Blinky With One Function

Shinu Shaji C0761203

Stebin Yohannan C0770947

Hima Sijin C0769744

Sani Thomas C0761838

Objective:

0) all LEDs OFF
1) RED LED On for 1 second
2) all LEDs are off 0.5 seconds
3) GREEN LED On for 1 second
4) all LEDs are off 0.5 seconds
5) BLUE LED On for 1 second
6) all LEDs are off for 0.5 seconds
7) go to 1)

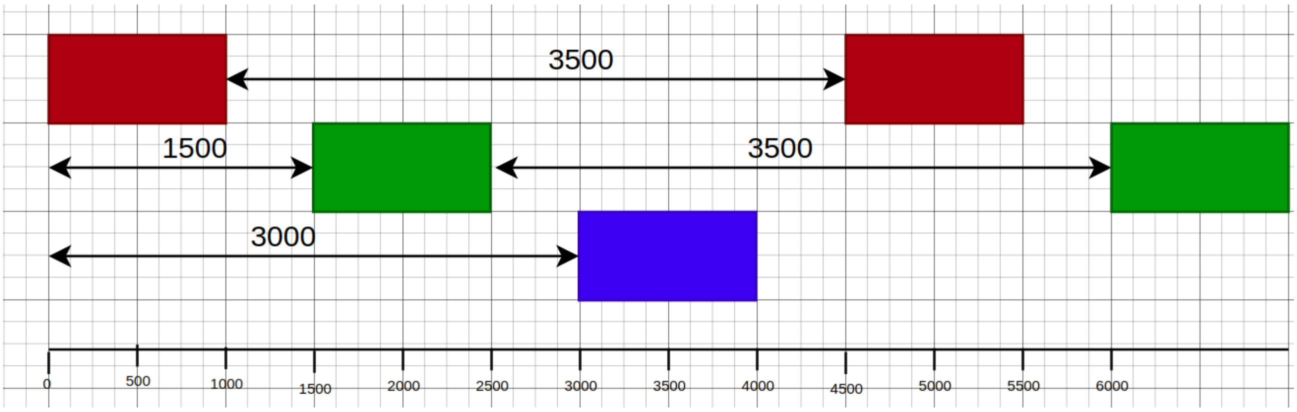
RED LED Task priority = 3
GREEN LED Task priority = 2
BLUE LED Task priority = 1

1 s = 1000 ticks

0.5 s = 500 ticks

For making delay vTaskDelay(t) function is used .

Task timing diagram



Working :

Declare a structure which store the values for led number and its corresponding initial delay

```
struct datta{
        int del;
        int led;
}data[3];

        data[0].del = 0;
        data[0].led = 0;
        data[1].del = 1500;
        data[1].led = 1;
        data[2].del = 3000;
        data[2].led = 2;
```

Make 3 led task create function and pass data[ ] to the tasks

```
        xTaskCreate(vLEDTask2, (signed char *) "vTaskLed1",
        configMINIMAL_STACK_SIZE, &data[0], (tskIDLE_PRIORITY + 1UL),
        (xTaskHandle *) NULL);
```

Create led tasks which can take the void pointer and cast to the datta struct type and retrive the led number and initial delay .

```
static void vLEDTask2(void *pvParameters) {
/*cast the void pointer*/
        struct datta *dd = pvParameters ;
        bool LedState = false;
        vTaskDelay((*dd).del);
        while (1) {
                Board_LED_Set((*dd).led, false);
                vTaskDelay(1000);
                Board_LED_Set((*dd).led, true);
                vTaskDelay(3500);
        }
        }
```

RESULT

- **Red led glows for 1 second**

- **All LED turn off for 0.5 s**
- **Green LED turn on for 1s**

- **All LED turn off for 0.5 s**
- **Blue LED turn on for 1s**

- **Back to red LED**

Video :

https://drive.google.com/file/d/1FcBMjZxTeV4Q2F22Pni915t-nRrNlkl3/view?usp=sharing

Code

```
/*
* @brief FreeRTOS Blinky example
*
* @note
* Copyright(C) NXP Semiconductors, 2014
* All rights reserved.
*

#include "board.h"
#include "FreeRTOS.h"
#include "task.h"
/*declare a structure for storing delay and led number*/
struct datta{
        int del;
        int led;
}data[3];
/* setup system hardware */
static void prvSetupHardware(void)
{
        SystemCoreClockUpdate();
        Board_Init();
        /* Initial LED0 state is off */
        Board_LED_Set(0, true);
        Board_LED_Set(1, true);
        Board_LED_Set(2, true);
        }
/*led task*/
static void vLEDTask2(void *pvParameters) {
/*cast the void pointer*/
        struct datta *dd = pvParameters ;
        bool LedState = false;
        vTaskDelay((*dd).del);
        while (1) {
                Board_LED_Set((*dd).led, false);
                vTaskDelay(1000);
                Board_LED_Set((*dd).led, true);
                vTaskDelay(3500);
        }
        }
int main(void)
{
        data[0].del = 0;
        data[0].led = 0;
        data[1].del = 1500;
        data[1].led = 1;
        data[2].del = 3000;
        data[2].led = 2;

        prvSetupHardware();

        xTaskCreate(vLEDTask2, (signed char *) "vTaskLed1",
        configMINIMAL_STACK_SIZE, &data[0], (tskIDLE_PRIORITY + 1UL),
        (xTaskHandle *) NULL);

        xTaskCreate(vLEDTask2, (signed char *) "vTaskLed2",
        configMINIMAL_STACK_SIZE,&data[1], (tskIDLE_PRIORITY + 1UL),
        (xTaskHandle *) NULL);
```

```
xTaskCreate(vLEDTask2, (signed char *) "vTaskLed3",
configMINIMAL_STACK_SIZE, &data[2], (tskIDLE_PRIORITY + 1UL),
(xTaskHandle *) NULL);

vTaskStartScheduler();
/* Should never arrive here */
return 1;
}
```