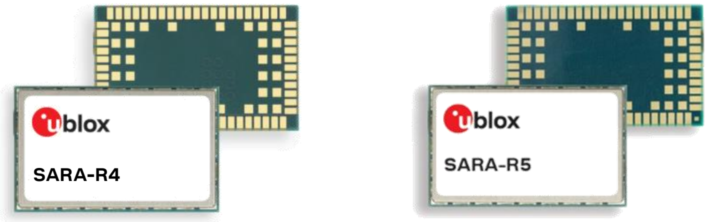
SARA-R422 / SARA-R5

Configure MQTT on AWS IoT core

Application note



Abstract

This document provides examples of how to use AT commands to connect the AWS IoT service with u-blox SARA-R422S, SARA-R422M8S, SARA-R422M10S and SARA-R5 series modules.

UBX-20044809 - R04
C1-Public

www.u-blox.com

Document information

Title	**SARA-R422 / SARA-R5**	
Subtitle	Configure MQTT on AWS IoT core	
Document type	Application note	
Document number	UBX-20044809	
Revision and date	R04	28-Sep-2022
Disclosure restriction	C1-Public	

This document applies to the following products:

Product name
SARA-R422S
SARA-R422M8S
SARA-R422M10S
SARA-R5 series

Contents

1 AWS IoT setup

To get started with the Amazon Web Services (AWS) IoT service, it is necessary to set up the AWS account and permissions. For details on how to create an AWS account, see the AWS official website link:

https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/

Detailed instructions are available in sections **Sign up for an AWS account** and **Create a user and grant permissions,** at https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html.

For more details on the use of AT commands, see the SARA-R5 series AT commands manual [2] / SARA-R4 series AT commands manual [6]. Further details on the IP data connection configuration are available in the SARA-R4 / SARA-R5 internet applications development guide [4].

⚠ Due to AWS's continuous evolution, some information provided in this document can be not up to date.

1.1 Policy creation

To set up an AWS connection, first create a policy. From the AWS IoT Core console at console.aws.amazon.com/iot, go to **Secure > Policies**, then click **Create**. This enables the creation of a new policy that will be adopted in future devices.

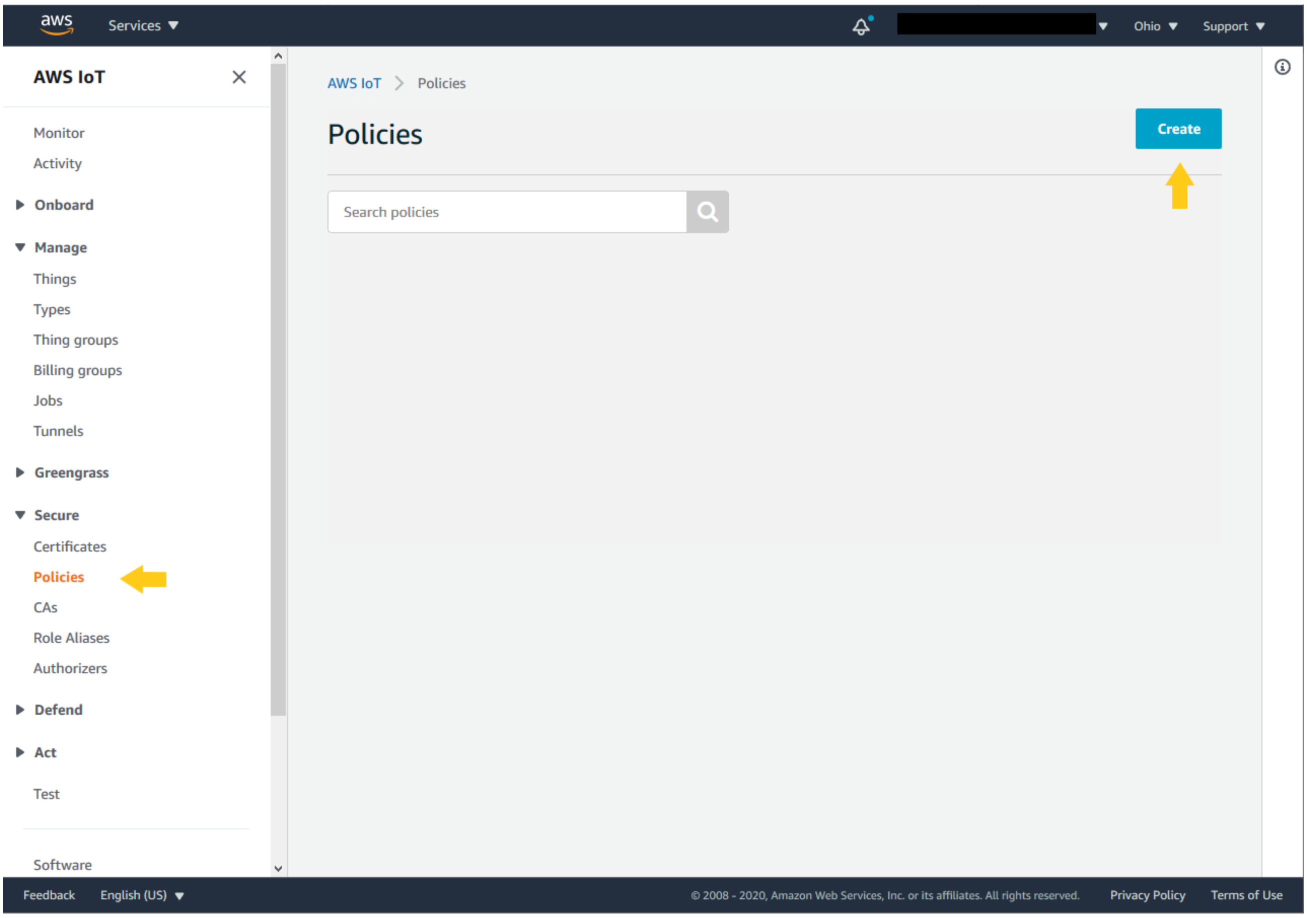
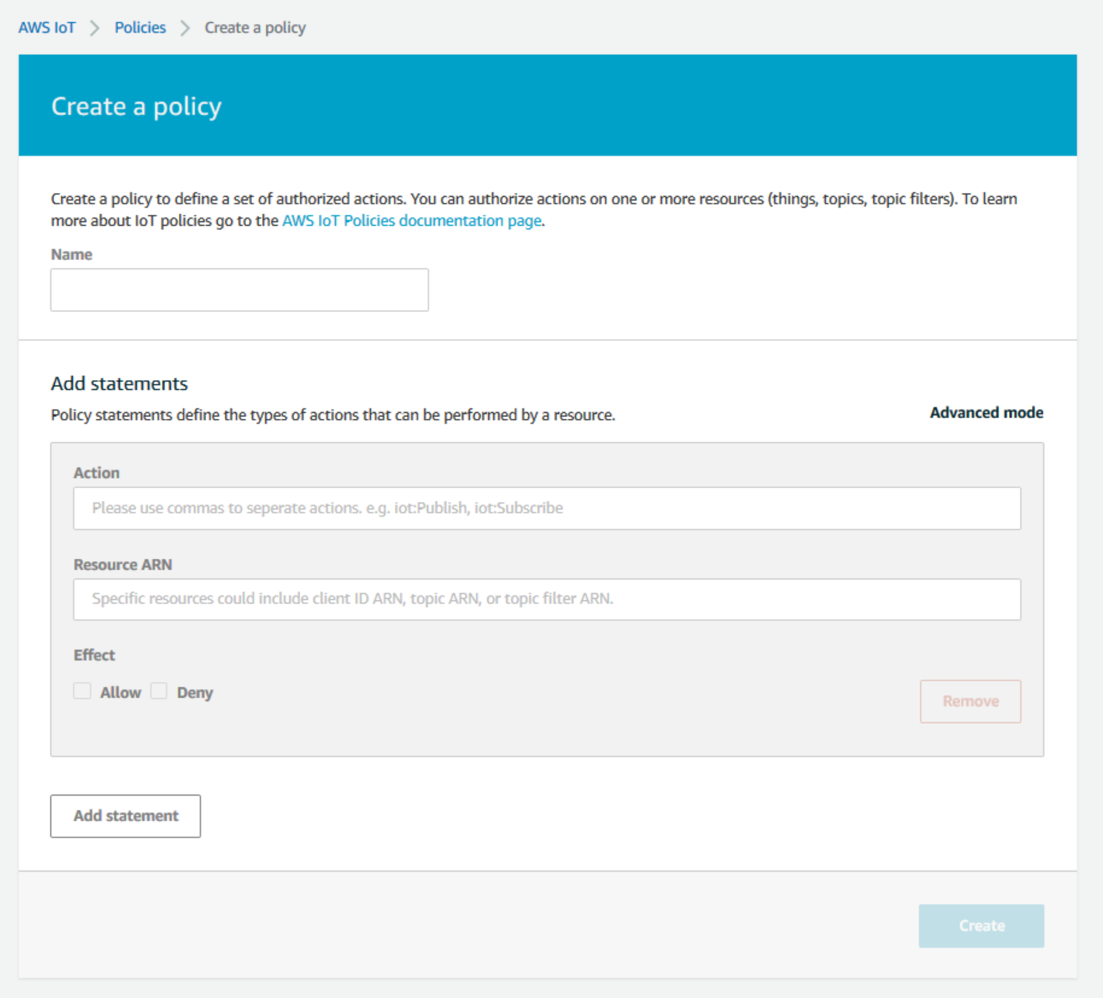


