Week 5

7-Segment I2C
(HT16K33)

1.การทำงานรูปแบบ I2C คืออะไร?

รูปแบบการสื่อสารแบบอนุกรม (Serial Communication) ที่ถูกออกแบบมาเพื่อใช้เชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์ในระยะใกล้บนแผงวงจรเดียวกัน โดยใช้สายสัญญาณเพียง 2 เส้น

หลักการทำงาน: Master และ Slave

Master (ตัวควบคุม): คือไมโครคอนโทรลเลอร์ (เช่น Arduino หรือ ESP32) ที่เป็นผู้เริ่มต้นการสื่อสารและสั่งงาน

SDA (Serial Data Line) and SCL (Serial Clock Line)

สายสัญญาณ 2 เส้นสำคัญ

SDA (Serial Data Line): สายสำหรับส่งและรับข้อมูล (Data)

SCL (Serial Clock Line): สายสำหรับส่ง สัญญาณนาฬิกา (Clock) เพื่อควบคุมจังหวะ (Synchronization) ในการส่งข้อมูล

2. Libraries ที่ใช้งานกับ 7-Segment แบบ I2C

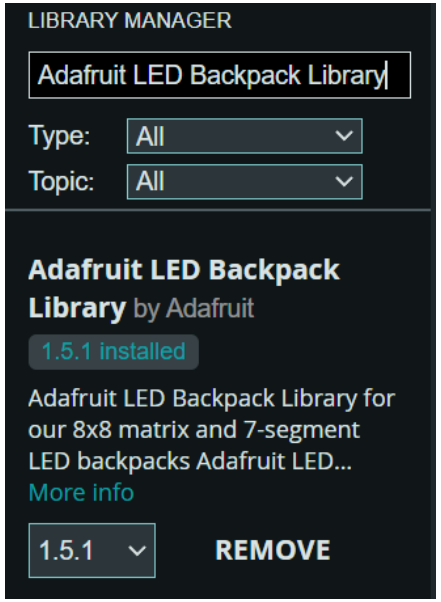
#include <Wire.h> ไลบรารีพื้นฐานของ Arduino ESP32 ที่จำเป็นสำหรับการจัดการสาย SDA/SCL เพื่อส่งข้อมูล I2C

2. Libraries ที่ใช้งานกับ 7-Segment แบบ I2C

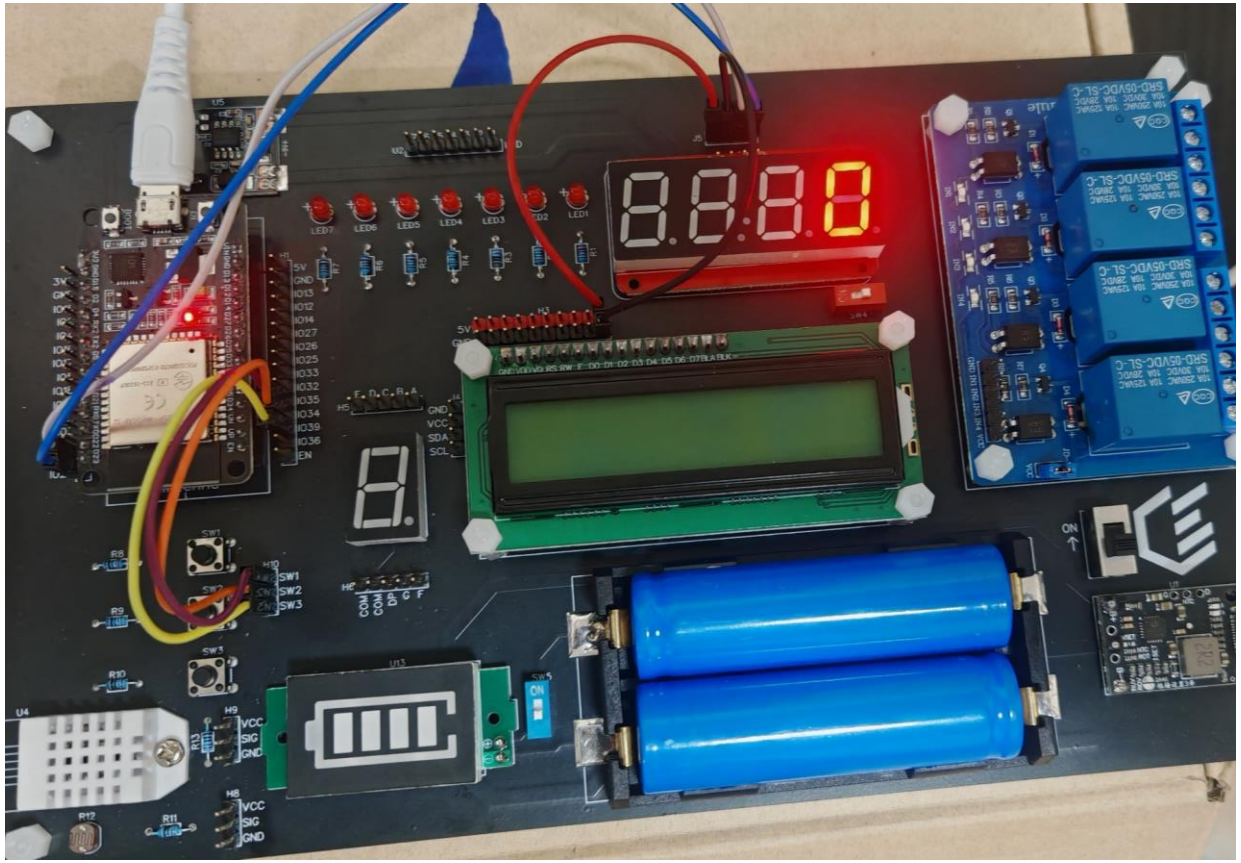
#include <Adafruit_GFX.h> ไลบรารีพื้นฐานสำหรับงานกราฟิก (เช่น การจัดการพิกเซลและตำแหน่ง) ที่ไลบรารีของ Adafruit ส่วนใหญ่นำไปใช้ต่อยอด

