UG435.05: Using Real-Time Operating Systems with Silicon Labs Connect v3.x

This chapter of the *Connect v3.x User's Guide* describes how to run the Silicon Labs Connect stack on top of one of the supported Real-Time Operating Systems (RTOS). The Connect stack is delivered as part of the Silicon Labs Proprietary Flex SDK v3.0 and higher. The *Connect v3.x User's Guide* assumes that you have already installed the Simplicity Studio development environment and the Flex SDK, and that you are familiar with the basics of configuring, compiling, and flashing Connect-based applications. Refer to *UG435.01: About the Connect v3.x User's Guide* for an overview of the chapters in the *Connect v3.x User's Guide*.

The *Connect v3.x User's Guide* is a series of documents that provides in-depth information for developers who are using the Silicon Labs Connect Stack for their application development. If you are new to Connect and the Proprietary Flex SDK, see *QSG168: Proprietary Flex SDK v3.x Quick Start Guide*.

Proprietary is supported on all EFR32FG devices. For others, check the device's data sheet under Ordering Information > Protocol Stack to see if Proprietary is supported. In Proprietary SDK version 2.7.n, Connect is not supported on EFR32xG22.

KEY POINTS

- Introduces Real-Time Operating System support.
- Discusses the CMSIS-RTOSv2 API.
- Discusses the Connect Stack IPC component features.
- Describes the Virtual NCP architecture details.

1. Introduction

The Silicon Labs Connect RTOS support is available as software components in Simplicity Studio. To enable this support requires very little effort on the part of application developers.

The RTOS-specific calls of Connect and other Silicon Labs components are implemented using the ARM CMSIS-RTOSv2 API. This allows support of any RTOS implementation (or kernel) that provides the CMSIS-RTOSv2 API. However, Silicon Labs recommends using the implementations that are available in Simplicity Studio as components in the RTOS group, because these were validated against Silicon Labs stacks, including Connect. As of this writing, Micrium OS and FreeRTOS are supported in Flex SDK 3.1. For more details on CMSIS-RTOSv2, see the CMSIS-RTOS2 Documentation.

Silicon Labs does not recommend using the CMSIS-RTOSv2 API from the application unless the developed application needs to support multiple kernels. If the application needs to support a single RTOS, using the API provided by the kernel is simpler and cleaner.

For details on how to use the RTOS implementation of your choice, refer to the documentation for the given RTOS. For documentation on the Micrium OS, see http://doc.micrium.com.

2. Getting Started

Any Connect application can be easily turned into an application running on an RTOS. To complete this task, install the RTOS component of your choice—for example, **Micrium OS kernel** or **FreeRTOS**. This will install all dependencies in the connect stack required to support RTOS operation.

The default setup should be good for most usecases, but if you need to fine-tune Connect's RTOS implementation, some options are configurable in `config/cmsis-rtos-ipc-config.h`.

- `EMBER_AF_PLUGIN_CMSIS_RTOS_CPU_USAGE`: If 1, the Application Framework compiles in some additional code that allows certain debug tools to have extended debug information from the OS.

- `EMBER_AF_PLUGIN_CMSIS_RTOS_CONNECT_STACK_PRIO`: The priority of the task running the stack. Note that this is defined in CMSIS-RTOSv2 priority.

- `EMBER_AF_PLUGIN_CMSIS_RTOS_CONNECT_STACK_SIZE`: The size in bytes of the call stack used by the stack task.

- `EMBER_AF_PLUGIN_CMSIS_RTOS_APP_FRAMEWORK_PRIO`: The priority of the task running the application. Note that this is defined in CMSIS-RTOSv2 priority.

- `EMBER_AF_PLUGIN_CMSIS_RTOS_APP_FRAMEWORK_STACK_SIZE`: The size in bytes of the call stack used by the Application Framework task.

- `EMBER_AF_PLUGIN_CMSIS_RTOS_MAX_CALLBACK_QUEUE_SIZE`: Defines the maximum supported number of simultaneous callback messages from the stack task to the application tasks.

Installing any RTOS can change the behavior of other components as well. For example, the CLI Core component will also create an RTOS task for itself, while normally it is called from the main while loop.

3. Virtual NCP Architecture Details

With the CMSIS Stack IPC component enabled, the Connect stack runs within an RTOS task while the Application Framework code runs within a separate RTOS task. This two-task model is also known as the **virtual Network Co-Processor** architecture (**vNCP**) because the processing and communication between the two tasks is the same as for the host/NCP architecture. The difference is that instead of running on separate processors and passing messages to each other through the serial port, the application and NCP run on the same processor but in different RTOS tasks. Message passing is handled using RTOS message queues or protected global data structures and is transparent to the application.

By enabling an RTOS Kernel component, the requested kernel code is added to the project. The CMSIS Stack IPC component will automatically start the two required main tasks for the stack to operate:

- A **Connect stack task** (higher priority task, 39 by default) responsible for running the Connect stack. This task is responsible for the following operations:
  - Periodically tick the Connect stack by invoking the `emberTick()` API.
  - Process incoming IPC (inter-process communication) commands from application tasks (if any) and send out a response.
  - Send callback IPC commands (if any) to the application tasks.
  - Attempt whenever possible to suspend itself to allow lower priority application tasks to run.
- An **Application Framework task** (lower priority, 38 by default) responsible for running the Application Framework code. This task is responsible for the following operations:
  - Periodically calls `emberAfTickCallback()` and `emberAfTick()`, ticking the application and connect components respectively.
  - Run application events.
  - Process incoming callback IPC commands (if any) from the stack task.
  - Attempt whenever possible to suspend itself to allow lower priority custom application tasks to run.

Of course, you can create additional RTOS tasks, but Silicon Labs recommends that you keep the Connect stack task priority as high as possible.

Custom application tasks can call stack APIs freely because the underlying IPC protocol ensures that all stack APIs are thread-safe. Keep in mind that only stack APIs are channeled through the above described vNCP. HAL functions and any MCU-related code might not be thread-safe. You should rely on the documentation for these to verify whether or not they are thread-safe. For non-volatile storage, only NVM3 is supported, because neither simEEv1 nor simEEv2 are thread-safe.

Smart. Connected.
Energy-Friendly.

IoT Portfolio
www.silabs.com/products

Quality
www.silabs.com/quality

Support & Community
www.silabs.com/community

Disclaimer
Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs products shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Labs product in such unauthorized applications.
Note: This content may contain offensive terminology that is now obsolete. Silicon Labs is replacing these terms with inclusive language wherever possible. For more information, visit www.silabs.com/about-us/inclusive-lexicon-project

Trademark Information
Silicon Laboratories Inc.®, Silicon Laboratories®, Silicon Labs®, SiLabs® and the Silicon Labs logo®, Bluegiga®, Bluegiga Logo®, EFM®, EFM32®, EFR, Ember®, Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals®, WiSeConnect , n-Link, ThreadArch®, EZLink®, EZRadio®, EZRadioPRO®, Gecko®, Gecko OS, Gecko OS Studio, Precision32®, Simplicity Studio®, Telegesis, the Telegesis Logo®, USBXpress® , Zentri, the Zentri logo and Zentri DMS, Z-Wave®, and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.

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

SILICON LABS

www.silabs.com