

**GigaDevice Semiconductor Inc.**

**GD32450Z-EVAL**  
**FreeRTOS**  
**Getting Started Guide**

1.1 版本

( 2020 年 5 月 )

## 开始使用 GD32450Z-EVAL

本文介绍了如何使用 GD32450Z-EVAL 开发板运行 FreeRTOS 示例工程。如果您还没有 GD32450Z-EVAL 开发板，可以通过访问 [GigaDevice](#) 官网购买。

在 GD32450Z-EVAL 开发板上运行 FreeRTOS 例程之前，AWS IoT 以及 FreeRTOS 需要根据用户实际情况进行配置以为链接 AWS 云做准备。具体操作请参考 FreeRTOS 用户指南 [First steps](#) 章节。请从 GitHub 上下载适配 GD32450Z-EVAL 开发板的 FreeRTOS [示例代码](#)，在后续的介绍中，使用 `<amazon-freertos>` 作为您下载的示例代码位置。

## 概述

本文介绍的主要内容有：

1. 安装 IDE 并配置开发环境；
2. 编译从 GitHub 上下载的源码；
3. 设置 GD32450Z-EVAL 开发板的跳线帽；
4. 运行示例代码。

## 开发环境设置

请使用 Keil MDK v5 版本编译示例工程。

### 安装 Keil MDK

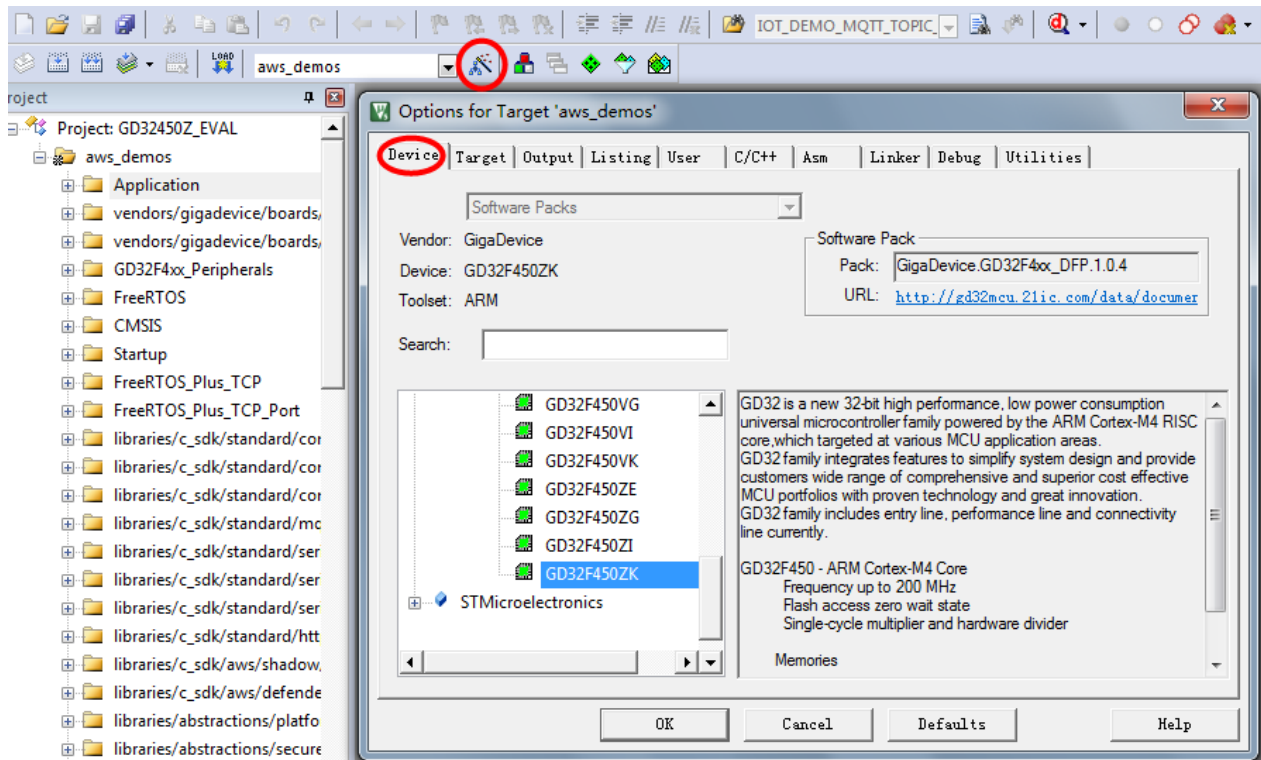
1. 从 Keil 官网中下载 Keil MDK 安装包；
2. 使用 license 正确安装软件包，Keil MDK 中包含了  $\mu$ Vision IDE、C/C++ 编译器以及  $\mu$ Vision 调试器；
3. 可以直接通过 Keil 中“Pack Installer”窗口获取对应的 GD32F4xx 系列芯片的 Pack 包，也可以通过访问 [GigaDevice](#) 官网获取。

