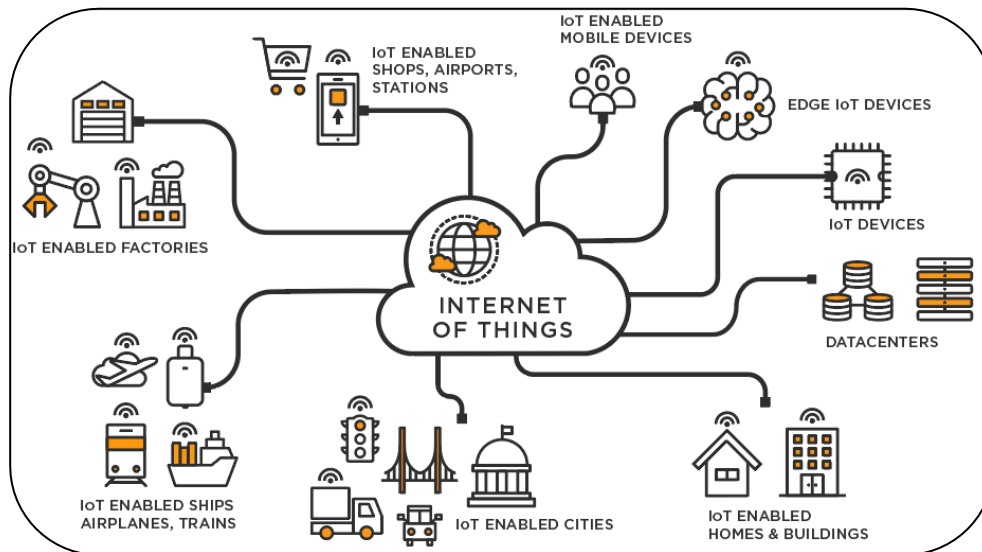


บทที่ 4 การประยุกต์ใช้ Arduino ควบคุม Sensor แสง อุณหภูมิ และความชื้น



<http://www.siam2dev.com>

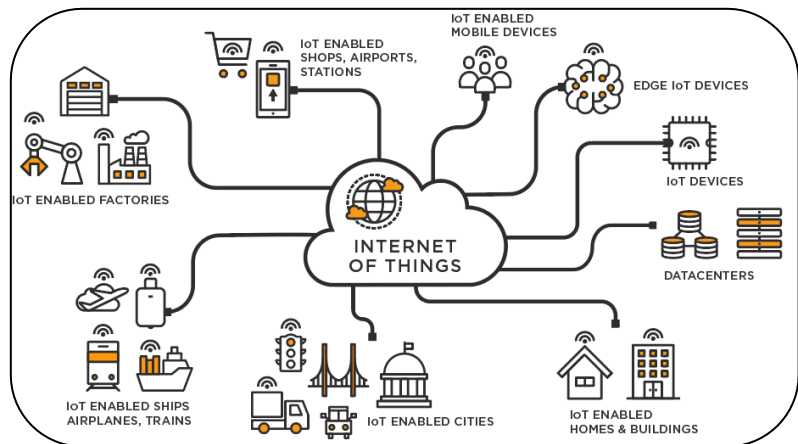
Last update :
02.08.2568

IOT: Internet of Things

Asst. Dr. Nattapong Songneam

<http://www.siam2dev.net>

Agenda



4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และความชื้น

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ ตรวจสอบแสง

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.5 soil module sensor

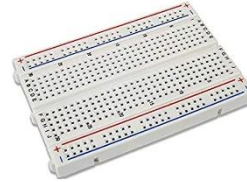
การเตรียมอุปกรณ์ (เซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และความชื้น)



Arduino uno r3



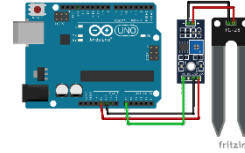
LED



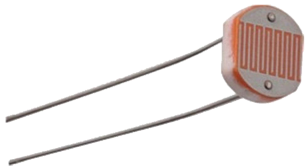
BREADBORD



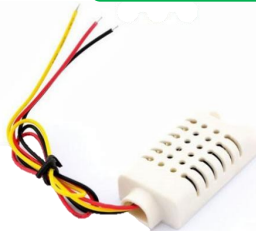
Resister



soil module sen
sor



LDR



DHT22



Jumper



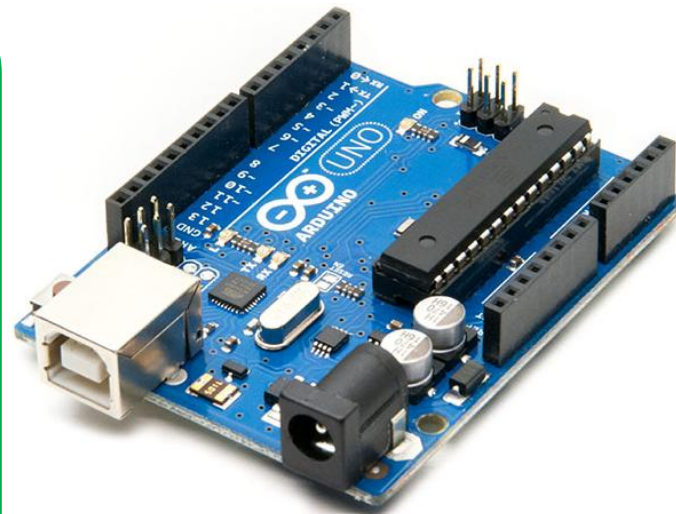
4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจจับ แสง อุณหภูมิ และความชื้น

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจจับ แสง อุณหภูมิ และ ความชื้น

4.1.1 Arduino Uno R3

อุปกรณ์หลักที่ใช้ในการพัฒนาระบบด้วยแนวคิด IoT เป็นเหมือนจุดที่ใช้ในการจ่ายไฟ หรือส่งต่อข้อมูลไปยังอุปกรณ์ต่าง ๆ โดยวงจรจะสามารถจะต้องมีแหล่งจ่ายไฟ ซึ่งแหล่งจ่ายไฟพื้นฐานได้แก่

- GND ย่อมาจาก GROUND คือ จุดต่ำสุดย่นรวม แรงดันอ้างอิง 0 V โดยอุปกรณ์ภายในวงจร จะมีกระแสไหลมาครบรอบที่ทราวด์ ถูกเรียกว่า ครบวงจร
- VIN คือ ขั้วต่อแหล่งจ่ายภายนอก ที่เข้าบอร์ด จากกรณีของ NodeMCU เราใช้ USB อะแดปเตอร์เสียบจ่ายไฟเข้าวงจร ดังนั้น vin ก็คือ แรงดัน (Voltage) ที่ได้จาก USB 5V
- VCC หรือ VDD คือ แรงดันที่ปรับค่าแล้ว ให้เหมาะสมกับวงจรนั้น ๆ ซึ่งกรณี NodeMCU เราใช้ USB อะแดปเตอร์เสียบจ่ายไฟ 5V จากนั้นในวงจรจะมีการปรับลดแรงดันลงเหลือ 3.3 v



ภาพที่ 4.1 Arduino Uno R3

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และ ความชื้น

4.1.2 หลอด LED

LED ย่อมาจาก Light Emitting Diode เป็นหลอดชนิดหนึ่งที่สามารถเปล่งแสงได้เมื่อจ่ายไฟในปริมาณที่เหมาะสมที่ 2 ขา คือ แอโนด (Anode) และ แคโทด (Cathode) โดย แอโนดจะต่อกับไฟขับเคลื่อน สังเกตได้จาก แอโนดจะมีขาที่ยาว ส่วนแคโทดจะต่อกับ GND และมีขาที่สั้นกว่า



ภาพที่ 4.2 หลอด LED

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และ ความชื้น

4.1.3 ตัวต้านทาน

เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่งที่มีคุณสมบัติในการต้านการไหลผ่านของกระแสไฟฟ้า ทำด้วยลวดต้านทานหรือถ่านคาร์บอน เป็นต้น นั่นคือ ถ้าอุปกรณ์นั้นมีความต้านทานมาก กระแสไฟฟ้าที่ไหลผ่านจะน้อยลง เป็นอุปกรณ์ไฟฟ้าชนิดพาสซีฟสองขั้ว ที่สร้างความต่างศักย์ไฟฟ้าคร่อมขั้วทั้งสอง (V) โดยมีสัดส่วนมากน้อยตามปริมาณกระแสไฟฟ้าที่ไหลผ่าน (I) อัตราส่วนระหว่างความต่างศักย์ และปริมาณกระแสไฟฟ้า ก็คือ ค่าความต้านทานทางไฟฟ้า หรือค่าความต้านทานของตัวนำมีหน่วยเป็นโอห์ม (สัญลักษณ์ : Ω)



ภาพที่ 4.3 ตัวต้านทาน

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และ ความชื้น

4.1.3 ตัวต้านทาน

ตารางที่แสดงตัวอย่างอุปกรณ์ IoT และขนาดของตัวต้านทานที่เหมาะสม:

อุปกรณ์ IoT	ฟังก์ชัน	ขนาดของตัวต้านทานที่เหมาะสม	รายละเอียด
เซ็นเซอร์อุณหภูมิ (DS18B20)	วัดอุณหภูมิ	4.7kΩ	ใช้ในวงจร Pull-Up เพื่อเชื่อมต่อกับบัส One-Wire
เซ็นเซอร์ความชื้น (DHT22)	วัดความชื้น	10kΩ	ใช้ในวงจร Pull-Up สำหรับสายสัญญาณ Data
ESP8266	การสื่อสาร Wi-Fi	10kΩ	ใช้ในวงจร Pull-Up หรือ Pull-Down บางพิน
รีเลย์ (Relay)	เปิด/ปิดอุปกรณ์ไฟฟ้า	1kΩ ถึง 10kΩ	ขนาดตัวต้านทานขึ้นอยู่กับทรานซิสเตอร์ที่ใช้ควบคุมรีเลย์
มอเตอร์เซอร์โว (Servo Motor)	การควบคุมทิศทางและตำแหน่ง	220Ω ถึง 1kΩ	ใช้ตัวต้านทานเพื่อจำกัดกระแสในวงจรควบคุม
โมดูล Bluetooth (HC-05)	การสื่อสาร Bluetooth	1kΩ ถึง 10kΩ	ใช้ในวงจร Level Shifter สำหรับพิน RX/TX
LED	แสดงผล	220Ω ถึง 1kΩ	ขนาดขึ้นอยู่กับแรงดันที่จ่ายและค่ากระแสที่ต้องการสำหรับ LED

คำอธิบายเพิ่มเติม:

• **ตัวต้านทาน Pull-Up/Pull-Down:** ใช้เพื่อให้แน่ใจว่าพินของไมโครคอนโทรลเลอร์หรือเซ็นเซอร์มีสถานะที่แน่นอนเมื่อไม่มีการเชื่อมต่อกับแหล่งสัญญาณอื่น

• **ตัวต้านทานจำกัดกระแส:** ใช้เพื่อป้องกันการไหลของกระแสเกินที่อาจทำให้อุปกรณ์เสียหาย

• **วงจร Level Shifter:** ใช้ในการปรับระดับแรงดันไฟฟ้าให้เหมาะสมระหว่างอุปกรณ์ที่มีแรงดันไฟฟ้าแตกต่างกัน

หากคุณต้องการรายละเอียดเพิ่มเติมเกี่ยวกับการใช้งานหรือการคำนวณขนาดตัวต้านทานในวงจรเฉพาะเจาะจง สามารถสอบถามเพิ่มเติมได้ครับ

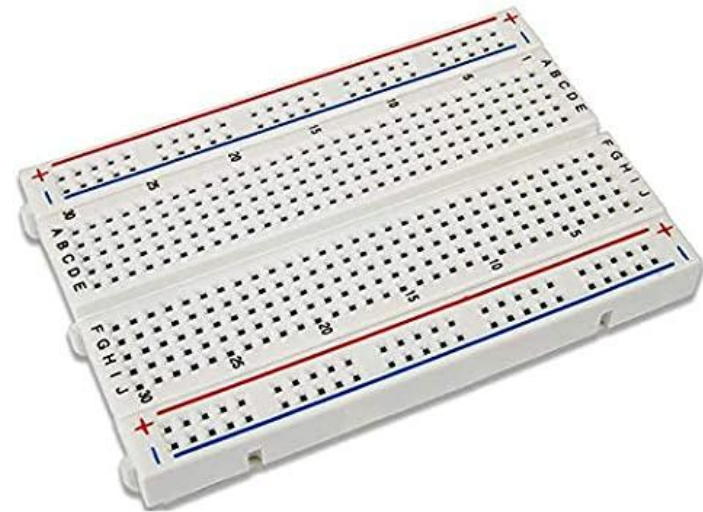


ภาพที่ 4.3 ตัวต้านทาน

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจวัด แสง อุณหภูมิ และ ความชื้น

4.1.4 บอร์ดทดลอง (Bread Board)

บอร์ดที่ใช้ทดลองวงจรอิเล็กทรอนิกส์ ลักษณะเป็นแผ่นพลาสติกหนาสีขาว บนแผ่นมีรูเรียงกันจำนวนมาก ภายในรูมีตัวนำไฟฟ้าซึ่งเชื่อมต่อกันในรูปแบบที่มีการกำหนดไว้ เวลาทดลองก็เสียบขาของอุปกรณ์อิเล็กทรอนิกส์ลงไปให้ตัวนำภายในเชื่อมวงจรถึงกัน และอาจใช้สายไฟเสียบลงรูเพื่อเชื่อมวงจรไฟฟ้าได้เช่นกัน ข้อดีของโปรโตบอร์ดคือไม่ต้องออกแบบแผงวงจรและไม่ต้องบัดกรี แต่มีข้อเสียคือใช้ทดลองวงจรที่ทำงานที่ความถี่สูง ๆ ไม่ได้เนื่องจากมีปัญหาเรื่องสัญญาณรบกวนในวงจร

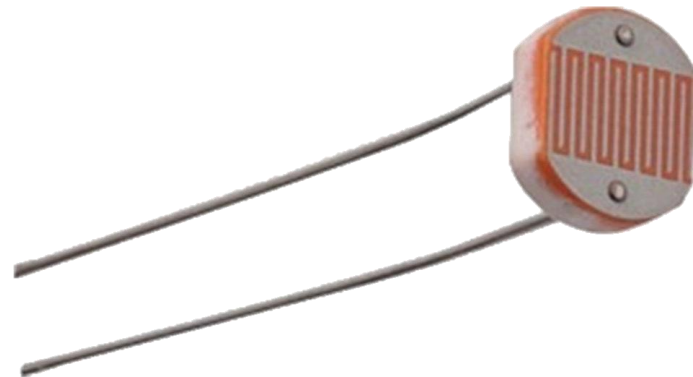


ภาพที่ 4.4 บอร์ดทดลอง (Bread Board)

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และ ความชื้น

4.1.5 เซ็นเซอร์ตรวจสอบแสง LDR

แอลดีอาร์ (LDR) หรือชื่อเต็ม ๆ คือ Light Dependent Resistor หรือตัวต้านทานที่แปรค่าตามแสง คือ ตัวต้านทานชนิดที่เปลี่ยนสภาพความนำไฟฟ้า (Conductance) ได้เมื่อมีแสงมาตกกระทบ ทำจากวัสดุสารกึ่งตัวนำที่ไวต่อแสง บางครั้งเราเรียก LDR เซ็นเซอร์ชนิดนี้ว่าโฟโตรีซิสเตอร์ (Photoresistor) หรือ โฟโตคอนดักเตอร์ (Photoconductor) ภาพที่ ...



ภาพที่ 4.5 LDR (Light Dependent Resistor)

4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจวัด แสง อุณหภูมิ และความชื้น

4.1.6 เซ็นเซอร์วัดอุณหภูมิ และความชื้น

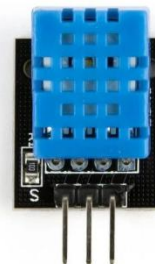
DHT22 เป็นโมดูลวัดอุณหภูมิและความชื้นที่ความละเอียดและช่วงการวัดที่สูงกว่า DHT11 ใช้ไฟได้ 3-5V สามารถวัดอุณหภูมิได้ตั้งแต่ -40 ถึง 80°C ที่ความแม่นยำ $\pm 0.5^{\circ}\text{C}$ และความชื้น 0-100% คลาดเคลื่อน 2-5% อัตราการอ่านค่าอุณหภูมิและความชื้นสูงสุด 0.5Hz โมดูล DHT22 ใช้งานง่าย มีไลบรารีพร้อมใช้



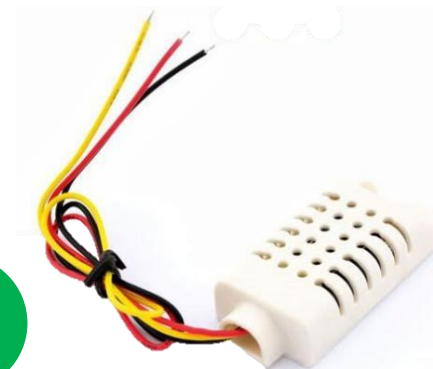
ภาพที่ 4.6 เซ็นเซอร์วัดอุณหภูมิ และความชื้น (DHT22)

แนะนำอุปกรณ์

เซ็นเซอร์วัดอุณหภูมิ และความชื้น
(DHT22)



DHT11

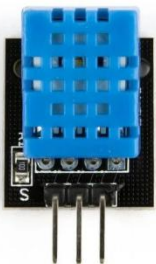


DHT22

DHT11 กับ DHT22 ต่างกันอย่างไร

แนะนำอุปกรณ์

DHT11

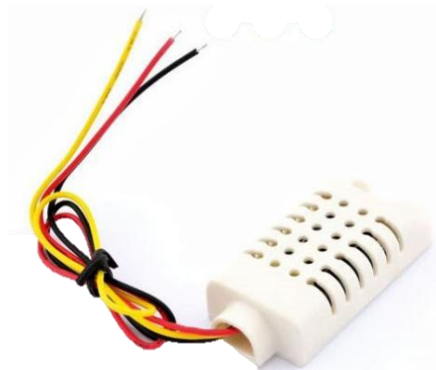


DHT11 เป็นเซนเซอร์วัดอุณหภูมิและความชื้นที่มีความทันสมัยและราคาถูก โดยมีความสามารถที่มีนัยสำคัญดังนี้:

- 1.วัดอุณหภูมิ: DHT11 สามารถวัดอุณหภูมิในช่วง 0°C ถึง 50°C และมีความแม่นยำเป็น $\pm 2^{\circ}\text{C}$.
- 2.วัดความชื้น: เซนเซอร์นี้สามารถวัดความชื้นในช่วง 20% ถึง 80% และมีความแม่นยำเป็น $\pm 5\% \text{ RH}$.
- 3.ราคาถูก: DHT11 เป็นทางเลือกที่คุ้มค่าและราคาถูกเมื่อเทียบกับเซนเซอร์อื่นที่ให้ความสามารถเทียบเท่า.
- 4.การใช้งานง่าย: การใช้งาน DHT11 ไม่ซับซ้อนมาก มีไลบรารีที่สามารถใช้กับหลายแพลตฟอร์มไมโครคอนโทรลเลอร์ และมีข้อมูลการใช้งานที่มีอยู่มากมายในชุมชนออนไลน์.
- 5.การเชื่อมต่อ: DHT11 มีอินเตอร์เฟซที่ใช้งานได้ง่ายและใช้แบบดิจิทัล.
- 6.การสื่อสารแบบสัญญาณดิจิทัล: DHT11 ส่งข้อมูลอุณหภูมิและความชื้นผ่านสัญญาณดิจิทัล, ทำให้มีความเสถียรและสามารถใช้งานได้ในระบบที่ไม่สามารถรับสัญญาณแอนะล็อกได้.

