[redacted]

ESP32 Smart Environment Monitor

ระบบตรวจวัดและแจ้งเตือนสภาพแวดล้อมแบบอัจฉริยะ
พร้อม 3 โหมดการทำงานที่หลากหลาย

สมาชิกกลุ่ม

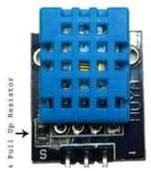
นาย รัชพล ลองซุม2311310862

อภิโชค ม่วงกล่ำ 2311310417

นรภัทร จั้นเอียบ 2311340083

ภาพรวมระบบ
Hardware component

Input



DHT11

Button switch

PC keyboard

Process

SSD1306

Interrupt

Timer

ภาพรวมระบบ
Hardware component

Output

OLED

LED

Project detail / requirement

ระบบตรวจสอบอุณหภูมิและความชื้นแบบเรียลไทม์ด้วยเซนเซอร์ DHT11
มีจอ OLED แสดงผลสถานะและข้อความแจ้งเตือน
รองรับการทำงาน 3 โหมดหลัก
Mode 1 : Live Monitor – แสดงค่า Temp/Humid ต่อเนื่อง
Mode 2 : Alert Mode – แจ้งเตือนเมื่อเกินเกณฑ์ (Temp > 30°C, Humid > 80%) ด้วย LED และข้อความ
Mode 3 : Quiz Game – เกมคิดเลขเร็ว 30 วินาที/รอบ ตอบใน Serial Monitor พร้อมเก็บคะแนนและแสดงประวัติ 3 รอบล่าสุด
มีปุ่มกดเปลี่ยนโหมด และกดค้างเพื่อเริ่มรอบใหม่
เหมาะกับการใช้งานเพื่อการเรียนรู้ IoT, Microcontroller และการออกแบบระบบ Embedded

Project specification

ฮาร์ดแวร์
บอร์ด ESP32 DOIT Devkit V1
เซนเซอร์ DHT11 สำหรับวัดอุณหภูมิและความชื้น
จอ OLED SSD1306 ขนาด 128x64 (I2C)
LED 2 ดวง (แดง/เขียว)
Push button 1 ปุ่ม
ซอฟต์แวร์
ใช้ Arduino Framework ผ่าน PlatformIO
Libraries: Adafruit GFX, Adafruit SSD1306, DHT sensor
การทำงาน
อ่านค่าเซนเซอร์ทุก 1 วินาที
Mode 1: แสดง Temp/Humid บน OLED และ LED เขียวติดค้าง
Mode 2: ถ้า Temp > 30°C หรือ Humid > 80% → LED แดงกระพริบทุก 200 ms และ OLED แสดง “HIGH TEMP/HUMID”
Mode 3: เริ่มเกมเมื่อกดปุ่มค้าง ~0.8s → มีเวลา 30 วินาที ตอบโจทย์บวกเลขทาง Serial Monitor, แสดงคะแนนและประวัติ 3 รอบล่าสุด
กดปุ่มสั้น = เปลี่ยนโหมด

สถาปัตยกรรมระบบ

1

Data Collection

DHT11 อ่านค่าอุณหภูมิและความชื้นทุกวินาที

2

Processing

- ESP32 ตรวจสอบเงื่อนไข (Temp > 30°C หรือ Humid > 80%)
- จัดการโหมดการทำงาน (Live, Alert, Quiz)

