

ใบงานการทดลองที่ 1

เรื่อง สมอกลฝั่งตัว เบื้องต้น

1. จุดประสงค์

1.5.1 ทดลองใช้งาน ESP32 เบื้องต้น

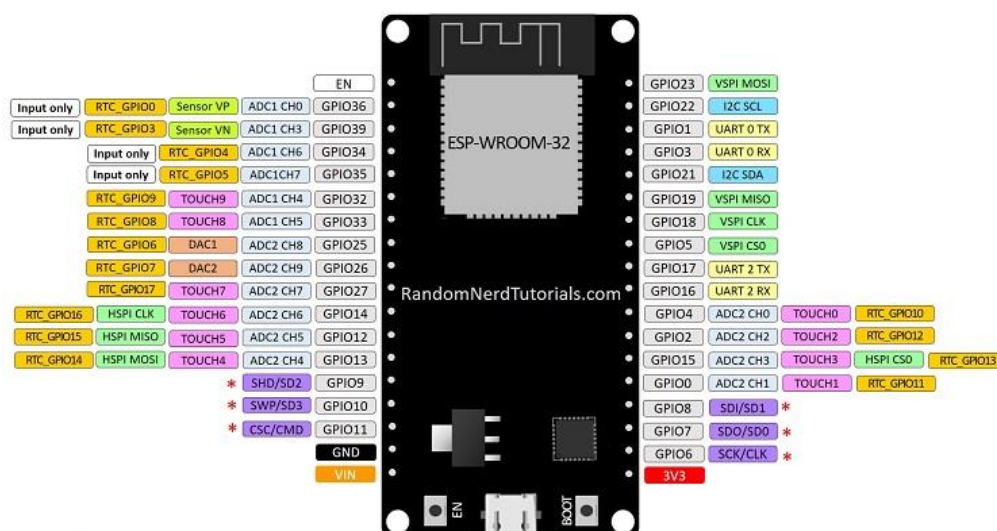
2. ทฤษฎี

2.1 ESP32

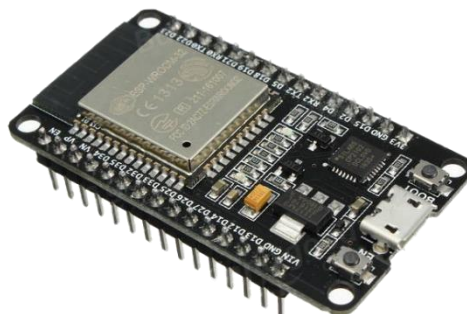
ESP32 คือ wifi microcontroller ที่ถูกพัฒนาต่อจาก ESP8266 โดยเพิ่ม CPU เป็น 2 core, Wi-Fi ที่เร็วขึ้น, มีขา GPIO ให้ใช้งานมากขึ้น และรองรับ Bluetooth อีกด้วย นอกจากนี้ ESP32 ยังมาพร้อมกับ touch-sensitive pins ที่สามารถใช้ปลุก ESP32 จากโหมด deep sleep และยังมี hall effect sensor และ temperature sensor ในตัว (รุ่นล่าสุดของ ESP32 ไม่มี temperature sensor ในตัวอีกแล้ว) และแน่นอนว่าด้วยคุณสมบัติที่ดีกว่ามากจึงทำให้ ESP32 มีราคาแพงกว่า ESP8266 พอสมควร

ESP32 DEVKIT V1 – DOIT

version with 36 GPIOs



* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on ESP-WROOM-32 and are not recommended for other uses.



รูปที่ 1.1 ESP32

2.2 เว็บไซต์จำลองการทำงานของ ESP32

<https://wokwi.com/> เป็นเว็บไซต์ที่ให้บริการจำลอง การทำงานของบอร์ดไมโครคอนโทรลเลอร์ โดยผู้ใช้สามารถเขียนโปรแกรมและจำลอง (Simulator) การทำงานของบอร์ด Arduino, Raspberry Pi Pico และ ESP32 หรือบอร์ดไมโครคอนโทรลเลอร์ ที่ออกแบบมาเพื่อเรียนรู้การเขียนโปรแกรมโดยไม่ต้องใช้ฮาร์ดแวร์จริง

นอกจากนี้ <https://wokwi.com/> ยังสามารถเพิ่มบอร์ดแบบกำหนดเองได้อีกด้วย ดังนั้นจึงไม่จำกัด เฉพาะบอร์ดที่รองรับอย่างเป็นทางการเท่านั้น ผู้ใช้สามารถเริ่มโครงการใหม่ได้ ด้วยการเลือกบอร์ดและเพิ่มอุปกรณ์ และเขียนโค้ดด้วยตัวเอง หรือเริ่มจากโครงการที่มีอยู่แล้ว เช่น โครงการเซ็นเซอร์อุณหภูมิและความชื้น ESP32 + DHT22 อีกทั้งยังสามารถเขียนโปรแกรมด้วย Arduino sketch เพื่อเริ่มการจำลองการทำงานของบอร์ด โปรแกรมจะแสดงค่าอุณหภูมิและความชื้นในแบบจำลองได้

