

MYS-6ULX Application Note

Alexa voice assistant

MYiRTM Make Your Idea Real

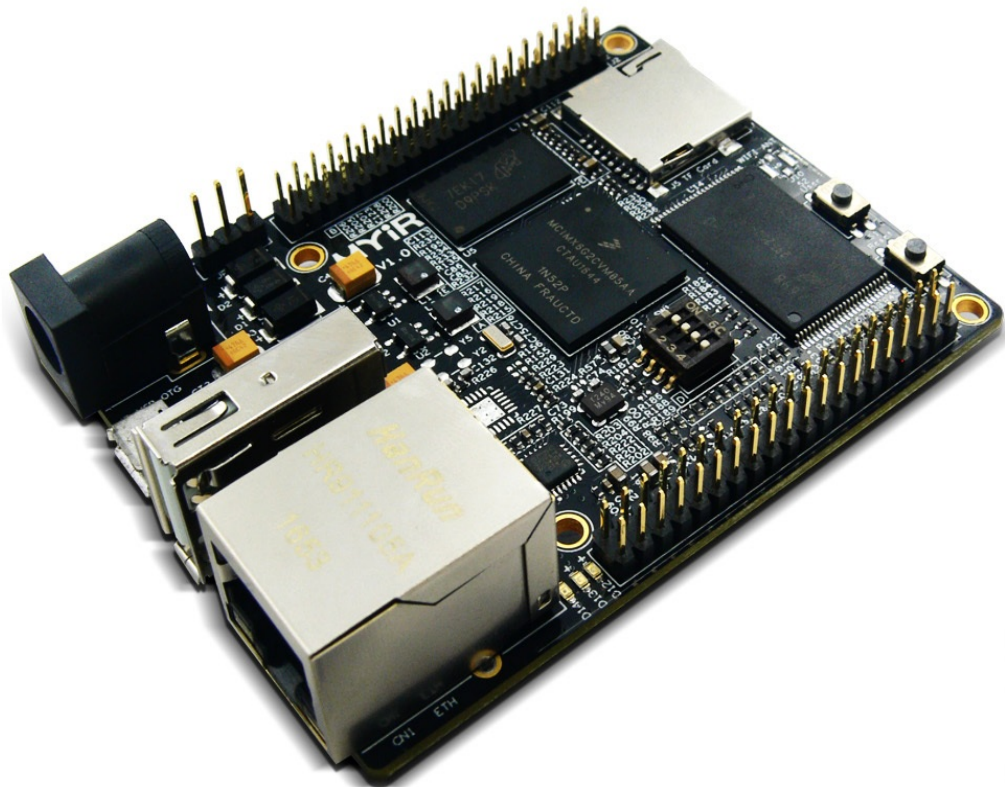


Table of Contents

Preface	0
1 Introduction	1
2 Amazon AVS	2
3 Amazon Alexa Skill	3
4 Amazon Lambda	4
5 Amazon IoT	5
6 Deploy application to MYS-6ULX board	6

MYS-6ULX Amazon Alexa

This document introduce using Amazon Alexa voice assistant on the MYS-6ULX development board and control LED(D12).

Version History

Version	Description	Date
V1.0	Initial version	2017.05.10

Hardware Version

This document suit for MYS-6ULX-IND and MYS-6ULX-IoT board.

Amazon Alexa Voice Application

Introduction

Amazon Alexa is virtual personal assistant. It supports user ability to dicate commands to assistant to control products throughout their home, listen to music, and more.

Alexa service has two services, AVS and ASK.AVS means Alexa Voice Service supports basic voice communication function, such as check weather, play music.ASK means Alexa Skill Kit supports user to define different skill to communicate with other device or service.

This application uses Python as main language.The use must have much experance on Python and Linux programming.

