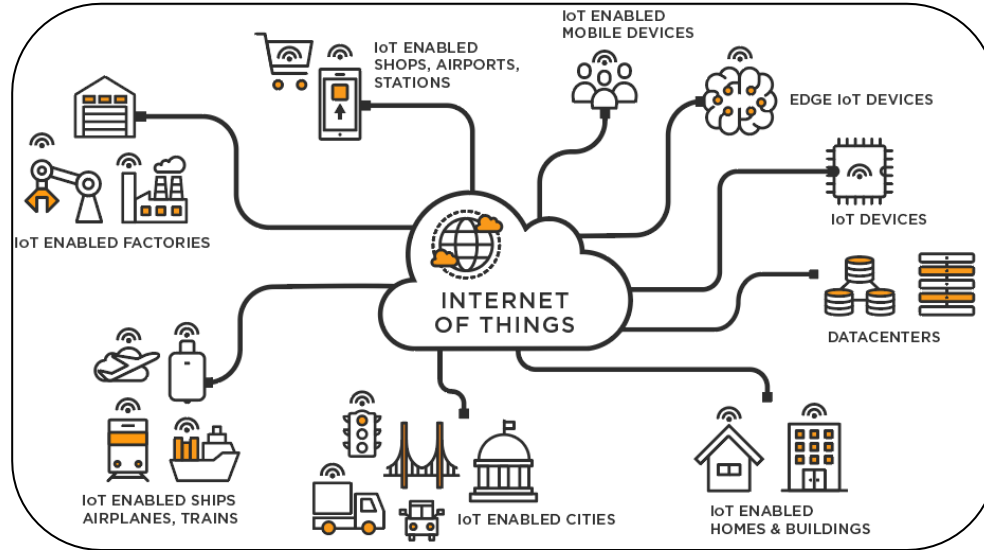


วิชา อินเทอร์เน็ตในทุกสิ่ง

บทที่ 5 การประยุกต์ใช้ Arduino ควบคุมเซ็นเซอร์วัดระยะ, หน้าจอLCD, นาฬิกา



<http://www.siam2dev.com>

Last update :
06.07.2569

IOT: Internet of Things

Asst.Prof. Dr. Nattapong Songneam

<http://www.siam2dev.net>

ตัวอย่างโจทย์ ข้อสอบ

ตัวอย่างโจทย์ ข้อสอบ

ให้ออกแบบการทดลองวัดอุณหภูมิห้อง แล้วแสดงผลบนหน้าจอ LCD

1. อุปกรณ์ที่ใช้มีอะไรบ้าง บอกรุ่น บอกชื่อ บอกหน้าที่ ของอุปกรณ์แต่ละชนิด
2. อธิบายการต่อวงจรมาอย่างละเอียด
3. เขียนโค้ด หรือคำสั่ง ลงในกระดานคำตอบ
4. เขียนคำตอบหรือผลการทดลอง
5. ถ่ายรูป หรือคลิปวิดีโอผลการทดลอง

การนำเสนองานกลุ่มครั้งที่ 1

-
-

- ชื่อเรื่อง
- สมาชิกในกลุ่ม
- ที่มาและความสำคัญของปัญหา
- วัตถุประสงค์ 3 ข้อ
- ขอบเขตของงาน
- แผนการดำเนินงาน
- วิธีการดำเนินงาน / วิธีทำเบื้องต้น
- เครื่องที่ใช้
- งบประมาณ

- **ทำเป็น PPT นำเสนอประมาณ 20 นาที**

- 1) ทักท้วง
- 2) ออกแบบและพัฒนาระบบ ...
- 3) ประเมินความพึงพอใจระบบ....

บทที่ 5 มีหัวข้อดังนี้



1

แนะนำอุปกรณ์

2

โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

3

โปรแกรมแสดงข้อความออกทางหน้าจอ LCD

4

โปรแกรมแสดง อุณหภูมิ และความชื้นออกทางหน้าจอ LCD

5

โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

5.1

แนะนำอุปกรณ์

5.1 แนะนำอุปกรณ์

5.1.1 Arduino Uno R3

อุปกรณ์หลักที่ใช้ในการพัฒนาระบบด้วยแนวคิด IoT เป็นเหมือนจุดที่ใช้ในการจ่ายไฟ หรือส่งต่อข้อมูลไปยังอุปกรณ์ต่าง ๆ โดยวงจรจะสามารถจะต้องมีแหล่งจ่ายไฟ ซึ่งแหล่งจ่ายไฟพื้นฐานได้แก่

- GND ย่อมาจาก GROUND คือ จุดต่ำสุดยี่รวม แรงดันอ้างอิง 0 V โดยอุปกรณ์ภายในวงจร จะมีกระแสไหลมาครบรอบที่ทราวด์ ทุกเรียกว่า ครบวงจร
- VIN คือ ขั้วต่อแหล่งจ่ายภายนอก ที่จ่ายเข้าบอร์ด จากกรณีของ Node MCU 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



5.1 แนะนำอุปกรณ์

5.1.2 หลอด LED

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



5.1 แนะนำอุปกรณ์

5.1.3 ตัวต้านทาน

เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่งที่มีคุณสมบัติในการต้านการไหลผ่านของกระแสไฟฟ้า ทำด้วยลวดต้านทานหรือถ่านคาร์บอนเป็นต้น นั่นคือ ถ้าอุปกรณ์นั้นมีความต้านทานมาก กระแสไฟฟ้าที่ไหลผ่านจะน้อยลง เป็นอุปกรณ์ไฟฟ้าชนิดพาสซีฟสองขั้ว ที่สร้างความต่างศักย์ไฟฟ้าคร่อมขั้วทั้งสอง (V) โดยมีสัดส่วนมากน้อยตามปริมาณกระแสไฟฟ้าที่ไหลผ่าน (I) อัตราส่วนระหว่างความต่างศักย์และปริมาณกระแสไฟฟ้า ก็คือ ค่าความต้านทานทางไฟฟ้า หรือค่าความต้านทานของตัวนำมีหน่วยเป็นโอห์ม (สัญลักษณ์ : Ω)

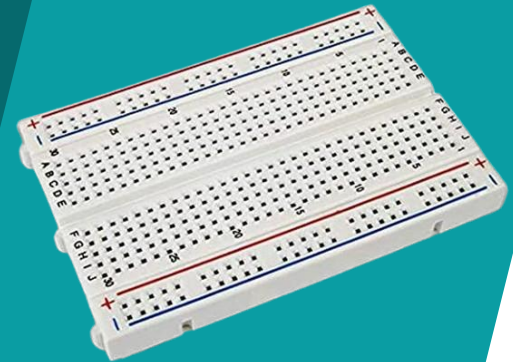


5.1 แนะนำอุปกรณ์

5.1.4 บอร์ดทดลอง

บอร์ดที่ใช้ทดลองวงจรอิเล็กทรอนิกส์ ลักษณะเป็นแผ่นพลาสติกหนาสีขาว บนแผ่นมีรูเรียงกันจำนวนมากภายในรูมีตัวนำไฟฟ้าซึ่งเชื่อมต่อกันในรูปแบบที่มีการกำหนดไว้ เวลาทดลองก็เสียบขาของอุปกรณ์อิเล็กทรอนิกส์ลงไปให้ตัวนำภายในเชื่อมวงจรถึงกัน และอาจใช้สายไฟเสียบลงรูเพื่อเชื่อมวงจรไฟฟ้าได้เช่นกัน ข้อดีของโปรโตบอร์ดคือ ไม่ต้องออกแบบแผงวงจรและไม่ต้องบัดกรี แต่มีข้อเสียคือใช้ทดลองวงจรที่ทำงานที่ความถี่สูง ๆ ไม่ได้เนื่องจากมีปัญหาเรื่องสัญญาณรบกวนในวงจร

บอร์ดทดลอง
(Breadboard)



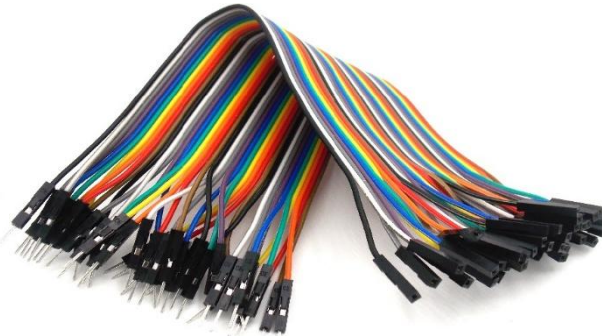
5.1 แนะนำอุปกรณ์

5.1.5 สายต่อ Jumper

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



ผู้ - ผู้



ผู้ - เมีย



เมีย - เมีย

5.1 แนะนำอุปกรณ์

5.1.6 เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04

- นักศึกษาต้องค้นคว้าเพิ่มเติม
- วัดระยะได้ไกลสุด เท่าไร
- หน่วยเป็นอะไร
- และ จะประยุกต์ใช้ได้อย่างไร ?



5.1 แนะนำอุปกรณ์

5.1.6 เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04

- หุ่นยนต์หลบหลีกสิ่งกีดขวาง (Obstacle Avoidance Robot): ใช้เซ็นเซอร์วัดระยะทางในการตรวจจับสิ่งกีดขวางและเปลี่ยนเส้นทางการเคลื่อนที่ของหุ่นยนต์
- ระบบจอดรถอัจฉริยะ (Smart Parking System): ตรวจจับระยะห่างระหว่างรถกับกำแพงหรือรถคันอื่นเพื่อช่วยในการจอดรถ
- ระบบรักษาความปลอดภัย (Security System): ใช้เซ็นเซอร์ในการตรวจจับการเคลื่อนไหวภายในบ้านหรือบริเวณที่ต้องการเฝ้าระวัง
- เครื่องวัดระดับน้ำ (Water Level Indicator): ตรวจสอบระดับน้ำในถังหรือบ่อเก็บน้ำ เพื่อแสดงระดับน้ำในระบบอัตโนมัติ
- ระบบอัตโนมัติในบ้าน (Home Automation): ใช้เซ็นเซอร์ในการเปิดปิดไฟหรืออุปกรณ์ไฟฟ้าอื่นๆ ตามการเคลื่อนไหวของผู้ใช้งาน
- หุ่นยนต์ดูดฝุ่น (Robotic Vacuum Cleaner): ตรวจจับระยะห่างจากผนังหรือเฟอร์นิเจอร์เพื่อวางแผนเส้นทางการทำความสะอาด
- เครื่องวัดความสูง (Height Measurement Device): ใช้ในการวัดความสูงของวัตถุหรือบุคคล
- ระบบขนส่งอัตโนมัติ (Automated Transport System): ใช้เซ็นเซอร์ในการตรวจจับเส้นทางและสิ่งกีดขวางในระบบการขนส่งสินค้า
- เครื่องวัดความเร็วลม (Anemometer): ใช้ในการตรวจวัดระยะห่างของอุปกรณ์ในการวัดความเร็วลม
- อุปกรณ์เสริมการนำทาง (Navigation Aid Devices): ช่วยในการนำทางสำหรับคนตาบอดหรือผู้ที่มีปัญหาทางสายตา

5.1 แนะนำอุปกรณ์

5.1.7 จอ 1604 LCD พร้อม I2C Interface

จอ LCD 1602 พร้อม I2C Interface คือจอแสดงผลแบบ LCD ที่มีขนาด 16 คอลัมน์ และ 2 แถว (16x2) ซึ่งมาพร้อมกับอินเตอร์เฟซ I2C (Inter-Integrated Circuit) ที่ช่วยให้การเชื่อมต่อและควบคุมจอ LCD ทำได้ง่ายขึ้นมากเมื่อเทียบกับการใช้การเชื่อมต่อแบบดั้งเดิมที่ใช้ขา GPIO หลายขาในการควบคุมจอแสดงผล

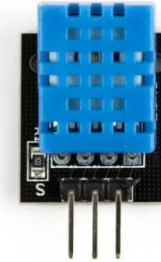


5.1 แนะนำอุปกรณ์

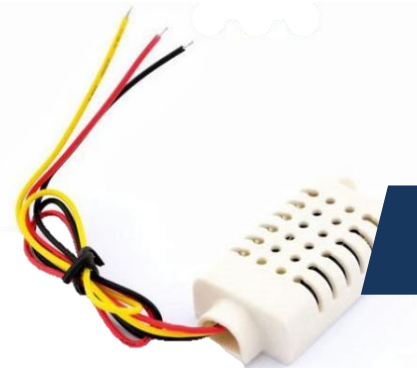
5.1.8 เซ็นเซอร์วัดอุณหภูมิ และความชื้น (DHT22)

เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT22 คือเซ็นเซอร์ที่ใช้ในการตรวจวัดอุณหภูมิและความชื้นในสิ่งแวดล้อม มีความแม่นยำสูงและสามารถนำไปใช้งานในหลายๆ โครงการที่ต้องการตรวจวัดสภาพแวดล้อม

DHT11 กับ DHT22 ต่างกันอย่างไร



DHT11



DHT22

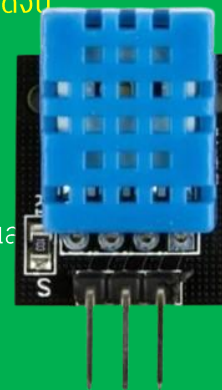
5.1 แนะนำอุปกรณ์

5.1.8 เซ็นเซอร์อุณหภูมิและความชื้น (DHT11)

DHT11

DHT11 เป็นเซ็นเซอร์อุณหภูมิและความชื้นที่มีความทันสมัยและราคาถูก โดยมีความสามารถที่มีนัยสำคัญดังนี้:

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



5.1 แนะนำอุปกรณ์

5.1.8 เซ็นเซอร์อุณหภูมิ และความชื้น (DHT22)

DHT22

DHT22 (หรือ AM2302) เป็นเซ็นเซอร์อุณหภูมิและความชื้นที่มีความแม่นยำและสามารถให้ข้อมูลในเงื่อนไขที่หลากหลาย. นี่คือคุณสมบัติหลักของ DHT22:

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



5.1 แนะนำอุปกรณ์

5.1.8 เปรียบเทียบความแตกต่างของ DHT11 และ DHT22

โปรโตคอลการสื่อสารที่ใช้ใน DHT11 และ DHT22 มีความแตกต่างกัน ซึ่งอาจมีผลต่อประสิทธิภาพและความแม่นยำในการวัดข้อมูล แต่ทั้งสองเซ็นเซอร์นี้มีการใช้งานทั่วไปในการวัดอุณหภูมิและความชื้นในสภาพแวดล้อมทั่วไป

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	สัญญาณดิจิตอลแบบชนิดซีรียัล	สัญญาณดิจิตอลแบบชนิดซีรียัล

5.1 แนะนำอุปกรณ์

5.1.8 โมดูลนาฬิกา DS3231 Module

- โมดูลนาฬิกา DS3231 เป็นโมดูล Real-Time Clock (RTC) ที่ใช้ในการเก็บรักษาเวลาและวันที่อย่างแม่นยำ มักใช้ในโครงการที่ต้องการการจัดการเวลาที่ต่อเนื่อง เช่น นาฬิกาเครื่องบันทึกข้อมูล และระบบควบคุมอัตโนมัติ
- เอาไปประยุกต์ใช้ทำอะไรได้บ้าง
- งานอะไรบ้างที่ต้องใช้ module นาฬิกา

