Assignment 8

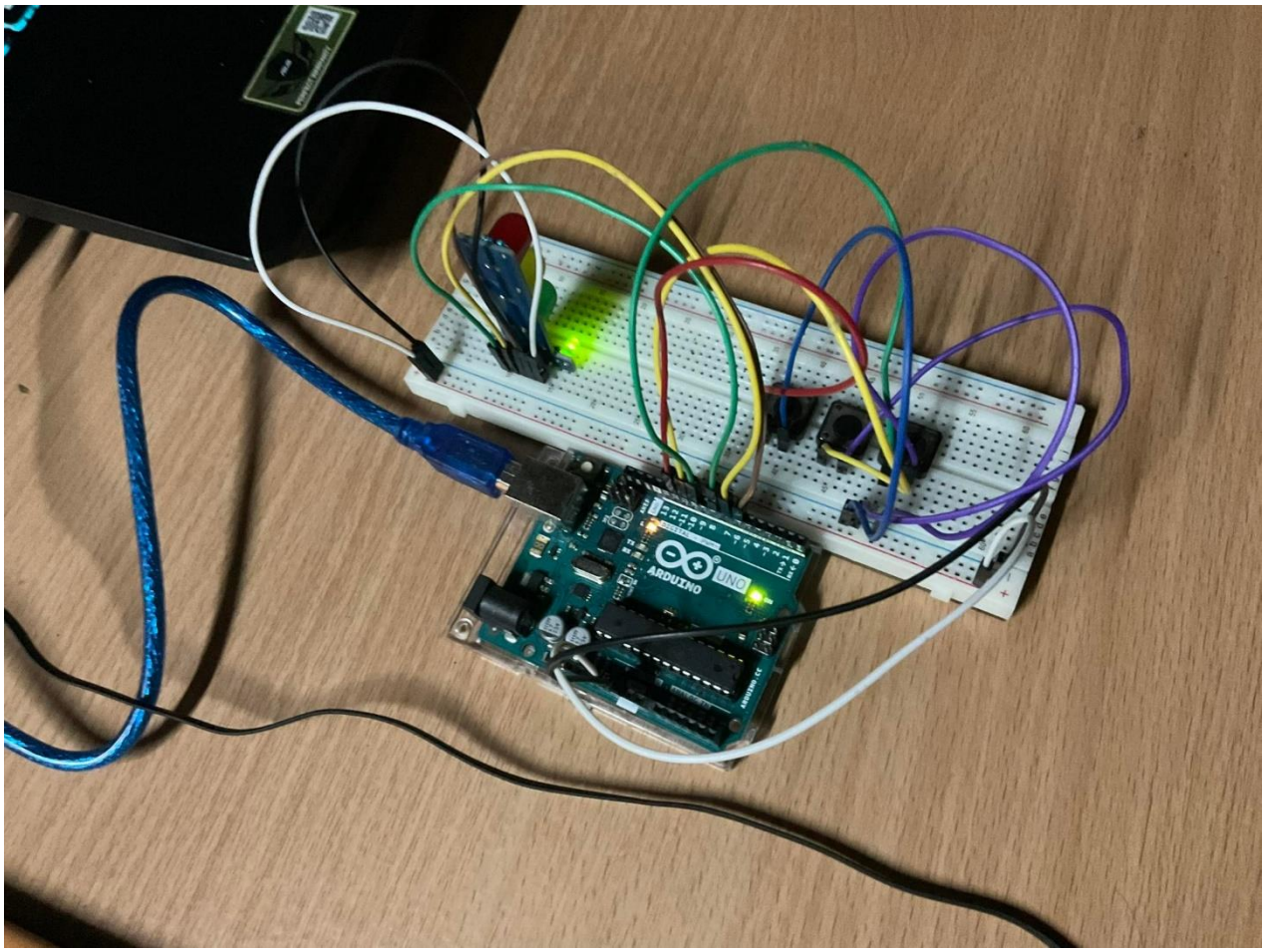
ชื่อกลุ่ม : 9 A.M.