This application need tools:

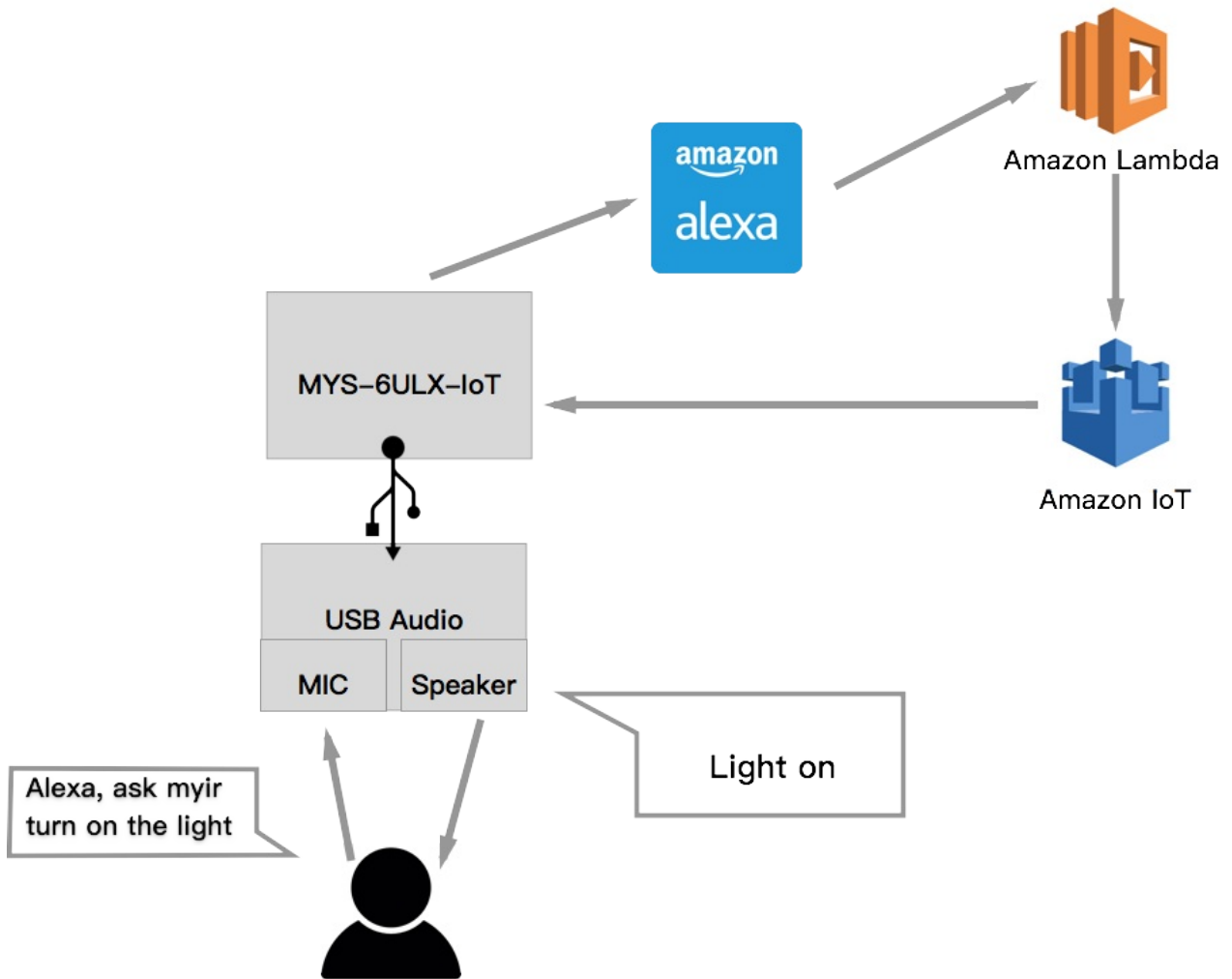
- Amazon developer account
- Amazon Alexa Skill Service
- Amazon Alexa Voice Service
- Amazon Lambda service
- Amazon IoT Service
- Python
- Debian system
- USB Sound Card
- MIC and Speaker

MYS-6ULX already supports Debian image.The image has installed some Python developing libraries and pip tool.Here use a GreenUnion C-Media USB audio codec.You need prepare the Amazon developer account before start.

Application Priceple

The user says "ask myir, turn on the light " to MIC.The python program will record and send to Alexa cloud server, it will parser the sentense follow predefined schema and utterance output the intent json.The json data will be send to Amazon Lambda cloud

service, then parser the intent and update the device of AWS IoT Shadow service. MYS-6ULX has other python program to receive the status of AWS IoT and operating the LED(D12).



2 Amazon Voice Service

Please visit <https://developer.amazon.com/avs/home.html> website and find "Building with Alexa Voice Service" webpage. Click the right corner "Register a Product" popdown button, choose "Device" item, then the bottom show wizard webpage.

The left panel has three menus "Device Type Info", "Security Profile" and "Device Detail".