แนะนำอุปกรณ์

DHT22



DHT22 (หรือ AM2302) เป็นเซนเซอร์วัดอุณหภูมิและความชื้นที่มีความแม่นยำและสามารถให้ข้อมูลในเงื่อนไขที่หลากหลาย. นี่คือคุณสมบัติหลักของ DHT22:

- 1.วัดอุณหภูมิ: DHT22 สามารถวัดอุณหภูมิในช่วง -40°C ถึง 80°C และมีความแม่นยำเป็น $\pm 0.5^{\circ}\text{C}$.
- 2.วัดความชื้น: เซนเซอร์นี้สามารถวัดความชื้นในช่วง 0% ถึง 100% และมีความแม่นยำเป็น $\pm 2\% \text{ RH}$.
- 3.ความละเอียดของการวัด: DHT22 มีความละเอียดสูงในการวัดอุณหภูมิและความชื้น, โดยมีความละเอียดถึง 0.1°C สำหรับอุณหภูมิและ $0.1\% \text{ RH}$ สำหรับความชื้น.
- 4.การสื่อสารแบบชนิดดิจิทัล: DHT22 ส่งข้อมูลผ่านสัญญาณดิจิทัล, ทำให้มีความเสถียรและสามารถใช้งานได้ในระบบที่ไม่สามารถรับสัญญาณแอนะล็อกได้.
- 5.ช่วงการทำงานของแรงดันไฟฟ้า: DHT22 สามารถทำงานในระหว่าง 3.3V ถึง 6V, ทำให้เหมาะสำหรับการใช้งานกับหลายแหล่งจ่ายไฟ.
- 6.ความเสถียร: DHT22 มีความเสถียรที่สูงและมีการป้องกันต่อการแปรผันไฟฟ้าที่เป็นไปได้.
- 7.การใช้งานที่หลากหลาย: DHT22 มีไลบรารีที่สามารถใช้งานได้กับหลายแพลตฟอร์มไมโครคอนโทรลเลอร์.

แนะนำอุปกรณ์

เปรียบเทียบความแตกต่างของ DHT11 และ DHT22

โปรโตคอลการสื่อสารที่ใช้ใน DHT11 และ DHT22 มีความแตกต่างกัน ซึ่งอาจมีผลต่อประสิทธิภาพและความแม่นยำในการวัดข้อมูล แต่ทั้งสองเซนเซอร์นี้มีการใช้งานทั่วไปในการวัดอุณหภูมิและความชื้นในสภาพแวดล้อมทั่วไป

ลักษณะ	DHT11	DHT22
ช่วงการวัดอุณหภูมิ	0°C ถึง 50°C	-40°C ถึง 80°C
ช่วงการวัดความชื้น	20% ถึง 80%	0% ถึง 100%
ความละเอียดของการวัด	1°C	0.1°C
	1% RH	0.1% RH
ความแม่นยำ	+/- 2°C	+/- 0.5°C
	+/- 5% RH	+/- 2% RH
อินเทอร์เฟซ	ชนิดไดจิตอล	ชนิดดิจิตอล
แรงดันไฟฟ้า	3.5V ถึง 5.5V	3.3V ถึง 6V
การส่งข้อมูล	สัญญาณดิจิตอลแบบชนิดซีรี่ย์	สัญญาณดิจิตอลแบบชนิดซีรี่ย์

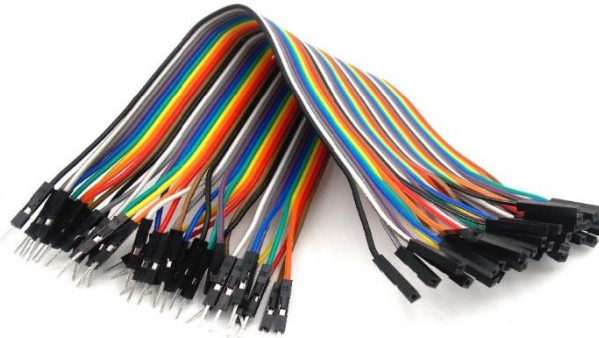
4.1 อุปกรณ์พื้นฐานเซ็นเซอร์ ตรวจสอบ แสง อุณหภูมิ และ ความชื้น

4.1.7 สายต่อ Jumper

ใช้สำหรับเสียบเข้ากับ Arduino และ บอร์ดทดลองเพื่อใช้ในการเชื่อมต่อวงจร ปลายหัวที่ใช้เสียบแบ่งออกเป็น ผู้ และ เมีย โดยผู้จะมีปลายแหลม ส่วนเมีย จะปลายเป็นรูเสียบ ซึ่งได้แบ่งสายจัมเปอร์ ออกเป็น 3 แบบ ดังนี้



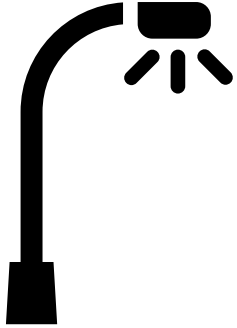
ผู้ - ผู้



ผู้ - เมีย



ภาพที่ 4.7 สายต่อจัมเปอร์ (Jumper)



4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ ตรวจจับแสง LDR



4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

หัวข้อนี้จะสอนใช้ ขา Arduino อ่านสัญญาณ analog 0-5v จากตัวต้านทานปรับค่าตามแสง LDR ซึ่งจะแปลงค่ามาเป็นข้อมูลแบบดิจิทัล 10 บิต จะมีค่า 1024 ระดับ โดยผลลัพธ์จะมีค่าระหว่าง 0 ถึง 1023 เราสามารถนำตัวต้านทานปรับค่าแสง LDR มาตรวจสอบความสว่างกลางวันกลางคืน มาควบคุมเปิดปิดไฟได้ โดยการใช้งาน Arduino สัญญาณ Analog Input ตัวต้านทานปรับค่าได้ตามแสง LDR ควบคุมเปิดปิดหลอดไฟ LED

คำสั่ง **Analog Read**

`val = analogRead(pin)`

pin หมายถึง ขา analog ของ arduino ที่ต้องการจะอ่าน

val หมายถึง ตัวแปร int สำหรับเก็บค่า สัญญาณ analog มีค่า **0-1023** ที่ขา pin

ตัวอย่างคำสั่ง Analog Read

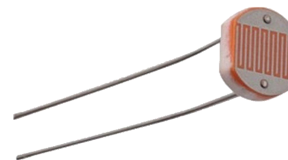
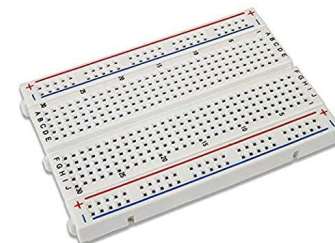
ต้องการอ่านค่า ขา Analog A0 เก็บในตัวแปร Val

`val = analogRead(A0)`

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

4.2.1 อุปกรณ์ที่ต้องใช้

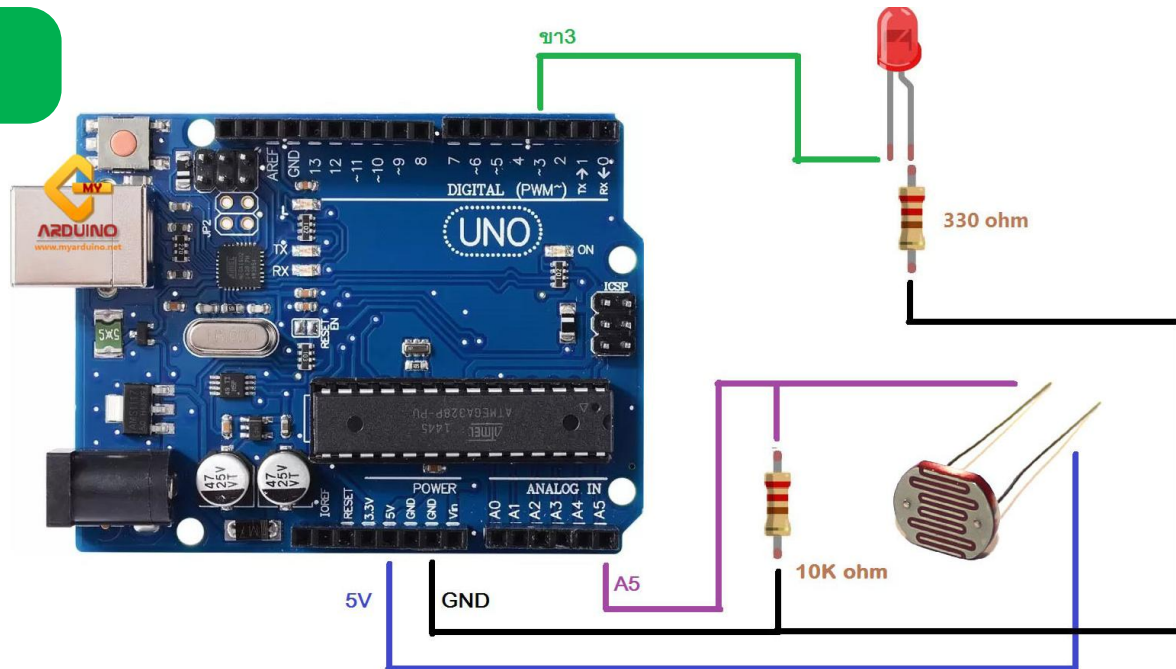
- Arduino UNO R3 พร้อม สาย USB
- เซ็นเซอร์ตรวจจับแสง LDR
- LED ขนาด 5mm สีแดง 1 ดวง
- บอร์ดทดลอง Breadboard
- Resistor ตัวต้านทาน 2 อัน
- สายไฟจัมเปอร์ คู่-คู่ 4 เส้น



4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

4.2.2 วิธีการต่อวงจร

- Arduino uno r3 -> หลอดไฟ LED
- ขา3 -> LED1
- Arduino uno r3 -> LDR
- ขาA5 -> ตัวต้านทานปรับค่าได้ LDR



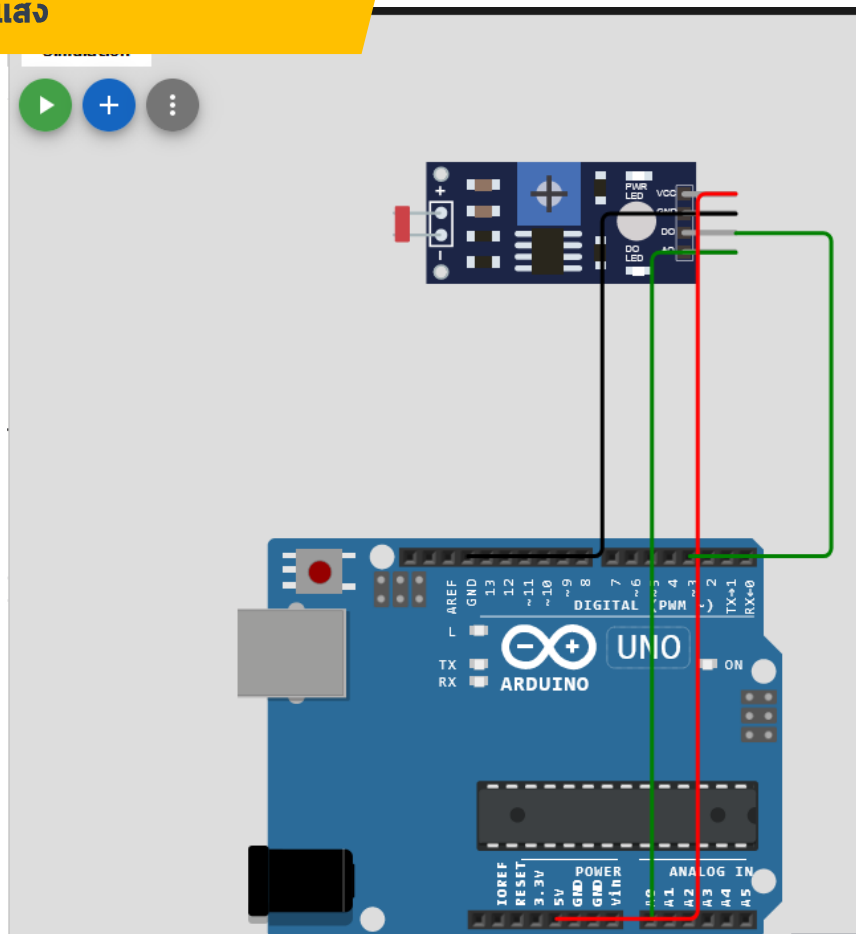
ภาพที่ 4.8 วิธีการต่อวงจร

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

WOKWI : LDR

sketch.ino • diagram.json • Library Manager ▾

```
1 int ledPin = 3;
2 int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่5
3 int val = 0;
4 void setup() {
5   pinMode(ledPin, OUTPUT); // sets the pin as output
6   Serial.begin(9600);
7 }
8 void loop() {
9   val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ LDR
10  Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
11  Serial.println(val); // พิมพ์ค่าของตัวแปร val
12  if (val < 100) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
13    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
14  }
15  else {
16    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
17  }
18  delay(100);
19 }
20
```



4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

sketch.ino

diagram.json

Library Manager

```
1 int ledPin = 3;
2 int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่:
3 int val = 0;
4 void setup() {
5   pinMode(ledPin, OUTPUT); // sets the pin as output
6   Serial.begin(9600);
7 }
8 void loop() {
9   val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ
10  Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
11  Serial.println(val); // พิมพ์ค่าของตัวแปร val
12  if (val < 100) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
13    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
14  }
15  else {
16    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
17  }
18  delay(100);
19 }
20
```

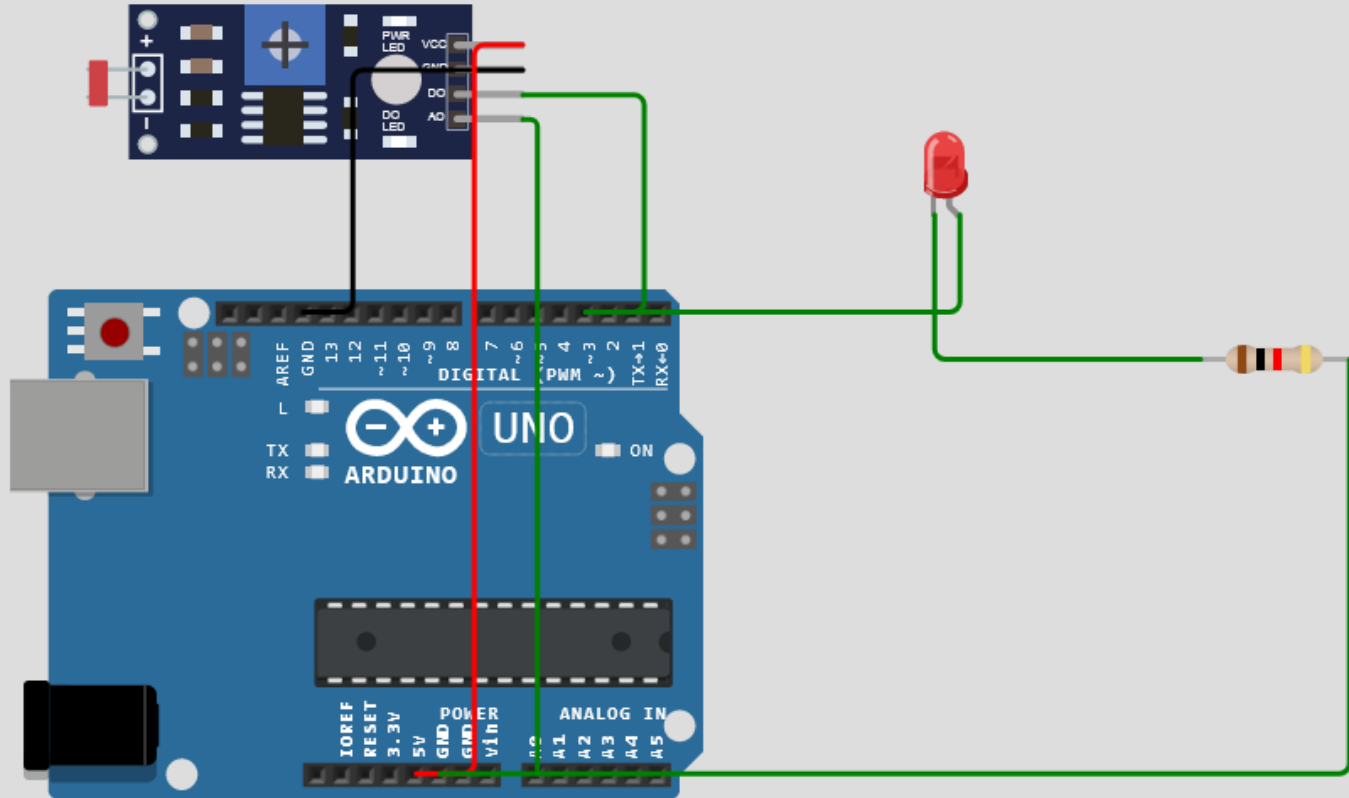
Photoresistor (LDR)

ILLUMINATION (LUX)

3467 lux

```
val = 473
val = 463
val = 513
val = 494
val = 456
val = 483
val = 498
```

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

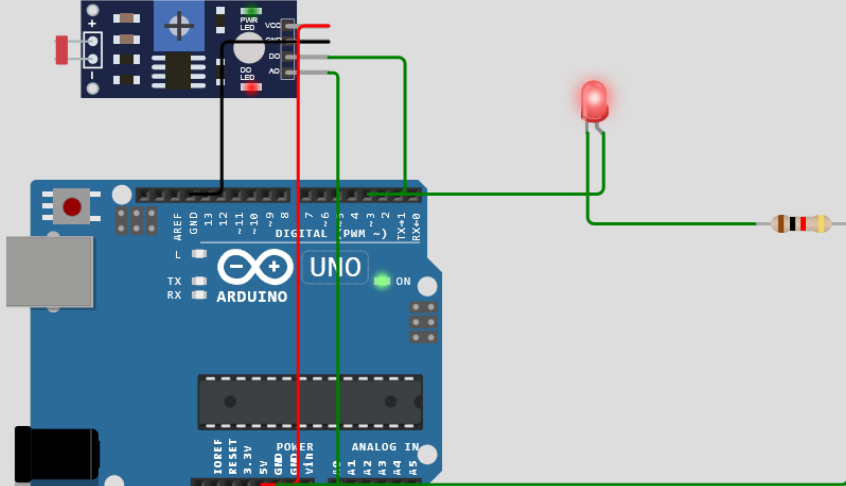


4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

```

1  int ledPin = 3;
2  int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่5
3  int val = 0;
4  void setup() {
5      pinMode(ledPin, OUTPUT); // sets the pin as output
6      Serial.begin(9600);
7  }
8  void loop() {
9      val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ LDR
10     Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
11     Serial.println(val); // พิมพ์ค่าของตัวแปร val
12     if (val < 500) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
13         digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
14     }
15     else {
16         digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
17     }
18     delay(1000);
19 }
20
                
```

