Nathan Bunnell
Embedded Real-Time Operating Systems
ECE-40290
Student ID: U08895857

Date: 9/28/2021

Assignment 1: Hello, RTOS!

The following will document completion of the first assignment for ECE-40290, with the stated goals of:

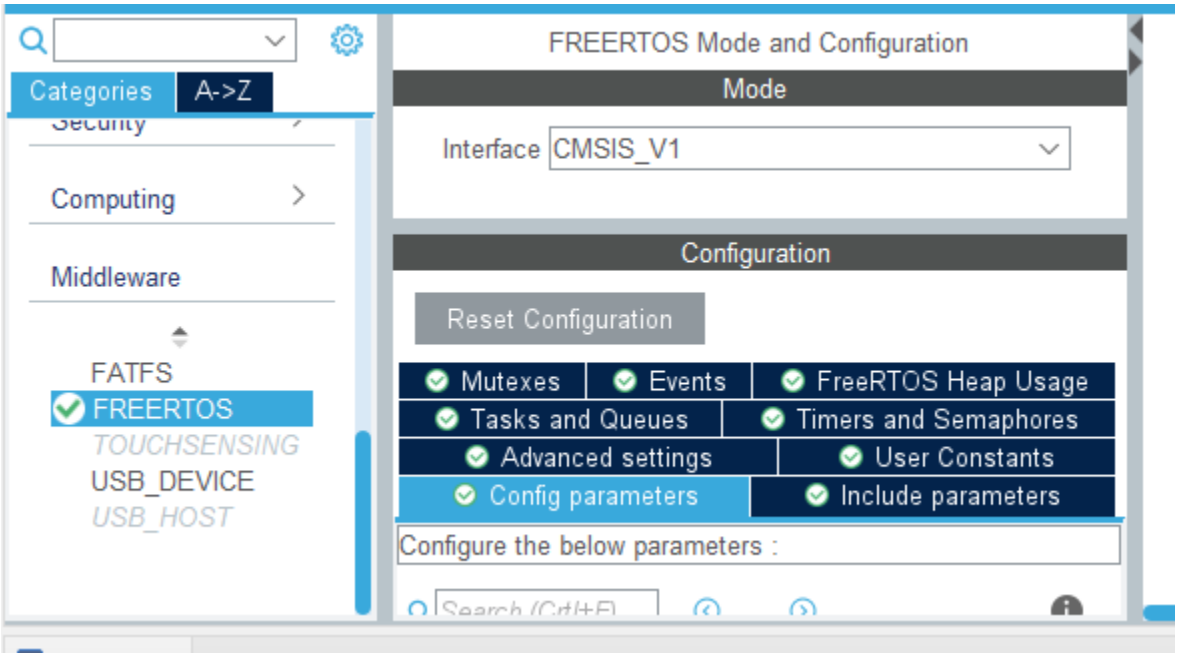
- Your skills for downloading and installing the STM32CubeIDE
- Your skills for configuring an STM32CubeIDE Project to use FreeRTOS
- Your skills for modifying the generated code the blink LED2 on the Discovery Board at a 1 second rate.

1. Your skills for downloading and installing the STM32CubeIDE

As this has been documented extensively in previous courses, I'll skip detailed steps and only note to open a new project as per the standard procedure, selecting the target Disco board and initializing all peripherals to their default settings.

2. Your skills for configuring an STM32CubeIDE Project to use FreeRTOS

In the project configurator, select the Middleware category, and choose CMSIS_V1 from the interface options. All defaults may be left as is for this example.

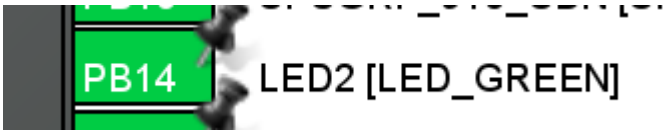


Nathan Bunnell
Embedded Real-Time Operating Systems
ECE-40290
Student ID: U08895857

Date: 9/28/2021

3. **Your skills for modifying the generated code the blink LED2 on the Discovery Board at a 1 second rate**

The target LED should have been pre-configured to an output on PB14 as the project was initiailized, we can do a quick confirmation of that by examining the pinout graphic.



From there, we can insert into the *StartDefaultTask()* function def some simple code to toggle the LED with the HAL API and implement a short delay using the *osDelay()* call defined under the RTOS APIs

```
678 void StartDefaultTask(void const * argument)
679 {
680   /* USER CODE BEGIN 5 */
681   /* Infinite loop */
682   for(;;)
683   {
684       // Simple tast to indicate the OS is doing something
685       // Toggle green LED and delay for one second
686       HAL_GPIO_TogglePin(LED2_GPIO_Port, LED2_Pin);
687       osDelay(1000);
688   }
689   /* USER CODE END 5 */
690 }
```

From there, we download and flash as in any other example and can observe the LED toggling on and off at a one second interval.

Closing Thoughts

This is my first experience with FreeRTOS, I greatly appreciate how simple it is to pull into a project via the IDE, especially compared to other middleware examples like the USB or WiFi stacks. Out of curiosity, I'd like to implement the same barebones example using the Makefile-oriented toolchain I normally would use for simple projects to compare the difficulty between the two. I'm very much looking forward to expanding on the capabilities an RTOS presents through the course of this class.