Figure 1: AWS policy creation

Then, on the next page, type a name for the new policy and type the required actions in the field **Action** (e.g., `iot:Publish, iot:Receive, iot:Subscribe`) considering the resource identification reported in the field **Resource ARN**.



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

Add statements

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

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

Figure 2: Create a policy

Remember to check the **Allow** box. Multiple statements can be added in the same policy. Complete the procedure by clicking **Create**. See a sample of a policy with two statements in the following script.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive",
        "iot:Subscribe"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:us-east-1:XXXXXXXX:*"
    }
  ]
}
```

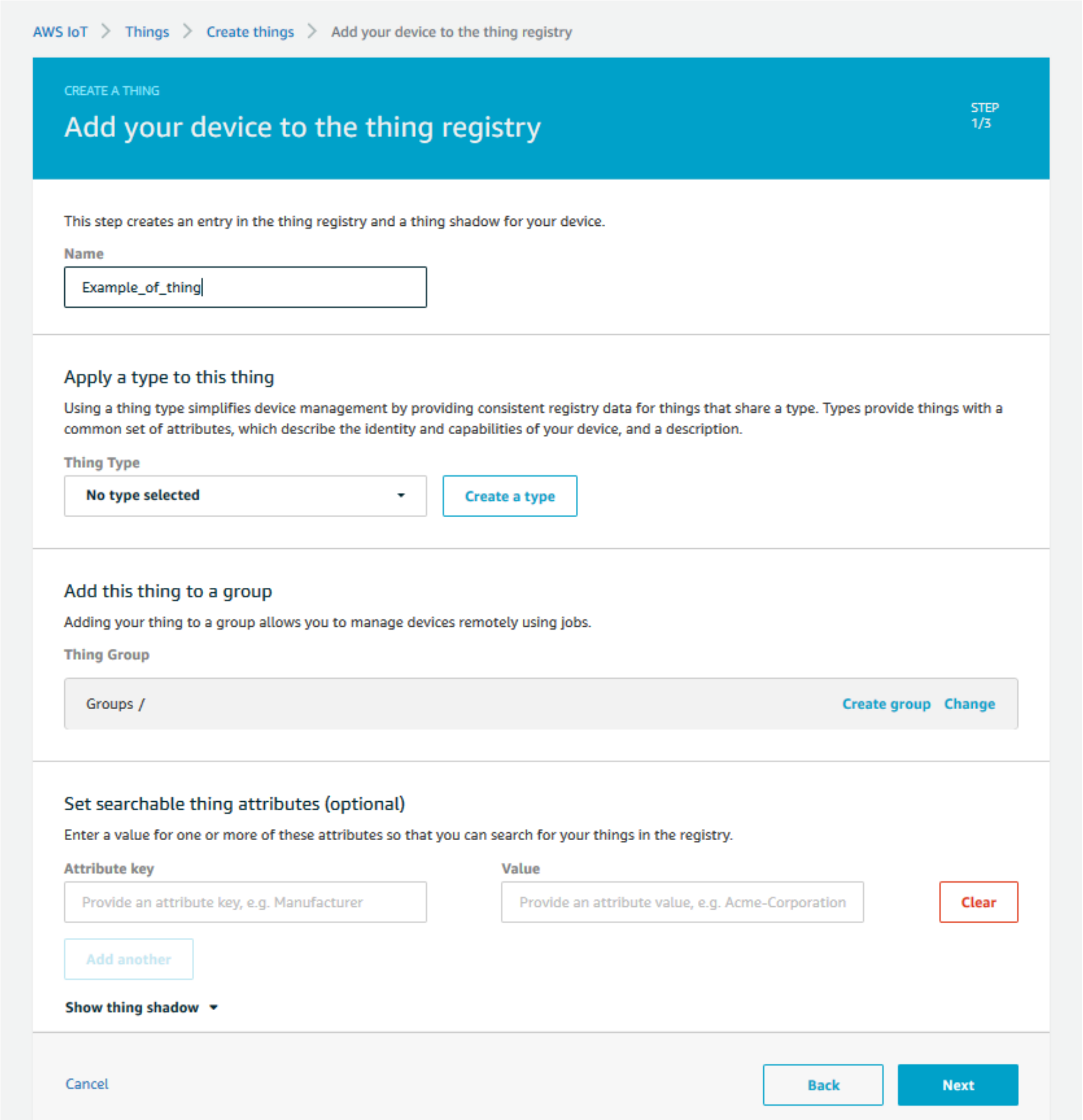
☞ For non-dev environments, all devices in your fleet must have credentials with privileges that authorize intended actions only, which include, but are not limited to, AWS IoT MQTT actions such as publishing messages or subscribing to topics with specific scope and context. The specific permission policies can vary for your use cases. Identify the permission policies that best meet your business and security requirements.

For additional sample policies, refer to:
https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html
https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html

1.2 Thing creation

As the next step, navigate to **Manage > Things** using the menu on the left-hand side of the AWS IoT Core console and select **Create** to initialize a new “thing”.

On the next page, select **Create a single thing** and proceed. Here, insert a thing name in the box. no further settings should be configured on this page. Complete the procedure by clicking **Next**. For clarity, see the example shown in the image below:



AWS IoT > Things > Create things > Add your device to the thing registry

CREATE A THING

Add your device to the thing registry

STEP 1/3

This step creates an entry in the thing registry and a thing shadow for your device.

Name: Example_of_thing

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type: No type selected

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group: Groups /

Set searchable thing attributes (optional)

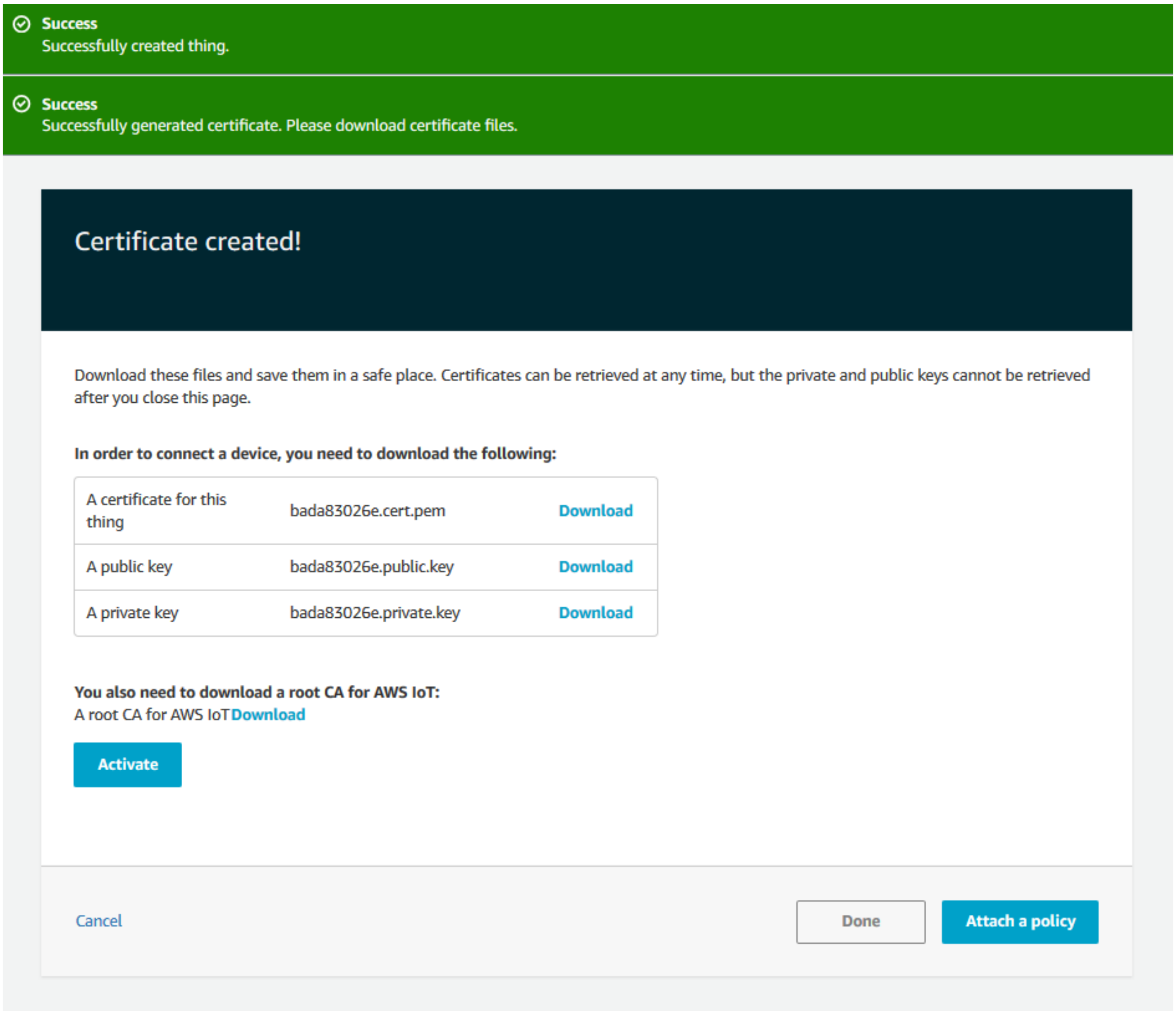
Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key: Provide an attribute key, e.g. Manufacturer

Value: Provide an attribute value, e.g. Acme-Corporation

Figure 3: Add your device to the thing registry

