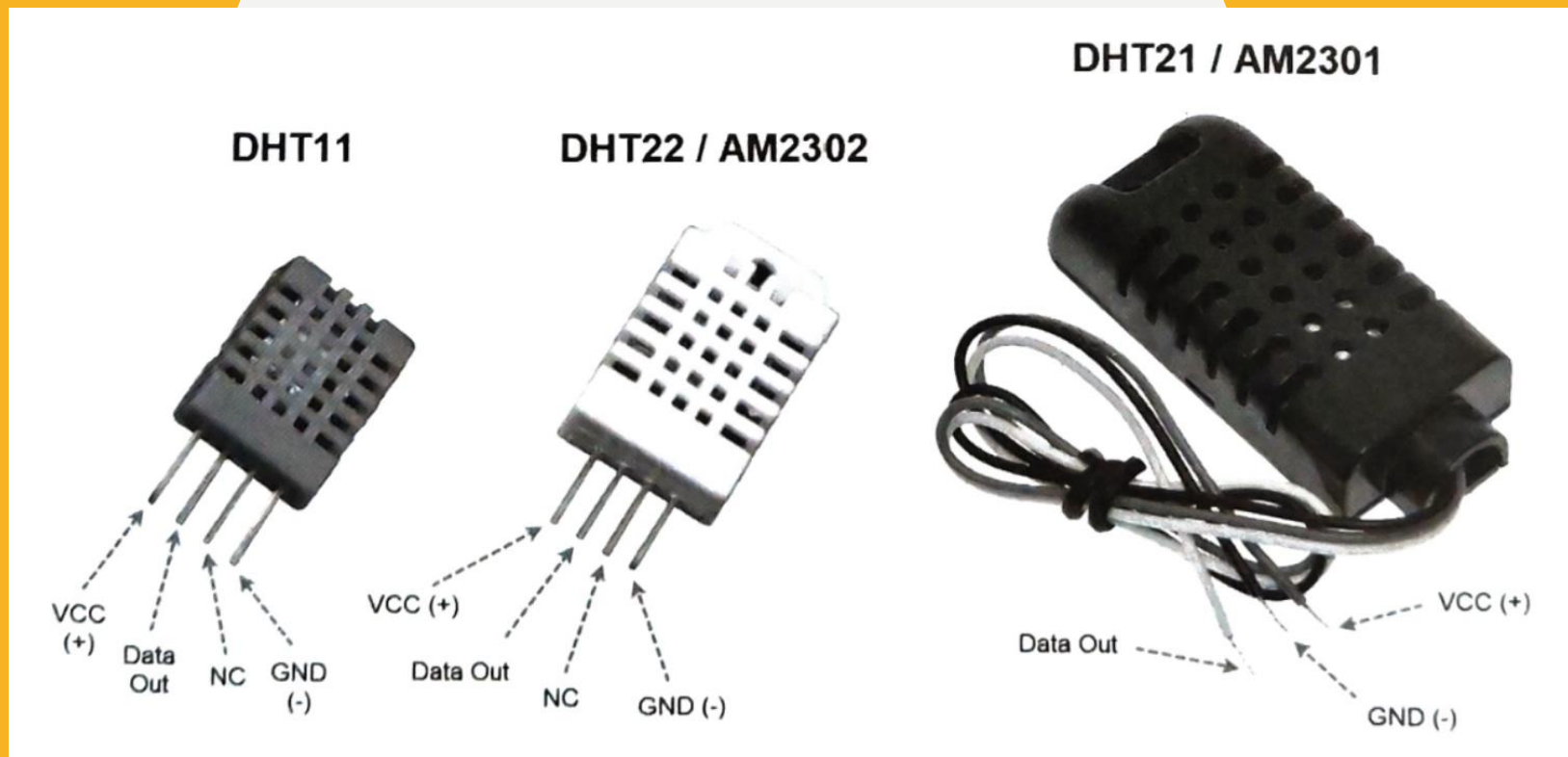


บทที่ 16

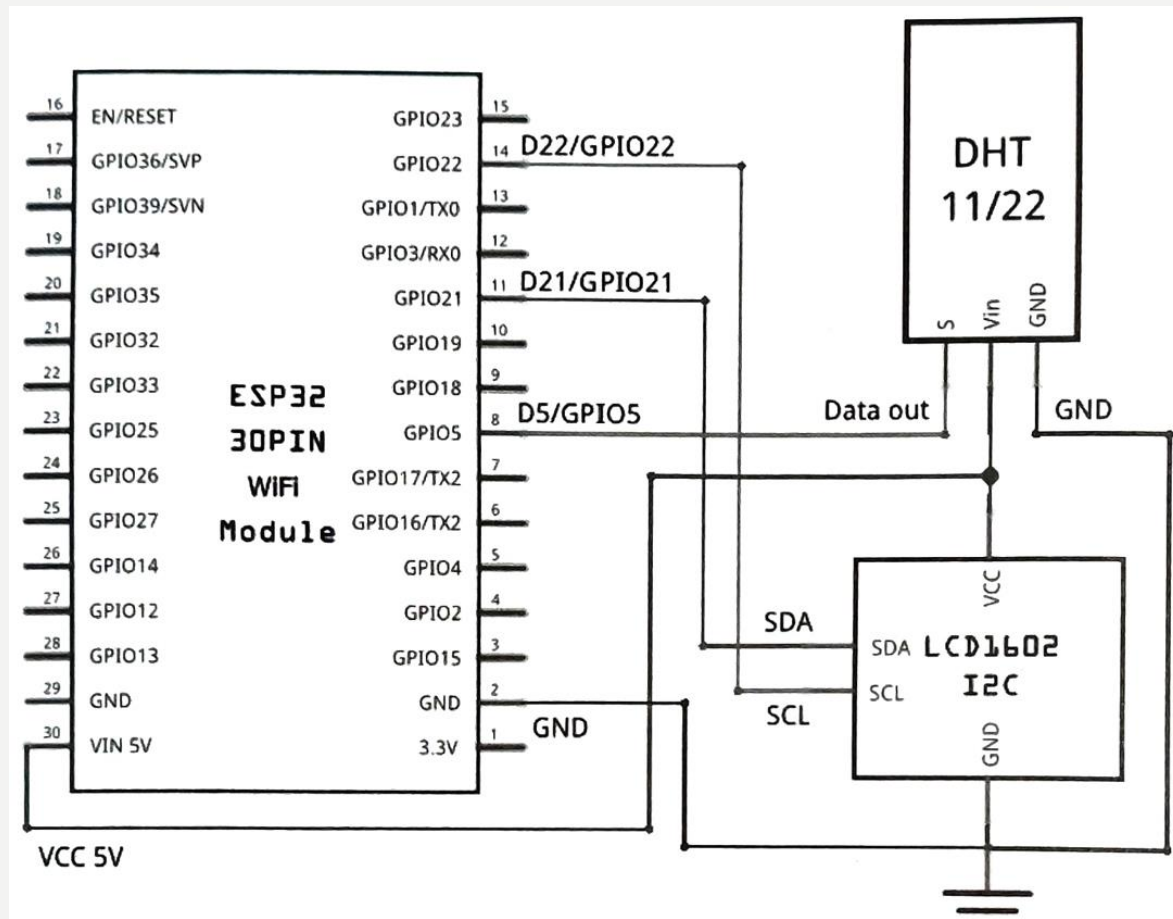
เซ็นเซอร์และการทำงาน

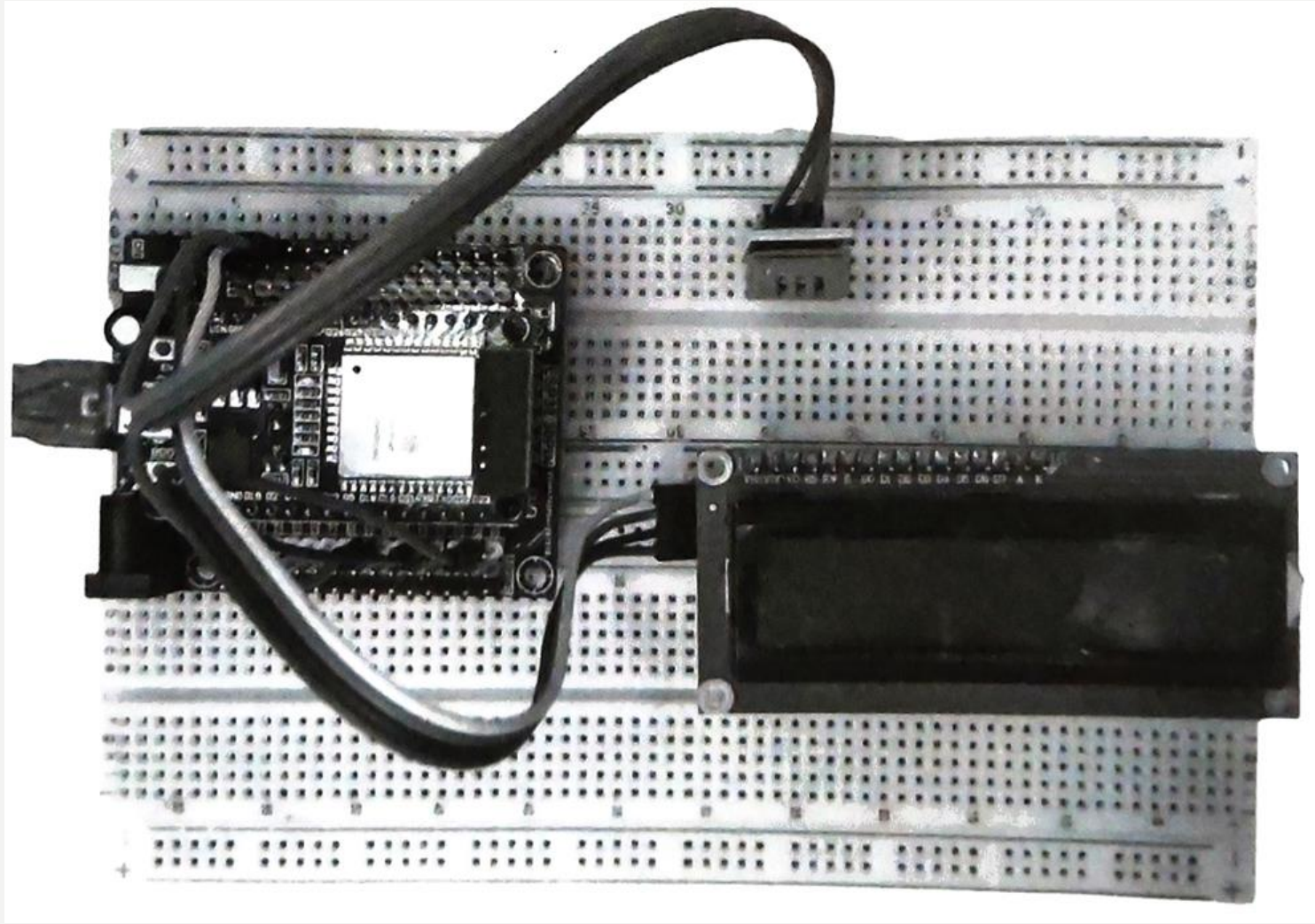


จุดประสงค์การเรียนรู้

1. ศึกษาการแสดงผลอุณหภูมิและความชื้นออกทางโมดูลจอ LCD
2. ศึกษาการเปิด/ปิดหลอดไฟ LED ด้วยเซ็นเซอร์ LDR
3. ศึกษาการวัดระยะทางและแจ้งเตือนด้วย Ultrasonic Sensor

การทดลองที่ 1 การแสดงอุณหภูมิและความชื้นออกทางโมดูลจอ LCD





Library Manager

Type Topic 2

DFRobot_DHT20
by fengli DFRobot
Provide an Arduino library to get Humidity and Temperature by reading data from dht20.
Product Link: <https://www.dfrobot.com/product-2391.html>
Product Link: <https://www.dfrobot.com.cn/goods-3282.html>
[More info](#)

DHT sensor library
by Adafruit
Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
[More info](#) Version 1.4.2 3

DHT sensor library for ESPx
by beegee_tokyo
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized libray to match ESP32 requirements. Last changes: Fix negative temperature problem (credits @helijunky)
