

Getting Started with the Nuvoton NuMaker-IoT-M487

Before you begin, see [Prerequisites](#).

If you do not already have the Nuvoton NuMaker-IoT-M487 board, you can purchase one from [Nuvoton Direct](#).

Setting Up Your Environment

The Keil MDK v5 Essential, Plus, or Pro version should work for the Nuvoton M487 (Cortex-M4 core) MCU. One more special version with price discount for Nuvoton Cortex-M4 series MCU, you can download Keil MDK Nuvoton edition ([here](#)).

Install Development Tool for NuMaker-IoT-M487

1. Download the Keil MDK Nuvoton Edition on [Keil MDK website](#) or standard edition on [Keil MDK-Arm website](#).
2. Install Keil MDK with your license.

If you experience issues during installation, please contact us ([here](#)).

Download and Configure Amazon FreeRTOS

After your environment is set up, you can download Amazon FreeRTOS.

Download Amazon FreeRTOS

1. In the AWS IoT console, browse to the [Amazon FreeRTOS page](#).
 2. In the navigation pane, choose **Software**.
 3. Under **Amazon FreeRTOS Device Software**, choose **Configure download**.
 4. Choose **Download FreeRTOS Software**.
-

5. Under **Software Configurations**, find **Connect to AWS IoT-Nuvoton**, and then choose **Download**.
6. Unzip the downloaded file to a folder, and make a note of the folder path. In this tutorial, this folder is referred to as `BASE_FOLDER`.

Note

The maximum length of a file path on Microsoft Windows is 260 characters. The longest path in the Amazon FreeRTOS download is 122 characters. To accommodate the files in the Amazon FreeRTOS projects, make sure that the path to the AmazonFreeRTOS directory is fewer than 98 characters long. For example, `C:\Users\Username\Dev\AmazonFreeRTOS` works, but `C:\Users\Username\Documents\Development\Projects\AmazonFreeRTOS` causes build failures.

Configure Your Project

To run the demo, you must configure your project to work with AWS IoT and a Wi-Fi network.

Configure your AWS IoT endpoint

1. Browse to the [AWS IoT console](#).
2. In the navigation pane, choose **Settings**.
3. Copy your AWS IoT endpoint from the **Endpoint** text box. It should look like `<1234567890123>.iot.<us-east-1>.amazonaws.com`.
4. Open `demos\common\include\aws_clientcredential.h` in your IDE.
5. Set `clientcredentialMQTT_BROKER_ENDPOINT` to your AWS IoT endpoint.

Configure your Wi-Fi

1. Open the same `aws_clientcredential.h` file.
2. Specify values for the following #define constants:

`clientcredentialMQTT_BROKER_ENDPOINT`

Your AWS IoT endpoint

`clientcredentialIOT_THING_NAME`

The AWS IoT thing for your board

clientcredentialWIFI_SSID

The SSID for your Wi-Fi network

clientcredentialWIFI_PASSWORD

The password for your Wi-Fi network

clientcredentialWIFI_SECURITY

The security type of your Wi-Fi network.

Valid security types are:

- eWiFiSecurityOpen (Open, no security)
- eWiFiSecurityWEP (WEP security)
- eWiFiSecurityWPA (WPA security)
- eWiFiSecurityWPA2 (WPA2 security)

Configure your AWS IoT credentials

Amazon FreeRTOS is a C language project, and the certificate and private key must be specially formatted to be added to the project. You need to format the certificate and private key for your device.

1. In a browser window, open
tools\certificate_configuration\CertificateConfigurator.html.
2. Under **Certificate PEM file**, choose the <ID>-certificate.pem.crt that you downloaded from the AWS IoT console.
3. Under **Private Key PEM file**, choose the <ID>-private.pem.key that you downloaded from the AWS IoT console.
4. Choose **Generate and save aws_clientcredential_keys.h**, and then save the file in demos\common\include. This overwrites the existing file in the directory.

Note

The certificate and private key should be hard-coded for demonstration purposes only. Production-level applications should store these files in a secure location.

