

**Embedded Real Time Operating Systems**

# ASSIGNMENT - 2

**FALL 2019**

**INSTRUCTOR: Takis**  **Zourntos**

# CLASS: ESE 3025

|  |  |  |
| --- | --- | --- |
| Student Name | Student ID | Signature\* |
| FRANCIS VIJITH A J | C0742980 | FV |
| VIJAYAPRAKASH | C0744450 | VI |
| PRATHYUSHA MUNI | C0743277 | PM |

*\*By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, an “F” in the course, or possibly more severe penalties.*

**Assignment -2**

**FreeRTOS blinky**

**Introduction:**

FreeRTOS is a real time kernel, on which we can build embedded applications to meet their hard-real time applications. It can allow different independent threads of execution to be organized and executed according to their priorities and implement hard-real time applications successfully. The scheduler is one of the key features in FreeRTOS kernel, that is responsible for deciding which task to be executed at particular time. It can suspend and resume tasks according to their priorities.

Here, in the following assignment, we use a demo project ‘FreeRTOS blinky’ that is already provided with MCUXpresso IDE, which toggles the red LED light continuously when we launch the code into our LPC 1769 board. So, we are modifying the FreeRTOS blinky to flash red, green and blue colour on LED with appropriate time delay and in same sequence as described.

**Code:**

**#include** "board.h"

**#include** "FreeRTOS.h"

**#include** "task.h"

/\* Sets up system hardware \*/

**static** **void** **prvSetupHardware**(**void**)

{

SystemCoreClockUpdate();

Board\_Init();

/\* Initial LED0 state is off \*/

Board\_LED\_Set(0, false);

Board\_LED\_Set(1, false);

      Board\_LED\_Set(2, false);

}

/\* task function for led0\*/

**static** **void** **vLEDTask0**(**void** \*pvParameters) {

bool LedState;

**while** (1) {

                       LedState = false;

Board\_LED\_Set(0, LedState);

vTaskDelay(configTICK\_RATE\_HZ);

LedState = true;

Board\_LED\_Set(0, LedState);

vTaskDelay(configTICK\_RATE\_HZ\*2);

}

}

/\* task function for led1 \*/

**static** **void** **vLEDTask1**(**void** \*pvParameters) {

bool LedState;

**while** (1) {

LedState = true;

Board\_LED\_Set(1, LedState);

vTaskDelay(configTICK\_RATE\_HZ);

LedState = false;

Board\_LED\_Set(1, LedState);

vTaskDelay(configTICK\_RATE\_HZ);

bool LedState = true;

Board\_LED\_Set(1, LedState);

vTaskDelay(configTICK\_RATE\_HZ);

}

}

/\* task function for led2 \*/

**static** **void** **vLEDTask2**(**void** \*pvParameters) {

bool LedState;

**while** (1) {

LedState = true;

Board\_LED\_Set(2, LedState);

vTaskDelay(configTICK\_RATE\_HZ\*2);

LedState = false;

Board\_LED\_Set(2, LedState);

vTaskDelay(configTICK\_RATE\_HZ);

        }

}

**int** **main**(**void**)

{

prvSetupHardware();

/\* Creating task for blinking led0(red)\*/

xTaskCreate(vLEDTask0,(**signed** **char**\*) "vTaskLed0",configMINIMAL\_STACK\_SIZE,NULL, 1, (xTaskHandle \*)NULL);

/\* Creating task for blinking led1(green) \*/

xTaskCreate(vLEDTask1, (**signed** **char**\*)"vTaskLed1",configMINIMAL\_STACK\_SIZE,NULL, 2, (xTaskHandle \*)NULL);

/\* Creating task for blinking led2(blue)\*/

xTaskCreate(vLEDTask2, (**signed** **char**\*)"vTaskLed2",configMINIMAL\_STACK\_SIZE,NULL, 3,(xTaskHandle \*)NULL);

/\* Start the scheduler \*/

vTaskStartScheduler();

/\* Should never arrive here \*/

**return** 1;

}

**Demonstration link for Assignment 2:**

[**https://youtu.be/LdCGA5zPggw**](https://youtu.be/LdCGA5zPggw)

**Conclusion:**

Here, in the above modified program, we created three tasks: vTaskLed0, vTaskLed1 and vTaskLed2 to display red, green and blue lights with priorities 1,2 and 3 with a delay. In the above video link, we can see the LPC 1769 board flashing red, green and blue lights distinctively.