

รายงาน

**Smartest Light**

จัดทำโดย

นายสรวิศ ธิษฐาพงศ์	6330528521
นายอรุชา คงสุวรรณ	6330580021
นายอัศววัฒน์ บุรโชติวัฒน์	6330585221

เสนอ

คณาจารย์วิชาวิศวกรรมคอมพิวเตอร์

รายงานนี้เป็นส่วนหนึ่งของวิชา 2110366 Embedded System Laboratory

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ภาคปลาย ปีการศึกษา 2564

## Name and description of the application

Name: Smartest Light

Description:

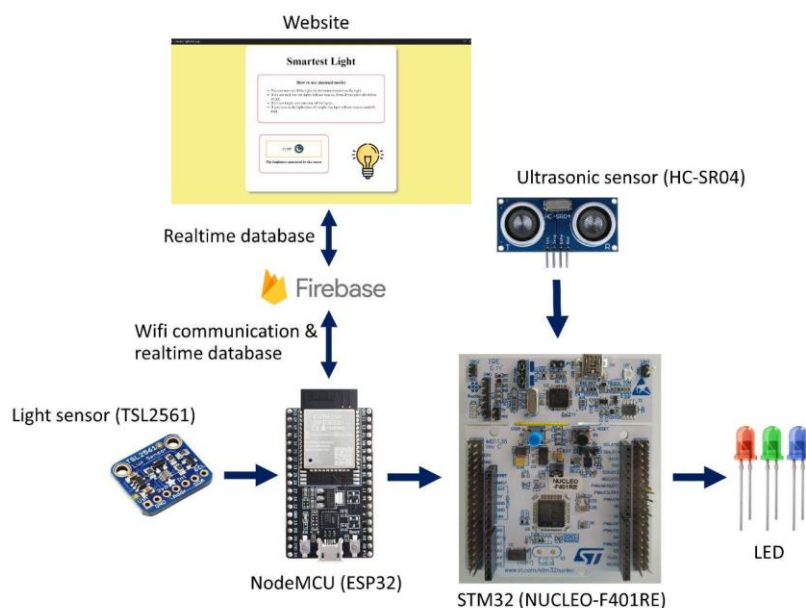
Smartest Light เป็นเว็บแอปพลิเคชันที่มีขึ้นเพื่อช่วยลดการสัมผัส ซึ่งเป็นไปตามจุดประสงค์ contactless society และยังมีประโยชน์ดังนี้

- ลดการสัมผัสซึ่งเป็นความเสี่ยงต่อการติดโรค
- ป้องกันการลืมนเปิดหรือปิดไฟ
- ประหยัดพลังงานไฟฟ้า

รวมไปถึงอื่นๆ อีกมาก เช่น สามารถบอกค่าความเข้มแสงและสามารถควบคุมไฟได้หากต้องการ

## Overview of our system

ระบบ smartest light ทำงานโดยแบ่งเป็น 2 ส่วน ส่วนแรกเป็น embedded system ควบคุมโดย STM32 (NUCLEO-F401RE) เป็นการรับข้อมูลจาก Ultrasonic sensor และข้อมูลจาก NodeMCU (ESP32) และส่งคำสั่งเปิด/ปิดไฟไปยัง LED ส่วนที่สองเป็นส่วนเว็บไซต์ ซึ่งแสดงค่าจาก Light sensor โดยเมื่อค่าความสว่างไม่เกิน 25 จะเปิดไฟอัตโนมัติ และถ้าค่าความสว่างมากกว่า 25 ไฟก็จะปิดอัตโนมัติ และสามารถกดปุ่มเปิด/ปิดไฟได้ทันทีด้วย เราใช้ Realtime database บน Firebase ในการรับส่งข้อมูลระหว่างเว็บไซต์กับ NodeMCU (ESP32)



## How to use the Web Application

URL website: <https://smartest-light.firebaseio.com>

Link to Github repository: <https://github.com/EarthAkkharawat/2110366-Final-project.git>

