AN1264: Using Open Thread with FreeRTOS™

The Silicon Labs OpenThread SDK provides support for running on top of FreeRTOS, a full-featured Real-Time Operating System (RTOS) for microcontrollers and small microprocessors. Support for FreeRTOS is integrated into Simplicity Studio.

FreeRTOS is supported on the EFR32MGxx family. For documentation on Free RTOS, see http://doc.micrium.comhttps://www.freertos.org/.

KEY POINTS

- Introduces FreeRTOS.
- Describes how to install FreeRTOS.
- Describes how to work with the FreeRTOS component most effectively.

1   Getting Started

Integrating FreeRTOS into your application is simply a matter of installing the FreeRTOS component for the project in Simplicity Studio. The ot-ble-dmp sample application runs on FreeRTOS by default and you can add FreeRTOS to any OpenThread project. The ot-cli-ftd sample application is a good starting example.

1. In your project, double-click the **.slcp file** for the project in the Project Explorer to open the project window.
2. Click the SOFTWARE COMPONENTS tab to see a complete list of Component categories.
3. Find the FreeRTOS component located under RTOS > FreeRTOS in the list of components.
4. Select **FreeRTOS** and then click **Install**.
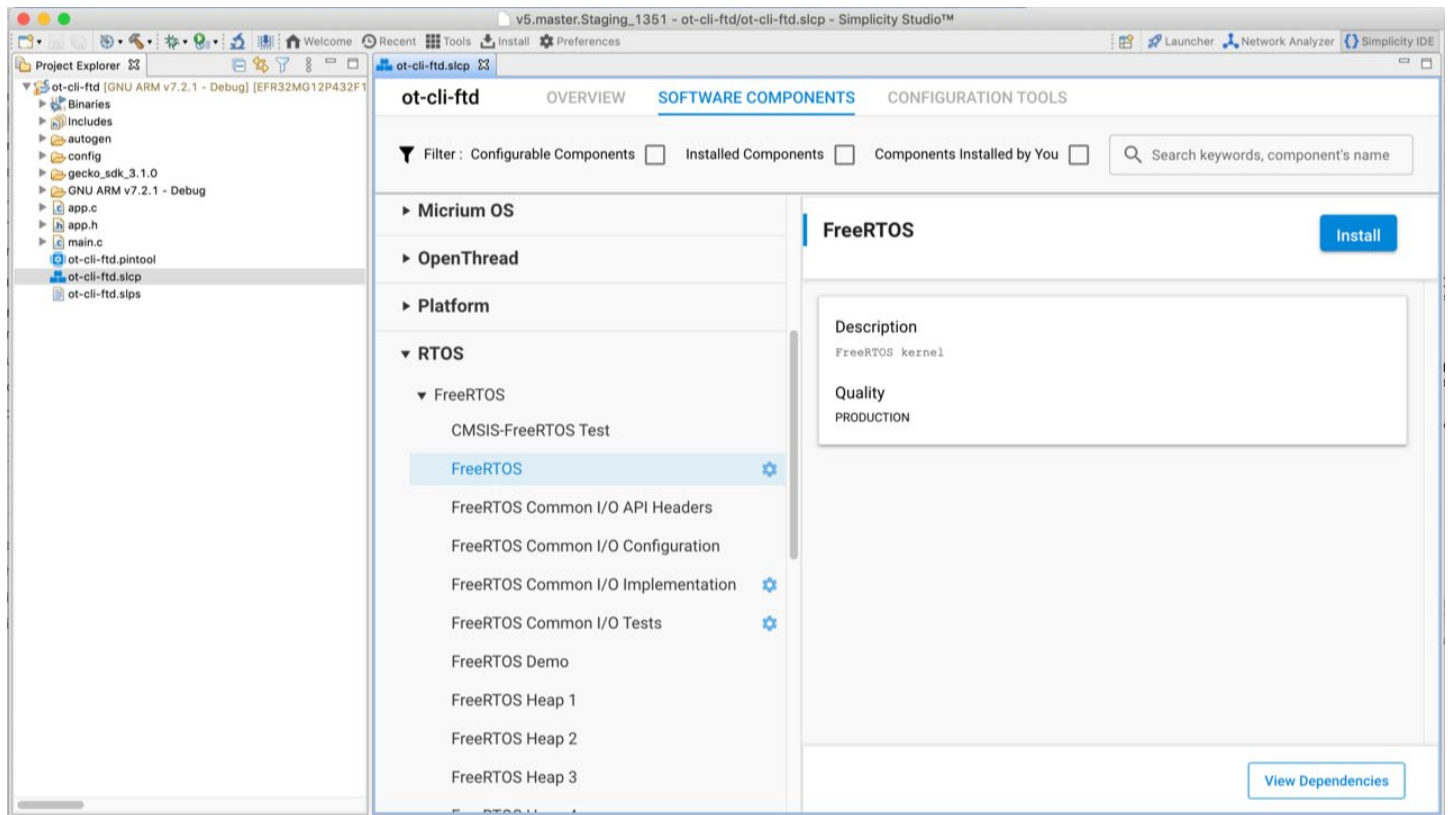
This component brings all the FreeRTOS kernel files into your project, along with some integration files and some additional components it depends on.

5. Click **View Dependencies** to display the components.

One dependency is the CMSIS-RTOS2 component, which is an RTOS abstraction layer used by the integration files. Silicon Labs recommends that application developers use the FreeRTOS API directly rather than using the CMSIS-RTOS2 API.

Another dependency is the heap implementation used by FreeRTOS. FreeRTOS comes with five different heap implementations. Each appears as a component. By default, the FreeRTOS Heap 4 component is added to the project. You can change this by selecting a different heap component and clicking **Install**. For example, FreeRTOS HEAP 3 uses the system `malloc()` and `free()` implementation and is a common choice.

2   Main Function and Task Model

The FreeRTOS component is designed to be used along with the standard Silicon Labs main.c template. The standard main function works for both bare metal and kernel-based projects and takes care of all the required system initializations and task creation.

For OpenThread running on FreeRTOS, a single task is created that runs both the OpenThread stack and the application logic. It is not safe to call the OpenThread API from other tasks.

In a bare metal OpenThread application, application logic is placed in the `app_process_action()` callback. Instead, when running on FreeRTOS, application logic is placed in the `sl_ot_rtos_application_tick()` callback.

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

IoT Portfolio
www.silabs.com/IoT

SW/HW
www.silabs.com/simplicity

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

SILICON LABS

Silicon Laboratories Inc.
400 West Cesar Chavez
Austin, TX 78701
USA

http://www.silabs.com