

# 大联大商贸（深圳）有限公司

## 品佳集团

---

### LPC54018 Iot 例程概述

编号：SAC-118732-1807

版本：1.0

编写部门：TSC2410

编写人：will

审核人：peter

2018 年 4 月 13 日



## 修订记录

[illegible]

目 录

1	前言 .....	1
1.1	说明 .....	1
1.2	开发板概述 .....	1
1.3	例程准备 .....	1
1.3.1	NXP 软件环境准备 .....	1
1.3.2	AWS 相关账户 .....	1
2	例程运行 .....	8
2.1	环境准备 .....	8
2.2	导入编译工程 .....	8
2.2.1	导入工程 .....	8
2.2.2	工程修改 .....	9
2.2.3	编译工程 .....	11
2.3	调试 .....	12
2.4	其它说明 .....	16

# LPC54018 Iot 例程概述

## 1 前言

### 1.1 说明

本文对 NXP 官方“[https://www.nxp.com/support/developer-resources/reference-designs/lpc54018-iot-module-for-the-lpc540xx-family-of-mcus:0M40007?tab=In-Depth\\_Tab](https://www.nxp.com/support/developer-resources/reference-designs/lpc54018-iot-module-for-the-lpc540xx-family-of-mcus:0M40007?tab=In-Depth_Tab)”（文中简称例程网址）资料进行中文简述及补充部分细节信息，方便用户运行例程。文中部分内容引用了“Amazon FreeRTOS - User Guide”文档。

### 1.2 开发板概述

恩智浦与 Embedded Artists 合作开发的 LPC54018 物联网模块是一款高性能的自给自足微控制器模块，支持 IEEE802.11，适用于开发基于 LPC540xx MCU 系列的产品。



图 1 LPC54018 IoT module

### 1.3 例程准备

#### 1.3.1 NXP 软件环境准备

##### 1.3.1.1 下载 SDK

从例程网址（[https://www.nxp.com/support/developer-resources/reference-designs/lpc54018-iot-module-for-the-lpc540xx-family-of-mcus:0M40007?tab=In-Depth\\_Tab](https://www.nxp.com/support/developer-resources/reference-designs/lpc54018-iot-module-for-the-lpc540xx-family-of-mcus:0M40007?tab=In-Depth_Tab)）下载需要的 SDK。

在例程网址的“1.2”部分，点击图 2 中的“Get MCUXpresso SDK”等待 SDK 下载完成。



图 2 下载 SDK

##### 1.3.1.2 安装 IDE

若您已安装 MCUXpresso IDE，则忽略此步骤；

在例程网址的“1.3”部分，点击图 3 中的“Get MCUXpresso IDE”，等待下载完成后，按一般的 windows 程序完成安装。



图 3 下载 IDE

#### 1.3.2 AWS 相关账户

##### 1.3.2.1 AWS 账户

打开网址：“<https://aws.amazon.com/>”，如图 4 所示，点击“Create AWS Account”；

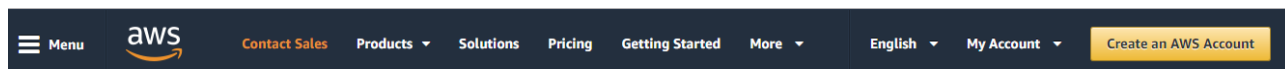
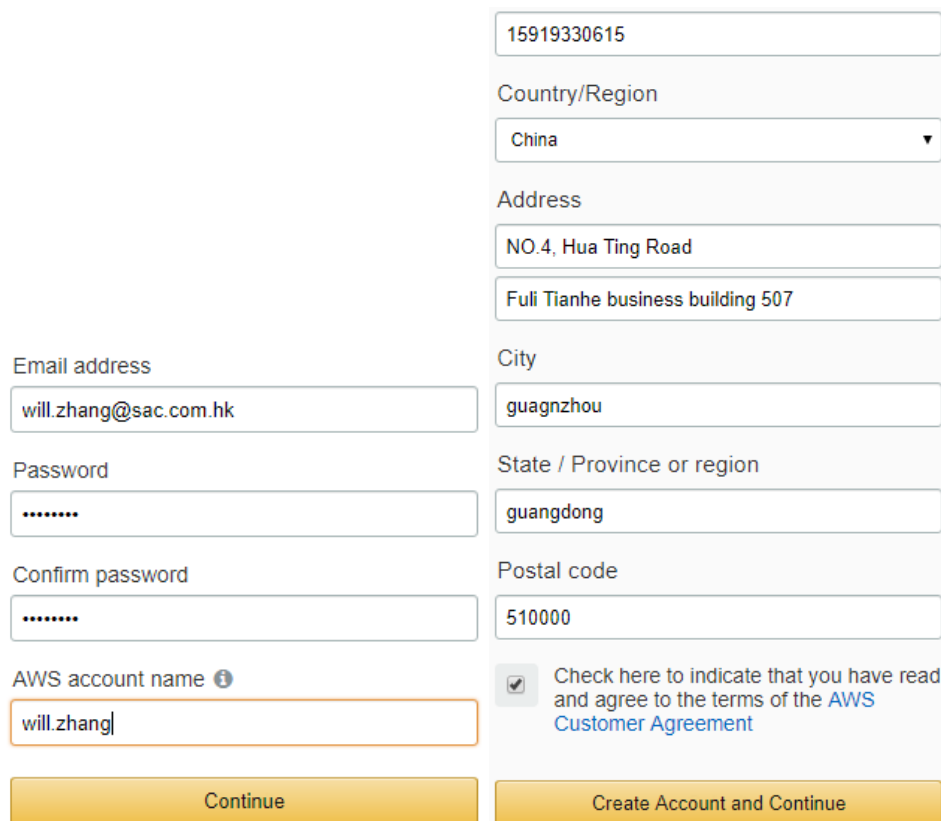


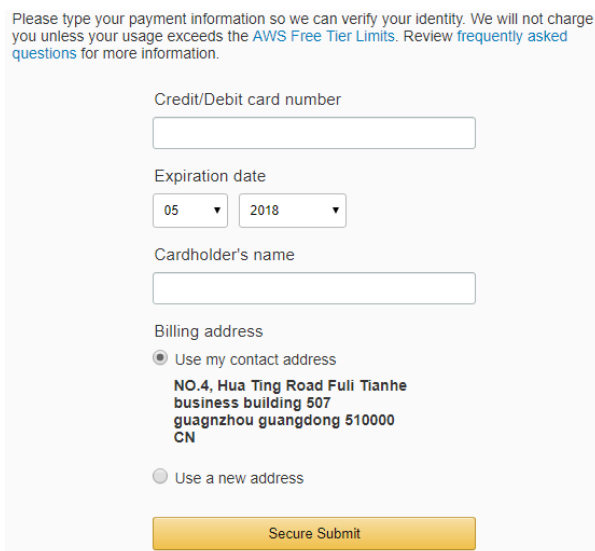
图 4 AWS 网址

按要求填写相关资料（根据自己的资料填写），如图 5；在账户使用前需要验证银行账号，该步骤会产生\$1.0 的消费，如图 6；银行账户验证后，会对手机进行验证（手机收到验证码，填入即可）。



The form is divided into two main sections. The left section contains fields for 'Email address' (will.zhang@sac.com.hk), 'Password' (masked with dots), 'Confirm password' (masked with dots), and 'AWS account name' (will.zhang). The right section contains fields for 'Phone number' (15919330615), 'Country/Region' (China), 'Address' (NO.4, Hua Ting Road, Fuli Tianhe business building 507), 'City' (guangzhou), 'State / Province or region' (guangdong), and 'Postal code' (510000). At the bottom right, there is a checkbox for 'Check here to indicate that you have read and agree to the terms of the AWS Customer Agreement'. Two buttons are at the bottom: 'Continue' and 'Create Account and Continue'.

