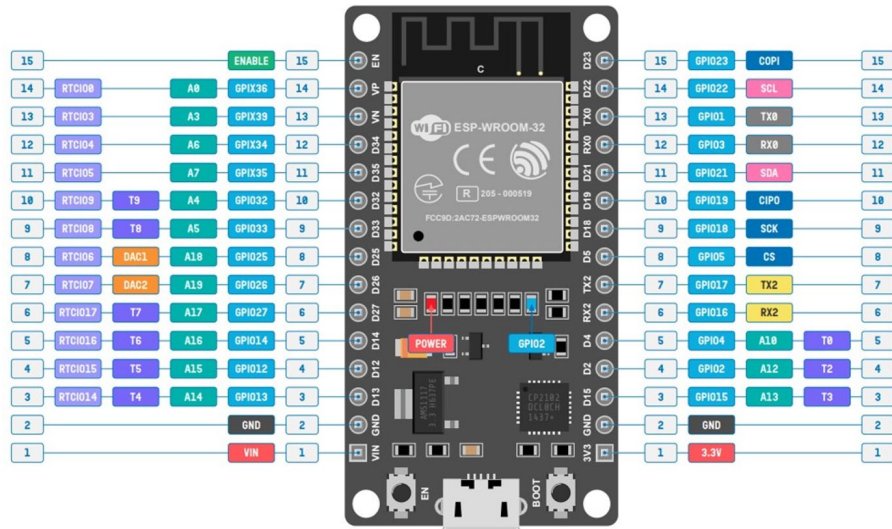


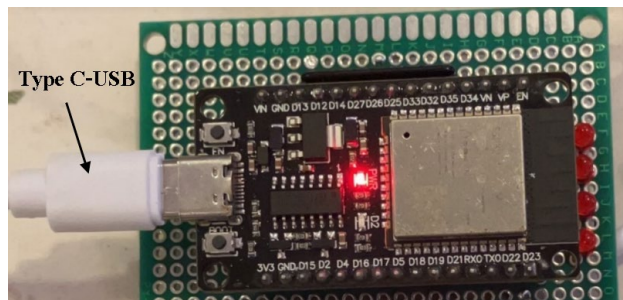
การทดลองที่ 1

ESP32 Microcontroller Board



ขั้นตอนการทดลอง

1. ต่อ ESP32 Microcontroller Board เข้ากับคอมพิวเตอร์ผ่าน USB Port (บน ESP32 Microcontroller Board จะเป็น Type-C USB) ตามรูปที่ 1



รูปที่ 1 บอร์ด ESP32-WROOM-DA Module

2. เปิดโปรแกรม Arduino IDE ไปที่ File ⇒ NEW Sketch เพื่อสร้าง Project ใหม่ ตามรูปที่ 2



