

Limit Number In the Room Detection

กลุ่ม กะเพราไก่ไข่ดาว

6331302121	จริยาภรณ์	อัมพรพันธ์
6332009521	ชินพัทธ์	นิธิพรศรี
6332015221	ณิษภัทร	เกริกกิตติกุล
6332020321	ธนวัฒน์	ผลเกิด

2110366

Embedded System Laboratory

ระบบปฏิบัติการและโปรแกรมระบบ

คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

จุฬาลงกรณ์มหาวิทยาลัย

Table of Contents

1. UI Design and Development
2. Front-end Development
3. System Architecture
4. Embedded System Development

UI Design and Development

- 6332020321 ธนวัฒน์ ผลเกิด

หน้าที่รับผิดชอบ:

1. ออกแบบวงจรโดยรวม
2. ออกแบบหน้าจอแสดงผล

1. ออกแบบวงจรโดยรวม

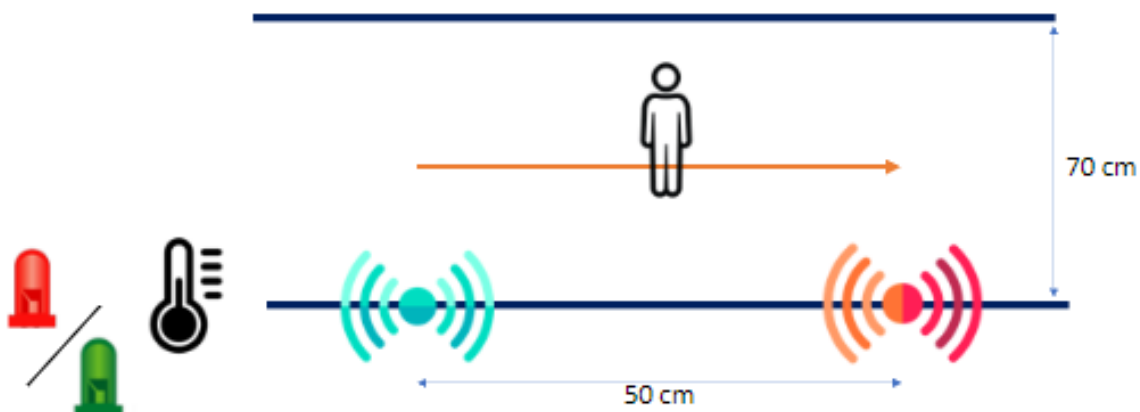
วงจรโดยรวมประกอบด้วย LED สีแดง LED สีเขียว LED สีเหลือง เซนเซอร์ Infrared และ Ultrasonic ตามลำดับจากทางเข้าไปยังห้องที่เราต้องการจะนับจำนวนคน โดยไฟ LED

สีเขียวจะติดเมื่อจำนวนคนในห้องยังไม่ถึงจำนวนคนที่เรากำหนด และไฟ LED

สีแดงจะติดเมื่อจำนวนคนในห้องถึงจำนวนคนที่เรากำหนดไว้แล้ว ไฟ LED สีเหลืองจะติดเมื่อวัตถุคนผ่าน ส่วน Infrared จะเอาไว้วัดอุณหภูมิก่อนเข้าห้องเพื่อเช็คค่าสิ่งที่ผ่าน Ultrasonic ไปเป็นคน ไม่ใช่สิ่งของหรืออื่น ๆ

ถัดมาเป็น Ultrasonic ซึ่งจะใช้สำหรับเช็คคนเข้าออกห้อง โดยเราจะให้ Ultrasonic ตัวที่ 1 และตัวที่ 2 มีระยะห่างกัน 50 เซนติเมตร เนื่องจากเป็นระยะห่างที่มากพอที่จะไม่ทำให้ Ultrasonic ทั้ง 2 ตัวติดพร้อมกัน และให้ความกว้างของทางเดินมีขนาด 70 เซนติเมตร เพื่อที่จะให้คนไม่สามารถเดินเข้าและออกพร้อมกันได้

