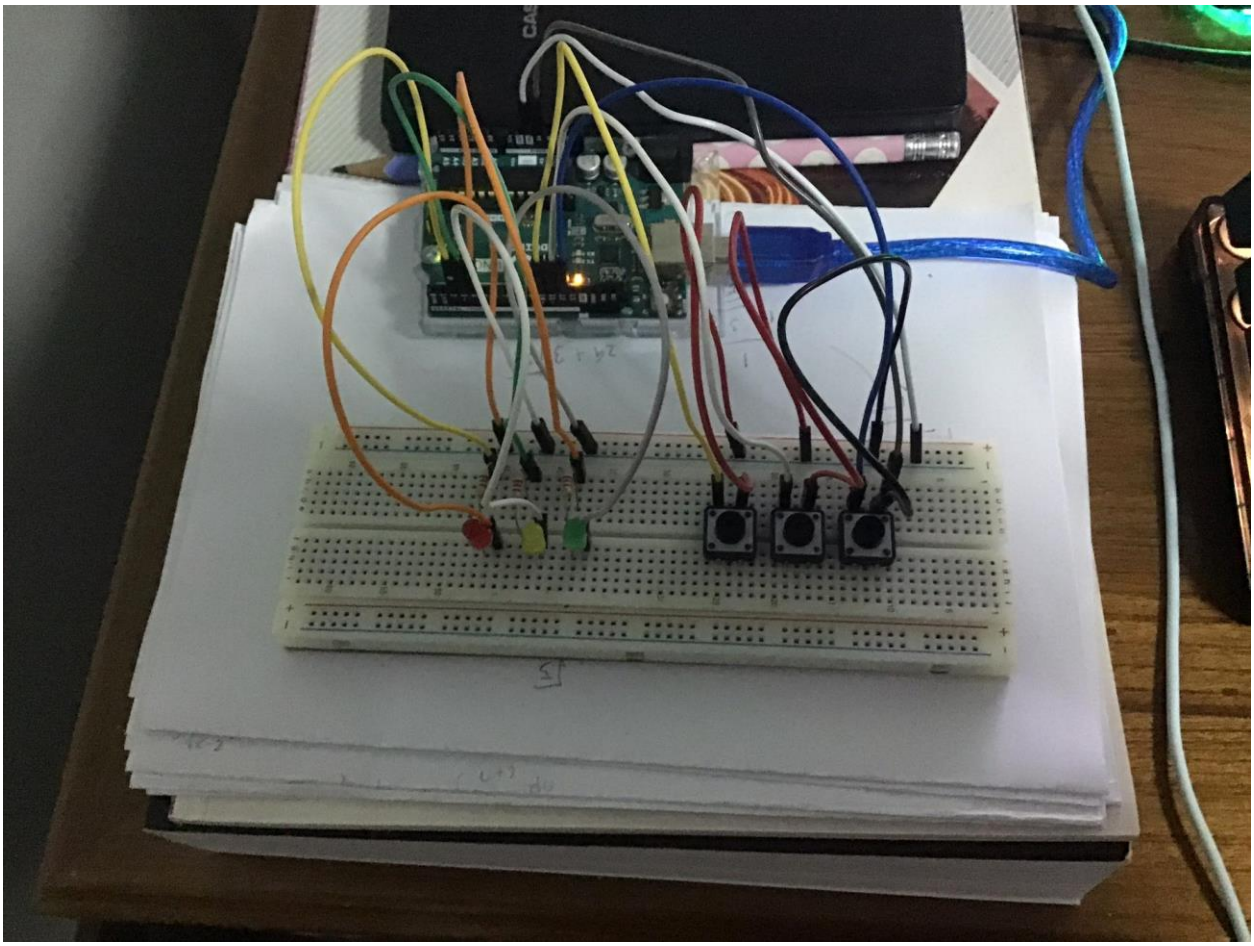
Assignment #9

ชื่อกลุ่ม : ฟ้ารักพ่อ

สมาชิกในกลุ่ม : 64010270 นายคลพัฒน์ โพธิ์น้อย

64010315 นายธนธร แตงอ่อน

รูปถ่ายชิ้นงาน



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

#define red     2
#define yellow  3
#define green   4

#define SW1    10
#define SW2    11
#define SW3    12

// Red LED and Green LED state
int red_state = 0;
int green_state = 0;

// Timer of R, Y and G
unsigned long CurrentTime_R = 0;
unsigned long CurrentTime_Y = 0;
unsigned long CurrentTime_G = 0;

// DebounceTime of R, Y and G
unsigned long Debounce_R = 0;
unsigned long Debounce_Y = 0;
unsigned long Debounce_G = 0;

// Queue of Button
QueueHandle_t ledQueue;

// Token of LED priority
SemaphoreHandle_t token[3];

void setup()
{
```

```
  Serial.begin(9600);
  ledQueue = xQueueCreate(10, sizeof(int32_t));
  xTaskCreate(vSenderTask, "Red SW",     100, SW1,     1,
NULL); // Send
  xTaskCreate(vSenderTask, "Yellow SW",  100, SW2,     1,
NULL);
  xTaskCreate(vSenderTask, "Green SW",   100, SW3,     1,
NULL);

  xTaskCreate(redLED,    "Red LED",    100, NULL, 1,
NULL);
  xTaskCreate(yellowLED, "Yellow LED", 100, NULL, 1,
NULL);
  xTaskCreate(greenLED,  "Green LED",  100, NULL, 1,
NULL);