[More info](#)


```
#include <Wire.h> //เรียกใช้ไลบรารี Wire.h
#include <LiquidCrystal_I2C.h> //เรียกใช้ไลบรารี LiquidCrystal_I2C.h
#include <DHT.h> //เรียกใช้ไลบรารี DHT.h

#define DHTPIN 5 //กำหนดให้ขา GPIO5/D5 เป็นขาอินพุตที่รับข้อมูลมา จากขา Data Out
ของโมดูล DHT11 มาเก็บไว้ในตัวแปร DHTPIN
#define DHTTYPE DHT11 //กำหนดให้ตัวแปร DHTTYPE ใช้เก็บชนิดของตัวเซ็นเซอร์

LiquidCrystal_I2C lcd(0x27, 16, 2); //สร้างออบเจ็กต์จากคลาส LiquidCrystal_I2C แล้วนำไป
เก็บไว้ในตัวแปร lcd โดยจะต้องผ่านค่าแอดเดรส 0x27 และขนาด
ของหน้าจอ LCD (แสดงผล 2 แถว แถวละ 16 ตัวอักษร)

DHT dht(DHTPIN, DHTTYPE); //สร้างออบเจ็กต์จากคลาส DHT แล้วนำไปเก็บไว้ในตัวแปร dht
โดยใช้ค่าที่เก็บไว้ในตัวแปร DHTPIN และ DHTTYPE
```

```
void setup() {  
    dht.begin();           //เริ่มต้นใช้งานเซ็นเซอร์ DHT  
    lcd.begin();          //เริ่มต้นการสื่อสารกับโมดูลจอ LCD  
    lcd.backlight();  
    lcd.print("DHT11 Test!"); //แสดงข้อความใน " " ออกทางหน้าจอ LCD  
}
```

