

# Smart Light



Final Project, Fall 2018: Internet of Things (IoT)

Present to

Dr.Pitchaya Sitthi-amorn

Presented by

Gun Kaewngarm	5931006121
Nattichai Sutipanwihan	5931023821
Kamolnadda Dansuputra	5931001021
Ittiwat Sukhumdhanakul	5931077721

Embedded System Laboratory 2110366

Semester 1/2018

Department of Computer Engineering,  
Faculty of Engineering, Chulalongkorn University

## **Table of content**

<b>1.Introduction</b>	<b>3</b>
<b>2.Objective</b>	<b>3</b>
<b>3. Roles and Responsibility</b>	<b>4</b>
<b>3.1 System Architecture</b>	<b>4</b>
<b>3.2 UI designer and development</b>	<b>6</b>
<b>3.3 Front-end development</b>	<b>9</b>
<b>3.4 Embedded System Development</b>	<b>11</b>
<b>4.Problem</b>	<b>13</b>

## 1.Introduction

เนื่องจากปัจจุบันความต้องการทางตลาดในด้าน smart home ได้เพิ่มมากขึ้น และการใช้งานการเปิด/ปิดไฟนั้นสามารถใช้การสั่งงานผ่านอินเทอร์เน็ต และผ่านคำสั่งเสียงเพื่ออำนวยความสะดวกแก่ผู้ใช้งานได้ ดังนั้นทางทีมงานของเราจึงได้ริเริ่มโปรเจกต์ชิ้นนี้พร้อมทั้งออกแบบ prototype เพื่อทำให้การใช้งานสะดวกสบายมากขึ้น

## 2.Objective

ในปัจจุบันนั้น การที่จะเปิด/ปิดไฟยังต้องทำผ่านการเดินไปที่สวิตช์ไฟด้วยตนเอง ในบางครั้งจึงเป็นโง่งที่เสียเวลาและไม่สะดวกอย่างมาก หากสามารถสั่งการผ่านอินเทอร์เน็ตหรือผ่านเสียงได้ย่อมที่จะแก้ปัญหาความไม่สะดวกสบายนี้ได้

ทางผู้จัดทำจึงออกแบบฮาร์ดแวร์เพื่อแก้ปัญหาดังกล่าวด้วยการใช้เซนเซอร์เสียง(ไมค์โครโฟน)เพื่อทำการเปิดหรือปิดไฟ ผ่านตรวจจับเสียงคบบมือติดต่อกันสองครั้ง และสามารถสั่งให้เปิดหรือปิดไฟได้ผ่าน ทางเว็บแอปพลิเคชัน

### 3. Roles and Responsibility

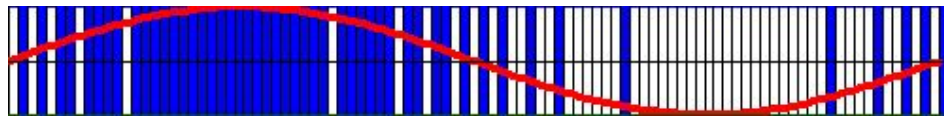
### 3.1 System Architecture

Person in charge: Gun Kaewngarm

### Responsibility:

- ศึกษาmicrophone MP45DT02 Data sheet :  
<https://www.marutsu.co.jp/contents/shop/marutsu/ds/MP45DT02.pdf>
- ศึกษาข้อมูลวิธีการใช้จากวิดีโอและsource code ตัวอย่าง:  
<https://www.youtube.com/watch?v=YBh65gcw0w8>  
<https://github.com/gizTM/STM32F4Discovery-IC-and-application>
- ศึกษาประเภทของข้อมูลที่MP45รับเสียงมาแล้วส่งให้บอร์ด ซึ่งMP45DT02ส่งข้อมูลเสียงเป็นแบบPDM (Pulse-density modulation)<sup>1</sup>

```
0101011011110111111111111111111101111101101101010100100100000010000000
0000000000000001000010010101
```



An example of PDM of 100 samples of one period of a sine wave. 1s represented by blue, 0s represented by white, overlaid with the sine wave.

