GigaDevice Semiconductor Inc.

GD32207i-EVAL
FreeRTOS
Getting Started Guide

Revision 1.1

(June 2020)

Getting started with the GD32207i-EVAL

This tutorial provides instructions for getting started with the Gigadevice GD32207i-EVAL board. If you do not already have the Gigadevice GD32207i-EVAL board, visit the Gigadevice website to purchase one.

Before you begin, you must configure AWS IoT and your FreeRTOS software to connect your development board to the AWS Cloud. For instructions, see First steps in FreeRTOS User Guide. Then download FreeRTOS code for Gigadevice from GitHub. In this tutorial, the path to the FreeRTOS download directory is referred to as <*amazon-freertos*>.

Overview

This tutorial guides you through the following steps:

1. Install software on your host machine for developing and debugging embedded applications for your microcontroller board.
2. Compile the FreeRTOS demos application.
3. Set up GD32207i-EVAL board.
4. Load the application image to your board and run the application.

Set Up Your Development Environment

Using keil MDK v5 to compile FreeRTOS demos for GD32207i-EVAL board. You can download keil MDK v5 at Keil MDK website. The Keil MDK v5 Essential, Plus, or Pro version should also work for the GD32F20x (Cortex-M3 core) MCU.

To install the development tool for the GD32207i-EVAL board

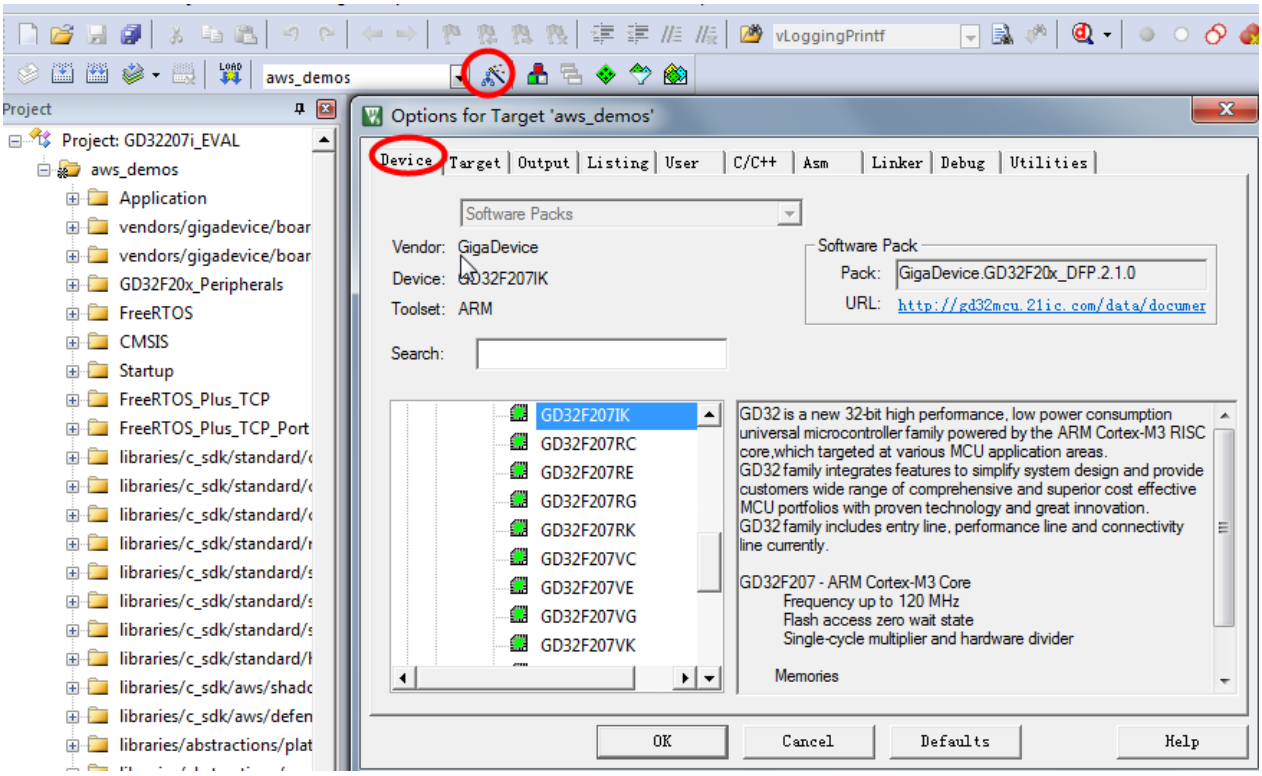
1. Download the Keil MDK.
2. Install the Keil MDK on your host machine by using your license. The Keil MDK includes the Keil µVision IDE, a C/C++ compilation toolchain, and the µVision debugger.
3. Install the GD32F20x Series Pack file from Pack Installer window of Keil IDE, you can also get pack file from the Gigadevice website.