- Device Type Info menu: "Device Type ID" fill ID of create device, this application use "mys6ulx". The "Display name" is for end user, also fill "mys6ulx".
- Security Profile menu: "Security Profile" choose "Create a new profile". The "Security Profile Name" value is "mys6ulx-profile", the "Security Profile Description" value is "mys6ulx security profile". And click right corner "Next" button, it will generate three ID values. "Web Settings" tab is URL address of MYS-6ULX, "Allowed Return URLs" such as "<http://192.168.1.165:5000/code>". The Alexa will allow device get "refresh token" from the URL.

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to [Learn More](#). If you are building an Android or iOS app, click here to [Learn More](#).

Security Profile ? *

A security profile is how Amazon identifies your device. Edit

General | Web Settings | Android/Kindle Settings | iOS Settings

Security Profile Description
Choose a description for your security profile for Amazon services to use in communicating with you. **mys6ulx security profile**

Security Profile ID
This ID will identify your security profile in Amazon services. **amzn1.application.4cb41cbc31934f3c9dfa30ad1f1eb62c**

Client ID ?
This is a value specific to you that is assigned to you when you register with Login with Amazon. **amzn1.application-0a2-client.0f4b5c3916534379b3c059de5b9725c1**

Client Secret ?
This is a secret specific to you that is assigned to you when you register with Login with Amazon. Confidential. **893f7d43d59105abaefef1b73fb638f8e8ddfbf4991416c4575a19e62fc8f590**

* Fields required

[AVS Program](#)
[AVS Content F](#)

You need a security profile to identify your device. Your security profile credentials - client ID and client secret - allow your device to securely identify itself to the Alexa Voice Service. If you are building a website, click here to [Learn More](#). If you are building an Android or iOS app, click here to [Learn More](#).

Security Profile ? *

A security profile is how Amazon identifies your device.

mys6ulx-profile

Edit

General

Web Settings

Android/Kindle Settings

iOS Settings

Allowed Origins ?

Your website origin, when using Login with Amazon.

Allowed Return URLs ?

If you make HTTPs calls to Login with Amazon with redirect_uris, specify them here.

http://192.168.1.165:5000/code

- Device Details: "Category" choose "Home appliance", "Description" fill with "mys6ulx"

3 Amazon Alexa Skill

Create Amazon Alexa Skill

Firstly, use Amazon developer account to login and visit <https://developer.amazon.com/edw/home.html#/skills>, click "Add a New Skill " on the right corner, and will show "Create a New Alexa Skill" webpage. The left panel is all process of ASK, right side is detail information on each item.

- Skill Information: Some basic information
- Interaction Model: Interactive information
- Configuration: Setting Alexa action when receive voice command
- Test: Test the existing model
- Publishing Information: Information on publishing Skill
- Privacy & Compliance: Setting privacy options

Skill Information

- Skill type choose "Custom Interaction Model"
- Name is Skill name, example use "mys6ulx"
- Invocation Name is a wakeup word, example use "myir"

English (U.S.)
Add a New Language

Skill Information	Skill Type Define a custom interaction model or use one of the predefined skill APIs. Learn more Custom
Interaction Model	Language Language of your skill English (U.S.)
Configuration	Application Id The ID for this skill amzn1.ask.skill.9cdd8de5-7107-4d48-a38c-9aed3af23a9c
Test	Name Name of the skill that is displayed to customers in the Alexa app. Must be between 2-50 characters. <input style="width: 80%;" type="text" value="mys6ulx"/>
Publishing Information	Invocation Name The name customers use to activate the skill. For example, "Alexa ask Tide Pooler...". Invocation Name Guidelines <input style="width: 80%;" type="text" value="myir"/>
Privacy & Compliance	Global Fields

Skills Beta Testing NEW

Status: NOT STARTED

Interaction Model

Intent Schema used config Intent.

```
{
  "intents": [
    {
      "intent": "LightOn"
    },
    {
      "intent": "LightOff"
    }
  ]
}
```

Generally, a Skill have more Intents

English (U.S.) ✔

Add a New Language

Try the skill builder (beta), an intuitive interface for building interaction model and creating dialog prompts.

