# ASSIGNMENT

Shinu Shaji C0761203

Stebin Yohannan C0770947

Hima Sijin C0769744

Sani Thomas C0761838

## Objective : create a light show that will sync to your song

## Working :

- Select a song and find the beat of the song .
- Select different beat harmonics of the song and put that as delay .
- Make a queue and send all the delays to the queue.
- Make LED tasks and run the tasks.
- Read from the queue.
- Blink the LED with the delay .

## Video Output:

https://drive.google.com/file/d/1yPkY7A__elbfXo7txG5yJoGCFoEV7F_b/view?usp=sharing

## Song selected:

https://www.youtube.com/watch?v=5O9q0NB2HL0

Program

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

```
#include "task.h"
#include "semphr.h"
#include "queue.h"
xQueueHandle que = NULL;
/* Sets up system hardware */
struct data{
	int led;
	int delay1;
	int delay2;
}dd[3];
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);
	}
/* LED1 toggle thread */
static void vLEDTask1(void *pvParameters) {
struct data d;
if(xQueueReceive(que,&d,10)){

	int led_n = (d).led;
	while (1) {
		Board_LED_Set(led_n, false);
		vTaskDelay((d).delay1);
		Board_LED_Set(led_n, true);
		vTaskDelay((d).delay2);
		}
	}
}
int main(void)
	{ int bps_r = 750;
```

```
int bps_g = 1500;
int bps_b = 3000;
que = xQueueCreate(4,sizeof(dd[0])); /* create queue of length 4 */
prvSetupHardware();
/* LED1 toggle thread */
if(que != NULL){

        dd[0].led = 0;
        dd[0].delay1 = 500;
        dd[0].delay2 = bps_r -dd[0].delay1;

        dd[1].led = 1;
        dd[1].delay1 = 100;
        dd[1].delay2 = bps_b -dd[1].delay1;

        dd[2].led = 2;
        dd[2].delay1 = 1000;
        dd[2].delay2 = bps_g -dd[2].delay1;

        xQueueSend(que,&dd[0],0); /*send the delays to the queue*/
        xQueueSend(que,&dd[1],0);
        xQueueSend(que,&dd[2],0);
        xTaskCreate(vLEDTask1, (signed char *) "vTaskLed1",
        configMINIMAL_STACK_SIZE,NULL,(tskIDLE_PRIORITY+1UL),
        (xTaskHandle *) NULL);


        xTaskCreate(vLEDTask1, (signed char *) "vTaskLed2",
        configMINIMAL_STACK_SIZE,NULL,(tskIDLE_PRIORITY + 1UL),
        (xTaskHandle *) NULL);


        xTaskCreate(vLEDTask1, (signed char *) "vTaskLed3",
        configMINIMAL_STACK_SIZE, NULL,(tskIDLE_PRIORITY + 1UL),
        (xTaskHandle *) NULL);
        vTaskStartScheduler();
```

```
        }
        /* Should never arrive here */
        return 1;
}
```