To create and download the necessary certificates, click **Create certificate**. Proceed to download of the certificate and the public and private keys that have been generated for this thing.



Success
Successfully created thing.

Success
Successfully generated certificate. Please download certificate files.

Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

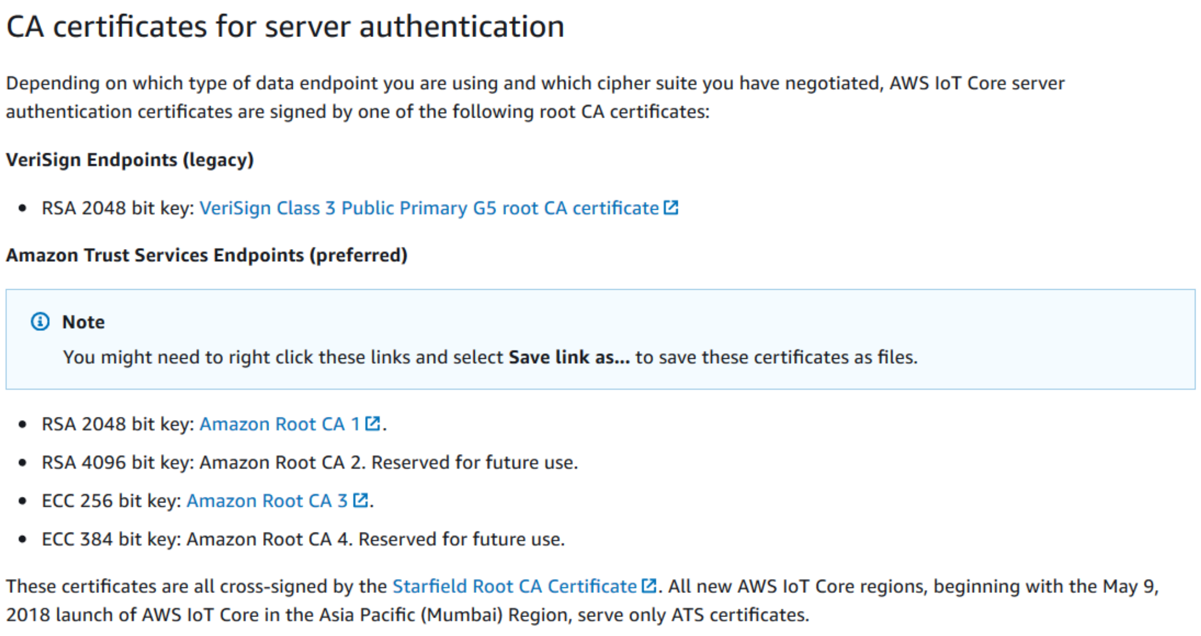
In order to connect a device, you need to download the following:

A certificate for this thing	bada83026e.cert.pem	Download
A public key	bada83026e.public.key	Download
A private key	bada83026e.private.key	Download

You also need to download a root CA for AWS IoT:
A root CA for AWS IoT Download

Figure 4: Certificate and keys creation

It is required to download a root CA certificate for AWS IoT, which is available in the dedicated link. A new page will be opened, as shown in Figure 5. Select and download an **RSA 2048 bit key: Amazon Root CA 1** certificate to complete the process.



CA certificates for server authentication

Depending on which type of data endpoint you are using and which cipher suite you have negotiated, AWS IoT Core server authentication certificates are signed by one of the following root CA certificates:

VeriSign Endpoints (legacy)

- RSA 2048 bit key: VeriSign Class 3 Public Primary G5 root CA certificate

Amazon Trust Services Endpoints (preferred)

> **Note**
>
> You might need to right click these links and select **Save link as...** to save these certificates as files.

- RSA 2048 bit key: Amazon Root CA 1.
- RSA 4096 bit key: Amazon Root CA 2. Reserved for future use.
- ECC 256 bit key: Amazon Root CA 3.
- ECC 384 bit key: Amazon Root CA 4. Reserved for future use.

These certificates are all cross-signed by the Starfield Root CA Certificate. All new AWS IoT Core regions, beginning with the May 9, 2018 launch of AWS IoT Core in the Asia Pacific (Mumbai) Region, serve only ATS certificates.

Figure 5: Download AWS Root CA

Once all the certificates and keys are downloaded, click **Activate** and finally click **Done**.

Section 2 shows how to use these certificates and keys with the module.

1.3 Attach policy to created thing

At this point, AWS will permit attaching a “policy” to the thing. This is the last necessary step to correctly use MQTT protocol and services. Thus, proceed by clicking: **Manage > Thing** in the left-hand navigation menu. Then click the thing you just created.

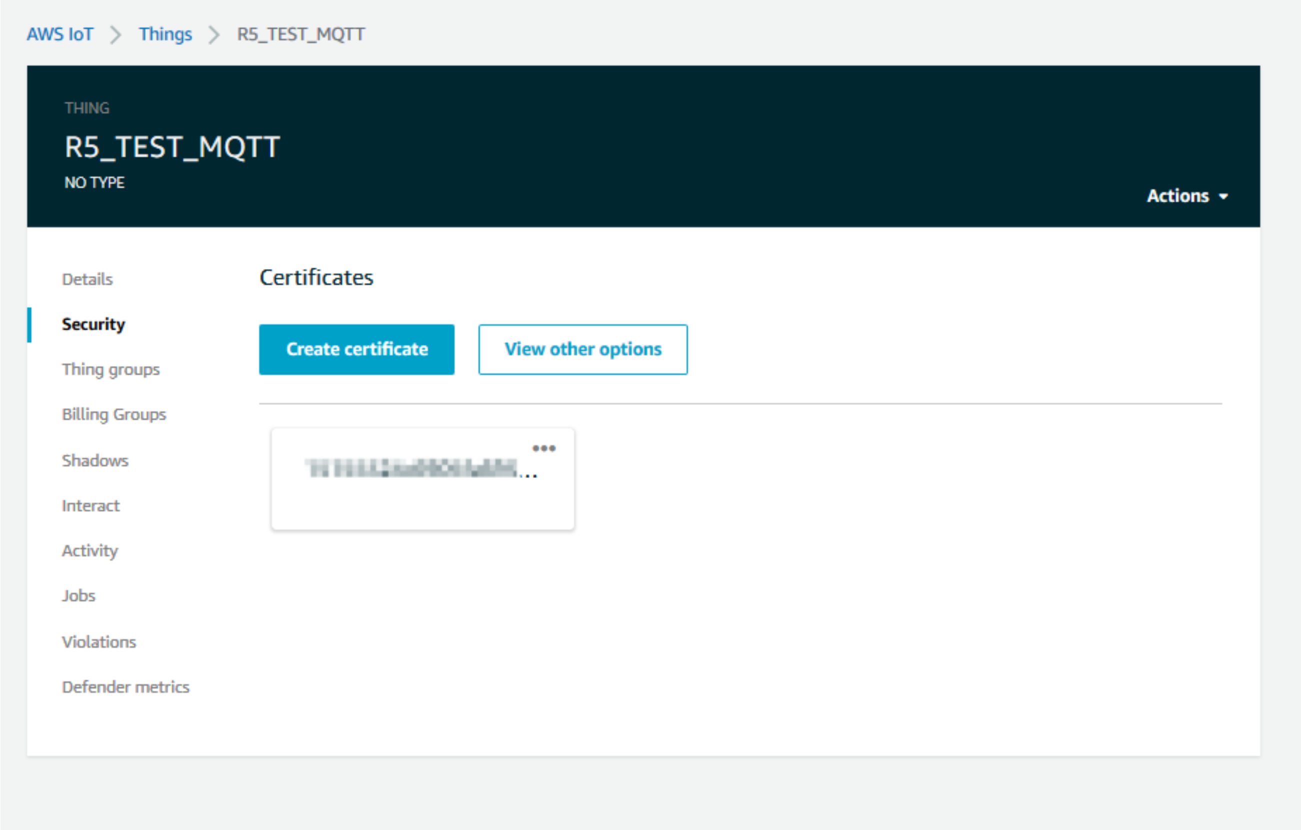


Figure 6: Select certificate

On the navigation menu click **Security**, then click the certificate created in the previous steps.