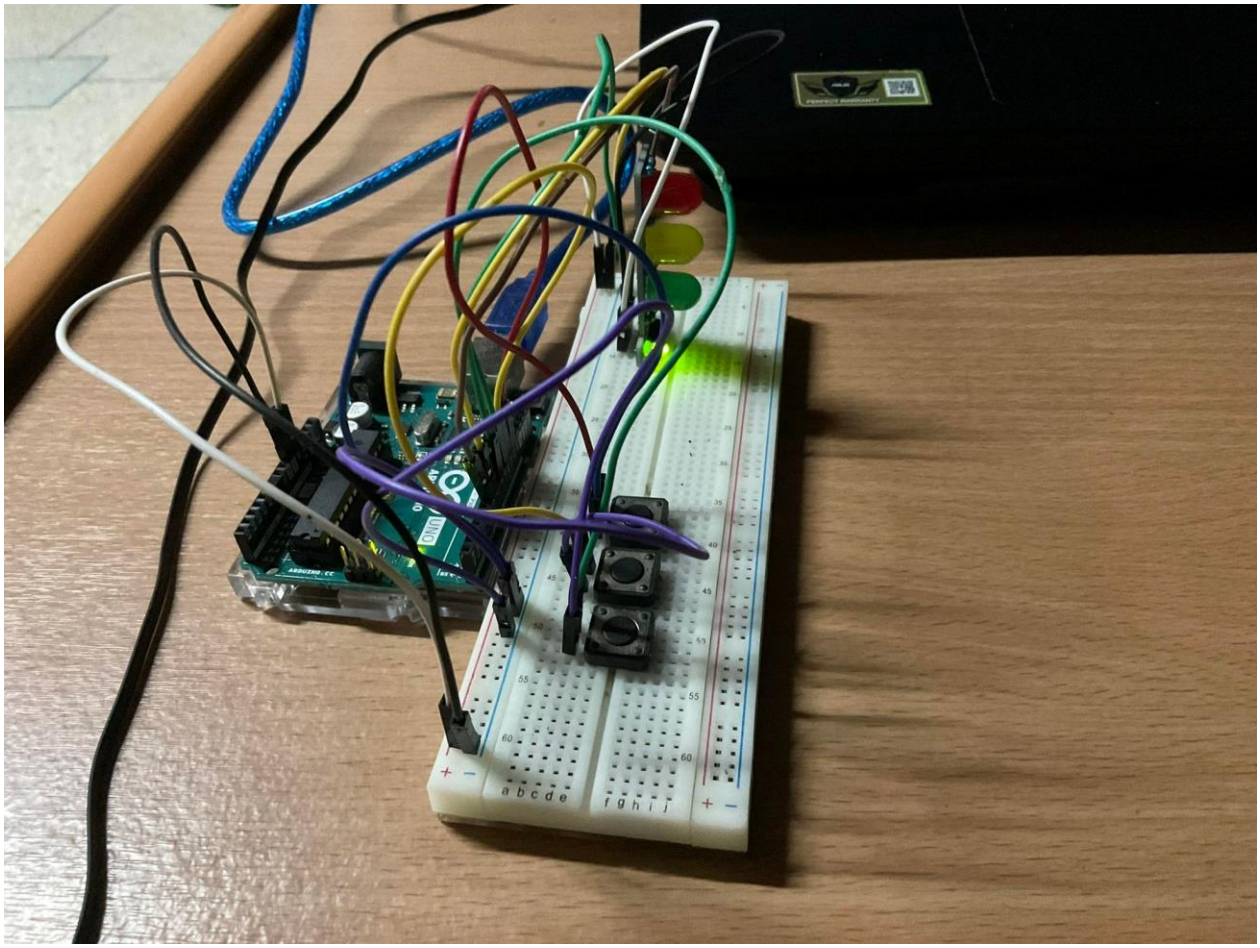
สมาชิกกลุ่ม : 64010761 นายวรพล รังษี

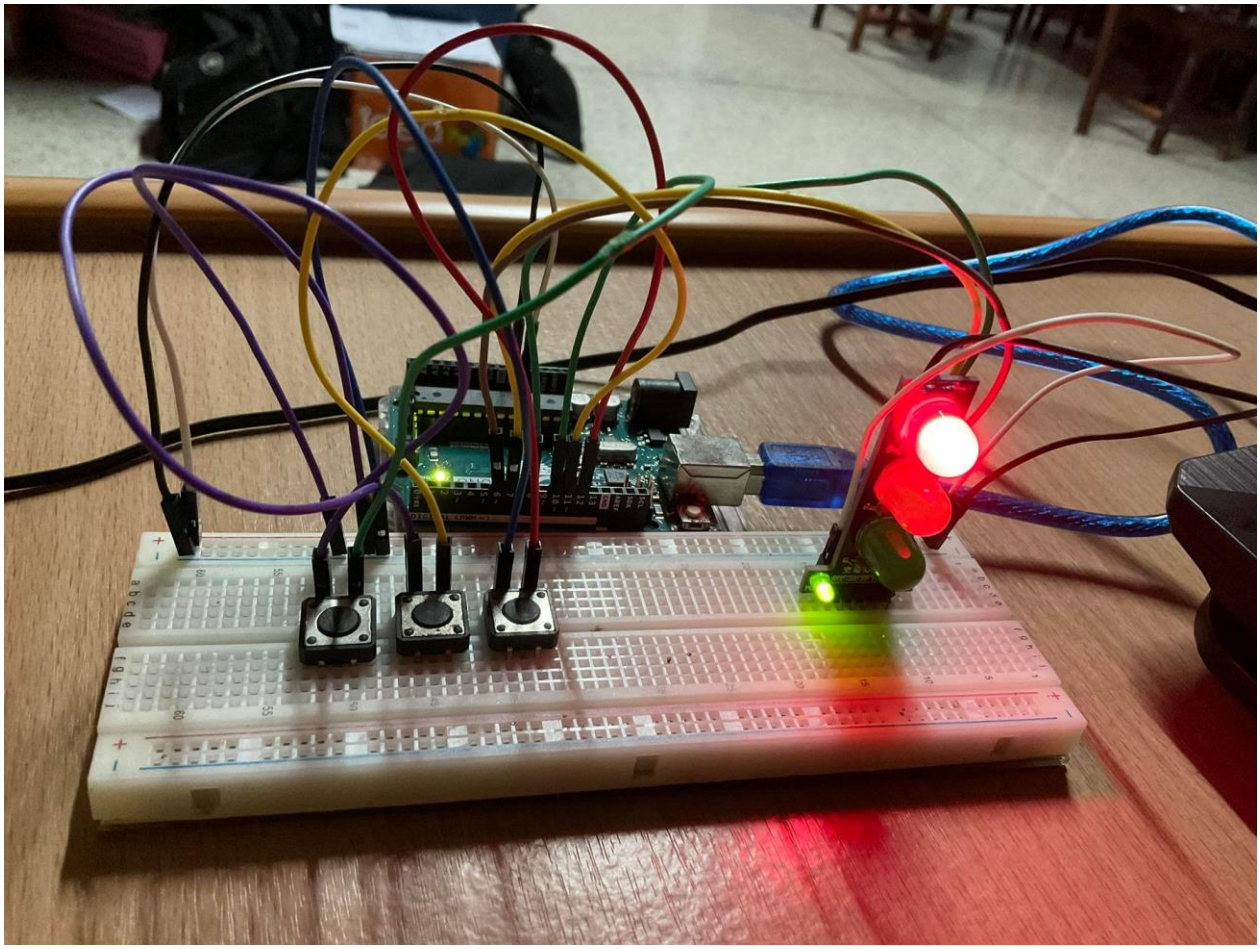
64010757 นายวรโชติ ใจเร็ว

รายละเอียดโปรแกรมโดยย่อ : มีการกำหนด Task ในส่วนของ SetUp ประกอบด้วย SenderTask สำหรับรับค่าสวิตซ์ และส่งค่าที่รับได้ให้ ReceverTask ผ่าน Queue เข้าสู่ function และแสดงผล LED ตามโจทย์

รูปถ่ายชิ้นงาน :







Code : https://github.com/worachote1/itc/blob/main/ITC_Assignment_08.ino

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

#define RED     6 //8
#define YELLOW  7 //9
#define GREEN   8 //10

#define SW1     12 //5  //control red
#define SW2     11 //6  //control yellow
#define SW3     10 //7  //control green