การเชื่อมโยงกันของระบบย่อยต่างๆ เช่น กลุ่มSensor กลุ่มส่งข้อมูล ได้ภาพรวมของระบบดังนี้

## การต่อวงจร

## 1. តំបន់ MP45DT02 ក្នុង STM32

ในที่นี้MP45DT02นั้นอยู่บนboardอยู่แล้วจึงไม่จำเป็นต้องต่อ

<sup>1</sup> "Pulse-density modulation - Wikipedia." [https://en.wikipedia.org/wiki/Pulse-density\\_modulation](https://en.wikipedia.org/wiki/Pulse-density_modulation). Accessed 18 Dec. 2018.



ภาพไมโครโฟนMP45DT02(วงกลมสีแดง)บนบอร์ด STM32F4Discovery

## 2.ต่อ STM32 กับ NodeMCU

ต่อ PA2,PA3 ซึ่งเป็น USART2\_TX,USART2\_RX ของSTM32 กับ D8,D7 ซึ่งเป็น TX,RX ของ NodeMCUตามลำดับ โดยจะส่งข้อมูลแบบSerial Data

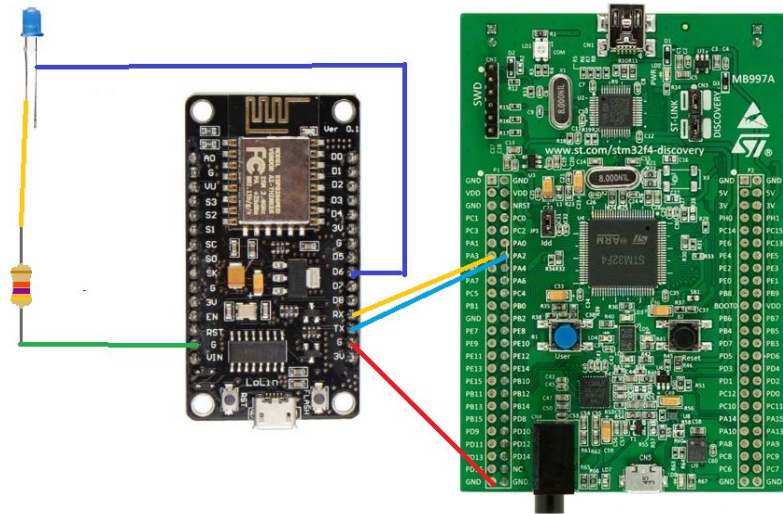
## 3.ต่อ NodeMCUกับ LED และตัวต้านทานเพื่อจำลองไฟที่จะถูกปิด/เปิด

ต่อ D6 ของ NodeMCUกับขั้วบวกของLEDเพื่อจ่ายไฟเข้าตามคำสั่ง และต่อขั้วลบเข้ากับตัวต้านทาน และGND ของ NodeMCU

## 4. เชื่อมต่อ NodeMCU กับ NETPIE ผ่านlibrary microgear ของ NETPIE

โดยมีwifiจะส่งข้อมูลไปยังNETPIE โดยใช้ฟังก์ชัน microgear.chat เมื่อNETPIEได้รับข้อมูลแล้ว Websiteจะ ดึงข้อมูลจากNETPIEมาใช้ (NETPIE คือ Platform as a service บริการเชื่อมต่อข้อมูลและ แลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ต่างๆได้ เหมาะสำหรับ IoT เช่น Arduino, ESP8266 และสามารถหา Library ของอุปกรณ์ต่างๆได้ที่ GitHub NETPIE ให้บริการ โดย NECTEC)

5. ถ้ามีการกดปุ่มสั่งจากWebsite ก็จะส่งข้อมูลติดต่อกับNETPIEผ่านlibrary microgear แล้วส่งกลับไปยังNodeMCU โดยในที่นี้จะใช้ไฟแสดงแทนว่าได้รับคำสั่งจากWebsite



ภาพรวมการต่อวงจรทั้งหมด

### 3.2 UI designer and development

Person in charge : Kamolnadda Dansuputra

Responsibility :

- ออกแบบหน้าจอผู้ใช้ วางแผนด้านการจัดวางองค์ประกอบของ website เพื่อให้ Frontend Developer จะได้นำไป implement ต่อไป