2. Libraries ที่ใช้งานกับ 7-Segment แบบ I2C

#include <Adafruit_LEDBackpack.h> ไลบรารีเฉพาะที่ทำหน้าที่แปลงตัวเลขหรือตัวอักษรให้เป็นรูปแบบที่ชิป HT16K33 เข้าใจ จากนั้นใช้ฟังก์ชัน I2C ใน Wire.h เพื่อส่งคำสั่งนั้นไปแสดงผลบนจอ 7-Segment



การต่อวงจร ESP32 7-Segment I2C



7-S ➔ ESP32

VCC ➔ 5V

GND ➔ GND

SCL ➔ 22

SDA ➔ 23

การเขียนโปรแกรม

#include <Wire.h> // ไลบรารีสำหรับสื่อสาร I2C

#include <Adafruit_GFX.h> // ไลบรารีกราฟิกพื้นฐาน

#include <Adafruit_LEDBackpack.h> // ไลบรารีสำหรับโมดูล 7-seg HT16K33

```
#include <Wire.h>                // ไลบรารีสำหรับสื่อสาร I2C
#include <Adafruit_GFX.h>        // ไลบรารีกราฟิกพื้นฐาน
#include <Adafruit_LEDBackpack.h> // ไลบรารีสำหรับโมดูล 7-seg HT16K33
```

การเขียนโปรแกรม

```
Adafruit_7segment matrix = Adafruit_7segment();
```

Adafruit_7segment = ชื่อคลาส

matrix = ชื่อตัวแปรที่สร้างไว้ใช้งาน

= Adafruit_7segment() = เรียก Constructor เพื่อสร้างออบเจกต์

การเขียนโปรแกรม

```
#include <Wire.h>                   // ไลบรารีสำหรับสื่อสาร I2C
#include <Adafruit_GFX.h>           // ไลบรารีกราฟิกพื้นฐาน
#include <Adafruit_LEDBackpack.h> // ไลบรารีสำหรับโมดูล 7-seg HT16K33
// ---------------- 7-Segment I2C HT16K33 ----------------
Adafruit_7segment matrix = Adafruit_7segment();

// ---------------- ตัวแปรควบคุม ----------------
int currentNum = 0;    // เลขที่รับจาก Serial
void showNumberI2C(int num) {
  matrix.clear();
  matrix.print(num);      // แสดงตัวเลข (0-9999)
  matrix.writeDisplay();
}
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  // เริ่มต้นใช้งาน I2C สำหรับ HT16K33
  Wire.begin(22, 23);    // กำหนด SDA=22, SCL=23
  matrix.begin(0x70);    // ที่อยู่ I2C ปกติคือ 0x70
  // เริ่มต้นแสดงเลข 0
  showNumberI2C(999);
}

void loop() {
  // put your main code here, to run repeatedly:

}
```

การรับค่าจากคีย์บอร์ด

```
void setup() {
  Serial.begin(115200);   // เริ่ม Serial
  Serial.println("กรอกตัวเลขแล้วกด Enter:");
}

void loop() {
  // ถ้ามีข้อมูลจากคีย์บอร์ดเข้ามา
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n'); // อ่านจนจบแถว
    input.trim();                                // เอา space / \r ออก

    // แปลงเป็นตัวเลข (int)
    int value = input.toInt();

    // แสดงผลกลับไปที่คอมพิวเตอร์
    Serial.print("คุณส่งตัวเลข: ");
    Serial.println(value);
  }
}
```

การรับค่าจากคีย์บอร์ด + คำนวณผล

```
void setup() {
  Serial.begin(115200);
  Serial.println("กรอกตัวเลข (0-9999) แล้วกด Enter:");
}

void loop() {
  if (Serial.available() > 0) {
    String input = Serial.readStringUntil('\n');
    input.trim();

    int num = input.toInt();

    // ---- แยกหลัก ----
    int thousands = (num / 1000);   // หลักพัน
    int hundreds  = (num / 100) % 10;    // หลักร้อย
    int tens      = (num / 10) % 10;     // หลักสิบ
    int ones      = num % 10;            // หลักหน่วย

    // ---- แสดงผล ----
    Serial.println("==== ผลการแยกหลัก ====");
    Serial.print("เลขที่รับเข้า: ");
    Serial.println(num);

    Serial.print("หลักพัน : "); Serial.println(thousands);
    Serial.print("หลักร้อย: "); Serial.println(hundreds);
    Serial.print("หลักสิบ : "); Serial.println(tens);
    Serial.print("หลักหน่วย: "); Serial.println(ones);
    Serial.println("====================");
  }
}
```