QueueHandle_t BlinkQueue;

void setup()
{
  Serial.begin(9600);
  BlinkQueue =  xQueueCreate(10, sizeof(int32_t));

  xTaskCreate(vSenderTask, "Sender Task 1", 100, SW1, 1, NULL);
  xTaskCreate(vSenderTask, "Sender Task 1", 100, SW2, 1, NULL);
  xTaskCreate(vSenderTask, "Sender Task 1", 100, SW3, 1, NULL);

  xTaskCreate(vReceiverTask, "Receiver Task ", 100, RED, 1, NULL);
  xTaskCreate(vReceiverTask, "Receiver Task", 100, YELLOW, 1, NULL);
  xTaskCreate(vReceiverTask, "Receiver Task", 100, GREEN, 1, NULL);
}

int lastPress = 0;
void vSenderTask(void *pvParameters)
{
  BaseType_t qStatus;
  int32_t valueToSend = 0;
  int SW = (int32_t)pvParameters;
  const TickType_t xTicksToWait = pdMS_TO_TICKS(100);
  pinMode(SW, INPUT_PULLUP);
  while (1)
  {

    if (digitalRead(SW) == LOW && millis() - lastPress >= 50)
    {
      lastPress = millis();
      valueToSend = SW;
    }
    else
    {
      valueToSend = 0;
```

```
    }

    if (valueToSend != 0) //ทำเงื่อนไขเมื่อ valueToSend != 0
    {
      qStatus = xQueueSend(BlinkQueue, &valueToSend, xTicksToWait);
      vTaskDelay(10); //10
    }
  }
}

//for control time with red LED
int count_red = 0;
unsigned long  pastTime_red = 0;

//for control time with yellow LED
unsigned long  pastTime_yellow = 0;

void vReceiverTask(void *pvParameters)
{
  int time_red = 0;
  int LED = (int32_t)pvParameters;
  int32_t valueReceived;
  BaseType_t qStatus;
  const TickType_t xTicksToWait = pdMS_TO_TICKS(100);

  //these LED are active LOW (LED active when LOW)
  pinMode(RED, OUTPUT); digitalWrite(RED, HIGH);
  pinMode(YELLOW, OUTPUT); digitalWrite(YELLOW, HIGH);
  pinMode(GREEN, OUTPUT); digitalWrite(GREEN, HIGH);

  bool YELLOW_status = false;
  while (1)
  {
    qStatus = xQueueReceive(BlinkQueue, &valueReceived, xTicksToWait);
    if (qStatus  == pdPASS)
    {
      Serial.println(valueReceived);
      if (valueReceived == 12 && LED == RED)
      {
        Serial.print("LED = ");
        Serial.print(LED);
        Serial.print("valueReceived = ");
        Serial.println(valueReceived);

        count_red += 1;
        //time_red = count_red * 3000;
        digitalWrite(RED, LOW);
//        vTaskDelay(300);
```

```
//
//        digitalWrite(RED, 1);
      }
      else if (valueReceived == 11 && LED == YELLOW)
      {
        YELLOW_status = !(YELLOW_status);
        digitalWrite(YELLOW,HIGH);
      }
      else if (valueReceived == 10 && LED == GREEN)
      {
        Serial.print("LED = ");
        Serial.println(LED);

        for(int i = 1 ; i <= 3 ; i++)
        {
          digitalWrite(GREEN, 0);
          vTaskDelay(50); //คล้ายๆ delay(500)
          digitalWrite(GREEN, 1);
          vTaskDelay(50);
        }
      }
//      else if (valueReceived != 10)
//      {
//        int32_t valueReceived = valueReceived;
//        xQueueReceive(BlinkQueue, &valueReceived, xTaskCreate);
//        vTaskDelay(1);
//      }

      //test
      Serial.print("qStatus = ");
      Serial.println(qStatus );
    }

    //how to display Yellow LED
    if (YELLOW_status && LED == YELLOW)
    {
        if(millis() - pastTime_yellow >= 400 )
        {
          pastTime_yellow = millis();
          digitalWrite(YELLOW,digitalRead(YELLOW) ^ 1);
        }
    }

    //how to do display Red LED
    if(count_red > 0)
    {
      Serial.print("count = ");
      Serial.println(count_red);
```

```
      if( millis() - pastTime_red >= count_red * 3000)
      {
        pastTime_red = millis();
        digitalWrite(RED,HIGH); // turn off red LED
        count_red = 0;
//        Serial.print("finally , count = ");
//        Serial.println(count_red);
      }
    }


  }

}

void loop()
{


}
```