1. คุณสามารถเปิด/ปิดหลอดไฟได้จากปุ่มรูปหลอดไฟ ดังรูป

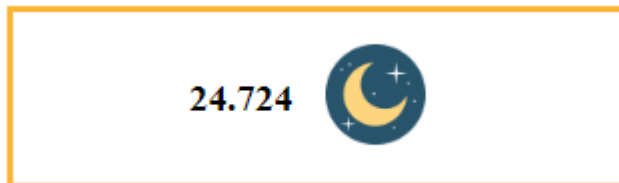


เปิดไฟ



ปิดไฟ

2. ถ้าหากต้องการเปิดไฟในตอนที่มีความสว่างเพียงพออยู่แล้วให้กดปุ่มเปิดไฟที่กล่าวถึงในข้อที่หนึ่ง ไฟจะยังไม่เปิดทันทีเมื่อกดปุ่มแต่จะเปิดอัตโนมัติเมื่อถึงตอนที่มีความสว่างไม่เพียงพอหากสว่างไม่เพียงพอหลอดไฟจะเปิดขึ้นมาเอง ถ้าหากต้องการปิดหลอดไฟให้กดปุ่มปิดไฟที่กล่าวมาในข้อหนึ่ง
3. ความสว่างที่วัดได้จาก Light Sensor (หน่วย lux) สามารถดูได้จากเว็บไซต์ ดังรูป



## Role and responsibility

เราได้แบ่งหน้าที่ความรับผิดชอบออกเป็น 3 ส่วน ได้แก่

1. System Architecture & Embedded System Development
2. UI/UX Designer and Development - Design and develop user interface & Team Management
3. Web Development - Create a database and develop the website
  - a. Frontend development
  - b. Backend development

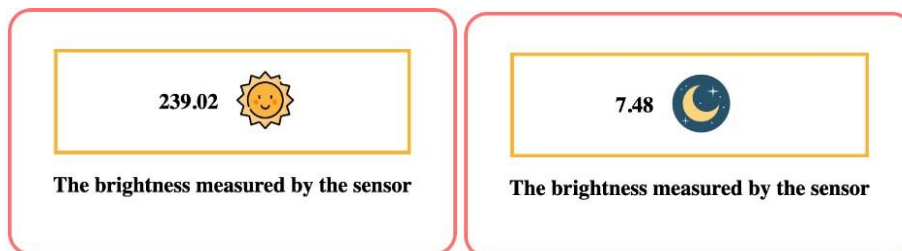
## Role and responsibility

### UI/UX Designer and Development - Design and develop user interface & Team Management

(นายอรุชา คงสุวรรณ 6330580021)

การออกแบบ Web Application จะต้องแสดงข้อมูลที่เข้าใจง่าย และไม่ซับซ้อน โดย Smartest Light Web Application จะแบ่งข้อมูลทั้งหมดเป็น 3 ส่วน คือ วิธีการใช้งาน ข้อมูลค่าความสว่างที่วัดได้ ปุ่มควบคุมหลอดไฟ(เปิด/ปิด) ซึ่งส่วนที่ User จะควบคุมอุปกรณ์ได้ จะมีเพียงแต่ส่วนที่เป็นปุ่มควบคุมหลอดไฟ นอกเหนือจากนี้จะเป็นส่วนที่แสดงข้อมูลเพียงอย่างเดียว ไม่สามารถทำอะไรได้

สำหรับ User Interface ได้มีการวางแผนไว้ให้มีสีขาว และเหลือง โดยจะประกอบไปด้วย 3 ส่วน ตามที่กล่าวไว้ ส่วนบนจะเป็นคำแนะนำการใช้ Web Application และส่วนล่างจะประกอบไปด้วย ค่าความสว่างและปุ่มควบคุมการทำงานของหลอดไฟ โดยส่วนหน้าต่างแสดงค่าความสว่างจะแสดงค่าความสว่างที่มีหน่วยเป็น lux และมีสัญลักษณ์กำกับว่าความสว่างนั้น เป็นตอนกลางคืนหรือกลางวัน ซึ่งเมื่อความสว่าง ที่วัดได้มีค่าต่ำกว่าที่กำหนด (25 lux) จะทำการเปิดหลอดไฟโดยอัตโนมัติ



อีกส่วนที่เป็นปุ่มเปิดปิดหลอดไฟ User สามารถควบคุมหลอดไฟได้จากปุ่มนี้ไม่ว่ากรณีใดๆ โดยเมื่อไฟเปิดหรืออยู่ปุ่มจะแสดงไอคอนต่างกัน