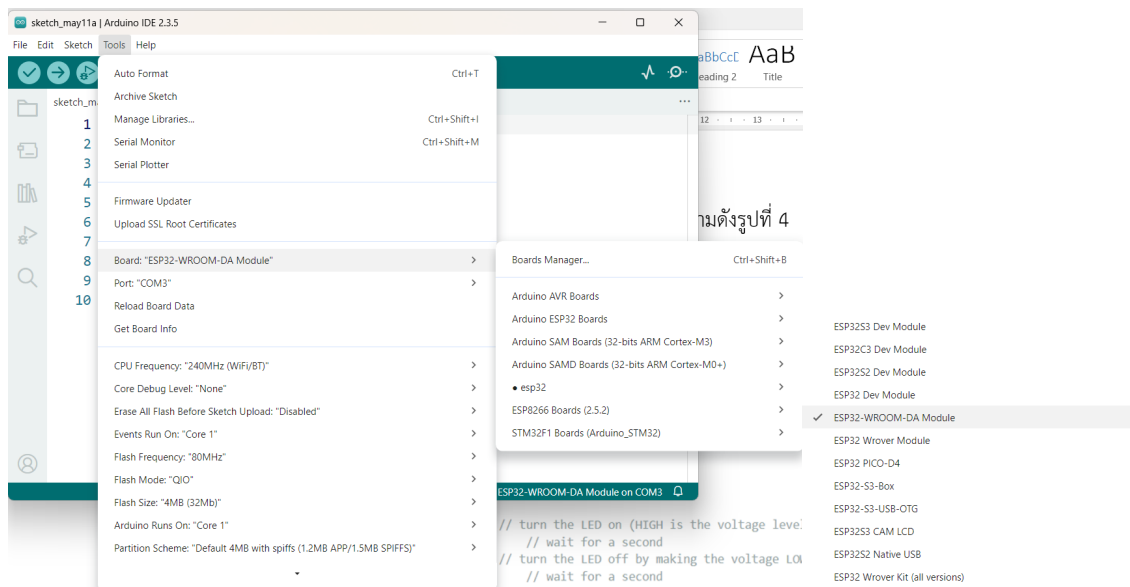
รูปที่ 2 การสร้าง Project ใหม่

3. ติดตั้งบอร์ด ESP32 โดยไปที่ Boards Manager โดยเลือกบอร์ดเป็น ESP32 ตามรูปที่ 3



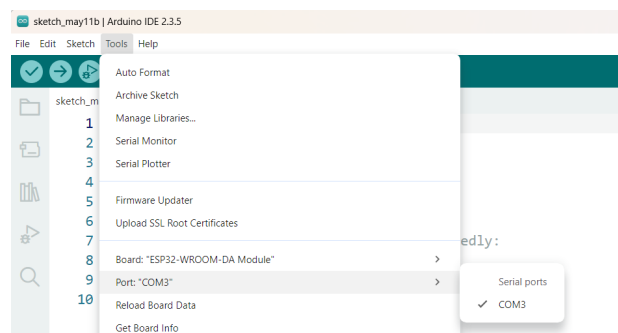
รูปที่ 3 การติดตั้งบอร์ด ESP32

4. เลือกบอร์ด ESP32 โดยไปที่ Tools ⇒ Board: ⇒ • esp32 ⇒ ESP32-WROOM-DA Module




รูปที่ 4 การติดตั้งบอร์ด ESP32

5. ทดสอบการติดต่อระหว่างบอร์ด ESP32 กับเครื่องคอมพิวเตอร์ผ่าน Port Type C-USB โดยไปที่ Tools ⇒ PORT “COMx” ถ้าติดต่อกับบอร์ด ESP32 สำเร็จสามารถกดเครื่องหมาย ✓ และจะมีตัวเลข COM Port ปรากฏขึ้น ตามรูปที่ 5



รูปที่ 5 การติดต่อระหว่างบอร์ด ESP32 กับเครื่องคอมพิวเตอร์

6. ทดสอบการทำงานพื้นฐานของบอร์ดโดยทำการป้อนโปรแกรม ตามรูปที่ 6



```
Blink | Arduino IDE 2.3.5
File Edit Sketch Tools Help
ESP32-WROOM-DA M...
Blink.ino
1 #define LED_ON_BOARD 2
2 // กำหนดให้ LED บนบอร์ด ชื่อ LED_ON_BOARD และเป็น GPIO2
3 void setup() {
4 // กำหนดให้ GPIO ที่ต่อกับ LED เป็น Output Port
5   pinMode(LED_ON_BOARD, OUTPUT);
6 }
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_ON_BOARD, HIGH); // turn the LED on (HIGH is the voltage level)
10  delay(1000); // wait for a second (1000msec)
11  digitalWrite(LED_ON_BOARD, LOW); // turn the LED off by making the voltage LOW
12  delay(1000); // wait for a second
13 }
14
```

รูปที่ 6 โปรแกรมพื้นฐานการทำงานของบอร์ด ESP32

7. ทำการ Upload โปรแกรมไปที่ Board โดยกดปุ่ม Upload ตามรูปที่ 7



```
Blink | Arduino IDE 2.3.5
File Edit Sketch Tools Help
ESP32-WROOM-DA M...
Blink.ino
1 #define LED_ON_BOARD 2
2 // กำหนดให้ LED บนบอร์ด ชื่อ LED_ON_BOARD และเป็น GPIO2
3 void setup() {
4 // กำหนดให้ GPIO ที่ต่อกับ LED เป็น Output Port
5   pinMode(LED_ON_BOARD, OUTPUT);
6 }
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_ON_BOARD, HIGH); // turn the LED on (HIGH is the voltage level)
10  delay(1000); // wait for a second (1000msec)
11  digitalWrite(LED_ON_BOARD, LOW); // turn the LED off by making the voltage LOW
12  delay(1000); // wait for a second
13 }
14
```