Simulation



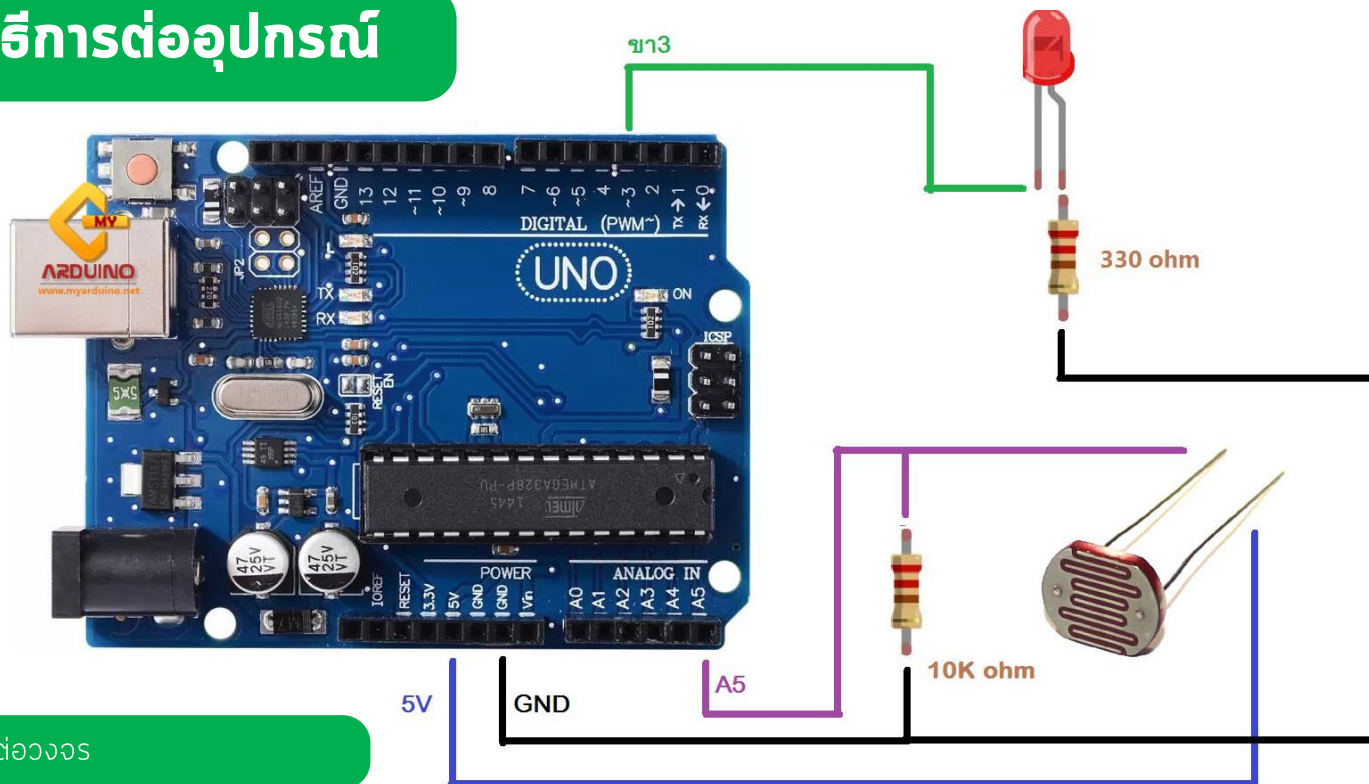
```

val = 257
val = 241
val = 303
val = 305
val = 336
val = 309
                
```

Val < 500 หลอดไฟติด
Val >= 500 หลอดไฟดับ

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

4.2.2 วิธีการต่ออุปกรณ์



ภาพที่ 4.9 วิธีการต่อวงจร

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

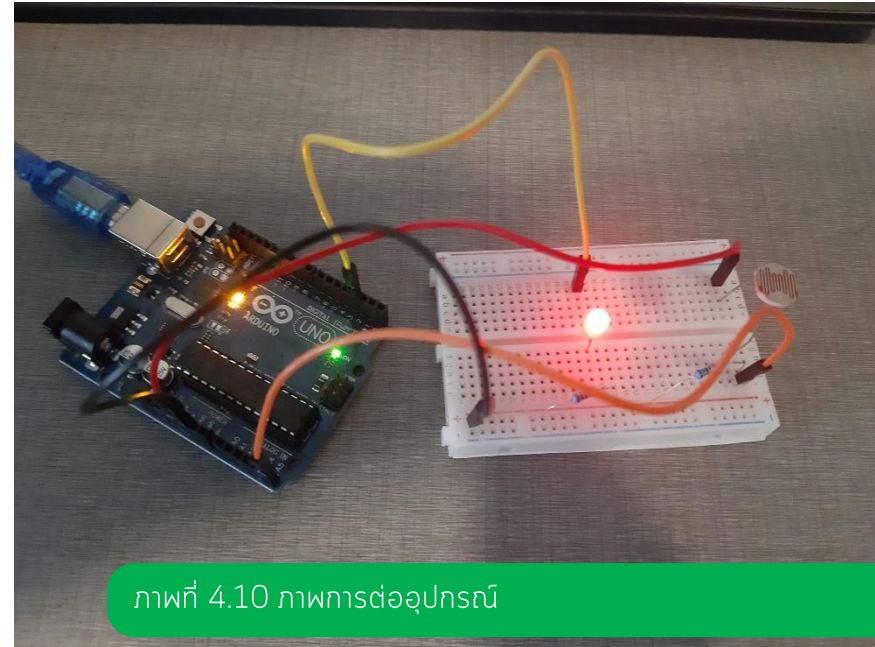
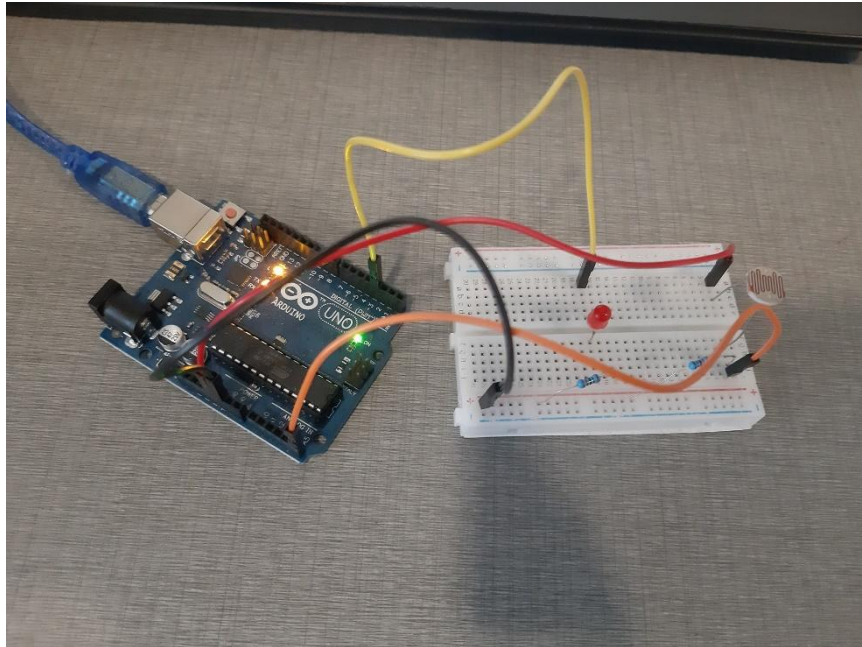
4.2.3 โปรแกรม (Code)

0 - 1023

```
int ledPin = 3;
int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนค่า analog ขาที่5
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT); // sets the pin as output
  Serial.begin(9600);
}
void loop() {
  val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ LDR
  Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
  Serial.println(val); // พิมพ์ค่าของตัวแปร val
  if (val < 100) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
  }
  else {
    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
  }
  delay(100);
}
```

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

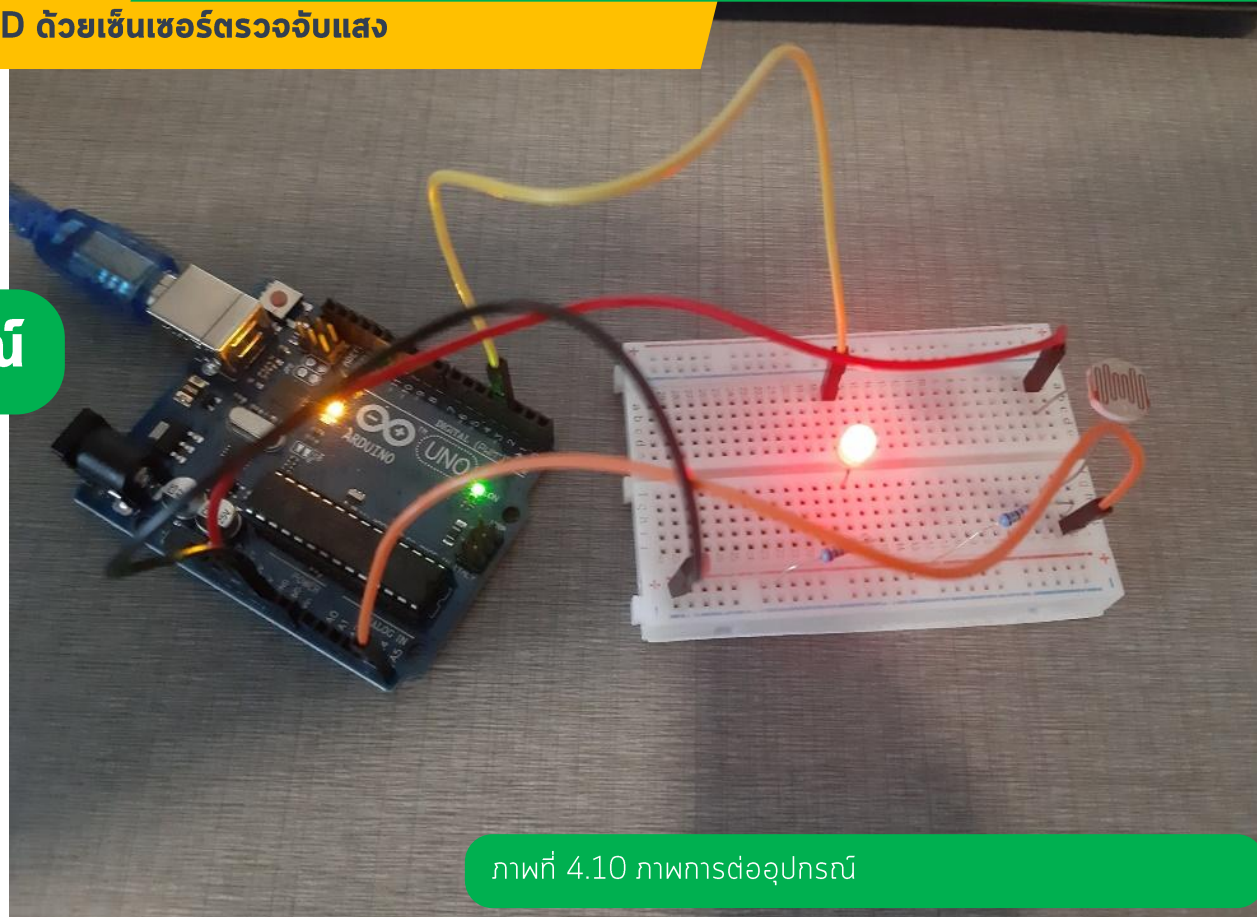
4.2.4 ภาพการต่ออุปกรณ์



ภาพที่ 4.10 ภาพการต่ออุปกรณ์

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

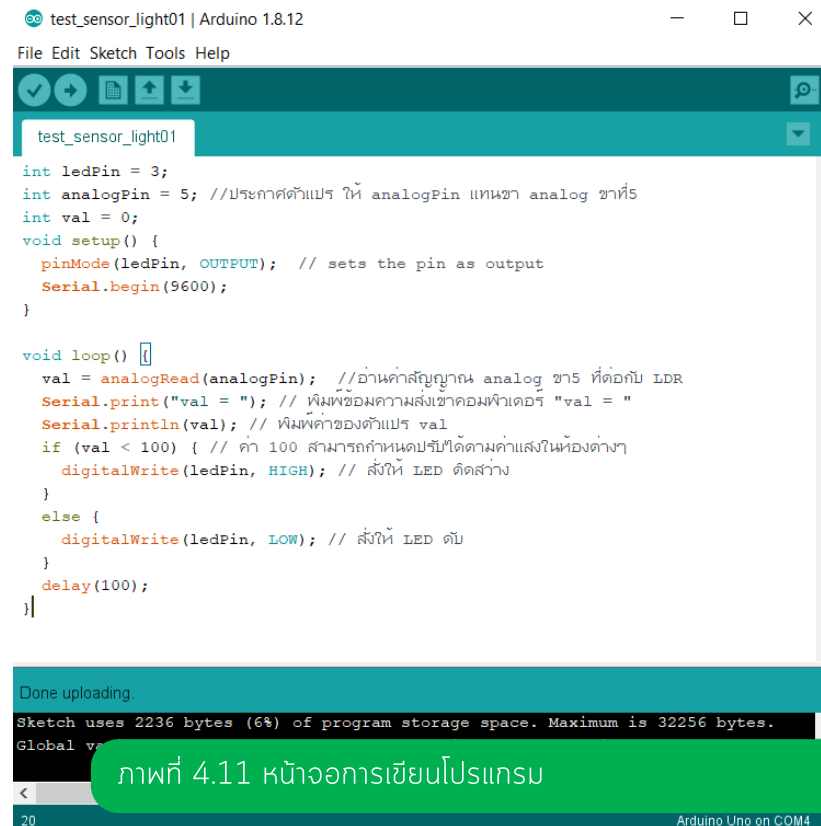
4.2.4 ภาพการต่ออุปกรณ์



ภาพที่ 4.10 ภาพการต่ออุปกรณ์

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

4.2.5 หน้าจตัวอย่าง Code



```

test_sensor_light01 | Arduino 1.8.12
File Edit Sketch Tools Help

test_sensor_light01

int ledPin = 3;
int analogPin = 5; //ประกาศตัวแปร ให้ analogPin แทนขา analog ขา5
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT); // sets the pin as output
  Serial.begin(9600);
}

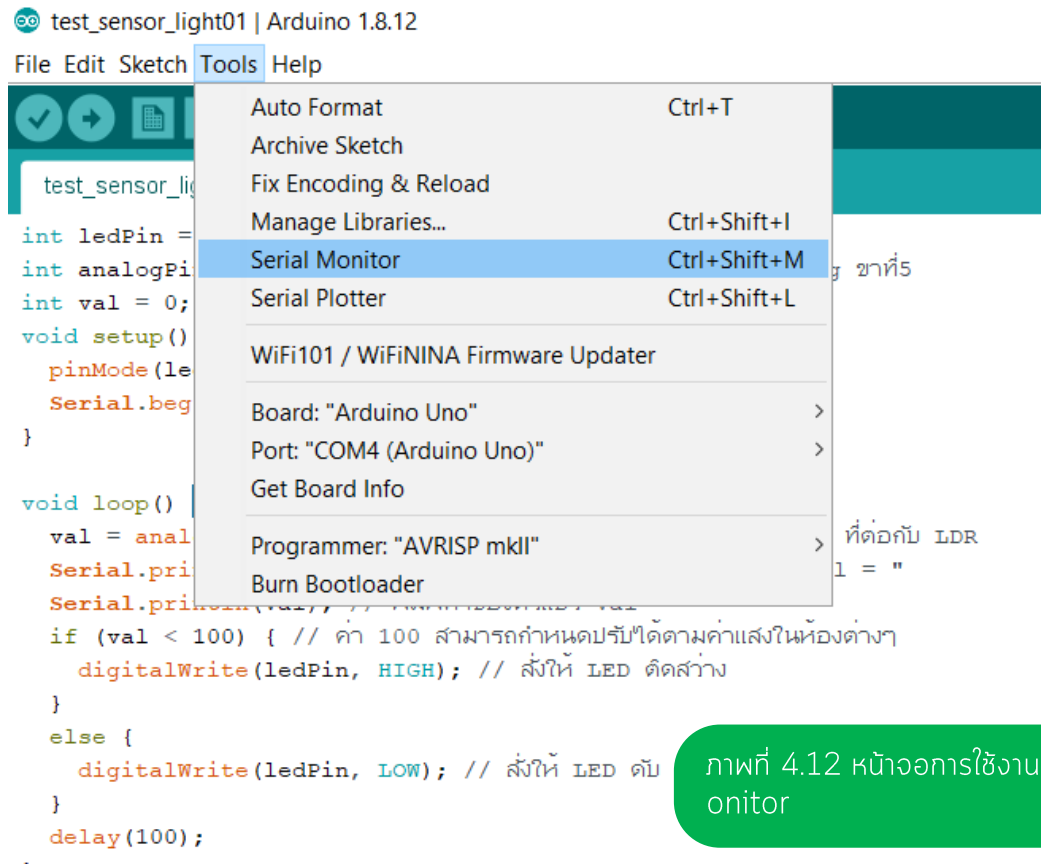
void loop() {
  val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ LDR
  Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
  Serial.println(val); // พิมพ์ค่าของตัวแปร val
  if (val < 100) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
  }
  else {
    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
  }
  delay(100);
}

Done uploading.
Sketch uses 2236 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables used: 0
20 Arduino Uno on COM4
  
```

ภาพที่ 4.11 หน้าจอการเขียนโปรแกรม

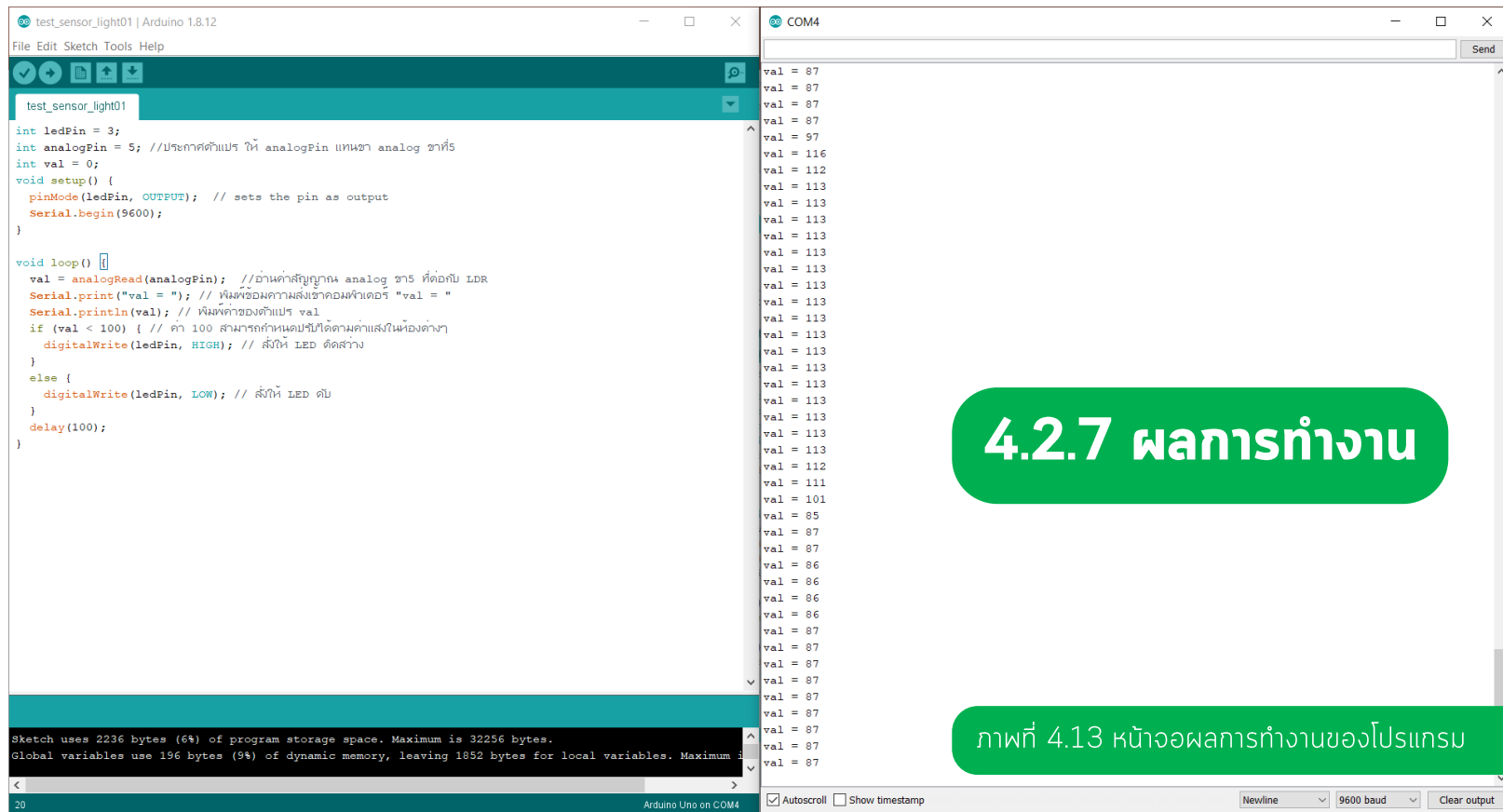
4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

