UG235.05: Using Micrium OS (RTOS) with Silicon Labs Connect v2.x

This chapter of the *Connect v2.x User's Guide* describes how to run the Silicon Labs Connect stack on top of the Micrium OS kernel, a full-featured real-time operating system (RTOS) for embedded systems. The Connect stack is delivered as part of the Silicon Labs Proprietary Flex SDK. The *Connect v2.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 *UG235.01: Developing Code with Silicon Labs Connect v2.x* for an overview of the chapters in the *Connect v2.x User's Guide*.

The *Connect v2.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 *QSG138: Proprietary Flex SDK v2.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

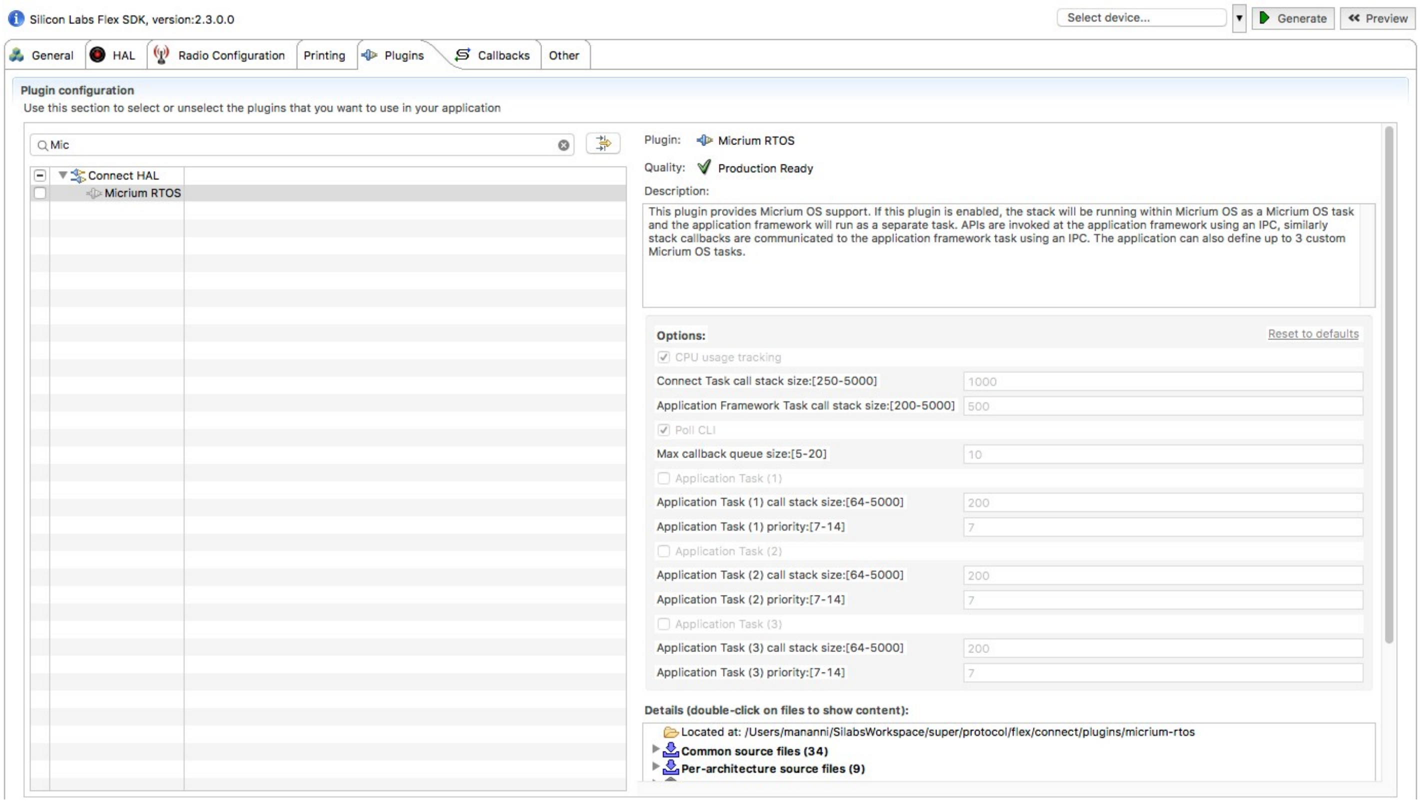
- Introduces Micrium OS support.
- Discusses the Micrium RTOS plugin features.
- Describes the Virtual NCP architecture details.