เปิดไฟ



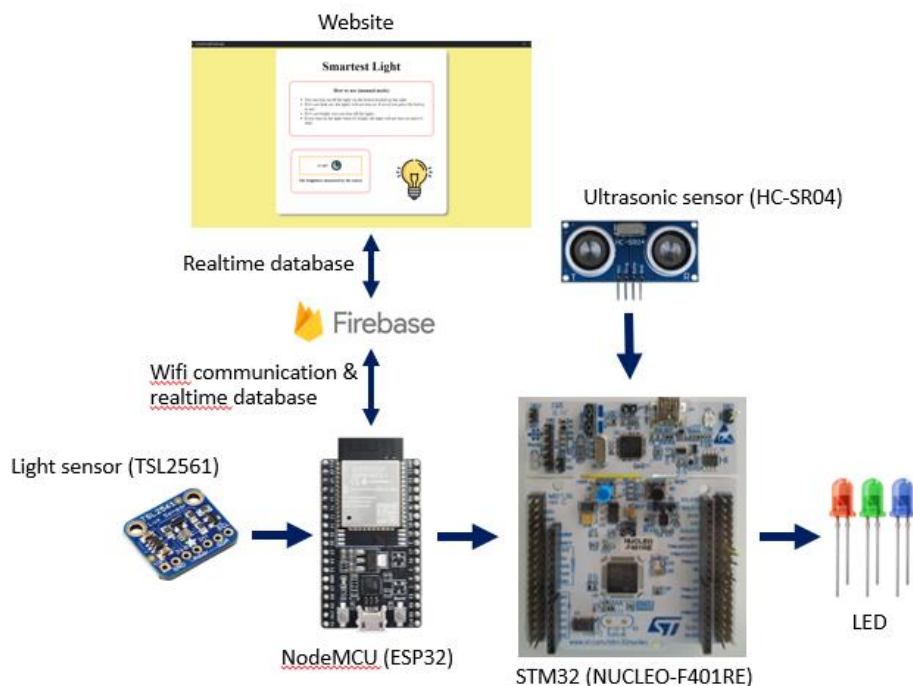
ปิดไฟ

## System Architecture & Embedded System Development - Design and integrate the system

(นายสรวิศ ธิษฐาพงศ์ 6330528521)

ได้ทำการออกแบบ ระบบและโครงสร้าง รวมถึงพัฒนา code ทั้งใน STM และ MCU บางส่วน โดยเริ่มแรกทำการศึกษาวิธีการใช้ Sensor ทั้ง Ultrasonic Sensor และ Light Sensor รวมถึงการใช้งาน MCU เบื้องต้น และการเชื่อมต่อ WIFI โดยระบบของเรา จะใช้ Ultrasonic Sensor ต่อกับ STM โดยมีการรับค่าจาก Sensor โดยการใช้งาน Input capture และ Light Sensor จะเชื่อมต่อกับ MCU โดยมีการใช้ Library SparkFunTSL2561 เข้ามาช่วยในการรับค่าและประเมินผล

โดยในตอนแรกมีปัญหาเกี่ยวกับการใช้ Ultrasonic Sensor ในตอนแรกใช้ UART ในการรับส่งข้อมูล แต่มีปัญหาคือ ค่าที่ได้รับเข้ามาค่อนข้างจะผิดไปจากความเป็นจริง จึงทำการศึกษาเพิ่มเติมเกี่ยวกับการทำงานของ Ultrasonic Sensor และเปลี่ยนมาใช้วิธี Input Capture จึงได้ค่าที่ถูกต้องมากขึ้น จากนั้นเมื่อใช้งาน Sensor ได้แล้ว จึงพยายามทำให้ STM และ MCU มีการสื่อสารกันผ่าน UART โดยการใช้ Pin Rx2 และ Tx2 จึงเกิดปัญหขึ้นทำให้ไม่สามารถ สื่อสารกันได้ จึงเปลี่ยนไปใช้ Pin อื่น และใช้ Library ESPSoftwareSerial เข้ามาช่วย จึงทำให้สามารถแก้ไขปัญหบางส่วนไปได้ จากนั้นพอระบบทุกอย่างเรียบร้อย ก็ได้ทำการรวมทุกอย่างเข้าด้วยกันและได้คิดการทำงานออกมาดังนี้



1. Light Sensor จะคอยรับข้อมูลแสงและส่งเข้าไปใน MCU
2. MCU จะเก็บค่าจาก Light Sensor และทำการประมวลผล
3. MCU คำนวณ state ออกมา โดยเพื่อนช่วยทำหน้าที่ webpage ที่จะมีปุ่มเปิดปิดไฟอยู่ ซึ่งถ้าไม่มีการกดปุ่มเปิด แล้ว LUX > 25 จะเป็น State "00" ถ้ามีการกดปุ่มเปิดแล้ว LUX > 25 จะเป็น State "01" ถ้าไม่มีการกดเปิด แล้ว LUX <= 25 จะเป็น State "10" ถ้ามีการกดเปิดแล้ว LUX <= 25 จะเป็น State "11" หลังจากคำนวณ State แล้ว MCU ก็จะส่งค่า State ไปยัง STM