4.2.6 การใช้งาน Tool >> Serial Monitor



ภาพที่ 4.12 หน้าจอการใช้งาน Serial Monitor

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง



The screenshot displays the Arduino IDE interface. The left pane shows the sketch 'test_sensor_light01' with the following code:

```
int ledPin = 3;
int analogPin = 5; //ประกาศตำแหน่ง pin ให้ analogPin แทนขา analog ขาที่5
int val = 0;
void setup() {
  pinMode(ledPin, OUTPUT); // sets the pin as output
  Serial.begin(9600);
}
void loop() {
  val = analogRead(analogPin); //อ่านค่าสัญญาณ analog ขา5 ที่ต่อกับ LDR
  Serial.print("val = "); // พิมพ์ข้อความส่งเข้าคอมพิวเตอร์ "val = "
  Serial.println(val); // พิมพ์ค่าของตัวแปร val
  if (val < 100) { // ค่า 100 สามารถกำหนดเป็นขีดตามค่าแสงในห้องต่างๆ
    digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
  }
  else {
    digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
  }
  delay(100);
}
```

The right pane shows the serial monitor output for COM4, displaying a series of 'val' values ranging from 85 to 116, indicating the sensor's readings over time.

Sketch uses 2236 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 196 bytes (9%) of dynamic memory, leaving 1852 bytes for local variables. Maximum is 2048 bytes.

Arduino Uno on COM4

Autoscroll Show timestamp Newline 9600 baud Clear output

4.2.7 ผลการทำงาน

ภาพที่ 4.13 หน้าจอผลการทำงานของโปรแกรม

4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

ตัวอย่างการต่อวงจร LDR (Light Dependent Resistor) กับบอร์ด Arduino เพื่อตรวจจับระดับแสงที่ต่างกัน โดยใช้ค่าความต้านทานของ LDR เพื่อตรวจสอบระดับแสงที่ภายนอก นี่คือการต่อวงจรและได้ตัวอย่าง:

อุปกรณ์ที่ต้องใช้:

- 1.Arduino Board (เช่น Arduino Uno)
- 2.LDR (Light Dependent Resistor)
- 3.Resistor 10K ohm
- 4.ตัวต้านทาน 220 ohm
- 5.LED (ในกรณีที่ต้องการให้มีการแสดงผล)

การต่อวงจร:

- 1.ต่อขา GND ของ Arduino กับขาต่อสัญญาณกลางของ LDR
- 2.ต่อขา A0 ของ Arduino กับขาต่อสัญญาณอีกข้างของ LDR
- 3.ต่อขา D13 ของ Arduino กับขายาวของ LED
- 4.ต่อขาสัญญาณของ LED กับตัวต้านทาน 220 ohm และต่อขาอีกข้างของตัวต้านทานกับ GND

4.2.7 ผลการทำงาน

ภาพที่ 4.13 หน้าจอผลการทำงานของโปรแกรม

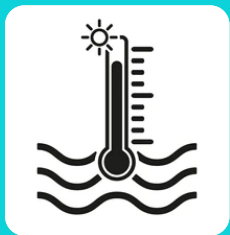
4.2 โปรแกรม เปิด-ปิด หลอด LED ด้วยเซ็นเซอร์ตรวจจับแสง

ตัวอย่างการต่อวงจร LDR (Light Dependent Resistor) กับบอร์ด Arduino เพื่อตรวจจับระดับแสงที่ต่างกัน โดยใช้ค่าความต้านทานของ LDR เพื่อตรวจสอบระดับแสงที่ภายนอก นี่คือการต่อวงจรและโค้ดตัวอย่าง:

```
// กำหนดขาที่เชื่อมต่อ LDR
const int ldrPin = A0;
// กำหนดขาที่เชื่อมต่อ LED
const int ledPin = 13;
void setup() {
  // กำหนดให้ขา LED เป็น Output
  pinMode(ledPin, OUTPUT);
  // เริ่มต้นให้ Serial Monitor เปิดใช้งาน
  Serial.begin(9600);
}
void loop() {
  // อ่านค่าแสงที่ LDR
  int lightValue = analogRead(ldrPin);
  // แสดงค่าความสว่างที่ได้จาก LDR บน Serial Monitor
  Serial.print("Light Value: ");
  Serial.println(lightValue);
  // ถ้าความสว่างมากกว่าหรือเท่ากับค่าที่กำหนด (ค่าสว่างเท่ากับ 500 คือเปิด LED)
  if (lightValue >= 500) {
    digitalWrite(ledPin, HIGH); // สั่งให้ LED เปิด
  } else {
    digitalWrite(ledPin, LOW); // สั่งให้ LED ปิด
  }
  delay(1000); // หน่วงเวลา 1 วินาที
}
```

4.2.7 ผลการทำงาน

ภาพที่ 4.13 หน้าจอผลการทำงานของโปรแกรม



4.3 โปรแกรม เซ็นเซอร์อุณหภูมิ และความชื้น

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.1 อุปกรณ์ที่ต้องใช้

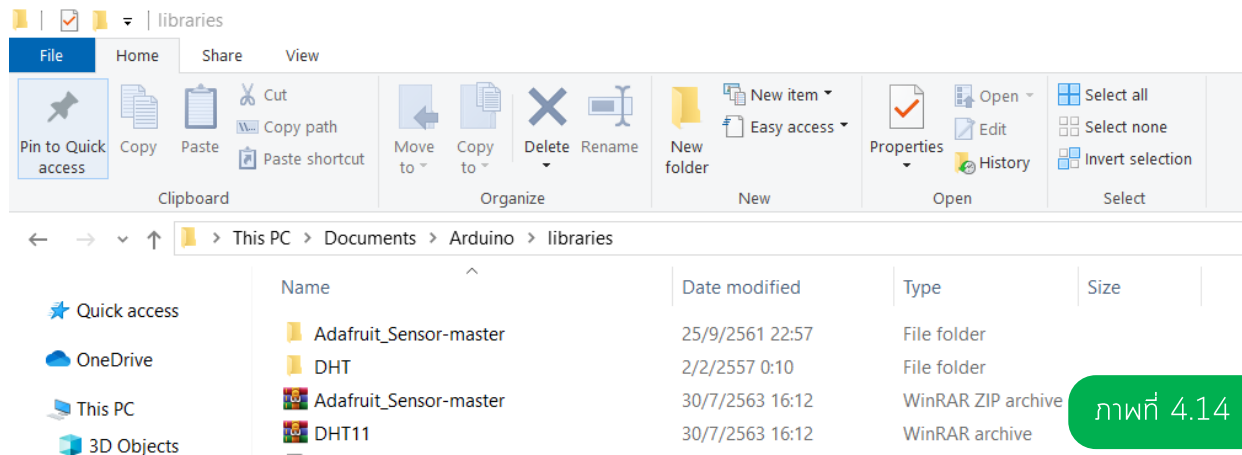
- Arduino UNO R3 พร้อม สายUSB
- เซ็นเซอร์วัด อุณหภูมิ และความชื้น (DHT22)
- บอร์ดทดลอง Breadboard
- Resistor ตัวต้านทาน 1 อัน
- สายไฟจัมเปอร์ ผุ้-ผุ้ 3 เส้น

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.2 ติดตั้ง Libraries Sensor วัดอุณหภูมิและความชื้น

ดาวน์โหลด Libraries ทั้ง 2 ไฟล์ และทำการแตกไฟล์ลงใน Folder: libraries

- <http://www.mediafire.com/download/6qh8q1g0kmokl4g/DHT11.rar>
- http://www.mediafire.com/file/1xqqnpk4g42wdug/Adafruit_Sensor-master.zip/file

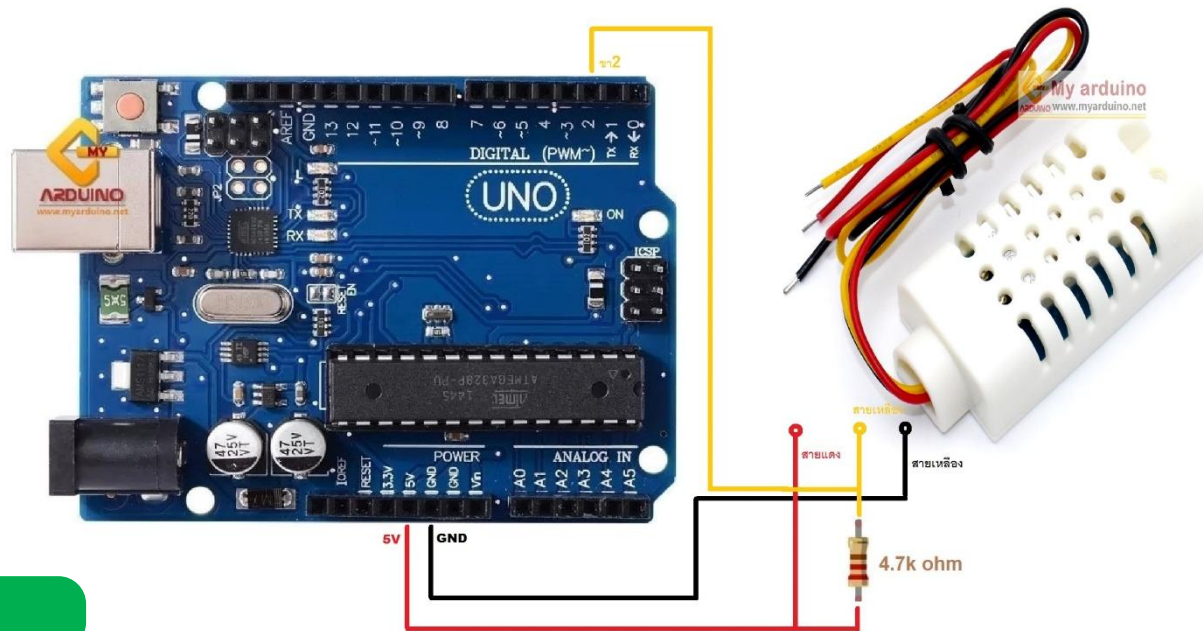


ภาพที่ 4.14 หน้าจอการติดตั้ง Libraries

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.3 วิธีการต่อวงจร

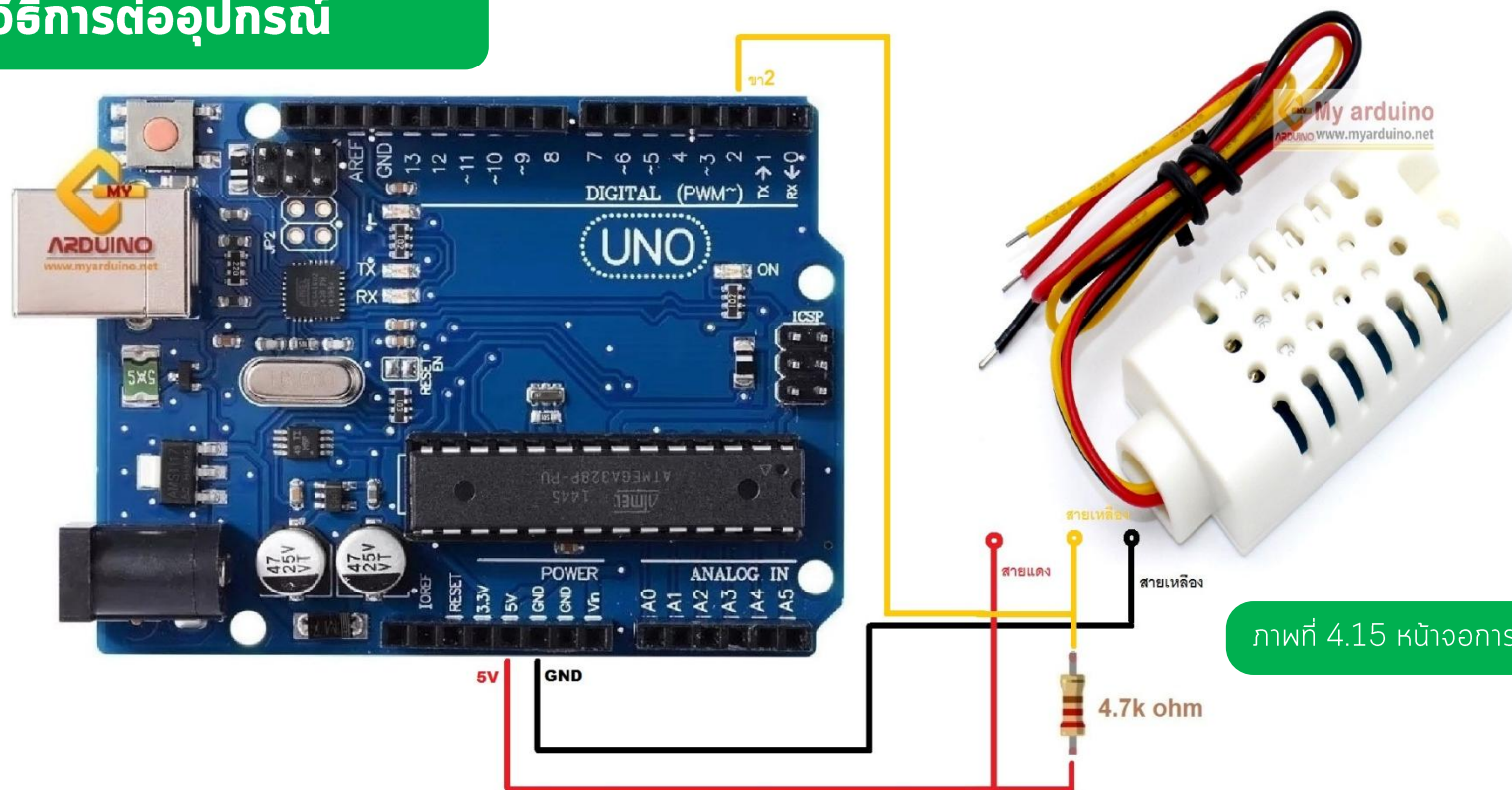
- ขา2 = ขาสีเหลือง
- GND = สายสีดำ
- 5V = สายสีแดง