图 5 AWS 注册账户填写资料



The form is titled 'Please type your payment information so we can verify your identity. We will not charge you unless your usage exceeds the AWS Free Tier Limits. Review frequently asked questions for more information.' It contains fields for 'Credit/Debit card number', 'Expiration date' (05 / 2018), and 'Cardholder's name'. Below these is the 'Billing address' section with a radio button selected for 'Use my contact address', showing the address: 'NO.4, Hua Ting Road Fuli Tianhe business building 507, guangzhou guangdong 510000, CN'. There is also an option for 'Use a new address'. A 'Secure Submit' button is at the bottom.

图 6 AWS 注册账户验证

### 1.3.2.2 IAM user

#### 1.3.2.2.1 创建用户

打开网址：“https://console.aws.amazon.com/iam/home”，使用上一小节中创建的账户登陆；在网页左侧导航栏点击“Users”，在打开的网页中点击“Add user”，如图 7；

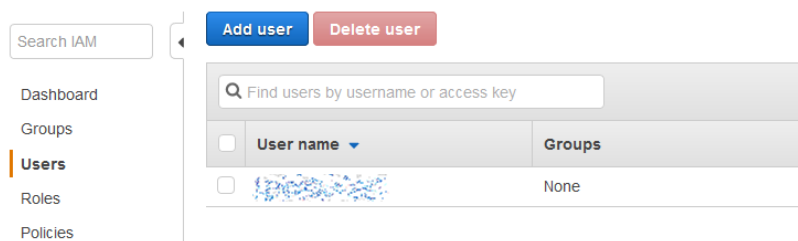


图 7 添加用户

在打开的网页中, 填入用户名如“Administrator”, Access type 选择“AWS Management Console access”, Console Password 选择“custom password”, 并填写密码; 点击“Next: Permissions”, 如图 8。

**Add user**

1 2 3 4

**Set user details**

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name\*

[Add another user](#)

**Select AWS access type**

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type\* ☐ Programmatic access  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☒ **AWS Management Console access**  
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password\* ☐ Autogenerated password  
☒ Custom password  
  
☐ Show password

Require password reset ☒ User must create a new password at next sign-in

\* Required

[Cancel](#) [Next: Permissions](#)

图 8 添加用户设置信息

### 1.3.2.2.2 用户权限设置

用户创建后, 在后续页面中选择“Attach existing policies directly”, 并在搜索框中输入“AmazonFreeRTOSFullAccess”, 勾选搜索结果, 如图 9;

以相同方式勾选“AWSIoTFullAccess”, “AdministratorAccess”;

**Grant permissions**

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

Attach one or more existing policies directly to the users or create a new policy. [Learn more](#)

[Create policy](#) [Refresh](#)

Filter: Policy type

	Policy name	Type	Attachments	Description
<input checked="" type="checkbox"/>	AmazonFreeRTOSFullAccess	AWS managed	0	Full Access Policy for Amazon FreeRTOS

图 9 添加用户设置用户权限

在新页面中确认用户已拥有“AmazonFreeRTOSFullAccess”、“AWSIoTFullAccess”、“AdministratorAccess”权限, 如图 10, 点击“Create user”;

Add user

1

2

3


4

Review

Review your choices. After you create the user, you can view and download the autogenerated password and access key.

User details

User name



AWS access type

AWS Management Console access - with a password

Console password type

Custom

Require password reset

Yes

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	<a href="#">AmazonFreeRTOSFullAccess</a>
Managed policy	<a href="#">AdministratorAccess</a>
Managed policy	<a href="#">AWSIoTFullAccess</a>

图 10 添加用户设置用户权限完成

完成创建用户后，记录图 11 所示的登陆地址，退出当前的账户，使用新建的用户登陆控制台。

图 11 完成添加用户

### 1.3.2.3 创建策略

使用 IAM User 创建的用户登陆: "http://console.aws.amazon.com/iot";

如果是第一次登陆，点击页面中的“Get started”，如图 12；

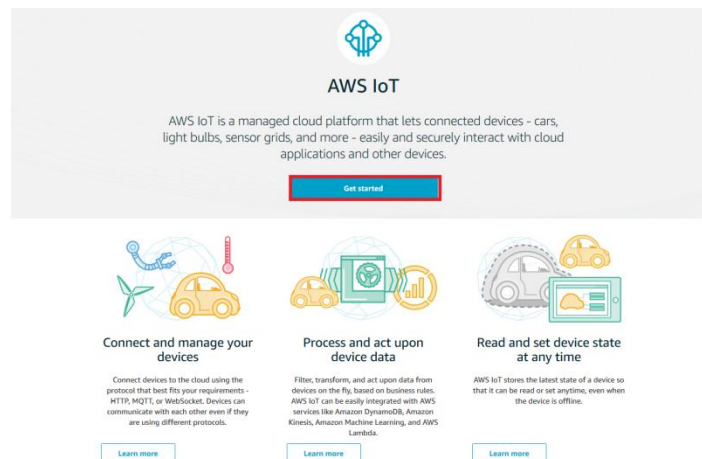


图 12 准备创建策略

在新页面的左侧导航栏中点击“Secure”-->“Policies”;如果是首次创建,则会出现图 13 所示界面,点击“Create a policy”;



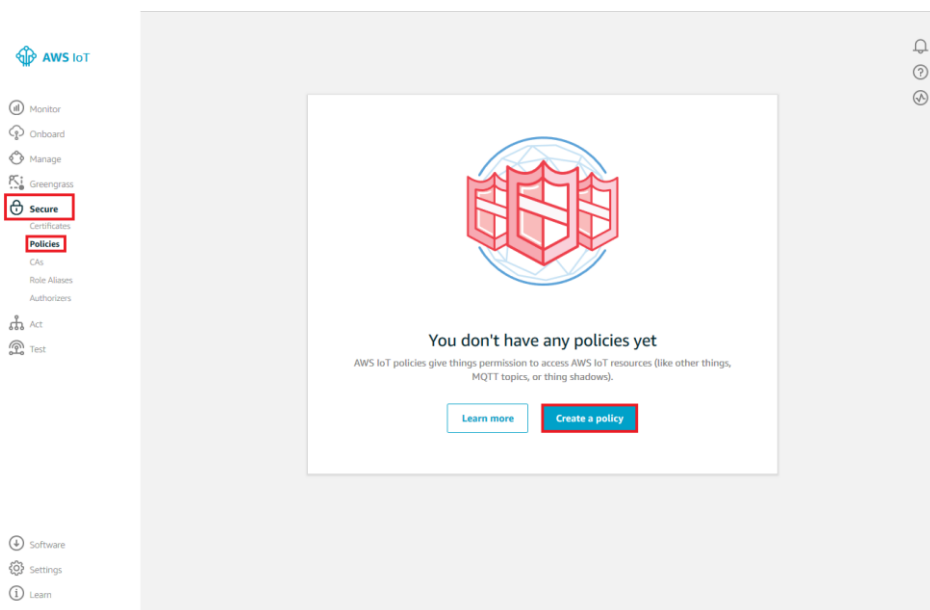


图 13 创建策略

在创建策略页面，填入策略名称如“myIoTPolicy”，在“Add statements”部分点击“Advanced mode”按下文修改 5 到 7 行，修改后如图 14（不要修改 Version），完成修改后点击“Create”；

```
{
  "Effect": "Allow",
  "Action": "iot:*",
  "Resource": "*"
}
```

### Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

**Name**

---

**Add statements**

Policy statements define the types of actions that can be performed by a resource.

