Getting Started with AWS IoT on Moxa UC-8100-LX Using Embedded C SDK

Jimmy Chen (陳永達)
Field Application Engineer
jimmy.chen@moxa.com

Contents

Released on January 11, 2017

About Moxa

Moxa is a leading manufacturer of industrial networking, computing, and automation solutions. With over 25 years of industry experience, Moxa has connected more than 30 million devices worldwide and has a distribution and service network that reaches customers in more than 70 countries. Moxa delivers lasting business value by empowering industry with reliable networks and sincere service for automation systems. Information about Moxa's solutions is available at www.moxa.com. You may also contact Moxa by email at info@moxa.com.

How to Contact Moxa

Tel: +886-2-8919-1230
Fax: +886-2-8919-1231

MOXA®
Reliable Networks ▲ Sincere Service

The instructions in this tech note apply only to the Moxa UC-81XX series of computers. These instructions MAY NOT be suitable for use in other computers because of the different Debian distribution versions, hardware peripherals, and firmware versions.

*The instructions in this document require familiarity with the Amazon AWS IoT console. Additional details on the Amazon AWS IoT console are available at: **https://aws.amazon.com/documentation/iot/***

1 Background

The purpose of this document is to provide step-by-step instructions on how to run the Amazon AWS IoT embedded C software development kit (SDK) on the UC-8100-LX computer.

2 Requirement

- UC-8100-LX/ UC-8100-LX-ME
- Amazon AWS IoT account

3 Overview

The UC 8100-LX computer comes preinstalled with the Debian Linux distribution. You should first install all dependent libraries and tools for this distribution, especially the ones for C SDK, before building software packages on the computer.

Once you have installed all the libraries and tools required for this distribution on your computer, follow the instruction given in the sections:

1. Preparing the AWS IoT C SDK environment on the UC-8100-LX
2. Building the C SDK source code
3. Connecting to the AWS IoT console

3.1 Preparing the AWS IoT C SDK Environment on the UC-8100-LX

3.1.1 Libraries and Tools

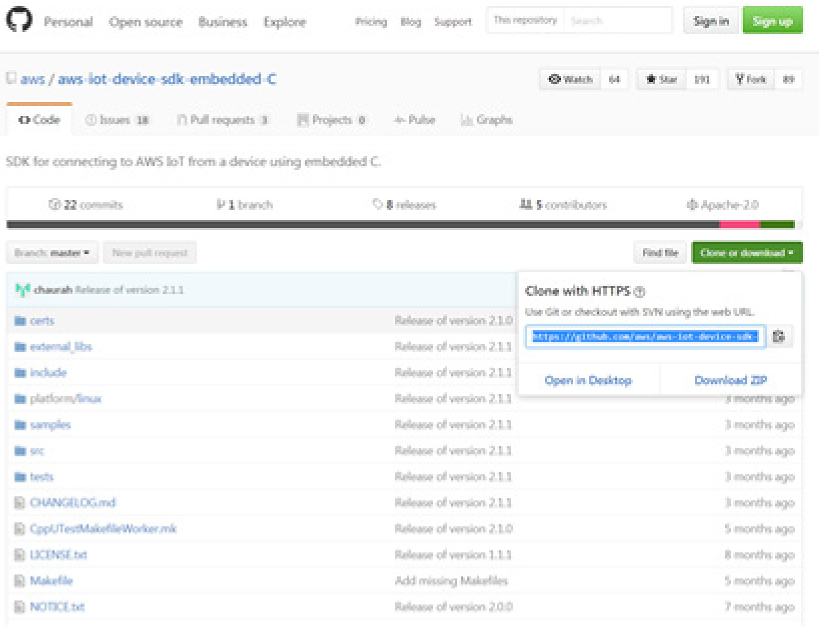
You will need a native compiler environment and SSL libraries to compile the C source code on the UC-8100-LX. Run the following commands on the UC-8100-LX with root authentication to connect to the repository and install the libraries and tools required to compile C code:

```
1. #apt-get update
2. #apt-get upgrade
3. #apt-get install build-essential -y
4. #apt-get install git -y
5. #apt-get install cmake -y
```

3.1.2 Building the C Source Code

Amazon C SDK is available free of cost at:
https://github.com/aws/aws-iot-device-sdk-embedded-C

Figure 1



SDK for connecting to AWS IoT from a device using embedded C.