โดยเราจะนับว่ามีคนเข้าห้องเมื่อ Infrared Ultrasonic ตัวที่ 1 และ Ultrasonic ตัวที่ 2 ทำงานตามลำดับ และเราจะนับว่ามีคนออกจากห้องเมื่อ Ultrasonic ตัวที่ 2 และ Ultrasonic ตัวที่ 1 ทำงานตามลำดับ



2. ออกแบบหน้าจอแสดงผล

หน้าจอแสดงผลจะประกอบไปด้วยชื่อโปรเจกต์ และจำนวนคนที่อยู่ในห้องแบบ Real time และเมื่อมีคนอยู่ในห้องถึงจำนวนที่กำหนดแล้วหน้าจอแสดงผลจะเปลี่ยนไปเป็นข้อความเตือน

Front-end Development

- 6331302121 จริยาภรณ์ อัมพรพันธ์

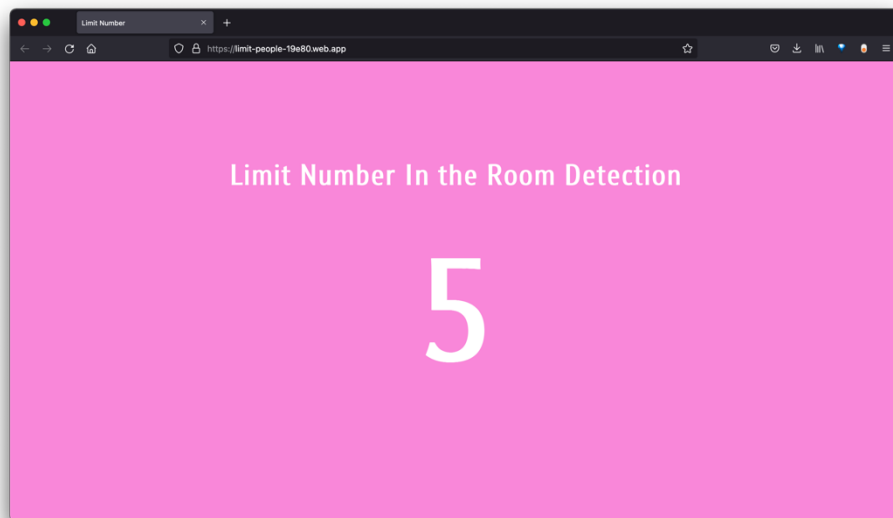
หน้าที่รับผิดชอบ:

1. ทำหน้าเว็บเพื่อแสดงจำนวนคนในห้อง ณ เวลาจริง
2. ดึงข้อมูลจำนวนคนจาก Firebase Realtime Database มาแสดงบนหน้าเว็บ

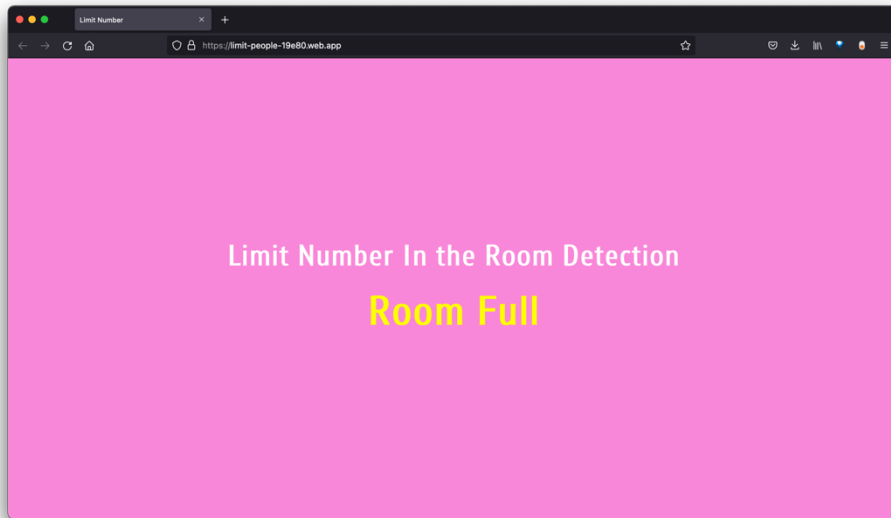
1. ทำหน้าเว็บเพื่อแสดงจำนวนคนในห้อง ณ ปัจจุบัน