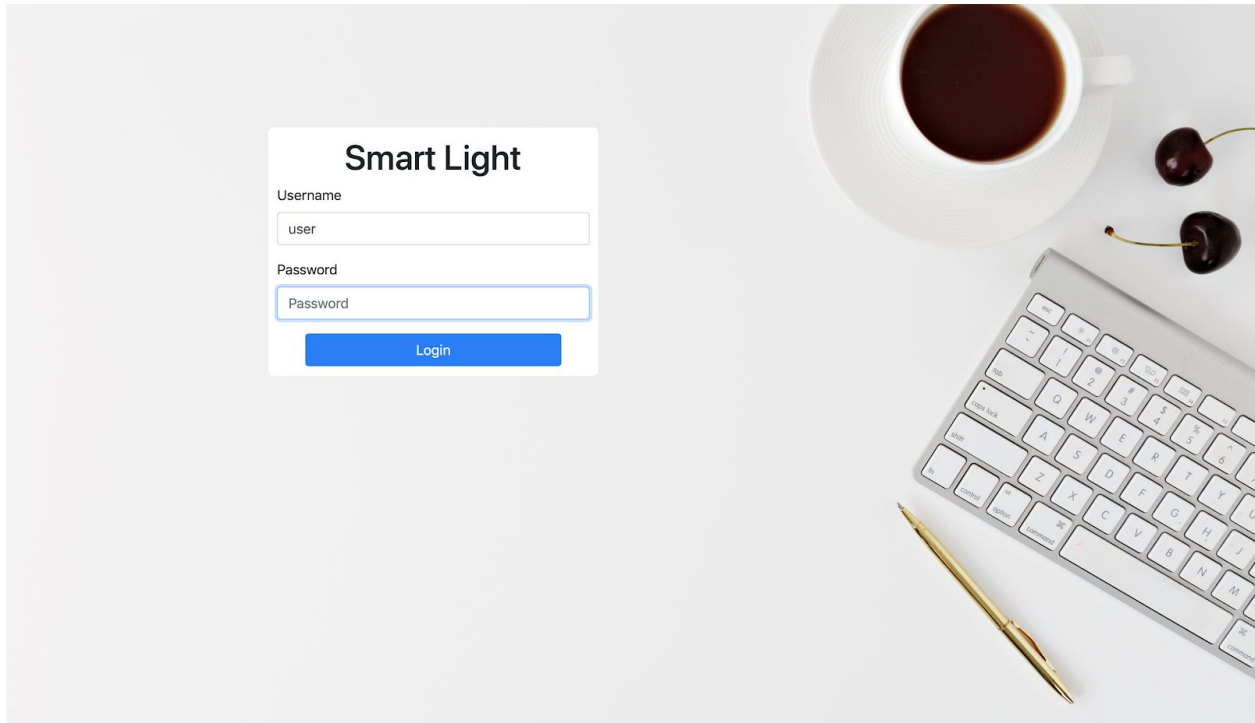
UX/UI ของโปรเจกนี้ เน้นในด้านการใช้งานและสะดวกกับผู้ใช้เป็นหลัก โดยเราได้แบ่ง หน้าจอออกเป็นสองส่วน นั่นคือ ส่วน login และ ส่วนฟังก์ชันเปิดปิดหลอดไฟ