Use Git or checkout with SVN using the web URL.

https://github.com/aws/aws-iot-device-sdk-

chaurah Release of version 2.1.1

certs	Release of version 2.1.0	
external_libs	Release of version 2.1.1	
include	Release of version 2.1.1	
platform/linux	Release of version 2.1.1	3 months ago
samples	Release of version 2.1.1	3 months ago
src	Release of version 2.1.1	3 months ago
tests	Release of version 2.1.1	3 months ago
CHANGELOG.md	Release of version 2.1.1	3 months ago
CppUTestMakefileWorker.mk	Release of version 2.1.0	5 months ago
LICENSE.txt	Release of version 1.1.1	8 months ago
Makefile	Add missing Makefiles	5 months ago
NOTICE.txt	Release of version 2.0.0	7 months ago

Download the C SDK from the Amazon website and then use the **Git** tool to download the source code and dependent libraries to the UC-8100-LX/home/moxa directory on the computer using the following commands:

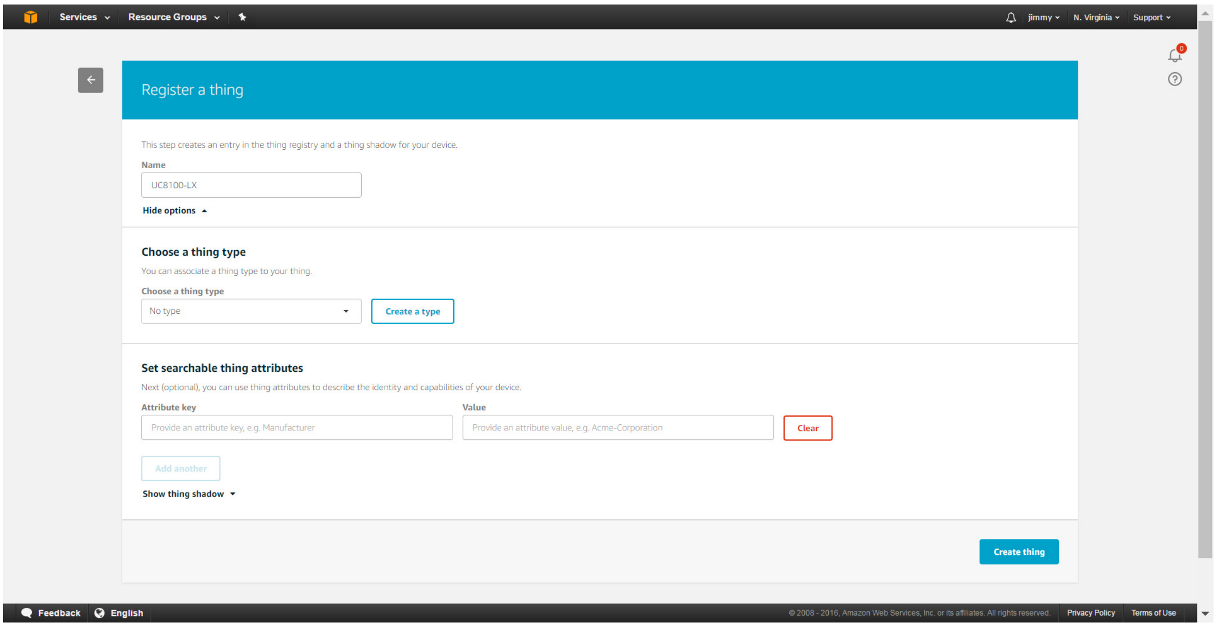
```
1.  #cd /home/moxa
2.  #git clone https://github.com/aws/aws-iot-device-sdk-embedded-C.git
3.  #git clone https://github.com/cpputest/cpputest.git
4.  #cp -a cpputest/* aws-iot-device-sdk-embedded-C/external_libs/CppUTest/
5.  #git clone https://github.com/ARMmbed/mbedtls.git
6.  #cp -a mbedtls/* aws-iot-device-sdk-embedded-C/external_libs/mbedTLS/
7.  #cd aws-iot-device-sdk-embedded-C
8.  #make
```

3.2 Building the C SDK Source Code

Once you have the source code and libraries that you need on your computer, it is time to build the source code samples. Before you go any further, register a *thing* account for your resource with AWS by login in to the **Amazon Management Console** and clicking on the **AWS IoT** link. This will enable you to download certification files on your computer.

The following screenshots show an example for registering an Amazon thing for a UC-8100-LX device (Figure 2 and Figure 3) and downloading the corresponding certification files (Figure 4).

Figure 2



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

Name: UC8100-LX

Choose a thing type

You can associate a thing type to your thing.

