

Getting Started with the AT&T M2X Service on Moxa's UC-81XX Computer

Jimmy Chen
Field Application Engineer
jimmy.chen@moxa.com

Contents

- 1 Background..... 2**
- 2 Requirements 2**
- 3 Overview..... 2**
 - 3.1 Preparing the AT&T M2X Environment on the UC-8100-LX..... 3
 - 3.2 Connecting to the M2X Service..... 5
- 4 Additional Reading..... 8**

Copyright © 2017 Moxa Inc.

Released on January 23, 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



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 AT&T IoT platform. Additional details on the AT&T IoT platform is available at: <https://m2x.att.com>

1 Background

The purpose of this document is to provide step-by-step instructions on how to run the AT&T M2X service using the M2X libraries on Moxa's UC-8100-LX computer.

2 Requirements

- UC-8100-LX/UC-8100-LX-ME
- AT&T IoT Platform account

3 Overview

The UC 8100-LX computer comes preinstalled with the Debian Linux distribution. Debian's powerful package management tool, APT makes it easy for you to install all dependent packages required by the M2X libraries to run the AT&T M2X service on your computer. In our example, we show the simplest way to connect to the M2X service using just a few Linux commands. The UC-8100-LX also supports other languages such as Ruby, Python, C, and Node.js. For code examples in these languages, go to <https://m2x.att.com/developer/tutorials/raspberry>.

Once you have installed all the libraries and tools for the Debian Linux distribution on your computer, follow the instruction in the sections below:

1. Preparing the AT&T M2X environment on the UC-8100-LX
2. Creating a simple bash script
3. Connecting to the M2X service

3.1 Preparing the AT&T M2X Environment on the UC-8100-LX

3.1.1 Libraries and Tools

You will need to install the `curl` tool and library on the UC-8100-LX to be able to connect to the AT&T M2X service (M2X service). Ensure that the UC-8100-LX is connected to the Internet and run the following commands using `root` authentication:

1. `#apt-get update`
2. `#apt-get upgrade`
3. `#apt-get install curl -y`
4. `#apt-get install libcurl3 -y`

These commands will install the `curl` tool and all the dependent libraries.

3.1.2 Creating a Simple Bash Script

Use the Vim editor to create a script with the following content and save it in the `/home/moxa/connect.sh` file as shown in [Figure 1](#):

```
#!/bin/bash

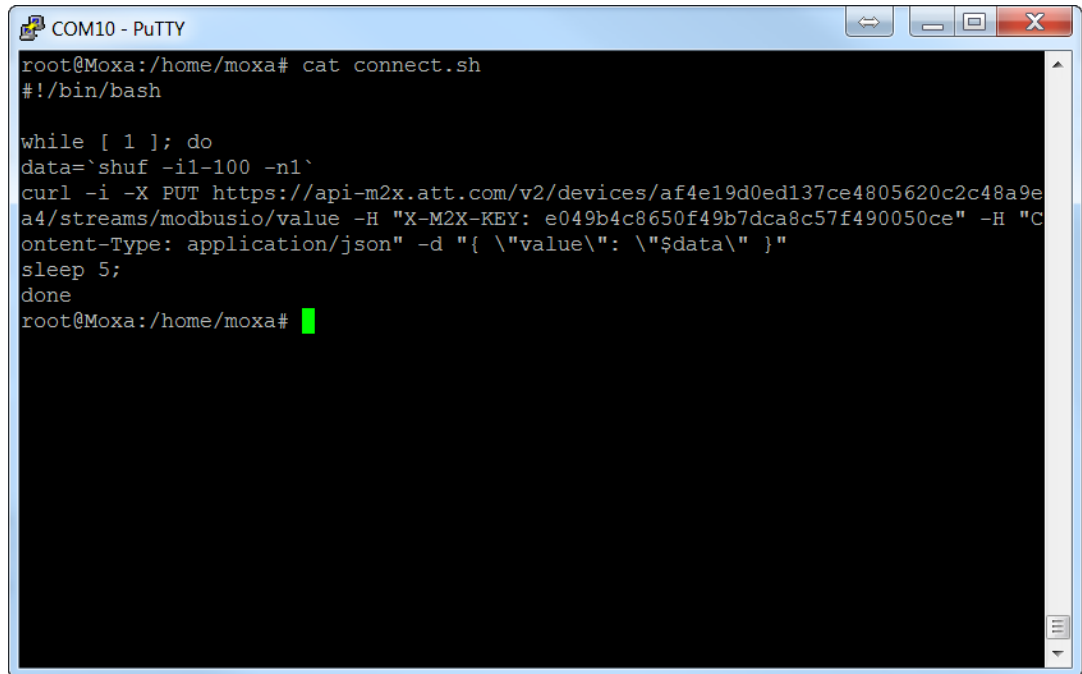
while [1]; do

data='shuf -il-100 -n1'

curl -i -X PUT
https://api-m2x.att.com/v2/devices/af4e19d0ed137ce4805620c2c48a9ea4
/streams/modbusio/value -H "X-M2X-KEY:
e049b4c8650f49b7dca8c57f490050ce" -H "Content-Type: application/json"
-d "{ \"value\": \"$data\" }"

sleep 5;
```