ผู้ใช้งานสามารถเพิ่มอุปกรณ์เสริมอื่นๆ เช่น ไฟ LED, Potentiometers, ปุ่ม, เซ็นเซอร์, VCC, GND และอื่นๆ สามารถบันทึกไฟล์โค้ดของซอร์สโค้ดในรูปแบบของไฟล์ JSON ได้

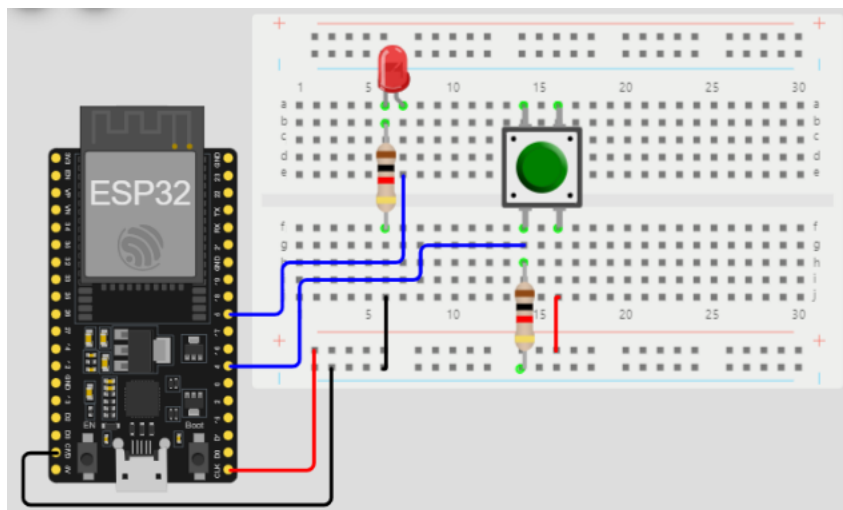
3. เครื่องมือและอุปกรณ์การทดลอง

3.1 โปรแกรม Arduino IDE 1.8.16 หรือสูงกว่า	1	โปรแกรม
3.2 สาย USB สำหรับ ESP32	1	เส้น
3.3 ชุดทดลอง ESP32	1	ชุด
3.4 สายต่อวงจร	1	ชุด
3.5 เครื่องคอมพิวเตอร์แบบพกพา	1	เครื่อง
3.6 แผงต่อวงจร	1	ตัว
3.7 ตัวต้านทาน 1 k Ω	6	ตัว
3.8 ไดโอดเปล่งแสง	6	ตัว

4. ลำดับขั้นตอนการทดลอง

การทดลองที่ 1

1.1. ทำการต่อวงจรตามรูปที่ 1.3 และ รูปที่ 1.4



รูปที่ 1.3

1.2. ทำการเขียนโปรแกรมที่ลงใน Arduino IDE และทำการโหลดโปรแกรมลงบอร์ดดูผลการทำงาน

```
const int buttonPin = 4;
const int ledPin = 5;
int buttonState = 0;
void setup() {
  Serial.begin(115200);
  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);
}
void loop() {
  buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);
  if (buttonState == HIGH) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }
}
```

1.3. สังเกตผลการทดลองและบันทึกผลการทดลอง

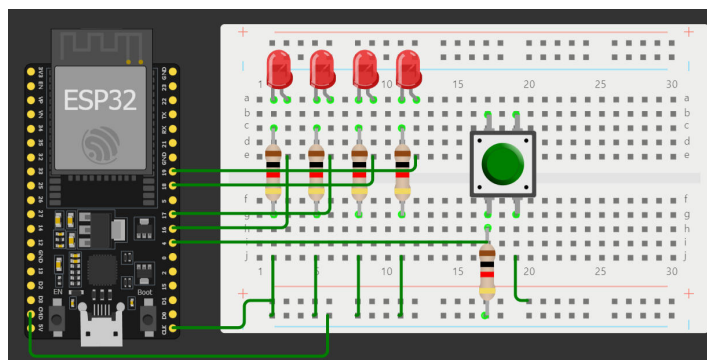
เมื่อ กดปุ่ม จะพบว่า LED ที่ติด และ เมื่อปล่อยปุ่ม LED จะดับ

1.4. สรุปผลการทดลอง

วิธีนี้ สามารถนำหลักการนี้ไปประยุกต์ใช้กับสิ่งอื่นได้ เช่น เมื่อขยับเท้าจะสั่งให้มอเตอร์