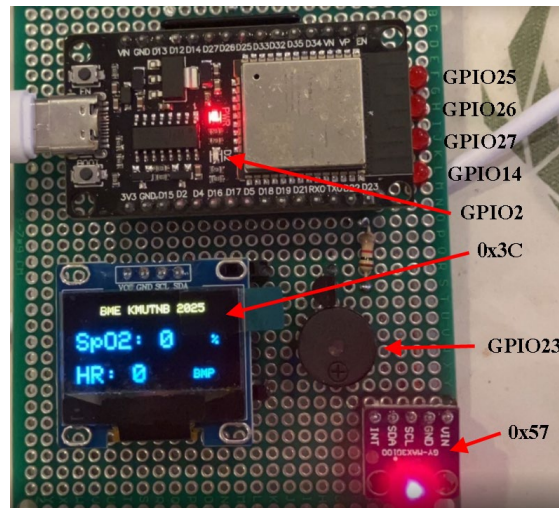
รูปที่ 7

8. สังเกต LED บนบอร์ด ESP32 และทดลองปรับค่า delay(xxxx) แล้วทำการ Upload ใหม่

การทดลองที่ 2

Basic Output Port

บนบอร์ดทดลอง ESP32 ได้ต่อ LED และ Buzzer เข้ากับ GPIO2 GPIO14 GPIO23 GPIO25 GPIO26 และ GPIO27 ตามรูปที่ 1



รูปที่ 1 การต่อ LED และ Buzzer เข้ากับ GPIO ของ ESP32

การทดลองที่ 2.1 การเขียนโปรแกรมควบคุม LED

ขั้นตอนการทดลอง

1. ป้อนโปรแกรม ตามรูปที่ 1 เพื่อทดสอบ LED ที่ต่ออยู่ภายนอกบอร์ด ESP32 โดยเริ่มที่ LED_1 หรือ GPIO25 สังเกตผลที่ LED ภายนอก

```
Blink | Arduino IDE 2.3.5
File Edit Sketch Tools Help
ESP32-WROOM-DA M...
Blinkino
1 #define LED_ON_BOARD 2
2 #define LED_1 25
3 #define LED_2 26
4 #define LED_3 27
5 #define LED_4 14
6 // กำหนดให้ LED บนบอร์ด ชื่อ LED_ON_BOARD LED 1-4 เป็น GPIO2 14 25-27
7 void setup() {
8   // กำหนดให้ GPIO ที่ต่อกับ LED เป็น Output Port
9   pinMode(LED_1, OUTPUT);
10 }
11 // the loop function runs over and over again forever
12 void loop() {
13   digitalWrite(LED_1, HIGH); // turn the LED on (HIGH is the voltage level)
14   delay(1000); // wait for a second (1000msec)
15   digitalWrite(LED_1, LOW); // turn the LED off by making the voltage LOW
16   delay(1000); // wait for a second
17 }
```