Build and Run Amazon FreeRTOS Samples

Open the Amazon FreeRTOS Sample Code into Keil uVision

1. Open IDE Keil uVision and choose “File” from the top menu, then click “Open”
2. If want to open the ethernet demo project, please select target project “aws_demos.uvproj” from
<BASE_FOLDER>\AmazonFreeRTOS\demos\nuvoton\numaker-iot-m487\numaker_iot_m487_eth\keil.
3. If want to open the Wi-Fi demo project, please select target project “aws_demos_wifi.uvproj” from
<BASE_FOLDER>\AmazonFreeRTOS\demos\nuvoton\numaker-iot-m487\numaker_iot_m487_wifi\keil.
4. From the **Project** menu, choose **Build Target**.

Run the Amazon FreeRTOS Samples

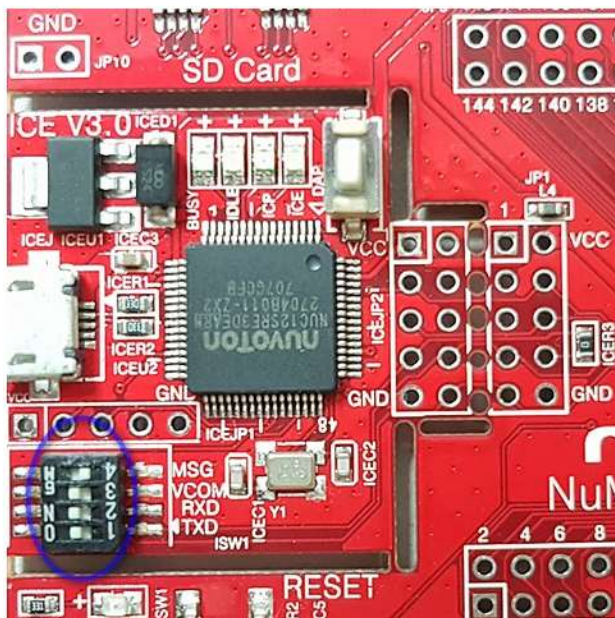
1. Use a USB cable to connect your Numaker-IoT-M487 board to your computer.
2. Rebuild your project.
3. Sign in to the [AWS IoT console](#).
4. In the navigation pane, choose **Test** to open the MQTT client.
5. In **Subscription topic**, enter freertos/demos/echo, and then choose **Subscribe to topic**.
6. From the **Flash** menu, choose **Download**.
7. From the **Debug** menu, choose **Start/Stop Debug Session**.
8. When the debugger stops at the breakpoint in main(), from the **Run** menu, choose **Run (F5)**.

You should see MQTT messages sent by your device in the MQTT client in the AWS IoT console.

Troubleshooting

VCOM driver & switch for Nu-Link

1. If your windows can't recognize device VCOM, please install NuMaker windows serial port driver from <https://goo.gl/4VGca6>.
2. For Keil IDE connection with device through Nu-Link, please ensure MSG switch (No.4 of ISW1 on ICE) as OFF:



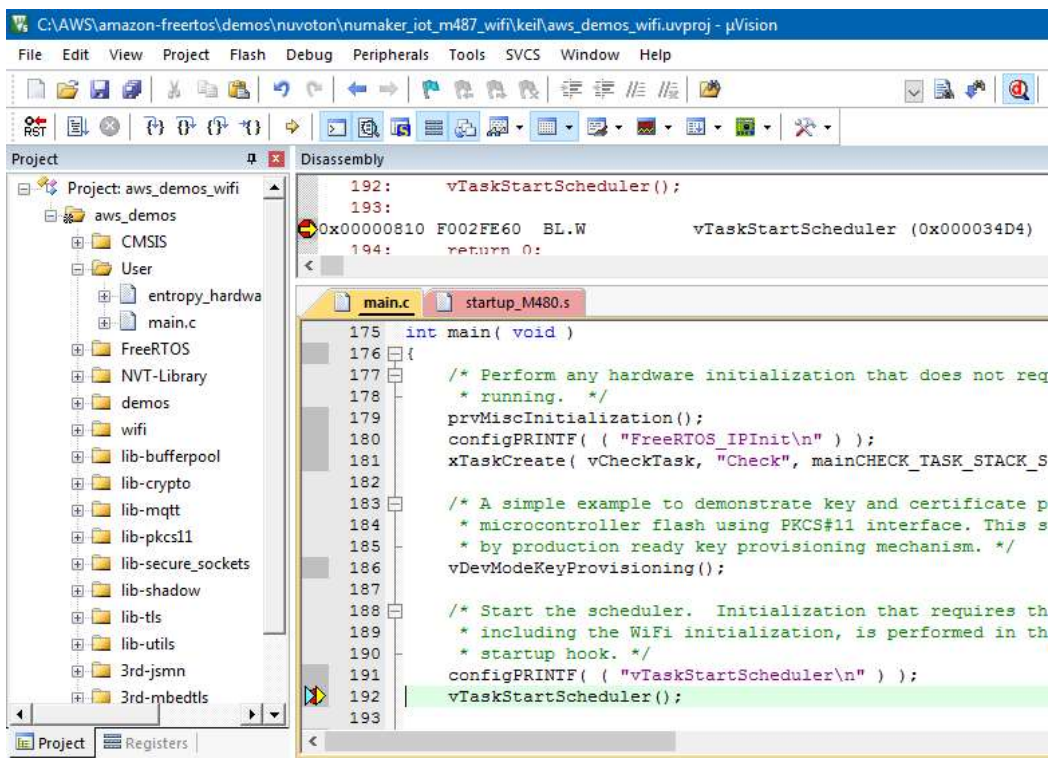
Debugging Amazon FreeRTOS Projects in Keil μ Vision