DS3231 Module

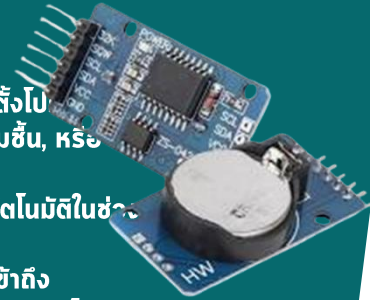


5.1 แนะนำอุปกรณ์

5.1.8 โมดูลนาฬิกา DS3231 Module

DS3231 Module

- นาฬิกาดังโต๊ะ (Desktop Clock) ใช้โมดูล DS3231 ในการแสดงเวลาและวันที่บนจอแสดงผลแบบ LCD หรือ LED
- ระบบปลุกอัตโนมัติ (Automatic Alarm System) สร้างระบบปลุกที่สามารถตั้งเวลาปลุกได้ตามต้องการและสามารถตั้งไป
- เครื่องบันทึกข้อมูล (Data Logger) ใช้โมดูล DS3231 ในการเก็บข้อมูลเวลาที่แน่นอนสำหรับการบันทึกอุณหภูมิ, ความชื้น, หรือเซ็นเซอร์อื่นๆ
- *ระบบควบคุมไฟ (Lighting Control System) ใช้โมดูล DS3231 ในการเปิด-ปิดไฟตามเวลาที่กำหนด เช่น เปิดไฟอัตโนมัติในช่วงเช้าและปิดในช่วงเย็น
- ระบบควบคุมการเข้าถึง (Access Control System) ใช้ในการบันทึกเวลาการเข้าออกของบุคคลในระบบควบคุมการเข้าถึง
- ระบบน้ำหยดอัตโนมัติ (Automatic Drip Irrigation System) ใช้ในการควบคุมการเปิด-ปิดระบบน้ำหยดตามเวลาที่กำหนดไว้
- ระบบเตือนความจำ (Reminder System) สร้างระบบเตือนความจำสำหรับงานต่างๆ เช่น การกินยา การตรวจสอบอุปกรณ์ หรือการทำงานประจำวัน
- ระบบการบันทึกเวลาเรียน (Attendance System) ใช้ในการบันทึกเวลาเข้าเรียนและเวลาเลิกเรียนของนักเรียน
- เครื่องควบคุมอุณหภูมิ (Temperature Control Device) ใช้ในการควบคุมอุณหภูมิในห้องหรืออุปกรณ์ต่างๆ โดยตั้งโปรแกรมให้ทำงานตามเวลาที่กำหนด
- นาฬิกาปลุกสำหรับผู้สูงอายุ (Senior Citizen Alarm Clock) ใช้ในการเตือนผู้สูงอายุในเรื่องต่างๆ เช่น การกินยา หรือการออกกำลังกาย



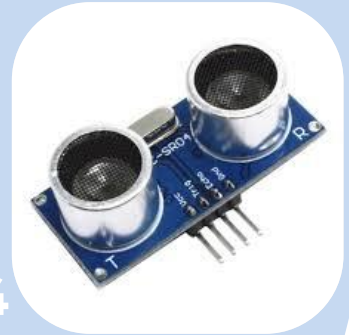
5.2

โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.1 อุปกรณ์ที่ต้องใช้

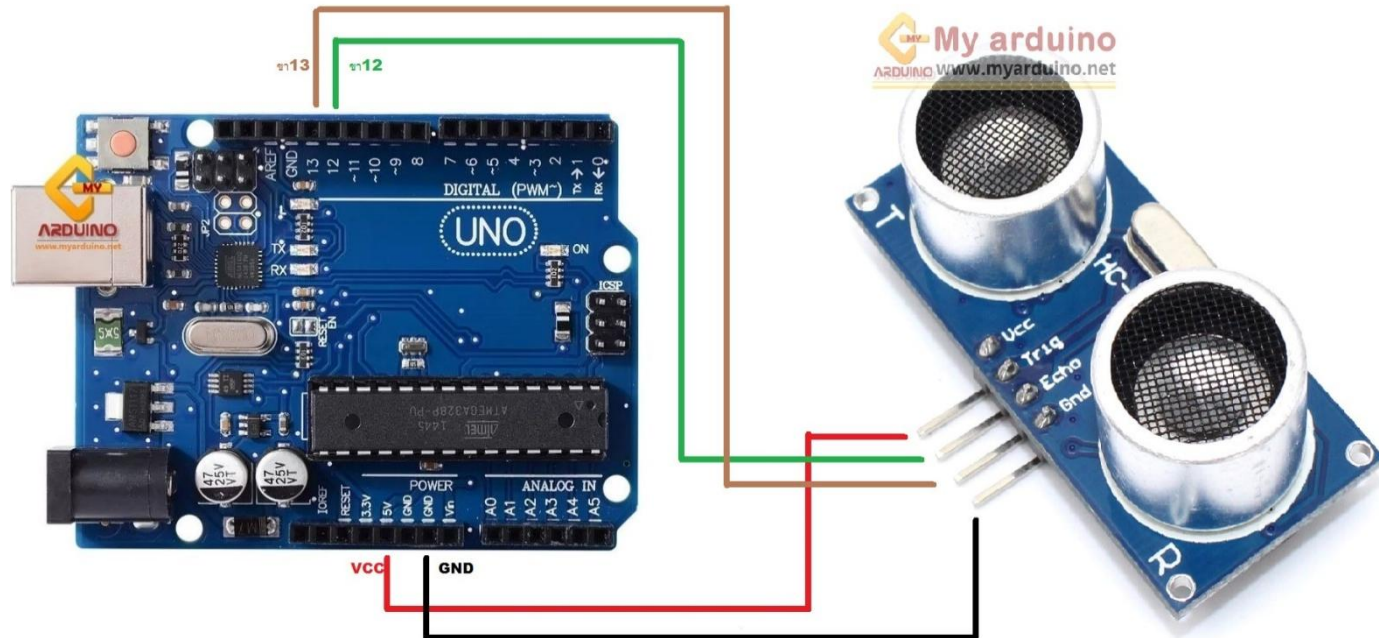
- **Arduino UNO R3 พร้อม สายUSB**
- **บอร์ดทดลอง Breadboard**
- **สายไฟจัมเปอร์ คู่-คู่ 4 เส้น**
- **เซ็นเซอร์วัดระยะทาง Ultrasonic Module HC-SR04**



5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

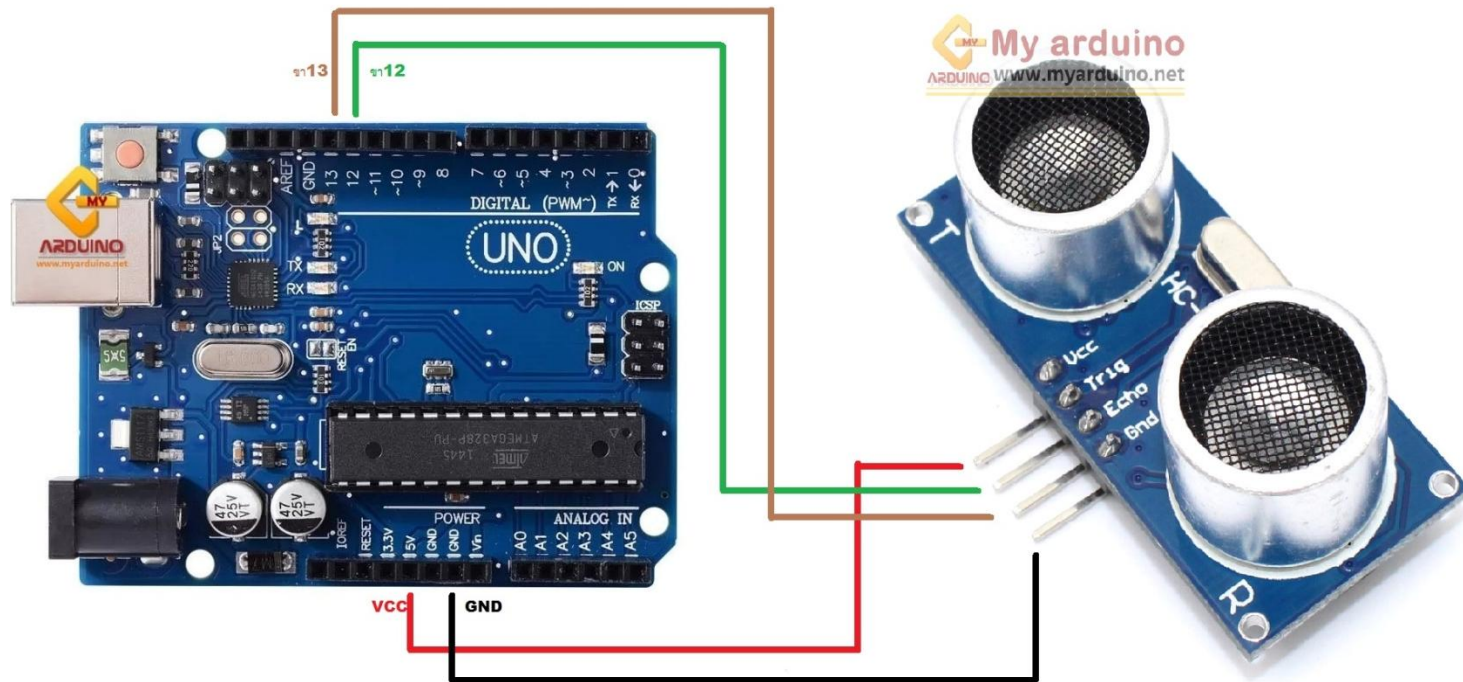
5.2.2 วิธีการต่ออุปกรณ์

- ขา12 -> Echo
- ขา13 -> Trig
- 5V -> Vcc
- GND -> GND



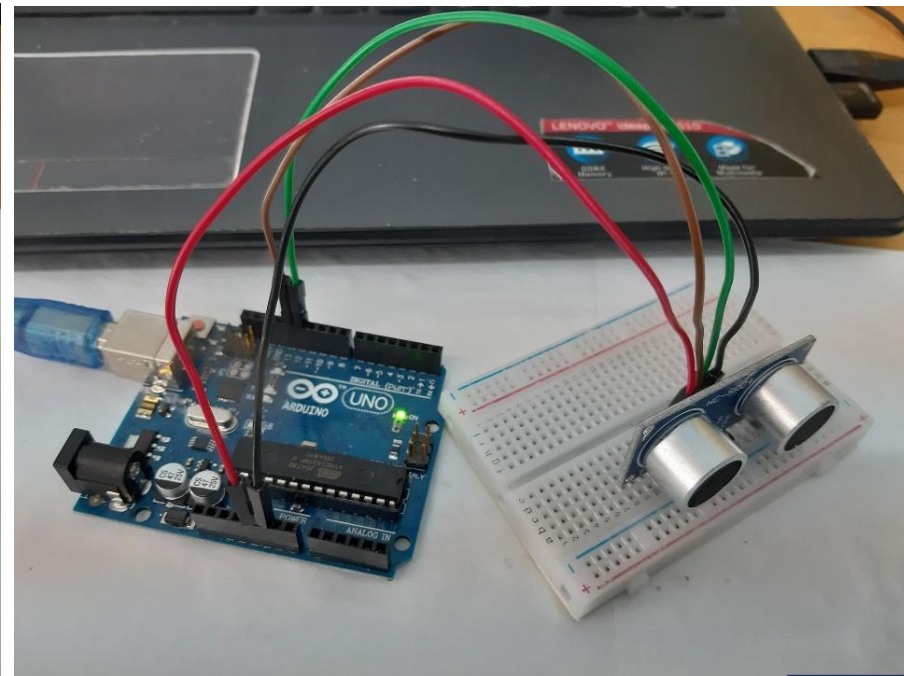
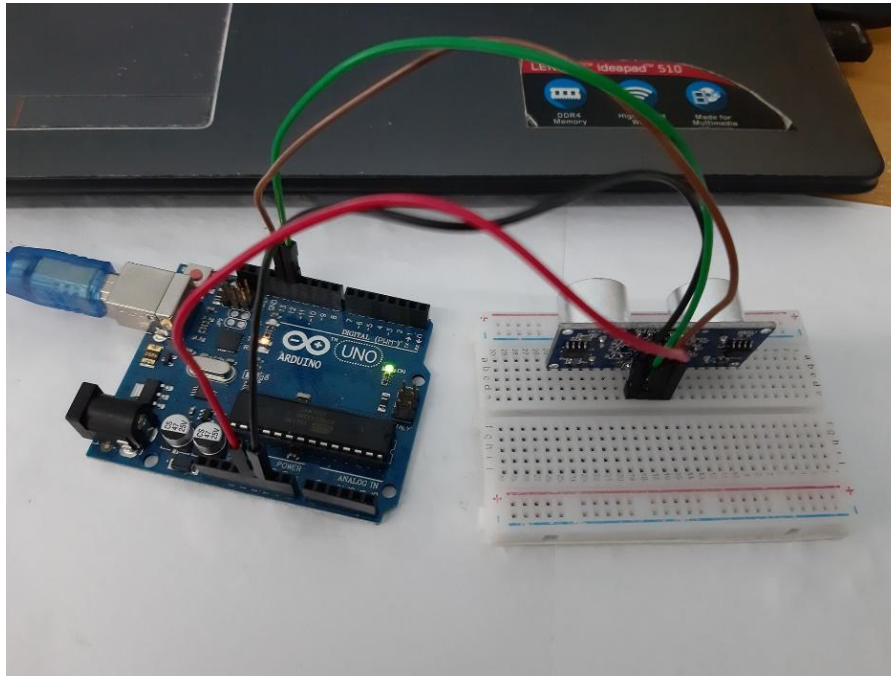
5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.2 วิธีการต่ออุปกรณ์



5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.3 ภาพการต่ออุปกรณ์



5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.4 โปรแกรม

1

```
const int pingPin = 13;
int inPin = 12;
void setup() {
  Serial.begin(9600);
}
void loop()
{
  long duration, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
```

Code โปรแกรม

2

```
digitalWrite(pingPin, LOW);
pinMode(inPin, INPUT);
duration = pulseIn(inPin, HIGH);
cm = microsecondsToCentimeters(duration);
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
}
long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
```

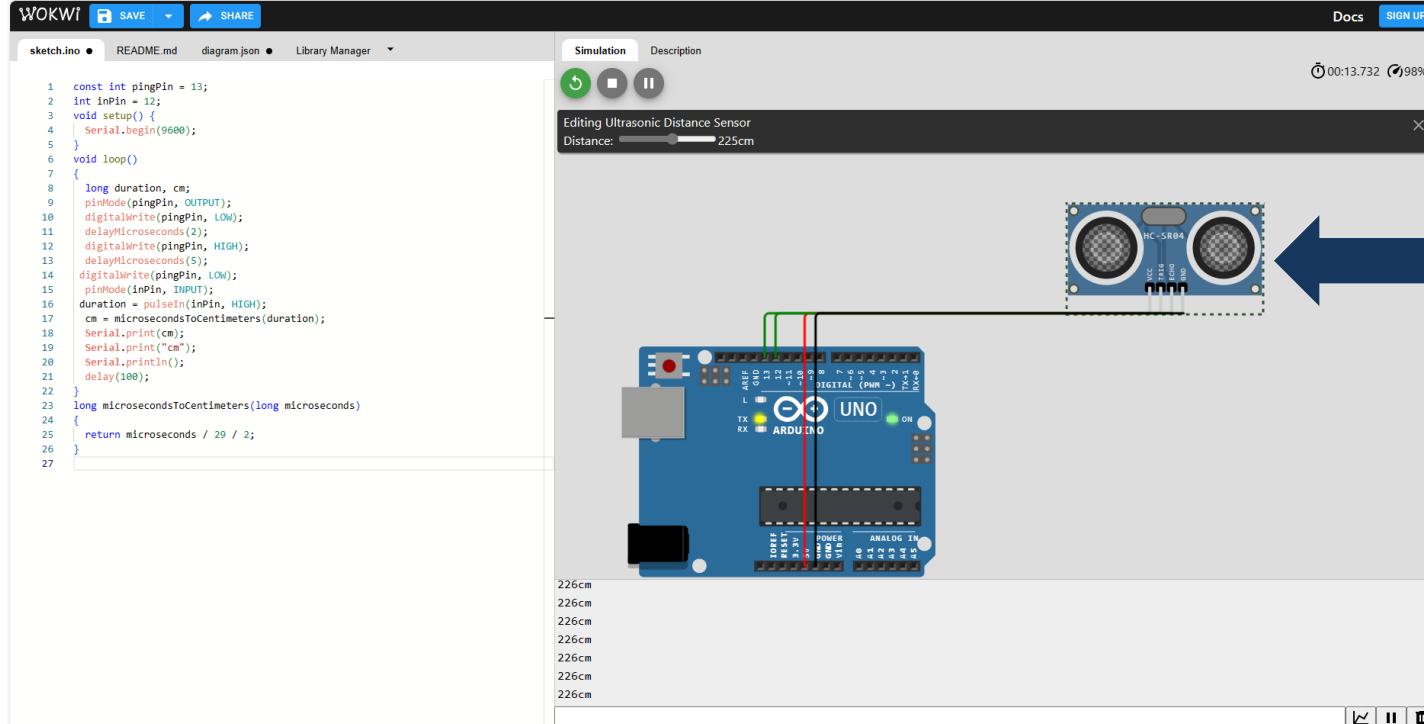
5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.5 หน้าจอตัวอย่าง Code

```
Tools Help
nic01
const int pingPin = 13;
int inPin = 12;
void setup() {
  Serial.begin(9600);
}
void loop()
{
  long duration, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(pingPin, LOW);
  pinMode(inPin, INPUT);
  duration = pulseIn(inPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  delay(100);
}
long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
```