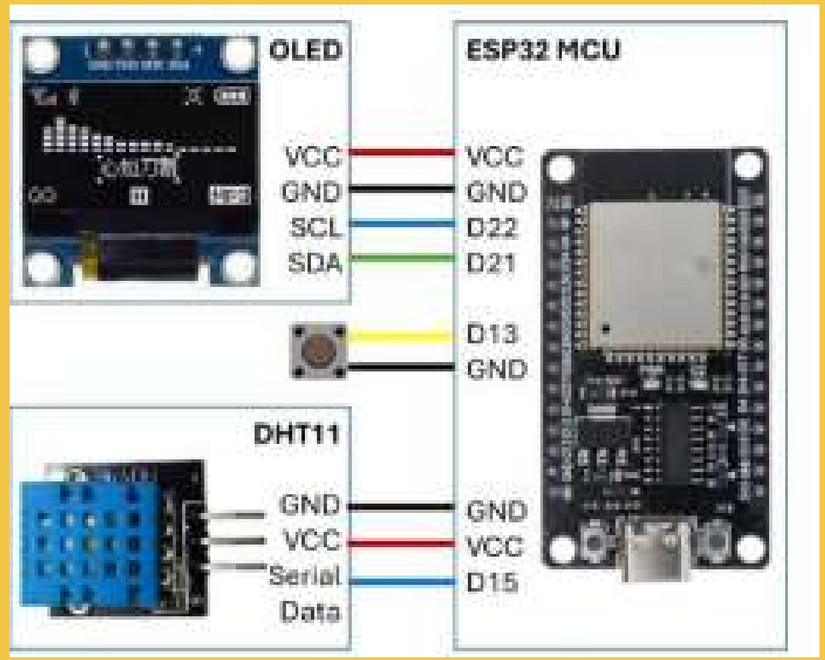
3

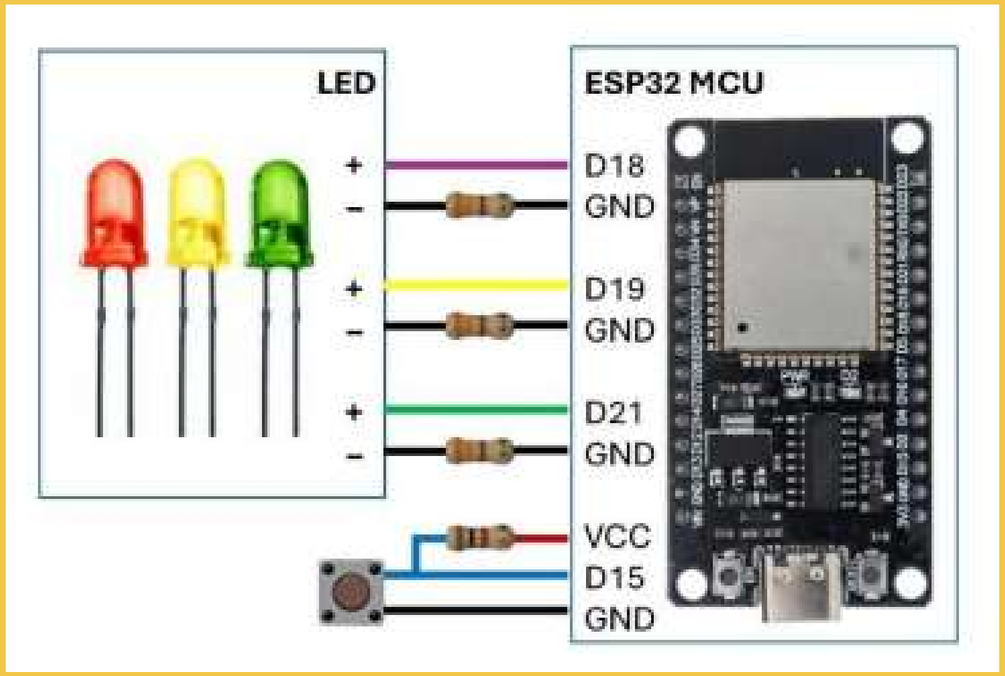
Display & Alert

- แสดงผลบน OLED (ค่าเซนเซอร์/ข้อความ/โจทย์ Quiz)
- ควบคุม LED (เขียวติดปกติ / แดงกระพริบแจ้งเตือน / ใช้ใน Quiz)
- เก็บประวัติคะแนน 3 รอบล่าสุด (Mode 3)