### 设置开发板

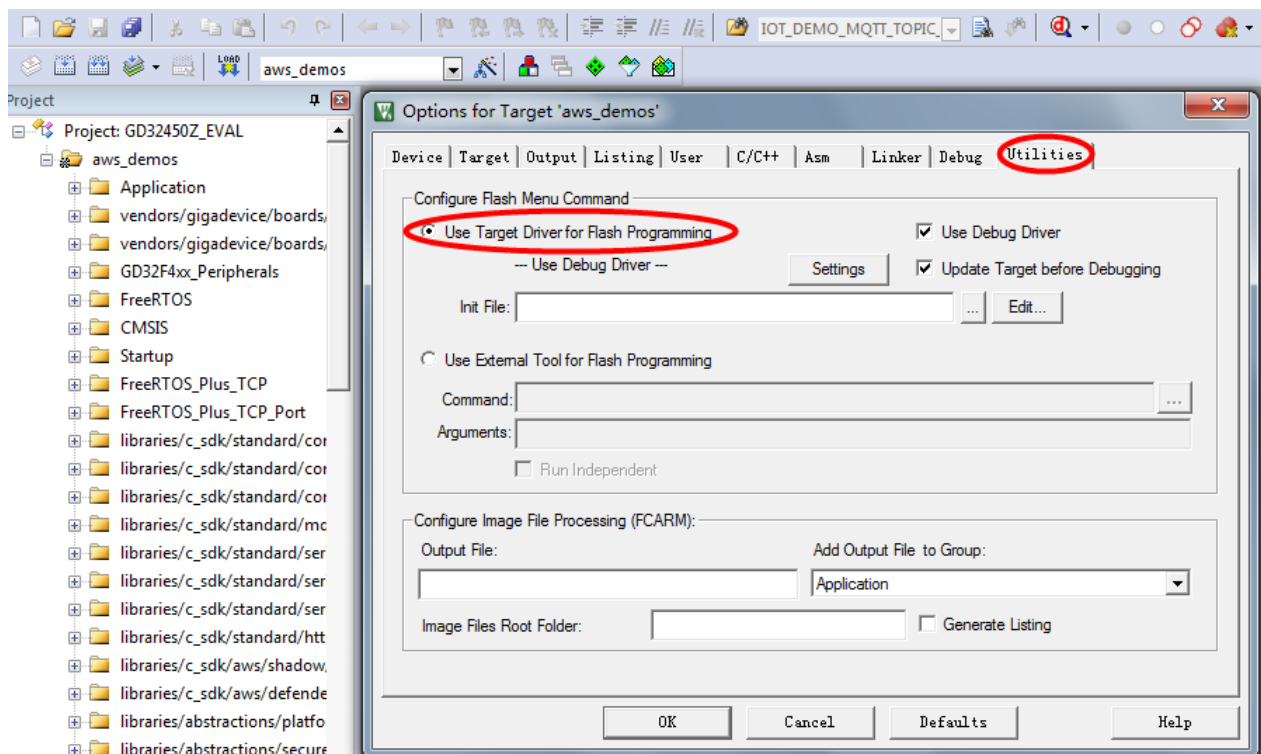
在运行示例工程前请检查开发板上的跳线帽是否设置正确，JP4、JP5、JP6 跳线帽需要选择至“Eth”选项，JP13 跳线帽需要选择至“Usart0”选项。