5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.6 Wokwi : New Arduino Uno Project - Wokwi Simulator



WOKWI SAVE SHARE Docs SIGN UP

sketch.ino • README.md diagram.json • Library Manager

```

1  const int pingPin = 13;
2  int inPin = 12;
3  void setup() {
4    Serial.begin(9600);
5  }
6  void loop()
7  {
8    long duration, cm;
9    pinMode(pingPin, OUTPUT);
10   digitalWrite(pingPin, LOW);
11   delayMicroseconds(2);
12   digitalWrite(pingPin, HIGH);
13   delayMicroseconds(5);
14   digitalWrite(pingPin, LOW);
15   pinMode(inPin, INPUT);
16   duration = pulseIn(inPin, HIGH);
17   cm = microsecondsToCentimeters(duration);
18   Serial.print(cm);
19   Serial.print("cm");
20   Serial.println();
21   delay(100);
22 }
23 long microsecondsToCentimeters(long microseconds)
24 {
25   return microseconds / 29 / 2;
26 }
27

```

Simulation Description 00:13.732 98%

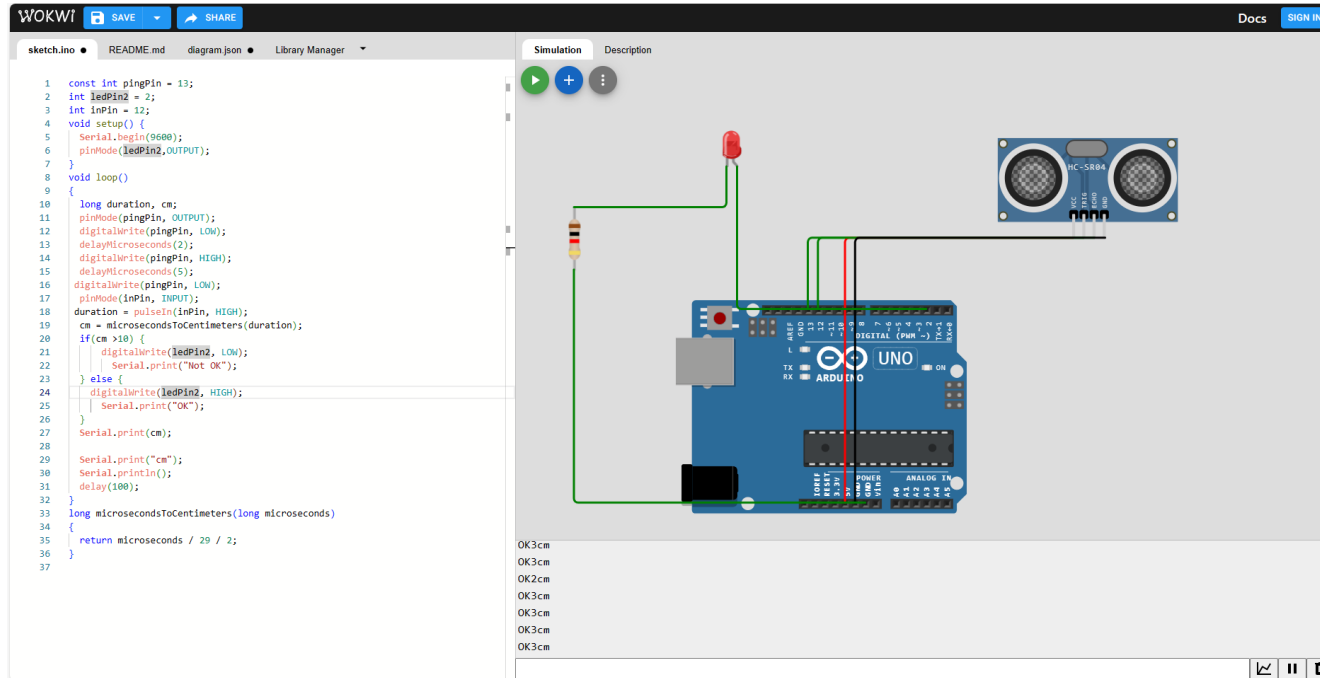
Editing Ultrasonic Distance Sensor
Distance: 225cm

Click ที่นี่เพื่อปรับ ระยะ

226cm
226cm
226cm
226cm
226cm
226cm
226cm

5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.7 ตัวอย่างการประยุกต์ : ถ้าระยะที่วัดได้น้อยกว่า 10 cm ให้หลอดไฟ LED สดสว่าง แต่ถ้ามากกว่าให้หลอด LED ดับ



5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.7 ตัวอย่างการประยุกต์ : ถ้าระยะที่วัดได้น้อยกว่า 10 cm ให้หลอดไฟ LED สดสว่าง แต่ถ้ามากกว่าให้หลอด LED ดับ

The image shows a WOKWI simulation of an Arduino Uno microcontroller board connected to an HC-SR04 ultrasonic sensor module and a red LED. The sensor's VCC pin is connected to the 5V pin on the Arduino, and its GND pin is connected to a GND pin. The sensor's Trig pin is connected to digital pin 12, and its Echo pin is connected to digital pin 13. The red LED's anode is connected to digital pin 2, and its cathode is connected to GND. The Arduino's 5V and GND pins are also connected to the breadboard power rails.

The code in the sketch.ino file is as follows:

```

1 const int pingPin = 13;
2 int ledPin = 2;
3 int inPin = 12;
4 void setup() {
5   Serial.begin(9600);
6   pinMode(ledPin, OUTPUT);
7 }
8 void loop() {
9   long duration, cm;
10  digitalWrite(pingPin, LOW);
11  delayMicroseconds(2);
12  digitalWrite(pingPin, HIGH);
13  delayMicroseconds(5);
14  digitalWrite(pingPin, LOW);
15  pinMode(inPin, INPUT);
16  duration = pulseIn(inPin, HIGH);
17  cm = microsecondsToCentimeters(duration);
18  if(cm < 10) {
19    digitalWrite(ledPin, LOW);
20    Serial.print("Not OK");
21  } else {
22    digitalWrite(ledPin, HIGH);
23    Serial.print("OK");
24  }
25  Serial.print(cm);
26  Serial.print("cm");
27  Serial.println();
28  delay(100);
29 }
30 long microsecondsToCentimeters(long microseconds)
31 {
32   return microseconds / 29 / 2;
33 }

```

The simulation interface shows the code on the left, a central workspace with the hardware components, and a right-hand panel with simulation controls and a description tab. The timer at the top right indicates 00:03.532 and 95% completion.

5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

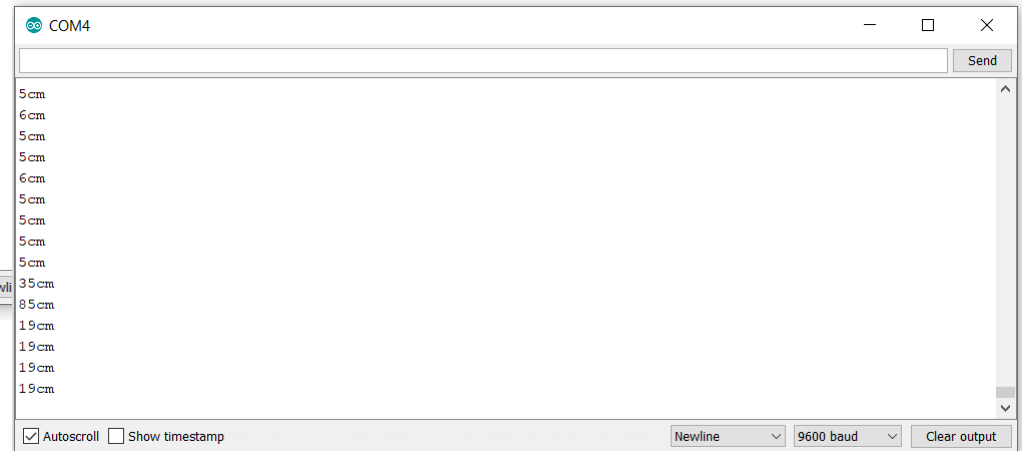
5.2.7 ตัวอย่างการประยุกต์ : ถ้าระยะที่วัดได้น้อยกว่า 10 cm ให้หลอดไฟ LED สดสว่าง แต่ถ้ามากกว่าให้หลอด LED ดับ

ถ้าระยะมากกว่า 10 cm ให้ไฟดับ ถ้ามากกว่าให้ไฟติด

```
const int pingPin = 13;
int inPin = 12;
int led1 = 2; // กำหนดขาใช้งาน
void setup() {
  pinMode(led1, OUTPUT); // กำหนดขาทำหน้าที่ให้
  ขา 2 เป็น OUTPUT
  digitalWrite(led1,LOW);
  Serial.begin(9600);
}
void loop()
{
  long duration, cm;
  pinMode(pingPin, OUTPUT);
  digitalWrite(pingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(pingPin, HIGH);
  delayMicroseconds(5);
```

```
  digitalWrite(pingPin, LOW);
  pinMode(inPin, INPUT);
  duration = pulseIn(inPin, HIGH);
  cm = microsecondsToCentimeters(duration);
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  if(cm > 10) {
    digitalWrite(led1,LOW); // ไฟ LED 1 ติด 50 ms
  } else {
    digitalWrite(led1,HIGH); // ไฟ LED 1 ติด 50 ms
    Serial.print("no");
  }
  delay(50);
  delay(1000);
}
long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
```

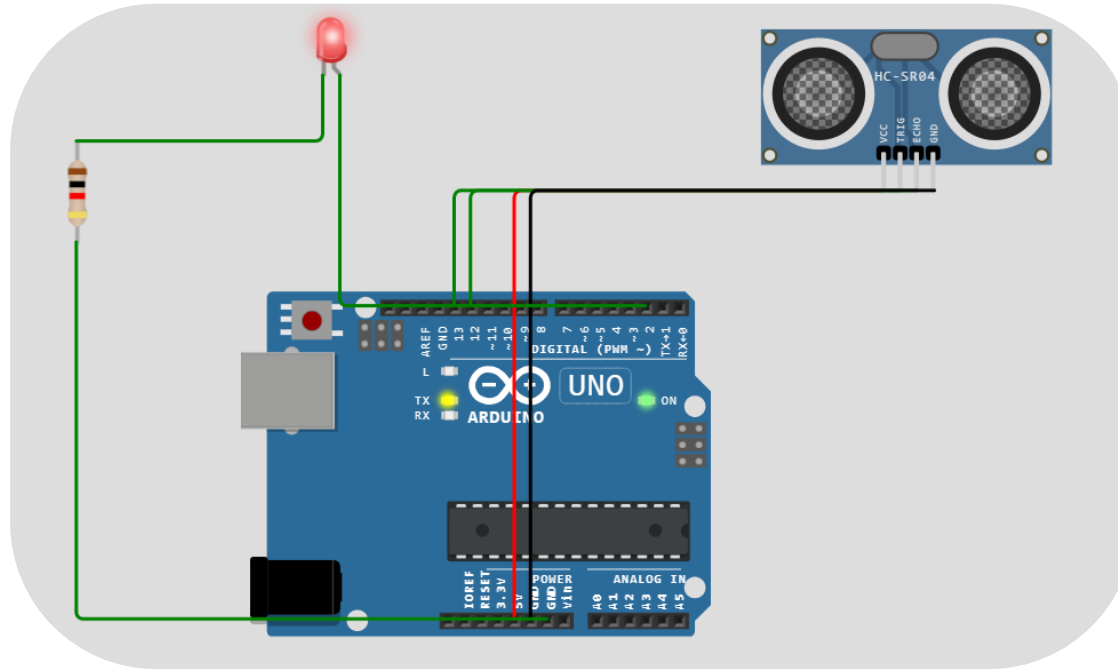
5.2.8 ภาพแสดงระยะห่างจากวัตถุ



5.2 โปรแกรมวัดระยะทางด้วย Ultrasonic Module HC-SR04

5.2.9 ตัวอย่างการประยุกต์ :

ยังเข้าใกล้ ให้หลอดไฟ LED ค่อย ๆ สว่างขึ้น แต่ถ้ายังออกห่าง ค่อย ๆ ให้หรี่ลง จนดับ



5.3

โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface



5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.1 อุปกรณ์ที่ต้องใช้

- Arduino UNO R3 พร้อม สายUSB
- 1602 LCD (Blue Screen) 16x2 โมดูลจอ LCD พร้อม I2C Interface
- สายไฟจัมเปอร์ ผุ้-เมีย 4 เส้น

5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.2 ติดตั้ง Libraries หน้าจอ LCD

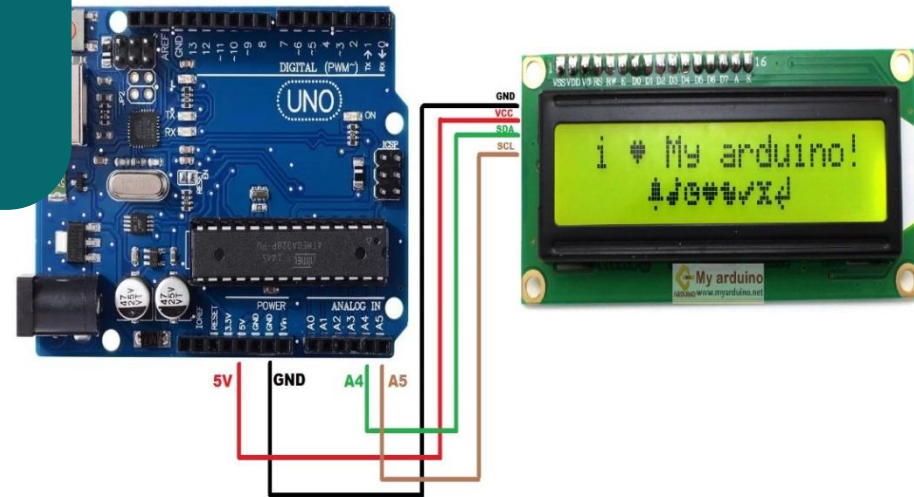
ดาวน์โหลด Libraries และทำการแตกไฟล์ลงใน Folder: libraries

- http://www.mediafire.com/file/iavimlofgo1aq6l/LiquidCrystal_i2c.rar

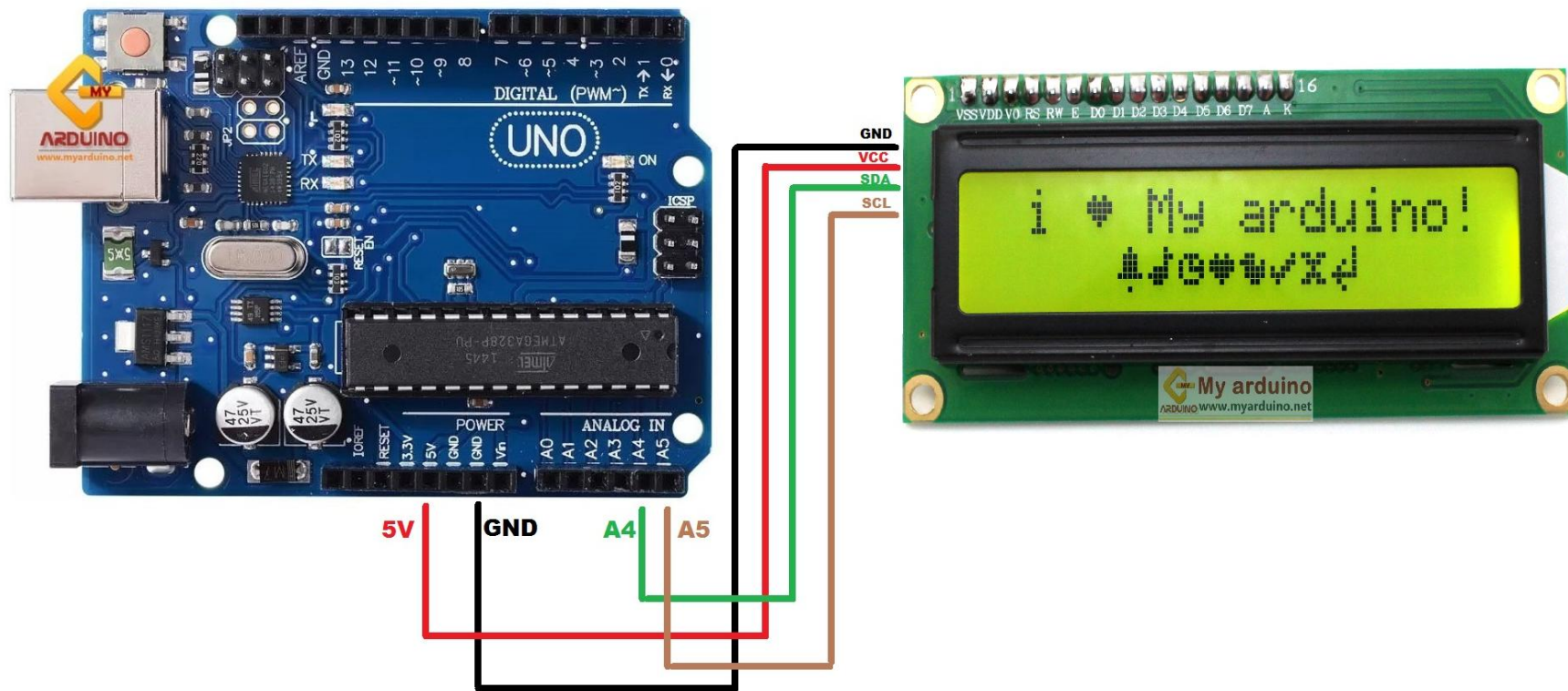
5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.3 วิธีการต่ออุปกรณ์