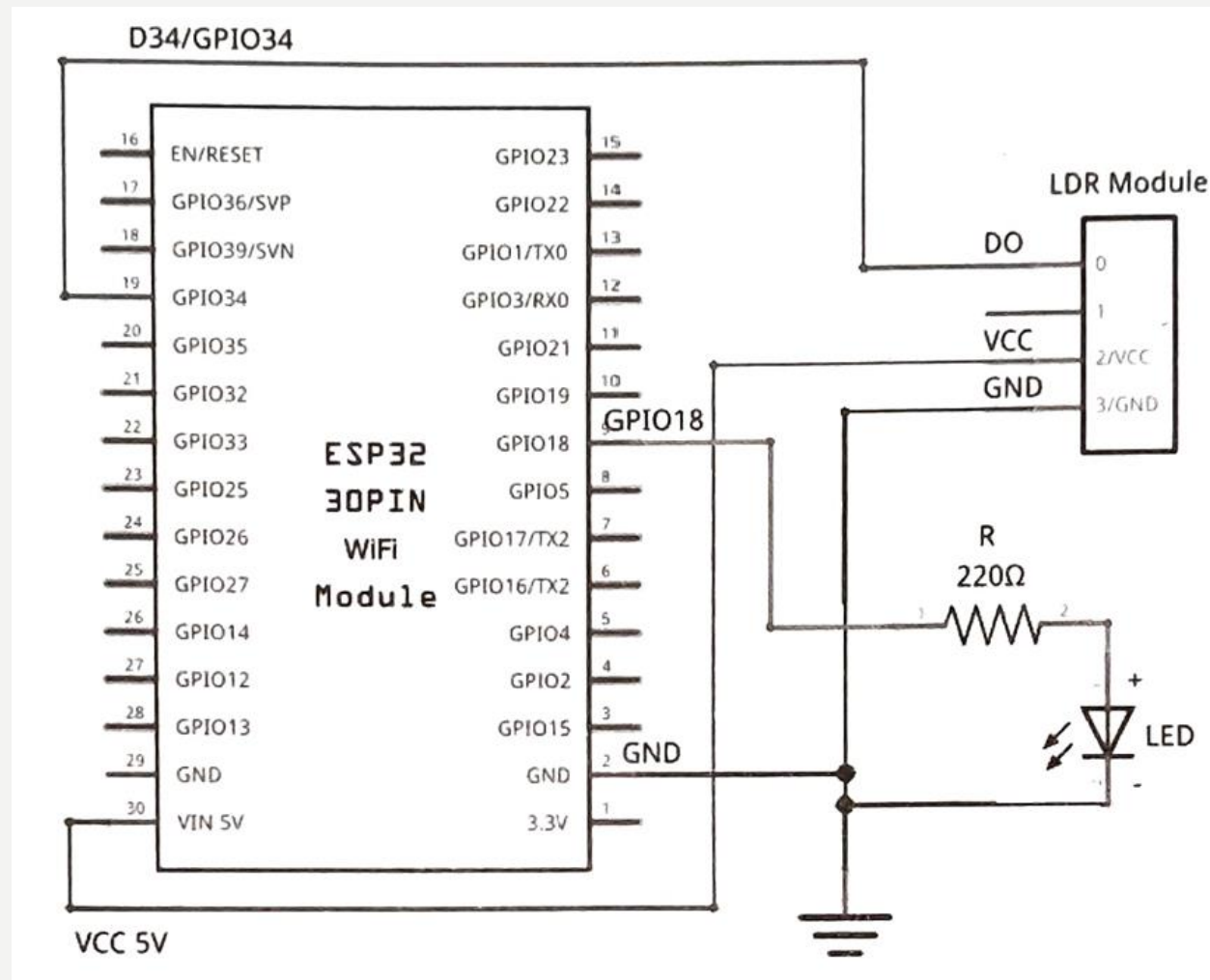
```
void loop() {  
    delay(2000);           //หน่วงรอเป็นเวลา 2 วินาที  
    float t = dht.readTemperature(); //ตัวแปร t ชนิด float ใช้เก็บค่าอุณหภูมิที่อ่านได้จาก  
                                     ตัวเซ็นเซอร์  
    float h = dht.readHumidity(); //ตัวแปร h ชนิด float ใช้เก็บค่าความชื้นที่อ่านได้จากตัวเซ็นเซอร์
```