โดยใช้หลักการการออกแบบดังนี้

1. เน้นความใช้ง่ายของผู้ใช้
2. เน้นการจัดวางที่ดูสะอาด ไม่รก

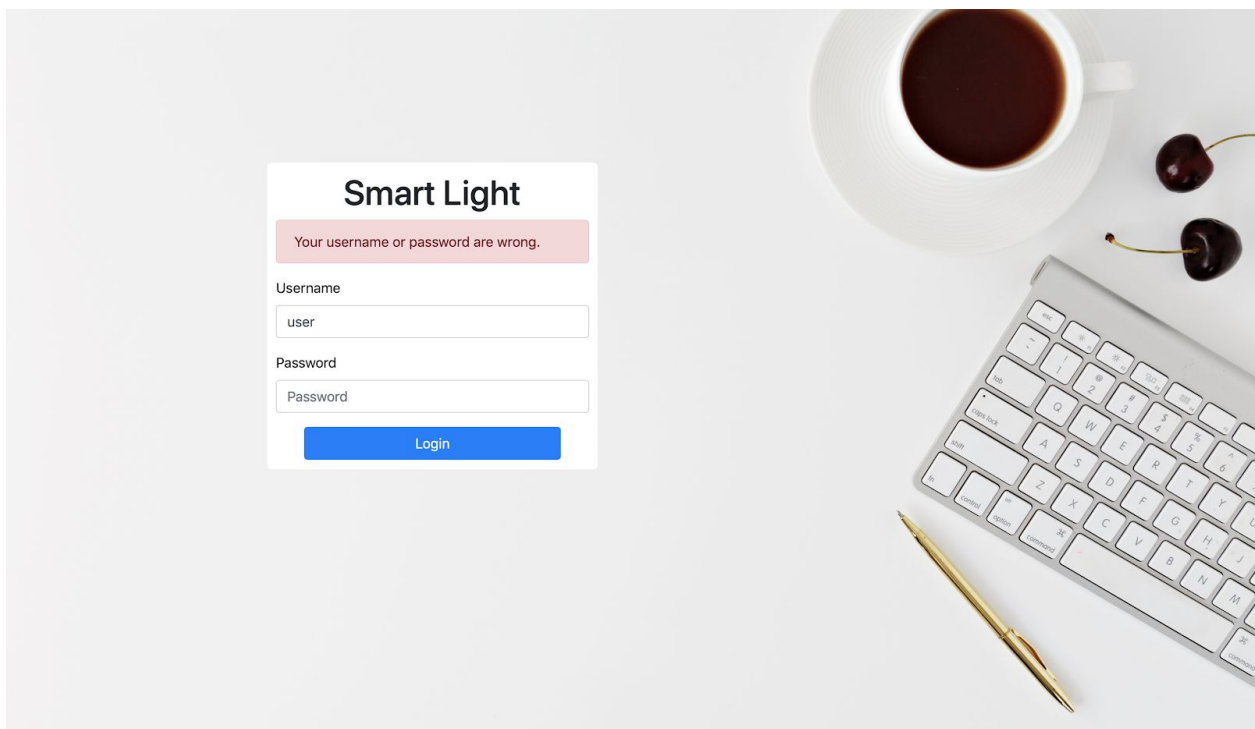
แต่ละหน้ามีองค์ประกอบดังต่อไปนี้

1. หน้าแรก จะเป็นหน้า log in user ต้องกรอก username password ที่ถูกต้องลงในฟอร์ม แล้วกด login