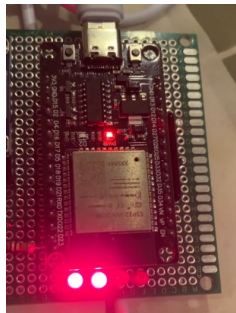
รูปที่ 1 โปรแกรมทดสอบ LED ที่อยู่ภายนอก

2. ให้เปลี่ยนโปรแกรมทดสอบ LED_2 – LED_4 ว่าทำงานถูกต้องหรือไม่ โดยทดลองที่ละ GPIO ถ้า LED ติดไม่ครบต้องแก้ไขให้เรียบร้อยก่อนจึงจะทดลองในขั้นตอนต่อไปได้
3. ป้อนโปรแกรมตามรูปที่ 2 และสังเกตผลการทดลองว่า LED ทั้ง 4 ดวง เป็นอย่างไร

```
Arduino IDE 2.3.5
Sketch Tools Help
ESP32-WROOM-DA M...
b2-1.ino
1  #define LED_ON_BOARD 2
2  #define LED_1 25
3  #define LED_2 26
4  #define LED_3 27
5  #define LED_4 14
6  // กำหนดให้ LED บนบอร์ด ชื่อ LED
7  void setup() {
8  // กำหนดให้ GPIO ที่ต่อกับ LED เป็น
9  pinMode(LED_1, OUTPUT);
10 pinMode(LED_2, OUTPUT);
11 pinMode(LED_3, OUTPUT);
12 pinMode(LED_4, OUTPUT);
13 }
14 // the loop function runs over
15 void loop() {
16 digitalWrite(LED_1, HIGH);
17 digitalWrite(LED_2, HIGH);
18 digitalWrite(LED_3, HIGH);
19 digitalWrite(LED_4, HIGH);
20 delay(1000);
21 digitalWrite(LED_1, LOW);
22 digitalWrite(LED_2, LOW);
23 digitalWrite(LED_3, LOW);
24 digitalWrite(LED_4, LOW);
25 delay(1000);
26 }
27
```

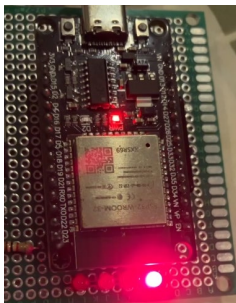
รูปที่ 2 โปรแกรมสำหรับควบคุม LED ผ่าน GPIO 14 25-27

4. ให้เขียนโปรแกรมให้ LED ทั้ง 4 ดวงทำงานตามคลิปต่อไปนี้ (Lab2-100)



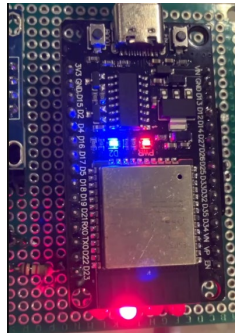
รูปที่ 3 คลิป Lab2-100

5. ให้เขียนโปรแกรมให้ LED ทั้ง 4 ดวงทำงานตามคลิปต่อไปนี้ (Lab2-101)



รูปที่ 4 คลิป Lab2-101

6. ให้เขียนโปรแกรมให้ LED ทั้ง 4 ดวงทำงานตามคลิปต่อไปนี้ (Lab2-102)



รูปที่ 5 คลิป Lab2-102

การทดลองที่ 2.2 Buzzer

Buzzer ถูกต่อเข้ากับ ESP32 ผ่าน GPIO23 ตามรูปที่ 1 โดยถ้าต้องการให้ Buzzer มีเสียง ที่ GPIO23 จะต้องส่งลอจิก “High” ถ้าต้องการให้ Buzzer ไม่มีเสียงต้องส่งลอจิก “LOW”

ขั้นตอนการทดลอง

1. ป้อนโปรแกรมตามรูปที่ 6 และสังเกตผลว่า Buzzer เป็นอย่างไร

```
lab2_200 | Arduino IDE 2.3.5
File Edit Sketch Tools Help

ESP32-WROOM-DA M...

lab2_200.ino
1  #define buzzer 23
2  void setup() {
3      // initialize digital pin buzzer as an output.
4      pinMode(buzzer, OUTPUT);
5  }
6
7  // the loop function runs over and over again forever
8  void loop() {
9      digitalWrite(buzzer, LOW); // turn the buzzer off by making the voltage LOW
10     delay(100);
11     digitalWrite(buzzer, HIGH); // turn the buzzer on by making the voltage HIGH
12     delay(4); // ปรับความดัง
13     digitalWrite(buzzer, LOW); // turn the buzzer off by making the voltage LOW
14     delay(100);
15 }
```