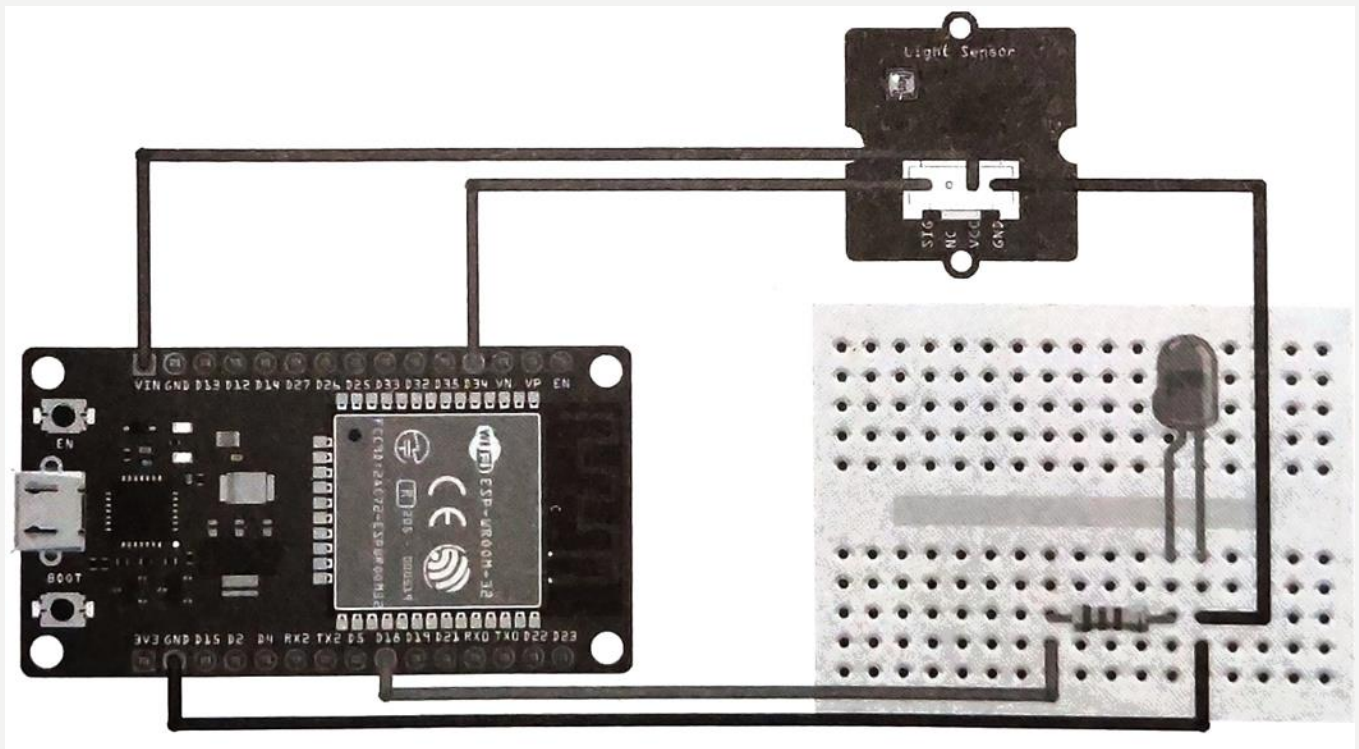
```
if (isnan(t) || isnan(h)) { //ถ้าไม่สามารถอ่านค่าอุณหภูมิและความชื้นที่เก็บไว้ในตัวแปร
                             t และ h ได้ แสดงว่าเกิดข้อผิดพลาด
    lcd.clear();           //เคลียร์หน้าจอ
    lcd.print("DHT Failed!"); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
} else {                   //แต่ถ้าไม่
    lcd.setCursor(0, 0);   //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 0 แถว 0
    lcd.print("Temp : "); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
    lcd.setCursor(7, 0);  //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 7 แถว 0
    lcd.print(t);         //แสดงค่าตัวเลขของอุณหภูมิที่เก็บไว้ในตัวแปร t
    lcd.setCursor(13, 0); //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 13 แถว 0
    lcd.print((char)223); //แสดงสัญลักษณ์องศา (°)
    lcd.setCursor(14, 0); //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 14 แถว 0
    lcd.print("C");       //แสดงข้อความใน " " ออกทางหน้าจอ LCD
    lcd.setCursor(0, 1);  //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 0 แถว 1
    lcd.print("Humid : "); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
    lcd.setCursor(8, 1);  //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 8 แถว 1
    lcd.print(h);         //แสดงค่าตัวเลขของความชื้นที่เก็บไว้ในตัวแปร h
    lcd.setCursor(14, 1); //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 14 แถว 1
    lcd.print("%");       //แสดงข้อความใน " " ออกทางหน้าจอ LCD
```



```
}  
  delay(1000);  
}
```

