Getting Started Guide

ExpressLink

ExpressLink: Getting Started Guide

Table of Contents

Getting started with AWS IoT ExpressLink

The following sections will guide you through the required steps to connect your AWS IoT ExpressLink Evaluation Kit to the cloud to send and receive data directly with your AWS account. For module specific guides, see the Getting Started Guide for your specific AWS IoT ExpressLink hardware module in the AWS Partner Device Catalog.

If your ExpressLink module manufacturer supplies an Evaluation Kit, you will want to follow the specific steps provided there. Also, the general steps you follow for a manufacturer-specific Evaluation Kit are provided in section "11.2.1 Run the Quick Connect demo application" of the AWS IoT ExpressLink Programmer's Guide .

If you have questions or issues that are not answered here, please visit the AWS re:Post for AWS IoT ExpressLink page.

Topics

Set up your host machine

AWS IoT ExpressLink evaluation kits can be connected to a host machine serial interface using a terminal application.

Prerequisites: To establish a serial interface connection between your host machine and the evaluation kit, you must install the corresponding USB to UART bridge Virtual Communication Port drivers. Refer to your hardware module's *Getting Started Guide* for any specific requirements.

1. Open a terminal application for your host machine (for example, TeraTerm for Windows, CoolTerm for Mac) and select the port corresponding to the evaluation kit.
2. Configure the terminal application with the following settings:

```
Baudrate:     115,200
Bits:          8
```

```
Parity:        None
Stop:          1
Flow control:  None
Local Echo:    Yes
End of Line:   Line Feed
```

3. To check your connection, enter the following command from the terminal:

   ```
   AT
   ```

   If you receive the answer 'OK', then you've successfully connected the evaluation kit to your host machine.

Keep the terminal window open. You'll use the terminal later in this procedure.

AWS account set up and console login

Before you use AWS IoT Core for the first time, complete the tasks in this section.

You can expect to spend about 5 minutes setting up your AWS account.

If you already have an AWS account, you can skip ahead to the section called "Open the AWS IoT console".

Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

To sign up for an AWS account

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call or text message and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

Sign in as the user with administrative access

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

Assign access to additional users

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

Open the AWS IoT console

Most of the console-oriented topics in this section start from the AWS IoT console. If you aren't already signed in to your AWS account, sign in, then open the AWS IoT console and continue to the next section to continue getting started with AWS IoT.

Register an AWS IoT ExpressLink module to your account

To create an AWS IoT thing and add it to your account you must retrieve the AWS IoT ExpressLink module Thing Name and its corresponding certificate. Follow these steps:

1. Open the AWS IoT console. In the navigation pane choose **Manage** then choose **Things**. Choose **Create things**, select **Create single thing**, then choose **Next**.
2. Open the terminal application on your host machine and enter the command:

   ```
   AT+CONF? ThingName
   ```

   Copy the returned string (a sequence of alphanumeric characters) from the terminal.
3. Return to the AWS IoT console, and on the **Specify thing properties** page under **Thing properties**, paste the string you copied from the terminal into the **Thing name** field. Leave other fields with their default values, then choose **Next**.
4. In the terminal application, enter the command:

```
AT+CONF? Certificate pem
```

5. Copy the returned string (a longer sequence of alphanumeric symbols), and save it in a text file on your host machine as "`ThingName.cert.pem`".
6. In the AWS IoT console, on the **Configure device certificate** page, select **Use my certificate**, then select **CA is not registered with AWS IoT**.
7. Under **Certificate**, choose **Choose file**. Select the file "`ThingName.cert.pem`" that you created in a previous step, then choose **Open**.
8. Under **Certificate status**, select **Active**, then choose **Next**.
9. Under **Attach policies to certificate**, choose **Create policy**.
10. Enter a **Policy name** (for example, "IoTDevPolicy"), then under **Policy document** select **JSON**.
11. Copy the the following into the console **Policy document**:

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

    > **Warning**
    >
    > 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, see Example IAM identity-based policies and Security Best practices in the *IAM Identity and Access Management User Guide*.

12. Choose **Create**. Return to the **Attach policies to certificate** page and select the policy you just created (for example, "IoTDevPolicy"), then choose **Create thing** to complete the thing creation.
13. In the AWS IoT console, in the navigation pane, choose **Settings**. Under **Device data endpoint** select the **Endpoint** to make a copy of the endpoint for your account.
14. In the terminal application, type this command using the endpoint you just copied:

    ```
    AT+CONF Endpoint=your endpoint string here
    ```

Set up for Wi-Fi modules

AWS IoT ExpressLink modules that support Wi-Fi connectivity require access to a local Wi-Fi router in order to connect to the internet. You can enter the required security credentials with the following additional steps:

1. In the terminal application, enter the command:

   ```
   AT+CONF SSID=your router ssid
   ```

2. In the terminal application, enter the command:

   ```
   AT+CONF Passphrase=your router passphrase
   ```

   > **Note**
   >
   > Your local router's SSID and passphrase are stored securely inside the AWS IoT ExpressLink module. While the SSID can be retrieved later (for debugging purposes) any attempt to retrieve the Passphrase will return an error.

Completion

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

Connect and interact with AWS Cloud.

In this section, you 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.
2. In the navigation pane, choose **Test** and then **MQTT Test Client**.
3. In **Subscribe to a topic**, enter **#**, and then choose **Subscribe**.

Connect

n this section, you learn how to connect to AWS Cloud. This is a two step process.