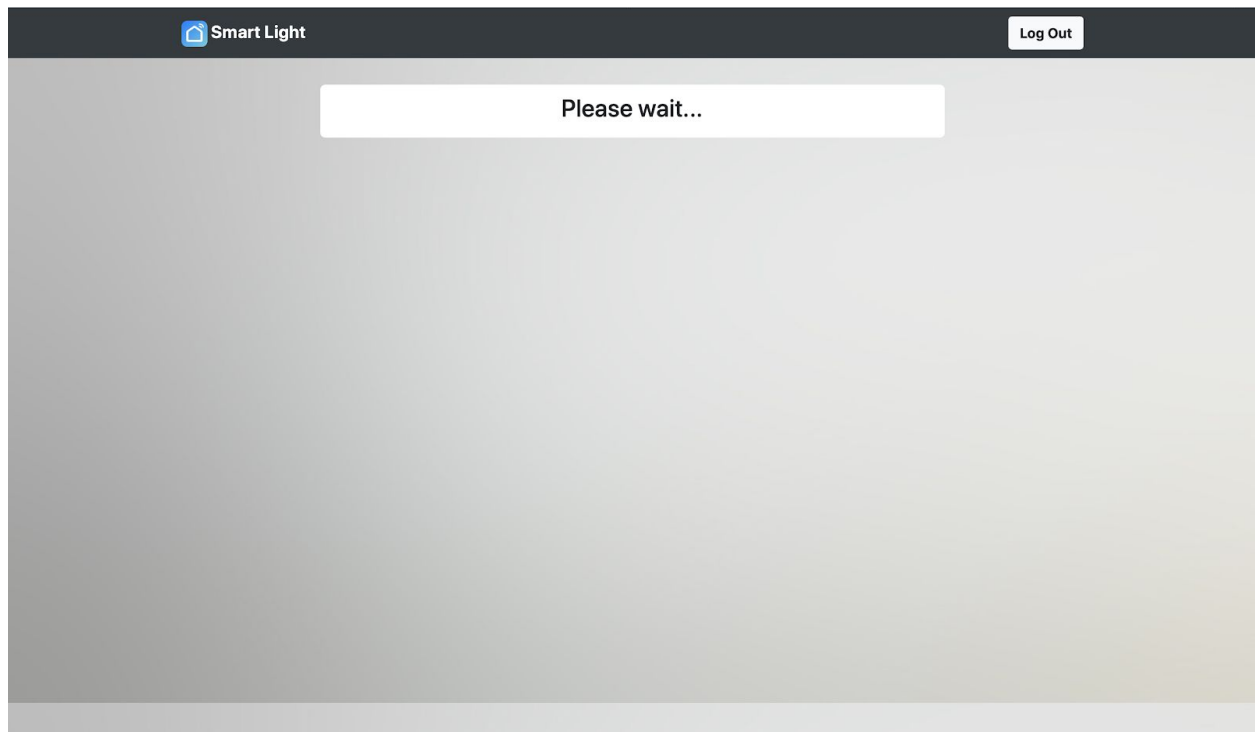
รูปที่1 หน้าจอ log in

2. ถ้าผู้ใช้กรอก username หรือ password ไม่ถูกต้อง จะขึ้นข้อความแจ้งเตือน ดังนี้



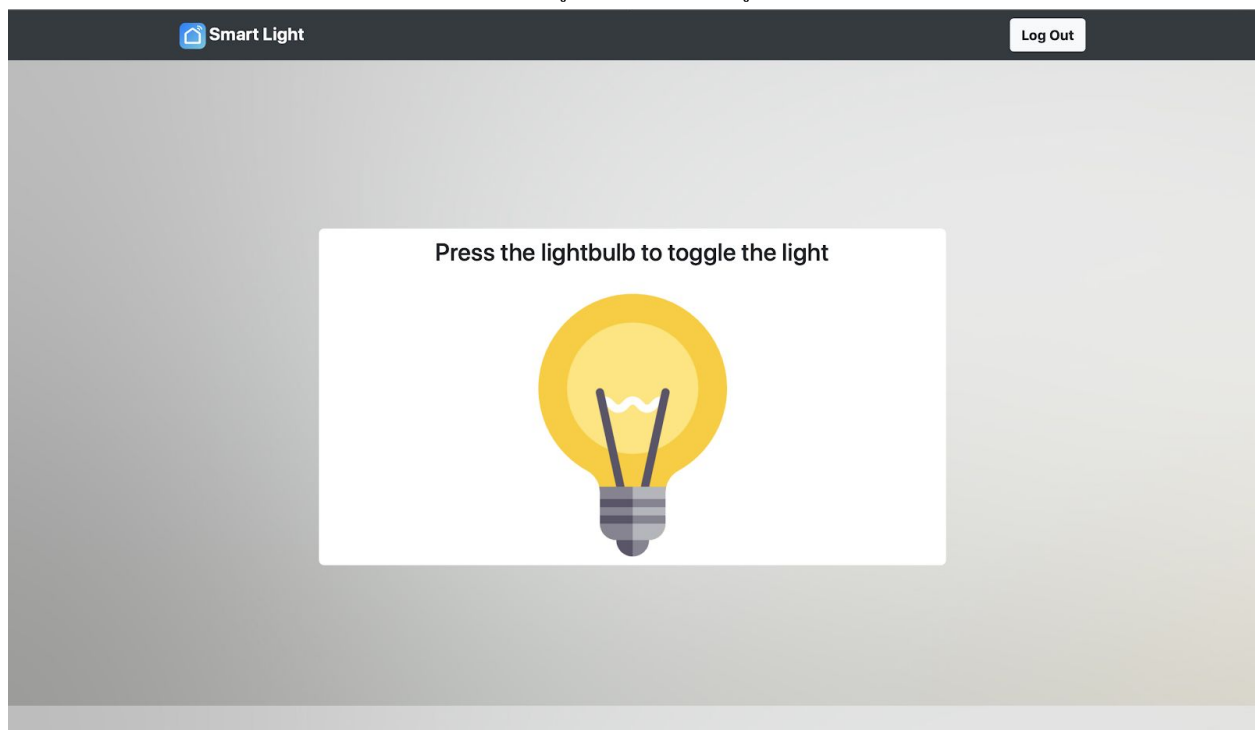
รูปที่ 2 หน้าจอ log in ที่มี alert