การทดลองที่ 2 การเปิด/ปิดหลอดไฟ LED ด้วยเซ็นเซอร์ LDR





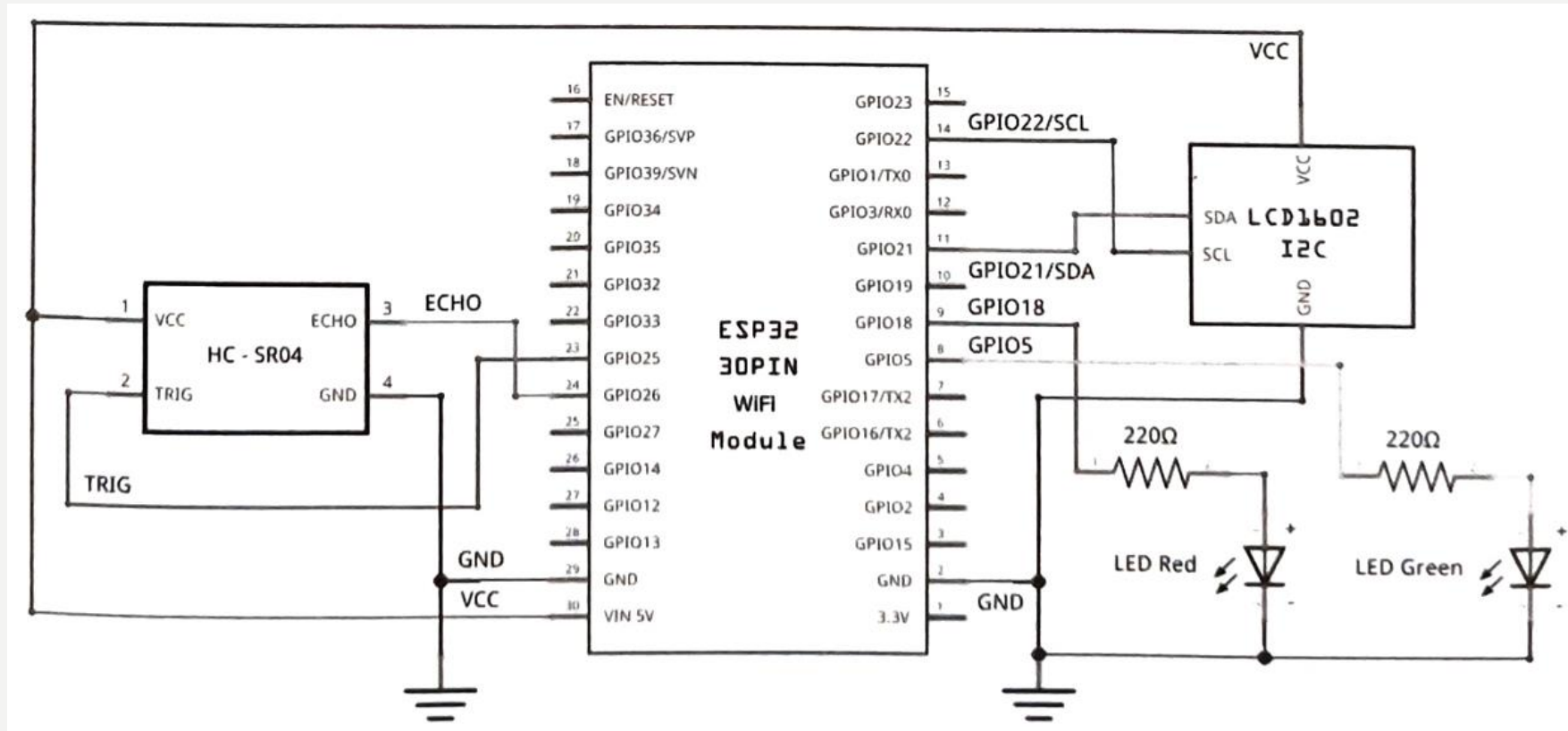
```
const int ledPin = 18;           //ประกาศตัวแปรขาที่ต่อกับหลอดไฟ LED
const int ldrPin = 34;          //ประกาศตัวแปรขาอินพุต Digital

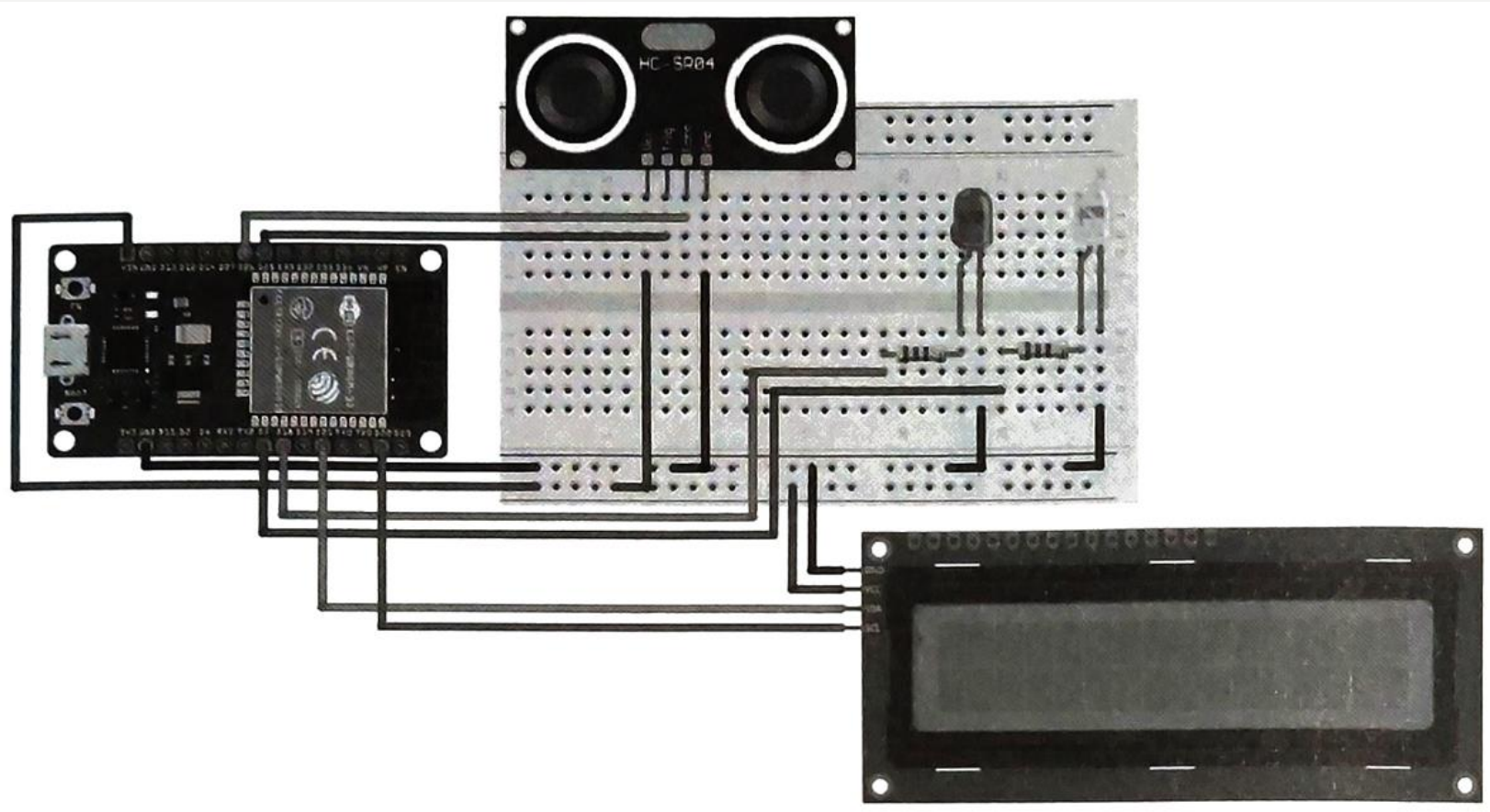
void setup() {

Serial.begin(115200);
pinMode(ledPin, OUTPUT);        //กำหนดให้ตัวแปรหรือขา D18/GPIO18 เป็น Output
pinMode(ldrPin, INPUT);        //กำหนดให้ตัวแปรหรือขา D34/GPIO34 เป็น Input
}

void loop() {
int ldrStatus = digitalRead(ldrPin); //อ่านค่าอินพุต Digital จาก ldrPin ไปเก็บไว้ที่ตัวแปร
if (ldrStatus == HIGH) {        //ตรวจสอบว่าถ้าค่าในตัวแปรสูงกว่าค่าที่ถูกกำหนดจากตัวต้านทานปรับค่าได้
digitalWrite(ledPin, HIGH);     //ให้ Output ที่ขา GPIO18 เป็น HIGH หลอดไฟ "ติด"
Serial.println("LDR --> DARK, LED --> ON"); //แสดงข้อความใน " " ออกทาง Serial Monitor
} else {                        //แต่ถ้าไม่
digitalWrite(ledPin, LOW);      //ให้ Output ที่ขา GPIO18 เป็น LOW หลอดไฟ "ดับ"
Serial.println("LDR --> Bright, LED --> OFF"); //แสดงข้อความใน " " ออกทาง Serial Monitor
}
delay(1000);
}
```