1. Introduction

The Silicon Labs Connect support for Micrium is integrated into the Silicon Labs Application Builder (AppBuilder) utility in Simplicity Studio. To enable this support requires very little effort on the part of application developers. 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 Micrium OS by simply enabling the **Micrium RTOS** plugin as shown in the following figure.



Plugin configuration
Use this section to select or unselect the plugins that you want to use in your application

Plugin: Micrium RTOS
Quality: ✓ Production Ready
Description:
This plugin provides Micrium OS support. If this plugin is enabled, the stack will be running within Micrium OS as a Micrium OS task and the application framework will run as a separate task. APIs are invoked at the application framework using an IPC, similarly stack callbacks are communicated to the application framework task using an IPC. The application can also define up to 3 custom Micrium OS tasks.

Options:

[✓] CPU usage tracking	
Connect Task call stack size:[250-5000]	1000
Application Framework Task call stack size:[200-5000]	500
[✓] Poll CLI	
Max callback queue size:[5-20]	10
[ ] Application Task (1)	
Application Task (1) call stack size:[64-5000]	200
Application Task (1) priority:[7-14]	7
[ ] Application Task (2)	
Application Task (2) call stack size:[64-5000]	200
Application Task (2) priority:[7-14]	7
[ ] Application Task (3)	
Application Task (3) call stack size:[64-5000]	200
Application Task (3) priority:[7-14]	7

Details (double-click on files to show content):
Located at: /Users/mananni/SilabsWorkspace/super/protocol/flex/connect/plugins/micrium-rtos
Common source files (34)
Per-architecture source files (9)

Figure 2.1. Micrium RTOS Plugin

The **Micrium RTOS** plugin has the following plugin options that allow application developers to customize certain settings:

- **CPU usage tracking:** If enabled, the Application Framework compiles in some additional code that allows certain debug tools to have extended debug information from the OS.
- **Connect Task call stack size:** The size in bytes of the call stack used by the stack task.
- **Application Framework Task call stack size:** The size in bytes of the call stack used by the Application Framework task.
- **Poll CLI:** If enabled, the application framework task (which is responsible for processing the incoming serial CLI commands) is allowed to yield only for small periods of time. This results in a more responsive CLI user experience.
- **Max callback queue size:** Defines the maximum supported number of simultaneous callback messages from the stack task to the application tasks.
- **Application Task (*N*)**: Application developers can also optionally enable up to three custom application tasks. For each of these tasks, application developers can specify the call stack size and the priority.

  Note: Although the allowed priority range is nominally 7-14, each custom application task must be assigned a lower priority than both the stack task and the Application Framework task.

3. Virtual NCP Architecture Details

With the Micrium RTOS plugin enabled, the Connect stack runs within a Micrium Task while the Application Framework code runs within a separate Micrium 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 Micrium tasks. Message passing is handled using Micrium message queues or protected global data structures and is transparent to the application.

By enabling the Micrium RTOS plugin, the Micrium kernel code is added to the project. The plugin code automatically starts all the OS tasks used in the configuration. At a minimum, two tasks are created:

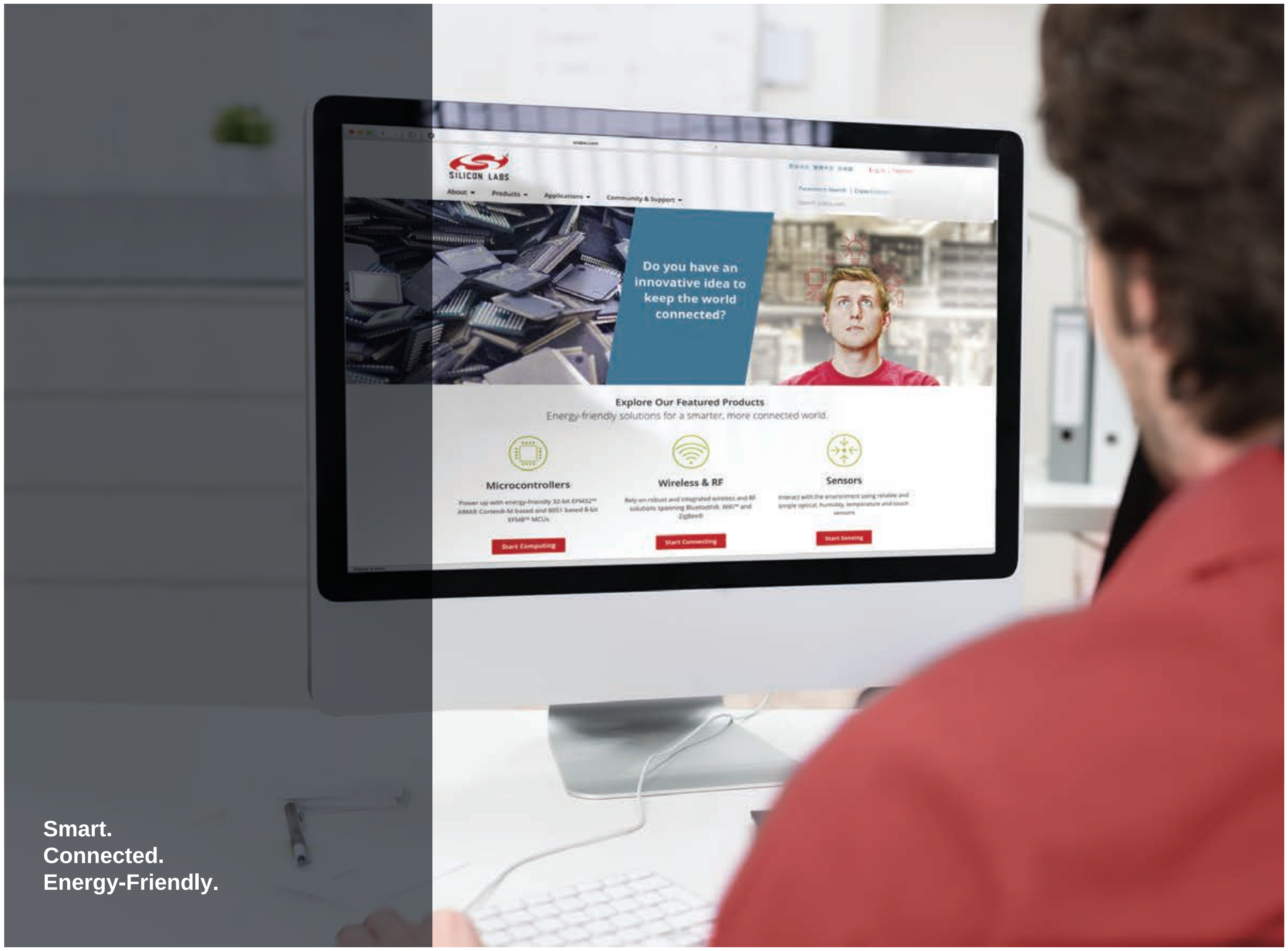
- A **Connect stack task** (highest priority task) responsible for running the Connect stack. This task is responsible for the following operations:
  - Periodically tick the Connect stack by invoking the `emberTick()` API.
  - Process incoming IPC (inter-process communication) commands from application 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** responsible for running the Application Framework code. This task is responsible for the following operations:
  - Periodically tick the application framework plugins (for those that implement a tick callback). This includes, for instance, the processing of incoming CLI commands.
  - 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.

The Micrium RTOS plugin also supports the creation of up to three custom application tasks. The custom application tasks must have a lower priority than both the Connect stack task and the Application Framework task. Each of these custom application tasks is started automatically, and should implement the following two callbacks:

- `emberAfPluginMicriumRtosAppTaskNInitCallback(void)` – This callback is invoked at initialization time to allow the *N*th custom application task to execute some initialization code.
- `emberAfPluginMicriumRtosAppTaskNMainLoopCallback(void *p_arg)` – This callback shall contain the *N*th task main loop.

Custom application tasks can call stack APIs freely because the underlying IPC protocol ensures that all stack APIs are thread-safe.

Note: As of Flex SDK 2.4, Connect does not support writing tokens from the Application Framework task. Silicon Labs intends to restore this functionality in an upcoming release. For more information on tokens, see *UG235.09: Using Other Services with Silicon Labs Connect*.


Smart.
Connected.
Energy-Friendly.
Do you have an innovative idea to keep the world connected?
Explore Our Featured Products
Energy-friendly solutions for a smarter, more connected world.
Microcontrollers
Wireless & RF
Sensors
Start Computing
Start Connecting
Start Sensing

Products
www.silabs.com/products

Quality
www.silabs.com/quality

Support and Community
community.silabs.com

SILICON LABS

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

http://www.silabs.com