supported

The firmware provided on the u-blox AWS IoT ExpressLink SARA-R5 module in this evaluation kit is a preview release and hence it does NOT support all the features of the AWS IoT ExpressLink command set. The following commands are supported:

**CONNECT**, **DISCONNECT**, **SLEEP**[2], **RESET**, **FACTORY_RESET**, **SEND**[3], **GET**[4], **SUBSCRIBE**, **UNSUBSCRIBE**, **CONF**, **EVENT**, **DIAG**, **TIME**[5].

For the avoidance of doubt, the following commands are not currently supported:

**CONFMODE**, **OTA**, **HOTA**, **WHERE**.

☞ OTA of the module is supported but only through u-blox uFOTA since it must work within the constraints applied by the cellular network operators.

For the **CONF** command, only the following parameters are supported:

**About**, **Version**, **ThingName**, **Certificate** (*without* **pem** after it), **EndPoint**, **APN**, **TopicRoot**, **TopicX** (where X is 1 to 16, i.e., MaxTopic is 16).

☞ The power-on behavior of the module is defined by the underlying SARA-R5 HW, i.e. the module does not power on until the `PWR_ON` pin is asserted for 1 second and then released, see the u-blox SARA-R5 data sheet for more details; the pin MUST be released as if it is held for more than 16 seconds an emergency restart of the module will be performed. On the SparkFun MicroMod Asset Tracker board a "SARA On" button is provided which will assert the `PWR_ON` pin while pressed.

☞ The `PWR_ON` pin is a toggle, i.e., if it is asserted for 1 second and then released while the module is powered on then the module will power off again. It is ALWAYS a good idea to power the module off in this way before physically removing power.

☞ From the perspective of the SARA-R5 module, "asserted" means "pulled low" but the SparkFun MicroMod Asset Tracker Carrier Board includes an inverter in the line and hence, from the perspective of an MCU on a MicroMod processor board, asserted means "held high".

☞ The behavior of the `WAKE` pin described in the AWS ExpressLink documentation is not supported.

☞ The behavior of the `RST` pin described in the AWS ExpressLink documentation is offered by the `RESET_N` pin of the module. On the SparkFun MicroMod Asset Tracker Board a "Reset" button is provided that will effect a reset of the module.

[2] While the command is supported the module does NOT currently enter a low power state, i.e. the command is only "emulated".

[3] The maximum AT command-line length in this release is 512 bytes, which will set a limit on the maximum size of an MQTT message being sent.

[4] The maximum receive message size is 2048 bytes.

[5] Must be connected to work and not all cellular networks support indication of time to the mobile terminal.

# Related documentation

[1] AWS IoT ExpressLink website
[2] AWS IoT ExpressLink SARA-R5 Starter Kit product page
[3] AWS IoT ExpressLink programmers manual
[4] AWS IoT ExpressLink example code
[5] u-blox SARA-R5 series data sheet, UBX-19016638
[6] u-blox SARA-R5 series system integration manual, UBX-19041356
[7] ESP32 development guide

☞ For product change notifications and regular updates of u-blox documentation, register on our website, www.u-blox.com.

# Revision history

| Revision | Date | Name | Comments |
|---|---|---|---|
| R01 | 30-Nov-2021 | rmea | Initial release. |
| R02 | 01-Dec-2021 | rmea | Correct picture in section 5.2 sub-bullet 9. Some minor style corrections. |

# Contact

For complete contact information, visit us at www.u-blox.com.

**u-blox Offices**

**North, Central and South America**

**u-blox America, Inc.**

Phone: +1 703 483 3180
Email: info_us@u-blox.com

**Regional Office West Coast:**

Phone: +1 408 573 3640
Email: info_us@u-blox.com

**Technical Support:**

Phone: +1 703 483 3185
Email: support_us@u-blox.com

**Headquarters**
**Europe, Middle East, Africa**

**u-blox AG**

Phone: +41 44 722 74 44
Email: info@u-blox.com
Support: support@u-blox.com

**Asia, Australia, Pacific**

**u-blox Singapore Pte. Ltd.**

Phone: +65 6734 3811
Email: info_ap@u-blox.com
Support: support_ap@u-blox.com

**Regional Office Australia:**

Phone: +61 3 9566 7255
Email: info_anz@u-blox.com
Support: support_ap@u-blox.com

**Regional Office China (Beijing):**

Phone: +86 10 68 133 545
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Chongqing):**

Phone: +86 23 6815 1588
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Shanghai):**

Phone: +86 21 6090 4832
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office China (Shenzhen):**

Phone: +86 755 8627 1083
Email: info_cn@u-blox.com
Support: support_cn@u-blox.com

**Regional Office India:**

Phone: +91 80 405 092 00
Email: info_in@u-blox.com
Support: support_in@u-blox.com

**Regional Office Japan (Osaka):**

Phone: +81 6 6941 3660
Email: info_jp@u-blox.com
Support: support_jp@u-blox.com

**Regional Office Japan (Tokyo):**

Phone: +81 3 5775 3850
Email: info_jp@u-blox.com
Support: support_jp@u-blox.com

**Regional Office Korea:**

Phone: +82 2 542 0861
Email: info_kr@u-blox.com
Support: support_kr@u-blox.com

**Regional Office Taiwan:**

Phone: +886 2 2657 1090
Email: info_tw@u-blox.com
Support: support_tw@u-blox.com