Tech Talks LIVE Schedule – Presentation will begin shortly

Tuesday, Mar 9th	**Add Free RTOS to Your Bluetooth Application**
Tuesday, Mar 23rd	Unboxing the BGM220 Explorer Kit
Tuesday, Apr 13th	Discover the Security Features of Secure Vault
Tuesday, Apr 27th	Uncover Sub-GHz and Proprietary Solutions within Simplicity Studio v5
Tuesday, May 11th	Optimize Your Battery Power with BG22
Tuesday, May 25th	Get to Know OpenThread Resources and Examples
Available On Demand Now	Design with Z-Wave to Extend Your Wireless Range 1 Mile

Respond to the poll to enter to win a BG22 Thunderboard

Recording and slides will be posted to:
www.silabs.com/training

We will begin in: 4:00

SILICON LABS

WELCOME

Add FreeRTOS to Your Bluetooth Application

Claudio Filho

Silicon Labs Confidential

Agenda

1. What is FreeRTOS?
2. BLE Stack related Tasks
3. FreeRTOS Memory Allocation Options
4. Adding Application Tasks and Queues
5. FreeRTOS Demo

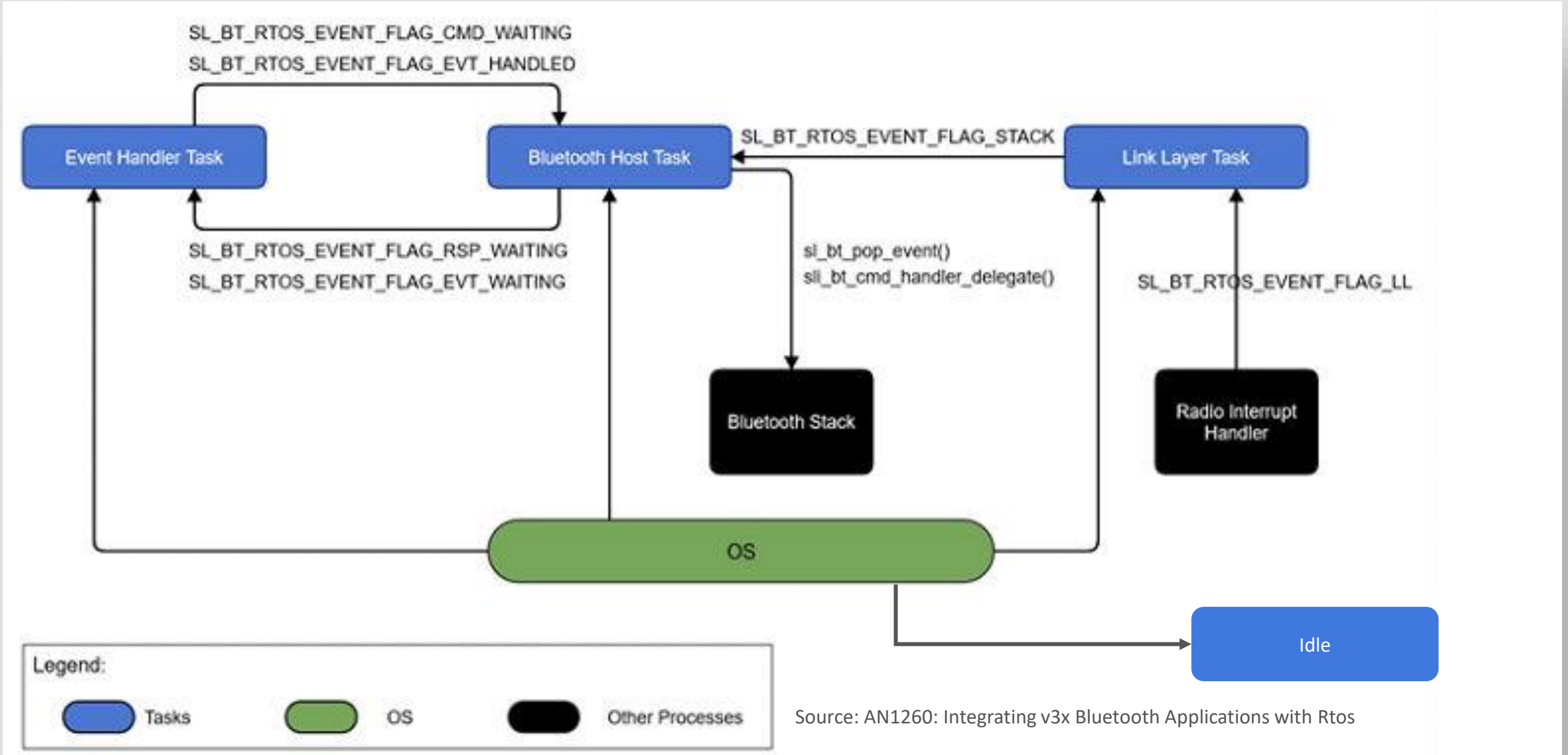
What is the FreeRTOS?