On the certificate page click **Policies** in the left-hand navigation menu. Next, click the **Actions** drop down on the right of the page. In the **Actions** drop down menu click **Attach Policy**.

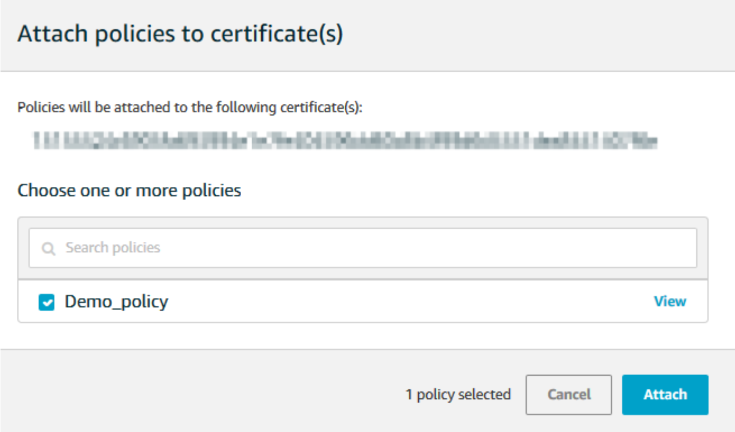

Attach policies to certificate(s)
Policies will be attached to the following certificate(s):
[redacted]
Choose one or more policies
Search policies
☑ Demo_policy
View
1 policy selected
Cancel
Attach

Figure 7: Attach policies

To conclude the process, select the policy created in section 1.1, then click **Attach**.

2 u-blox module setup

2.1 Store certificates in module file system

After downloading the CA and CC certificates and PK from AWS, store them in the module file system via AT commands or using m-center.

2.1.1 AT commands procedure to store the file in the module

Use the +UDWNFILE AT command to store all the certificates and keys required for communication in the flash file system of the module.

Command	Response	Description

`AT+UDWNFILE="aws_ca.pem",1188`	`>`
	`-----BEGIN CERTIFICATE-----`
	`hDKXJioaldXgjUkK642M4UwtBV8ob2x...`
	`jgSubJrIqg0CAwEAAaNCMEAwDwYDQnoZsG4q5WTP468SQvvG5`
	`-----END CERTIFICATE-----`
	`OK`
		After character “>” copy/paste the entire certificate.
		The file is stored successfully.

Repeat the same procedure for the other certificates that may be necessary: e.g., for CC and PK.

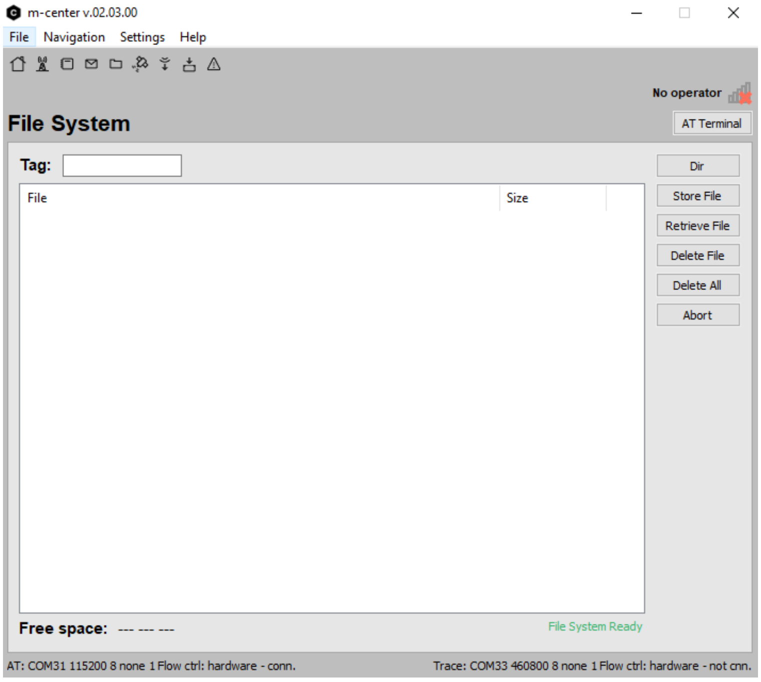
2.1.2 m-center procedure to store the file in the module

Similarly, m-center evaluation software can be used to store the certificates file into the module. The software uses the same +UDWNFILE AT command but it is masked by a simple GUI.

Any file can be stored in the module via the **File System Tab** (see Figure 8), by clicking **Store file**. A window will open where the chosen file can be selected from the Windows Explorer. Even in this case, repeat the same procedure for the other certificates that may be necessary: e.g., for CC and PK.

By clicking **Dir**, the m-center window will display all the stored files.

The u-blox m-center can be downloaded free-of-charge from our website (http://www.u-blox.com).



Free space: --- --- ---

File System Ready

AT: COM31 115200 8 none 1 Flow ctrl: hardware - conn.

Trace: COM33 460800 8 none 1 Flow ctrl: hardware - not cnn.

Figure 8: m-center File System tab

2.2 Check CA, CC, and PK in the file system

Command	Response	Description
AT+ULSTFILE=2,"aws_ca.pem"	+ULSTFILE: 1188 OK	CA availability in the module.
AT+ULSTFILE=2,"383847e4d4-certificate.pem.crt"	+ULSTFILE: 1224 OK	CC availability in the module.
AT+ULSTFILE=2,"383847e4d4-private.pem.key"	+ULSTFILE: 1679 OK	PK availability in the module

