

ublox USB-NORA-W256AWS AWS IoT ExpressLink Multiradio Development Kit User Guide

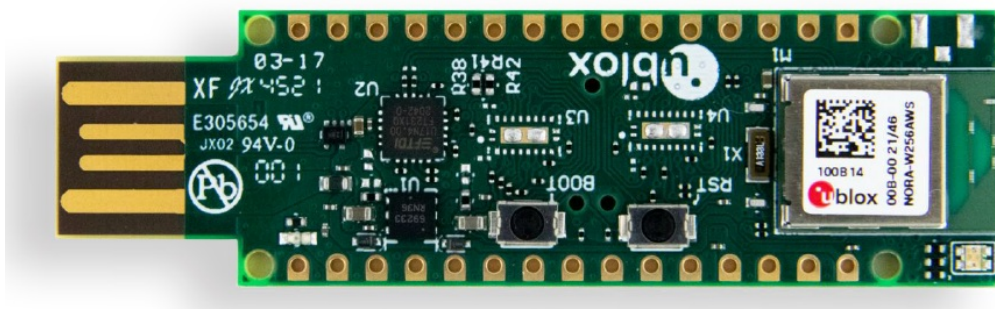
[Home](#) » [Ublox](#) » ublox USB-NORA-W256AWS AWS IoT ExpressLink Multiradio Development Kit User Guide

Contents [[hide](#)]

- 1 [ublox USB-NORA-W256AWS AWS IoT ExpressLink Multiradio Development Kit](#)
- 2 [Introduction](#)
 - 2.1 [Key features](#)
- 3 [Getting started](#)
- 4 [Run the “Quick Connect” demo application](#)
- 5 [Registering an AWS IoT ExpressLink in your development account](#)
- 6 [Connecting and interacting with AWS cloud](#)
- 7 [Over-The-Air updates for AWS IoT ExpressLink](#)
- 8 [Related Documentation](#)
- 9 [Documents / Resources](#)
 - 9.1 [References](#)
- 10 [Related Posts](#)



ublox USB-NORA-W256AWS AWS IoT ExpressLink Multiradio Development Kit



Introduction

The USB-NORA-256AWS evaluation kit comprises a small 16×22 mm evaluation board that conveniently exposes the supported interfaces of the NORA-W256AWS module, which is mounted directly on the board. The board has a USB form factor and plugs directly into the USB port of your computer.

This document explains how to setup the USB-NORA-W256AWS evaluation kit that includes the USB-NORA-W256AWS evaluation board and NORA-W256AWS module. Note that the module has not yet been provisioned in production for commercial use, which means that the provisioned security certificate and associated keys are not for use in production.

NORA-W2 AWS series stand-alone modules

NORA-W2 AWS stand-alone, multi-radio modules integrate a powerful 32-bit, dual core microcontroller unit (MCU) and radio for wireless communication. The modules support either an internal antenna (NORA-W256AWS) or external antennas connected through a dedicated antenna pin (NORA-W251AWS). The radio provides support for Wi-Fi 802.11b/g/n in the 2.4 GHz ISM band.

The embedded AWS IoT ExpressLink-compliant software includes secured certificates that are pre-flashed in the modules. This allows the modules to provide “out of the box” connectivity with Amazon Web Services (AWS) without any additional effort from the customer. NORA-W2 AWS modules also support secure Host and Firmware Over the Air (OTA) upgrades to sustain a high level of quality. Control and data communication is handled through the module with stateless AT-commands over a serial interface.

The evaluation kit supports modification of the endpoint to your development account. In the commercial releases of this module this endpoint is pre-defined to the u-blox staging account and is transferred to the customer AWS account at a later stage.