- gnd -> GND
- Vcc -> 5V
- SDA -> ขาA4 (สำหรับ Arduino uno r3)
- SCL -> ขาA5 (สำหรับ Arduino uno r3)



5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface



5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

จอแสดงผล TFT LCD ซึ่งเป็นจอสีที่สามารถแสดงผลข้อความ ภาพ และกราฟิกต่างๆ ได้ มักใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ เช่น Arduino, Raspberry Pi หรือ ESP8266

คุณสมบัติหลักของจอ TFT LCD:

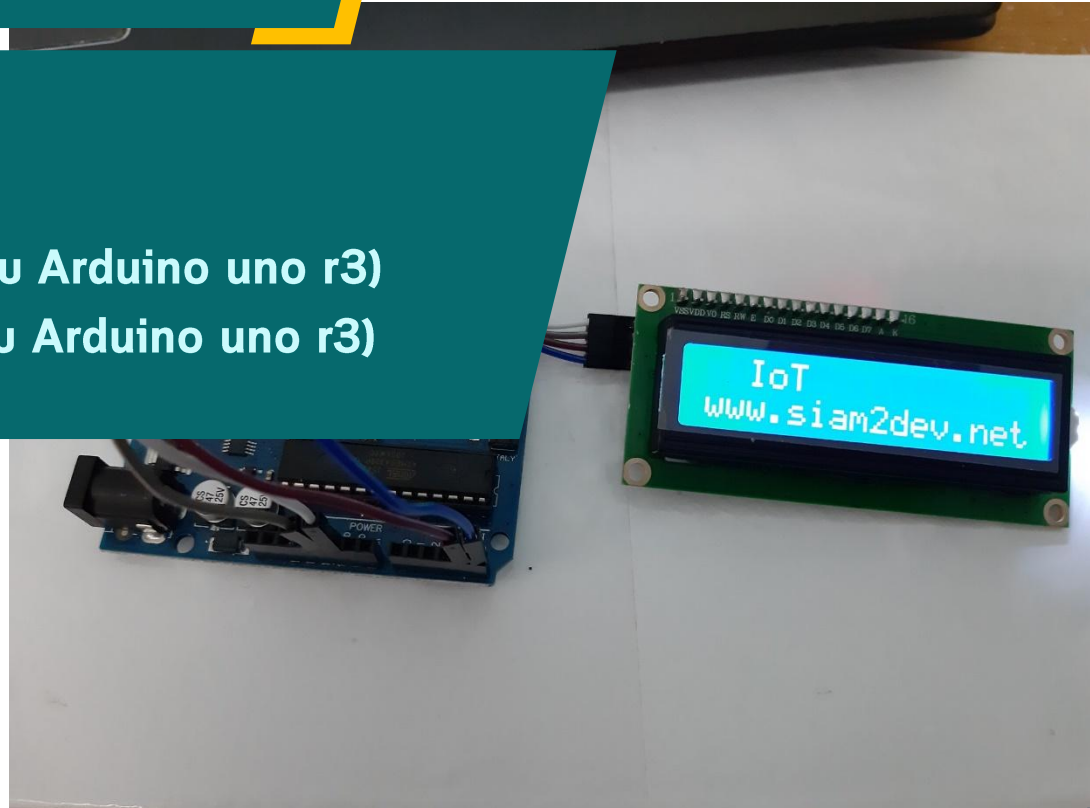
- 1.ความละเอียดสูง: ให้ความละเอียดของภาพที่คมชัดและมีสีสันสดใส
- 2.อินเตอร์เฟซ SPI หรือ Parallel: ใช้การเชื่อมต่อแบบ SPI (Serial Peripheral Interface) หรือ Parallel ที่ช่วยให้การสื่อสารกับไมโครคอนโทรลเลอร์ทำได้ง่ายและรวดเร็ว
- 3.การแสดงผลแบบสี: สามารถแสดงผลสีได้มากมาย ทำให้เหมาะกับการแสดงผลกราฟิกและภาพถ่าย
- 4.ทัชสกรีน (ถ้ามี): บางรุ่นมาพร้อมกับฟังก์ชันทัชสกรีนที่ช่วยให้สามารถอินเตอร์แอคทีฟกับผู้ใช้งานได้



5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.3 วิธีการต่ออุปกรณ์

- gnd -> GND
- Vcc -> 5V
- SDA -> ขาA4 (สำหรับ Arduino uno r3)
- SCL -> ขาA5 (สำหรับ Arduino uno r3)



5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface



ภาพการต่อวงจร

5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.4 Wokwi : New Arduino Uno Project - Wokwi Simulator

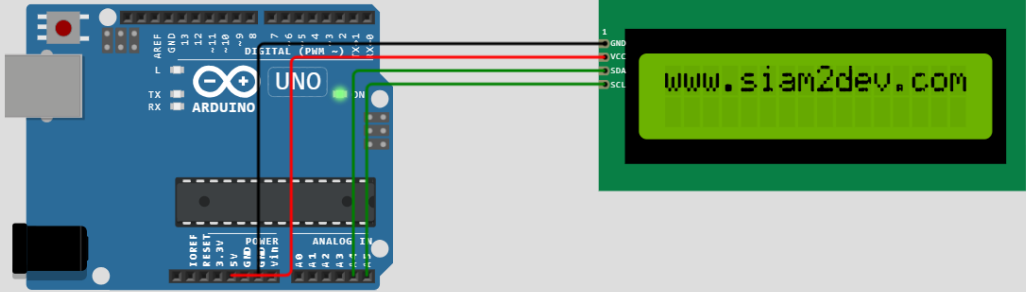
WOKWI SAVE SHARE Docs SIGN UP

sketch.ino • README.md • diagram.json • libraries.txt • Library Manager

Simulation Description

00:04.615 99%

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 // ระบุขนาดของหน้าจอ LCD (cols x rows)
5 LiquidCrystal_I2C lcd(0x27, 16, 2); // ตัวอย่าง 16x2
6
7 void setup() {
8   lcd.begin(16, 2); // ระบุขนาดและจำนวนหลักแถวใน LiquidCrystal_I2C::begin()
9   lcd.backlight(); // สำหรับเปิดไฟหลัง LCD
10  lcd.print("www.siam2dev.com");
11 }
12
13 void loop() {
14   // โค้ดใน loop ถูกทำซ้ำตลอดเวลา
15 }
16
```



The image shows a Wokwi simulator interface. On the left, there is a code editor with a sketch named 'sketch.ino'. The code includes the necessary libraries for an Arduino Uno to communicate with an I2C LCD (LiquidCrystal_I2C). The setup function initializes the LCD at address 0x27 with 16 columns and 2 rows, and the loop function prints the website 'www.siam2dev.com' to the LCD. On the right, the simulation shows an Arduino Uno board connected to an LCD display. The LCD screen displays the text 'www.siam2dev.com'. The connection is made via I2C pins (SDA and SCL) and power pins (VCC and GND). The simulator also shows a 'Simulation' tab with a play button and a 'Description' tab. The top bar includes 'WOKWI', 'SAVE', 'SHARE', 'Docs', and 'SIGN UP' buttons. The bottom status bar shows the simulation time as 00:04.615 and a 99% completion rate.

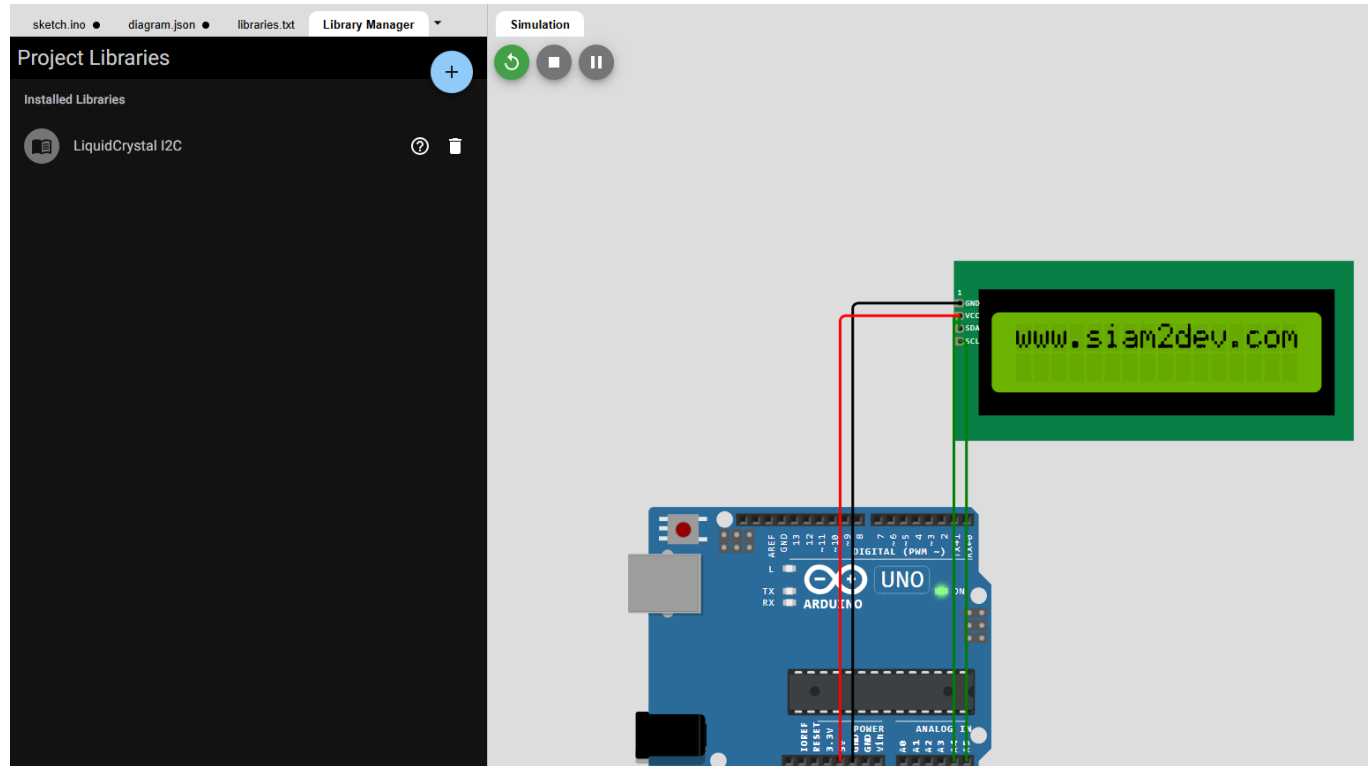
5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.4 Wokwi : New Arduino Uno Project - Wokwi Simulator

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
// ระบุขนาดของหน้าจอ LCD (cols x rows)
LiquidCrystal_I2C lcd(0x27, 16, 2); // ตัวอย่าง 16x2
void setup() {
  lcd.begin(16, 2); // ระบุขนาดและจำนวนหลักแถวใน LiquidCrystal_I2C::begin()
  lcd.backlight(); // สำหรับเปิดไฟหลัง LCD
  lcd.print("www.siam2dev.com");
}
void loop() {
  // ใ้ติดใน loop ถูกทำซ้ำตลอดเวลา
}
```

5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

Install libraries I2C



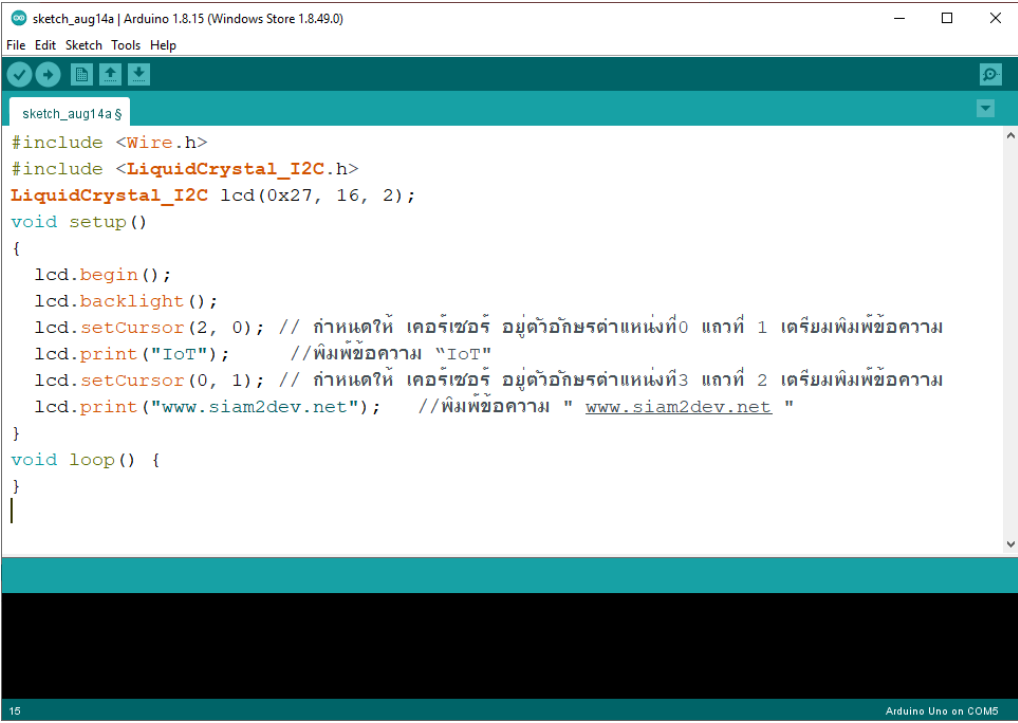
5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.5 Code โปรแกรม

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(2, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 1 เตรียมพิมพ์ข้อความ
  lcd.print("IoT"); //พิมพ์ข้อความ "IoT"
  lcd.setCursor(0, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่3 แถวที่ 2 เตรียมพิมพ์ข้อความ
  lcd.print("www.siam2dev.com"); //พิมพ์ข้อความ " www.siam2dev.com "
}
void loop() {
}
```


5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.5 Code โปรแกรม

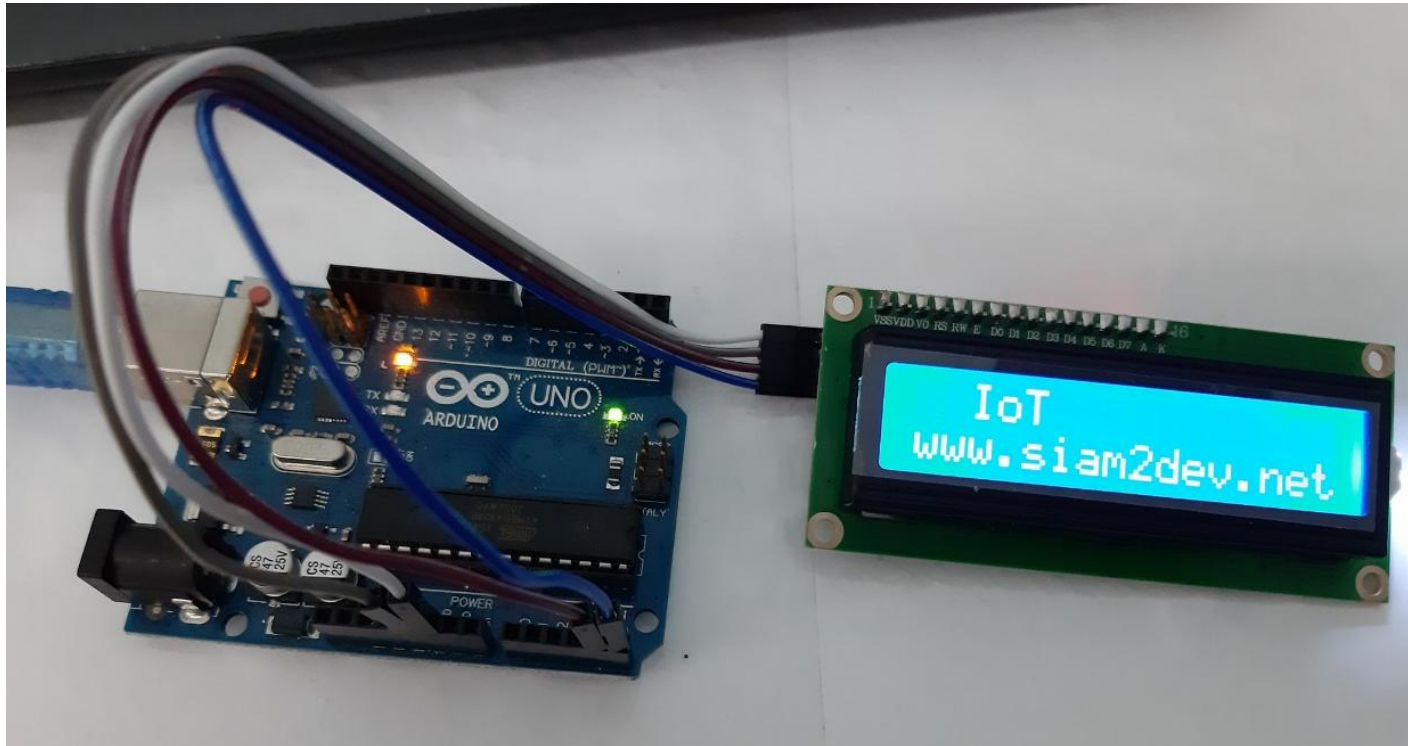