2.3 Certificates manager configuration

Command	Response	Description
AT+USECMNG=1,0,"AWS_CA","aws_ca.pem"	+USECMNG: 1,0,"AWS_CA","CB17E431673EE209FE455793F30AFA1C" OK	Import the CA in the certificates manager.
AT+USECMNG=1,1,"AWS_Client","383847e4d4-certificate.pem.crt"	+USECMNG: 1,1,"AWS_Client","50C3004AAE690124E3D7F96F904D7084" OK	Import the CC in the certificates manager.
AT+USECMNG=1,2,"Client_Key","383847e4d4-private.pem.key"	+USECMNG: 1,2,"Client_Key","CD879AA22744A7211D3AF5D3BEFAFF29" OK	Import the client PK in the certificates manager.

2.4 Security profile configuration

Command	Response	Description
AT+USECPRF=0,0,1	OK	Set the certificate validation level 1.
AT+USECPRF=0,2,0	OK	Set automatic the cipher suite.
AT+USECPRF=0,3,"AWS_CA"	OK	Set the trusted root certificate internal name.
AT+USECPRF=0,5,"AWS_Client"	OK	Set the client certificate internal name.
AT+USECPRF=0,6,"Client_Key"	OK	Set the client key internal name.
AT+USECPRF=0,10,"xxx-ats.iot.us-east-2.amazonaws.com"	OK	Set the Server Name Indication. ☞ SNI is a feature of SSL/TLS which uses an additional SSL/TLS extension header to specify the server name to which the client is connecting to. SNI configuration may be required to support the certificate handling used with virtual hosting provided by the various SSL/TLS enabled servers mostly in cloud-based infrastructures.

3 Example of MQTT session between module and AWS IoT

The best way to describe the interaction between a u-blox module and AWS IoT is through a simple use case. The following example describes a MQTT session that simulates a form of remote temperature control. The u-blox module is the MQTT client responsible for publishing temperature messages and receiving action messages from the AWS IoT server.

The module sends the temperature messages to the "`building/groundfloor/office_1/temperature`" topic and is also subscribed to the "`building/groundfloor/office_1/heating`" topic for receiving the AWS IoT commands. Both module AT commands and AWS actions are manually performed. On AWS it is possible to automate the operations by defining rules and actions but this topic is beyond the scope of this document.

3.1 Module setup: start a MQTT session and subscribe to a topic

☞ Correctly activate an IP data connection before using the AT commands in this section. This is necessary because a packet switched (PS) data connection must be activated before creating a socket and connecting to the AWS server.

Go to the AWS IoT Core console at console.aws.amazon.com/iot. In the navigation panel, choose **Settings**. The endpoint address is listed under **Custom endpoint**.