**Basic mode**

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": "iot:*",
7       "Resource": "*"
8     }
9   ]
10 }

```

**Add statement**

**Create**

图 14 准备创建策略设置

#### 1.3.2.4 创建 Things

在控制台左侧导航栏选择“Manage”—>“Things”，如果没有任何“IoT things”则出现图 15 界面，点击“Register a thing”；否则点击“Create”；

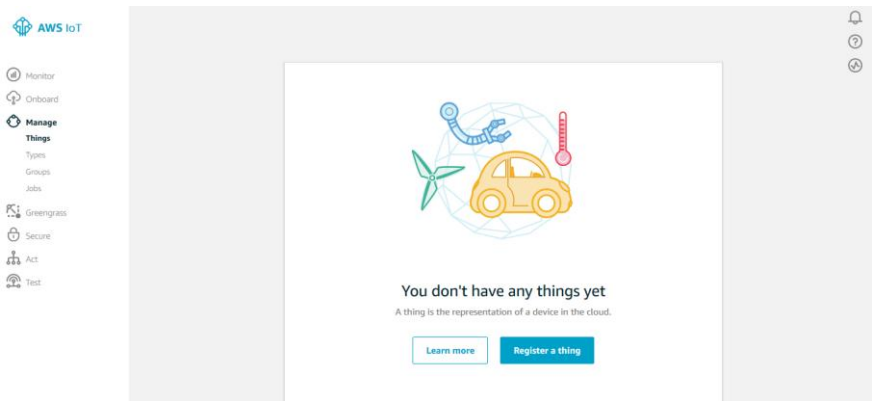


图 15 准备注册 thing

在新页面中点击“Create a single thing”，如图 16；

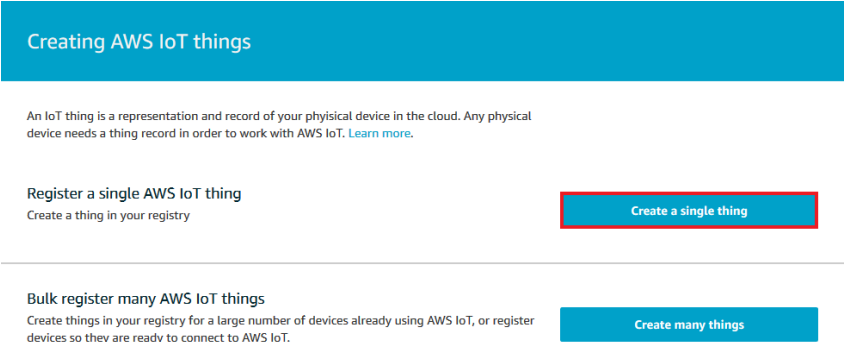


图 16 创建 thing

在新打开的页面中，名称填写自己需要创建的名称如“myShadowThing”（需要与工程中的名称一致），如图 17，点击“Next”；

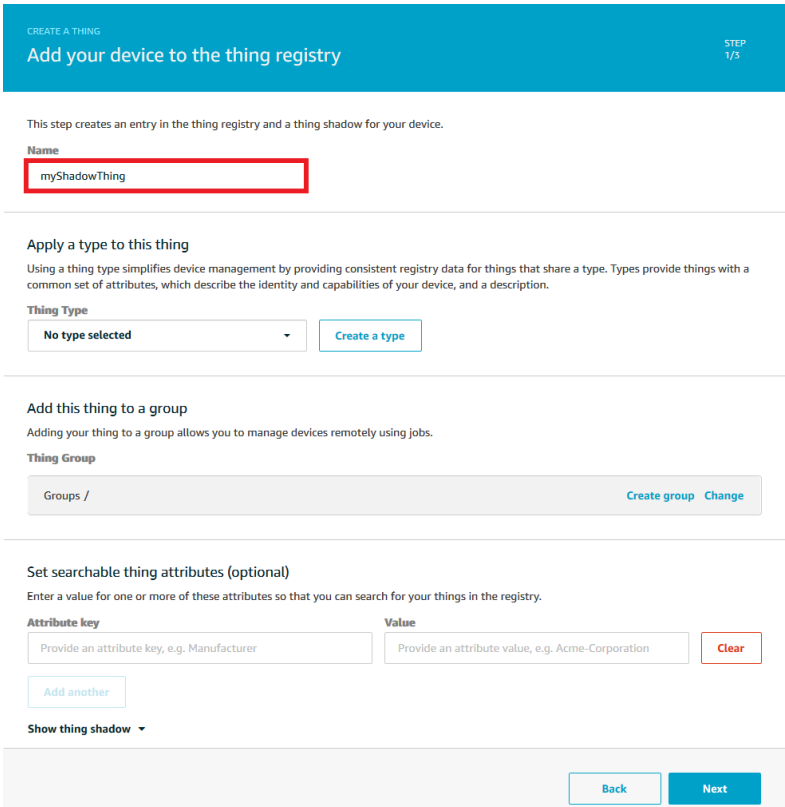


图 17 创建 thing 设置

在如图 18 所示页面中，点击“Create certificate”；

CREATE A THING  
Add a certificate for your thing

STEP 2/3

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 with CSR**  
Upload your own certificate signing request (CSR) based on a private key you own.

**Use my certificate**  
Register your CA certificate and use your own certificates for one or many devices.