```
sketch_aug14a | Arduino 1.8.15 (Windows Store 1.8.49.0)
File Edit Sketch Tools Help

sketch_aug14a $
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup()
{
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(2, 0); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่0 แถวที่ 1 เตรียมพิมพ์ข้อความ
  lcd.print("IoT");    //พิมพ์ข้อความ "IoT"
  lcd.setCursor(0, 1); // กำหนดให้ เคอร์เซอร์ อยู่ตัวอักษรตำแหน่งที่3 แถวที่ 2 เตรียมพิมพ์ข้อความ
  lcd.print("www.siam2dev.net"); //พิมพ์ข้อความ " www.siam2dev.net "
}
void loop() {
}
```

5.3 โปรแกรมแสดงข้อความบนหน้าจอ LCD 1602 พร้อม I2C Interface

5.3.6 ผลการทำงาน



5.4

โปรแกรมแสดงอุณหภูมิ และความชื้นออกทาง
หน้าจอ LCD 1602

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.1 อุปกรณ์ที่ต้องใช้

Arduino UNO R3 พร้อม สายUSB
เซ็นเซอร์วัด อุณหภูมิ และความชื้น DHT22
1602 LCD (Blue Screen) 16x2 โมดูลจอ LCD พร้อม I2C Interface
บอร์ดทดลอง Breadboard
Resistor ตัวต้านทาน 1 อัน
สายไฟจัมเปอร์ คู่-คู่
สายไฟจัมเปอร์ คู่-เมีย

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.2 ติดตั้ง Libraries หน้าจอ LCD

ดาวน์โหลด Libraries และทำการแตกไฟล์ลงใน Folder: libraries

http://www.mediafire.com/file/iavimlofgo1aq6l/LiquidCrystal_i2c.rar

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

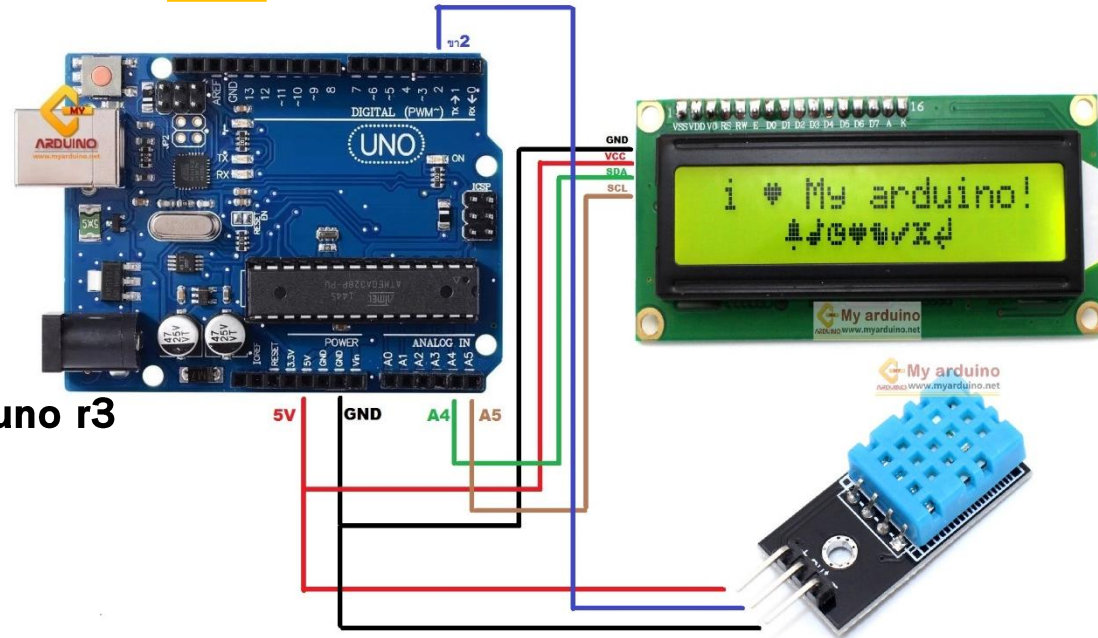
5.4.3 วิธีการต่ออุปกรณ์

จอแสดงผล LCD 1602 -> Arduino uno r3

- gnd -> GND
- Vcc -> 5V
- SDA -> ขา A4
- SCL -> ขา A5

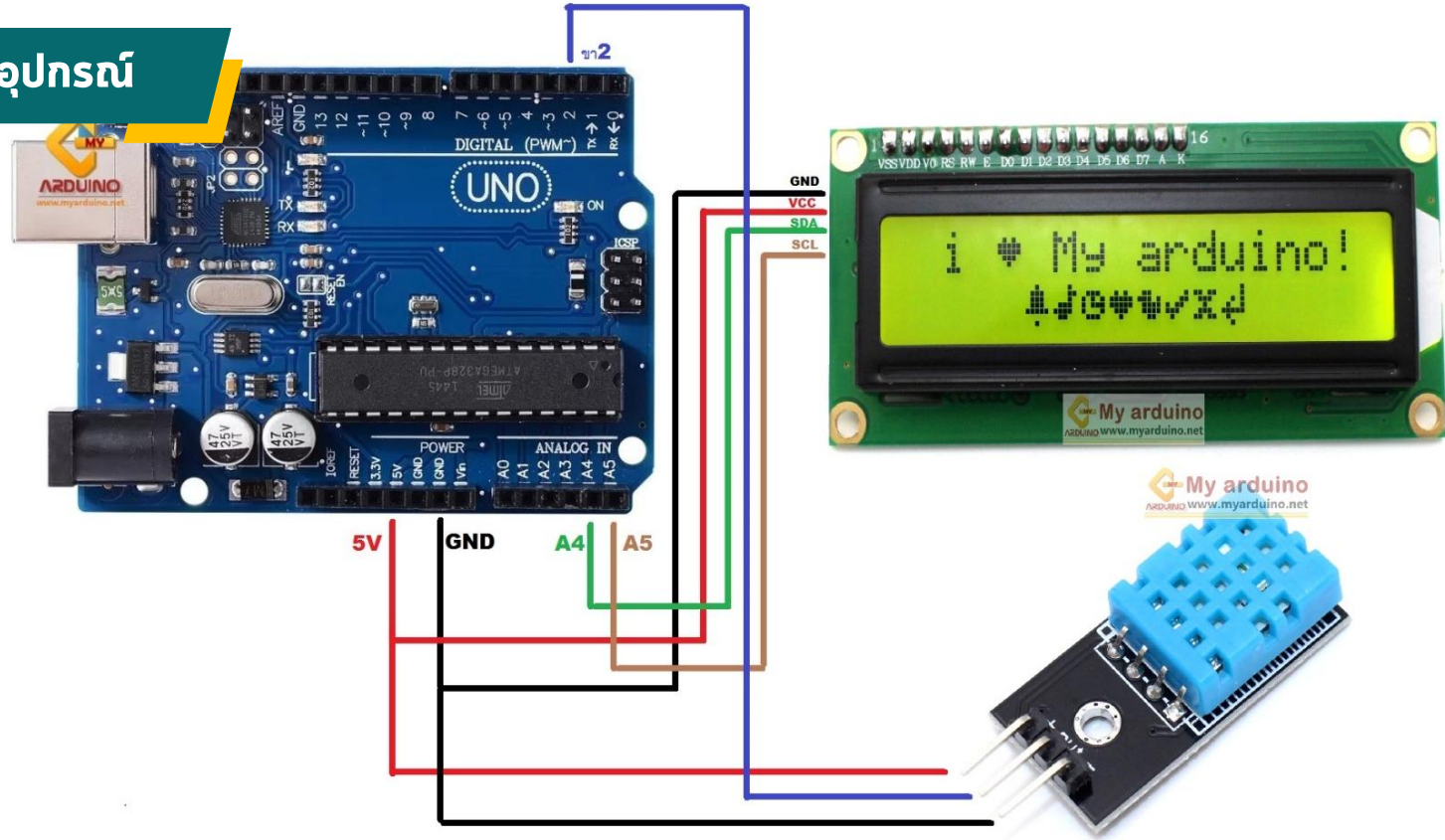
Sensor วัดอุณหภูมิและความชื้น -> Arduino uno r3

- gnd -> GND
- Vcc -> 5V
- Out -> ขา2



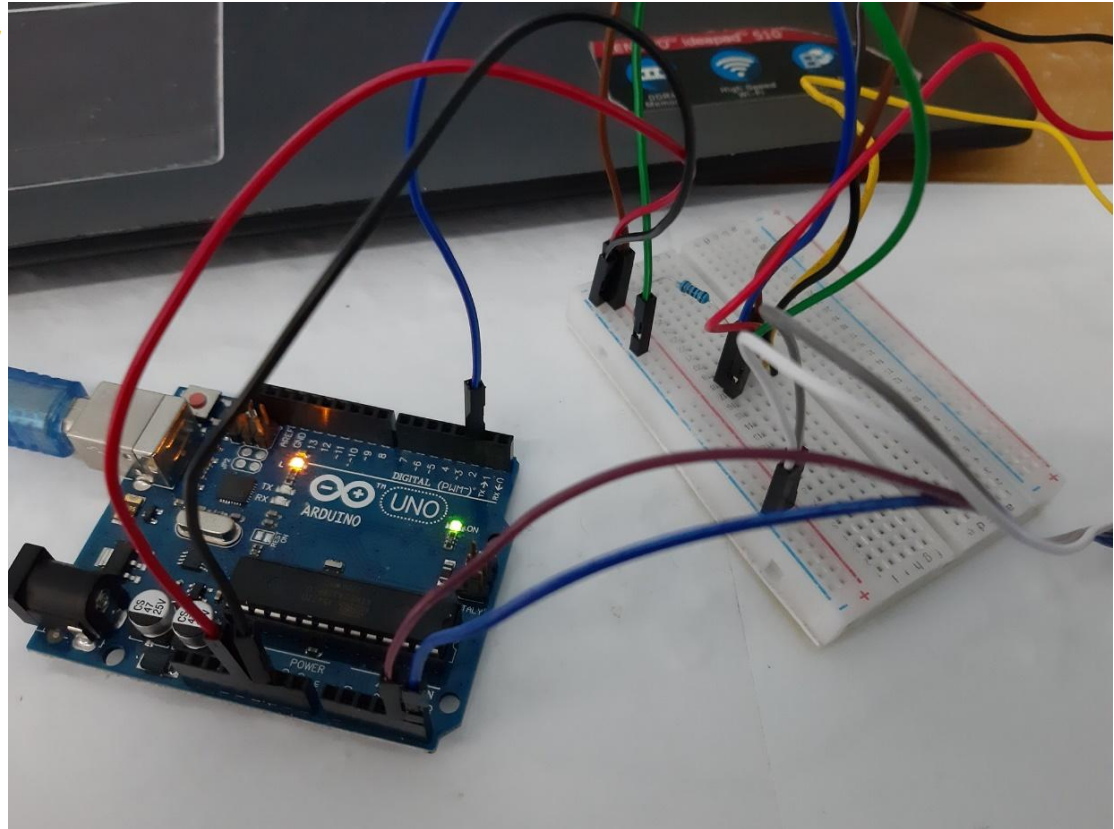
5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.3 วิธีการต่ออุปกรณ์



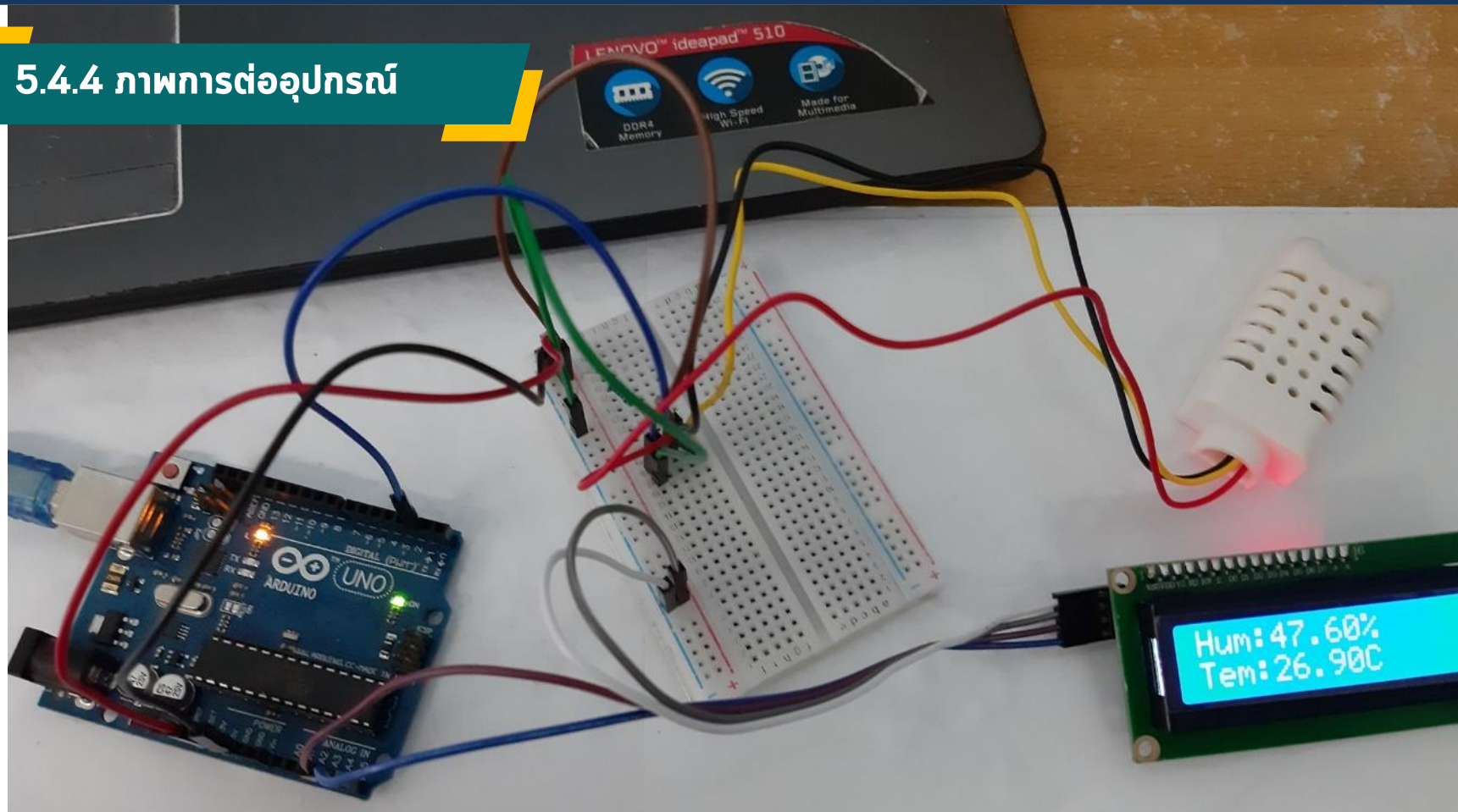
5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.4 ภาพการต่ออุปกรณ์



5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.4 ภาพการต่ออุปกรณ์



5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

1

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht;
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");
  dht.setup(2); // data pin 2
  lcd.begin();
  lcd.backlight();
}
```

