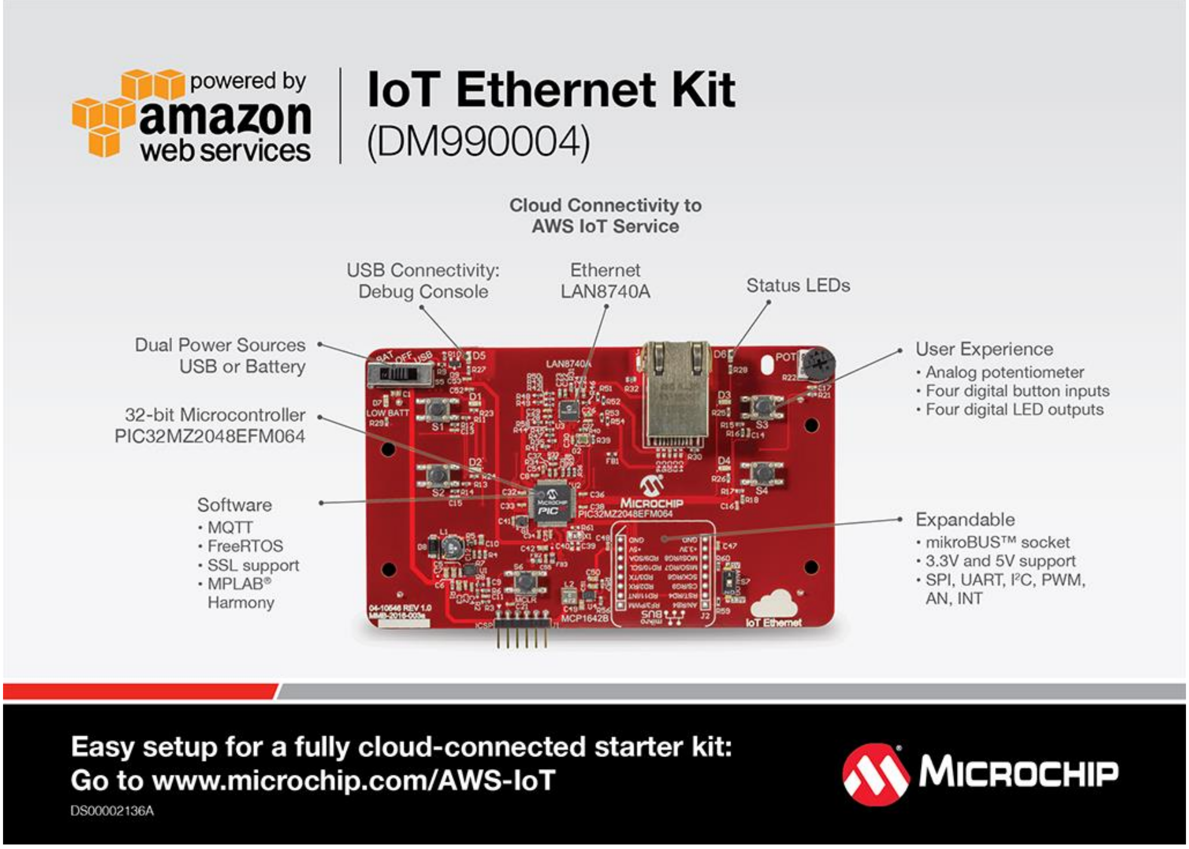
Introduction

AWS IoT is a service that will allow internet of things (IoT) devices to be easily and securely connected to Amazon Web Services (AWS). The IoT Ethernet Kit has been design to work with this service and allow you to develop new IoT based designs.



powered by amazon web services

IoT Ethernet Kit
(DM990004)

Cloud Connectivity to AWS IoT Service

USB Connectivity: Debug Console

Ethernet LAN8740A

Status LEDs

Dual Power Sources USB or Battery

32-bit Microcontroller PIC32MZ2048EFM064

Software
- MQTT
- FreeRTOS
- SSL support
- MPLAB® Harmony

User Experience
- Analog potentiometer
- Four digital button inputs
- Four digital LED outputs

Expandable
- mikroBUS™ socket
- 3.3V and 5V support
- SPI, UART, I²C, PWM, AN, INT

Easy setup for a fully cloud-connected starter kit:
Go to www.microchip.com/AWS-IoT
DS00002136A

MICROCHIP

Required Tools and Applications

Microchip Tools and Applications

You will need the following Microchip development tools to run out of box demo

- IoT Ethernet Kit (DM990004), available from Microchip Direct
- Download and install Insight on Things desktop application; Download latest version
- To learn more about building the project from source or just upgrading the firmware, read IoT Ethernet Firmware Compiling and Programing in our documents folder

AWS Tools and Applications

- You will need an AWS account to use AWS IoT.