Real-time operating system for microcontrollers

- Market-leading RTOS for microcontrollers
- Developed over a 15-year period,
- Open-source license with MIT License
- Stewardship owned by Amazon AWS

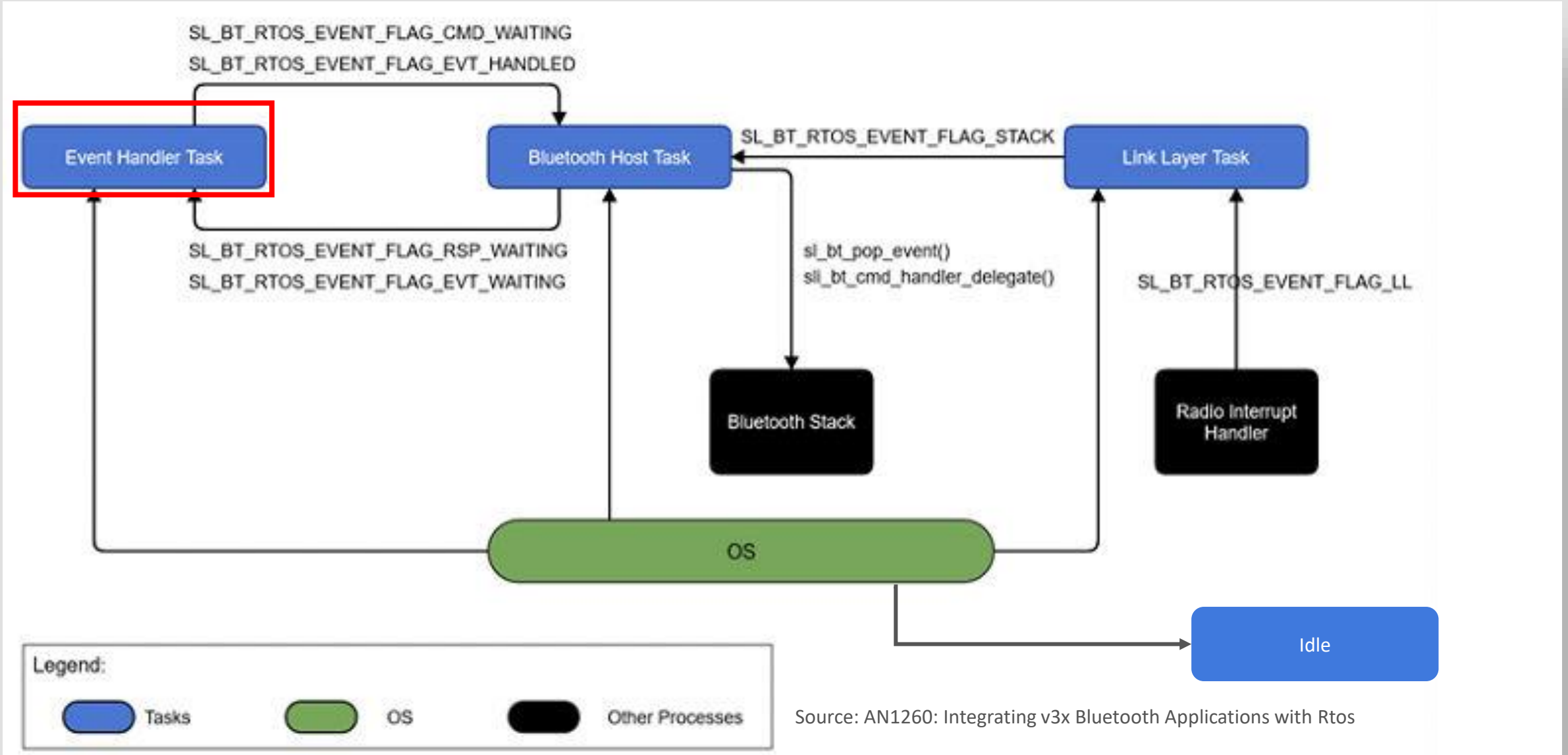
https://freertos.org

System Architecture – Bluetooth Tasks



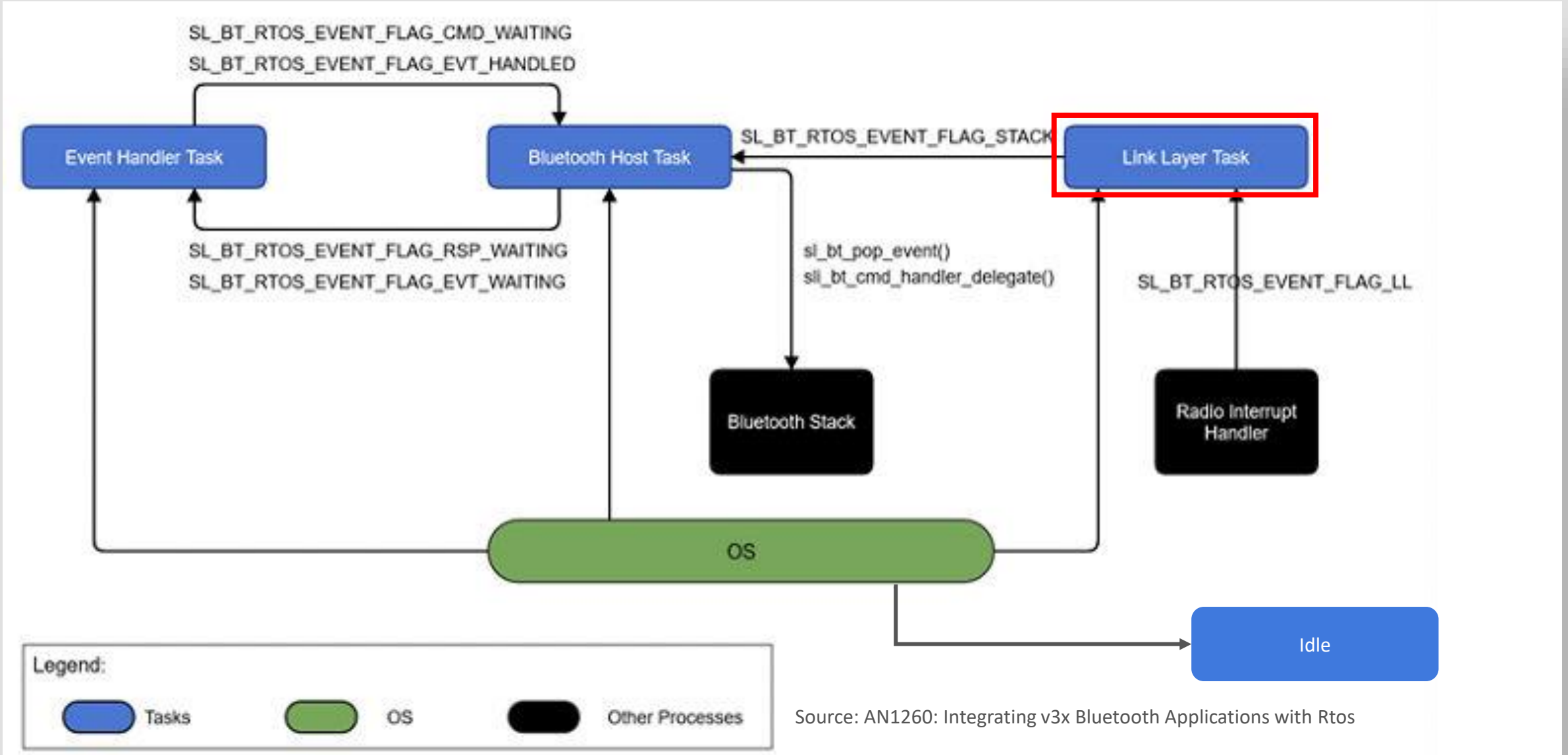
Source: AN1260: Integrating v3x Bluetooth Applications with Rtos