ภาพที่ 4.15 หน้าจอการต่อวงจร

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

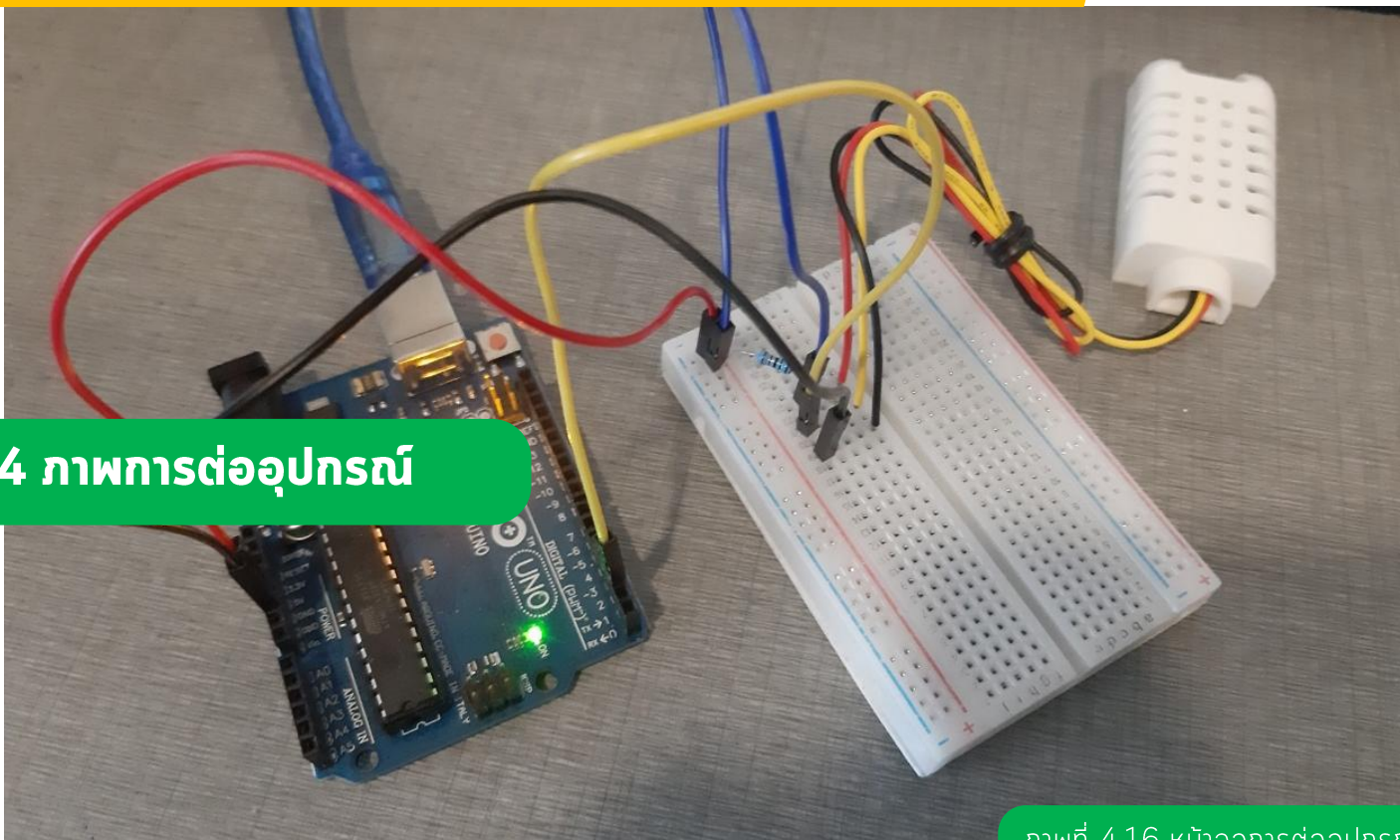
4.3.3 วิธีการต่ออุปกรณ์



ภาพที่ 4.15 หน้าจอการต่อวงจร

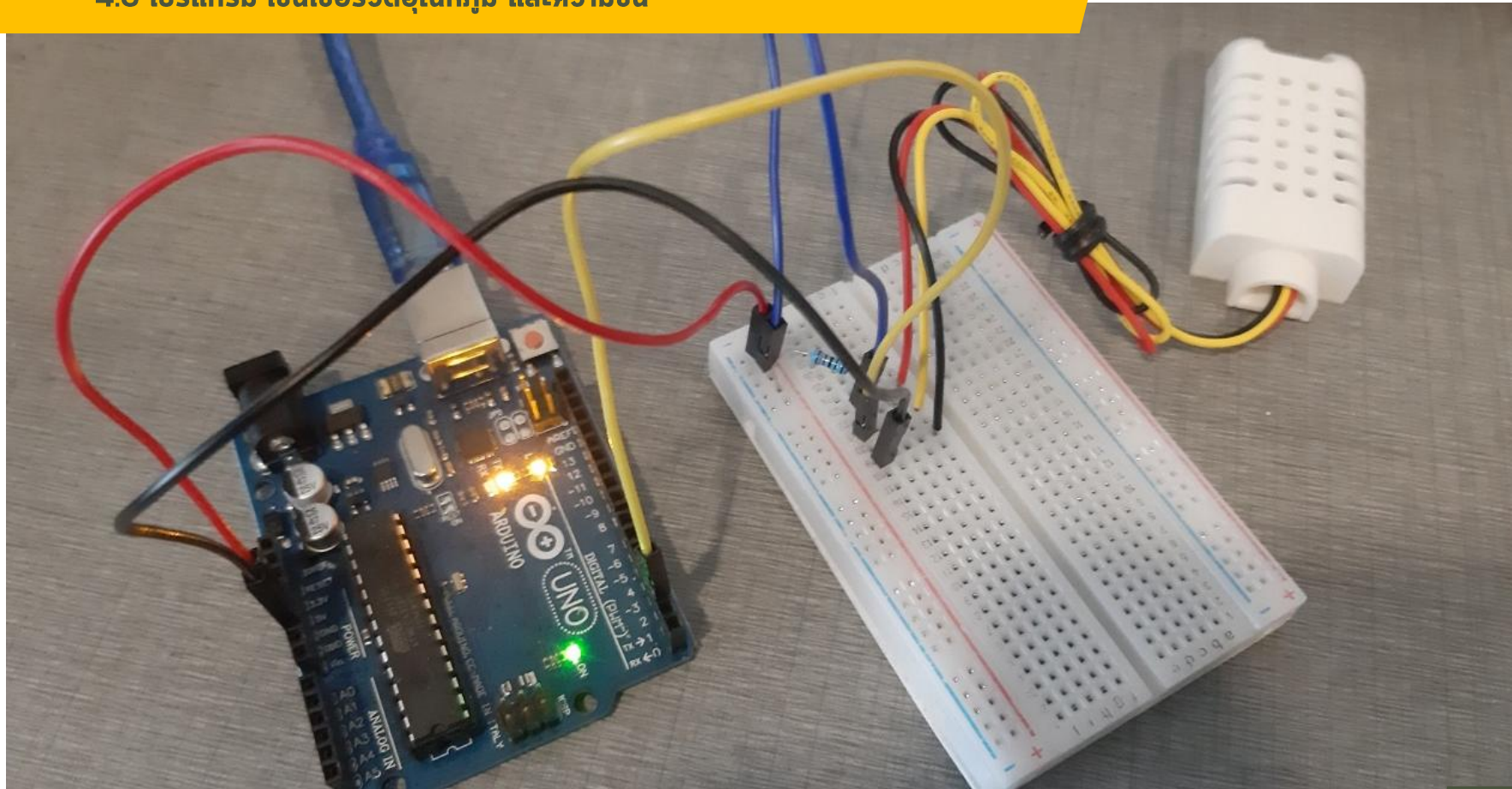
4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.4 ภาพการต่ออุปกรณ์



ภาพที่ 4.16 หน้าจอการต่ออุปกรณ์

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น



4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.4 โปรแกรม (Code)

1

```
#include "DHT.h"// เรียกใช้งาน libraries DHT
DHT dht;
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\t Humidity (%) \t Temperature (C)\t(F)");
};

dht.setup(2); // data pin 2
}
```

2

```
void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1); //แสดงค่าความชื้น
  Serial.print("\t\t");
  Serial.print(temperature, 1); //แสดงอุณหภูมิแบบเซลเซียส
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature), 1); //แสดงค่า
  อุณหภูมิแบบฟาเรนไฮต์
  delay(1000);
}
```

```
#include <DHT.h>
```

```
// กำหนดค่าที่เชื่อมต่อเซนเซอร์ DHT และชนิดของเซนเซอร์
```

```
#define DHTPIN 2
```

```
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  Serial.println("DHT Sensor Initializing...");
```

```
  dht.begin();
```

```
}
```

```
void loop() {
```

```
  delay(2000); // ค่าความถี่ในการอ่านข้อมูล (2 วินาที)
```

```
  // อ่านค่าความชื้นและอุณหภูมิ
```

```
  float humidity = dht.readHumidity();
```

```
  float temperature = dht.readTemperature();
```

```
  float fahrenheit = dht.readTemperature(true);
```

```
  // ตรวจสอบข้อผิดพลาดในการอ่านค่า
```

```
  if (isnan(humidity) || isnan(temperature) || isnan(fahrenheit)) {
```

```
    Serial.println("Failed to read from DHT sensor!");
```

```
    return;
```

```
}
```

```
  // แสดงค่าความชื้นและอุณหภูมิ
```

```
  Serial.print("Humidity: ");
```

```
  Serial.print(humidity);
```

```
  Serial.print(" %\n");
```

```
  Serial.print("Temperature: ");
```

```
  Serial.print(temperature);
```

```
  Serial.print(" °C ");
```

```
  Serial.print(fahrenheit);
```

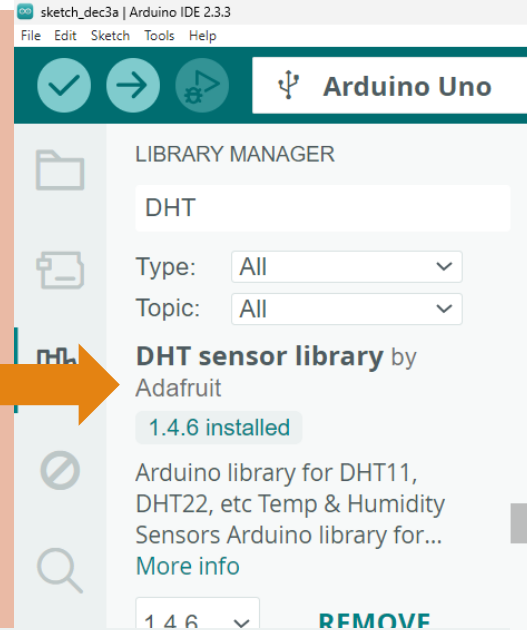
```
  Serial.println(" °F");
```

```
}
```

1. ไปที่เมนู Tools

2. Manage Libraries

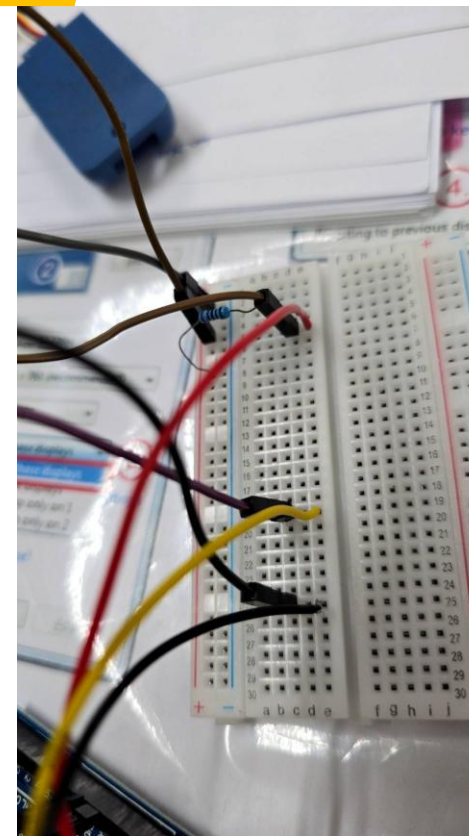
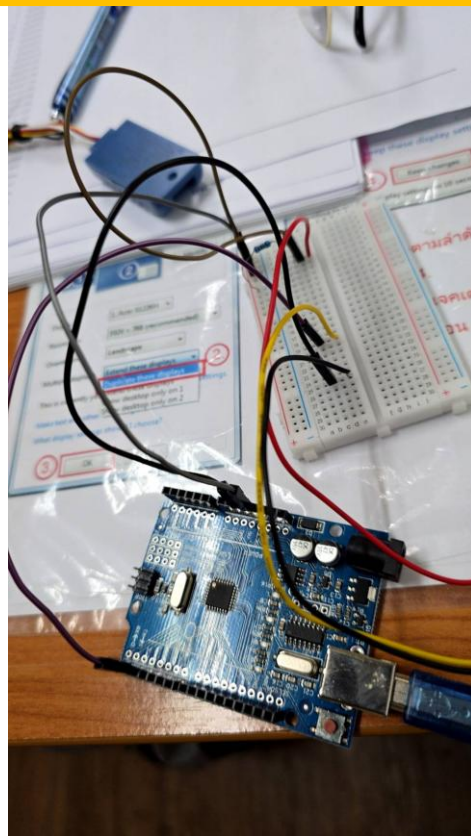
3



Not connected. Select a board and a port to connect automatically.

Humidity: 38.30 %	Temperature: 28.70 °C 83.66 °F
Humidity: 37.90 %	Temperature: 28.70 °C 83.66 °F
Humidity: 38.60 %	Temperature: 28.70 °C 83.66 °F
Humidity: 39.00 %	Temperature: 28.60 °C 83.48 °F
Humidity: 39.10 %	Temperature: 28.60 °C 83.48 °F
Humidity: 39.00 %	Temperature: 28.60 °C 83.48 °F
Humidity: 38.90 %	Temperature: 28.60 °C 83.48 °F
Humidity: 38.80 %	Temperature: 28.70 °C 83.66 °F
Humidity: 39.10 %	Temperature: 28.70 °C 83.66 °F
Humidity: 39.00 %	Temperature: 28.60 °C 83.48 °F
Humidity: 39.40 %	Temperature: 28.70 °C 83.66 °F

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น



Build failed!

```
sketch.ino:1:10: fatal error: DHT.h: No such file or directory  
#include "DHT.h" // เรียกใช้งาน libraries DHT
```

^~~~~~

compilation terminated.

Error during build: exit status 1

INSTALL "DHT SENSOR LIBRARY" LIBRARY

CLOSE

WOKWI



SAVE



SHARE

sketch.ino

diagram.json

libraries.txt

Library Manager

Project Libraries



Installed Libraries



DHT sensor library

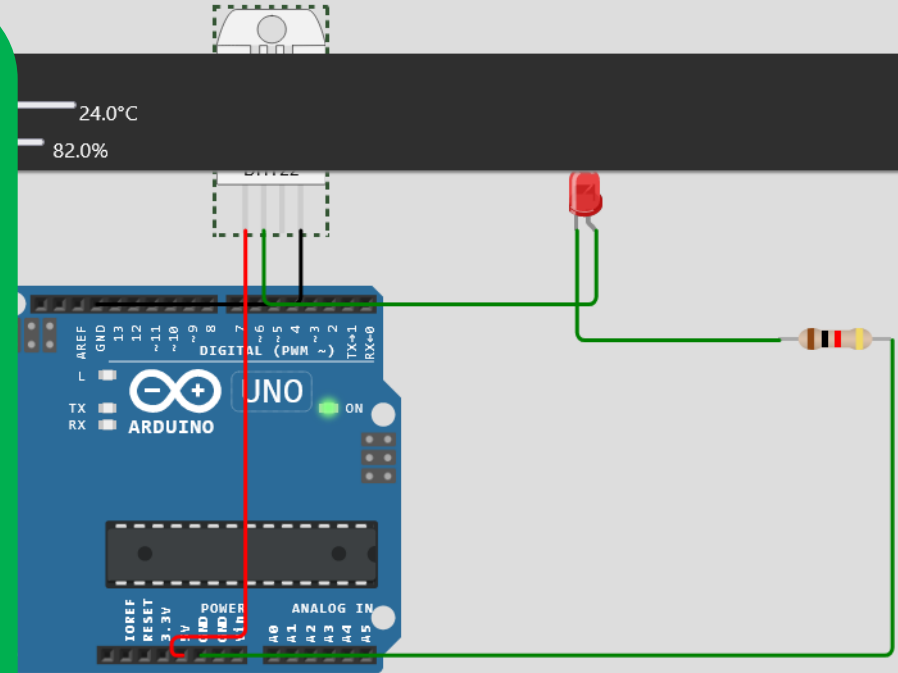


Wokwi

[illegible]

```
#include "DHT.h"
#define DHT_PIN 2
#define DHT_TYPE DHT22
```

```
DHT dht(DHT_PIN, DHT_TYPE);
void setup() {
  Serial.begin(9600);
  Serial.println("DHT22 Sensor Test");
  dht.begin();
}
void loop() {
  delay(2000); // หน่วงเวลา 2 วินาที เพื่อให้เซ็นเซอร์ทำงาน
  float humidity = dht.readHumidity(); // อ่านค่าความชื้น
  float temperature = dht.readTemperature(); // อ่านค่าอุณหภูมิ (เซลเซียส)
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("%  Temperature: ");
  Serial.print(temperature);
  Serial.println("'C");
  delay(1000);
}
```

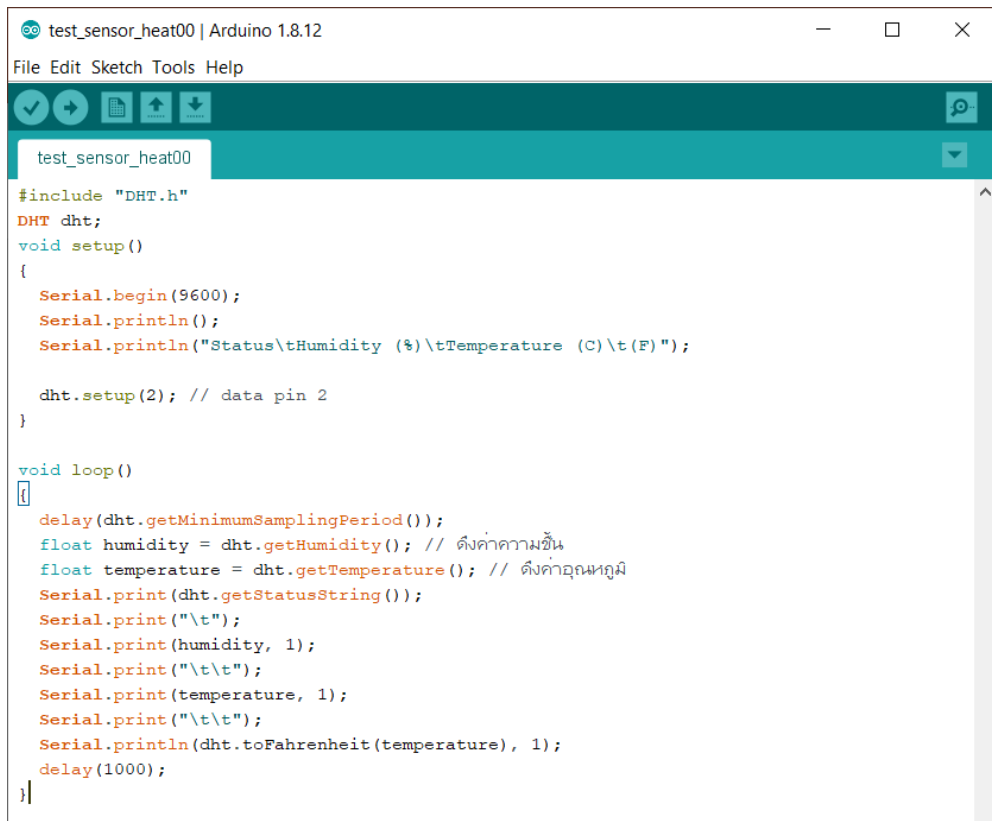


```
Temperature: 24.00°C
Temperature: 24.00°C
Temperature: 24.00°C
Temperature: 24.00°C
Humidity: 40.00% Temperature: 24.00°C
Humidity: 40.00% Temperature: 24.00°C
Humidity: 82.00% Temperature: 24.00°C
```

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.4 หน้าจอตาราง Code

ภาพที่ 4.17 หน้าจอการเขียนโปรแกรม



```
test_sensor_heat00 | Arduino 1.8.12
File Edit Sketch Tools Help

test_sensor_heat00

#include "DHT.h"
DHT dht;
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");

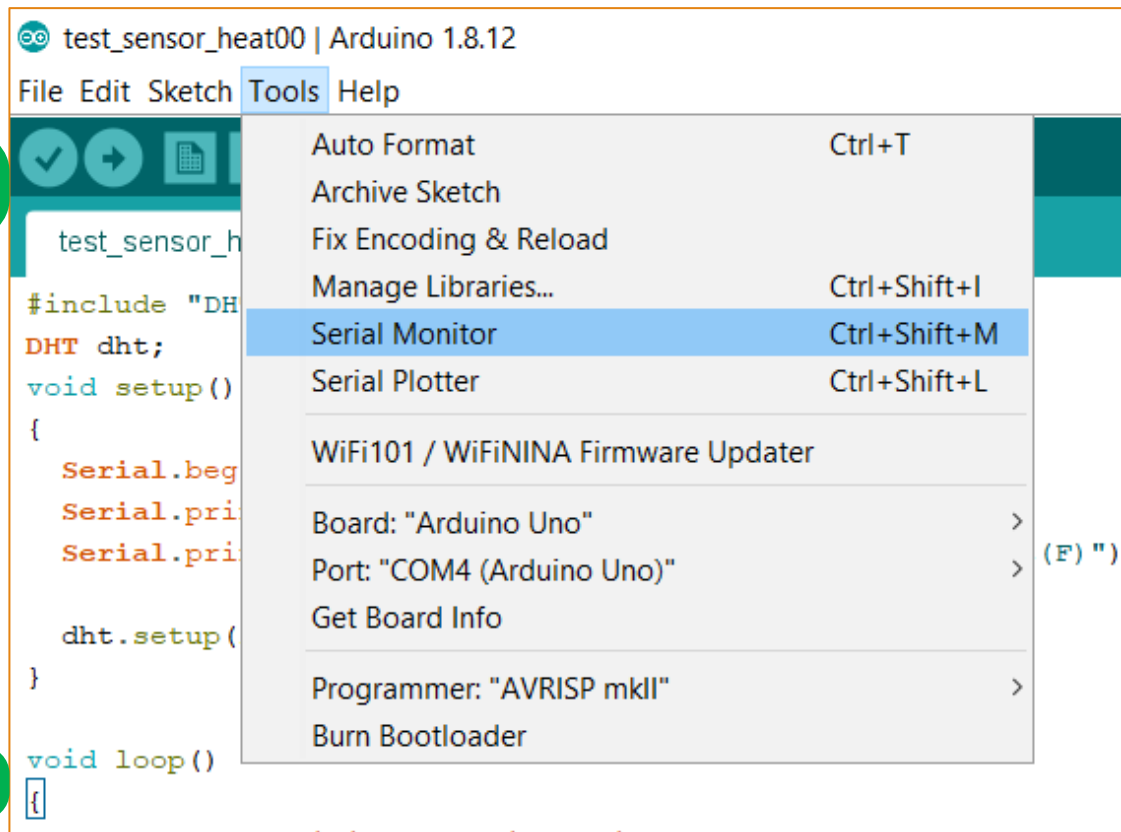
  dht.setup(2); // data pin 2
}

void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature), 1);
  delay(1000);
}
```