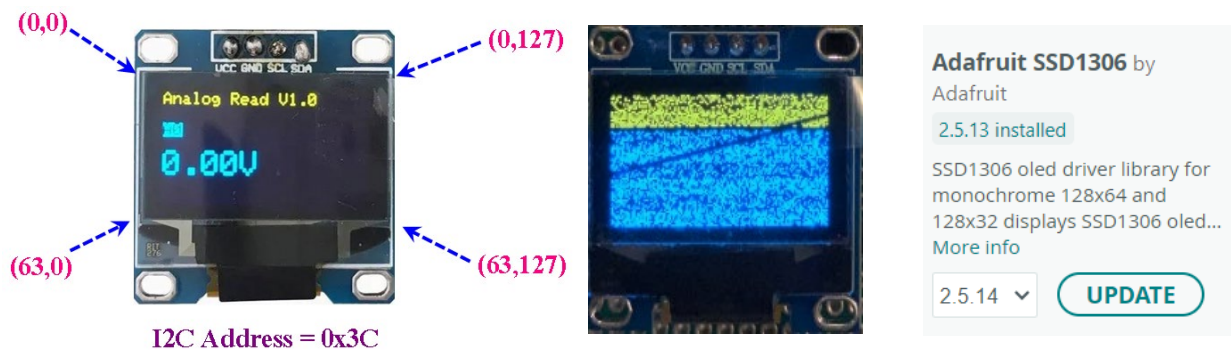
รูปที่ 6 โปรแกรมควบคุม Buzzer

2. ให้เขียนโปรแกรมกำหนดให้ Buzzer ดังเร็ว 3 ครั้งสลับกับเงียบ
3. ให้เขียนโปรแกรมให้ LED ทำงานตามคลิปที่ LAB2-102 พร้อมให้ Buzzer มีเสียงขณะที่ LED เปลี่ยน Step

การทดลองที่ 3

OLED 128 x 64 Display

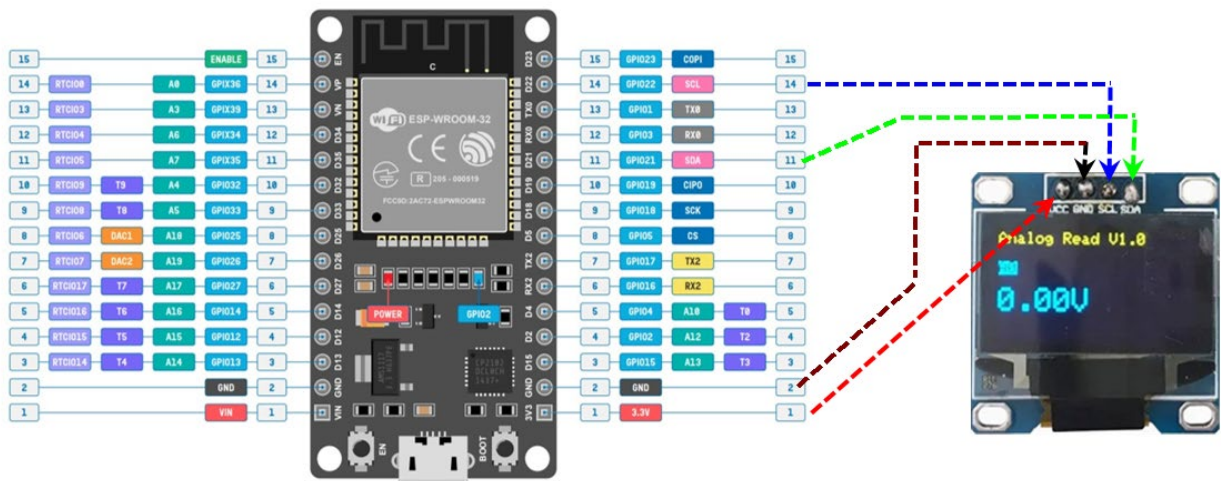
โมดูลที่ใช้ในการทดลองนี้เป็น OLED มีขนาด 0.96 นิ้ว ความละเอียด 128 x 64 จุด มีการเชื่อมต่อเป็นแบบ I2C โดยมีตำแหน่งอยู่ที่ 0x3C สามารถแสดงผลได้ 2 สีคือสีฟ้าและสีเหลือง โดยสีเหลืองจะอยู่บน สีฟ้าจะอยู่ด้านล่าง สำหรับ Library ที่ใช้ประกอบการเขียนโปรแกรมคือ Adafruit SSD1306 (สามารถ Down load Library Manager) แสดงดังรูปที่ 1



รูปที่ 1 OLED Adafruit SSD1306

ขั้นตอนการทดลอง

1. ต่อโมดูล OLED 128 x 64 Display เข้ากับบอร์ด ESP32 ผ่านทาง Port I2C ตามรูปที่ 1