การออกแบบโดยละเอียด

วงจร





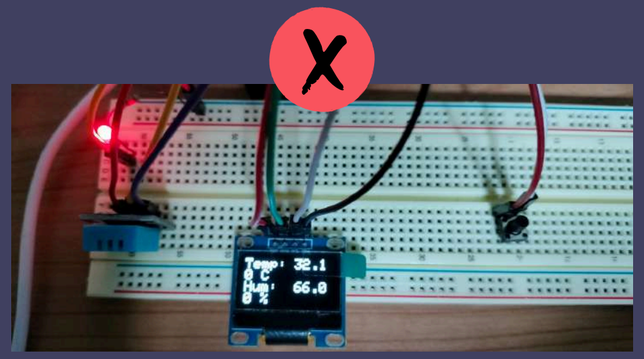
แบ่งเป็น 3 mode

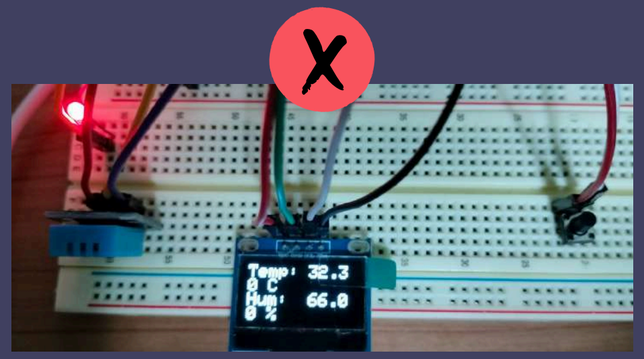
Mode 1

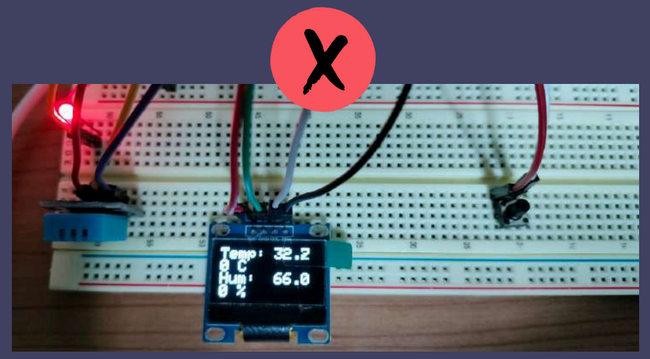
Real-time Monitor

โหมดแสดงผลต่อเนื่อง

แสดงค่าอุณหภูมิและความชื้นจาก DHT11 แบบเรียลไทม์ โดยใช้ Timer อ่านข้อมูลทุก 1 วินาที







Mode 1 : โค้ด

```
void runMode1() {
  // อ่านค่า DHT ทุก 1 วินาที
  if (millis() - tRead >= 1000) {
    tRead = millis();
    readDHT();
    showLive();   // แสดงผลเรียลไทม์
  }

  // สภาวะปกติ → LED เขียวติดค้าง, แดงดับ
  digitalWrite(LED_GREEN, HIGH);
  digitalWrite(LED_RED, LOW);
}
```

```
void showLive() {
  drawHeader("LIVE");

  display.setTextSize(2);
  display.setCursor(0, 16);
  display.print("T:");
  if (isnan(tempC)) display.print("--");
  else display.print((int)round(tempC));
  display.print((char)247); // °C
  display.print("C");

  display.setCursor(0, 40);
  display.print("H:");
  if (isnan(humi)) display.print("--");
  else display.print((int)round(humi));
  display.print("%");
```

```
// แสดงการ์ตูน: ขยับเมื่อไม่สบายตัว
drawMascotAnimated(comfortable(tempC, humi));
drawFooterHint();
display.display();
}
```

Mode 1: วิธีการทำงาน

Timer-based Reading

ใช้ Timer อ่านค่าจาก DHT11 ทุก 1,000 มิลลิวินาที

Display Update

แสดงผลบนจอ OLED รูปแบบ "Temp: 29°C, Hum: 60%"