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

4.3.4 หน้าจอ Serial Monitor

เลือก Tools > Serial Monitor



ภาพที่ 4.18 หน้าจอใช้งาน Serial Monitor

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น

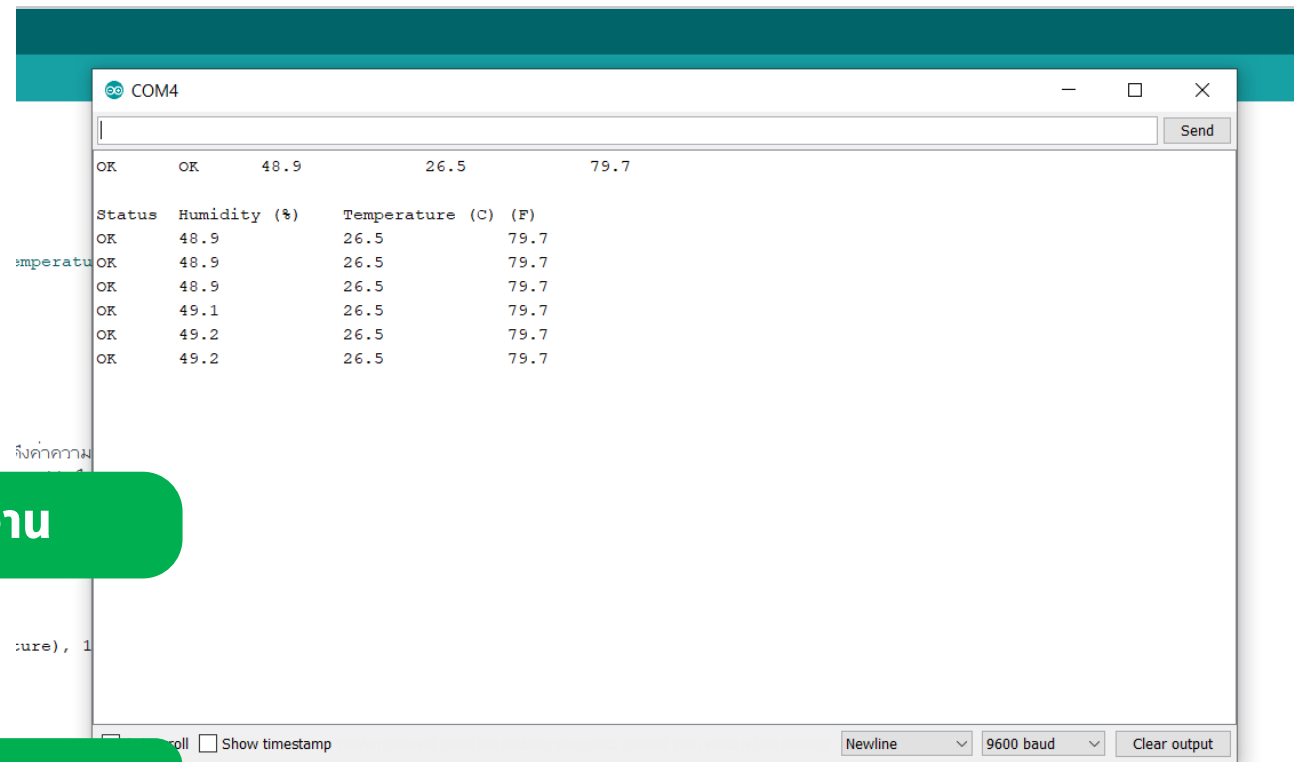
```
// กำหนดขาที่เชื่อมต่อ LDR
const int ldrPin = A0;
// กำหนดขาที่เชื่อมต่อ LED
const int ledPin = 13;

void setup() {
  // กำหนดให้ขา LED เป็น Output
  pinMode(ledPin, OUTPUT);
  // เริ่มต้นให้ Serial Monitor เปิดใช้งาน
  Serial.begin(9600);
}

void loop() {
  // อ่านค่าแสงที่ LDR
  int lightValue = analogRead(ldrPin);
  // แสดงค่าความสว่างที่ได้จาก LDR บน Serial Monitor
  Serial.print("Light Value: ");
  Serial.println(lightValue);
  // ถ้าความสว่างมากกว่าหรือเท่ากับค่าที่กำหนด (ค่าสว่างเท่ากับ 500 คือเปิด LED)
  if (lightValue >= 500) {
    digitalWrite(ledPin, HIGH); // สั่งให้ LED เปิด
  } else {
    digitalWrite(ledPin, LOW); // สั่งให้ LED ปิด
  }
  delay(1000); // หยุดเวลา 1 วินาที
}
```

ภาพที่ 4.18 หน้าจอใช้งาน Serial Monitor

4.3 โปรแกรม เซ็นเซอร์วัดอุณหภูมิ และความชื้น



4.3.5 ผลลัพธ์การทำงาน

ภาพที่ 4.19 หน้าจอผลการทำงาน



4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.1 อุปกรณ์ที่ต้องใช้

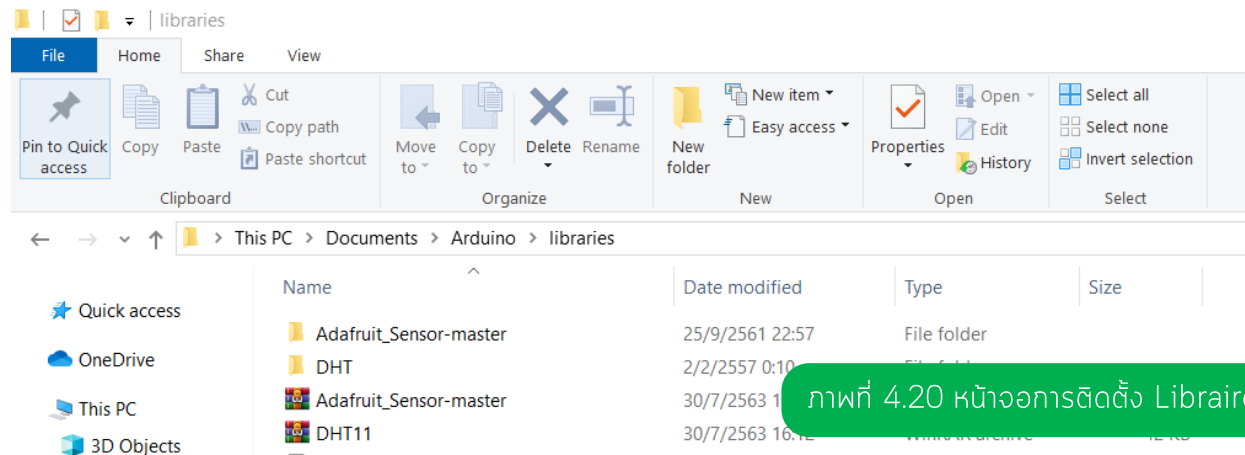
- **Arduino UNO R3 พร้อม สายUSB**
- **เซ็นเซอร์วัด อุณหภูมิ และความชื้น**
- **LED ขนาด 5mm สีแดง 1 ดวง**
- **บอร์ดทดลอง Breadboard**
- **Resistor ตัวต้านทาน 2 อัน**
- **สายไฟจัมเปอร์ ผุ้-ผุ้ 6 เส้น**

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.2 ติดตั้ง Libraries Sensor วัดอุณหภูมิและความชื้น

ดาวน์โหลด Libraries ทั้ง 2 ไฟล์ และทำการแตกไฟล์ลงใน Folder: libraries

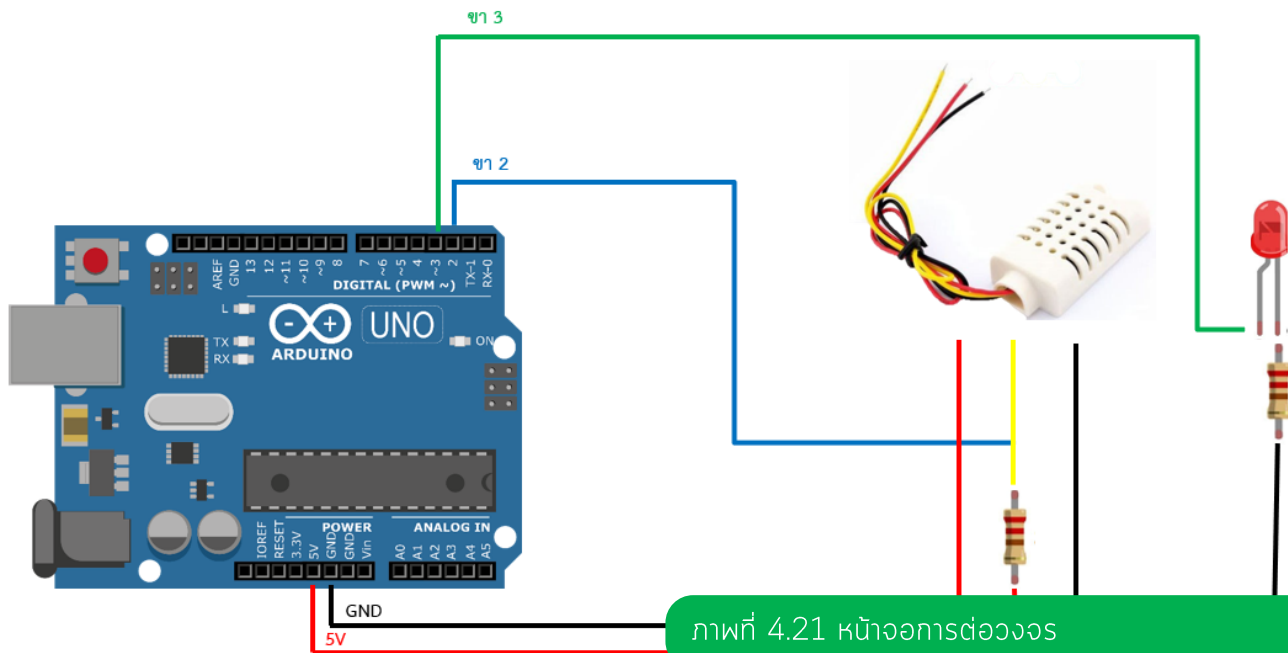
- <http://www.mediafire.com/download/6qh8q1g0kmokl4g/DHT11.rar>
- http://www.mediafire.com/file/1xqqnpk4g42wdug/Adafruit_Sensor-master.zip/file



4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.2 วิธีการต่อวงจร

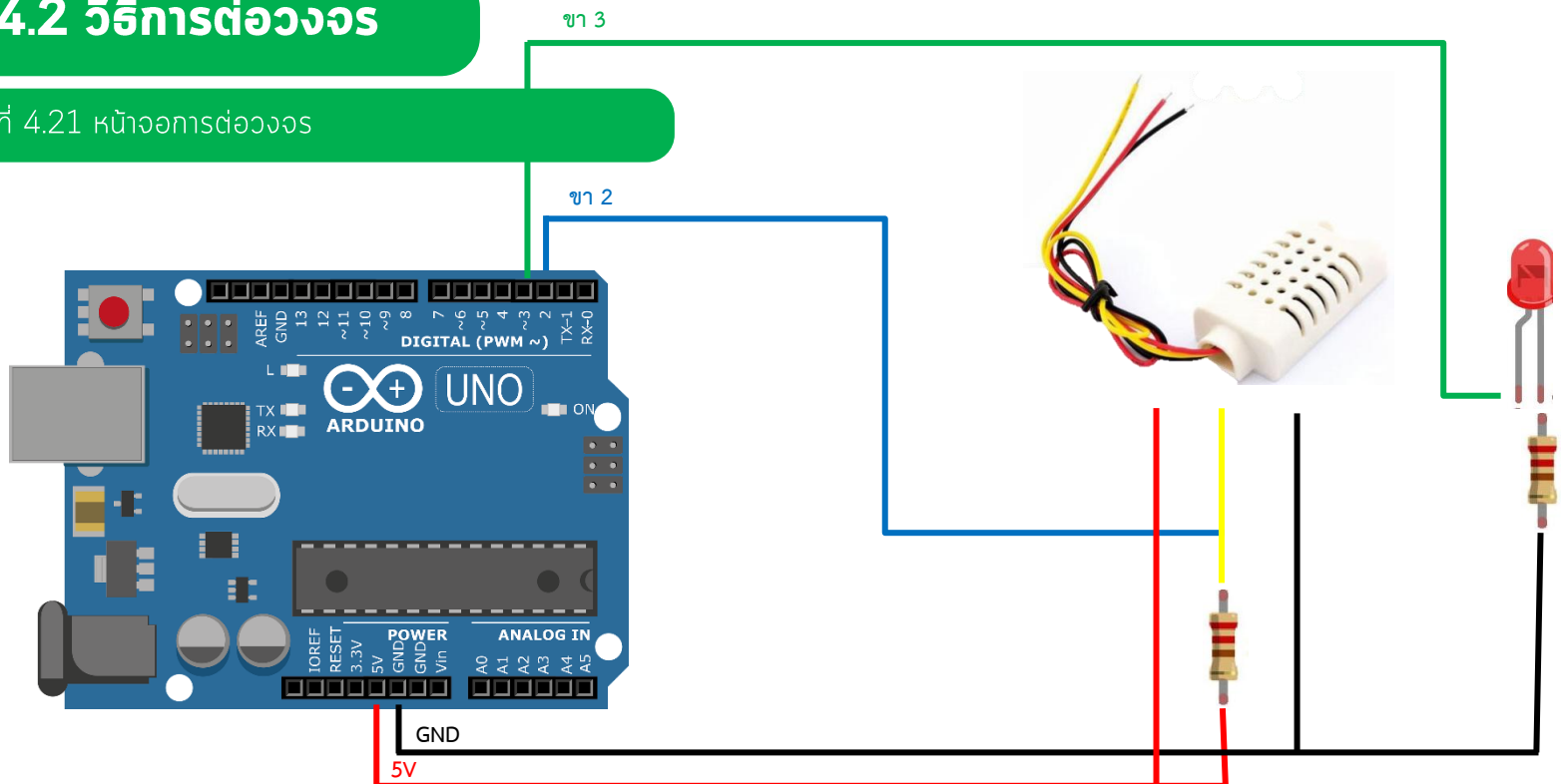
- ขา2 = ขาสีเหลือง
- ขา3 = LED1
- GND = สายสีดำ
- 5V = สายสีแดง



4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

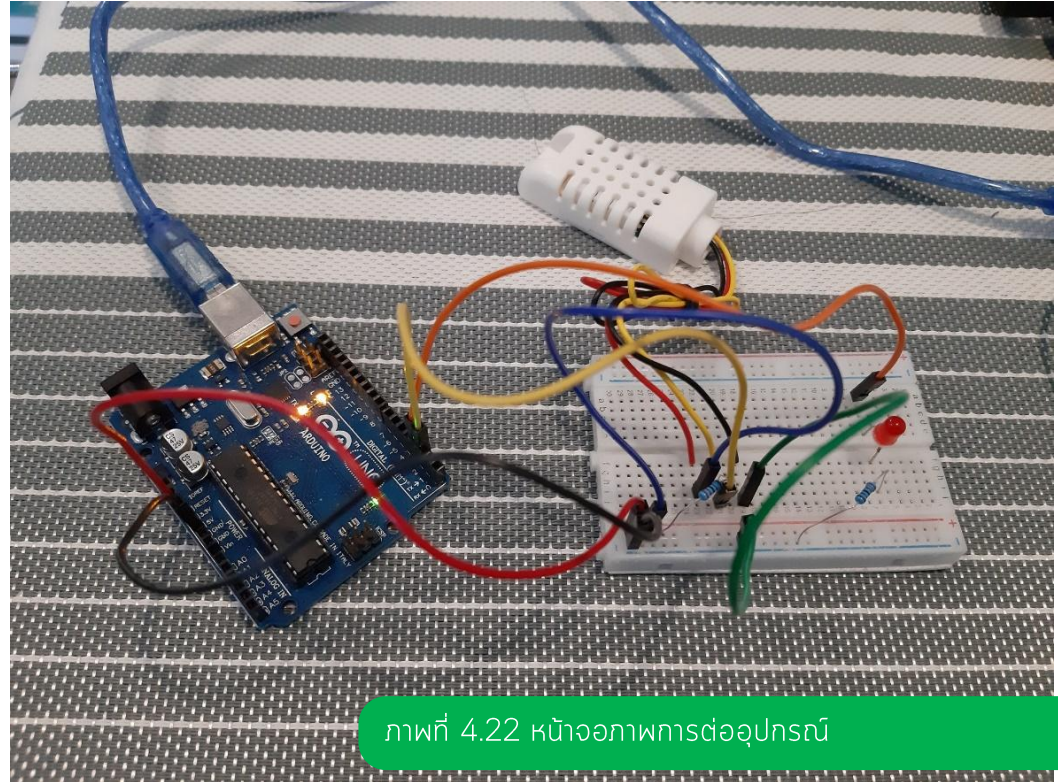
4.4.2 วิธีการต่อวงจร

ภาพที่ 4.21 หน้าจอการต่อวงจร



4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.3 ภาพการต่อวงจร



ภาพที่ 4.22 หน้าจอภาพการต่ออุปกรณ์

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.4 โปรแกรม (Code)

1

```
#include "DHT.h"// เรียกใช้งาน libraries DHT
DHT dht;
int led1 = 3; //ประกาศตัวแปรหลอด led1 มีค่าเท่ากับ 3
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
  pinMode(led1, OUTPUT);
  dht.setup(2); // data pin 2
}
```

2

```
void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1); //แสดงค่าความชื้น
  Serial.print("\t\t");
  Serial.print(temperature, 1); //แสดงอุณหภูมิแบบเซลเซียส
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature), 1); //แสดงค่าอุณหภูมิ
  แบบฟาเรนไฮต์
  delay(1000);

  if (temperature < 25){
    digitalWrite(led1, HIGH);
  } else {
    digitalWrite(led1, LOW);
  }
}
```