5.4.5 โค้ดโปรแกรม

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);
  lcd.setCursor(0, 0);
  lcd.print("Hum:   ");
  lcd.setCursor(4, 0);
```

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.5 โค้ดโปรแกรม

```
lcd.print(humidity);  
lcd.setCursor(9, 0);  
lcd.print("%");  
lcd.setCursor(0, 1);  
lcd.print("Tem:  ");  
lcd.setCursor(4, 1);  
lcd.print(temperature);  
lcd.setCursor(9, 1);  
lcd.print("C");  
delay(500);  
}
```

3

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.5 ไล่โปรแกรม

- ในเว็บไซต์ Wokwi, ขา NC ของ DHT22 หมายถึง "No Connection" หรือ "Not Connected" ซึ่งหมายความว่าคุณไม่ต้องเชื่อมต่อกับอะไรก็ตาม. ในการใช้งานปกติของ DHT22, คุณสามารถปล่อยขา NC นี้ไม่เชื่อมต่อไปกับอะไรก็ได้, ไม่มีการใช้งาน.
- ในกรณีที่คุณใช้เซ็นเซอร์ DHT22 บน Wokwi, คุณสามารถลองทำการสร้างวงจรขึ้นมาในโปรแกรม Wokwi Simulator เพื่อดูการทำงานของเซ็นเซอร์ DHT22 ได้.

3

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.5 ได้โปรแกรม

- ในเว็บไซต์ Wokwi, ขา NC ของ DHT22 หมายถึง "No Connection" หรือ "Not Connected" ซึ่งหมายความว่าเราไม่ต้องเชื่อมขา NC กับอะไรก็ตาม ในการใช้งานโมดูลของ DHT22 ตามภาพประกอบ

3

Editing DHT22
Temperature: -18.4°C
Humidity: 12.5%

```

12 lcd.begin(16, 2); // ระบุขนาดและจำนวนพินที่เชื่อมต่อ
13 lcd.backlight();
14 }
15
16 void loop() {
17   delay(2000); // รอให้เซ็นเซอร์ DHT ทำการตรวจวัด
18
19   float humidity = dht.readHumidity();
20   float temperature = dht.readTemperature();
21
22   Serial.print("OK\t");
23
24   Serial.print(humidity, 1);
25   Serial.print("\t\t");
26   Serial.print(temperature, 1);
27   Serial.print("\t\t");
28   Serial.println(dht.convertCtoF(temperature));
29
30   lcd.clear();
31   lcd.setCursor(0, 0);
32   lcd.print("Hum: ");
33   lcd.setCursor(6, 0);
34   lcd.print(humidity);
35   lcd.setCursor(13, 0);
36   lcd.print("%");
37   lcd.setCursor(0, 1);
38   lcd.print("Tem: ");
39   lcd.setCursor(6, 1);
40   lcd.print(temperature);
41   lcd.setCursor(13, 1);
42   lcd.print("C");
43
44   delay(500);
45 }
46
  
```

OK	nan	nan	nan
OK	nan	nan	nan
OK	nan	nan	nan
OK	6.2	-1647.6	-2933.7
OK	6.2	-1647.6	-2933.7
OK	6.2	-1647.6	-2933.7

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.5 ไลด์โปรแกรม

- ในเว็บไซต์ Wokwi, ขา NC ของ DHT22 หมายถึง "No Connection" หรือ "Not Connected" ซึ่งหมายความว่า คุณไม่ต้องเชื่อมขานี้กับอะไรก็ตาม. ในการใช้งานปกติของ DHT22, คุณสามารถปล่อยขา NC นี้ไม่เชื่อมต่อไปกับอะไรก็ได้, ไม่มีการใช้งาน.
- ในกรณีที่ผู้ใช้เซ็นเซอร์ DHT22 บน Wokwi, คุณสามารถลองทำการสร้างวงจรขึ้นมาในโปรแกรม Wokwi Simulator เพื่อดูการทำงานของเซ็นเซอร์ DHT22 ได้.
- เซ็นเซอร์ DHT22 ไม่มีขา SDA เนื่องจาก DHT22 เป็นเซ็นเซอร์ที่ใช้โปรโตคอล 1-Wire เท่านั้น ที่ใช้ขาเดียวสำหรับการสื่อสาร. ขาที่สำคัญสำหรับ DHT22 คือ:
- VCC (หรือ VDD): ใช้ในการจ่ายไฟเข้าสู่เซ็นเซอร์ (Voltage Supply).
- GND (หรือ G): ใช้ในการเชื่อมต่อกับ GND (Ground).
- Data (หรือ Signal): ใช้ในการรับข้อมูลจากเซ็นเซอร์ (Data Line).
- ในการใช้งานบน Wokwi Simulator, คุณสามารถลองเชื่อมต่อขาทั้งสามนี้ในโปรแกรม Wokwi Simulator เพื่อดูการทำงานของ DHT22 ได้.

The logo for Wokwi, consisting of the word "wokwi" in a white, lowercase, sans-serif font, centered within a blue rounded rectangle.

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.5 ได้โปรแกรม

test_lcd_heat01 | Arduino 1.8.12

File Edit Sketch Tools Help

test_lcd_heat01

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"
LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht;
void setup()
{
    Serial.begin(9600);
    Serial.println();
    Serial.println("Status\tHumidity (%) \tTemperature (C) \t(F)");
    dht.setup(2); // data pin 2
    lcd.begin();
    lcd.backlight();
}
void loop() {
    delay(dht.getMinimumSamplingPeriod());

    float humidity = dht.getHumidity(); // ดึงค่าความชื้น
```

test_lcd_heat01 | Arduino 1.8.12

File Edit Sketch Tools Help

test_lcd_heat01

```
Serial.println();
Serial.println("Status\tHumidity (%) \tTemperature (C) \t(F)");
dht.setup(2); // data pin 2
lcd.begin();
lcd.backlight();

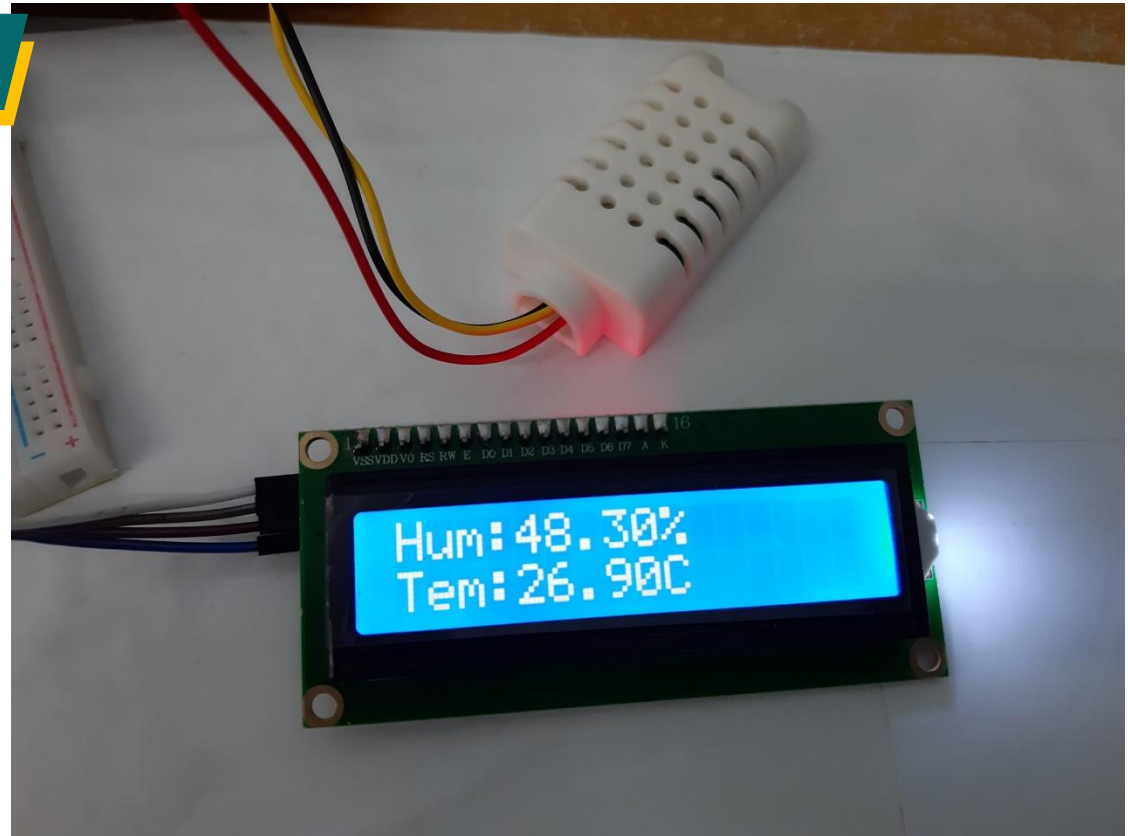
}
void loop() {
    delay(dht.getMinimumSamplingPeriod());

    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);
    lcd.setCursor(0, 0);
    lcd.print("Hum:      ");
    lcd.setCursor(4, 0);
    lcd.print(humidity);
    lcd.setCursor(9, 0);
    lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print("Tem:      ");
    lcd.setCursor(4, 1);
    lcd.print(temperature);
    lcd.setCursor(9, 1);
    lcd.print("C");
    delay(500);
```

5.4 โปรแกรมแสดงอุณหภูมิ และความชื้นออกทางหน้าจอ LCD 1602

5.4.6 ภาพการต่อวงจร



5.5

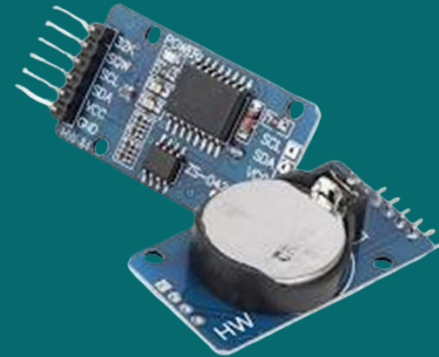
โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

***** สมมุติให้กำหนดเวลาในการวัดชิ้นและอุณหภูมิตอน
08.00 และ 18.00 น.**

5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

5.5.1 อุปกรณ์ที่ต้องใช้

- Arduino UNO R3 พร้อม สายUSB
- DS3231 Module โมดูลนาฬิกา
- สายไฟจัมเปอร์ ผุ้-ผุ้
- สายไฟจัมเปอร์ ผุ้-เมีย
- บอร์ดทดลอง Breadboard
- Resistor ตัวต้านทาน 1 อัน
- หลอด LED 1 อัน



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

5.5.2 ติดตั้ง Libraries นาฬิกา

ดาวน์โหลด Libraries และทำการแตกไฟล์ลงใน Folder: libraries

- <http://www.mediafire.com/download/w2vnjcmiv377hpj/RTCLib.rar>

5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

5.5.3 ตัวอย่างการใช้งานโมดูล DS3231

- นาฬิกาตั้งโต๊ะ: แสดงเวลาและวันที่บนจอแสดงผลแบบ LCD หรือ LED
- เครื่องบันทึกข้อมูล (Data Logger): เก็บข้อมูลเวลาที่แน่นอนสำหรับบันทึกข้อมูลจากเซ็นเซอร์ต่างๆ เช่น อุณหภูมิ ความชื้น หรือการเคลื่อนไหว
- ระบบเตือนความจำ: สร้างระบบเตือนความจำสำหรับการทำงานต่างๆ เช่น กินยา หรือการตรวจสอบเครื่องจักร
- ระบบควบคุมอัตโนมัติ: ใช้ในระบบควบคุมที่ต้องการการตั้งเวลา เช่น การเปิด-ปิดไฟ หรือการควบคุมระบบน้ำหยด
- ระบบการบันทึกเวลาเรียน: ใช้ในการบันทึกเวลาเข้าเรียนและเวลาเลิกเรียนของนักเรียน



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

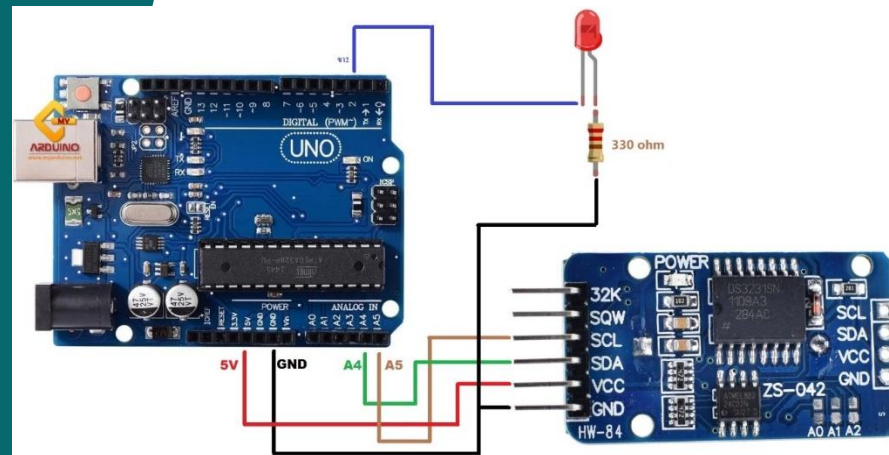
5.5.4 วิธีการต่ออุปกรณ์

Arduino uno r3 -> หลอดไฟ LED

- ขา2 -> LED1

Arduino uno r3 -> โมดูลนาฬิกา DS3231

- A4 -> SDA
- A5 -> SCL
- 5V -> VCC
- GND -> GND



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

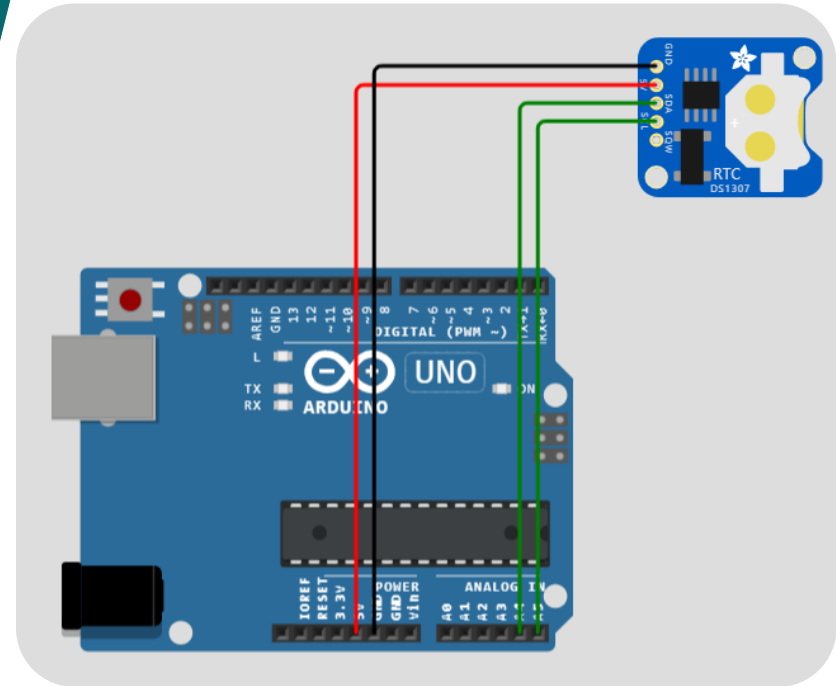
5.5.5 วิธีการต่ออุปกรณ์ Wokwi

Arduino uno r3 -> หลอดไฟ LED

- ขา2 -> LED1

Arduino uno r3 -> โมดูลนาฬิกา DS3231

- A4 -> SDA
- A5 -> SCL
- 5V -> VCC
- GND -> GND



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

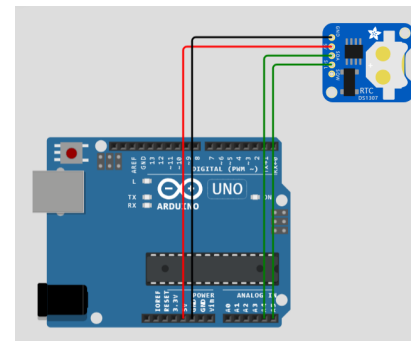
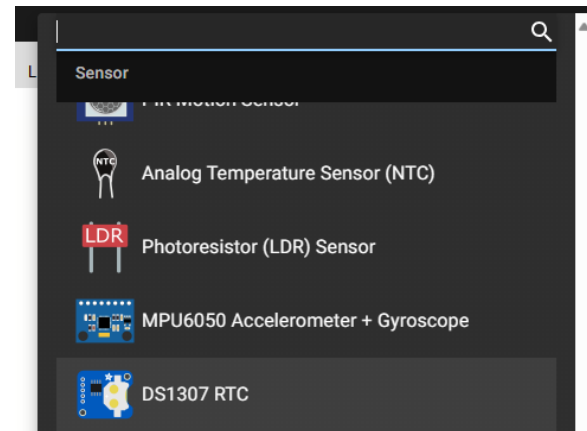
5.5.6 ติดตั้ง Libraries RTC