NORA-W2 AWS modules include wireless MCU, flash memory, crystal, and other components for matching, filtering, antenna, and decoupling, which means that these are very compact, stand-alone, multiradio modules. The modules can be used to design solutions with top-grade security, thanks to integrated cryptographic hardware accelerators. This enables secure boot, which ensures that the module boots up only in the presence of authenticated software. The small size and the embedded security capabilities make NORA-W2 AWS modules ideal for critical IoT applications where security is important. Intended applications include consumer products, telematics, low power sensors, connected factories, connected buildings (appliances and surveillance), point-of-sales, and health devices.

NORA-W2 AWS modules are globally certified, which significantly reduces the time to market for end products. To ensure operation in harsh professional environments, NORA-W2 AWS modules are of a professional grade, support an extended temperature range of –40 °C to +85 °C, and are qualified according to ISO 16750. See also the NORA-W2 series data sheet [1].

USB-NORA-W256AWS evaluation board

USB-NORA-W256AWS is a versatile development platform that allows quick prototyping of a variety of extremely low-powered Internet of Things (IoT) applications. The radio provides support for Bluetooth Low Energy 5.01 and Wi-Fi 802.11 b/g/n in the 2.4 GHz ISM band.

USB-NORA-W256AWS includes an internal PCB antenna and NORA-W256AWS module that is pre-flashed with AWS IoT ExpressLink compliant software.



Figure 1: USB-NORA-W256AWS evaluation board

Kit includes

The kit includes the USB-NORA-W256AWS evaluation board with USB connector. The NORA-W256AWS module is mounted on the evaluation board.

With a USB Type A form factor, the USB-NORA-W256AWS evaluation board is inserted directly into the PC host. For more information, see also the USB-NORA-W2 product web page.

User provided items

Item	Description
PC	Computer with USB type A port
Wi-Fi Access Point/Router	Wi-Fi Access Point with public internet access. WPA, WPA2 or WPA3 Wi-Fi authentication must be supported.

Table 1: User provided items

Purchasable third-party items

No additional items are required.

Key features

Pre-provisioned with AWS cloud

With the prototype USB-NORA-256AWS evaluation kit, the user needs to manually register the device in their own AWS account. This manual process is only applicable to the prototype version of the evaluation board and is subject to change. The manual process of Registering an AWS IoT ExpressLink in your development account.

The production release of the NORA-W256AWS module is provisioned with securely stored keys and certificates for secure connection to the AWS cloud during production. No module setup, apart from the verification of Wi-Fi credentials, is required to connect the end-product to the AWS cloud.

Simple integration with stateless commands

Communication from the host to the module is performed using a simple and easy-to-use AT-command set over a serial interface. The USB-NORA-W256AWS evaluation board is designed to conveniently insert into a computer USB port for prototyping and development. For details about the supported commands, see also the AWS AT-command information on the AWS IoT ExpressLink developer page [4].

Wi-Fi 802.11b/g/n

USB-NORA-W256AWS communicates with the AWS Cloud over Wi-Fi 4.

Enhanced security features

NORA-W2AWS2 modules contain a multistage secure boot that ensures that the running software, as well as the hardware, is authentic. All provisioned certificates and keys are stored in the secure memory of the module. Keys cannot be read or modified externally. NORA-W2AWS modules support MQTT TLS 1.2 and Wi-Fi WPA, WPA2 and WPA3 authentication protocols for internet communication. The prototype kits hosting the module have security limitations³.