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.5 หน้าจอตาราง Code

ตั้งเงื่อนไข ถ้าอุณหภูมิต่ำกว่า 25 องศา
ให้หลอด LED เปิดไฟ



```
test_sensor_heat | Arduino 1.8.12
File Edit Sketch Tools Help

test_sensor_heat

#include "DHT.h"
DHT dht;
int led1 = 3;
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");

  pinMode(led1, OUTPUT);

  dht.setup(2); // data pin 2
}

void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t");
  Serial.print(temperature, 1);
  Serial.print("\t");
  Serial.println(dht.toFahrenheit(temperature), 1);
  delay(1000);

  if (temperature < 25) {
    digitalWrite(led1, HIGH);
  } else {
    digitalWrite(led1, LOW);
  }
}

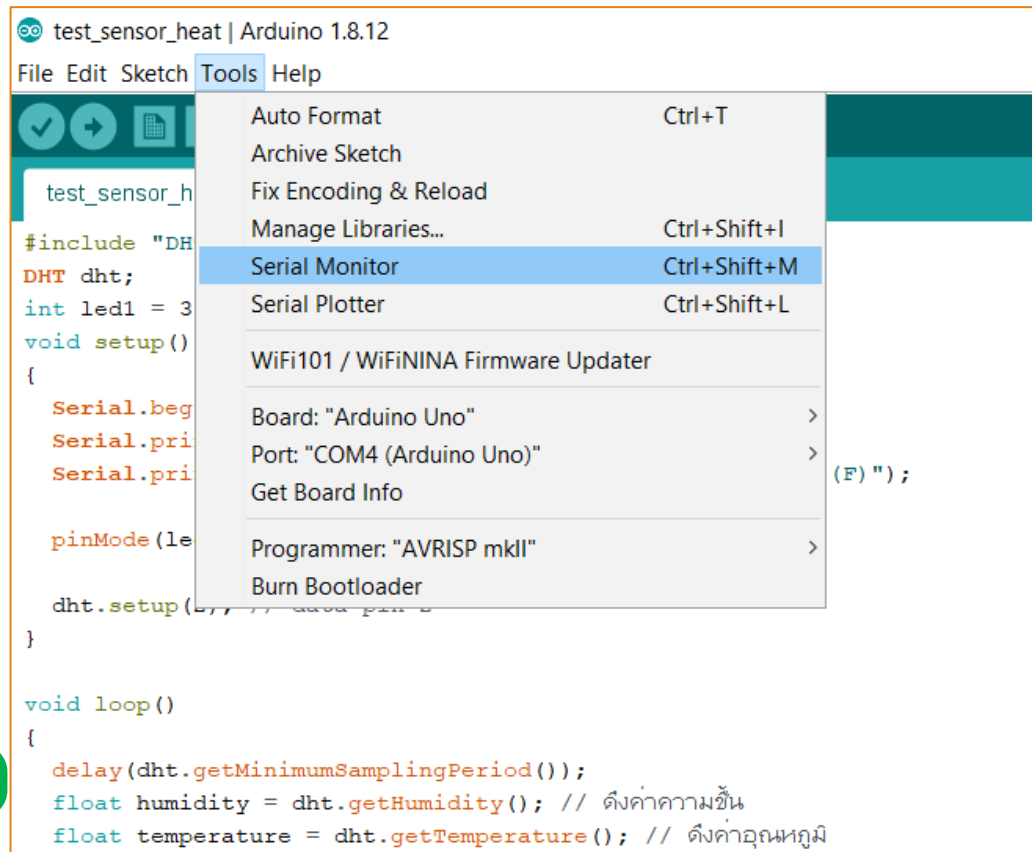
Sketch uses 4884 bytes (15%) of program storage space. Maximum is 32256 bytes.
Global variables use 279 bytes (13%) of dynamic memory, leaving 1769 bytes for local variables.
```

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.6 การใช้ Serial Monitor

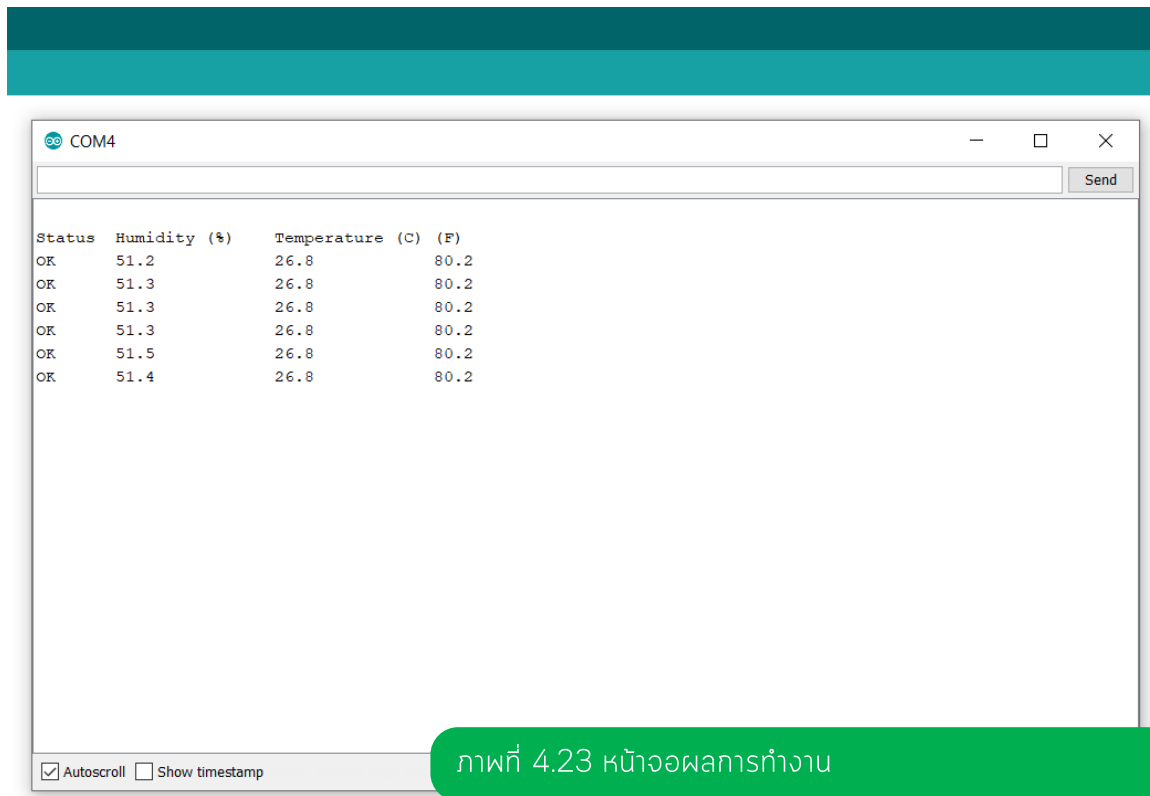
เลือก Tools > Serial Monitor

ภาพที่ 4.22 หน้าจอการใช้งาน Serial Monitor



4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

4.4.7 หน้าจอแสดง ความชื้น และ อุณหภูมิ



ภาพที่ 4.23 หน้าจอผลการทำงาน

4.4 โปรแกรม เปิด-ปิด หลอดไฟ LED อัตโนมัติ ด้วยอุณหภูมิ

WOKWI

SAVE

SHARE

sketch.ino

diagram.json

libraries.txt

Library Manager

```

20 Serial.print( Humidity: );
21 Serial.print(humidity);
22 Serial.print("% Temperature: ");
23 Serial.print(temperature);
24 Serial.println("°C");
25 if(temperature >25) {
26   digitalWrite(3,HIGH);
27   delay(3000);
28 } else {
29   digitalWrite(3,LOW);
30   delay(3000);
31 }
32
33 delay(1000);
34 }
35

```

Simulation

↺

⏏

⏸

DHT22 Sensor Test

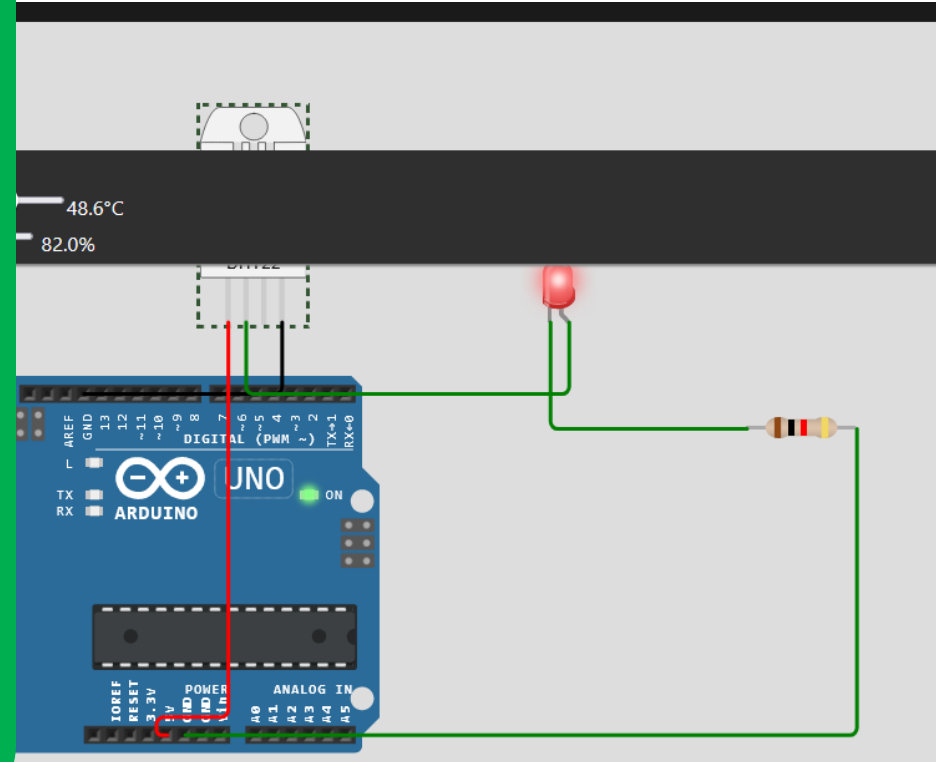
Humidity: 82.00% Temperature: 24.00°C

Humidity: 82.00% Temperature: 48.60°C

WOKWI : DHT22 & LED

```
#include "DHT.h"
#define DHT_PIN 2
#define DHT_TYPE DHT22
DHT dht(DHT_PIN, DHT_TYPE);
void setup() {
  Serial.begin(9600);
  Serial.println("DHT22 Sensor Test");
  dht.begin();
}
void loop() {
  delay(2000); // รอเวลา 2 วินาที เพื่อให้เซ็นเซอร์ทำงาน
  float humidity = dht.readHumidity(); // อ่านค่าความชื้น
  float temperature = dht.readTemperature(); // อ่านค่าอุณหภูมิ (เซลเซียส)

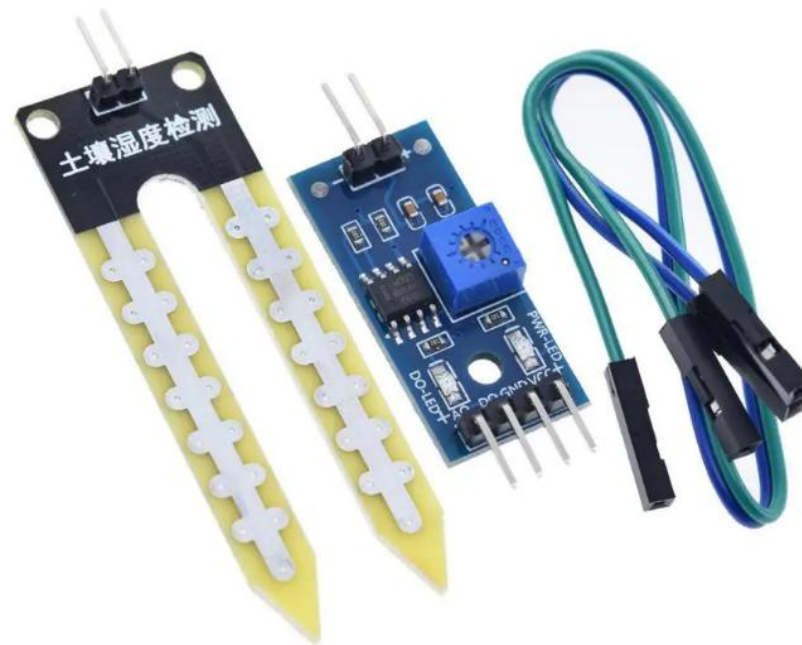
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print("% Temperature: ");
  Serial.print(temperature);
  Serial.println("°C");
  if(temperature > 25) {
    digitalWrite(3,HIGH);
    delay(3000);
  } else {
    digitalWrite(3,LOW);
    delay(3000);
  }
  delay(1000);
}
```



4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

soil moisture sensor

4.5.1 เซ็นเซอร์วัดความชื้น ในดิน



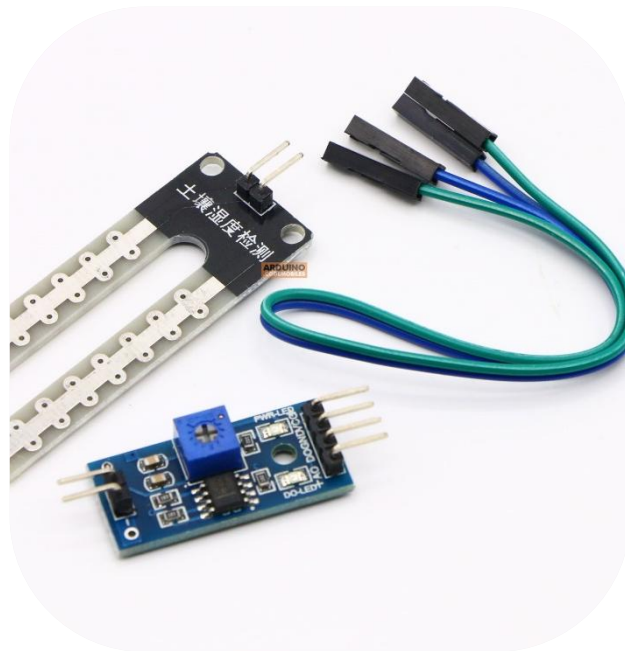
ภาพที่ 4.23 เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

soil module sensor

"soil module sensor" หมายถึงเซ็นเซอร์หรือเซ็นเซอร์โมดูลที่ใช้ในการวัดค่าทางธรรมชาติของดิน เช่น ความชื้นในดิน, อุณหภูมิดิน, ระดับน้ำในดิน เป็นต้น การวัดค่าเหล่านี้ช่วยให้เกษตรกรและผู้ที่ทำสวนหรือปลูกพืชสามารถตรวจสอบสภาพดินและเพิ่มประสิทธิภาพในการจัดการปลูกพืชให้เหมาะสมและเกิดผลผลิตที่ดีขึ้นได้

เซ็นเซอร์โมดูลดินส่วนใหญ่มักมีรูปแบบที่ติดตั้งในดินหรือประกอบในตู้เครื่อง และสามารถเชื่อมต่อกับระบบอิเล็กทรอนิกส์หรืออุปกรณ์อื่นๆ เพื่อรับค่าเซ็นเซอร์และนำไปใช้ในการตรวจวัดหรือบันทึกข้อมูล ยังมีรูปแบบเซ็นเซอร์ที่ไร้สาย (Wireless) ที่ให้ความสะดวกในการติดตั้งและใช้งาน รวมถึงช่วยให้สามารถตรวจสอบสภาพดินจากระยะไกลได้ ซึ่งเป็นประโยชน์สำหรับการจัดการพื้นที่กว้างใหญ่หรือสวนที่ใหญ่ รวมถึงใช้ในการวิจัยและพัฒนาทางด้านการเกษตรและสิ่งแวดล้อมอย่างกว้างขวาง

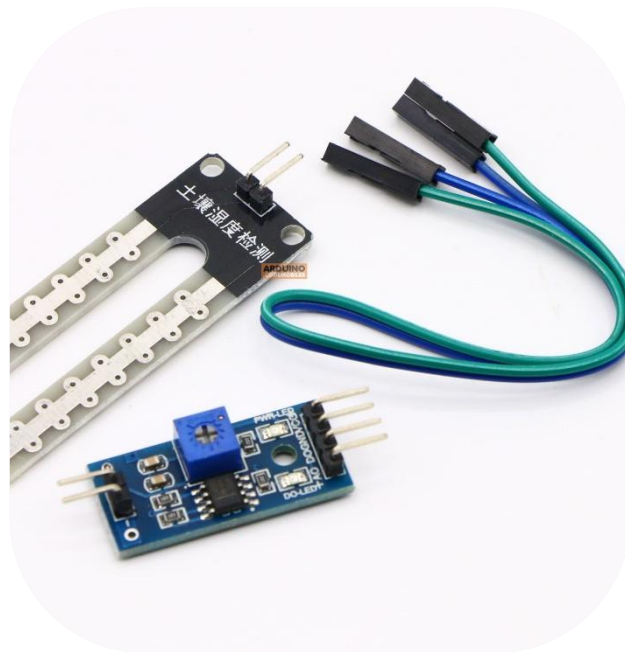


4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

soil module sensor

อุปกรณ์ที่ต้องใช้:

- 1.Arduino Board (เช่น Arduino Uno)
- 2.LDR (Light Dependent Resistor)
- 3.Resistor 10K ohm
- 4.ตัวต้านทาน 220 ohm
- 5.LED (ในกรณีที่ต้องการให้มีการแสดงผล)