3. เมื่อผู้ใช้งาน log in เข้าไปแล้ว จะมีข้อความ “Please Wait ” ขึ้นตอนรอโหลด



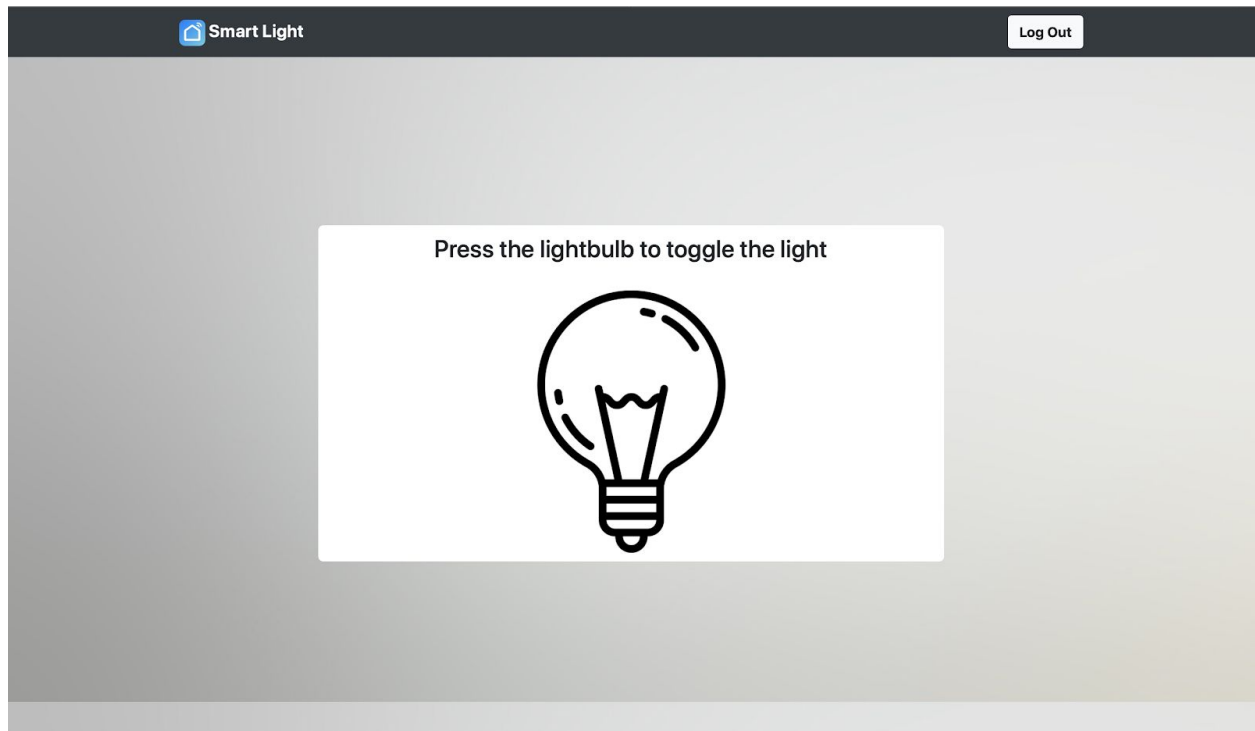
รูปที่ 3 หน้าจอ loading

4. เมื่อโหลดเสร็จจะขึ้นสถานะของหลอดไฟ ผู้ใช้สามารถคลิกที่รูปหลอดไฟเพื่อเปิดหรือปิดไฟได้