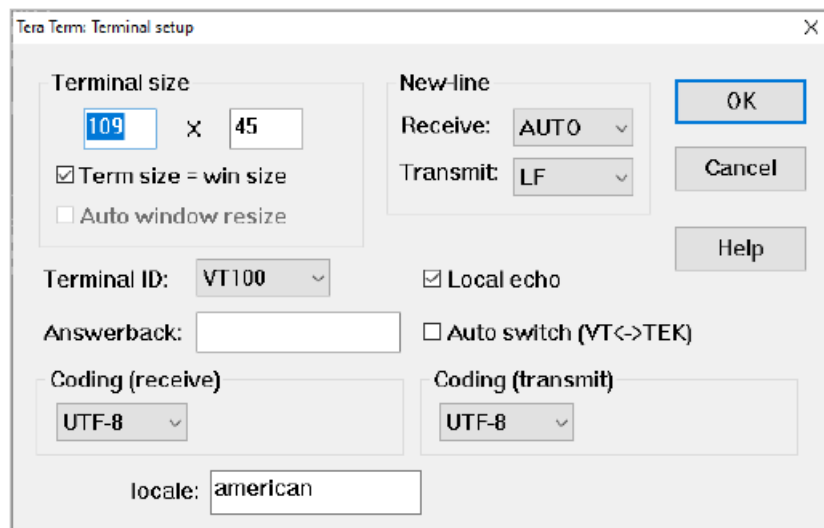
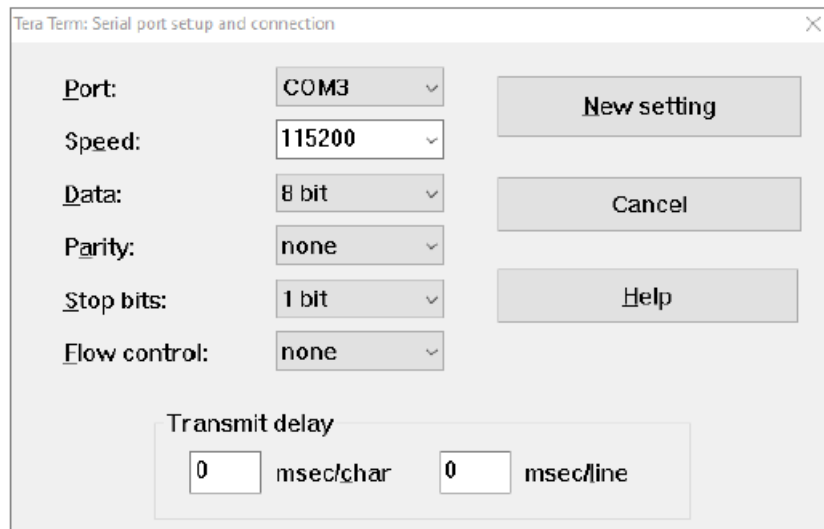
Getting started

Setting up the host

Use the following procedure to set up the USB-NORA-W256AWS evaluation board on a host machine. Any operating system is supported that supports running a terminal application.

Depending on your computer configuration, it may be necessary to download and install USB-to-Serial drivers to enable the computer to perform serial communication over USB. The drivers and driver installation instructions can be found [here](#).

1. Connect USB-NORA-W256 to the host machine
2. Open a terminal application on your host machine, like TeraTerm for Windows or CoolTerm for Mac.
3. Select the port corresponding to the evaluation kit. Consult the documentation for your operating system to find out how to detect the port used by the evaluation kit.
4. Configure the terminal application as follows:
 - **Baudrate:** 115200
 - **Bits:** 8
 - **Parity:** None
 - **Stop:** 1
 - **Flow control:** None
 - **Local Echo:** Yes



Verifying evaluation board connection

Verify that you have a working connection to the evaluation board:

1. Open the terminal window
2. Type "AT" and press return.
3. Verify that a "OK" is written in the terminal windows. This confirms that the connected the evaluation kit is successfully connected to your host machine.

```
AT
OK
```

Keep the terminal open, as it is needed for Connecting and interacting with AWS cloud and Troubleshooting.

AWS account and permissions for IoT development

Go to "Set up your AWS account" [2] and complete the following tasks:

- Sign up for an AWS account
- Create a user and grant permissions
- Open the AWS IoT console

See also Registering an **AWS IoT ExpressLink** in your development account.