### 编译 FreeRTOS 示例工程

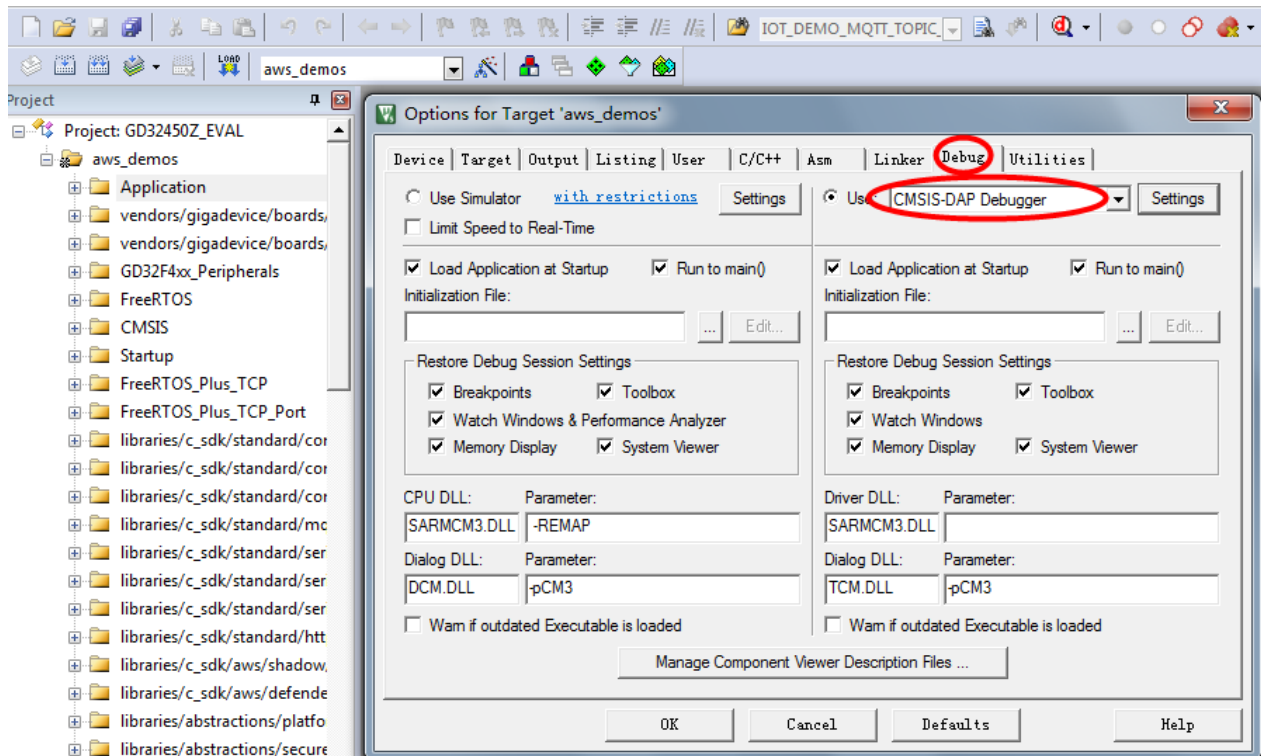
1. 确保 Keil MDK 已被正确安装；
2. 打开位于 `<amazon-freertos>\projects\gigadevice\gd32450z_eval\uvision` 路径下的 `aws_demos` 文件夹；
3. 双击 `GD32450Z_EVAL.uvprojx` 工程文件打开示例工程；
4. 打开“Options for Target”窗口中的“Device”选项卡确保芯片选型为 GD32F450ZK；



5. 选择“Utilities”选项卡，确保“Use Target Driver for Flash Programming”选项被勾选；



6. 选择“Debug”选项卡，根据实际使用的调试器选择对应调试器类型，GD32450Z-EVAL 开发板上包含有 GD Link，对应的调试器选项为 CMSIS-DAP Debugger；



7. 检查设置正确后点击导航栏中的“Build”按钮开始编译示例工程。

## 运行 FreeRTOS 示例工程

1. 正确编译后点击导航栏中的“Download”按钮下载镜像至开发板上；
2. 通过串口 1 打印出的信息检查示例工程是否正确运行；

注意：如果使用的是 2016-10 V1.1 及以前版本的开发板，开发板上串口 1 的丝印标注为 COM1，如果使用的是 2019-4 V1.1 版本的开发板，开发板上串口 1 的丝印标注为 COM0。

```
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'
(208.67.222.222) is: 192.168.62.138, 255.255.255.0, 192.168.62.1, 208.67.222.222

[INFO ][DEMO][8000] 2 publishes received.
[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 lib
rary 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-----
```

## 通过 AWS 控制台订阅 MQTT 消息

用户也可以通过登录 AWS IoT 控制台订阅 MQTT 消息来查看示例工程的运行情况，操作步骤如下：

1. 登录 AWS IoT 控制台；
2. 在控制台的导航窗口中，选择“Test”进入 MQTT 客户端；
3. 在 MQTT 客户端界面中的“Subscription topic”对话框中，输入对应的“iotdemo/#”，点击“Subscribe to topic”订阅消息；