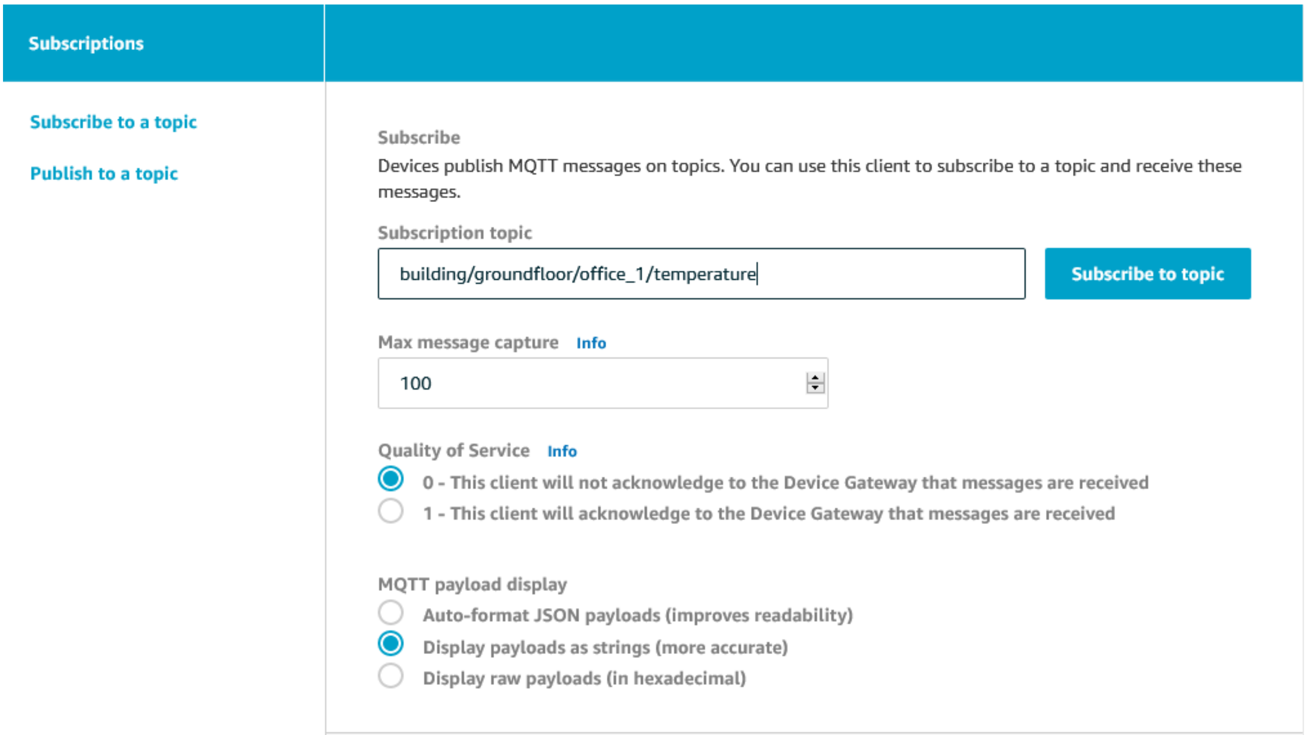
☞ The string "-ats" needs to be removed from the endpoint address in case a legacy certificate is used as root CA. On the opposite, as in the example here reported, if an Amazon root CA (certificate from Amazon Trust Services – see Figure 5) is used, the string "-ats" needs to be used in the endpoint address. Once identified the correct endpoint to use, the same endpoint address needs to be used as the remote server name in the +UMQTT AT command configuration but also as SNI in the +USECPRF AT command configuration.

☞ Furthermore, AWS IoT Core is currently supported using the legacy root CA certificate in a limited number of AWS regions. For the list of supported AWS region visit the following page: https://docs.aws.amazon.com/general/latest/gr/greengrass.html#greengrass-legacy-endpoints.

Command	Response	Description
`AT+UMQTT=2,"xxx-ats.iot.us-east-2.amazonaws.com",8883`	`OK`	Set the remote server name (the above endpoint address) and the server port (TLS MQTT).
`AT+UMQTT=11,1,0`	`OK`	Enable the secure connection option using the profile 0. See section 2.4.
`AT+UMQTTC=1`	`OK`	Connect to the AWS IoT broker and start a secure MQTT session.
	`+UUMQTTC: 1,1`	
`AT+UMQTTC=4,0,"building/groundfloor/office_1/heating"`	`OK`	Subscribe to the heating system control of the ground floor office #1.
	`+UUMQTTC: 4,1,0,"building/groundfloor/office_1/heating"`	

3.2 Configure AWS IoT: subscribe to a topic

From the AWS IoT Core console, select **Test** from the navigation pane, and choose **MQTT test client**. Enter the topic and click **Subscribe to topic,** as shown in Figure 9:



Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic: building/groundfloor/office_1/temperature

Max message capture Info: 100

Quality of Service Info

● 0 - This client will not acknowledge to the Device Gateway that messages are received

○ 1 - This client will acknowledge to the Device Gateway that messages are received

MQTT payload display

○ Auto-format JSON payloads (improves readability)

● Display payloads as strings (more accurate)

○ Display raw payloads (in hexadecimal)

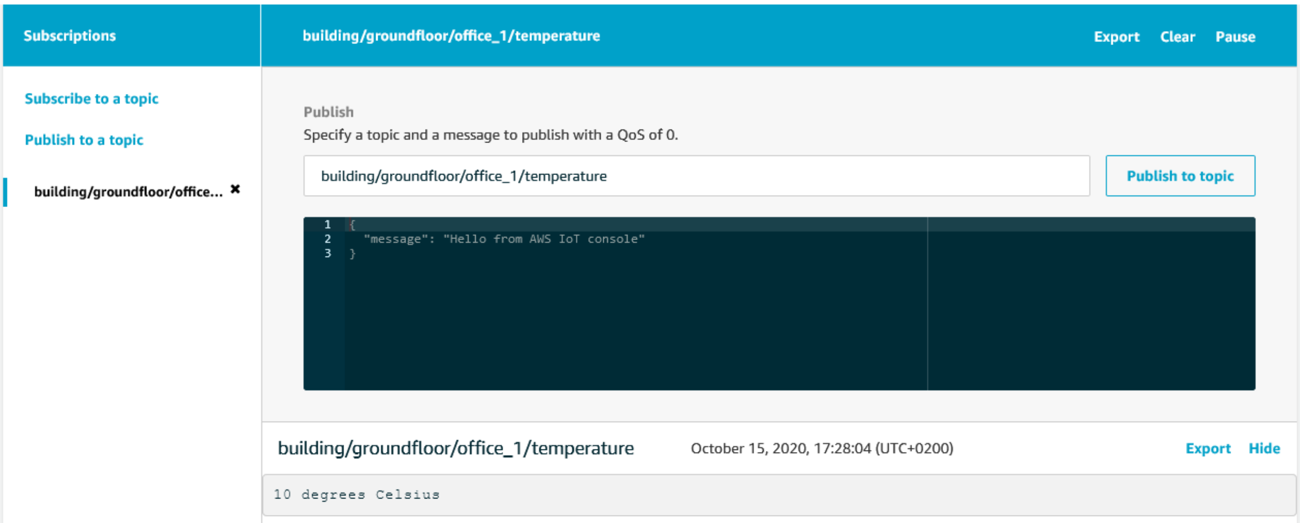
Figure 9: AWS subscribe to a topic

3.3 Module: publish a message to AWS IoT

Publish the temperature:

Command	Response	Description
`AT+UMQTTC=2,0,0,0,"building/groundfloor/office_1/temperature","10 degrees Celsius"`	`OK`	Publish the temperature of the ground floor office #1.
	`+UUMQTTC: 2,1`	

3.4 AWS IoT: read the received message



Specify a topic and a message to publish with a QoS of 0.

building/groundfloor/office_1/temperature