Choose a thing type: No type

Set searchable thing attributes

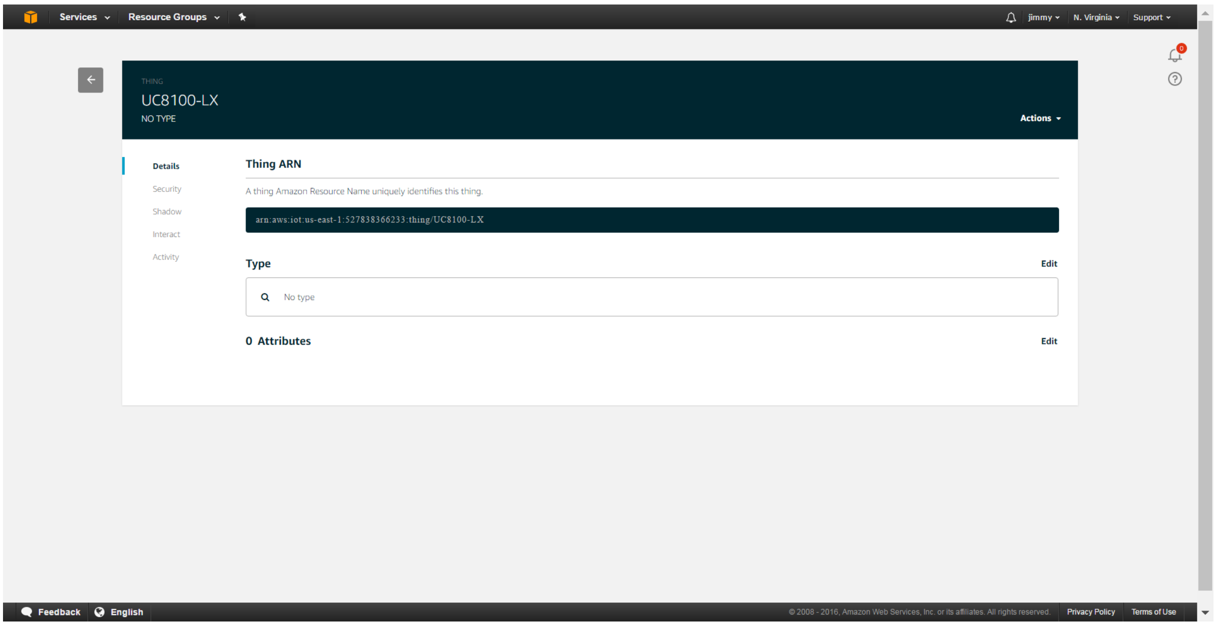
Next (optional), you can use thing attributes to describe the identity and capabilities of your device.

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

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

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 3



THING
UC8100-LX
NO TYPE

Thing ARN

A thing Amazon Resource Name uniquely identifies this thing.

```
arn:aws:iot:us-east-1:527838366233:thing/UC8100-LX
```