System Architecture – Bluetooth Tasks



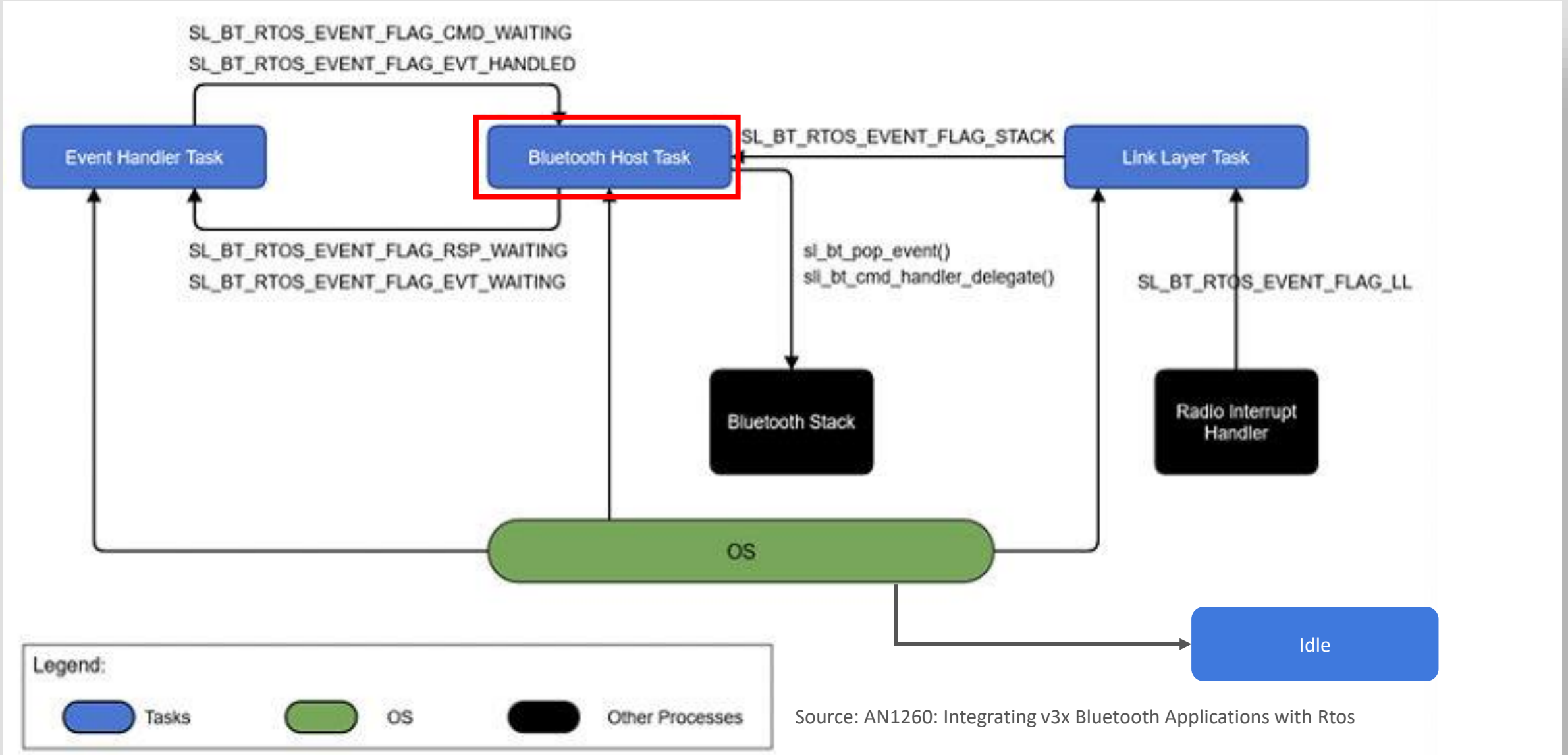
Source: AN1260: Integrating v3x Bluetooth Applications with Rtos