```
{
  "message": "Hello from AWS IoT console"
}
```

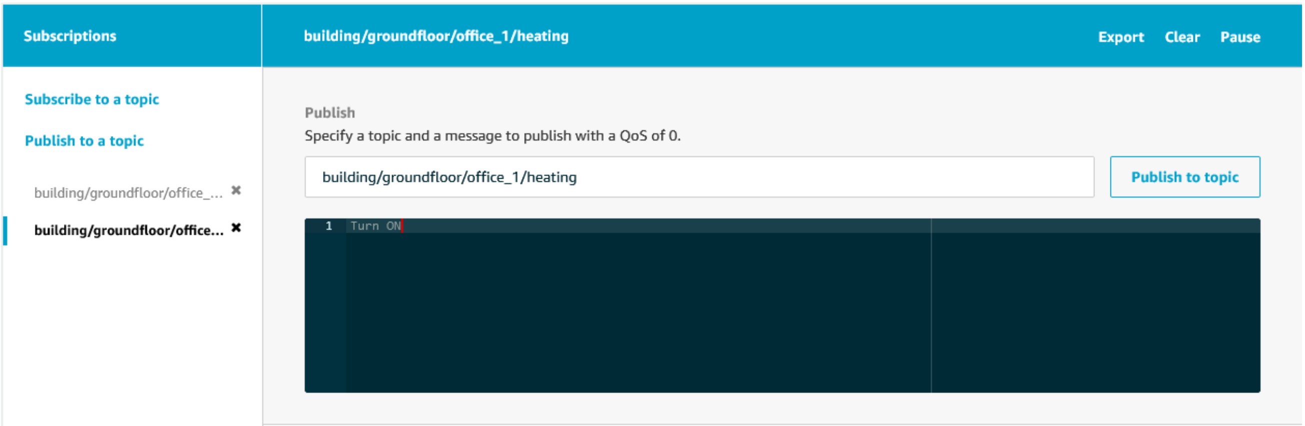
building/groundfloor/office_1/temperature October 15, 2020, 17:28:04 (UTC+0200)