Arduino uno r3 -> หลอดไฟ LED

- ขา2 -> LED1

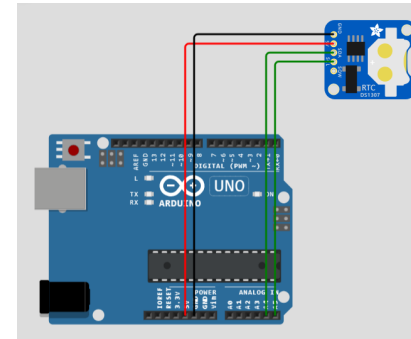
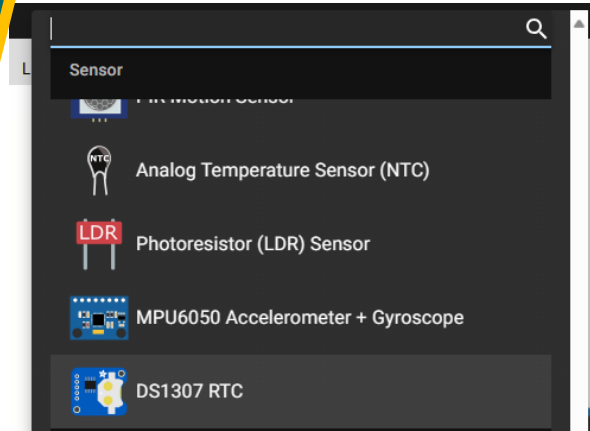
Arduino uno r3 -> โมดูลนาฬิกา DS3231

- A4 -> SDA
- A5 -> SCL
- 5V -> VCC
- GND -> GND



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

5.5.6 ติดตั้ง Libraries RTC

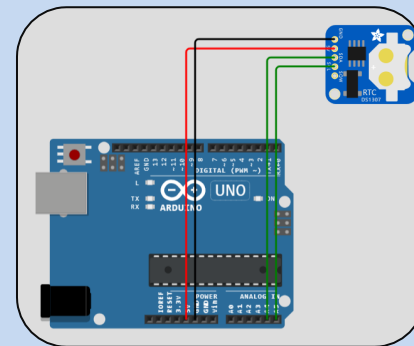


Add Libraries

5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

โปรแกรมแสดงเวลา ด้วย DS1307 RTC ใน wokwi (Part_1)

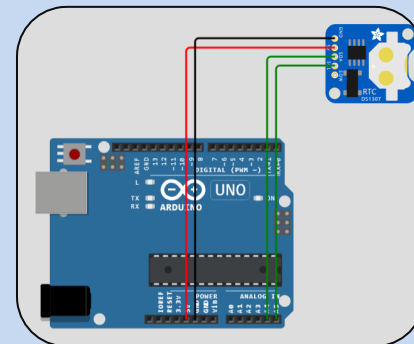
```
#include <Wire.h>
#include <RTClib.h>
RTC_DS1307 rtc;
void setup() {
  Serial.begin(9600);
  if (!rtc.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if (!rtc.isrunning()) {
    Serial.println("RTC is NOT running!");
    // The following line sets the RTC to the date & time this sketch was compiled
    rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}
```



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

โปรแกรมแสดงเวลา ด้วย DS1307 RTC ใน wokwi (Part_2)

```
void loop() {  
  DateTime now = rtc.now();  
  Serial.print(now.year(), DEC);  
  Serial.print('/');  
  Serial.print(now.month(), DEC);  
  Serial.print('/');  
  Serial.print(now.day(), DEC);  
  Serial.print(" ");  
  Serial.print(now.hour(), DEC);  
  Serial.print(':');  
  Serial.print(now.minute(), DEC);  
  Serial.print(':');  
  Serial.print(now.second(), DEC);  
  Serial.println();  
  delay(1000);  
}
```



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

โปรแกรมแสดง เวลา ด้วย DS1307 RTC ใน wokwi (Part_2)

WOKWI

SAVE

SHARE

sketch.ino

README.md

diagram.json

libraries.txt

Library Manager

Simulation

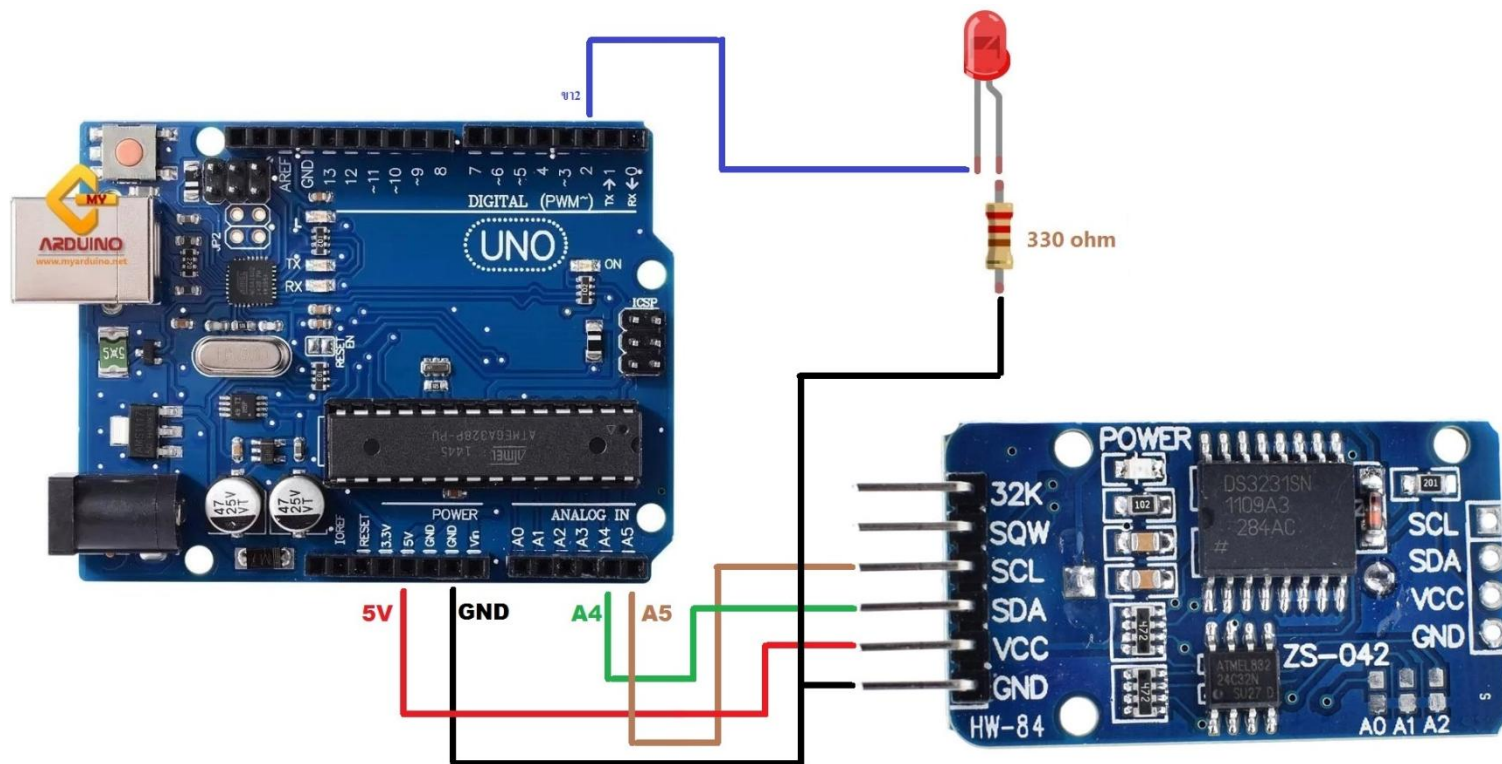
Description

```

1  #include <Wire.h>
2  #include <RTClib.h>
3
4  RTC_DS1307 rtc;
5
6  void setup() {
7    Serial.begin(9600);
8
9    if (!rtc.begin()) {
10     Serial.println("Couldn't find RTC");
11     while (1);
12    }
13
14    if (!rtc.isrunning()) {
15     Serial.println("RTC is NOT running!");
16     // The following line sets the RTC to the date & time this sketch was
17     // compiled
18     rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
19    }
20
21    void loop() {
22     DateTime now = rtc.now();
23
24     Serial.print(now.year(), DEC);
25     Serial.print('/');
26     Serial.print(now.month(), DEC);
27     Serial.print('/');
28     Serial.print(now.day(), DEC);
29     Serial.print(" ");
30     Serial.print(now.hour(), DEC);
31     Serial.print(':');
32     Serial.print(now.minute(), DEC);
33     Serial.print(':');
34     Serial.print(now.second(), DEC);
35     Serial.println();
36
37     delay(1000);
38    }
39

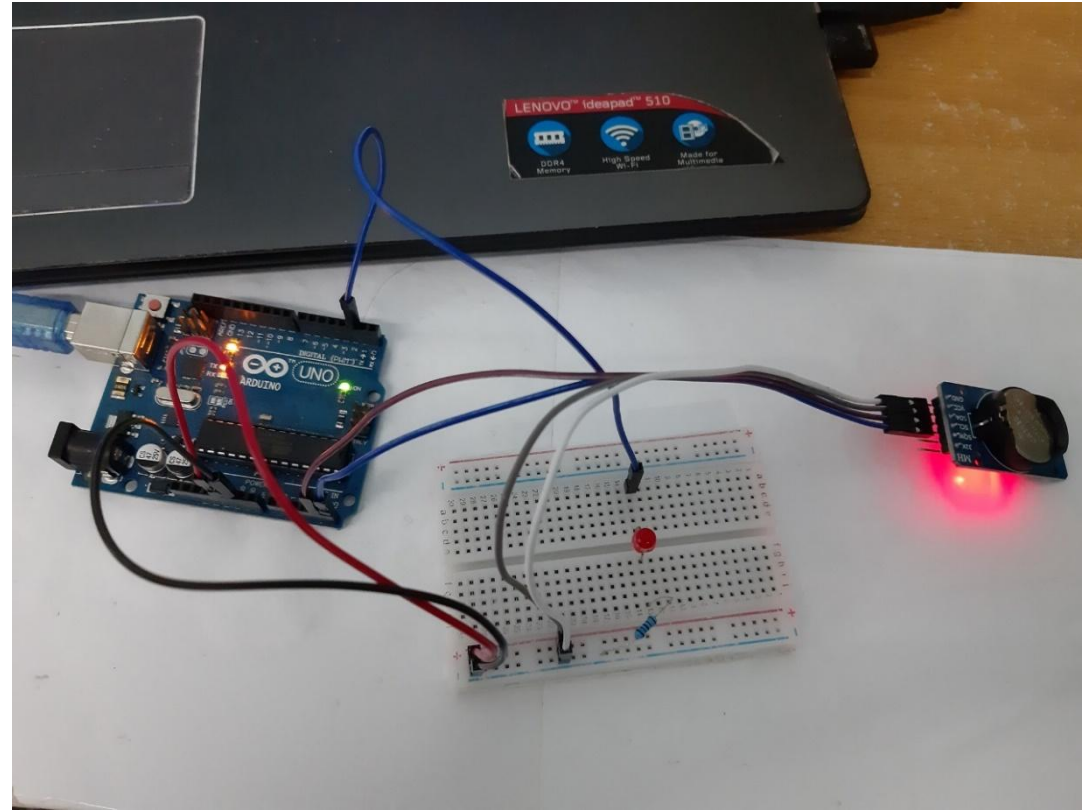
```

5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

ภาพการต่อ
วงจร



5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

Code
โปรแกรม

1

```
#include "Wire.h"
#include "SPI.h"
#include "RTCLib.h"

RTC_DS3231 RTC;
int ledPin1 = 2;
void setup () {
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  pinMode(ledPin1, OUTPUT);
  RTC.adjust(DateTime(__DATE__, __
_TIME__));
}
```

2

```
void loop () {
  DateTime now = RTC.now();
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
}
```

5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

Code
โปรแกรม

```
if (now.hour() == 14 && now.minute() == 30 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 30 จะสั่งให้เปิดไฟ LED
    digitalWrite(ledPin1, HIGH); // สั่งให้ LED ติดสว่าง
    Serial.println("LED ON");
} else if (now.hour() == 14 && now.minute() == 35 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 35 จะสั่งให้ปิดไฟ LED
    digitalWrite(ledPin1, LOW); // สั่งให้ LED ดับ
    Serial.println("LED OFF");
} else if (now.hour() == 14 && now.minute() == 40 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 40 จะสั่งให้เปิดไฟ LED
    digitalWrite(ledPin1, HIGH); // สั่งให้ LED ติดสว่าง
    Serial.println("LED ON");
} else {
    Serial.println("LED OFF");
    digitalWrite(ledPin1, LOW); // สั่งให้ LED ดับ
}
Serial.println();
delay(1000);
}
```

3

หากต้องการแก้ไขการตั้งเวลา
ให้แก้ไขในส่วนนี้

5.5 โปรแกรมตั้งเวลา เปิด-ปิด หลอด LED

หน้าจอตัวอย่าง
Code

test_time_02 | Arduino 1.8.12

File Edit Sketch Tools Help



test_time_02 \$

```
#include "Wire.h"
#include "SPI.h"
#include "RTClib.h"
```

```
RTC_DS3231 RTC;
int ledPin1 = 2;
```

```
void setup () {
  Serial.begin(9600);
  Wire.begin();
  RTC.begin();
  pinMode(ledPin1, OUTPUT);
}
```

```
RTC.adjust(DateTime(__DATE__, __TIME__));
}
```

```
void loop () {
```

```
  DateTime now = RTC.now();
```

```
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
```

```
  if (now.hour() == 14 && now.minute() == 30 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 30 จะสั่งให้เปิดไฟ LED
    digitalWrite(ledPin1, HIGH); // สั่งให้ LED ติดสว่าง
```

test_time_02 | Arduino 1.8.12

File Edit Sketch Tools Help



test_time_02 \$

```
}
```

```
void loop () {
```

```
  DateTime now = RTC.now();
```

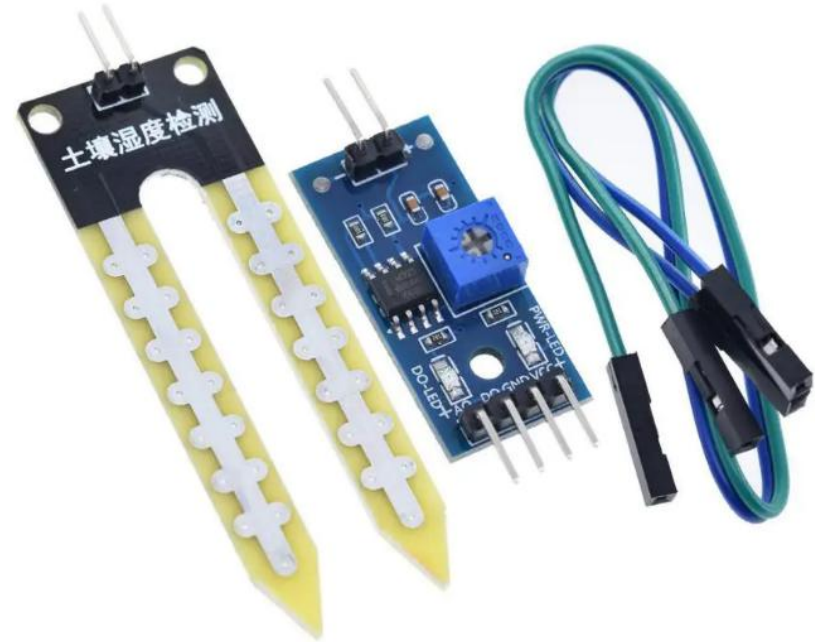
```
  Serial.print(now.year(), DEC);
  Serial.print('/');
  Serial.print(now.month(), DEC);
  Serial.print('/');
  Serial.print(now.day(), DEC);
  Serial.print(' ');
  Serial.print(now.hour(), DEC);
  Serial.print(':');
  Serial.print(now.minute(), DEC);
  Serial.print(':');
  Serial.print(now.second(), DEC);
  Serial.println();
```

```
  if (now.hour() == 14 && now.minute() == 30 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 30 จะสั่งให้เปิดไฟ LED
    digitalWrite(ledPin1, HIGH); // สั่งให้ LED ติดสว่าง
    Serial.println("LED ON");
  } else if (now.hour() == 14 && now.minute() == 35 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 35 จะสั่งให้เปิดไฟ LED
    digitalWrite(ledPin1, LOW); // สั่งให้ LED ดับ
    Serial.println("LED OFF");
  } else if (now.hour() == 14 && now.minute() == 40 ) { //ถ้าชั่วโมงเท่ากับ 14 และนาทีเท่ากับ 40 จะสั่งให้เปิดไฟ LED
    digitalWrite(ledPin1, HIGH); // สั่งให้ LED ติดสว่าง
    Serial.println("LED ON");
  } else {
    Serial.println("LED OFF");
    digitalWrite(ledPin1, LOW); // สั่งให้ LED ดับ
  }
  Serial.println();
  delay(1000);
}
```