System Architecture – Bluetooth Tasks



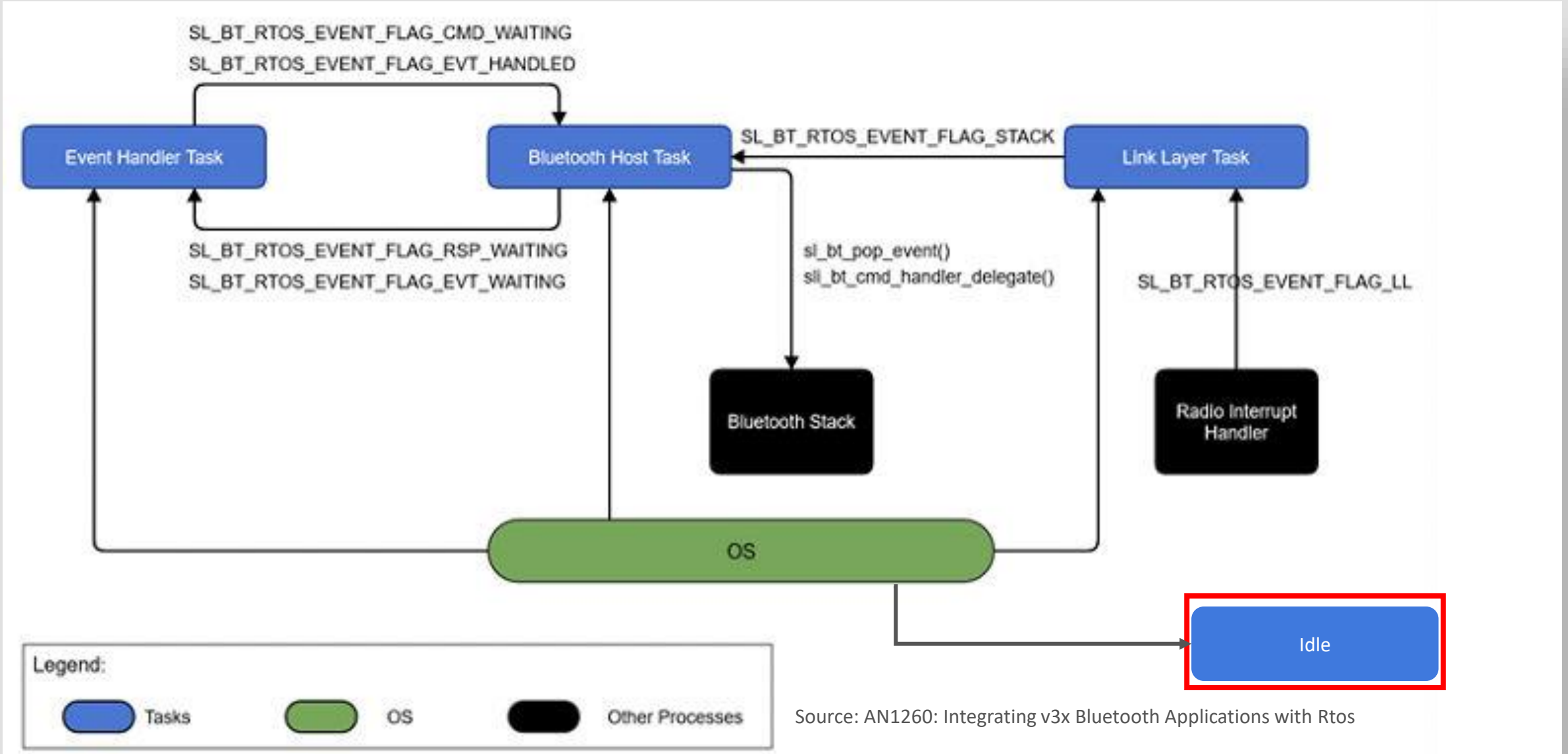
Source: AN1260: Integrating v3x Bluetooth Applications with Rtos

System Architecture – Bluetooth Tasks



Source: AN1260: Integrating v3x Bluetooth Applications with Rtos

System Architecture – Bluetooth Tasks



Source: AN1260: Integrating v3x Bluetooth Applications with Rtos

FreeRTOS Memory allocation Choices

Heap_1 – the very simplest, does not permit memory to be freed.

Heap_2 – permits memory to be freed, but does not coalescence adjacent free blocks.

Most Popular Options

Heap_3 – simply wraps the standard malloc() and free() for thread safety.

Heap_4 – coalesces adjacent free blocks to avoid fragmentation. Includes absolute address placement option.

Heap_5 – as per heap_4, with the ability to span the heap across multiple non-adjacent memory areas.

Source: FreeRTOS Developer Docs - https://www.freertos.org/a00111.html and Silicon Labs AN1260: Integrating v3x Bluetooth Applications with Rtos

Task Implementation

```
void led_task(void *p_arg)
{
  (void)p_arg;
  while (1)
    {

     sl_led_toggle(&sl_led_led0);

     vTaskDelay(500);
    }
}
```

Puts the task to Sleep

Source: https://www.freertos.org/implementing-a-FreeRTOS-task.html and Silicon Labs AN1260: Integrating v3x Bluetooth Applications with Rtos