รูปที่ 4.1 รูปสถานะหลอดไฟ ไฟเปิด





รูปที่ 4.1 รูปสถานะหลอดไฟ ไฟปิด

### 3.3 Front-end development

Person in charge: Ittiwat Sukhumdhanakul

Responsibility:

พัฒนาเว็บไซต์ด้วย HTML CSS และ Javascript

เชื่อมต่อเว็บไซต์กับ Server ของ Netpie ด้วย Microgear

โดยมีขั้นตอนต่างๆดังต่อไปนี้

- ทำการระบุข้อมูลของ device ที่ต้องการติดต่อ และสร้างตัวแปรที่เก็บ element ของ html

```
const APPID = "ClapClapLight";
const KEY = "i3V3f9P04j9XHo1";
const SECRET = "5RtkB5BtbOnhn6h50xkjUUSqW";
const ALIAS = "DigitalOUTPUT_HTML_web";
const thing1 = "NodeMCU1";

var connect = false;
var statuss = document.getElementById('status');
var loading = document.getElementById('loading');
```

- เขียนฟังก์ชันเพื่อ handle การติดต่อต่างๆระหว่างหน้าเว็บ และ device

```
function switchPress(logic){
    if(logic == 1 ){
        microgear.chat(thing1,"ON");

    }else if(logic == 0 ){
        microgear.chat(thing1,"OFF");
    }
}

var microgear = Microgear.create({
    key: KEY,
    secret: SECRET,
    alias : ALIAS
});

microgear.on('message', function(topic,data) {
    if(data=="ON"){
        lightbulb.setAttribute("src","./onn.png")
    }else if(data=="OFF"){
        lightbulb.setAttribute("src","./offf.png")
    }
});

microgear.on('connected', function() {
    microgear.setAlias(ALIAS);
    microgear.chat(thing1,"GET");
    connect = true;
});

microgear.on('present', function(event) {
    console.log(event);
});

microgear.on('absent', function(event) {
    console.log(event);
});

microgear.resettoken(function(err) {
    microgear.connect(APPID);
});
```

- เขียนโค้ดในส่วนที่ handle การกดที่หลอดไฟ

```

var lightbulb = document.getElementById('lightbulb');
lightbulb.onclick = function(){
    if(lightbulb.getAttribute("src") === './off.png'){
        switchPress(1);
    }
    else{
        switchPress(0);
    }
}

```

### 3.4 Embedded System Development

Person in charge : Nattichai Sutipanwihan

Responsibility :

ศึกษา datasheet ของ microphone MP45DT02

Data sheet : <https://www.marutsu.co.jp/contents/shop/marutsu/ds/MP45DT02.pdf>

- ทำการศึกษาการเขียนโดยศึกษาจากตัวอย่าง  
[https://github.com/tueytoma/Hardware\\_Lab\\_Project/blob/master/Sensors%205730625221/MP45DT02/Src/main.c](https://github.com/tueytoma/Hardware_Lab_Project/blob/master/Sensors%205730625221/MP45DT02/Src/main.c)
- Implement code ในส่วนของข้อมูลที่ได้รับจาก NetPie :  
<https://github.com/netpieio/microgear-esp8266-arduino>  
<https://netpie.gitbooks.io/nodemcu-esp8266-on-netpie/content/lab-7.html>

Sensor ที่เลือกใช้งาน

ชื่อ : MP45DT02

รายละเอียด : MP45DT02 นั้นคือไมโครโฟนประเภทดิจิทัล MEMS ซึ่งสามารถรับเสียงรับเสียงได้ดีในทุกๆ ทิศทาง (Omnidirectional) ซึ่งฝังตัวอยู่บนบอร์ด STM32 อยู่แล้วและสามารถใช้ library I2S ของบอร์ดในกับรับค่าจากเซนเซอร์นี้ได้เลย จึงง่ายต่อการใช้งาน