Status LED

LED ติดค้างแสดงสถานะการทำงานปกติ มีตัวการ์ตูนเล็กๆบ่งบอกถึงอารมณ์ต่อสภาพอากาศ

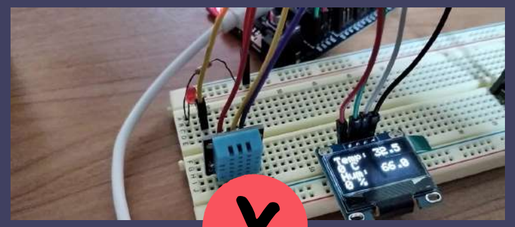
Mode 2

Alert Mode

โหมดแจ้งเตือนอัตโนมัติ

ตรวจจับค่าผิดปกติและแจ้งเตือนทันที เป็น Active monitoring ที่มีเงื่อนไขการแจ้งเตือน

*บ้านคนถ่ายอุณหภูมิต่ำสุดอยู่ที่ 31 องศา



เงื่อนไขการแจ้งเตือน
อุณหภูมิ > 30°C หรือ
ความชื้น > 80%

การแสดงผล
LED กระพริบเร็ว + ข้อความ
"HIGH Tem" หรือ "HIGH Humid"

Mode 2 : โค้ด

```
// --------- ALERT thresholds ---------
const float TEMP_HIGH = 30.0;   // อุณหภูมิสูงสุด
const float HUMI_HIGH = 80.0;   // ความชื้นสูงสุด

void runMode2() {
  if (millis() - tRead >= 1000) {   // อ่านค่าทุก 1 วินาที
    tRead = millis();
    readDHT();
  }

  bool highT = (!isnan(tempC) && tempC > TEMP_HIGH);
  bool highH = (!isnan(humi)  && humi  > HUMI_HIGH);
  bool alert = highT || highH;

  if (alert) {
    if (millis() - tBlink >= 200) {   // LED แดงกระพริบเร็ว
      tBlink = millis();
      ledFlip = !ledFlip;
      digitalWrite(LED_RED, ledFlip);
    }
    digitalWrite(LED_GREEN, LOW);
  } else {
    digitalWrite(LED_GREEN, HIGH);
    digitalWrite(LED_RED, LOW);
  }

  showAlert(highT, highH);  // แสดงผลบนจอ OLED
}
```

Mode 2 : การตรวจสอบเงื่อนไข

```
if (temperature > 30.0) {
  display.println("HIGH TEMP!");
  alertActive = true;
} else if (humidity > 80.0) {
  display.println("HIGH HUMID!");
  alertActive = true;
}
```

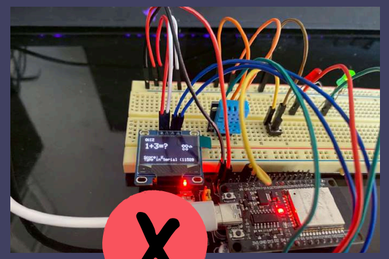
ใช้ Interrupt กับปุ่มกดเพื่อหยุดการแจ้งเตือนหรือเปลี่ยนโหมด

Mode 3

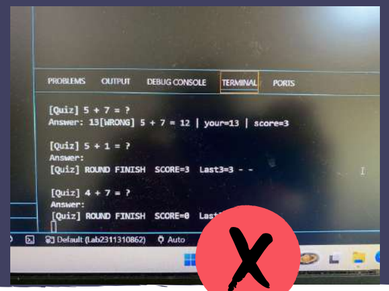
Quiz Game

โหมดตอบคำถามบวกเลข

โหมดตอบคำถามบวกเลข (0-9) รอบละ 30 วินาที
ผู้ใช้พิมพ์คำตอบใน Serial Monitor
ระบบตรวจคำตอบ → ถูกเพิ่มคะแนน, ผิดกระพริบไฟแดง
เมื่อหมดเวลา จอจะแสดง คะแนนรวมของรอบ และ ประวัติ 3 รอบล่าสุด
ผู้ใช้ต้อง กดปุ่มค้าง (~0.8s) เพื่อเริ่มรอบใหม่เอง



ในหน้าจอมอนิเตอร์ของบอร์ดแสดงผลโจทย์และคะแนนรวม



การแสดงผล แสดงโจทย์ ละให้พิมคำตอบบนคีบอด

Mode 3 : วิธีการทำงาน

🔵 Setup / Start
กดปุ่มค้าง (~0.8s) เพื่อเริ่มรอบใหม่ (30 วินาที)
ระบบรีเซ็ตคะแนนรอบนั้นเป็น 0