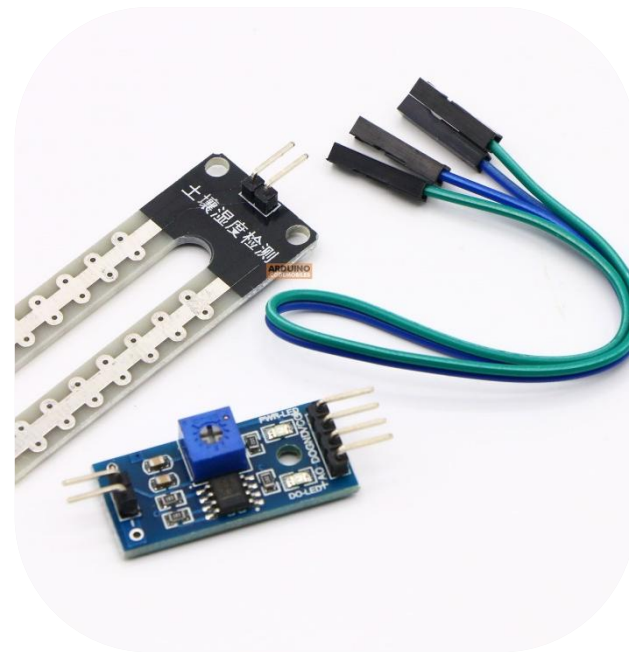


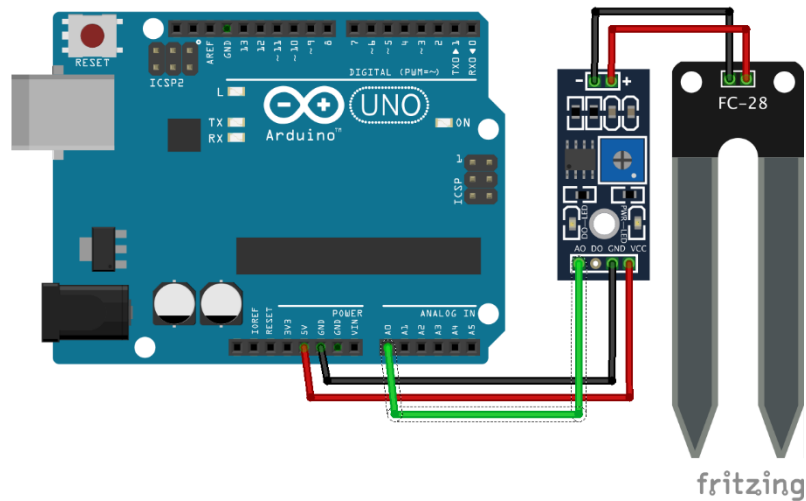
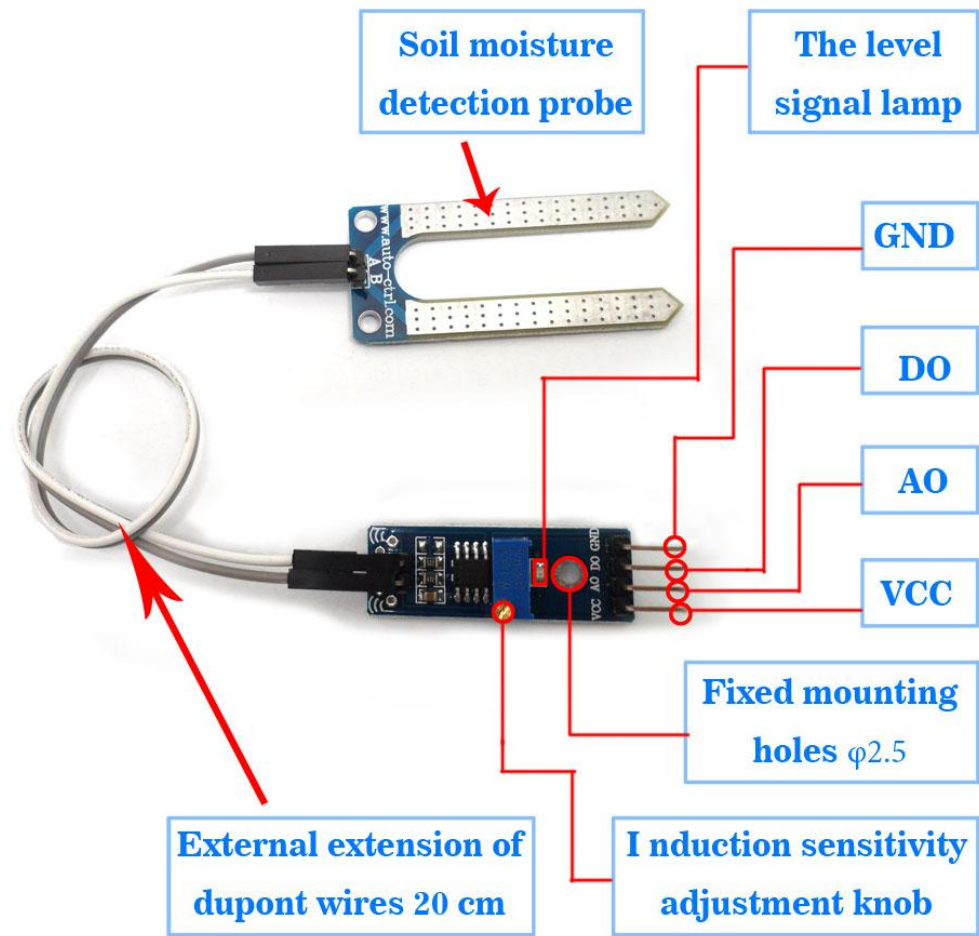
4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

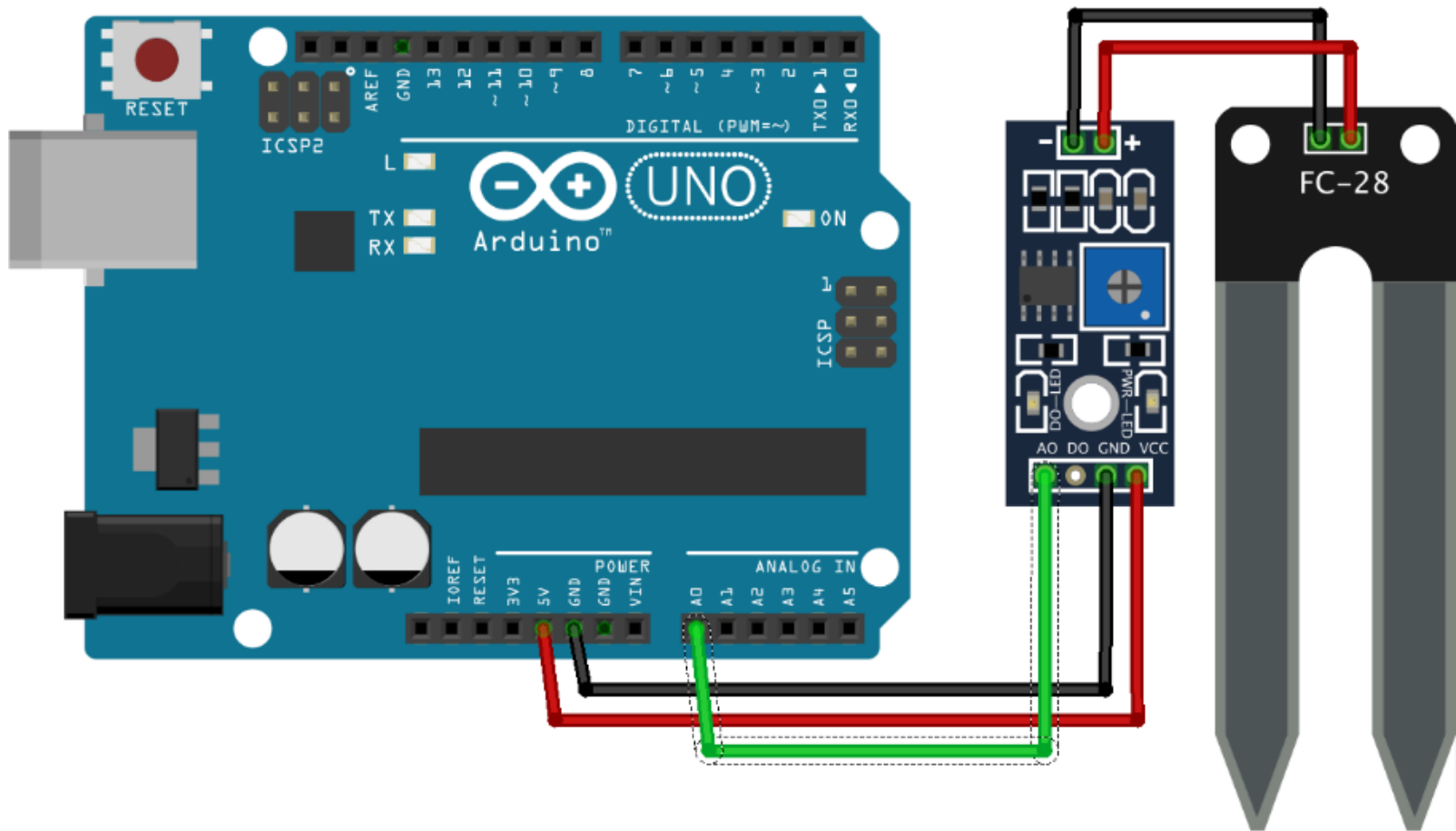
soil module sensor

อุปกรณ์ที่ต้องใช้:

- 1.Arduino Board (เช่น Arduino Uno)
- 2.LDR (Light Dependent Resistor)
- 3.Resistor 10K ohm
- 4.ตัวต้านทาน 220 ohm
- 5.LED (ในกรณีที่ต้องการให้มีการแสดงผล)







fritzing

4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

```
int sensorPin = A0; //กำหนดขาที่เชื่อมต่อกับเซ็นเซอร์ให้เป็นขา A0 ซึ่งในกรณีนี้เป็นเซ็นเซอร์วัดความชื้นในดิน
void setup() {
  Serial.begin(9600); // set up serial port for 9600 baud (speed)
  delay(500); // wait for display to boot up
}
void loop() {
  int sensorValue;
  sensorValue = analogRead(sensorPin);
  sensorValue = map(sensorValue, 0, 1023, 0, 100);
  Serial.print("Soil moisture: ");
  Serial.print(sensorValue);
  Serial.println(" %");
  delay(500); //wait for half a second, so it is easier to read
}
```

เปอร์เซ็นต์ความชื้นไม่ถูกต้อง

4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

```
int sensorPin = A0; //กำหนดขาที่เชื่อมต่อกับเซ็นเซอร์ให้เป็นขา A0 ซึ่งในกรณีนี้เป็นเซ็นเซอร์วัดความชื้นในดิน

void setup() {

  Serial.begin(9600); // set up serial port for 9600 baud (speed)

  delay(500); // wait for display to boot up

}

void loop() {

  int sensorValue;

  sensorValue = analogRead(sensorPin);

  sensorValue = map(sensorValue, 0, 1023, 0, 100);

  Serial.print("Soil moisture: ");

  Serial.print(sensorValue);

  Serial.println(" %");

  delay(500); //wait for half a second, so it is easier to read

}
```

แก้ไขความชื้นที่ถูกต้อง

4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

ในโค้ดที่คุณให้มา การแปลงค่าความชื้นเป็นเปอร์เซ็นต์มีบางประเด็นที่ควรพิจารณาเพิ่มเติม:

การแปลงค่าโดยใช้ฟังก์ชัน map:

```
sensorValue = map(sensorValue, 0, 1023, 0, 100);
```

ฟังก์ชัน map ถูกใช้ในการแปลงค่า sensorValue จากช่วง 0-1023 ไปเป็นช่วง 0-100 ซึ่งในทางทฤษฎีนี้ถูกต้องสำหรับการแปลงค่าความชื้นให้อยู่ในรูปแบบเปอร์เซ็นต์ อย่างไรก็ตาม ค่าที่อ่านได้จากเซ็นเซอร์วัดความชื้นในดินมักจะมีค่าชั๊วชอนกว่านั้น

การคาลิเบรต:

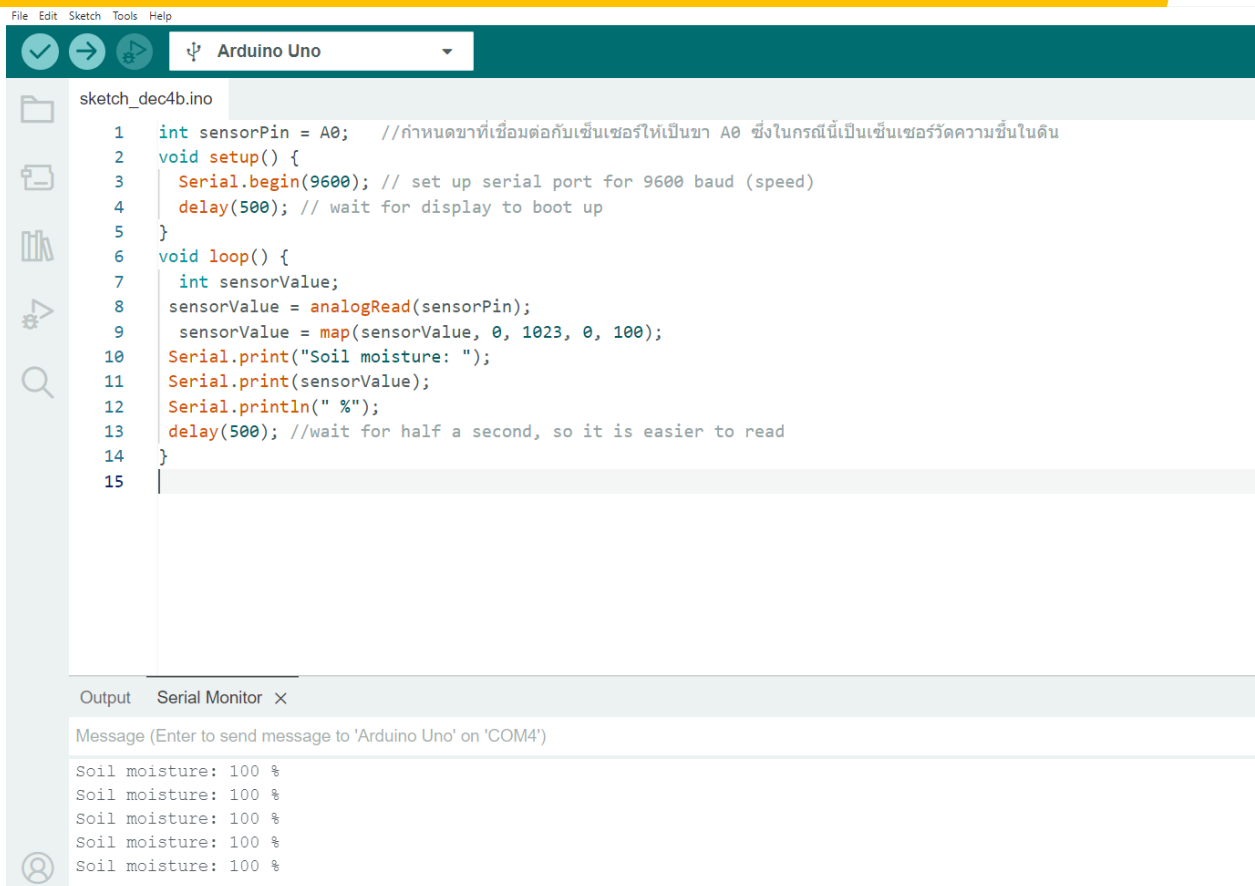
เพื่อให้ได้ค่าที่แม่นยำ คุณอาจต้องคาลิเบรตเซ็นเซอร์ โดยการทดสอบในสภาวะแวดล้อมที่แตกต่างกัน เช่น ดินแห้งสนิท (0%) และดินที่ชื้นเต็มที่ (100%) แล้วจดบันทึกค่าที่อ่านได้จากเซ็นเซอร์ จากนั้นนำค่าเหล่านี้มาแทนที่ในฟังก์ชัน map แทนที่จะใช้ 0 และ 1023 โดยตรง

การคำนวณค่าความชื้นจากเซ็นเซอร์:

ค่าที่อ่านได้จากเซ็นเซอร์วัดความชื้นอาจไม่จำเป็นต้องแปลงเป็นเปอร์เซ็นต์เสมอไป ขึ้นอยู่กับความต้องการของโปรแกรมและความแม่นยำที่ต้องการ

แก้ไขความชื้นที่ถูกต้อง

4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for checking, running, and uploading code, along with a dropdown menu set to 'Arduino Uno'. The main workspace displays a sketch named 'sketch_dec4b.ino' with the following code:

```
1 int sensorPin = A0; //กำหนดขาที่เชื่อมต่อกับเซ็นเซอร์ให้เป็นขา A0 ซึ่งในกรณีนี้เป็นเซ็นเซอร์วัดความชื้นในดิน
2 void setup() {
3   Serial.begin(9600); // set up serial port for 9600 baud (speed)
4   delay(500); // wait for display to boot up
5 }
6 void loop() {
7   int sensorValue;
8   sensorValue = analogRead(sensorPin);
9   sensorValue = map(sensorValue, 0, 1023, 0, 100);
10  Serial.print("Soil moisture: ");
11  Serial.print(sensorValue);
12  Serial.println(" %");
13  delay(500); //wait for half a second, so it is easier to read
14 }
15
```

At the bottom, the 'Serial Monitor' window is open, showing the output of the program:

```
Message (Enter to send message to 'Arduino Uno' on 'COM4')

Soil moisture: 100 %
Soil moisture: 100 %
Soil moisture: 100 %
Soil moisture: 100 %
Soil moisture: 100 %
```

4.5 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

```
int sensorPin = A0;  //กำหนดขาที่เชื่อมต่อกับเซ็นเซอร์ให้เป็นขา A0 ซึ่งในกรณีนี้เป็นเซ็นเซอร์วัดความชื้นในดิน
void setup() {
  Serial.begin(9600); // set up serial port for 9600 baud (speed)
  delay(500); // wait for display to boot up
}
void loop() {
  int sensorValue;
  sensorValue = analogRead(sensorPin);
  sensorValue = map(sensorValue, 0, 1023, 0, 100);
  Serial.print("Soil moisture: ");
  Serial.print(sensorValue);
  Serial.println(" %");
  delay(500); //wait for half a second, so it is easier to read
}
```

```
int ledPin = 3;
```

```
int sensorPin = A0; //กำหนดขาที่เชื่อมต่อกับเซ็นเซอร์ให้เป็นขา A0 ซึ่งในกรณีนี้เป็นเซ็นเซอร์วัดความชื้นในดิน
```

```
void setup() {
```

```
  Serial.begin(9600); // set up serial port for 9600 baud (speed)
```

```
  delay(500); // wait for display to boot up
```

```
  pinMode(ledPin,OUTPUT);
```

```
}
```

```
void loop() {
```

```
  int sensorValue;
```

```
  sensorValue = analogRead(sensorPin);
```

```
  sensorValue = map(sensorValue, 0, 1023, 0, 100);
```

```
  if(sensorValue >= 90) {
```

```
    Serial.println("OK");
```

```
    digitalWrite(ledPin,HIGHT);
```

```
  } else {
```

```
    Serial.println("Not OK");
```

```
    digitalWrite(ledPin,LOW);
```

```
}
```

```
  Serial.print("Soil moisture: ");
```

```
  Serial.print(sensorValue);
```

```
  Serial.println(" %");
```

```
  delay(500); //wait for half a second, so it is easier to read
```

```
}
```

soil module sensor

soil module sensor

```
// กำหนดขาที่เชื่อมต่อ LDR
const int ldrPin = A0;

// กำหนดขาที่เชื่อมต่อ LED
const int ledPin = 13;

void setup() {

    // กำหนดให้ขา LED เป็น Output
    pinMode(ledPin, OUTPUT);

    // เริ่มต้นให้ Serial Monitor เปิดใช้งาน
    Serial.begin(9600);
}
```

soil module sensor

ต่อ

```
void loop() {  
  int lightValue = analogRead(ldrPin); // อ่านค่าแสงที่ LDR  
  Serial.print("Light Value: "); // แสดงค่าความสว่างที่ได้จาก LDR บน Serial Monitor  
  Serial.println(lightValue);  
  if (lightValue >= 500) { // ถ้าความสว่างมากกว่าหรือเท่ากับค่าที่กำหนด (ค่าสว่างเท่ากับ 500 คือเปิด LED)  
    digitalWrite(ledPin, HIGH); // สั่งให้ LED เปิด  
  } else {  
    digitalWrite(ledPin, LOW); // สั่งให้ LED ปิด  
  }  
  delay(1000); // หน่วงเวลา 1 วินาที  
}
```

อ้างอิง

www.iot.codemobiles.com/product/174/เซ็นเซอร์วัดความชื้นในดิน-soil-moisture-sensor

<http://www.iot.codemobiles.com/>

Codemobiles. (n.d.). เซ็นเซอร์วัดความชื้นในดิน (Soil moisture sensor). Retrieved July 21, 2024, from <https://www.iot.codemobiles.com/product/174/เซ็นเซอร์วัดความชื้นในดิน-soil-moisture-sensor>

Codemobiles. (n.d.). Home page. Retrieved July 21, 2024, from <http://www.iot.codemobiles.com/>

อ้างอิง

Codemobiles. (n.d.). เซ็นเซอร์วัดความชื้นในดิน (Soil moisture sensor). Retrieved July 21, 2024, from <https://www.iot.codemobiles.com/product/174/เซ็นเซอร์วัดความชื้นในดิน-soil-moisture-sensor>

Codemobiles. (n.d.). Home page. Retrieved July 21, 2024, from <http://www.iot.codemobiles.com/>

Thank you