11:16:42.984 -> lux: 133.80	11:16:01.529 -> lux: 134.21
11:16:43.737 -> 00	11:16:02.140 -> 01

State 00

State 01

11:17:30.334 -> lux: 4.06	11:14:23.948 -> lux: 5.41
11:17:30.945 -> 10	11:14:24.559 -> 11

State 10

State 11

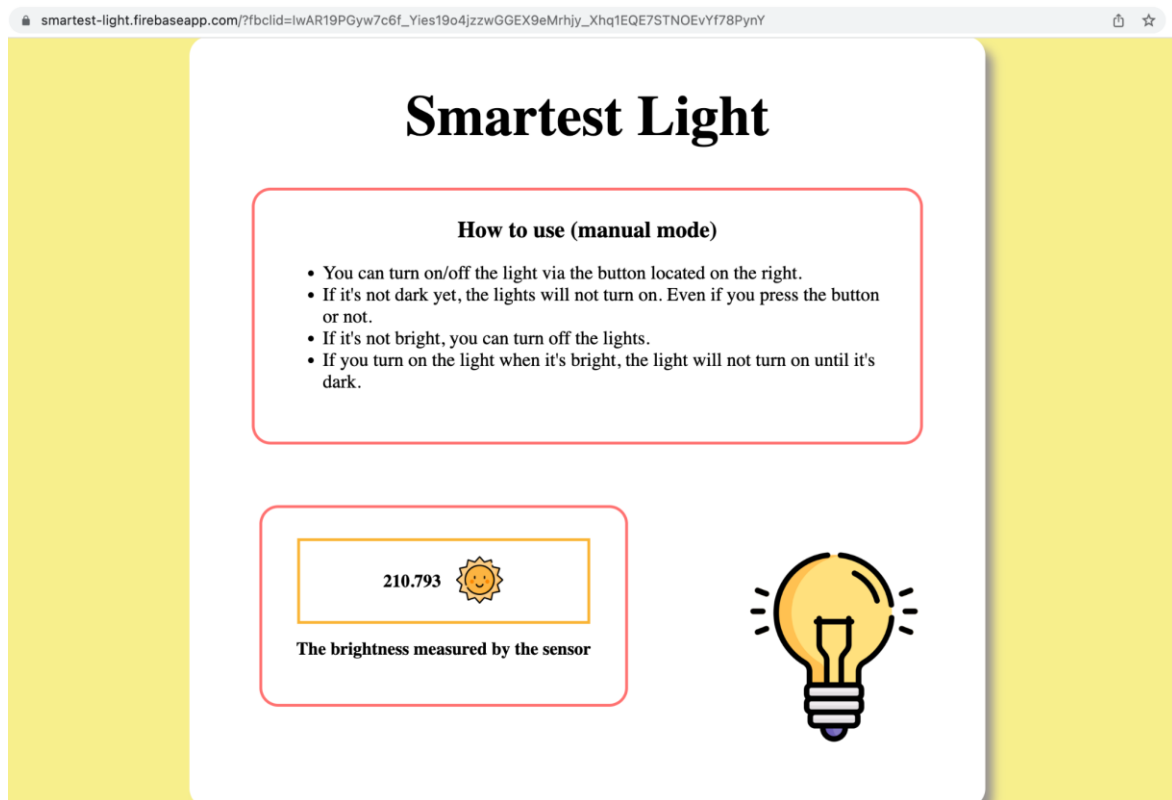
4. STM จะรับข้อมูล State ผ่าน UART1 ถ้าเป็น State "00" ก็จะไม่มีการทำอะไร ถ้าเป็น State "01" STM จะรอให้มีการส่ง State 10 จาก MCU จากนั้นทำการเปิดไฟ ถ้าเป็น state "10" STM จะรอให้มีการเคลื่อนไหวผ่าน Ultrasonic Sensor จากนั้นจะทำการเปิดไฟ แต่ถ้าเป็น State "11" STM จะทำการเปิดไฟทันที

## Web Development - Create a database and develop the website

(6330585221 อัครวัฒน์ บุรโชควิวัฒน์)

### Frontend development

เขียน layout และตกแต่งหน้าเว็บโดยใช้ภาษา html และ css โดยเว็บไซต์ที่ออกแบบเป็นดังภาพ



ซึ่งมีรายละเอียดแยกย่อยเป็นดังต่อไปนี้

1. ชื่อเว็บไซต์และวิธีการใช้

## Smartest Light

### How to use (manual mode)

- You can turn on/off the light via the button located on the right.
- If it's not dark yet, the lights will not turn on. Even if you press the button or not.
- If it's not bright, you can turn off the lights.
- If you turn on the light when it's bright, the light will not turn on until it's dark.