**Skip certificate and create thing**  
You will need to add a certificate to your thing later before your device can connect to AWS IoT.

图 18 创建证书

证书创建后，在如图 19 所示的页面年中下载“certificate”、“private key”，然后点击“Activate”；

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	<a href="#">Download</a>
A public key	<a href="#">Download</a>
A private key	<a href="#">Download</a>

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

[Activate](#)

Done [Attach a policy](#)

图 19 下载并激活证书

后续打开的页面如图 20，勾选创建的策略，然后点击“Register Thing”。

CREATE A THING  
Add a policy for your thing

STEP 3/3

Select a policy to attach to this certificate:

Search policies

☒ myIoTPolicy [View](#)

1 policy selected [Register Thing](#)

图 20 证书应用到 Thing

## 2 例程运行

### 2.1 环境准备

在导入工程前，需要安装 SDK；方法为，打开 MCUXpresso IDE，在右下方点击“Installed”，将下载的 SDK 拖放到空白处，按提示完成 SDK 安装，如图 21。

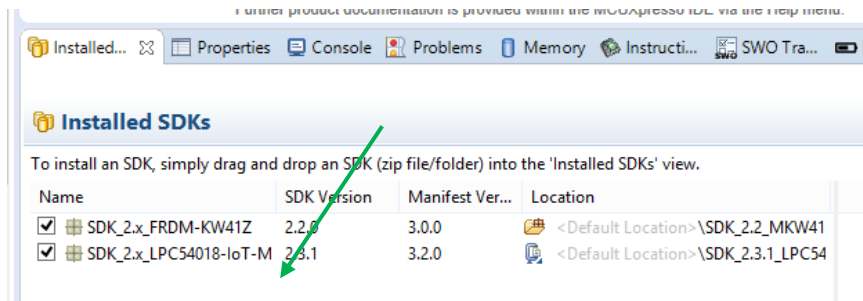


图 21 安装 SDK

### 2.2 导入编译工程

#### 2.2.1 导入工程

打开“MCUXpresso IDE”，点击左下方的“Import SDK Example(s)”，如图 22 所示；