Run the “Quick Connect” demo application

The Quick Connect demo application allows you to establish a connection with AWS IoT, all in the space of a few minutes; no dependencies to install, no source code to download and build, and no AWS account required. To run the demo, follow the steps below:

1. If you opened a terminal application in the previous step, be sure to disconnect that application from the serial port.
2. Download the Quick Connect executable:
 1. Download for Mac
 2. Download for Windows
 3. Download for Linux
3. Unzip the package. Open the config.txt file and enter the serial port corresponding to the evaluation kit. For example, COM14, /dev/cu.usbserial-12345, and so on in the serial port field.
4. Enter your Wi-Fi credentials in the SSID and Passphrase fields.
5. Run the “Start_Quick_Connect” executable.

The demo connects to AWS IoT Core and print an URL that you can use to visualize data flowing from the device to the cloud using AT+SEND commands. The demo runs for up to two minutes, and afterwards, you can type AT+SEND commands yourself and see the data in the Quick Connect Dashboard shown in **Figure 1**.

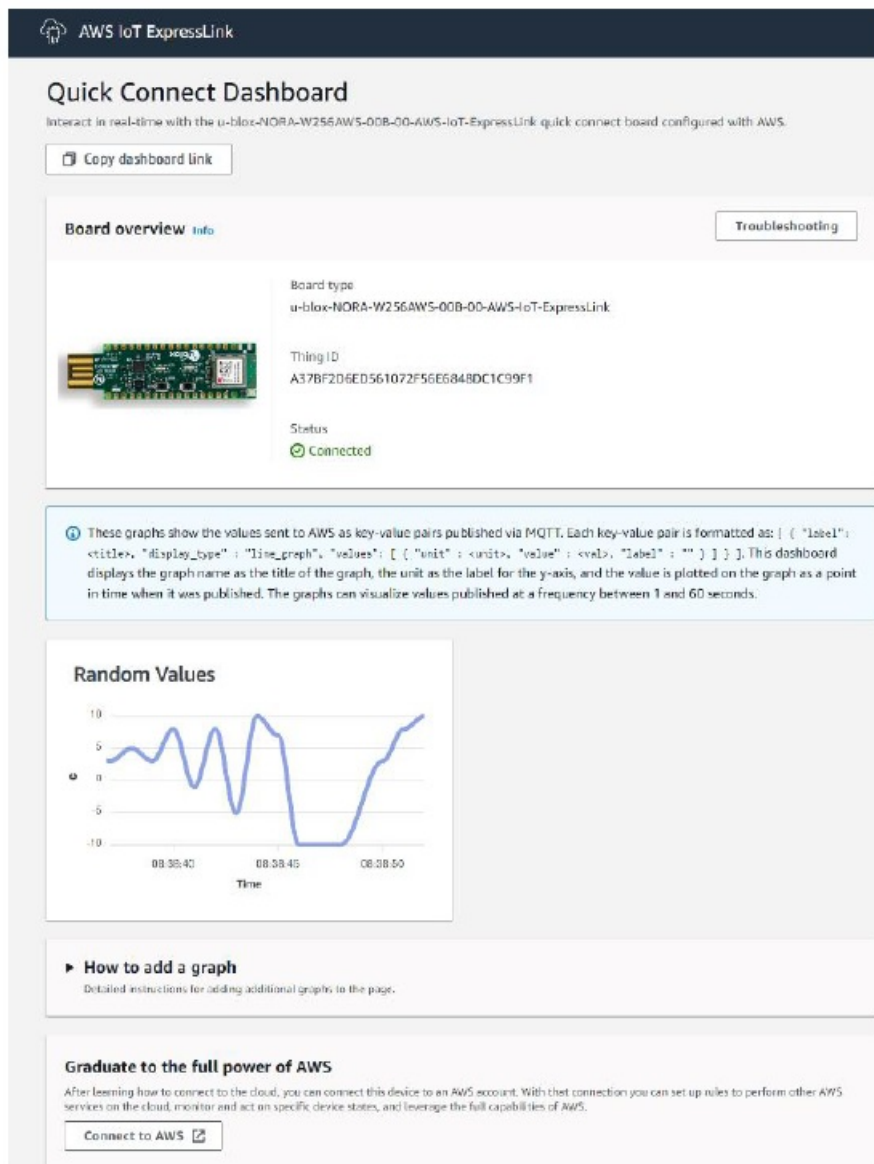


Figure 2: Quick Connect Dashboard

Registering an AWS IoT ExpressLink in your development account

To create an IoT “Thing” and add it to your account you must retrieve the AWS IoT ExpressLink “Thing Name” and the corresponding certificate of the module. The “Thing name” is a sequence as characters that identifies the NORA-W256AWS module and its virtual cloud representation.

Follow the procedure below to register your development account using the AWS Management Console [5]41.

1. Open the AWS IoT Console.
2. Select Manage then select Things.
3. Choose Create things, select Create single thing, and then click Next.
4. In the terminal application, type the command: AT+CONF? ThingName and copy the returned string (a sequence of alphanumeric characters) from terminal.
5. On the Specify thing properties page, paste the copied string from terminal into the Thing name under Thing properties on the console. Leave all other fields as default, then click Next.
6. In the terminal application, type the command: AT+CONF? Certificate pem
7. Copy the returned string (a longer sequence of alphanumeric symbols) and save the string as a text file called “ThingName.cert.pem” on your host machine.

8. On the Configure device certificate page, select Use my certificate and choose CA is not registered with AWS IoT.
9. Under Certificate, select Choose file and then double click on the "ThingName.cert.pem" file created in step 5.