## ขั้นตอนการเตรียมก่อนนำ sensor มาใช้งาน

### การ Implement ใน STM32F407VG

1. ทำการประกาศตัวแปร pdm, pcm และกำหนดค่าเริ่มต้น

```
uint16_t PDM_buffer[PDM_BUFFER_SIZE];  
uint16_t PDM_value = 0;  
uint16_t pcmCount = 0;  
uint8_t i;  
uint8_t PCM_value = 0;
```

```
float leaky_PCM_buffer = 0.0;  
float leaky_AMP_buffer = 0.0;  
char uart_temp_display_buffer[100];
```

```
float maxAmp = 0;  
int check = 0;
```

2. Initialize ค่า system clock และ peripherals ต่าง ๆ

```
HAL_Init();
```

```
SystemClock_Config();
```

```
MX_GPIO_Init();
```

```
MX_I2S2_Init();
```

```
MX_USART2_UART_Init();
```

3. เขียน Code ส่วน Main loop โดยมีลำดับการทำงานดังนี้

```
while (1) {
```

- 3.1 รับค่าจาก protocol I2S ลงใน PDM\_buffer

```
HAL_I2S_Receive(&hi2s2, PDM_buffer, PDM_BUFFER_SIZE, 1000);
```

- 3.2 คำนวณค่า amplitude ในช่วงเวลาหนึ่ง โดยเก็บไว้ในตัวแปร leaky\_AMP\_buffer และเพิ่มค่า pcmCount ไป 1 เพื่อนับจำนวนค่าที่คำนวณไปแล้ว

```
for (i = 0; i < PDM_BUFFER_SIZE; i++) {  
    PCM_value = -PDM_BLOCK_SIZE_BITS / 2;  
    PDM_value = PDM_buffer[i];
```

```
    while (PDM_value != 0)  
    {
```

```
        PCM_value++;  
        PDM_value ^= PDM_value & -PDM_value;
```

```

    }

    leaky_PCM_buffer += PCM_value;
    leaky_PCM_buffer *= LEAKY_KEEP_RATE;
    leaky_AMP_buffer += absFloat(leaky_PCM_buffer);
    leaky_AMP_buffer *= LEAKY_KEEP_RATE;
}
pcmCount++;

```

3.3 เก็บค่า amplitude ที่มากที่สุดไว้ในตัวแปร maxAmp

```

if (maxAmp < leaky_AMP_buffer)
    maxAmp = leaky_AMP_buffer;

```

3.4 หากจำนวนค่าที่คำนวณไปแล้วครบ 500 ค่า จะเปรียบเทียบค่า maxAmp ว่ามากกว่า 65000 หรือไม่ ถ้ามากกว่าให้ส่งข้อมูลไปทาง UART ว่าเสียงที่ได้รับมี amplitude มากกว่าที่กำหนดไว้ พร้อมกับ reset ค่าต่าง ๆ กลับไปเป็นค่าเริ่มต้น

```

if (pcmCount == 500) {
    if(maxAmp >= 65000){
        int d[1] = {1};
        sprintf(uart_temp_display_buffer, d, sizeof(d));
        HAL_UART_Transmit(&huart2,
(uint8_t*)uart_temp_display_buffer, strlen(uart_temp_display_buffer), 100);
    }
    pcmCount = 0;
    maxAmp = 0;
    leaky_PCM_buffer = 0;
    leaky_AMP_buffer = 0;
}
}

```

## 4.Problem

ในการทำงานครั้งนี้มีปัญหาเกิดขึ้นตามที่ดังจะกล่าวต่อไปนี้

1. ตบมือแล้วบางทีไฟไม่ติด (ไมค์โครโฟนไม่รับเสียง)
2. โค้ดติดบัค ตัวอย่างเช่น
  - 2.1 เกิดการ delay มากเวลาส่งข้อมูลจาก web server
  - 2.2 จาก log in มาหน้าหลัก ต้องใช้เวลารอ
 

แต่ปัญหาเหล่านี้ได้รับการแก้ไขแล้วเรียบร้อยแล้ว
3. สายที่เชื่อมต่อหลุดง่าย
4. ก่อนจะอัปโหลดโค้ดลง nodeMCU ต้องถอดสาย STM32 ออกก่อน เนื่องจากมี baudrate ที่ตรงกัน