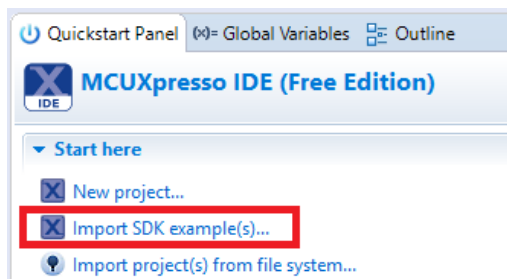


图 22 准备导入工程

在出现的窗口中，选择“lpc54018iotmodule”，如图 23；

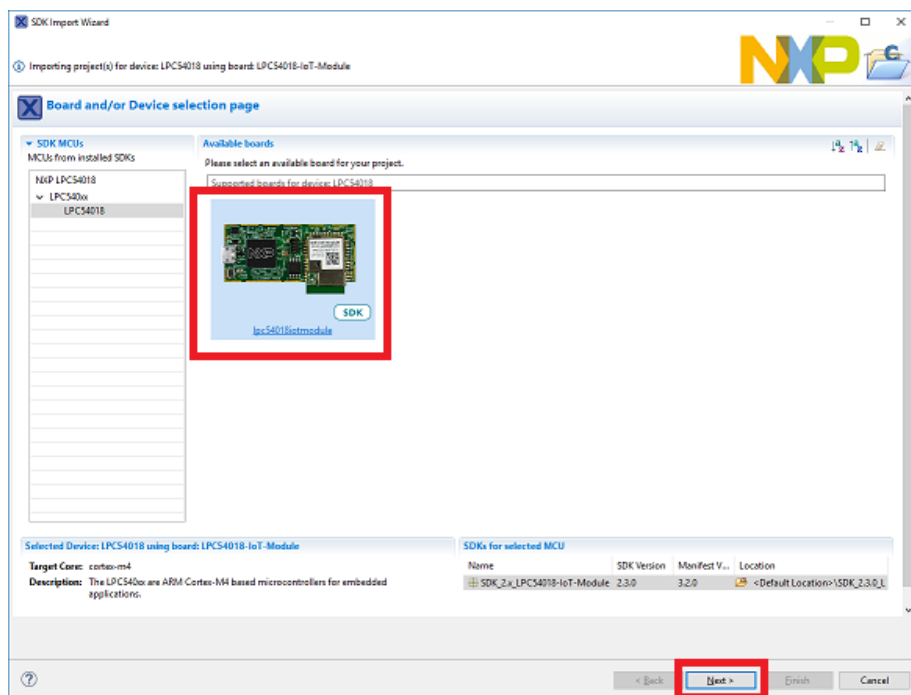


图 23 导入工程选择开发板

在打开的 SDK Import Wizard 窗口中, 搜索栏输入“aws\_shadow”, 选择“aws\_shadow\_wifi\_qspi\_xip”, 其它选项设置参考图 24, 点击“Finish”; 工程导入后如图 25。

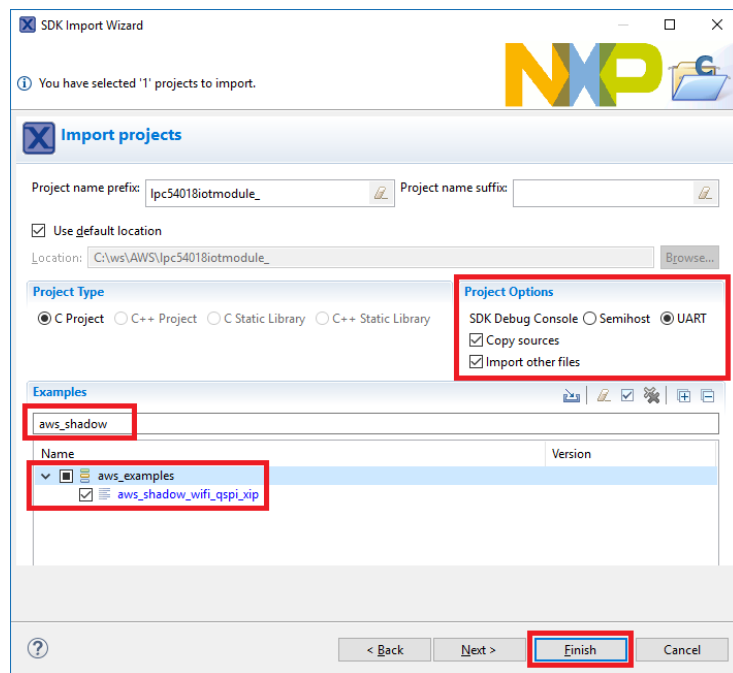


图 24 导入工程设置

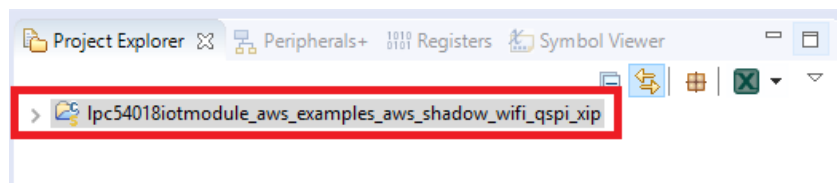


图 25 完成工程导入

## 2.2.2 工程修改

在工程目录下, 按如下路径“/amazon-freertos/include/”, 找到“aws\_clientcredential.h”文件, 修改如下参数:

- 1) Broker endpoint, 该参数值可通过“AWS IoT -> myShadowThing(创建 Things 的名称) -> Interact “查看 (建议直接复制);
- 2) Thing name, 该参数为创建 Things 的名称 (文本为 myShadowThing);
- 3) Wi-Fi SSID: 根据实际情况填写 (若公司 wifi 认证较为复杂时, 建议使用手机热点);
- 4) Wi-Fi Password: 根据实际情况填写。

操作结果如图 26 所示;

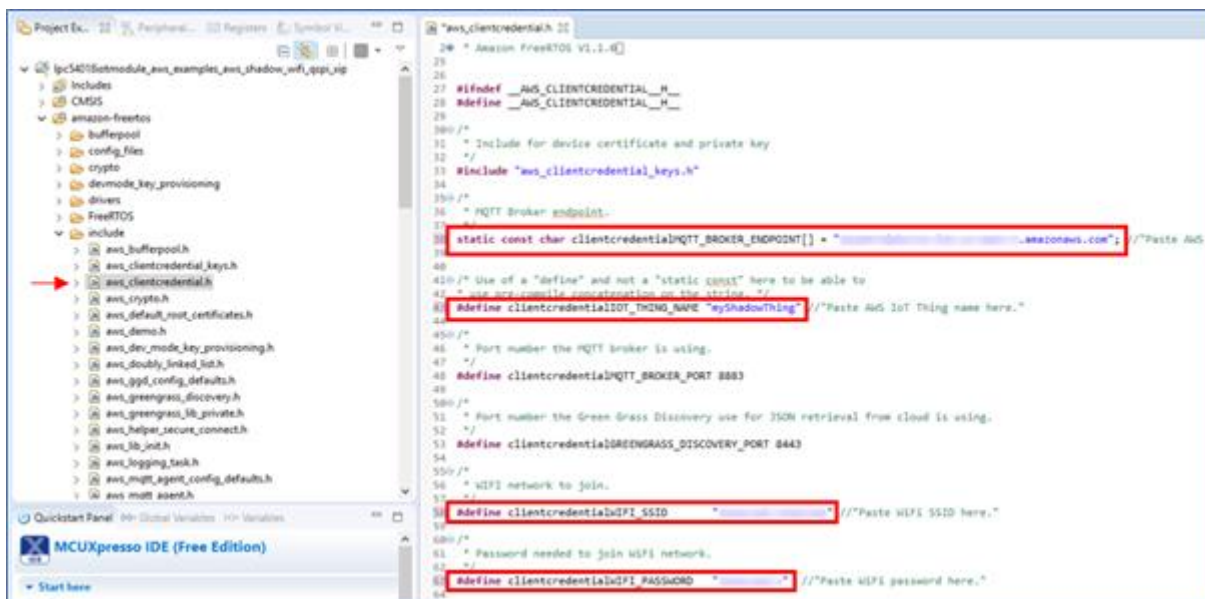


图 26 修改头文件

按路径“SDK\_2.3.0\_LPC54018-IoT-Module/rtos/amazon-freertos/demos/common/devmode\_key\_provisioning/CertificateConfigurationTool”找到“/CertificateConfigurator.html”文件；  
