

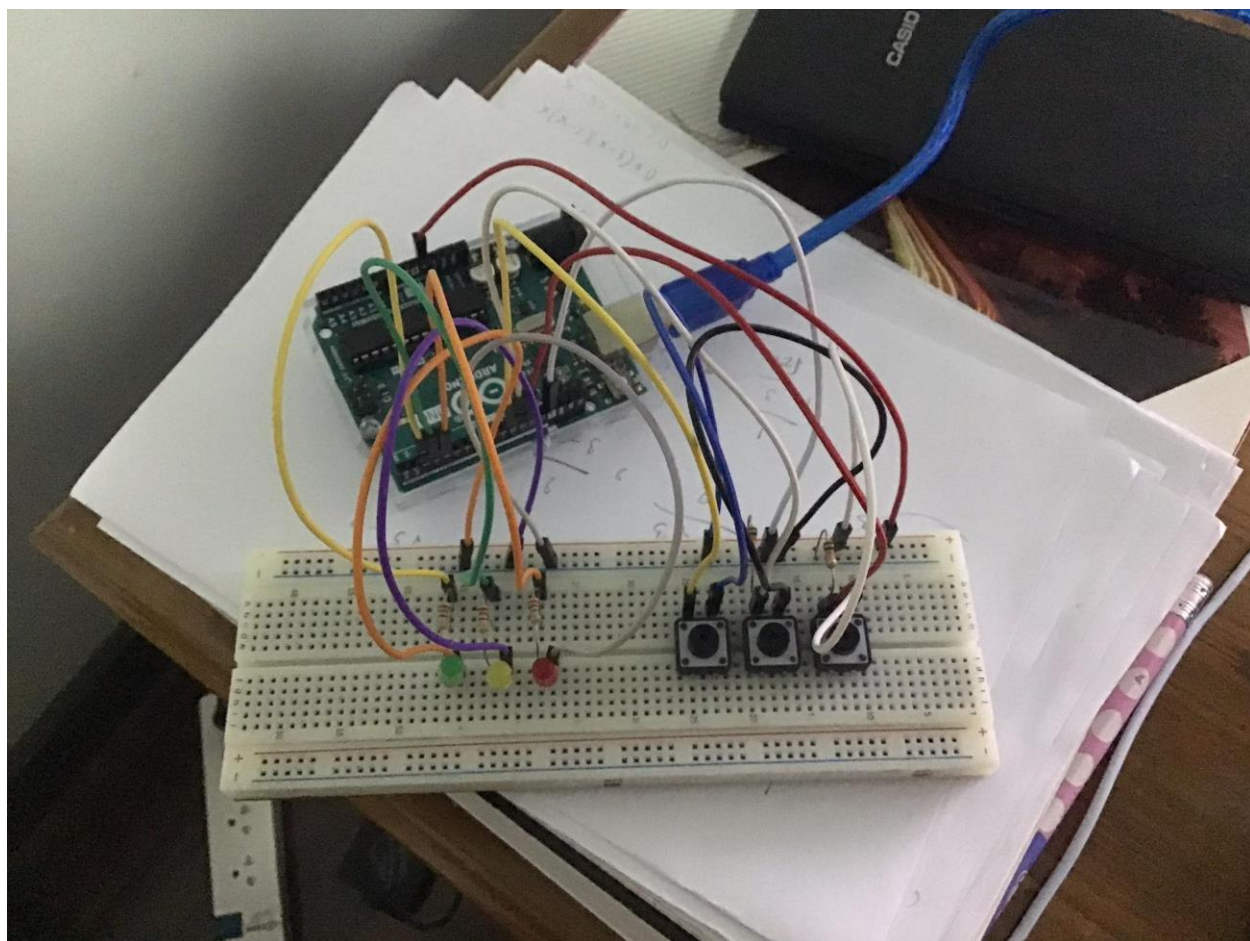
Assignment #8

ชื่อกลุ่ม : ฟ้ายรักพ่อ

สมาชิกในกลุ่ม : 64010270 นายคณพัฒน์ โพธิ์น้อย

64010315 นายธนธร แดงอ่อน

รูปถ่ายชิ้นงาน



```
#include <Arduino_FreeRTOS.h>
#include "queue.h"

#define RED_LED      4
#define YELLOW_LED   3
#define GREEN_LED    2

#define SW_1 12
#define SW_2 11
#define SW_3 10

//Check CurrentTime of LED
unsigned long now_RED = 0;
unsigned long now_YELLOW = 0;
unsigned long now_GREEN = 0;

//Type of LED Working (RED, YELLOW, GREEN)
unsigned long blink_RED = 3000; //  ON 3 Sec Click again
continue ON
unsigned long blink_YELLOW = 100; //  Blink again and again
until click again
unsigned long blink_GREEN = 500;  //  Blink 3 Time (500ms)

unsigned long debounceTime_YELLOW = 50;

int on_RED = 0;
int on_YELLOW = 0;
int on_GREEN = 0;
int off_GREEN = 0;

QueueHandle_t BlinkQueue;

void setup()
```

```

{
    Serial.begin(9600);
    BlinkQueue = xQueueCreate(5, sizeof(int32_t)); //
Length of queue and Size of queue

// Name of Function Task, Size of Stack Save for
Task(Parameters send to Task) Important Task Handle
    xTaskCreate(vSenderTask, "sw_RED", 100, SW_1, 1, NULL);
// Send
    xTaskCreate(vSenderTask, "sw_YELLOW", 100, SW_2, 1, NULL);
    xTaskCreate(vSenderTask, "sw_GREEN", 100, SW_3, 1, NULL);

    xTaskCreate(vReceiverTask, "led_RED", 100, RED_LED, 1,
NULL); // Receive
    xTaskCreate(vReceiverTask, "led_YELLOW", 100, YELLOW_LED,
1, NULL);
    xTaskCreate(vReceiverTask, "led_GREEN", 100, GREEN_LED,
1, NULL);

}

void vSenderTask(void *pvParameters)
{
    BaseType_t qStatus;
    int32_t valueToSend = 0;
    int SW = (int32_t)pvParameters;
    pinMode(SW, INPUT_PULLUP);
    while (1)
    {
        if (!digitalRead(SW))
        {
            valueToSend = SW;
            qStatus = xQueueSend(BlinkQueue, &valueToSend, 0);

```

```

        vTaskDelay(1); // Put little number for Task working
