Getting Started with NXP FRDM-K64F

If you do not have a FRDM-K64F board, visit the AWS Partner Device Catalog to purchase it from our partner. Use a USB cable to connect the FRDM-K64F board to your computer.

Before you begin, configure the AWS IoT and your Amazon FreeRTOS download to connect your device to the AWS Cloud. See prerequisites for instructions. In this tutorial, the path to the Amazon FreeRTOS download directory is called `BASE_FOLDER`.

Hardware requirements

1. FRDM-K64F development board
2. Ethernet cable

Setting Up Your Environment

Two toolchains are supported for the FRDM-K64F board: IAR Embedded Workbench and GNU Arm Embedded Toolchain.

Before you begin, install one of these IDEs.

To install IAR Embedded Workbench for ARM

3. Browse to Software for NXP Kits and choose **Download Software**.

   Note

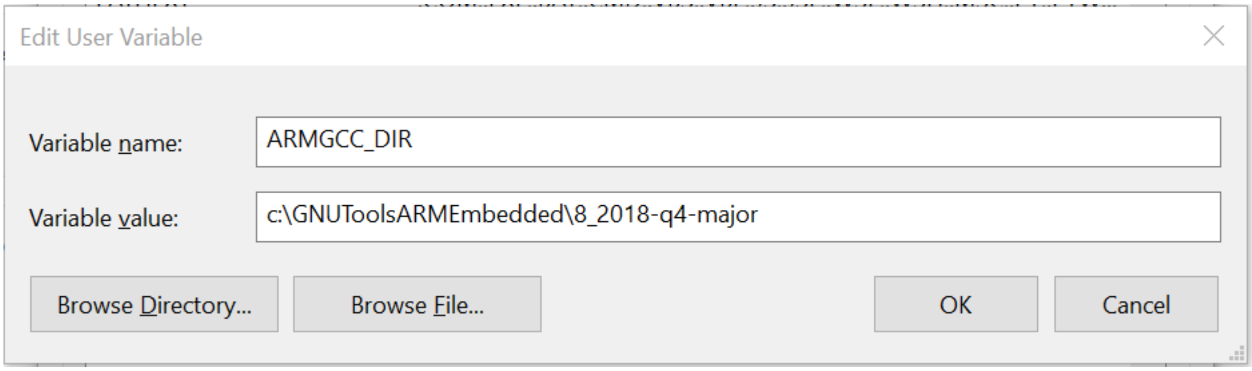
   IAR Embedded Workbench for ARM requires Microsoft Windows.

4. Unzip and run the installer. Follow the prompts.
5. In the **License Wizard**, select **Register with IAR Systems to get an evaluation license**.

To install GNU Arm Embedded Toolchain

1. GNU Arm Embedded toolchain is supported for Windows and Linux OS.

2. Download and run the GNU Arm Embedded toolchain installer from https://launchpad.net/gcc-arm-embedded on Windows OS.
   Extract gcc-arm-none-eabi-8-2018-q4-major-linux.tar.bz2. archive on Linux OS.
3. Download a cmake from https://cmake.org/.
4. Create an environment variable "ARMGCC_DIR" that contains a GCC location. For example:



Edit User Variable

Variable name: ARMGCC_DIR

Variable value: c:\GNUToolsARMEmbedded\8_2018-q4-major

   On Linux OS execute command “export ATMGCC_DIR=~/Downloads/ gcc-arm-none-eabi-8-2018-q4-major-linux” and append the command into ~/.bashrc file.

5. Builder scripts are located in "armgcc" project subdirectory. Run the "build_flexspi_nor_debug.bat" on Windows OS and “source build_flexspi_nor_debug.sh” on Linux OS.

Connecting to network

Plug an ethernet cable into the FRDM-K64F board’s Ethernet socket.

Connecting a JTAG debugger

Plug a micro USB cable to debug the SDA USB port (J26). Your PC OS should recognize and set up a new COM Port device that is used as the default UART for the development board. The debugger uses the preset CMSIS-DAP firmware. If you encounter any debugger related issues, please try to update the firmware (k20dx_frdmk64f_if_crc_legacy_0x5000.bin) according to instructions in this link.
It is possible to also use the J-Link OpenSDA - Board-Specific Firmware, available at https://www.segger.com/downloads/jlink/

Building and Running the Amazon FreeRTOS Demo Project

Importing Amazon FreeRTOS Demo into Your IDE

To import the Amazon FreeRTOS sample code into the IAR Embedded Workbench IDE, perform these steps:

1. Open the IAR Embedded Workbench and choose **Open Workspace** from the **File** menu.
2. Enter `<BASE_FOLDER>\demos\nxp\<BOARD>\iar`, into the **search-directory** text box and select **aws_demos.eww**.
3. From the **Project** menu, select **Rebuild All**.

Build the Amazon FreeRTOS sample code using CMake and GNU Arm Embedded Toolchain.

1. Enter the `<BASE_FOLDER>\demos\nxp\<BOARD>\armgcc` project subdirectory and run the "build_flexspi_nor_debug.bat" on Windows OS or “source build_flexspi_nor_debug.sh” on Linux OS.

Run the Amazon FreeRTOS Demo Project

1. Configure the Amazon FreeRTOS Demo Project by following the prerequisite steps found here:
    a. https://docs.aws.amazon.com/freertos/latest/userguide/freertos-account-and-permissions.html

   b. https://docs.aws.amazon.com/freertos/latest/userguide/get-started-freertos-thing.html
   c. https://docs.aws.amazon.com/freertos/latest/userguide/freertos-configure.html
2. Connect the USB port on the NXP board to your host computer, open a terminal program, and connect to the port identified as USB Serial Device.
3. In your IDE, from the **Project** menu, select **Build**.
4. If you are using IAR Embedded Workbench:
   a. From the **Project** menu, select **Download and Debug**.
   b. From the **Debug** menu, select **Start Debugging**.
   c. When the debugger stops at the breakpoint in `main`, select **Go** from the **Debug** menu.

   Note

   If a **J-Link Device Selection** dialog box opens, click **OK** to continue.

5. If you are using GNU Arm Embedded Toolchain with JLink:
   a. Download and install Ozone – The J-Link Debugger from https://www.segger.com/downloads/jlink/#Ozone.
   b. Run the Ozone and select "File > New Project Wizard" to download and debug the application. In first step, set the "Target device" to "MK64FN1M0xxx12". In the next step, set the "Target interface" to "SWD". In the last step, set the "Program File" to the application binary located in `<BASE_FOLDER>\demos\nxp\<BOARD>\armgcc\<TARGET>\aws_demos` and click "Finish".
   c. Run "Debug > Start Debugging > Download and Reset Program".
6. When the debugger stops at the breakpoint in `main()`, select **Continue** from the **Debug** menu.

You can use the MQTT client in the AWS IoT console to monitor the messages that your device sends to the AWS Cloud.

To subscribe to the MQTT topic with the AWS IoT MQTT client:

1. Sign in to the AWS IoT console.
2. In the navigation pane, select **Test** to open the MQTT client.

3. In **Subscription topic**, enter `freertos/demos/echo`, and select **Subscribe to topic**.