รูปที่ 1 การต่อ OLED Display เข้ากับ ESP32 ผ่าน I2C

2. ติดตั้ง Library Adafruit SSD1306 ผ่าน Library Manager

3. Upload โปรแกรมสำหรับการทดสอบการเชื่อมต่อ I2C ตามรูปที่ 2 ถ้า ESP32 พบ OLED Display จะแสดงข้อความตามรูปที่ 3

```
1 // I2C Scanner
2 #include <Wire.h>
3 void setup() {
4     Serial.begin (115200);
5     // Leonardo: wait for serial port to connect
6     while (!Serial)
7     {
8     }
9     Serial.println ();
10    Serial.println ("I2C scanner. Scanning ...");
11    byte count = 0;
12    Wire.begin();
13    for (byte i = 8; i < 120; i++)
14    {
15        Wire.beginTransmission (i);
16        if (Wire.endTransmission () == 0)
17        {
18            Serial.print ("Found address: ");
19            Serial.print (i, DEC);
20            Serial.print (" (0x");
21            Serial.print (i, HEX);
22            Serial.println (")");
23            count++;
24            delay (1); // maybe unneeded?
25        } // end of good response
26    } // end of for loop
27    Serial.println ("Done.");
28    Serial.print ("Found ");
29    Serial.print (count, DEC);
30    Serial.println (" device(s).");
31 } // end of setup
32 void loop() {}
```

รูปที่ 2 โปรแกรมสำหรับค้นหาอุปกรณ์ I2C ที่เชื่อมต่อกับ ESP32

```
I2C scanner. Scanning ...
Found address: 60 (0x3C)
Done.
Found 1 device(s).
```

รูปที่ 3 ผลการค้นพบอุปกรณ์ I2C ที่ตำแหน่ง 0x3C

4. ป้อนโปรแกรมตามรูปที่ 3 สังเกตผลการทดลองที่จอแสดงผล OLED

```
1 // Test Display DS_1306_001
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5
6 #define SCREEN_WIDTH 128 // OLED display width, in pixels
7 #define SCREEN_HEIGHT 64
8 #define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
9 #define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
10 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
11 unsigned char x1=24,x2=63;
12 void setup() {
13     Serial.begin(9600);
14
15     // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
16     if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
17         Serial.println(F("SSD1306 allocation failed"));
18         for(;;); // Don't proceed, loop forever
19     }
20 }
21 void loop() {
22     // put your main code here, to run repeatedly:
23     display.clearDisplay();
24     display.setTextColor(WHITE);
25     display.setTextSize(1);
26     display.setCursor(15,2);
27     display.print(" BME KMUTNB 2025");
28     display.setTextSize(2);
29     display.setCursor(1,22);
30     display.print("SpO2 = ");
31     display.print(x1);
32     display.setCursor(1,50);
33     display.print("HR = ");
34     display.print(x2);
35     display.display();
36 }
```

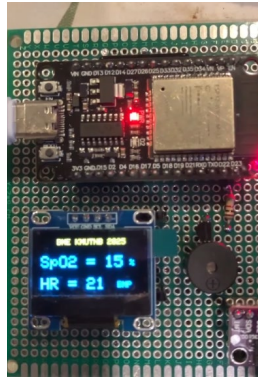
รูปที่ 3 โปรแกรมทดสอบ OLED แบบที่ 1

5. เพิ่มเติมโปรแกรมในรูปที่ 3 ด้วยรูปที่ 4 ตั้งแต่บรรทัดที่ 36-39 ทำการ Upload สังเกตผลที่ OLED

```
30     display.print("SpO2 = ");
31     display.print(x1);
32     display.setCursor(1,50);
33     display.print("HR = ");
34     display.print(x2);
35     display.display();
36     delay(500);
37     display.clearDisplay();
38     display.display();
39     delay(500);
40 }
```