🟠 Question & Answer
ระบบสุ่มโจทย์บวกเลขง่าย ๆ (0–9)
พิมพ์คำตอบลงใน Serial Monitor

🟣 Scoring
ตอบถูก → ได้คะแนน +1 และขึ้นติ๊กถูกบนจอ
ตอบผิด → ไฟ LED แดงกระพริบ

🟢 Round Finish
เมื่อครบเวลา 30 วินาที ระบบสรุปคะแนนรอบนั้น
แสดง ประวัติ 3 รอบล่าสุด (เช่น 5 3 7)
จอพักค้างรอผู้ใช้กดปุ่มเริ่มใหม่เอง

Mode 3 : โค้ด

```
// ------------------- QUIZ (Mode 3) -------------------
const uint32_t ROUND_MS = 30000;   // รอบละ 30 วินาที
bool roundRunning = false;         // สถานะรอบ
bool roundFinished = false;
unsigned long roundStart = 0;
int score = 0;
// ฟังก์ชันสร้างโจทย์ใหม่
void newQuestion() {
  aQ = random(0, 10);
  bQ = random(0, 10);
  ansQ = aQ + bQ;
  Serial.printf("[Quiz] %d + %d = ?\n", aQ, bQ);
}
```

```
// แสดงหน้าจอควิซ
void showQuizRunning(unsigned long remainMs) {
  drawHeader("QUIZ");
  display.setTextSize(2);
  display.setCursor(0, 14);
  display.printf("%d+%d=?", aQ, bQ);
  display.setTextSize(1);
  display.setCursor(0, 40);
  display.printf("Remain: %lu s", remainMs/1000);
  display.setCursor(0, 50);
  display.printf("Score: %d", score);
  display.display();
}
```

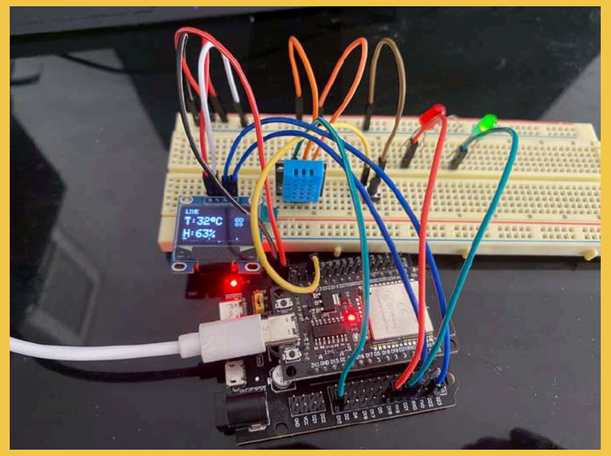
สรุปและประโยชน์ของโปรเจค

โหมด 1 (Live Monitor)
ประโยชน์: ตรวจวัดอุณหภูมิและความชื้นแบบเรียลไทม์ → ช่วยให้ผู้ใช้รู้สภาพแวดล้อมปัจจุบันทันที ใช้ได้กับบ้าน สำนักงาน หรือโรงเรือนที่ต้องเฝ้าระวังอากาศตลอดเวลา

โหมด 2 (Alert Mode)
ประโยชน์: แจ้งเตือนอัตโนมัติเมื่อเกินค่ากำหนด เช่น ร้อนเกิน 30°C หรือชื้นเกิน 80% ลดความเสี่ยง เช่น พืชเสียหาย เครื่องมือพัง หรือคนอยู่ในสภาพอากาศไม่เหมาะสม เหมาะสำหรับโรงเพาะปลูก ห้องเซิร์ฟเวอร์ หรือห้องเก็บของ

โหมด 3 (Quiz Mode)
ประโยชน์: เพิ่มการโต้ตอบ/ความสนุก → ใช้เป็นเกมการศึกษา (คิดเลขเร็ว) ช่วยให้ผู้ใช้ฝึกสมองไปพร้อม ๆ กับการใช้งานอุปกรณ์ มีการเก็บคะแนน และประวัติ 3 คนล่าสุด → เหมาะสำหรับแข่งขันกันในกลุ่มเล็ก ๆ