10. Under Certificate Status, select Active.
11. Click Next to Attach policies to certificate.
12. On the Attach policies to certificate page, select Create policy (opens a new window).
13. Include the policy name (e.g., IoTDevPolicy) and click Advanced mode.
14. Copy the following code snippet into the console.

```
{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "*", "Resource": "*" } ] }
```
15. Click Create to complete policy creation.
16. Close Create a policy window and return to Create single thing window.
17. Select the newly created IoTDevPolicy as policy.
18. Click Create thing to complete the Thing creation.
19. In the AWS IoT Console, choose Settings, copy the "Endpoint" string of your account under Device data endpoint.
20. In the terminal application, type the following command for example:

```
AT+CONF Endpoint=a3ixxxxxxx7i2-ats.iot.eu-north-1.amazonaws.com
```

The examples in this document are intended only for development 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, see also "Security best practices in AWS IoT Core" in the AWS documentation [6].

Setting up and connecting to Wi-Fi

The USB-NORA-W256AWS evaluation board requires access to a local Wi-Fi router to connect to the internet. Enter the required security credentials and enter the following commands in your terminal application:

```
AT+CONF SSID=<your router SSID>
AT+CONF Passphrase=<your router passphrase>
```

The SSID and passphrase of your local router are stored securely inside the AWS IoT ExpressLink module. While the SSID can be retrieved later (for debugging purposes and so on) any attempt to retrieve the Passphrase will return an error.

Validating the onboarding process

Having completed all previous steps in this chapter, enter the command AT+CONNECT in your Terminal application to validate the onboarding process. OK CONNECTED confirm successful connection with the cloud.

AT+CONNECT
OK 1 CONNECTED

You have now completed the registration of the evaluation kit as a "Thing" in your IoT account. As the AWS IoT ExpressLink remembers its configuration, the module automatically accesses your registered AWS account the next time you connect.

Connecting and interacting with AWS cloud

Use the MQTT client in the AWS IoT console to monitor the communication between your evaluation kit and the AWS Cloud.

1. Navigate to the AWS IoT console [3].
2. In the navigation pane, select Test and then MQTT Test Client to open the MQTT client.
3. In Subscribe to a topic, enter #. The multi-level wildcard subscribes and listens to all payloads published to your account.
4. Click Subscribe.