Figure 1



```
COM10 - PuTTY
root@moxa:/home/moxa# cat connect.sh
#!/bin/bash

while [ 1 ]; do
data=`shuf -i1-100 -n1`
curl -i -X PUT https://api-m2x.att.com/v2/devices/af4e19d0ed137ce4805620c2c48a9e
a4/streams/modbusio/value -H "X-M2X-KEY: e049b4c8650f49b7dca8c57f490050ce" -H "C
ontent-Type: application/json" -d "{ \"value\": \"$data\" }"
sleep 5;
done
root@moxa:/home/moxa# █
```

This is a simple script that creates an endless while loop, which is executed every 5 seconds. Each time the program loops, the `curl` command uses the HTTPS PUT command to push data to the M2X URL. The `data` variable holds the data to be used by the `curl` command.

Use the following command to change the script in the `connect.sh` file to an executable:

```
#chmod u+x /home/moxa/connect.sh
```

3.2 Connecting to the M2X Service

After logging into the AT&T IoT Platform, click on the **Create New** link to define a new device, UC8100 as shown in [Figure 2](#) and [Figure 3](#).

Figure 2

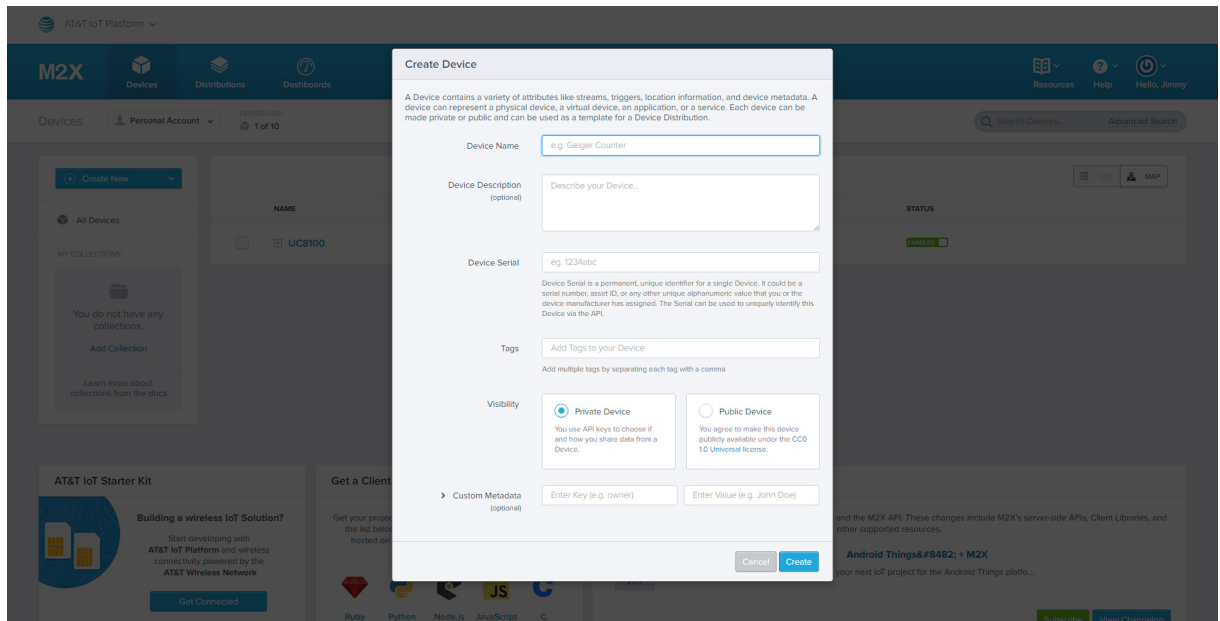
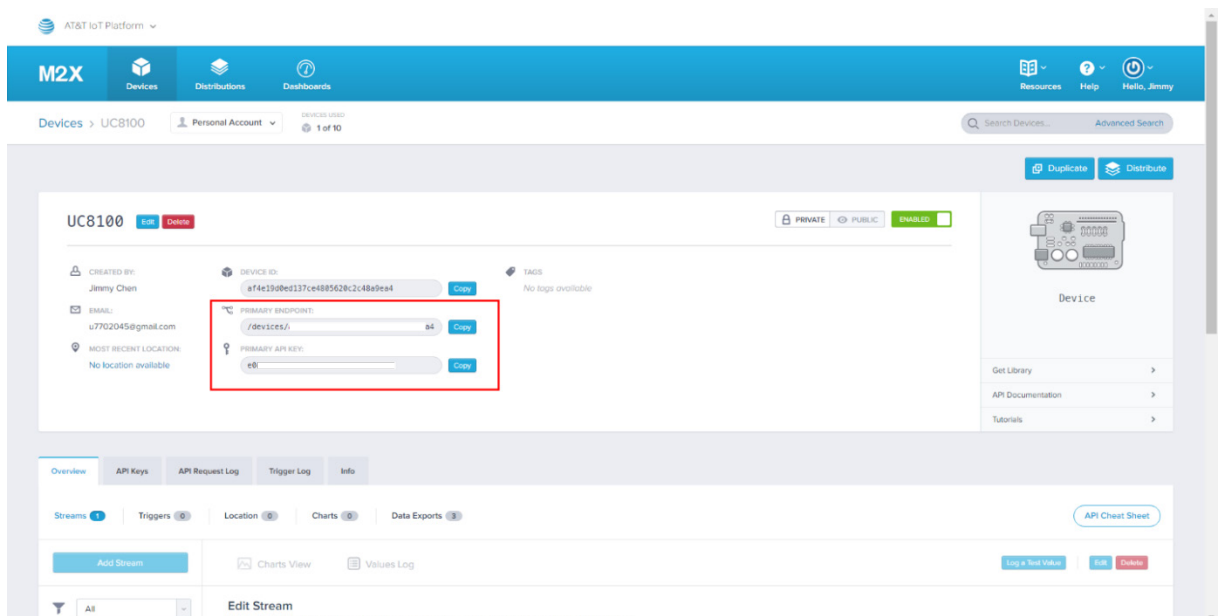
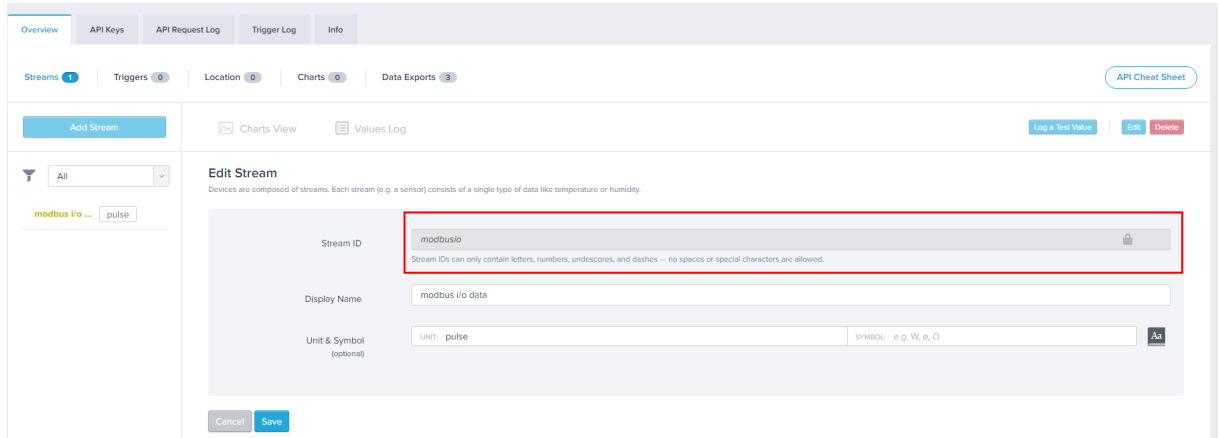