Launch Skill Builder BETA

Skill Information ✔
Interaction Model ✔
Configuration ✔
Test ✔
Publishing Information ✔
Privacy & Compliance ✔

Intent Schema

The schema of user intents in JSON format. For more information, see [Intent Schema](#).
Also see [built-in slots](#) and [built-in intents](#).

```

1 {
2   "intents": [
3     {
4       "intent": "LightOn"
5     },
6     {
7       "intent": "LightOff"
8     }
9   ]
10 }
```

Skills Beta Testing NEW

Status: NOT STARTED

☆ Beta Test Your Skill

The Sample Utterances used match Intent and trigger action when user speaks.

```
LightOn turn on the light
LightOn turn on light
LightOn light on

LightOff turn off the light
LightOff turn off light
LightOff light off
```

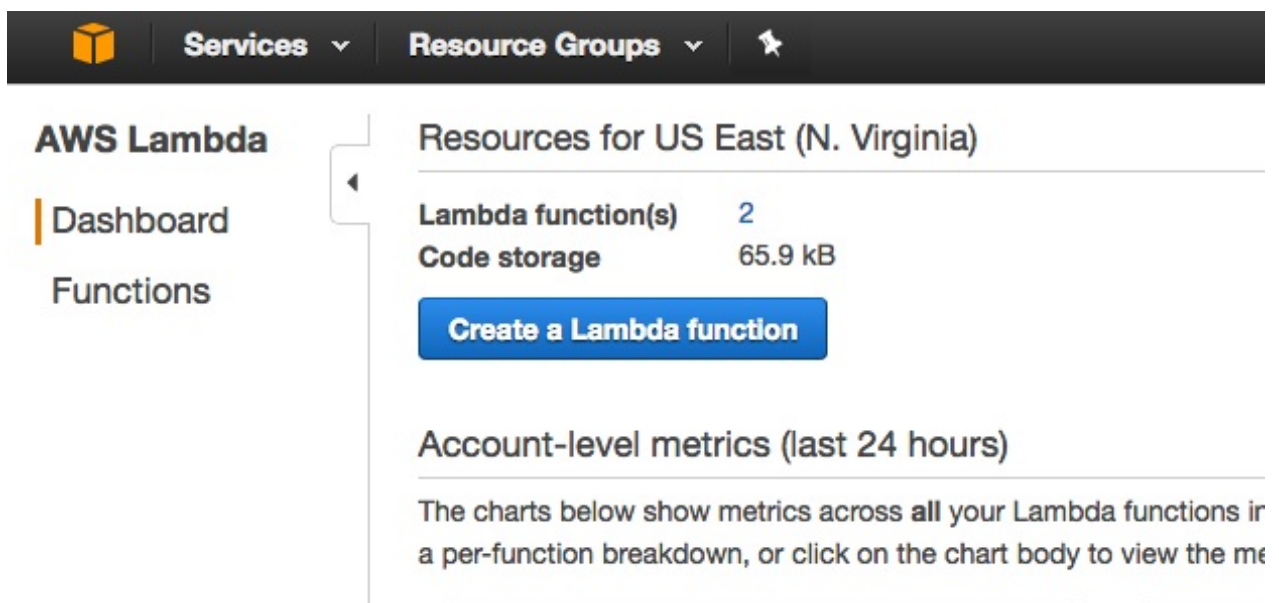
Configuration

This webpage config the Endpoint that used receive and parse Alexa Intent. This application use AWS Lambda service. You just fill the Lambda ARN URL to input box.