Connecting

In your terminal application, enter the command `AT+CONNECT` to establish a secure connection. `OK CONNECTED` confirms successful connection to the cloud.

Send data to AWS cloud

To check communication with the MQTT test client:

1. In your terminal application, enter the command: `AT+CONF Topic1=MyTopic`
2. In your terminal application, enter the command: `AT+SEND1 Hello from my IoT device`. This command sends the “Hello from my IoT device” string to the previously defined Topic1. After a short delay, terminal returns the prompt `OK` to confirm that the command has been sent.
3. Check that the message “Hello from my IoT device” is now displayed on the AWS IoT console under the topic “MyTopic”, as shown in **Figure 3**.

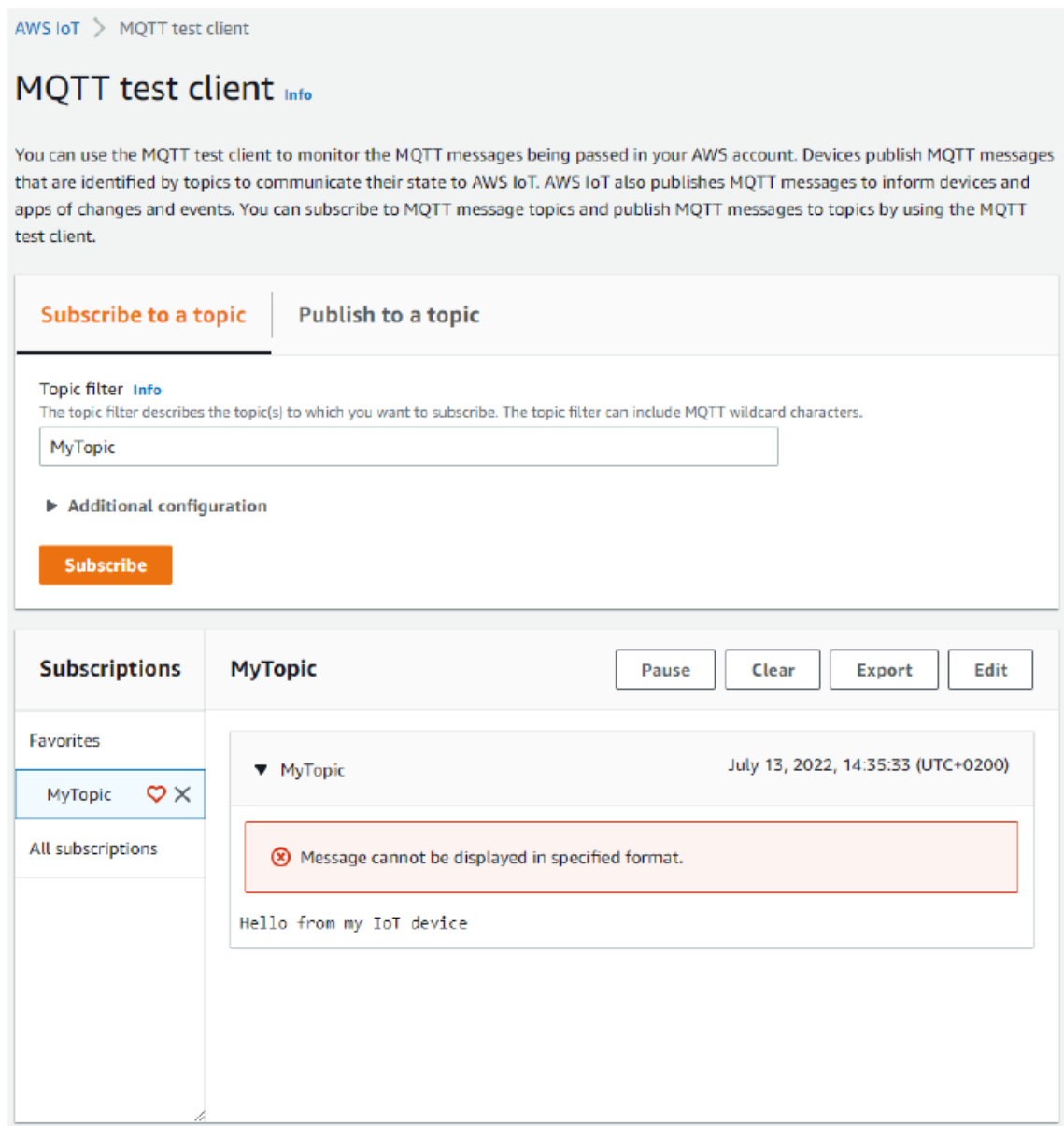


Figure 3: MQTT test client in AWS IoT console

Receiving data and commands from AWS cloud

Use your terminal application and AWS IoT console to check the receipt of data and commands from the AWS cloud.

Enter the commands below in the terminal application to subscribe to a topic:

1. Enter the command `AT+CONF Topic1=MyTopic`. After a short delay, terminal returns the prompt OK to confirm that the command has been sent.
2. Enter the command `AT+SUBSCRIBE1`

To publish a message on the same topic, enter the following commands in the AWS IoT console:

1. Select Publish to a topic
2. in Topic name field, type “MyTopic”.
3. Enter the text “Hello from the AWS IoT console” message then click “Publish”

In the terminal application enter the command AT+GET1. The message “OK Hello from the AWS IoT console” is returned at the prompt.

Over-The-Air updates for AWS IoT ExpressLink

Acquiring software updates

Firmware update images are available on the u-blox product webpage under the “Documentation & resources” tab [7].

Prerequisites

You should have a firmware image signed by the manufacturer of the ExpressLink module. Along with the firmware image, you should also receive additional signing metadata such as:

- The signature hashing algorithm used (Example: SHA-256)
- Signature encryption algorithm used (Example: RSA)
- Actual signature encoded using the base64 encoding format.
- The path name (a string) which identifies the location where the certificate is provisioned in the ExpressLink (optional)