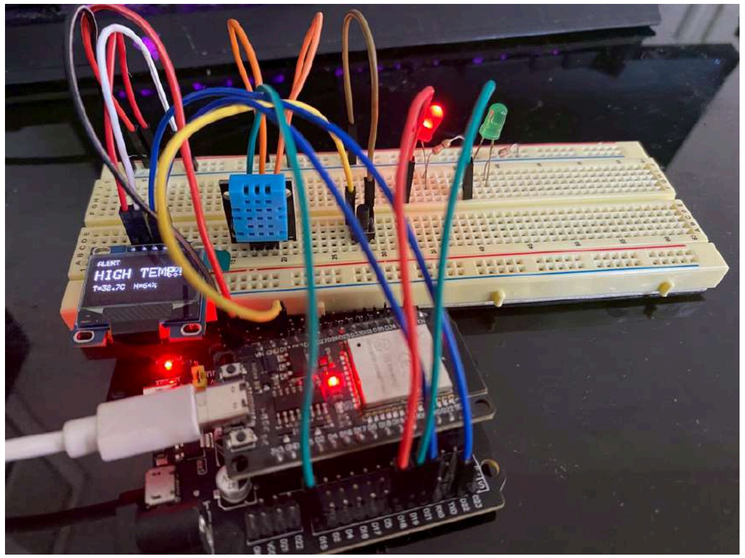
Picture of your project



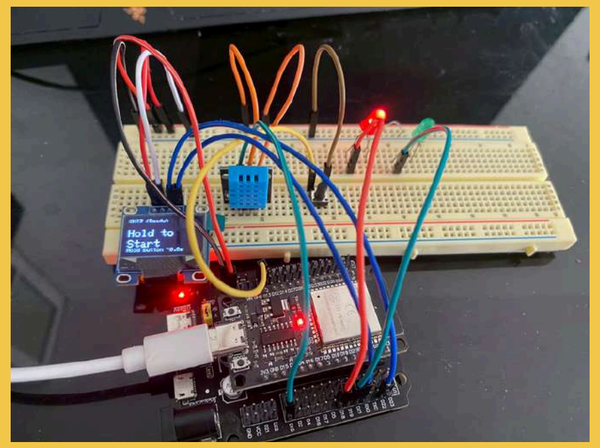
https://drive.google.com/drive/folders/1EJ-RjSEnCcKYMFGtikMLgeuCsUhLjqVN