Task Creation

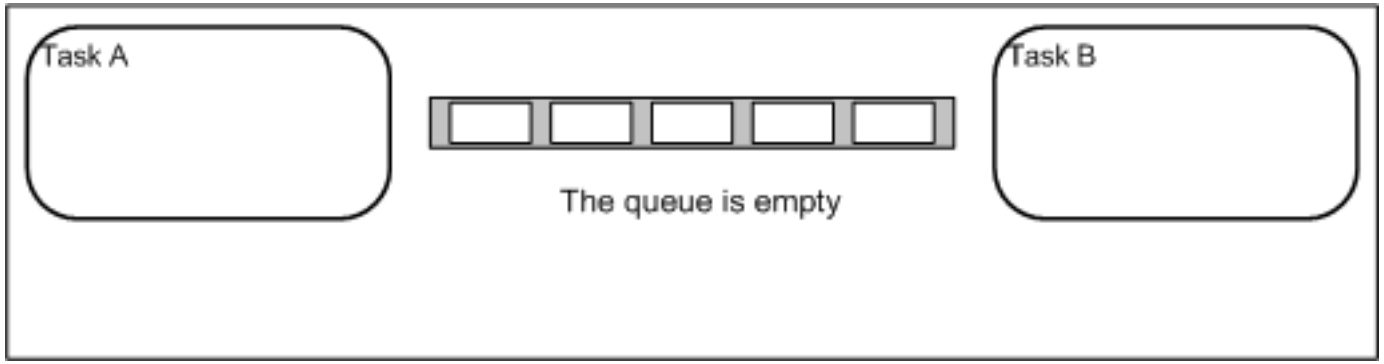
xTaskCreate(led_task, LED_TASK_NAME, LED_TASK_STACK_SIZE, argument, TASK_PRIORITY, &led_task_handle);

- Recommended to add task creations on the *app_init* function
- Each task maintains its own stack resulting in higher RAM usage.
- Priority numbers should be >=10 for application tasks
- Tasks can be deleted by calling *vTaskDelete()* and suspended by calling *vTaskSuspend()*

Source: https://www.freertos.org/taskandcr.html and Silicon Labs AN1260: Integrating v3x Bluetooth Applications with Rtos

Queues

- Provide a task-to-task, task-to-interrupt, and interrupt-to-task communication mechanism.
- Primary form of intertask communications
- It requires a Handle to be created



Source: https://www.freertos.org/Embedded-RTOS-Queues.html

Queue APIs

Queue Handle API:

xQueueHandle BLE_to_ADC_Queue_Handle = 0;

Queue creation (app_init function):

BLE_to_ADC_Queue_Handle = xQueueCreate(3,sizeof(int));

Sender task API:

xQueueSend(BLE_to_ADC_Queue_Handle,&NotiConnFlag,10);

Receiver task API:

xQueueReceive(BLE_to_ADC_Queue_Handle, &NotiConnFlag, 10)

Source: https://www.freertos.org/Embedded-RTOS-Queues.html

FreeRTOS and BLE Demo

FreeRTOS Resources

Main FreeRTOS website: https://freertos.org

FreeRTOS Documentation: https://freertos.org/Documentation/RTOS_book.html

Book: Mastering the FreeRTOS Real Time Kernel – a Hands On Tutorial Guide

FreeRTOS Reference Manual: https://freertos.org/fr-content-src/uploads/2018/07/FreeRTOS_Reference_Manual_V10.0.0.pdf

Developer Docs: https://www.freertos.org/features.html

Tasks: https://www.freertos.org/taskandcr.html

Queues: https://www.freertos.org/Embedded-RTOS-Queues.html

Documents / User Guides

Main source of Documents - docs.silabs.com

AN1260 - https://www.silabs.com/documents/public/application-notes/an1260-integrating-v3x-bluetooth-applications-with-rtos.pdf

Bluetooth Software API - https://www.silabs.com/documents/public/reference-manuals/bluetooth-api-reference.pdf

UG434: Silicon Labs *Bluetooth* ® C Application Developer's Guide - https://www.silabs.com/documents/public/user-guides/ug434-bluetooth-c-soc-dev-guide-sdk-v3x.pdf

Qsg169: Getting started with Bluetooth - https://www.silabs.com/documents/public/quick-start-guides/qsg169-bluetooth-sdk-v3x-quick-start-guide.pdf

Silicon Labs GitHub - https://github.com/siliconlabs

Hardware Peripheral Examples - https://github.com/SiliconLabs/peripheral_examples

Join our next Tech Talk LIVE

Live participants will receive a FREE BGM220 Explorer Kit after the event

Unboxing the BGM220 Kit

March 23rd | 10 AM CST / 17:00 CET

Silicon Labs Confidential

SILICON LABS

Q&A

Silicon Labs Confidential

SILICON LABS

THANK YOU

Silicon Labs Confidential