หน้าเว็บนี้มีการใช้ Hosting service ของ Firebase

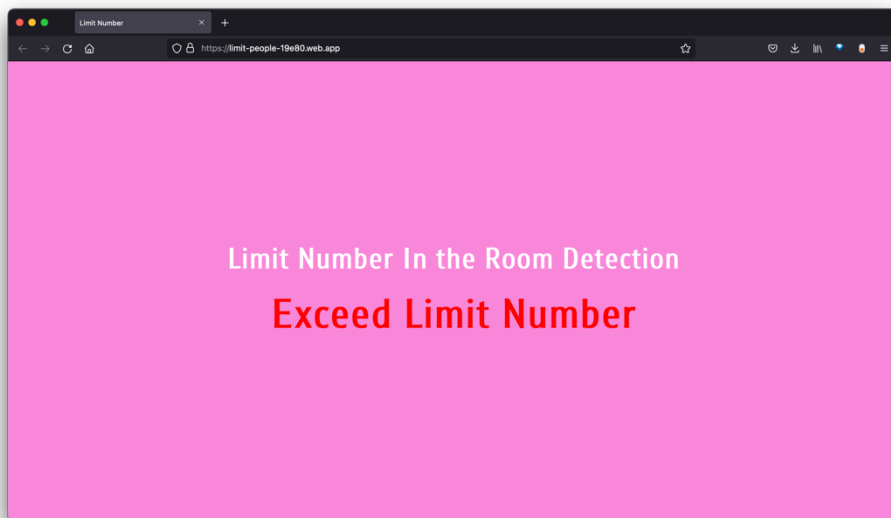
โดยองค์ประกอบของหน้าเว็บประกอบไปด้วยข้อความ 2 ส่วน คือ ชื่อโปรเจกต์ และ จำนวนคนในห้อง ณ เวลาจริง ดังภาพ



และจะแสดงข้อความ “Room Full” แทนตัวเลขเมื่อมีคนในห้องเท่ากับจำนวนที่ตั้งไว้ ดังภาพ



และจะแสดงข้อความ “Exceed Limit Number” แทนข้อความ “Room Full”
เมื่อมีคนในห้องเกินจำนวนที่ตั้งไว้ ดังภาพ