Create an OTA update job in AWS IoT

Create an OTA Update role in your account:

1. Open AWS IoT Console. Select Manage then select Jobs. Choose Create job, select Create FreeRTOS OTA Update Job and click Next.
2. Provide a job name which is unique within your AWS account. Provide an optional description. Click Next.
3. From Devices to update drop-down menu, choose the “Thing” name with which the USB-NORA-W256AWS has a registered account. Select MQTT as the protocol to use for transfer.
4. Choose Use my custom signed file and complete the displayed form. When filling the form, use the details supplied in the firmware package.
5. In the signature field provide the base64 encoded signature for the image. From the Original hashing algorithm drop-down menu, select the hashing algorithm given in the firmware package.
6. From the Original encryption algorithm drop-down menu, select the encryption algorithm given in the firmware package. For Path name of code signing certificate on device, enter the path name as stated in the same procedure. If path name is not provided, enter NA.
7. Select Upload a new file, click on Choose file and upload the image received from u-blox.
8. Select Create S3 bucket for the new uploaded image and proceed with creating a new bucket. If needed, choose an existing bucket in your account by selecting the Browse S3 option.
9. Under the Path Name of the file on the device, enter NA if the image is not targeted as an executable file within a filesystem.
10. In the File type input field, define the type of software to be updated – either the ExpressLink software (FOTA) or host software (HOTA). Set the value 101 for ExpressLink and 202 for the host software.
11. From the role dropdown under the IAM role, choose the OTA update role created above. Click Next.
12. Click Create Job. On successful creation, the job name and state are displayed as in progress.

Monitoring and applying for OTA jobs

The NORA-W256AWS IoT ExpressLink module automatically polls for firmware update jobs. Once a new job has been detected it downloads and validates the software image. It then enters a waiting state where the host application must accept the update to be applied. The process is described below.