การทดลองที่ 2

ทดสอบเขียนโปรแกรมไฟวิ่งจาก โดยใช้ LED 4 หลอดจากซ้ายไปขวาโดยจะวิ่งเพียง 1 ดวง เมื่อกดปุ่ม จะเปลี่ยน ให้วิ่งจากขวาไปซ้าย โดยให้วิ่งทีละ 1 ดวงเช่นกัน



```

1  const int buttonPin = 4;
2  int buttonState = 0;
3  void setup() {
4    Serial.begin(115200);
5    pinMode(buttonPin, INPUT);
6    for(int i =16 ; i<=19; i++){
7      pinMode(i,OUTPUT);
8    }
9  }
10 void led_left(){
11   for(int i = 16; i<=19;i++){
12     digitalWrite(i,HIGH);
13     delay(1000);
14     digitalWrite(i,LOW);
15   }
16 }
17 void led_inverse(){
18   for(int i = 19; i>=16;i--){
19     digitalWrite(i,HIGH);
20     delay(1000);
21     digitalWrite(i,LOW);
22   }
23 }
24 void loop() {
25   buttonState = digitalRead(buttonPin);
26   Serial.println(buttonState);
27   if(buttonState == HIGH){
28     led_inverse();
29   } else {
30     led_left();
31   }
32 }
33

```

2.1 สังเกตผลการทดลองและบันทึกผลการทดลอง

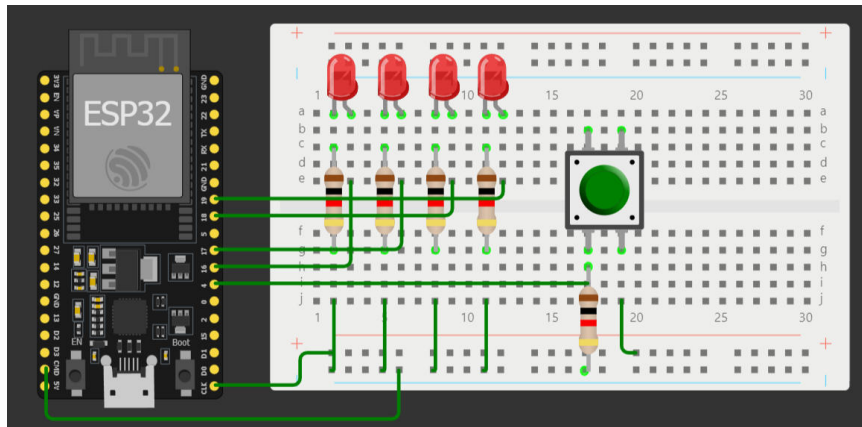
เมื่อเรื้อไปโปรแกรม มุมของ LED รังจากพวงโซ่ไปจน และ เมื่อ กดปุ่ม LED จะดับสว่างขึ้น

2.2 สรุปผลการทดลอง

เมื่อ กด LED ที่ตำแหน่ง i และ i จะดับด้วย $mr + 1$ และเมื่อ กดปุ่มของ i จะดับ $mr - 1$ แทน

ความคิดสร้างสรรค์

ควบคุม Led ด้วยการ delay



```
1  const int buttonPin = 4;
2  int buttonState = 0;
3  void setup() {
4      Serial.begin(115200);
5      pinMode(buttonPin, INPUT);
6      for(int i =16 ; i<=19; i++){
7          pinMode(i,OUTPUT);
8      }
9  }
10 void off_led(){
11     for(int i =16 ; i<=19; i++){
12         digitalWrite(i,LOW);
13     }
14 }
15 void loop() {
16     int delay_led = 0;
17     buttonState = digitalRead(buttonPin);
18     while(buttonState == HIGH){
19         delay(1000);
20         delay_led +=1000;
21         Serial.println(delay_led);
22         buttonState = digitalRead(buttonPin);
23         if (buttonState == LOW){
24             switch (delay_led) {
25                 case 1000:
26                     digitalWrite(16,HIGH);
27                     break;
28                 case 2000:
29                     digitalWrite(17,HIGH);
30                     break;
31                 case 3000:
32                     digitalWrite(18,HIGH);
33                     break;
34                 case 4000:
35                     digitalWrite(19,HIGH);
36                     break;
37                 default:
38                     break;
39             }
40         }
41     }
42     delay(1000);
43     off_led();
44 }
45
```

สังเกตผลการทดลอง

เมื่อเริ่มการทำงานจะไม่มีอะไรเกิดขึ้น แต่เมื่อกดปุ่มค้างเป็นเวลา 1,2,3,4 วินาที จะพบว่า LED จะติดสว่างขึ้นตามเวลาที่กดค้าง เช่น กดค้าง 1 วินาที LED ที่ 1 จะติด

สรุปผลการทดลอง

เมื่อเริ่มโปรแกรมจะทำการรอรับค่าจากปุ่มก่อนจนกว่าจะมีการกด และเมื่อทำการกดค้างจนถึงเวลาที่กดหนด จะเปิดไฟไปตามเวลาที่ตนเองเปิดกดค้าง โดยกำหนดเวลาด้วยการ delay นำไปบวกกันเรื่อยๆแล้วเช็คค่า 1000 2000 3000 และ 4000 ตามลำดับ