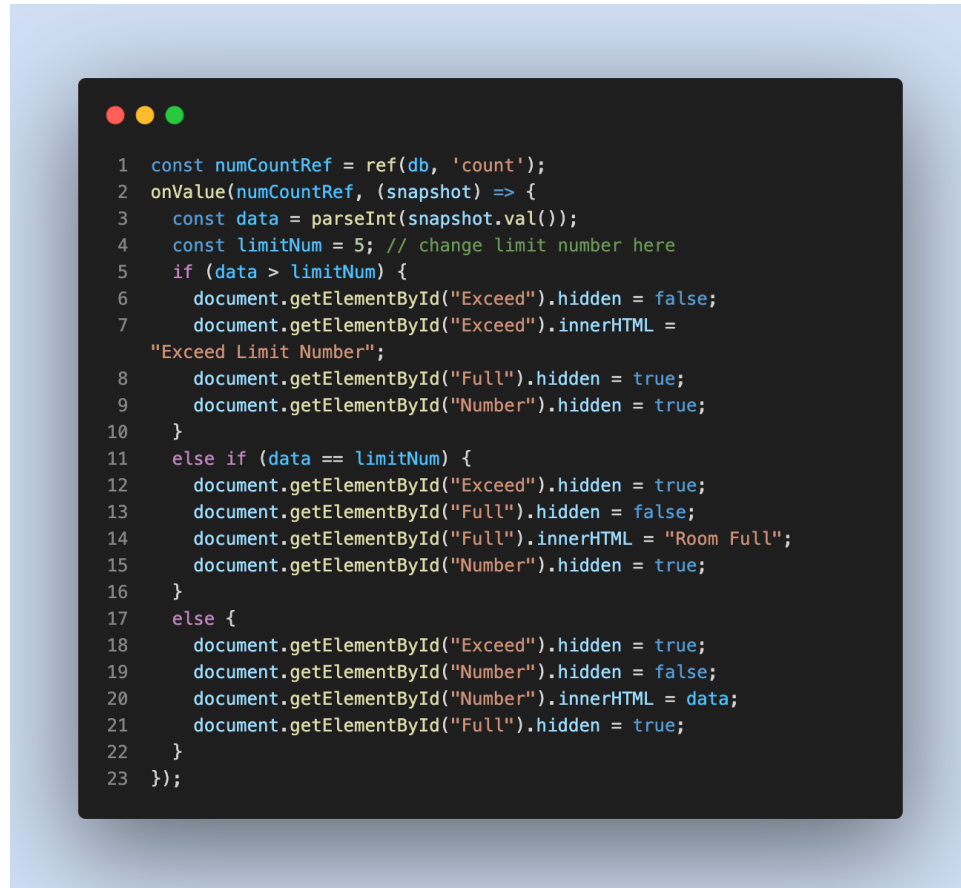
ในส่วนของการสร้าง มีการใช้ภาษาคอมพิวเตอร์ 3 ภาษา ได้แก่

1. HTML ใช้ในการสร้างโครงสร้างข้อมูลที่จะแสดงในเว็บ
2. CSS ใช้ในการตกแต่งหน้าเว็บ โดยมีการใช้ SCSS ซึ่งเป็น CSS Preprocessor มาดำเนินการ
3. Javascript ใช้ในการดึงข้อมูลจำนวนคนจาก Firebase มาแสดงผลในหน้าเว็บ

2. ดึงข้อมูลจำนวนคนจาก Firebase Realtime Database มาแสดงบนหน้าเว็บ

ในโปรเจกต์นี้มีการใช้บริการ Realtime Database ของ Firebase เพื่อเก็บและอัปเดตข้อมูลจำนวนคนในห้อง ณ เวลาจริง ซึ่งจะมีการอธิบายภายหลัง

ในส่วนของหน้าเว็บจะมีการดึงข้อมูลส่วนนี้จาก Firebase มาและนำมาแสดงผลบนหน้าเว็บโดยใช้ Javascript มาจัดการโดยใช้คำสั่ง ดังภาพ



โดยฟังก์ชัน [onValue\(\)](#) เป็นฟังก์ชันที่มีไว้เพื่ออ่านค่าใน database โดยจะถูกเรียกทุกครั้งข้อมูลใน database มีการเปลี่ยนแปลง ซึ่งในฟังก์ชันนี้จะมีการเรียก snapshot เพื่อดึงข้อมูลจำนวนคนออกมา จากนั้นจะนำข้อมูลนั้นไปแสดงบนหน้าเว็บโดยใช้คำสั่ง innerHTML ให้แสดงผล ณ ตำแหน่งที่ได้กำหนด id เท่ากับ “Number” ไว้ในไฟล์ HTML หากจำนวนคนน้อยกว่าค่าที่กำหนด ถ้าหากจำนวนคนเท่ากับค่าที่กำหนดจะแสดงข้อความ “Room Full” ณ ตำแหน่งที่ได้กำหนดค่า id เท่ากับ “Full” ในไฟล์ HTML แทน แต่หากเกิน จะแสดงข้อความ “Exceed Limit Number” ณ ตำแหน่งที่ได้กำหนดค่า id เท่ากับ “Exceed” ในไฟล์ HTML แทน ดังภาพ



```
1 <div class="container">
2   <h1>Limit Number In the Room Detection</h1>
3   <h1 class ="num" id="Number"></h1>
4   <h1 class ="exceed" id="Exceed"></h1>
5   <h1 class ="full" id="Full"></h1>
6 </div>
```