การทดลองที่ 3 การวัดระยะทางและแจ้งเตือนด้วย Ultrasonic Sensor





```
#include <Wire.h> //เรียกใช้ไลบรารี Wire.h
#include <LiquidCrystal_I2C.h> //เรียกใช้ไลบรารี LiquidCrystal_I2C.h
LiquidCrystal_I2C lcd(0x27, 16, 2); //สร้างออบเจกต์จากคลาส LiquidCrystal_I2C แล้วนำไป
//เก็บไว้ในตัวแปร lcd โดยจะต้องผ่านค่าแอดเดรสของโมดูล
//LCD (ในที่นี้แอดเดรสคือ 0x27) และขนาดของหน้าจอ LCD
// (แสดงผล 2 แถว แถวละ 16 ตัวอักษร)

#define TRIG 25 //กำหนดให้ขา GPIO25/D25 เป็นขา TRIG
#define ECHO 26 //กำหนดให้ขา GPIO26/D26 เป็นขา ECHO
#define RED_PIN 18 //กำหนดให้ขา GPIO18/D18 เป็นขา RED_PIN
#define GREEN_PIN 5 //กำหนดให้ขา GPIO5/D5 เป็นขา GREEN_PIN
long duration, distance; //ประกาศตัวแปรชนิด long ใช้เก็บระยะเวลาที่คลื่นเสียง
//เดินทางไปและกลับ (duration) และระยะทางหรือ
//ระยะห่างระหว่างตัวเซ็นเซอร์กับวัตถุ (distance)
```

```
void setup() {
  pinMode(TRIG, OUTPUT);           //กำหนดให้ขา TRIG เป็น Output
  pinMode(ECHO, INPUT);           //กำหนดให้ขา ECHO เป็น Input
  pinMode(RED_PIN, OUTPUT);       //กำหนดให้ขา RED_PIN เป็น Output
  pinMode(GREEN_PIN, OUTPUT);    //กำหนดให้ขา GREEN_PIN เป็น Output
  lcd.begin();                   //เริ่มต้นการสื่อสารกับโมดูลจอ LCD
  lcd.backlight();
  lcd.home();                    //เลื่อนเคอร์เซอร์ไปยังจุดเริ่มต้นที่ตำแหน่งคอลัมน์ 0 (ซ้ายสุด)
                                  //แถว 0 (บรรทัดบน)

  lcd.print("Check Distance");    //แสดงข้อความใน " " ออกทางหน้าจอ LCD
  lcd.setCursor(0, 1);           //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 0 แถว 1 (บรรทัดล่าง)
  lcd.print("Ready!");          //แสดงข้อความใน " " ออกทางหน้าจอ LCD
  delay(2000);                  //หน่วงรอเป็นเวลา 2 วินาที
}
```

```

void loop() {
    digitalWrite(TRIG, LOW);           //กำหนดให้ Output ที่ขา TRIG มีสถานะเป็น LOW
    delayMicroseconds(5);              //หน่วงรอเป็นเวลา 5 ไมโครวินาที (µs) เพื่อเตรียมพร้อม
                                        หรือเพื่อให้แน่ใจว่าจะยังไม่มีคลื่นวิทยุถูกส่งออกไปในตอนนี

    digitalWrite(TRIG, HIGH);         //กำหนดให้ Output ที่ขา TRIG มีสถานะเป็น HIGH

    delayMicroseconds(10);            //หน่วงรอเป็นเวลา 10 ไมโครวินาที (µs) เพื่อให้เกิด
                                        สัญญาณพัลส์ที่เป็น HIGH กว้างอย่างน้อย 10 µs

    digitalWrite(TRIG, LOW);          //เมื่อครบ 10 µs จึงกำหนดให้ Output ที่ขา TRIG มีสถานะเป็น LOW
                                        จากนั้นเซ็นเซอร์จะส่งคลื่นเสียงความถี่สูงจำนวน 8 Cycle ออกไป

    duration = pulseIn(ECHO, HIGH);    //ด้วยคำสั่งนี้ บอร์ดจะเริ่มนับเวลาตั้งแต่คลื่นเสียงเริ่มเดิน
                                        ทางไป หรือก็คือเมื่อขา ECHO เริ่มเปลี่ยนสถานะจาก LOW และจะนับ
                                        ไปเรื่อยๆ (คงสถานะ HIGH ไว้) จนกว่าเป็น HIGH คลื่นเสียงจะสะท้อน
                                        และเดินทางกลับมา ซึ่งจะทำให้ขา ECHO เปลี่ยนสถานะจาก HIGH เป็น
                                        LOW ค่าเวลาที่ได้ในช่วง HIGH จะถูกเก็บไว้ในตัวแปร duration

    distance = duration * 0.034 / 2;   //นำค่าในตัวแปร duration มาคำนวณหาค่าระยะทางหรือ
                                        ระยะห่างระหว่างตัวเซ็นเซอร์กับวัตถุ (distance)
}