```
10 degrees Celsius
```

Figure 10: AWS read a message

3.5 AWS IoT: publish a message to module

To publish a message to the “`building/groundfloor/office_1/heating`” topic, select **Publish to a topic**, enter the topic in the topic field, and then click **Publish to topic**.


Subscriptions
building/groundfloor/office_1/heating
Export
Clear
Pause
Subscribe to a topic
Publish to a topic
building/groundfloor/office_... ✖
building/groundfloor/office... ✖
Publish
Publish to topic

Specify a topic and a message to publish with a QoS of 0.

building/groundfloor/office_1/heating

```
Turn ON
```

Figure 11: AWS publish a message

3.6 Module read the received message

Command	Response	Description
	`+UUMQTTC: 6,1`	URC notifying the received publish message.

`AT+UMQTT=6,1`	`+UMQTTC: 6,0,44,37,"building/groundfloor/office_1/heating",7,"Turn ON"`
	`OK`
		Read the received message.

Appendix

A Glossary

Abbreviation	Definition
AWS	Amazon Web Services
CA	Certificate Authority
CC	Client Certificate
MQTT	Message Queuing Telemetry Transport
PK	Private Key
PS	Packet Switched
SNI	Server Name Indication
TLS	Transport Layer Security

Table 1: Abbreviations and terms

Related documentation

[1] u-blox SARA-R5 series data sheet, UBX-19016638

[2] u-blox SARA-R5 series AT commands manual, UBX-19047455

[3] u-blox SARA-R5 series system integration manual, UBX-19041356

[4] u-blox SARA-R4 / SARA-R5 internet application development guide, UBX-20032566

[5] u-blox SARA-R4 series data sheet, UBX-16024152

[6] u-blox SARA-R4 series AT commands manual, UBX-17003787

[7] u-blox SARA-R4 series system integration manual, UBX-16029218

☞ 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	26-Oct-2020	mreb	Initial release
R02	26-Jan-2021	mreb	Added more detailed instructions for connecting to the AWS IoT service
R03	29-Mar-2021	mreb	Extended document applicability to SARA-R422S and SARA-R422M8S
R04	28-Sep-2022	mreb	Extended document applicability to SARA-R422M10S

Contact

For further support and contact information, visit us at www.u-blox.com/support.