System Architecture

- 6332009521 ชินพัทธ์ นิธิพรศรี

หน้าที่รับผิดชอบ:

1. ส่งข้อมูลระหว่าง NodeMCU กับ STM32
2. ส่งข้อมูลจากเซนเซอร์ Ultrasonic มายัง STM32 และคำนวณระยะทางของวัตถุ
3. คำนวณจำนวนคนในห้อง

1. ส่งข้อมูลระหว่าง NodeMCU กับ STM32

ใช้ UART ในการส่งข้อมูลระหว่าง NodeMCU กับ STM32 โดยการใช้ SoftwareSerial เพื่อกำหนด พอร์ต D6 และ D7 เป็น RX และ TX ของ NodeMCU ตามลำดับ

```
const byte RX = D6;  
const byte TX = D7;  
SoftwareSerial mySerial = SoftwareSerial(RX, TX);
```

และเลือกใช้ USART1 จะได้พอร์ต PA10 และ PA15 เป็น RX และ TX ของ STM32 ตามลำดับ

Pin Name	Signal on Pin
PA10	USART1_RX
PA15	USART1_TX

ทำการเชื่อมต่อพอร์ต D6 กับ PA15 และพอร์ต D7 กับ PA10 เมื่อเชื่อมต่อ UART แล้วจึงทำการส่งข้อมูลของตัวแปร human ใน NodeMCU ซึ่งจะมีค่า 1 เมื่อตรวจวัดอุณหภูมิของมนุษย์ได้ และ มีค่า 0 เมื่อตรวจจับมนุษย์ไม่ได้ ให้ STM32 เพื่อทำการคำนวณจำนวนคนที่อยู่ในห้อง จากนั้น STM32 จะส่งจำนวนคนที่คำนวณได้ (ค่าของตัวแปร HUMAN[0] ใน STM32) กลับมาให้ NodeMCU เพื่อส่งค่าของจำนวนคนในห้องให้ Firebase เพื่อนำไปแสดงบนหน้าเว็บต่อไป

2. ส่งข้อมูลจากเซนเซอร์ Ultrasonic มายัง STM32 และคำนวณระยะทางของวัตถุ

ต่อ Trig Pin ของ Ultrasonic1 และ Ultrasonic2 กับ TIM1_CH1 และ TIM2_CH2 เพื่อใช้ Timer1 Channel1 และ Timer2 Channel2 ในการรับค่าของ Ultrasonic1 และ Ultrasonic2 ตามลำดับ

```
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    if(htim->Channel == HAL_TIM_ACTIVE_CHANNEL_1){ // if the interrupt source is channel1
        Trig_Ultrasonic_1(htim);
    }
    if(htim->Channel == HAL_TIM_ACTIVE_CHANNEL_2){ // if the interrupt source is channel2
        Trig_Ultrasonic_2(htim);
    }
}
```