4 AWS Lambda

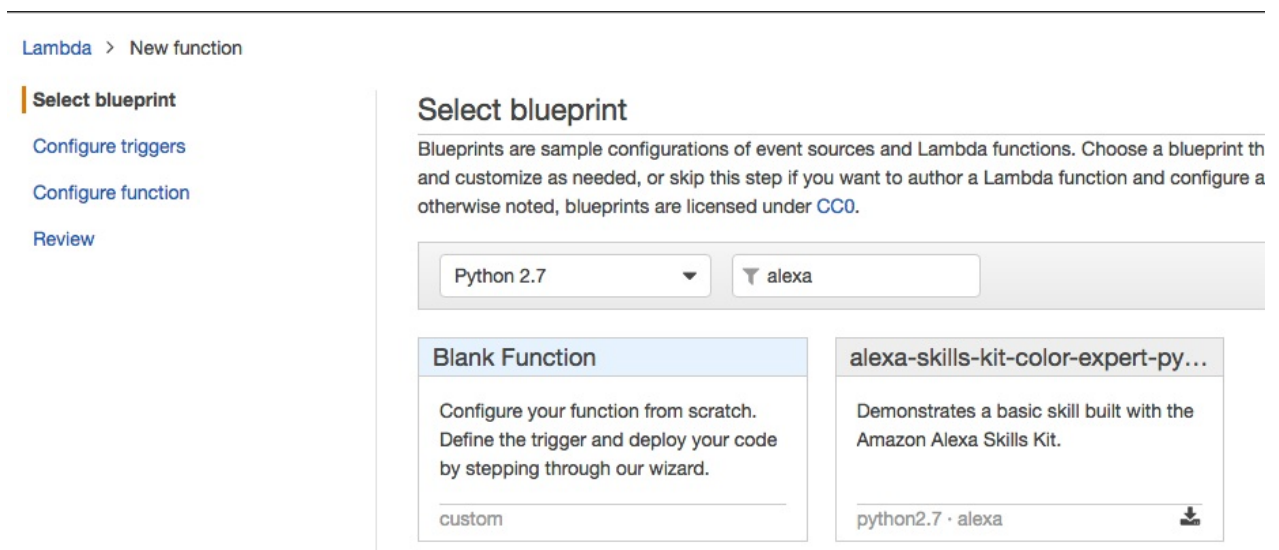
Visit AWS Console webpage, <https://console.aws.amazon.com/lambda/home>, Alexa only support us-east-1 region. Please choose N. Virginia.

Click "Create a Lambda function" button. In "Select blueprint" webpage, through filter to find Alexa. Then choose "alexa-skills-kit-color-expert-python". Switch to "Configure triggers" webpage, config the default trigger source is Alexa Skills Kit, and click "Next" button.



The screenshot shows the AWS Lambda console interface. At the top, there are navigation tabs for 'Services' and 'Resource Groups'. The main content area is titled 'Resources for US East (N. Virginia)'. It displays a summary of resources: 'Lambda function(s)' with a value of 2, and 'Code storage' with a value of 65.9 kB. A large blue button labeled 'Create a Lambda function' is centered below the summary. Underneath, there is a section for 'Account-level metrics (last 24 hours)' with a brief description: 'The charts below show metrics across all your Lambda functions in a per-function breakdown, or click on the chart body to view the metrics'.

"Configure function" webpage, the Lambda create sample code for Alexa. Now, we just skip it until upload our code lately.



The screenshot shows the 'Select blueprint' page in the AWS Lambda console. At the top left, there is a breadcrumb 'Lambda > New function'. On the left side, there is a sidebar with four options: 'Select blueprint' (highlighted), 'Configure triggers', 'Configure function', and 'Review'. The main content area is titled 'Select blueprint' and includes a description: 'Blueprints are sample configurations of event sources and Lambda functions. Choose a blueprint that you want to use as a starting point, and customize as needed, or skip this step if you want to author a Lambda function and configure it as otherwise noted, blueprints are licensed under CC0.' Below the description, there are two dropdown menus: 'Python 2.7' and 'alexa'. Below these, there are two blueprint options: 'Blank Function' and 'alexa-skills-kit-color-expert-py...'. The 'Blank Function' option has a description: 'Configure your function from scratch. Define the trigger and deploy your code by stepping through our wizard.' and a 'custom' label. The 'alexa-skills-kit-color-expert-py...' option has a description: 'Demonstrates a basic skill built with the Amazon Alexa Skills Kit.' and a 'python2.7 · alexa' label with a download icon.

Next we config Lambda function Handler and Role. The "Handler" part, keep default value. The "Lambda function handler and role" part setting relative with Lambda permission. In here, we just choose "Role" popdown item, choose "Create a custom role".

The screenshot shows the 'Lambda function handler and role' configuration section in the AWS Lambda console. At the top, there is a checkbox for 'Enable encryption helpers' which is unchecked. Below it is the 'Environment variables' section with a table with two columns: 'Key' and 'Value'. The main section is titled 'Lambda function handler and role' and contains three fields: 'Handler*' with the value 'lambda_function.lambda_handler', 'Role*' with a dropdown menu showing 'Choose an existing role', and 'Existing role*' with an empty dropdown menu. Each field has an information icon to its right. At the bottom left of this section is a 'Tags' link with a right-pointing arrow.