Type

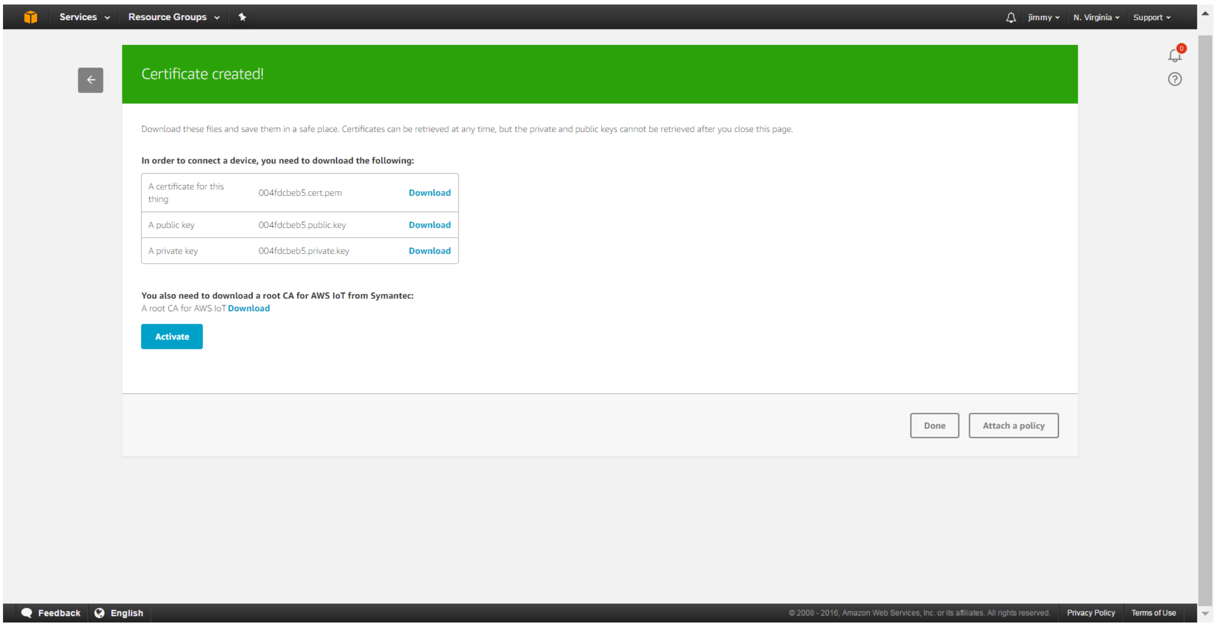
No type

0 Attributes

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Once you have registered a thing object, AWS assigns it a unique resource name and a certificate. You should be able to download the certificate files to your desktop computer via a browser. Use WinSCP or FileZilla (Figure 4) to upload these certificate files to the following Amazon C SDK source code directory on the UC-8100-LX machine: ***[PATH_OF_SOURCE_CODE]*/aws-iot-device-sdk-embedded-C/certs**.

Figure 4



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	004fdcbeb5.cert.pem	Download
A public key	004fdcbeb5.public.key	Download
A private key	004fdcbeb5.private.key	Download

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

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You can modify/edit the sample code header file (aws_iot-config.h) to best fit your thing resource and its policy. Sample header files are available at: https://github.com/aws/aws-iot-device-sdk-embedded-C/tree/master/samples

Run the commands listed below to build your source code:

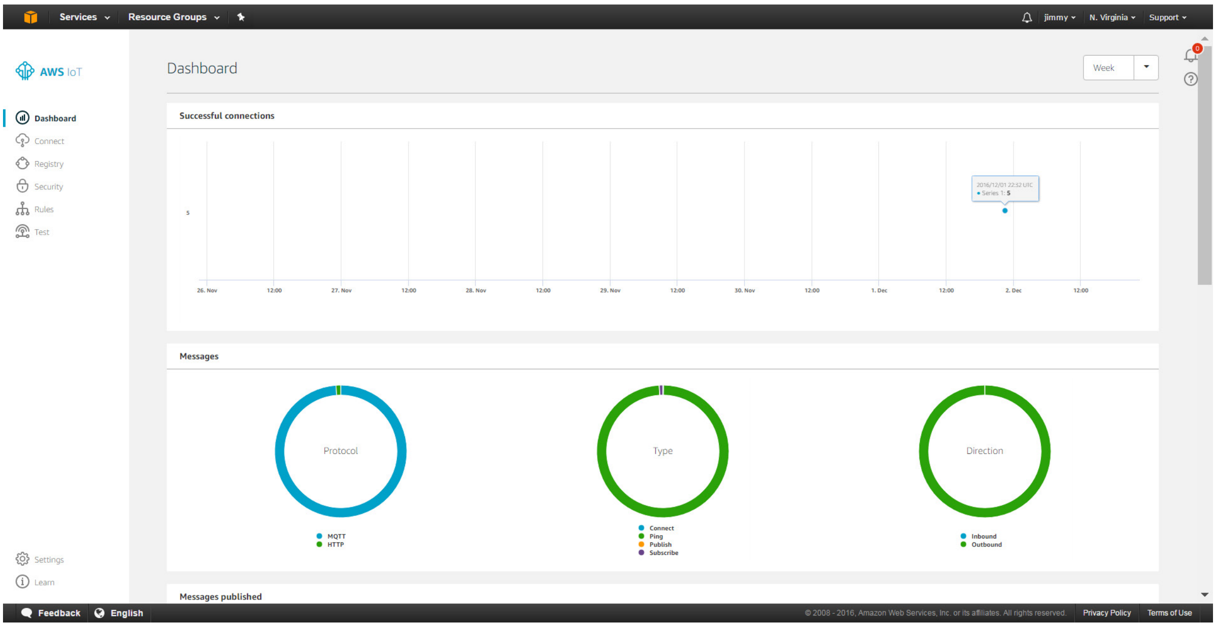
1. `#cd` ***[PATH_OF_SOURCE_CODE]***`/aws-iot-device-sdk-embedded-C`
2. `#make`
3. `#cd` ***[PATH_OF_SOURCE_CODE]***`/aws-iot-device-sdk-embedded-C/samples/linux/subscribe_publish_sample`
4. `#make`

You are now ready to run your sample programs.

3.3 Connecting to the AWS IoT Console

We run one of the sample programs using the `#./subscribe_publish_sample` command to view the AWS IoT Console as shown in Figure 5 below:

Figure 5



You are all set once you see the **Successful connection** status on your AWS IoT dashboard.

4 Additional Reading

- http://docs.aws.amazon.com/iot/latest/developerguide/iot-device-sdk-c.html
- http://www.moxa.com/product/uc-8100.htm