จากนั้นทำการรับค่าของ Ultrasonic ทั้งสองมาเก็บไว้ใน IC_Val1, IC_Val2 สำหรับ Ultrasonic1 และ IC_Val12, IC_Val22 สำหรับ Ultrasonic2 โดยใช้ฟังก์ชัน Trig_Ultrasonic_1 และ Trig_Ultrasonic_2 ตามลำดับ

```
void Trig_Ultrasonic_1(TIM_HandleTypeDef *htim){
    if (Is_First_Captured==0) // if the first value is not captured
    {
        IC_Val1 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1); // read the first value
        Is_First_Captured = 1; // set the first captured as true
        // Now change the polarity to falling edge
        __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_FALLING);
    }
    else if (Is_First_Captured==1) // if the first is already captured
    {
        IC_Val2 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_1); // read second value
        __HAL_TIM_SET_COUNTER(htim, 0); // reset the counter

        Distance = cal_dis(IC_Val1, IC_Val2);
        Is_First_Captured = 0; // set it back to false

        // set polarity to rising edge
        __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_1, TIM_INPUTCHANNELPOLARITY_RISING);
        __HAL_TIM_DISABLE_IT(&htim1, TIM_IT_CC1);
    }
}

void Trig_Ultrasonic_2(TIM_HandleTypeDef *htim){
    if (Is_First_Captured2==0) // if the first value is not captured
    {
        IC_Val12 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_2); // read the first value
        Is_First_Captured2 = 1; // set the first captured as true
        // Now change the polarity to falling edge
        __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_2, TIM_INPUTCHANNELPOLARITY_FALLING);
    }

    else if (Is_First_Captured2==1) // if the first is already captured
    {
        IC_Val22 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_2); // read second value
        __HAL_TIM_SET_COUNTER(htim, 0); // reset the counter

        Distance2 = cal_dis(IC_Val12, IC_Val22);
        Is_First_Captured2 = 0; // set it back to false

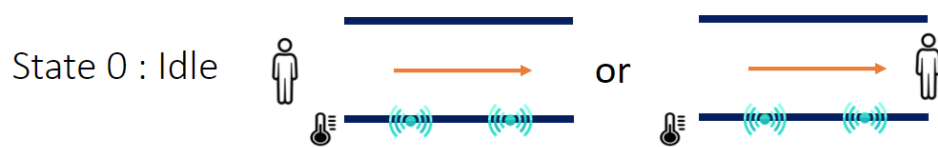
        // set polarity to rising edge
        __HAL_TIM_SET_CAPTUREPOLARITY(htim, TIM_CHANNEL_2, TIM_INPUTCHANNELPOLARITY_RISING);
        __HAL_TIM_DISABLE_IT(&htim2, TIM_IT_CC2);
    }
}
```

เมื่อทำการรับค่าแล้วจึงนำค่าทั้งสองไปคำนวณโดยใช้ฟังก์ชัน cal_dis ซึ่งจะคืนระยะห่างของ Ultrasonic กับวัตถุกลับมา โดยค่าระยะห่างของ Ultrasonic1 และ Ultrasonic2 ใน Distance และ Distance2 ตามลำดับ

```
uint8_t cal_dis(uint32_t val1, uint32_t val2){  
    if (val2 > val1){  
        Difference = val2-val1;  
    }else if (val1 > val2){  
        Difference = (0xffff - val1) + val2;  
    }  
    return Difference * .034/2;  
}
```

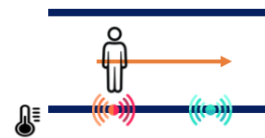
3. คำนวณจำนวนคนในห้อง

ทำการพิจารณาสถานการณ์ของการเดินเข้า-ออกแยกออกมาได้ 4 states ได้แก่



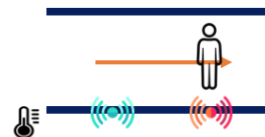
State 0 คือช่วงจังหวะที่คนยังไม่เดินเข้า หรือเดินผ่าน(เข้า/ออก)ไปแล้ว