2. ช่องแสดงค่าความสว่างที่วัดได้จาก light sensor โดยถ้าค่ามากกว่า 25 lux จะแสดง icon ดวงอาทิตย์ แต่ถ้าน้อยกว่าหรือเท่ากับ 25 lux จะแสดง icon ดวงจันทร์

239.02



**The brightness measured by the sensor**

7.48



**The brightness measured by the sensor**

3. ปุ่มเปิด/ปิดไฟ





## Backend development

ออกแบบ backend ของเว็บไซต์โดยใช้ภาษา javascript และ ภาษา c โดยผ่านทาง arduino IDE และใช้ realtime database ของ firebase โดยมี flow การทำงานดังภาพนี้



Reference: <https://randomnerdtutorials.com/esp32-firebase-realtime-database/>

การทำงานของ backend มี 4 ส่วน คือ

1. การแสดงค่าความสว่างบนหน้าเว็บโดยดึงค่าความสว่างมาจาก realtime database และการแสดง icon ที่เปลี่ยนไปตามค่าความสว่าง

```
// Attach an asynchronous callback to read the data
databaseFloat.on(
  "value",
  (snapshot) => {
    floatReading = snapshot.val();
    console.log(floatReading);
    document.getElementById("brightness").innerHTML = floatReading;
    if (floatReading <= 25) {
      document.getElementById("sicon").src = "image/moon.png";
    } else {
      document.getElementById("sicon").src = "image/sun.png";
    }
  },
  (errorObject) => {
    console.log("The read failed: " + errorObject.name);
  }
);
```

2. การแสดงปุ่มเปิดเป็นหลอดไฟที่เปิดอยู่และปุ่มปิดเป็นหลอดไฟที่ปิดอยู่ และการ write buttonState ลงบน realtime database

```
document.getElementById("on-off").addEventListener("click", function onClick() {
  if (check == 0) {
    console.log("test");
    document.getElementById("imgg").src = "image/nolightbut.png";
    check = 1;
    update(ref(db, "test/"), {
      buttonState: 0,
    });
  } else {
    console.log("test2");
    document.getElementById("imgg").src = "image/light-bulb.png";
    check = 0;
    update(ref(db, "test/"), {
      buttonState: 1,
    });
    console.log("test");
  }
});
```

3. การเชื่อมต่อกับ database ของเว็บไซต์

```
import { initializeApp } from "https://www.gstatic.com/firebasejs/9.6.11/firebase-app.js";
import { getDatabase, ref, onValue, update, get, set } from "https://www.gstatic.com/firebasejs/9.6.11/firebase-database.js";
var firebaseConfig = {
  apiKey: "AIzaSyDU3ZLPqguNfSx2TBqNUsPZnsruVY_H3tQ",
  authDomain: "smartest-light.firebaseio.com",
  databaseURL: "https://smartest-light-default-rtdb.firebaseio.com",
  projectId: "smartest-light",
  storageBucket: "smartest-light.appspot.com",
  messagingSenderId: "863791815861",
  appId: "1:863791815861:web:cc08cc550c2010b88772fc",
  measurementId: "G-3WEQ651SPY",
};
```

4. การอ่านข้อมูลและเขียนข้อมูลลงใน realtime database ของ nodeMCU โดยใช้ arduino IDE

```
// Write an Double number on the database path test/brightness
if (Firebase.RTDB.setDouble(&fbdo, "test/brightness", lux)) {
  Serial.println("PASSED");
  Serial.println("PATH: " + fbdo.dataPath());
  Serial.println("TYPE: " + fbdo.dataType());
}
```

```
// Get buttonState from the database
if (Firebase.RTDB.getInt(&fbdo, "/test/buttonState")) {
    if (fbdo.dataType() == "int") {
        buttonState = fbdo.intData();
        Serial.println(buttonState);
    }
}
```