```



```

lcd.setCursor(0, 0); //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 0 แถว 0 (บรรทัดบน)
lcd.print("Distance: "); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
lcd.print(distance); //แสดงค่าตัวเลขที่เก็บไว้ในตัวแปร distance
lcd.print(" cm. "); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
if (distance >= 15) { //ถ้าระยะทางที่คำนวณได้มากกว่าหรือเท่ากับ 15 เซนติเมตร
    digitalWrite(GREEN_PIN, HIGH); //กำหนดให้ขา GREEN_PIN เป็น HIGH หลอดไฟสีเขียวติด
    digitalWrite(RED_PIN, LOW); //กำหนดให้ขา RED_PIN เป็น LOW หลอดไฟสีแดงดับ
    lcd.setCursor(0, 1); //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 0 แถว 1 (บรรทัดล่าง)
    lcd.print("Safe!"); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
} else { //แต่ถ้าไม่
    digitalWrite(GREEN_PIN, LOW); //กำหนดให้ขา GREEN_PIN เป็น LOW หลอดไฟสีเขียวดับ
    digitalWrite(RED_PIN, HIGH); //กำหนดให้ขา RED_PIN เป็น HIGH หลอดไฟสีแดงติด
    lcd.setCursor(0, 1); //เลื่อนเคอร์เซอร์ไปที่ตำแหน่งคอลัมน์ 0 แถว 1 (บรรทัดล่าง)
    lcd.print("STOP! "); //แสดงข้อความใน " " ออกทางหน้าจอ LCD
}
delay(500);
}

```