State 1 : First sensor detect object



State 1 คือช่วงจังหวะที่ Ultrasonic1 ตรวจจับได้ว่ามีวัตถุผ่าน

State 2 : Second sensor detect object



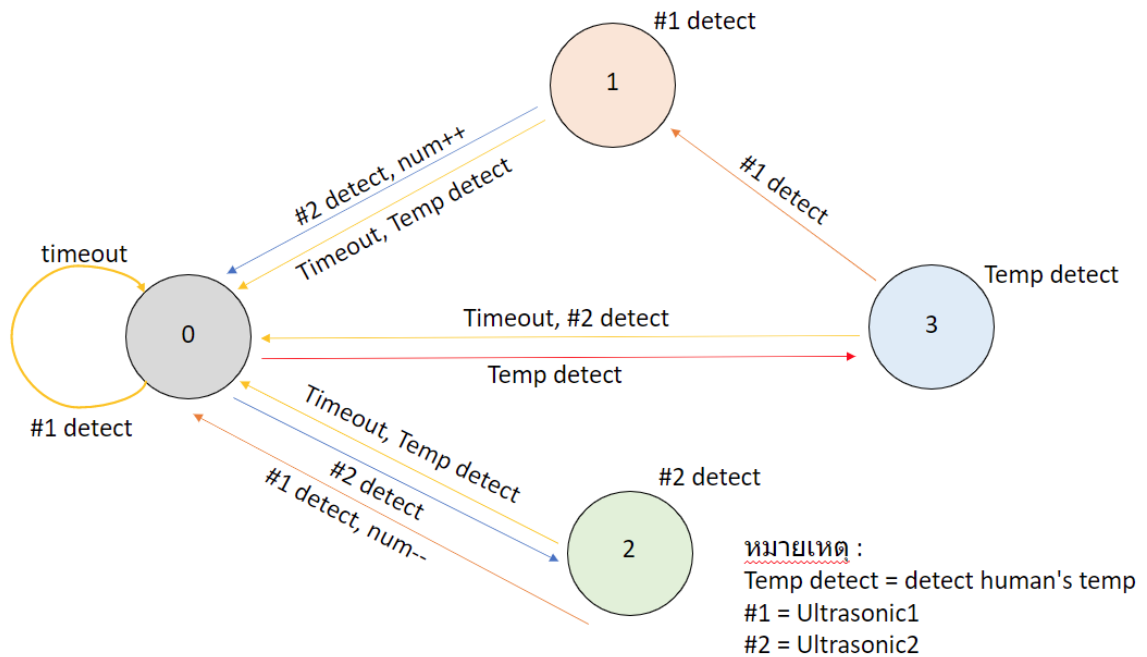
State 2 คือช่วงจังหวะที่ Ultrasonic2 ตรวจจับได้ว่ามีวัตถุผ่าน

State 3 : Temp detect



State 3 คือช่วงจังหวะที่ Temperature Sensor ตรวจจับอุณหภูมิของมนุษย์ได้

ซึ่งเมื่อทำการวาด State Diagram ออกมาแล้วจะได้ ดังภาพ



จากนั้นจึงนำ Diagram ดังกล่าวมาแปลงเป็น code จะได้ว่า

หาก Ultrasonic1 ตรวจจบบัวตฤได้จะเปลี่ยน state ดังนี้

```

if(Distance < object_dis){ // #1 detects
  counter_tim3 = 0;
  if(state == 3){
    state = 1;
    HAL_Delay(600);
  } else if(state == 2){
    if(num > 0) num--;
    state = 0;
    HAL_Delay(600);
  }
}

```

หาก Ultrasonic2 ตรวจจบบัวตฤได้จะเปลี่ยน state ดังนี้

```

}else if(Distance2 < object_dis){ // #2 detects
    counter_tim3 = 0;
    if(state == 0){
        state = 2;
        HAL_Delay(600);
    } else if(state == 1){
        num++;
        state = 0;
        HAL_Delay(600);
    } else if(state == 3){
        state = 0;
        HAL_Delay(600);
    }
}

```

หาก Temperature Sensor ตรวจจับอุณหภูมิของมนุษย์ได้จะเปลี่ยน state ดังนี้

```

HAL_UART_Receive_IT(&huart1, (uint8_t*) HUMAN, 1);
if(HUMAN[0] == '1'){
    counter_tim3 = 0;
    if(state == 0 || state == 3) state = 3;
    else if(state == 1 || state == 2) state = 0;
}

```

หากเกินเวลาที่กำหนดไว้ (ใช้ Timer 3 Channel 3 ในการนับเวลา) จะเปลี่ยน state ดังนี้

```

}else if(counter_tim3 > 100){
    counter_tim3 = 0;
    state = 0;
}