若 SDK 未解压，建议将“CertificateConfigurationTool”解压；  
可通过 IDE 软件右下方“Installed SDKs”打开 SDK，如图 27；

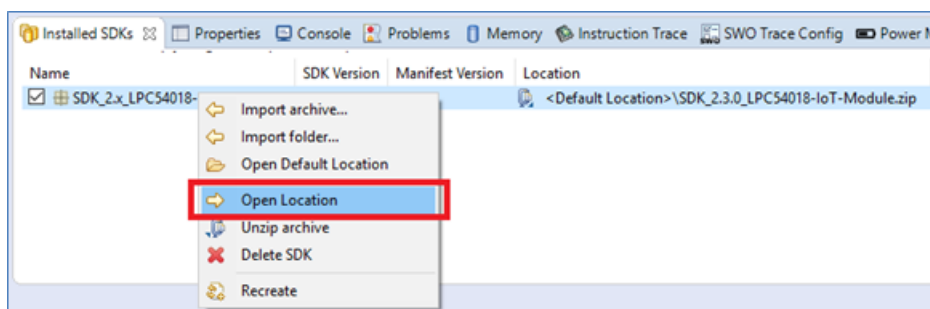


图 27 认证文件生成工具路径

打开“Certificate Configurator”文件，如图 28（建议使用 Google Chrome 浏览器）；

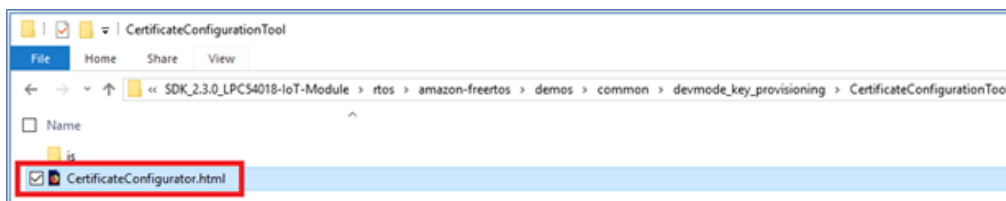


图 28 认证文件生成工具

在浏览器界面中按要求分别选择“创建 Things”时下载的证书及私钥文件，点击“Generate and save aws\_clientcredential\_keys.h”，如图 29；

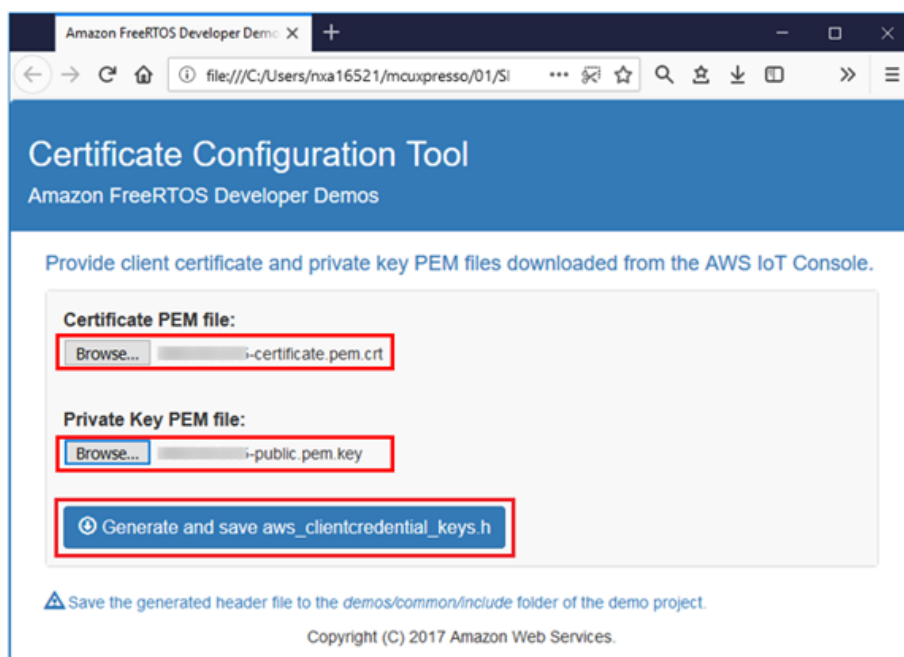


图 29 生成认证文件

此时浏览器会下载一个“aws\_clientcredential\_keys.h”文件；用该文件替换“lpc54018iotmodule\_aws\_examples\_aws\_shadow\_wifi\_qspi\_xip\amazon-freertos\include”下的同名文件，如图 30；

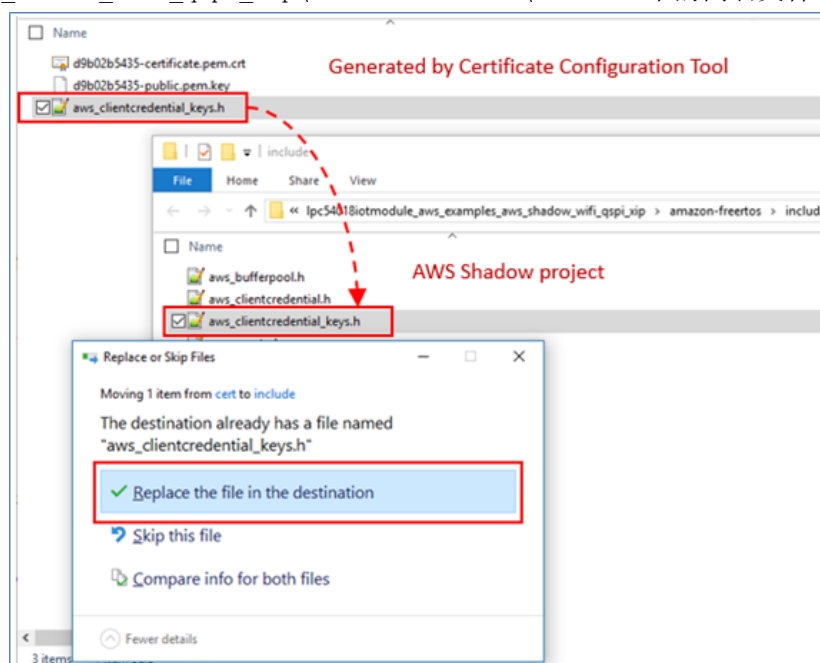


图 30 替换工程认证文件

### 2.2.3 编译工程

点击 IDE 左下方“Build”，开始编译工程如图 31；编译过程中如图 32，等待编译完成；

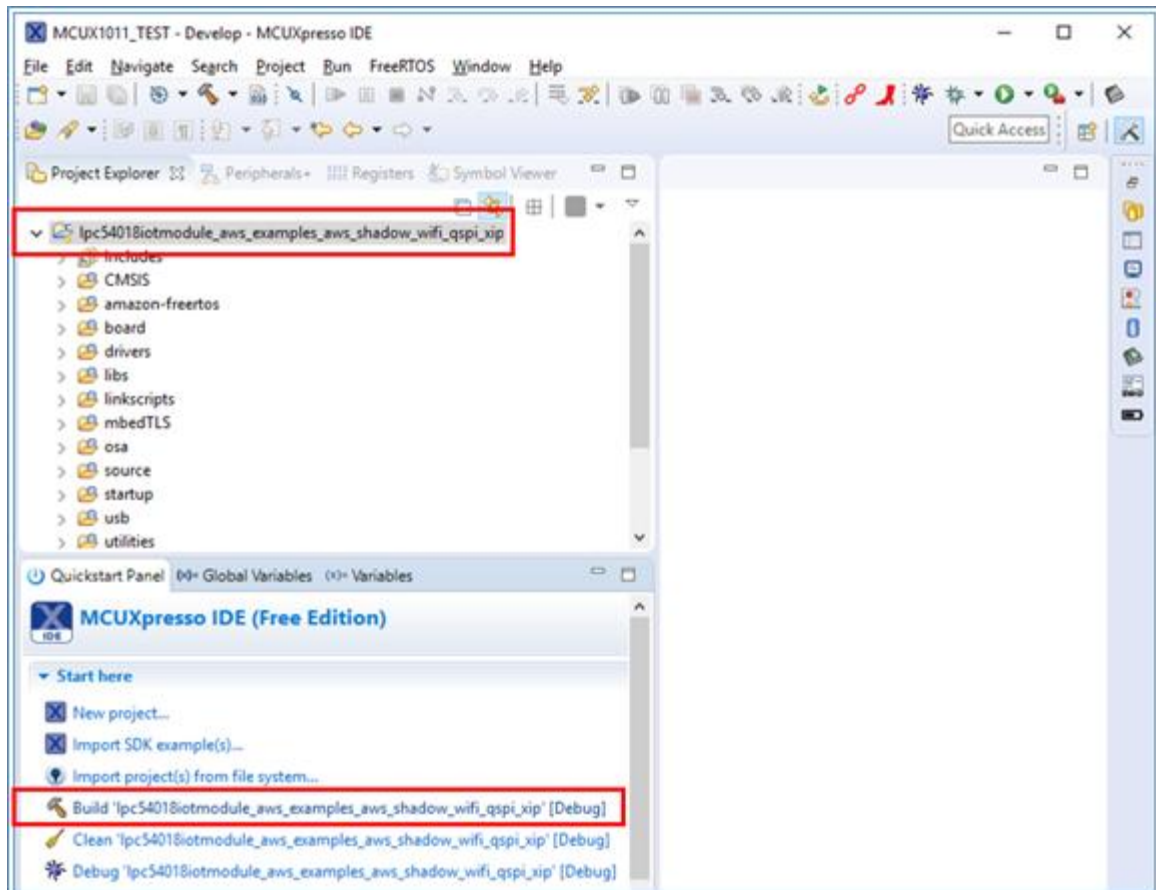


图 31 编译工程

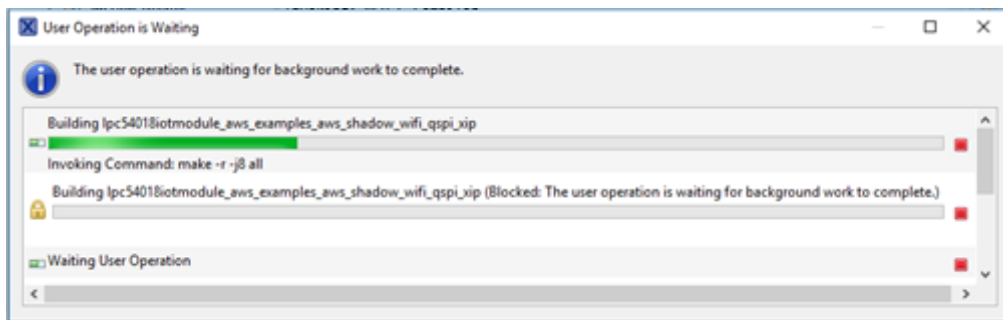


图 32 工程编译

工程编译完成后，在 IDE 右下方的“Console”中确认编译通过（没有 error），如图 33；

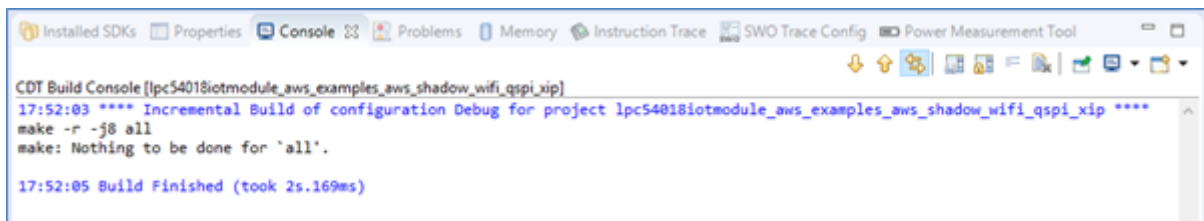


图 33 工程编译完成