รูปที่ 4 ส่วนเพิ่มเติมโปรแกรมทดสอบ OLED

6. ให้เขียนโปรแกรมให้ OLED แสดงผลพร้อม LED และ Buzzer ตามคลิปต่อไปนี้

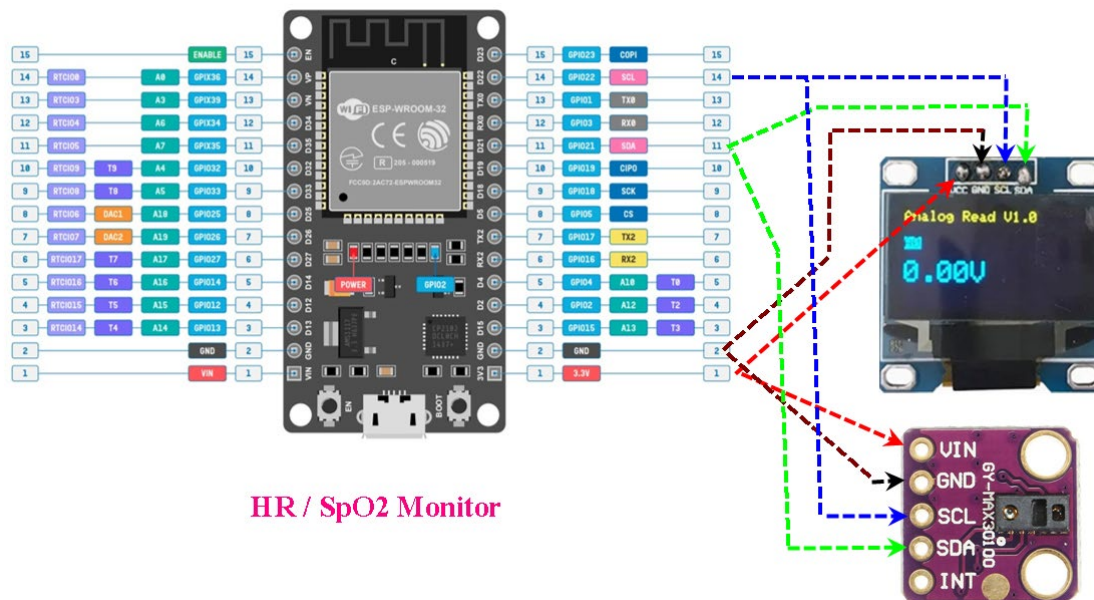


การทดลองที่ 4

MAX30100 Pulse Oximeter and Heart-Rate Sensor

ขั้นตอนการทดลอง

1. ต่อโมดูล OLED และ โมดูล MAX30100 เข้ากับบอร์ด ESP32 ตามรูปที่ 1



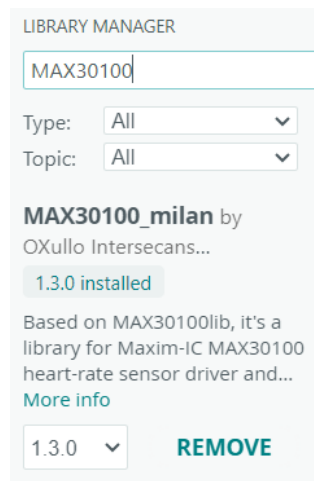
รูปที่ 1 การเชื่อมต่อ ESP32 เข้ากับ OLED และ Module MAX30100 (SpO2/HR)

2. Upload โปรแกรมสำหรับการทดสอบการเชื่อมต่อ I2C ตามรูปที่ 2 จากการทดลองที่ 3 ถ้า ESP32 พบ OLED Display และ Module MAX30100 จะแสดงข้อความตามรูปที่ 2

```
I2C scanner. Scanning ...  
Found address: 60 (0x3C)  
Found address: 87 (0x57)  
Done.  
Found 2 device(s).
```

รูปที่ 2 ผลการค้นพบอุปกรณ์ I2C ที่ตำแหน่ง 0x3C (OLED) และ 0x57 (MAX30100)

3. ติดตั้ง Library สำหรับ Module MAX30100 ผ่าน Library Manager ตามรูปที่ 3



รูปที่ 3 Library สำหรับ Module MAX30100

4. เปิดโปรแกรมตัวอย่างโดยไปที่ File \Rightarrow Examples \Rightarrow MAX30100_milan \Rightarrow MAX30100_Minimal

5. Upload โปรแกรม MAX30100_Minimal ให้ใช้รีเซ็ตบนตัว Sensor MAX30100 สังเกตผลการทดลองที่ Serial Monitor ตามรูปที่ 4

```
Heart rate:36.27bpm / SpO2:0%
Beat!
Heart rate:58.76bpm / SpO2:96%
Beat!
Heart rate:50.41bpm / SpO2:96%
Beat!
Heart rate:69.37bpm / SpO2:96%
Beat!
Beat!
Heart rate:86.40bpm / SpO2:96%
Beat!
Beat!
Heart rate:78.25bpm / SpO2:94%
```