It shows new window "AWS Lambda requires access to your resources", fill the item "Role Name" as "led_control". Click "View Policy Document" toggle button and click "Edit" button, add below code:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:*"
      ],
      "Resource": "arn:aws:iot:*:*:*"
    }
  ]
}
```

These code allow Lambda visit AWS IoT resource, then click "Allow" button.

The last is Review webpage, just click "Create function" button. Waiting webpage returned, the Lambda function has created. The right corner ARN address is Endpoint of current Lambda function, it needs by Alexa setting webpage.

Code: <https://github.com/MYiR-Dev/Alexa-AWSLambda-demo>

Download and fill AWSIoT information into config.py file. And package those files as zip format and upload to Lambda.

```
git clone https://github.com/aws/aws-iot-device-sdk-python.git
git clone https://github.com/MYiR-Dev/Alexa-AWSLambda-demo
cd Alexa-AWSLambda-demo
cp -r ../aws-iot-device-sdk-python/AWSIoTPythonSDK .
mkdir cert
cp config.example.py config.py
zip Archive.zip ./*
```

The zip file list:

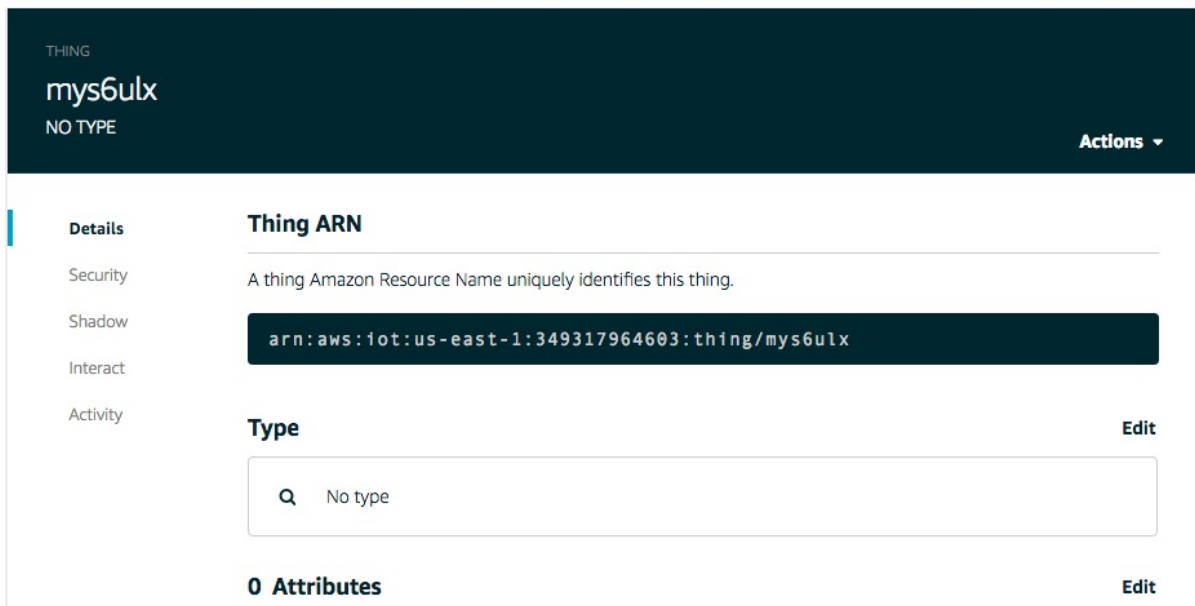
```
├─ AWSIoTPythonSDK
│  └─ MQTTLib.py
│  └─ __init__.py
│  └─ core
│  └─ exception
├─ README.md
├─ cert
│  └─ *-certificate.pem.crt
│  └─ *-private.pem.key
│  └─ root-CA.crt
├─ config.example.py
├─ config.py
└─ main.py
```

5 AWS IoT

AWS IoT service is connect to the device with other AWS services, such as store data, process action.

Visit <https://console.aws.amazon.com/iotv2/home>, enter AWS IoT DashBoard webpage.Below is create Things steps:

- Click left menu "Registry" and popdown "Things" item, click right "Create" button.
- Input the "mys6ulx" as name, click "Create Things" button will finish it.The left menu has different types information.



The screenshot displays the AWS IoT console interface for a specific Thing. At the top, the header shows 'THING' and the name 'mys6ulx' with 'NO TYPE' below it. An 'Actions' dropdown menu is visible in the top right corner. On the left, a navigation menu lists 'Details', 'Security', 'Shadow', 'Interact', and 'Activity'. The main content area is titled 'Thing ARN' and includes a description: 'A thing Amazon Resource Name uniquely identifies this thing.' Below this, the ARN is displayed in a dark box: 'arn:aws:iot:us-east-1:349317964603:thing/mys6ulx'. Underneath, the 'Type' field is shown with a search icon and the text 'No type', accompanied by an 'Edit' button. At the bottom, it indicates '0 Attributes' with another 'Edit' button.

- Click left menu "Security" and popdown "Certificates" item, click right "Create" button. Then click "create certificate" item in "Create a certificate" webpage.

Create a certificate

A certificate is used to authenticate your device's connection to AWS IoT.

One-click certificate creation (recommended)
This will generate a certificate, public key, and private key using AWS IoT's certificate authority. [Create certificate](#)

Create with CSR
Upload your own certificate signing request (CSR) based on a private key you own. [Create with CSR](#)

Use my certificate
Register your CA certificate and use your own certificates for one or many devices. [Get started](#)

- After that will show the "Certificate created!" webpage, and all "certificate" files were created. Download all files, include "certificate" and rootCA file, then click "Active" button.

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	f1fd9c723f.cert.pem	Download
A public key	f1fd9c723f.public.key	Download
A private key	f1fd9c723f.private.key	Download

You also need to download a root CA for AWS IoT from Symantec:
A root CA for AWS IoT [Download](#)

[Activate](#)

- Click left menu "Security" and popdown "Policies" item, click right "Create" button. In "Create a policy" webpage fill Name is "mys6ulx", "Action" is "iot:", "Resource ARN" is "arn:aws:iot:us-east-1:349317964603:topic/", "Effect" choose "Allow", click "Create" button.

POLICY

mys6ulx

Actions ▾

Overview

Certificates

Versions

Policy ARN

A policy ARN uniquely identifies this policy. [Learn more](#)