on queue now
    }
}
}

```

```

void vReceiverTask(void *pvParameters)
{
    int32_t valueReceived;
    int32_t LED = (int32_t) pvParameters;
    BaseType_t qStatus;
    const TickType_t xTicksToWait = pdMS_TO_TICKS(100);
    pinMode(RED_LED, OUTPUT);
    digitalWrite(RED_LED, LOW);
    pinMode(YELLOW_LED, OUTPUT);
    digitalWrite(YELLOW_LED, LOW);
    pinMode(GREEN_LED, OUTPUT);
    digitalWrite(GREEN_LED, LOW);

    while (1)
    {
        qStatus = xQueueReceive(BlinkQueue, &valueReceived,
xTicksToWait);
        if (qStatus == pdPASS)
        {
            if (valueReceived == SW_1 && LED == RED_LED)
            {
                on_RED = 1;
                now_RED = millis();
                digitalWrite(RED_LED, HIGH);
            }
            if (valueReceived == SW_2 && LED == YELLOW_LED &&

```

```

millis()-debounceTime_YELLOW >= 500)
{
    debounceTime_YELLOW = millis();
    on_YELLOW++;
    on_YELLOW %= 2; // Blink
}
if (valueReceived == SW_3 && LED == GREEN_LED && !
on_GREEN) // Green 0 1 0 (OFF, ON, OFF) Blink
{
    on_GREEN = 1;
    off_GREEN = 0;
}
}
LEDController();
vTaskDelay(10);
}
}

void LEDController()
{
    if(millis() - now_RED >= blink_RED && on_RED) // RED LED
    {
        on_RED = 0;
        digitalWrite(RED_LED, LOW);
    }
    if(on_YELLOW) // YELLOW LED
    {
        if(millis() - now_YELLOW >= blink_YELLOW)
        {
            digitalWrite(YELLOW_LED, digitalRead(YELLOW_LED) ^ 1);
            now_YELLOW = millis();
        }
    }
}
else

```

```
{
    digitalWrite(YELLOW_LED, LOW);
}
if(on_GREEN)// GREEN LED
{
    if(millis() - now_GREEN >= blink_GREEN)
    {
        digitalWrite(GREEN_LED, digitalRead(GREEN_LED) ^ 1);
        off_GREEN++;
        now_GREEN = millis();
    }
    if(off_GREEN >= 6)
    {
        on_GREEN = 0;
        off_GREEN = 0;
    }
}
}

void loop()
{

}
```