Set up GD32207i-EVAL board correctly

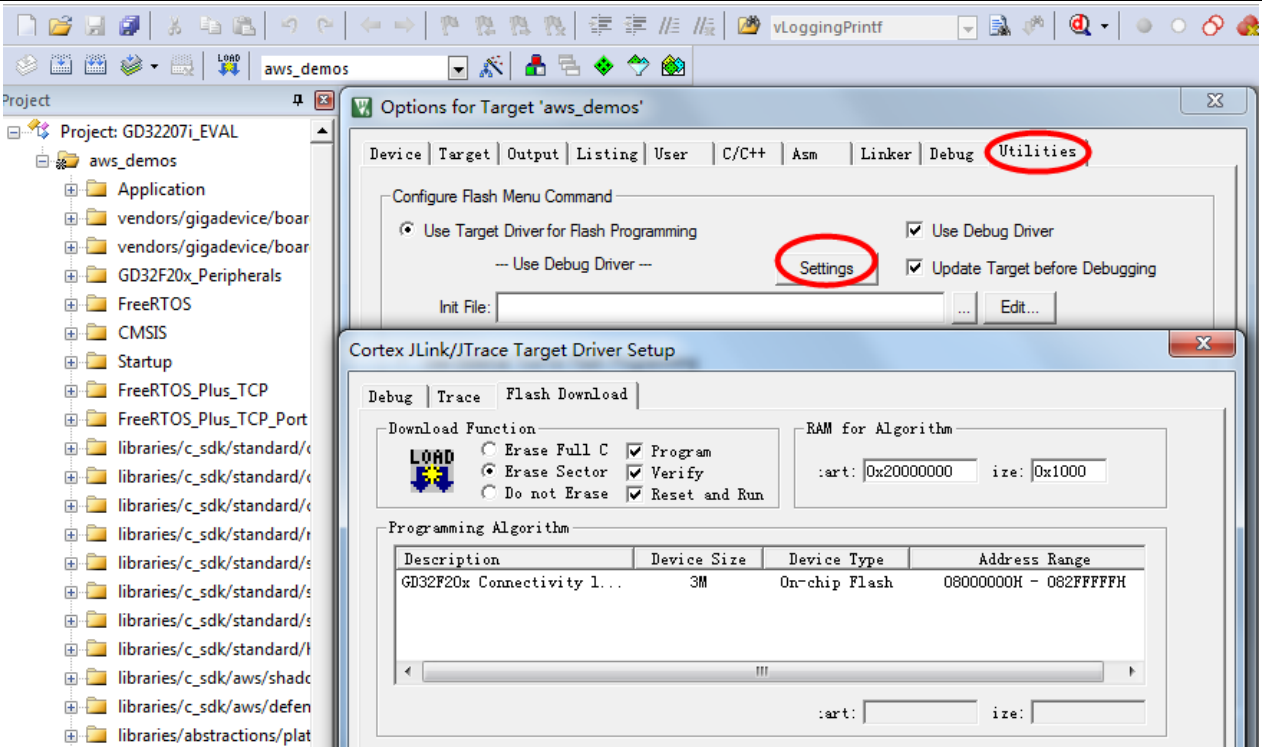
Make sure the jumper cap on the development board is set correctly. JP18, JP19, JP20,JP22,JP23 jump to ‘Eth’. JP5 jump to ‘USART0’.

Build the FreeRTOS Demo Project

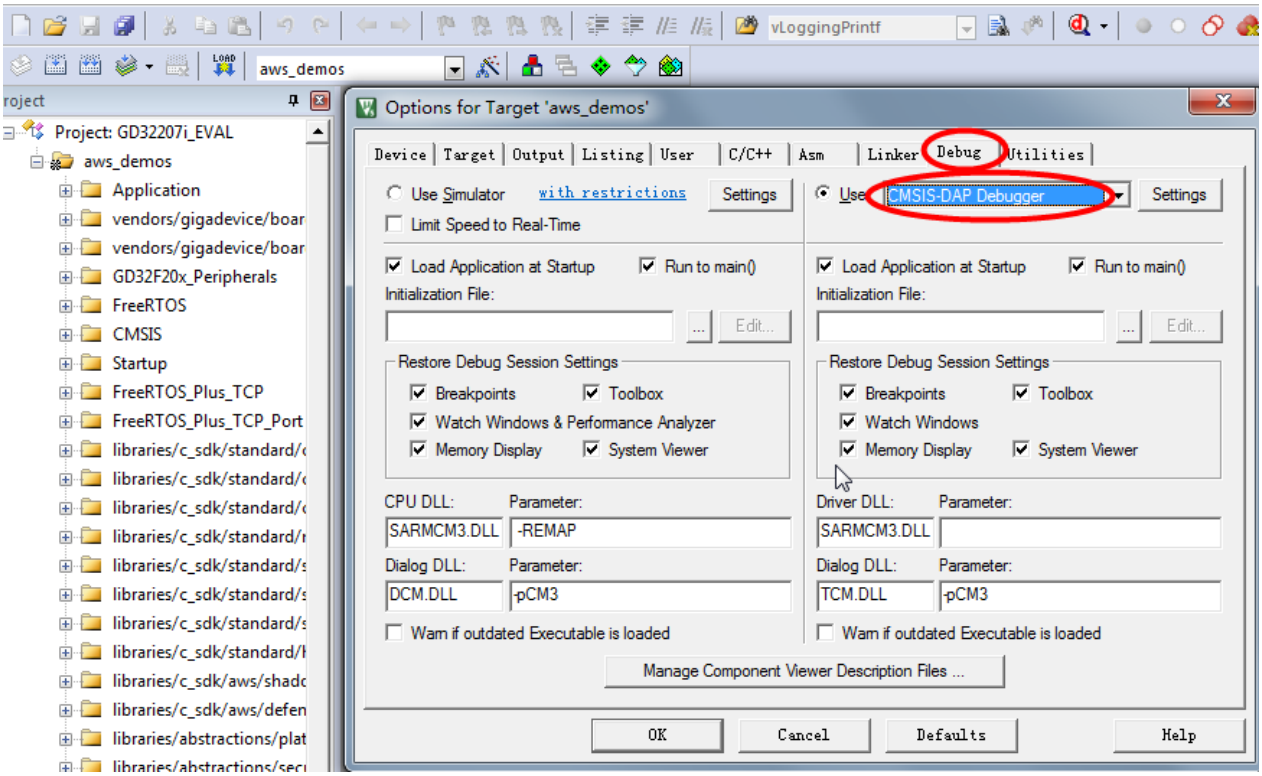
1. Make sure the Keil MDK v5 is installed correctly.
2. Open aws_demos folder in the <*amazon-freertos*> \ *projects* \ *gigadevice* \ *gd32207i_eval* \ *uvision* directory.
3. Double click GD32207i_EVAL.uvprojx project file to open demo project.
4. To make sure that your settings are correct to flash the chip, click the 'Options for Target' in the project navigation pane.
5. On the 'Device' tab, verify the chip type is set to GD32F207IK.