## 2.3 调试

如图 34 所示连接 LPC-Link2 调试器及 LPC54018-IoT Module（注意连接方向）；当然也可使用 J-Link 调试器（需要转接），连接方法如图 35；



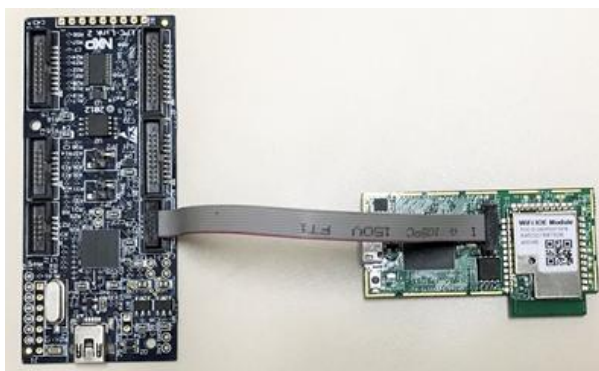


图 34 连接调试器与开发板



图 35 连接调试器与开发板 (Jlink)

通过 micro-USB 电缆连接 LPC54018 IoT Module 到 PC 机，如图 36；



图 36 开发板连接到电脑

点击 IDE 左下方“Debug lpc54018iotmodule\_aws\_examples\_aws\_shadow\_wifi\_qspi\_xip”，如图 37，开始调试工程；

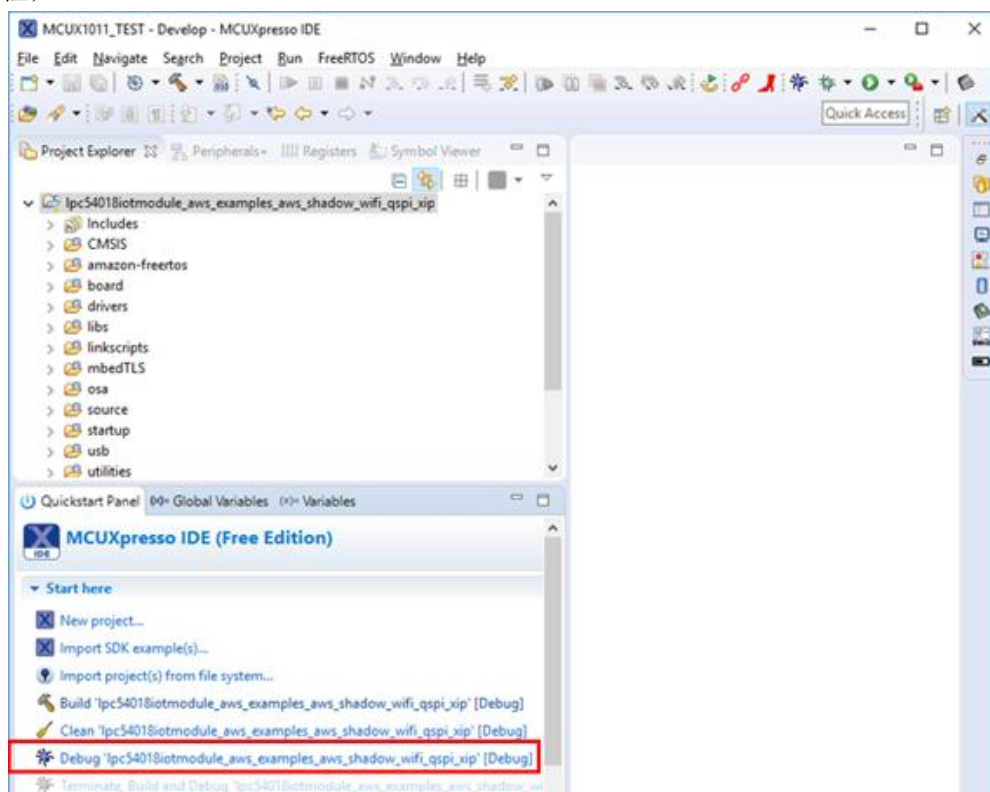


图 37 调试工程

在调试器选择窗口中选择调试器 (LPC-Link2 or J-Link)，点击“OK”后，开始将固件写入硬件；等待 p

rogram 完成后, 点击“Run”, 如图 38;

注: 若调试器没有停在 main() 函数, 点击“Restart”, 如图 39, 然后再一次点击“Run”;

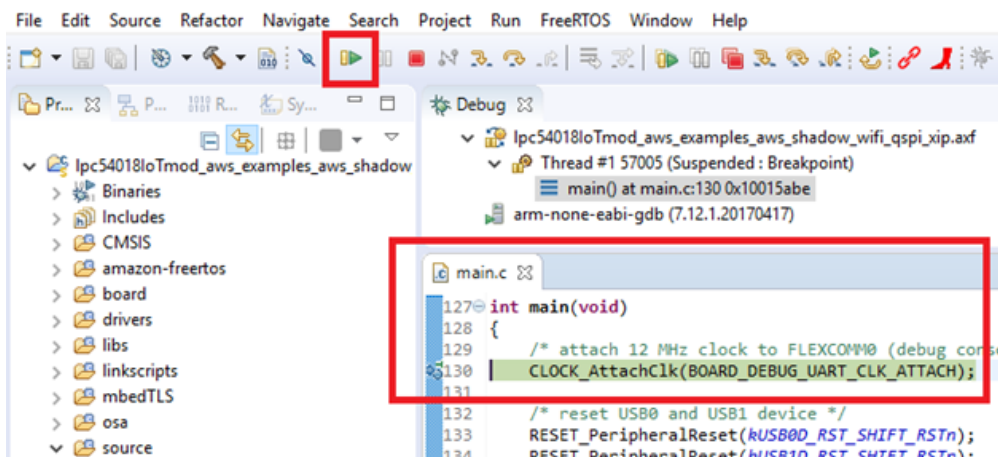


图 38 进入 main(函数)



图 39 Restart 工程

打开电脑的“Device Manager”, 会出现 IoT Module 虚拟的串口如图 40, 若串口驱动未正常安装如图 41, 可使用“\$SDK\_PATH\boards\lpc54018iotmodule\usb\_examples\usb\_device\_cdc\_vcom\inf”安装虚拟串口驱动;

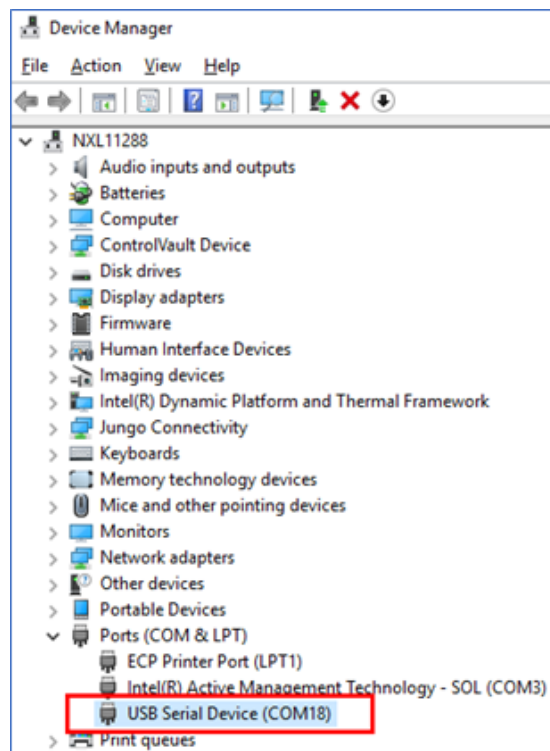


图 40 开发板对应的串口

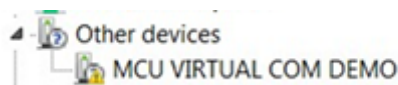


图 41 开发板串口驱动异常

使用串口终端软件 (推荐 TeraTerm、PuTTY) 打开开发板 (lpc54018iotmodule) 的对应的串口, 开发板串口输出的信息可参阅图 42;