รูปที่ 4 การแสดงค่า SpO2 / HR บน Serial Monitor

6. ป้อนโปรแกรมตามรูปที่ 5 เพื่อนำค่า SpO2 / HR จาก Module MAX30100 ไปแสดงผลที่ OLED

7. ให้ใช้รีเซ็ตบนตัว Sensor MAX30100 สังเกตผลการทดลองที่ OLED และฟังเสียงที่ Buzzer

```

1 // Test Display DS_1306_002
2 #include <Wire.h>
3 #include <Adafruit_GFX.h>
4 #include <Adafruit_SSD1306.h>
5 #include "MAX30100_PulseOximeter.h"
6
7 #define REPORTING_PERIOD_MS    500
8
9 #define SCREEN_WIDTH 128 // OLED display width, in pixels
10 #define SCREEN_HEIGHT 64
11 #define OLED_RESET     -1 // Reset pin # (or -1 if sharing Arduino reset pin)
12 #define SCREEN_ADDRESS 0x3C ///< See datasheet for Address; 0x3D for 128x64, 0x3C for 128x32
13 PulseOximeter pox;
14 uint32_t tsLastReport = 0;
15 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);
16 float x1,x2;
17 unsigned char x3,x4;
18
19 void onBeatDetected()
20 {
21   Serial.println("Beat!");
22 }
23
24 void setup() {
25   Serial.begin(115200);
26   Serial.print("Initializing pulse oximeter..");
27   pinMode(23, OUTPUT);
28   pinMode(14, OUTPUT);
29   pinMode(25, OUTPUT);
30   pinMode(26, OUTPUT);
31   pinMode(27, OUTPUT);
32   // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
33   if(!display.begin(SSD1306_SWITCHCAPVCC, SCREEN_ADDRESS)) {
34     Serial.println(F("SSD1306 allocation failed"));
35     for(;;); // Don't proceed, loop forever
36   }
37   display.setTextColor(WHITE);
38
39   if (!pox.begin()) {
40     Serial.println("FAILED");
41     for(;;);
42   } else {
43     Serial.println("SUCCESS");
44   }
45   //pox.setOnBeatDetectedCallback(onBeatDetected);
46   pox.setOnBeatDetectedCallback(beep001);
47 }
48

```

รูปที่ 5 โปรแกรมอ่านค่า SpO2 / HR จาก Module MAX30100 แสดงผลที่ OLED


```

49
50 void loop() {
51     // put your main code here, to run repeatedly:
52     pox.update();
53     display.clearDisplay();
54     yellow_display();
55     if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
56         Serial.print("Heart rate:");
57         x1 = pox.getSpO2();
58         x2 = pox.getHeartRate();
59         x3 = x1;
60         x4 = x2;
61         Serial.print(x2);
62         Serial.print("bpm / SpO2:");
63         Serial.print(x1);
64         Serial.println("%");
65         display.setTextSize(2);
66         display.setCursor(0,22);
67         display.print("SpO2: ");
68         display.print(x3);
69         display.setTextSize(1);
70         display.setCursor(110,27);
71         display.print(" %");
72         display.setTextSize(2);
73         display.setCursor(0,50);
74         display.print("HR: ");
75         display.print(x4);
76         display.setTextSize(1);
77         display.setCursor(100,55);
78         display.print("BMP");
79         display.display();
80         tsLastReport = millis();
81     }
82 }
83
84 void yellow_display()
85 {
86     display.setTextSize(1);
87     display.setCursor(15,2);
88     display.print(" BME KMUTNB 2025");
89 }
90 void beep001()
91 {
92
93     digitalWrite(23, LOW);
94     digitalWrite(25, LOW);
95     delay(2);
96     digitalWrite(23, HIGH);
97     digitalWrite(25, HIGH);
98     delay(2);
99     digitalWrite(23, LOW);
100    digitalWrite(25, LOW);
101
102 }

```

รูปที่ 5 โปรแกรมอ่านค่า SpO2 / HR จาก Module MAX30100 แสดงผลที่ OLED (ต่อ)