To establish a secure connection

1. Open the terminal application on your host machine and enter the command:

   ```
   AT+CONNECT
   ```

2. After a short time, you will receive the message:

   ```
   OK 1 CONNECTED
   ```

   Congratulations! You are now successfully connected to your AWS Cloud account.

Send data to AWS Cloud

In this section, you learn how to send a message to AWS Cloud. This is a three step process.

To send a "Hello World!" message

1. Open the terminal application on your host machine and enter the command:

   ```
   AT+CONF Topic1=data
   ```

   You should receive the response from the module:

   ```
   OK
   ```

2. In the terminal application, enter the command:

   ```
   AT+SEND1 Hello World!
   ```

   After a short time, you should receive the message `OK`.

3. In the AWS IoT console MQTT test client you will see `Hello World!` message with the topic `data`.

Receive data and commands from AWS Cloud

In this section, you learn how to receive data and commands from AWS Cloud. This is a four step process.

To receive messages and data

1. Open the terminal application on your host machine and enter the command:

   ```
   AT+CONF Topic1=MyTopic
   ```

   You should receive the response from the module:

   ```
   OK
   ```

2. In the terminal, enter the command:

   ```
   AT+SUBSCRIBE1
   ```

3. In the AWS IoT console MQTT Test Client, choose **Publish to a topic**, and enter **`MyTopic`** in the topic name field. Keep the default message ("Hello from AWS IoT console") in the message field, then choose **Publish**.
4. In the terminal application, enter the command:

   ```
   AT+GET1
   ```

   You should receive the message:

   ```
   OK Hello from AWS IoT console
   ```

Perform a firmware Over-The-Air update for AWS IoT ExpressLink

To perform a irmware Over-The-Air update for AWS IoT ExpressLink, the following must be completed:

Topics

Prerequisites

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

- signature hashing algorithm used (Example: SHA-256)
- signature encryption algorithm used (Example: ECDSA)
- actual signature encoded using the base64 encoding format.
- (Optional) path name (a string) which identifies the location where the certificate is provisioned in the ExpressLink module

Finally, before you proceed, you should create an OTA Update role in your AWS account using the steps outlined in Create an OTA Update service role.

Create a firmware update job in AWS IoT

1. Open the AWS IoT console. Choose **Manage** then choose **Jobs**. Choose **Create job**, **Create FreeRTOS OTA Update Job**, then choose **Next**.
2. Enter a job name which is unique within your AWS account. Enter an optional description. Choose **Next**.
3. From the **Devices to update** dropdown, choose the thing name associated with your ExpressLink module when it was registered with the account. Choose **MQTT** as the protocol that will be used to perform the transfer, and unselect **HTTP** if it is selected.
4. Choose **Use my custom signed file**; this will display a form to be filled in. Use the details from Prerequisites to fill in the form.
5. In the **signature** field, enter the base64 encoded signature for the image. From the **Original hashing algorithm** drop down, select the hashing algorithm provided by the manufacturer. From the **Original encryption algorithm** drop down, select the encryption algorithm provided by the manufacturer. In the **Path name of code signing certificate on device**, enter the path name, if any is provided by the manufacturer. (If the path name is not provided, then you can just enter 'NA'.)

6. Choose **Upload a new file**, then **Choose file** and upload the image you received from the manufacturer. Choose **Create S3 bucket** for the new uploaded image and proceed with creating a new bucket. If needed, you can also choose an existing bucket in your account by selecting **Browse S3** option.
7. Under **Path Name of file on device** you can enter 'NA' if the image is not targeted as an executable file within a filesystem.
8. From the **File type** drop down select a value (default is 0) to signify this is an ExpressLink firmware update as opposed to a host firmware update.
9. From the **role** dropdown under the **IAM role** section, select the OTA update role you created above. Choose **Next**.
10. Choose **Create Job**. If the job creation was successful, it should list the job name and state as 'in progress'.

Monitor and apply a new firmware update for AWS IoT ExpressLink

After you create a firmware update job as described in the previous section, the ExpressLink module polls for firmware update jobs, receives and validates a job, and enters a state waiting for the update to be accepted. The host application receives an OTA event that indicates a new firmware image is available for the ExpressLink module.

1. Use the host terminal application to query the state of the job. Enter the command:

   ```
   AT+OTA?
   ```

   You should see the module respond with 'OK 1 *version*' to verify that a module OTA firmware update was proposed.
2. To accept the new firmware update, use the host terminal application to issue the command:

   ```
   AT+OTA ACCEPT
   ```

3. The ExpressLink module should now start downloading the firmware update from the cloud. You can monitor the state of the job using the 'AT+OTA?' command.

   When the download is complete and the image signature validation is successful, the host terminal application receives an event that indicates the module is ready to apply the new image.

4. Direct the module to apply the new image by issuing the command:

   ```
   AT+OTA APPLY
   ```

5. The ExpressLink module now reboots and boots up with the new image. The host terminal application receives a 'STARTUP' event indicating the new image is booted. To see the event, issue the command:

   ```
   AT+EVENT?
   ```

   Note: the event queue is shown in FIFO order, so you may have to issue the 'AT+EVENT?' command multiple times, depending on how many events are in the queue.

6. Use the host terminal application to direct the module to reconnect to AWS IoT by issuing the command:

   ```
   AT+CONNECT
   ```

   The ExpressLink module should now connect to AWS IoT, complete the self-test and mark the image as valid (preventing any further rollback to the old image).

7. Return to the AWS IoT console and verify that the job status is marked as completed and succeeded.