Figure 3



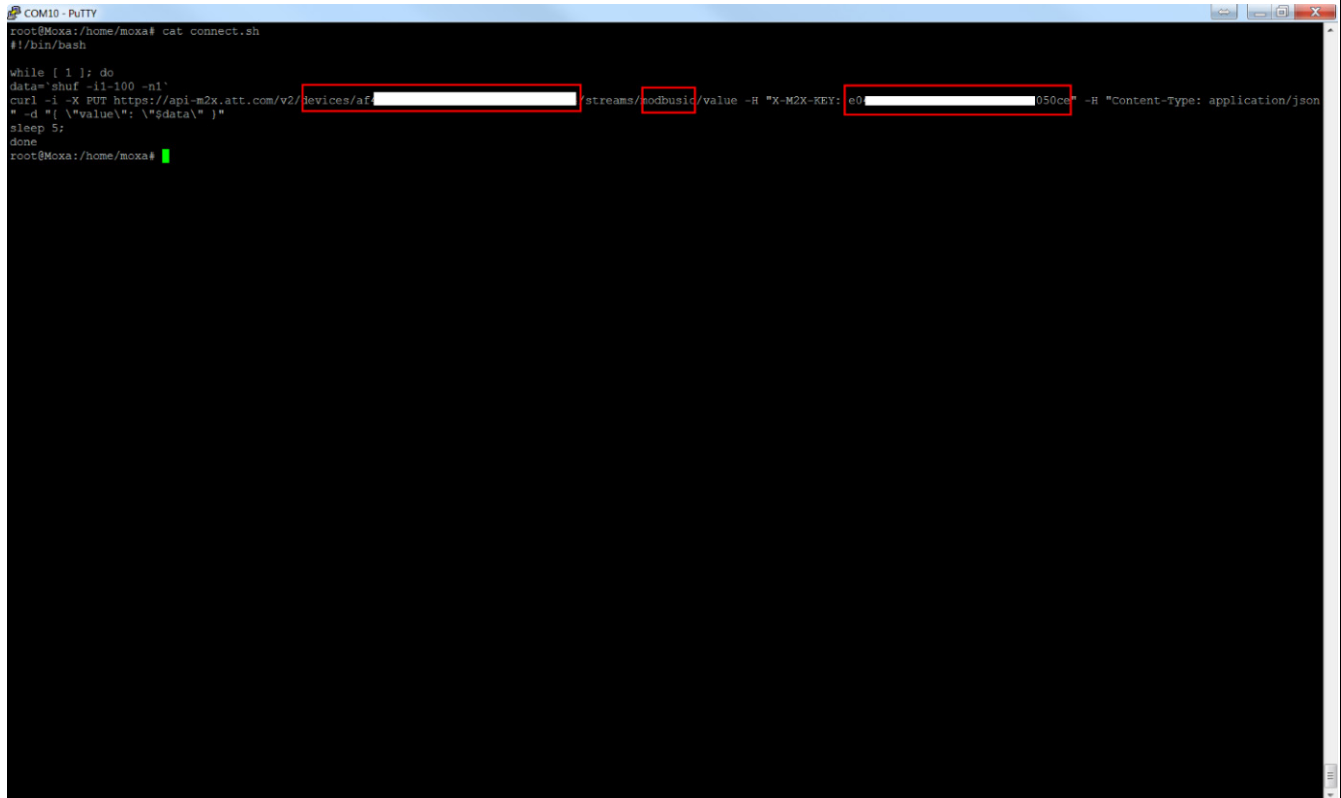
In the **Overview** tab, click **Add Stream** to create the Stream ID, `modbusio` as shown in [Figure 4](#):

Figure 4



Use the information from the uc8100 device to replace the device ID, primary endpoint, and stream id values in the script as indicated in [Figure 5](#).

Figure 5



```
COM10 - PuTTY
root@moxa:/home/moxa# cat connect.sh
#!/bin/bash

while [ 1 ]; do
data=$(shuf -i1-100 -n1)
curl -i -X PUT https://api-m2x.att.com/v2/devices/af/[redacted]/streams/modbusic/value -H "X-M2X-KEY: [redacted]050cc" -H "Content-Type: application/json" -d [{"value": \"$data\"}]
sleep 5;
done
root@moxa:/home/moxa#
```

Run the following commands to execute the `connect.sh` script:

1. `#cd /home/moxa`
2. `#./connect.sh`

You will see the AT&T IoT Platform dashboard with the Modbus I/O data graph as shown in [Figure 6](#).

Figure 6



4 Additional Reading

- <https://m2x.att.com/developer/tutorials/raspberry>
- <http://www.moxa.com/product/uc-8100.htm>