```

COM18 - Tera Term VT
File Edit Setup Control Window Help
0 0 [Inr Svc] Starting key provisioning...
1 0 [Inr Svc] Write root certificate...
2 1 [Inr Svc] Write device private key...
3 12 [Inr Svc] Write device certificate...
4 21 [Inr Svc] Key provisioning done...
5 22 [Inr Svc] Starting WiFi...
6 1662 [Inr Svc] WiFi module initialized.
7 7812 [Inr Svc] WiFi connected to AP JC.
8 8905 [Inr Svc] IP Address acquired 192.168.43.85
9 8915 [MainDemoTask] [Shadow 0] MQTT: Creation of dedicated MQTT client succeeded.
10 8917 [MainDemoTask] Sending command to MQTT task.
11 8920 [MQTT] Received message 10000 from queue.
12 9064 [MQTT] Looked up aize9r3jykcrl2.iot.us-east-2.amazonaws.com as 18.221.47.28
13 31310 [MQTT] MQTT Connect was accepted. Connection established.
14 31311 [MQTT] Notifying task.
15 31315 [MainDemoTask] Command sent to MQTT task passed.
16 31317 [MainDemoTask] [Shadow 0] MQTT: Connect succeeded.
17 31320 [MainDemoTask] Sending command to MQTT task.
18 31322 [MQTT] Received message 20000 from queue.
19 31557 [MQTT] MQTT Subscribe was accepted. Subscribed.
20 31558 [MQTT] Notifying task.
21 31562 [MainDemoTask] Command sent to MQTT task passed.
22 31564 [MainDemoTask] [Shadow 0] MQTT: Subscribe to callback topic succeeded.
23 31567 [MainDemoTask] Sending command to MQTT task.
24 31569 [MQTT] Received message 30000 from queue.
25 31804 [MQTT] MQTT Subscribe was accepted. Subscribed.
26 31805 [MQTT] Notifying task.
27 31809 [MainDemoTask] Command sent to MQTT task passed.
28 31811 [MainDemoTask] [Shadow 0] MQTT: Subscribe to callback topic succeeded.
29 31814 [MainDemoTask] Sending command to MQTT task.
30 31816 [MQTT] Received message 40000 from queue.
31 32051 [MQTT] MQTT Subscribe was accepted. Subscribed.
32 32052 [MQTT] Notifying task.
33 32056 [MainDemoTask] Command sent to MQTT task passed.
34 32058 [MainDemoTask] [Shadow 0] MQTT: Subscribe to callback topic succeeded.
35 32061 [MainDemoTask] Sending command to MQTT task.
36 32063 [MQTT] Received message 50000 from queue.
37 32305 [MQTT] MQTT Subscribe was accepted. Subscribed.
38 32306 [MQTT] Notifying task.
39 32310 [MainDemoTask] Command sent to MQTT task passed.
40 32312 [MainDemoTask] [Shadow 0] MQTT: Subscribe to accepted topic succeeded.
41 32315 [MainDemoTask] Sending command to MQTT task.
42 32317 [MQTT] Received message 60000 from queue.
43 32552 [MQTT] MQTT Subscribe was accepted. Subscribed.
44 32553 [MQTT] Notifying task.
45 32557 [MainDemoTask] Command sent to MQTT task passed.
46 32559 [MainDemoTask] [Shadow 0] MQTT: Subscribe to rejected topic succeeded.
47 32562 [MainDemoTask] Sending command to MQTT task.
48 32564 [MQTT] Received message 70000 from queue.
49 32587 [MQTT] Notifying task.
50 32591 [MainDemoTask] Command sent to MQTT task passed.
51 32593 [MainDemoTask] [Shadow 0] MQTT: Publish to operation topic succeeded.
52 32928 [Shd-IOT-0] Sending command to MQTT task.
53 32929 [MQTT] Received message 80000 from queue.
54 33163 [MQTT] MQTT Subscribe was accepted. Subscribed.
55 33164 [MQTT] Notifying task.
56 33166 [Shd-IOT-0] Command sent to MQTT task passed.
57 33168 [Shd-IOT-0] [Shadow 0] MQTT: Subscribe to accepted topic succeeded.
58 33172 [Shd-IOT-0] Sending command to MQTT task.
59 33173 [MQTT] Received message 90000 from queue.
60 33407 [MQTT] MQTT Subscribe was accepted. Subscribed.
61 33408 [MQTT] Notifying task.
62 33410 [Shd-IOT-0] Command sent to MQTT task passed.
63 33412 [Shd-IOT-0] [Shadow 0] MQTT: Subscribe to rejected topic succeeded.
64 33416 [Shd-IOT-0] Sending command to MQTT task.
65 33417 [MQTT] Received message a0000 from queue.
66 33446 [MQTT] Notifying task.
67 33448 [Shd-IOT-0] Command sent to MQTT task passed.
68 33450 [Shd-IOT-0] [Shadow 0] MQTT: Publish to operation topic succeeded.
69 33852 [Shd-IOT-0] Shadow Demo initialized.
70 33854 [Shd-IOT-0] Client request to change color of light bulb.
71 33856 [Shd-IOT-0] Sending command to MQTT task.
72 33858 [MQTT] Received message b0000 from queue.
73 33885 [MQTT] Notifying task.
74 33887 [Shd-IOT-0] Command sent to MQTT task passed.
75 33889 [Shd-IOT-0] [Shadow 0] MQTT: Publish to operation topic succeeded.
76 34248 [Shd-IOT-0] Client change done in thing shadow.
77 34250 [ShDenoUpdt] Performing thing update.
78 34253 [ShDenoUpdt] Sending command to MQTT task.
79 34255 [MQTT] Received message c0000 from queue.
80 34282 [MQTT] Notifying task.
81 34286 [ShDenoUpdt] Command sent to MQTT task passed.
82 34288 [ShDenoUpdt] [Shadow 0] MQTT: Publish to operation topic succeeded.

```

图 42 开发板串口输出信息

注: The terminal will only run when the software is running. Stop, reset or halt will disable USB CDC.

在 AWS 云端, 打开创建的“Things”页面在 shadow 的 shadowstate 中可查看信息变化, 如图 43。

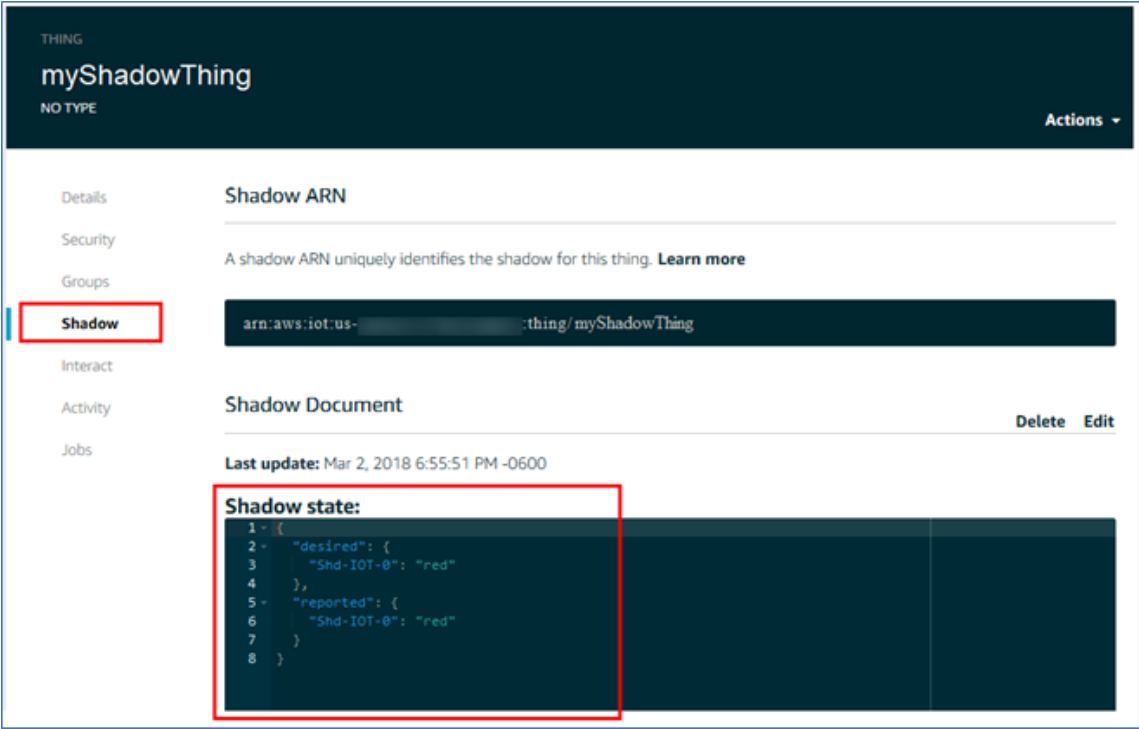


图 43 云端信息

2.4 其它说明

如何找到 Shadow，登陆 AWS 控制台后，点击“Services”，在“History”中点选“IoT Core”如图 44，在新页面中点击后展开“Manage”，点击“Things”后选择自己建立的 Things（如 LPC54018Iot）如图 45，而后会出现图 43 界面。

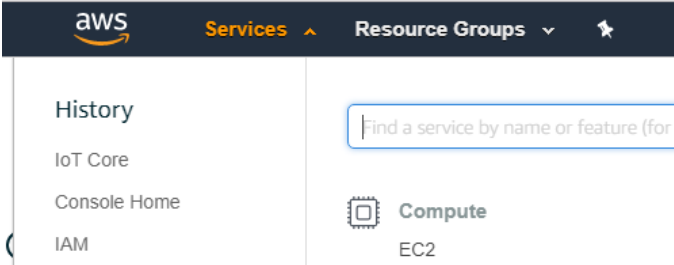


图 44 IoT Core

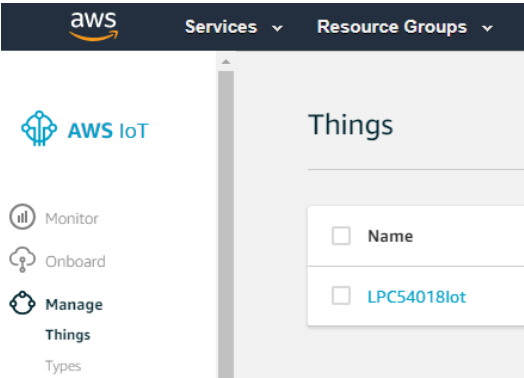


图 45 Things