1. The host application receives an OTA event indicating that a new firmware image is available for ExpressLink.
The host application can query the state of the job using the command AT+OTA?.
2. The module responds with OK 1 <version> to confirm that a module OTA firmware update is proposed.
3. The host application can accept a new firmware update for by issuing command AT+OTA ACCEPT.
4. The module starts downloading the firmware update from the AWS IoT cloud. During the download, the host can monitor the state of the job using AT+OTA?
5. After successful download and signature validation, the host receives an event to apply the new image.
6. The host application can apply the new image by issuing the command AT+OTA APPLY.
7. When the apply command has been sent, the module reboots using the new software image. The host receives a STARTUP event confirming that the new image is booted. To see the event, enter the command AT+EVENT?
8. The host application can connect back to the AWS IoT with the command AT+CONNECT.
9. The module now connects to AWS IoT, completes the self-test, and marks the image as valid to prevent rollback to any older image.
10. The job status in the AWS IoT console is shown as completed and succeeded.

Related Documentation

1. NORA-W2 series data sheet, UBX-21046925
2. Set up your AWS account, <https://docs.aws.amazon.com/iot/latest/developerguide/setting-up.html>
3. AWS IoT ExpressLink main page, <https://aws.amazon.com/iot-expresslink>
4. AWS IoT ExpressLink developer page, <https://docs.aws.amazon.com/expresslink>
5. AWS Management Console, <https://aws.amazon.com/console/>
6. Security best practices in AWS IoT Core,
<https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html>
7. NORA-W2 series <https://www.u-blox.com/en/product/nora-w2-series>

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

Revision history

Revision	Date	Name	Comments
R01	19-Nov-2021	magj	Initial release for demonstrator variant
R02	5-Aug-2022	magj	Updated for software v0.99

Contact

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

Documents / Resources

 <p>USB-NORA-W256AWS AWS IoT ExpressLink Multiradio development kit</p>	<p>ublox USB-NORA-W256AWS AWS IoT ExpressLink Multiradio Development Kit [pdf] User Guide</p> <p>USB-NORA-W256AWS, AWS IoT ExpressLink Multiradio Development Kit, USB-NORA-W256AWS AWS IoT ExpressLink Multiradio Development Kit, ExpressLink Multiradio Development Kit, Multiradio Development Kit, Development Kit</p>
--	---

References

- [🌐 console.aws.amazon.com/iot](https://console.aws.amazon.com/iot)
- [📺 Home | u-blox](#)
- [📺 Home | u-blox](#)
- [📺 Technical Support | u-blox](#)
- [aws AWS Management Console](#)
- [aws Quickly Develop Secure IoT Devices | AWS IoT ExpressLink | Amazon Web Services](#)
- [📖 Security best practices in AWS IoT Core - AWS IoT Core](#)
- [📖 Set up your AWS account - AWS IoT Core](#)
- [🌐 Error during processing.](#)
- [🌐 Error during processing.](#)
- [🔊 D2XX Drivers - FTDI](#)
- [🌐 quickconnectexpresslinkutility.s3.us-west-2.amazonaws.com/QuickConnect_v1.9_linux.x64.tar.gz](https://quickconnectexpresslinkutility.s3.us-west-2.amazonaws.com/QuickConnect_v1.9_linux.x64.tar.gz)
- [🌐 quickconnectexpresslinkutility.s3.us-west-2.amazonaws.com/QuickConnect_v1.9_macos.x64.tar.gz](https://quickconnectexpresslinkutility.s3.us-west-2.amazonaws.com/QuickConnect_v1.9_macos.x64.tar.gz)
- [📺 u-blox.com/docs/UBX-21046925](https://u-blox.com/docs/UBX-21046925)
- [📺 NORA-W2 series | u-blox](#)
- [📺 USB-NORA-W2 | u-blox](#)