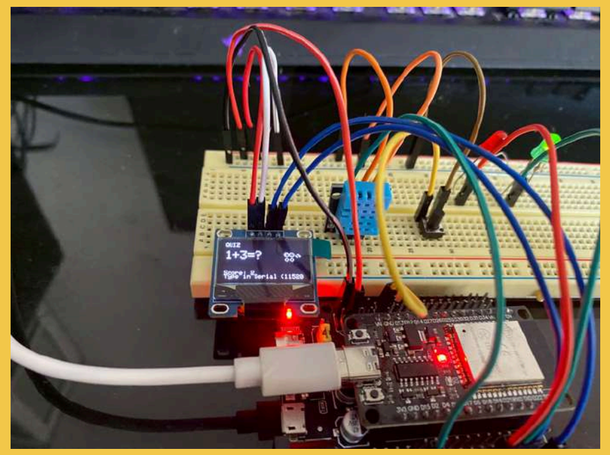
mode1

mode2

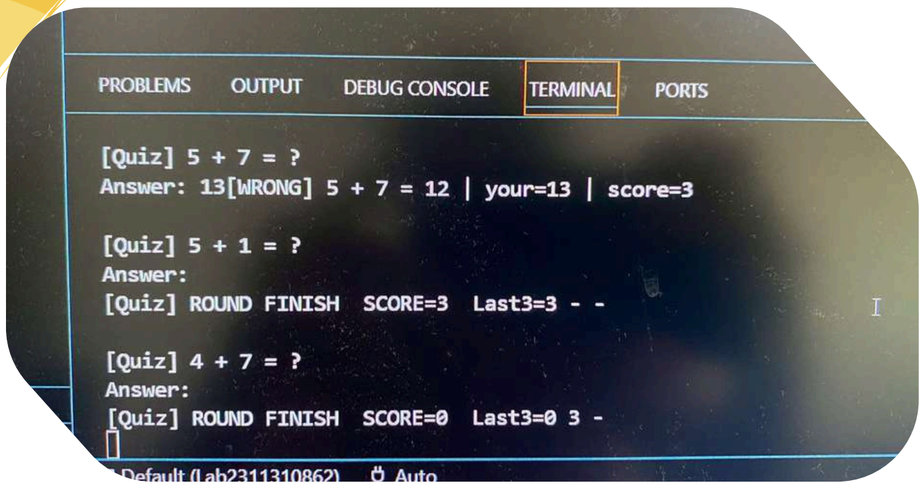


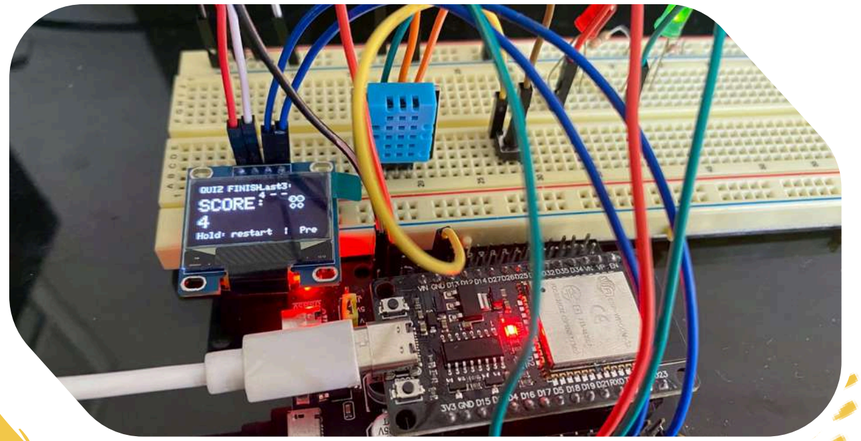
Picture of your project



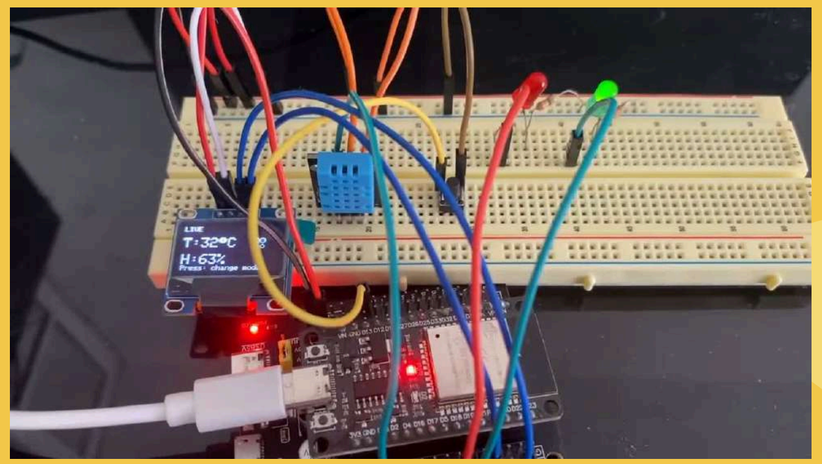


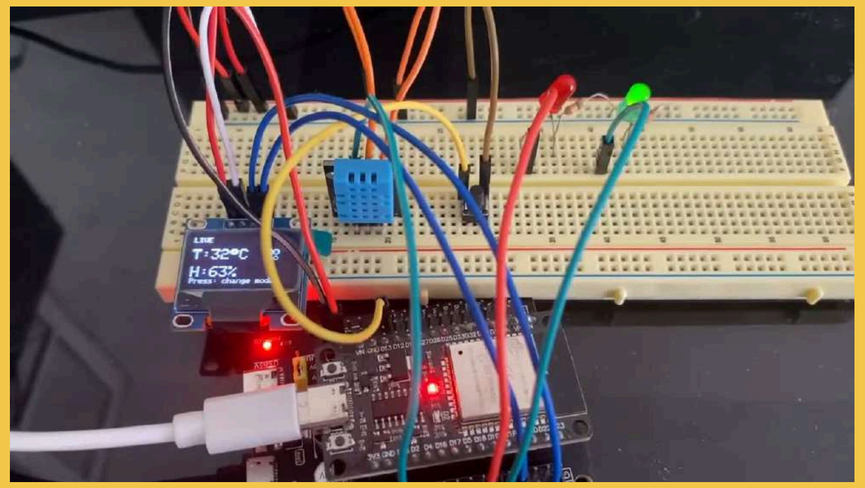
mode3

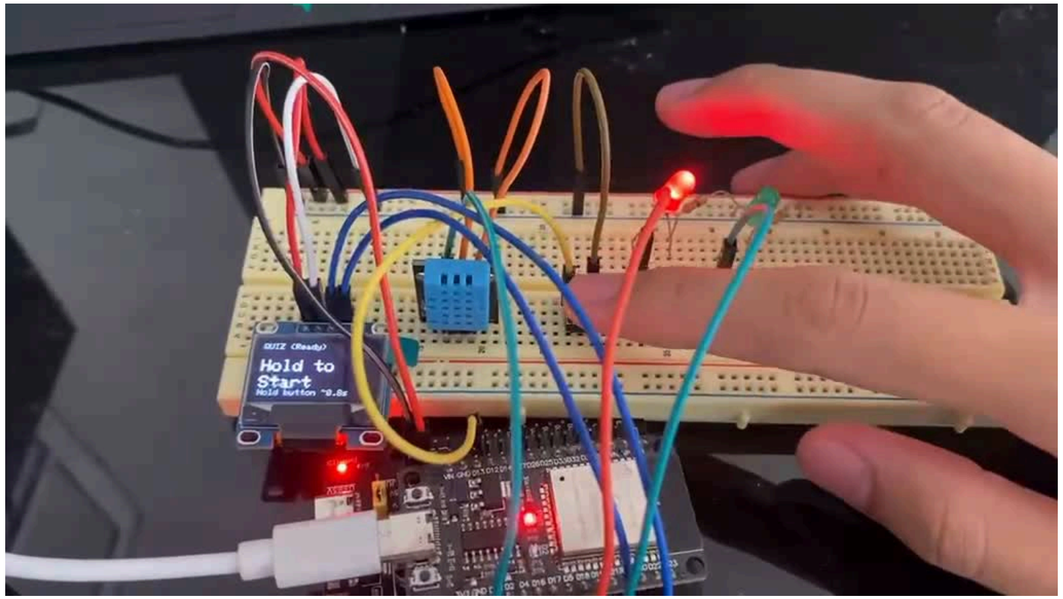




VDO demonstration







ปัญหาที่เจอ

-โค้ดมีความสับสนกันระหว่างการสรุปคะแนนรวม ที่ไม่ใช่คะแนนตอบถูกในแต่ละครั้ง

-ใช้ Github ครั้งแรก

-ต้องเพิ่มโค้ดใน ส่วนของ platformio ด้วยเพิ่ม การทำงานของโหมด3ในการรับค่าอินพุตจากคีบอด

Project gantt chart

	ผู้ดำเนินการ
-ในส่วนของการทำไลด์canva	/รัชพล ลองซุม,อภิโชค ม่วงกล่ำ,นรภัทร จั้นเอียบ
-ในส่วนของ code	/รัชพล ลองซุม,อภิโชค ม่วงกล่ำ,นรภัทร จั้นเอียบ
-githubการสร้างไฟล์	/รัชพล ลองซุม,
-ผู้ทดลองต่อบอดและทดสอบ	/รัชพล ลองซุม,อภิโชค ม่วงกล่ำ
-เช็คประเมินความคืบหน้า	/รัชพล ลองซุม