  // Givetoken of All LED
  for (int i = 0; i < 3; i++)
  {
    token[i] = xSemaphoreCreateBinary();
    xSemaphoreGive(token[i]);
  }
}

// Task of (SW) button
void vSenderTask(void *pvParameters)
{
  BaseType_t qStatus;
  int32_t valueToSend = 0;
  int32_t bt = (int32_t) pvParameters;
  pinMode(bt, INPUT_PULLUP);

  while (1)
  {
    // Check button
```

```
    if (!digitalRead(bt))
    {
      // Send button pin
      valueToSend = bt;

      // Add to queue
      qStatus = xQueueSend(ledQueue, &valueToSend, 0);
      vTaskDelay(1);
    }
  }
}

void redLED(void* pvParameters)
{
  int32_t valueReceived;
  pinMode(red, OUTPUT);
  BaseType_t qStatus;
  const TickType_t xTicksToWait = pdMS_TO_TICKS(100);

  while (1)
  {
    qStatus = xQueueReceive(ledQueue, &valueReceived,
xTicksToWait);
    if (qStatus == pdPASS)
    {
      // Check Red (SW1) Button
      if (valueReceived == SW1  && millis() - Debounce_R
>= 500)
      {
        Debounce_R = millis();

        //Check Red token
        if (xSemaphoreTake(token[0], 1) == pdTRUE)
        {
          CurrentTime_R = millis();
```

```
        red_state = 1;
      }
      else
      {
        // If click Red button (SW1) for second time,
Red = 0 (off)
        red_state = 0;
      }
    }
  }
  redLEDControl(red_state); // Controlling Red LED
 }
}


void yellowLED(void* pvParameters)
{
 int32_t valueReceived;
 pinMode(yellow, OUTPUT);
 BaseType_t qStatus;
 const TickType_t xTicksToWait = pdMS_TO_TICKS(100);

 while (1)
 {
  qStatus = xQueueReceive(ledQueue, &valueReceived,
xTicksToWait);
  if (qStatus == pdPASS)
  {
    // Check Yellow button (SW2)
    if (valueReceived == SW2  && millis() - Debounce_Y
>= 500)
    {
      Debounce_Y = millis();

      // Check Red token
```

```
      if (xSemaphoreTake(token[0], 1) == pdTRUE)
      {
        // Unlock Red token
        xSemaphoreGive(token[0]);

        // Check Green token
        if (xSemaphoreTake(token[2], 1) == pdTRUE)
        {
          // Unlock Greeen token
          xSemaphoreGive(token[2]);

          //  Check Yellow Token
          if (xSemaphoreTake(token[1], 1) == pdTRUE)
          {
            // Turn Yellow LED on
            yellowLEDControl(1);
          }
        }
      }
    }
  }
}

void greenLED(void* pvParameters)
{
  int32_t valueReceived;
  pinMode(green, OUTPUT);
  BaseType_t qStatus;
  const TickType_t xTicksToWait = pdMS_TO_TICKS(100);

  while (1)
  {
    qStatus = xQueueReceive(ledQueue, &valueReceived,
xTicksToWait);
```

```
    if (qStatus == pdPASS)
    {
      // Check Green button (SW3)
      if (valueReceived == SW3  && millis() - Debounce_G
>= 500)
      {
        Debounce_G = millis();

        // Check Red token
        if (xSemaphoreTake(token[0], 1) == pdTRUE)
        {
          // Unlock Red token
          xSemaphoreGive(token[0]);

          // Check Green token if self
          if (xSemaphoreTake(token[2], 1) == pdTRUE)
          {
            CurrentTime_G = millis();
            green_state = 1; //Turn on (1)
          }
          else
          {
            //Turn off(0) if click Green button when
green LED is on (1)
            green_state = 0;
          }
        }
      }
    }
    greenLEDControl(green_state); // Controlling Green LED
  }
}

void loop()
{
```

```
}

void redLEDControl(int state)
{
  // if Red state is on(1)
  if (state)
  {
    digitalWrite(red, HIGH);

    // Check Timer 3 sec
    if (millis() - CurrentTime_R >= 3000)
    {
      digitalWrite(red, LOW);
      red_state = 0;

      // Unlock Red token
      xSemaphoreGive(token[0]);
    }
  }
  else
  {
    // Push button for 2nd time
    digitalWrite(red, LOW);
    xSemaphoreGive(token[0]); //Unlock
  }
}

void yellowLEDControl(int state)
{
  // If Yellow lED on(1)
  if (state)
  {
    // Counter for Yellow Blink
    int cnt_Yellow = 0;
```

```
    while (cnt_Yellow < 4)
    {
      // Check for each Time of Blink
      if (millis() - CurrentTime_Y >= 500)
      {
        // Switching between on and off
       digitalWrite(yellow, digitalRead(yellow) ^ 1);
        cnt_Yellow++;
        CurrentTime_Y = millis();
      }
    }
    //Unlock Yellow token
    xSemaphoreGive(token[1]);
  }
}

void greenLEDControl(int state)
{
  // If Green Led state on (1)
  if (state)
  {
    digitalWrite(green, HIGH);

    // Check time 3 sec
    if (millis() - CurrentTime_G >= 3000)
    {
      digitalWrite(green, LOW);
      green_state = 0;

      //Unlock token
      xSemaphoreGive(token[2]);
    }
  }
  else
  {
```

```
    // Push Button 2nd time
    digitalWrite(green, LOW);

    // Unlock token
    xSemaphoreGive(token[2]);
  }
}
```