6. On the 'Utilities' tab, verify that 'Use Target Driver for Flash Programming' is selected, then click 'Settings' and verify Programming Algorithm was chosen to 'GD32F20x Connectivity line FMC'.



7. On the ‘Debug’ tab, you can select the corresponding debugger as required. The default debugger is CMSIS-DAP Debugger.



8. Click ‘Build’ button in the project navigation pane after the above operations are completed.

Run the FreeRTOS Demo Project

1. Click the ‘Download’ button in the project navigation pane to flash the program to the GD32207i-

EVAL board after the correct compilation.

2. You can view the demo running result through the serial port.

```
FreeRTOS App Ver:90002
FreeRTOS_IPInit
vTaskStartScheduler

Network buffers: 16 lowest 16

Write certificate...


IP Address: 192.168.62.138
Subnet Mask: 255.255.255.0
Gateway Address: 192.168.62.1
DNS Server Address: 208.67.222.222


[INFO ][DEMO][2323] ---------STARTING DEMO---------

[INFO ][INIT][2328] SDK successfully initialized.
[INFO ][DEMO][2332] Successfully initialized the demo. Network type for the
demo: 4
[INFO ][MQTT][2340] MQTT library successfully initialized.
[INFO ][DEMO][2345] MQTT demo client identifier is dyr_thing (length 9).
Network buffers: 15 lowest 15

DNS[0x06BF]: The answer to 'ak42ahou55ic7.iot.cn-north-1.amazonaws.com.cn'
```

```
[INFO ][MQTT][8004] (MQTT connection 200144c0) UNSUBSCRIBE operation
scheduled.
[INFO ][MQTT][8011] (MQTT connection 200144c0, UNSUBSCRIBE operation
20014660) Waiting for operation completion.

[INFO ][MQTT][8066] (MQTT connection 200144c0, UNSUBSCRIBE operation
20014660) Wait complete with result SUCCESS.
[INFO ][MQTT][8076] (MQTT connection 200144c0) Disconnecting connection.
[INFO ][MQTT][8083] (MQTT connection 200144c0, DISCONNECT operation
20014660) Waiting for operation completion.
[INFO ][MQTT][8093] (MQTT connection 200144c0, DISCONNECT operation
20014660) Wait complete with result SUCCESS.
[INFO ][MQTT][8103] (MQTT connection 200144c0) Connection disconnected.
[INFO ][MQTT][8109] (MQTT connection 200144c0) Network connection closed.
[INFO ][MQTT][8133] (MQTT connection 200144c0) Network connection
destroyed.
[INFO ][MQTT][8140] MQTT li
140] MQTT li
brary cleanup done.
[INFO ][DEMO][8144] memory_metrics::freertos_heap::before::bytes::95016
[INFO ][DEMO][8150] memory_metrics::freertos_heap::after::bytes::14096
[INFO ][DEMO][8157] memory_metrics::demo_task_stack::before::bytes::20308
[INFO ][DEMO][8164] memory_metrics::demo_task_stack::after::bytes::16632
[INFO ][DEMO][8170] Demo completed successfully.
[INFO ][INIT][8174] SDK cleanup done.
[INFO ][DEMO][8178] -------DEMO FINISHED-------
```

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

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

1. Sign in to the AWS IoT console.
2. In the navigation pane, choose ‘Test’ to open the MQTT client.
3. In ‘Subscription topic’, enter iotdemo/#, and then choose ‘Subscribe to topic’.