```

ซึ่งเมื่อเพิ่ม/ลดค่าของจำนวนคนแล้วก็จะส่งค่านี้ไปให้ NodeMCU ผ่าน USART1

โดยหากค่าของจำนวนคนเท่ากับหรือมากกว่าค่าที่กำหนดไว้ ไฟ LED สีเขียวจะดับ และไฟ LED สีแดงจะติด

แต่หากค่าของจำนวนคนยังไม่เกินค่าที่กำหนด ไฟ LED สีเขียวจะติด และไฟ LED สีแดงจะดับ

และหาก Temperature sensor ตรวจจับอุณหภูมิของมนุษย์ได้ ไฟ LED สีเหลืองจะติด

Embedded System Development

- 6332015221 ณิชภัทร เกริกกิตติกุล

หน้าที่รับผิดชอบ:

0. การเตรียม NodeMCU(ESP8266)
1. การส่งข้อมูลจากเซ็นเซอร์วัดอุณหภูมิไปยัง NodeMCU
2. การส่งข้อมูลระหว่าง NodeMCU และ Firebase

0. การเตรียม NodeMCU(ESP8266)



NodeMCU

```
1 void setup() {  
2     connectWifi();  
3     Serial.begin(115200);  
4     Firebase.begin(FIREBASE_HOST, FIREBASE_KEY);  
5     mlx.begin();  
6 }
```

ก่อนเริ่มการใช้งานนั้นจะทำการเชื่อมต่อ WiFi, setup temperature sensor และ config key ของ firebase ไว้

1. การส่งข้อมูลจากเซ็นเซอร์วัดอุณหภูมิไปยัง NodeMCU



Temp Sensor

จาก [Adafruit MLX90614 Library](#) ทำให้เราสามารถอ่านข้อมูลจาก sensor ได้จาก #include <Adafruit_MLX90614>

```
1 void callTemp(){
2   Serial.print("Ambient = ");
3   Serial.print(mlx.readAmbientTempC());
4   Serial.print("C\tObject = ");
5   Serial.print(mlx.readObjectTempC()); Serial.println("C");
6   if(mlx.readObjectTempC() > 30){
7     Serial.println("HUMAN DETECT");
8     human = 1;
9   } else {
10    Serial.println("NOTHING");
11    human = 0;
12  }
13  Serial.println();
14  delay(500);
15 }
```

โดยเราจะวัดอุณหภูมิอยู่สองอย่าง คือ Ambient และ Object ซึ่งจะตัดสินว่า Object นี้เป็นมนุษย์ ถ้า Object temperature นี้มากกว่า 30 องศา และทำงานนี้เรื่อยๆตลอดเวลาเพื่อคอยตรวจสอบการวัดอุณหภูมิ

2. การส่งข้อมูลระหว่าง NodeMCU และ Firebase

2.1 เชื่อมต่อ WiFi

```
1  #include <ESP8266WiFi.h>
2
3  void connectWifi() {
4      WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
5      while (WiFi.status() != WL_CONNECTED) {
6          Serial.print(".");
7          delay(500);
8      }
9      Serial.print("Connected Already!");
10 }
```

จาก [ESP8266WiFi Core Library](#) ทำให้การเชื่อมต่อ WiFi สามารถทำได้ตาม code ดังภาพ

2.2 ส่งข้อมูลไปที่ Cloud (Firebase Realtime Database)

```
1  #include <FirebaseESP8266.h>
2
3  #define FIREBASE_HOST "FIREBASE_HOST"
4  #define FIREBASE_KEY "FIREBASE_KEY"
5
6  void postValueTofirebase(String val){
7      if(Firebase.set(firebaseData, "/count", val)) {
8          Serial.println("Added");
9      } else {
10         Serial.println("Error : " + firebaseData.errorReason());
11     }
12 }
```


เราทำการเขียน function ชื่อ `postValueTofirebase(String val)` ที่จะทำการส่งค่า `String val` ไปที่ Realtime Database ในรูปแบบ `{ count : val }` (NoSQL)