To start debugging the project

1. Open IDE Keil uVision and choose "File" from the top menu, then click "Open"
2. If want to open the ethernet demo project, please select target project "aws_demos.uvproj" from
<BASE_FOLDER>\AmazonFreeRTOS\demos\nuvoton\numaker-iot-m487\numaker_iot_m487_eth\keil.
3. If want to open the Wi-Fi demo project, please select target project "aws_demos_wifi.uvproj" from

<BASE_FOLDER>\AmazonFreeRTOS\demos\nuvoton\numaker-iot-m487\numaker_iot_m487_wifi\keil.

4. From the **Project** menu, choose **Build Target**.
5. From the **Debug** menu, choose **Start/Stop Debug Session**. The **Call Stack + Locals** window opens when you start the debug session. μ Vision flashes the demo to the board, runs the demo, and stops at the beginning of the `main()` function.
6. You could set breakpoints in your project's source code, and run the code. The program counter stops like as the following:

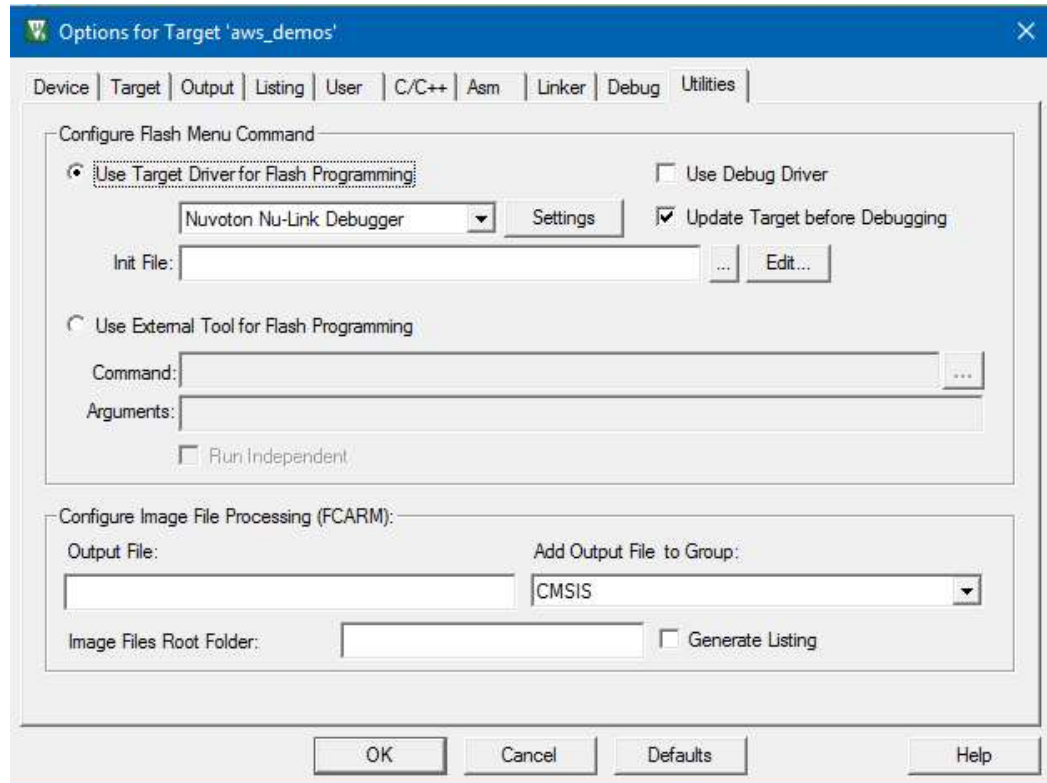


Troubleshooting the IDE Debugger Settings

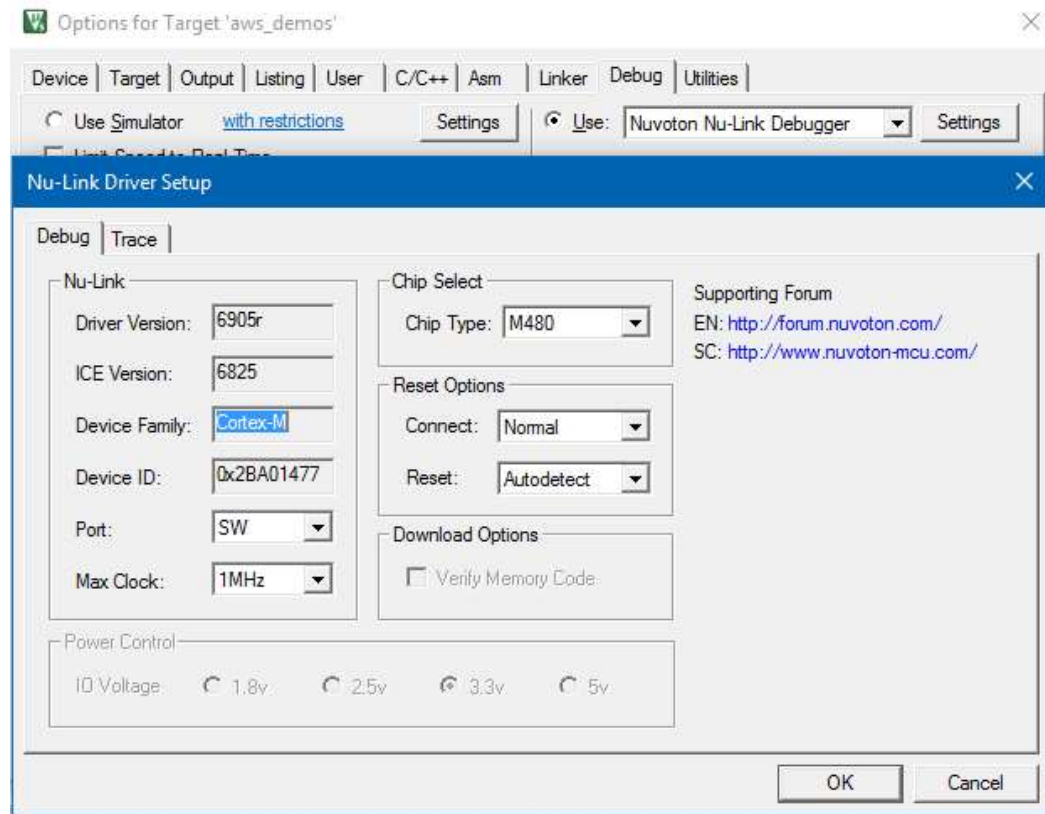
If you are having trouble debugging an application, your debugger settings might be incorrect.

To verify that your debugger settings are correct

1. Open Keil μ Vision.
2. Right-click the `aws_demos_wifi.uvproj` project, choose **Options**, and under the **Utilities** tab, ensure to set “**Nuvoton Nu-Link Debugger**” in the dropdown box.



3. choose **Options**, and under the **Debug** tab, choose **Settings**, next to “**Nuvoton Nu-Link Debugger**”. In “**Chip Type**” dropdown box, it should be “**M480**” as below:



For general troubleshooting information about Getting Started with Amazon FreeRTOS, see [Troubleshooting Getting Started](#).