    - For more information how to setup an AWS Account please go to AWS main page
    - Please check with your IT or IS department on your company's policies on using cloud computing, or to see if you have a corporate AWS account that should be used.

- To use the AWS IoT services you will need to make use of *AWS Command Line Interface* (awscli) tool. For more information on how to install, configure, and use the awscli tool please go to AWS Command Line Interface User Guide.

NOTE: If you are a **Microchip Employee** we have a corporate AWS account that must be used. To setup your AWS IoT device please go to https://setup.iot.microchip.com for more information

Setting Up the AWS Command Line Interface

Linux, Unix, and Windows Installations

Follow the instructions in the Getting Set Up with the AWS Command Line Interface for installations for your operating system.

Mac OS X Installation

This assumes that you have not installed previous version of Python, awscli, or openssl; if you have you will need to uninstall those versions. You will need to use a version of openssl 1.0.0 or later to connect to AWS IoT services as TLS 1.1 is required. To do this follow the these steps:

1. Use Homebrew to install the latest version of openssl with the following commands in a terminal window. You will need to force the links so that Python will use the right version of openssl.

```
2. brew update
3. brew install openssl
4. brew link --force openssl
```

5. Reinstall Python to use the latest openssl:

```
6. brew install python --with-brewed-openssl
```

7. To verify that python was updated run the following command and make sure the version os greater than 1.0.2d

```
8. python -c 'import ssl; print ssl.OPENSSL_VERSION'
```

9. Install/upgrade the AWS CLI:

```
10.pip install --upgrade awscli
```

11. To verify the version of the AWS CLI tool installed run the following command and look for version 1.10.x or greater:

12. `aws --version`

Setting Up the AWS IoT Service

To setup your AWS IoT service to work with the *IoT Ethernet Kit* you will need to work with the awscli that you installed in the previous section.

1. Using the awscli you will need to find the endpoint that will be used for your AWS IoT account. To do this run the following command in your terminal/command window.

2. `aws iot describe-endpoint`

   You will receive the following back that you will need to add to the firmware in the next section. The output will look like:`<random-string>.iot.us-east-1.amazonaws.com`

3. Create certificates that will be used in the application firmware by running the following command in your terminal/command prompt window:

4. `aws iot create-keys-and-certificate --set-as-active --output text`

   You will need to copy the output on the terminal/command window to text files for the certs and private key. Please also note the *certificate arn* value that will be used later in this section; it is the first long string of text that you will seen on the screen.

   NOTE: this is the only time that you will be able to see this information and make sure there is no text or spaces at the beginning of each of the files you create. For example in the output below you will need to remove the `KEYPAIR`text and spaces so that the first line of the private key starts with the dashes:

```
...
Gmh/2ib2NsA8Tf+8drTQXf3gQ4oemdOfnA4IlXiFGA9QAP9i/krzYvaVAlSogLWh
wRHaP1hzPuS9+0UjZkS1Uzyfqmd+vR4Om8S+t2VmEwYMJMG3mUaFXdhBhMDJ
-----END CERTIFICATE-----

KEYPAIR  -----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAynvp1r7jl9HzbAVCgVWzYmKu4O6nNM4wBZQ1YjHpSOL1+joF
KKcXM8eS8jmUbHRJ84JeDrr9A50ok1cyUCDBPkf6c7VO4xWGzXruR3WoVjRYq3vc
...
```

5. Create and attach the following policy that will allow your thing to publish and subscribe to the AWS IoT service

   - First create a file that will have the policy information in it and save it to you computer:

NOTE: Make sure you replace `<AWS-IoT Region>` and `<AWS Account Number>` with the AWS-IoT region and your AWS account number

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
        "iot:Connect"
    ],
    "Resource": [
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:client/${iot:ClientId}"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:Publish"
    ],
    "Resource": [
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/update",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/get"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:Subscribe"
    ],
    "Resource": [
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topicfilter/$aws/things/${iot:ClientId}/shadow/update/delta"
,
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topicfilter/$aws/things/${iot:ClientId}/shadow/update/accept
ed",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topicfilter/$aws/things/${iot:ClientId}/shadow/update/reject
ed",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topicfilter/$aws/things/${iot:ClientId}/shadow/get/accepted"
,
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topicfilter/$aws/things/${iot:ClientId}/shadow/get/rejected"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "iot:Receive"
    ],
    "Resource": [
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/update/delta",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/update/accepted",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/update/rejected",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/get/accepted",
      "arn:aws:iot:<AWS-IoT Region>:<AWS Account
Number>:topic/$aws/things/${iot:ClientId}/shadow/get/rejected"
    ]
  }]
}
```

- Load the policy to the AWS IoT services with the following command:

```
aws iot create-policy --policy-name 'MQTTandShadowPolicy' --policy-document file://<path-to-your-policy-document>
```

- You will need to attached this policy to a principal to allow the cert the rights to publish and subscribe to topics:

```
aws iot attach-principal-policy --principal '<certificate-arn>' --policy-name 'MQTTandShadowPolicy'
```

6. Create a thing shadow based on the MAC address of the board:

```
7. aws iot create-thing --thing-name '<MAC Address of the board>'
```

Running the Demo

There are two parts to running the demo. First you have to commission the demo so it knows how to talk to your AWS Account that you setup in the Setting Up the AWS IoT Service section of this out of box guide. Second is sending and receiving data from AWS IoT.

Commissioning

To setup and run the demo follow these instructions:

1. Plug the IoT Ethernet Kit into a router or switch using an ethernet cable
2. Power the IoT Ethernet Kit by connecting the supplied Micro-USB cable to a power source and setting the power switch on the board to USB
3. Connect your computer to the same network that the IoT Ethernet Kit is connected
   - Your board will register a Multicast DNS (mDNS) name with the local network that will allow you to access a configuration webpage.
4. Use a bonjour or multicast DNS service to find your board's IP address and enter the IP address into a web browser. The kit will have a service name such as `xxxxxx_IoT-E` where `xxxxxx` is the last six characters of the kit's MAC address.
   - Alternatively you can try entering `xxxxxx_iot-e.local` into a web browser to navigate to the webpage where`xxxxxx` is the last six characters of the MAC address on the board

5. Once the web browser finds and displays the IoT Ethernet Kit configuration webpage
    - Enter your AWS IoT Endpoint Address that you received from the previous AWS IoT Service Setup section. The endpoint will have the form `<random-string>.iot.us-east-1.amazonaws.com`
6. Copy and paste your AWS Certificate into the box labeled AWS Certificate.

    - The cert should look like

    ```
    -----BEGIN CERTIFICATE-----
    <cert data>
    -----END CERTIFICATE-----
    ```

7. Copy and paste your AWS Certificate Private Key into the box labeled AWS Certificate Private Key.

    - The key should look like

    ```
    -----BEGIN RSA PRIVATE KEY----
    <key data>
    -----END RSA PRIVATE KEY-----
    ```

8. Click the **Join** button and a *Connecting to server* page will appear showing the endpoint that the IoT Ethernet board will connect to along with the AWS Thing Name (MAC Address). > NOTE: The information you enter above will be sent in the clear and is not secured as if you where using https; this is an option that you can add to your code later
9. You will see a LEDs D1-D6 light up while the board gets initial data from the server.
    - If not, please see the Status and Error Code Table table below for more information
10. The board is now connect and running; there are two blue LED status indicator on the board.
    - LED D6 will flash briefly when transmitting data
    - LED D5 will flash briefly when receiving data

Sending and Receiving Data with AWS IoT

This demo is controlled though the AWS IoT shadow registors by the Insight on Things desktop application that we provide you. Click here download the latest version. You will need to follow the README.md guide on the github page to setup this application.

Troubleshooting

If you are having trouble connecting with the starter kit, check to make sure that each of the issues bellow are resolved.

Connection Issue

- Check that you have a valid internet connection on your network.
- Ensure that port 8883 is open to the internet.
- Ensure that your AWS IoT service is setup property.

Change the configuration

- Changing the configuration that you entered, or if you entered it incorrectly, simply power off the starter kit, press and hold S2 and S3 while turning on the power to the starter kit.
- This will erase the current configuration and you can begin this process again and re-enter the configuration information.

Association to AP Failed

- Check that your access point is not blocking MAC addresses.

DNS Resolved Failed

- Check that the AWS IoT Endpoint Address you configured your starter kit with is correct.
- Check that you have a valid internet connection on your network

Status and Error Code Table

D7	D6	D5	D4	D3	D2	D1	Diagnostic Report
-	0	F	0	0	0	0	Configuration Mode: Need to configure Connected to network
-	1	1	1	1	1	1	Connected to network and waiting on data
-	1	0	C	C	C	C	Associating to network
-	1	1	0	0	0	1	Connection Issue, no internet found
-	1	1	0	0	1	0	Association to network failed
-	1	1	0	1	0	1	DNS Unresolved: Bad endpoint or no connection

D7	D6	D5	D4	D3	D2	D1	Diagnostic Report
-	F	0	-	-	-	-	Normal Operation: D6 flashes each message transmission
-	0	F	-	-	-	-	Normal Operation: D5 flashes each subscription receive
1	-	-	-	-	-	-	Battery Voltage Low

F = Flashing, C = Rotating in a counter clockwise pattern, - = Don’t care, 0 = Off, and 1 = On