```
arn:aws:iot:us-east-1:349317964603:policy/mys6ulx
```

Policy document

The policy document defines the privileges of the request. [Learn more](#)

After those steps you will get below information:

- Thing ARN address
- Shadow ARN address
- Thing Shadow Rest API Endpoint
- MQTT topic
- rootCA
- public certificate file
- private certificate file

6 Deploy application to MYS-6ULX board

MYS-6ULX board has two programs, Alexa-MYS-6ULX and Alexa-MYS-6ULX-mqtt. The Alexa-MYS-6ULX used process voice related. The Alexa-MYS-6ULX-mqtt used receive command from AWS IoT and operate LED(D12).

MYS-6ULX Alexa

Code: <https://github.com/MYiR-Dev/Alexa-MYS-6ULX>

Download and copy to board, execute below command to install required libraries:

```
apt-get update
apt-get install libasound2-dev memcached python-pip
pip install -r requirements.txt
```

The program interactive with Alexa needs token. Firstly need fill the IP address of MYS-6ULX into Alexa AVS webpage on "Security Profile" -> "Allowed Return URLs" option. Then running the auth_web.py program.

```
cp example_creds.py creds.py
python auth_web.py
```

After auth_web.py run, use browser to visit the URL of auth_web.py service on PC, eg: "192.168.1.162:5000". The browser will jump to Amazon verify webpage, login the Amazon developer account and click "Verify" button. The webpage will show the refresh token string. Copy and put into variable refresh_token of creds.py file.

Now, the program can interactive with Alexa.

```
python main.py
```

The key(S2) will trigger to record and send to Alexa service. Press it and say something and release.

MYS-6ULX Amazon IoT

Code: <https://github.com/MYiR-Dev/Alexa-MYS-6ULX-mqtt>

The Alexa-MYS-6ULX communicate with AWS IoT service.

Download and execute below commands to install requirement libraries:

```
pip install -r requirements.txt
apt-get install dnsutils mpg123
```

Communicate AWS IoT needs some key and host address. Those information need fill into config.py file.

```
cp example.config.py config.py
```

The Alexa has supports two methods to help developer test program.

- In AWS IoT webpage send command and whether Alexa-MYS-6ULX-mqtt receive or not.
- In Alexa Skill Kit webpage input sentence and will trigger AWS Lambda to test AWS IoT and Alexa-MYS-6ULX-mqtt are correctly.

After tests are successfully, you can running it as background with Alexa-MYS-6ULX to control LED(D12).