5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

soil moisture sensor

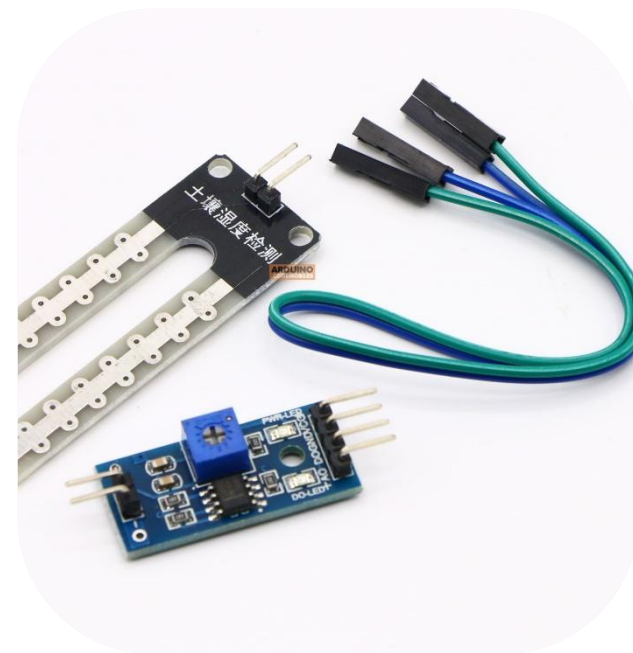
5.6.1 เซ็นเซอร์วัดความชื้น ในดิน



ภาพที่ 5.23 เซ็นเซอร์วัดความชื้น ในดิน (soil moisture sensor)

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

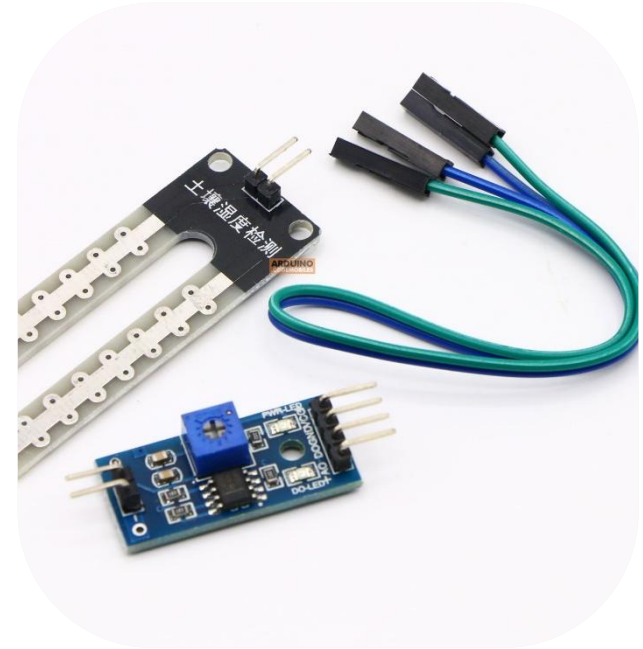
- "soil module sensor" หมายถึงเซ็นเซอร์หรือเซ็นเซอร์โมดูลที่ใช้ในการวัดค่าทางธรรมชาติของดิน เช่น ความชื้นในดิน, อุณหภูมิ, ระดับน้ำในดิน เป็นต้น การวัดค่าเหล่านี้ช่วยให้เกษตรกรและผู้ที่ทำสวนหรือปลูกพืชสามารถตรวจสอบสภาพดินและเพิ่มประสิทธิภาพในการจัดการปลูกพืชให้เหมาะสมและเกิดผลผลิตที่ดีขึ้นได้
- เซ็นเซอร์โมดูลดินส่วนใหญ่มักมีรูปแบบที่ติดตั้งในดินหรือประกอบในตู้เครื่อง และสามารถเชื่อมต่อกับระบบอิเล็กทรอนิกส์หรืออุปกรณ์อื่นๆ เพื่อรับค่าเซ็นเซอร์และนำไปใช้ในการตรวจวัดหรือบันทึกข้อมูลยังมีรูปแบบเซ็นเซอร์ที่ไร้สาย (Wireless) ที่ให้ความสะดวกในการติดตั้งและใช้งาน รวมถึงช่วยให้สามารถตรวจสอบสภาพดินจากระยะไกลได้ ซึ่งเป็นประโยชน์สำหรับการจัดการพื้นที่กว้างใหญ่หรือสวนที่ใหญ่ รวมถึงใช้ในการวิจัยและพัฒนาทางด้านการเกษตรและสิ่งแวดล้อมอย่างกว้างขวาง



5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

อุปกรณ์ที่ต้องใช้:

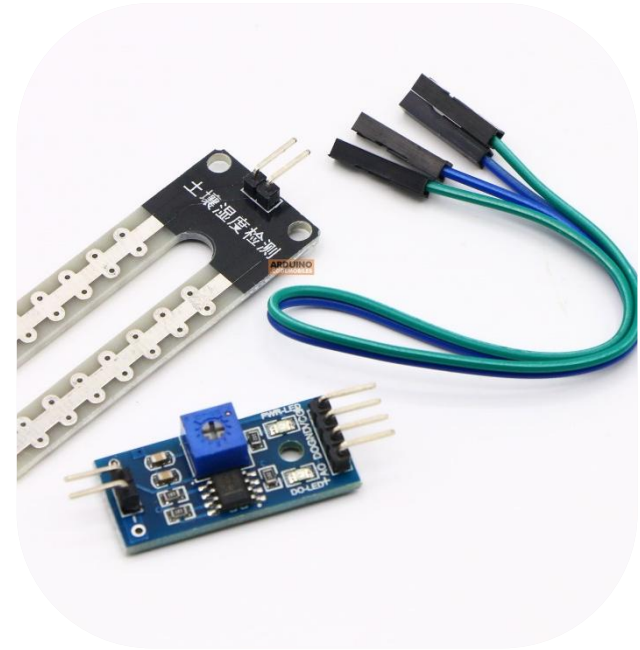
1. Arduino Board (เช่น Arduino Uno)
2. LDR (Light Dependent Resistor)
3. Resistor 10K ohm
4. ตัวต้านทาน 220 ohm
5. LED (ในกรณีที่ต้องการให้มีการแสดงผล)



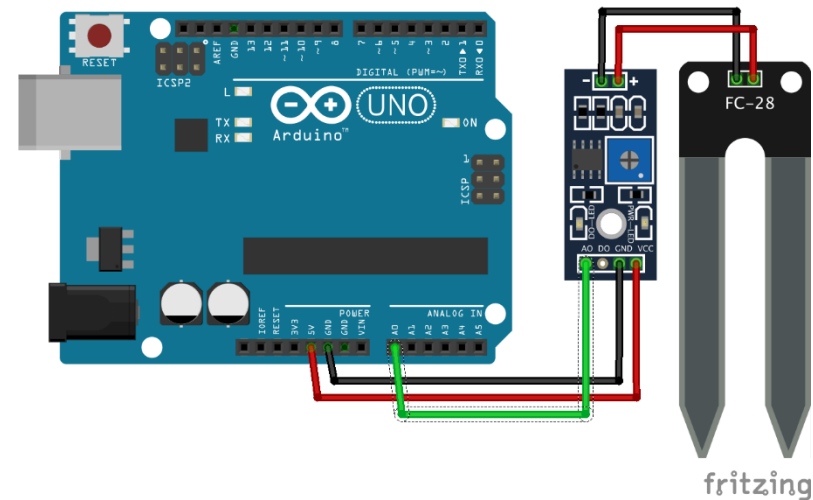
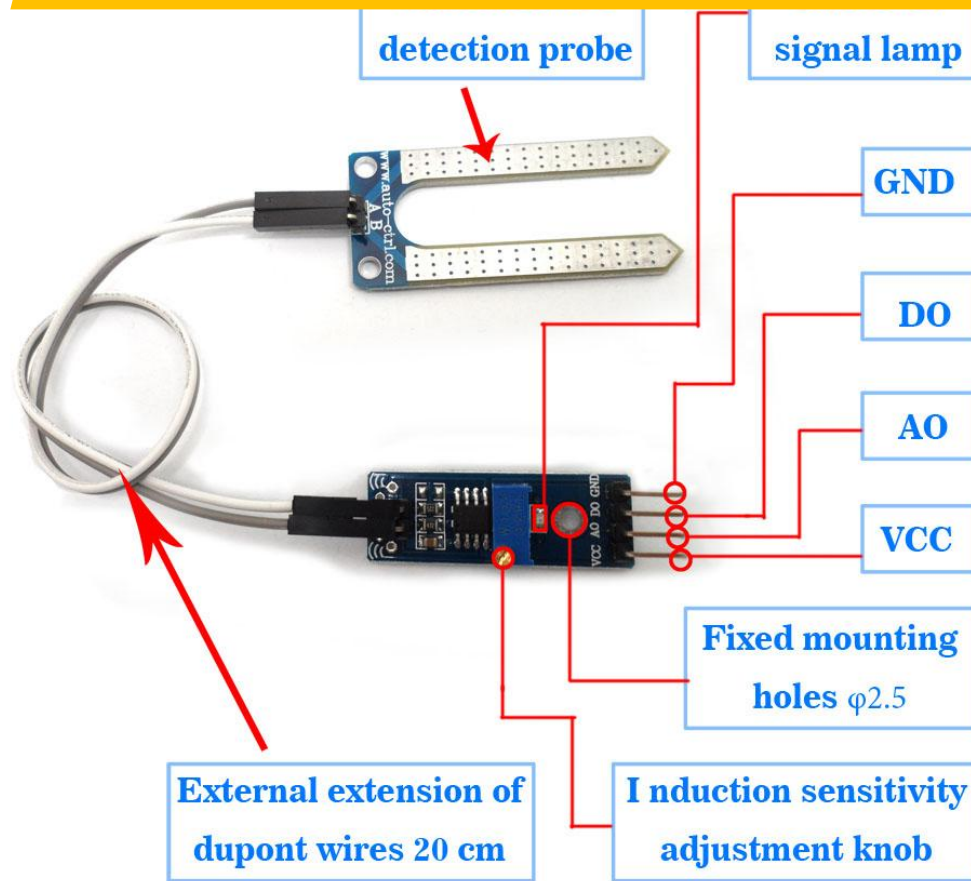
5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

อุปกรณ์ที่ต้องใช้:

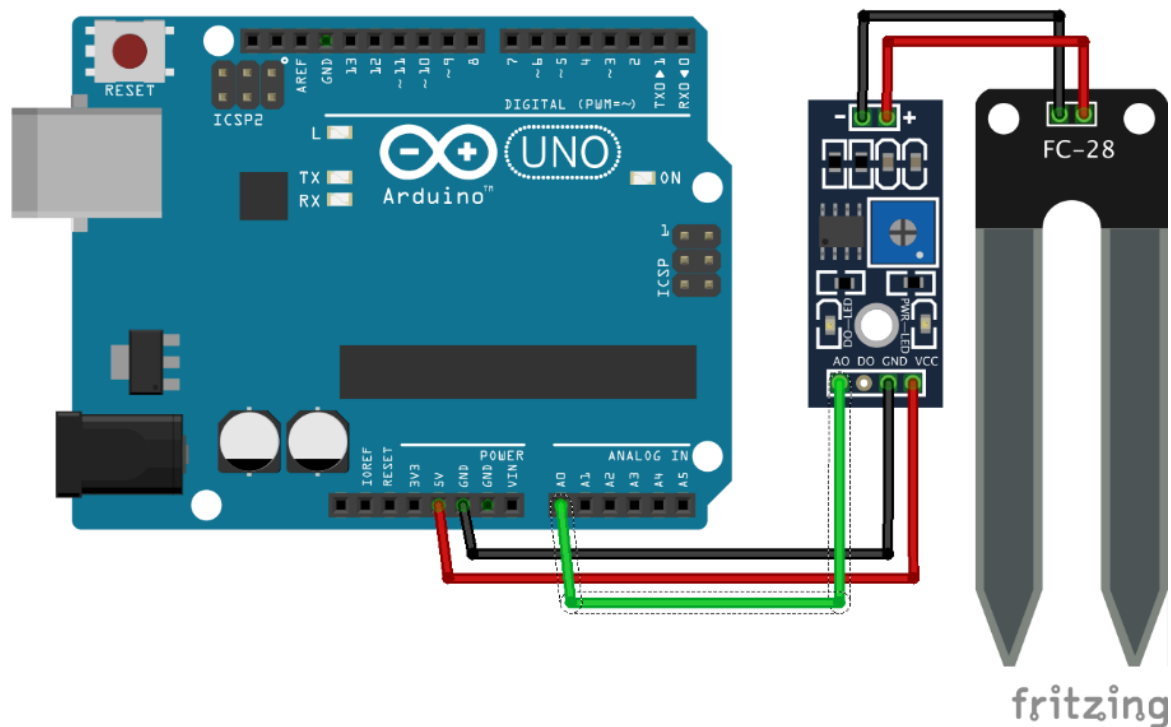
1. Arduino Board (เช่น Arduino Uno)
2. LDR (Light Dependent Resistor)
3. Resistor 10K ohm
4. ตัวต้านทาน 220 ohm
5. LED (ในกรณีที่ต้องการให้มีการแสดงผล)



5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน



5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน



5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

```
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

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

```
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. }
```

แก้ไขความชื้นที่ถูกต้อง

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

- ในโค้ดที่คุณให้มา การแปลงค่าความชื้นเป็นเปอร์เซ็นต์มีบางประเด็นที่ควรพิจารณาเพิ่มเติม:
การแปลงค่าโดยใช้ฟังก์ชัน map:

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

- ฟังก์ชัน map ถูกใช้ในการแปลงค่า sensorValue จากช่วง 0-1023 ไปเป็นช่วง 0-100 ซึ่งในทางทฤษฎีนี้
ถูกต้องสำหรับการแปลงค่าความชื้นให้อยู่ในรูปแบบเปอร์เซ็นต์ อย่างไรก็ตาม ค่าที่อ่านได้จากเซ็นเซอร์วัดความชื้น
ในดินมักจะมีความซับซ้อนกว่านั้น

การคาลิเบรต:

- เพื่อให้ได้ค่าที่แม่นยำ คุณอาจต้องคาลิเบรตเซ็นเซอร์ โดยการทดสอบในสภาวะแวดล้อมที่แตกต่างกัน เช่น ดินแห้ง
สนิท (0%) และดินที่ชื้นเต็มที่ (100%) แล้วจดบันทึกค่าที่อ่านได้จากเซ็นเซอร์ จากนั้นนำค่าเหล่านี้มาแทนที่ใน
ฟังก์ชัน map แทนที่จะใช้ 0 และ 1023 โดยตรง
- การคำนวณค่าความชื้นจากเซ็นเซอร์:
- ค่าที่อ่านได้จากเซ็นเซอร์วัดความชื้นอาจไม่ตรงกับค่าจริงที่ต้องการ

แก้ไขความชื้นที่ต้องการ

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

```
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
```

Output Serial Monitor X

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

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

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

```
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 = AO; //กำหนดขาที่เชื่อมต่อกับเซ็นเซอร์ให้เป็นขา AO 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

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

```
// 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);
}
```

5.6 โปรแกรม เซ็นเซอร์วัดความชื้น ในดิน

```
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  
}
```

ต่อ

แบบฝึกหัด

แบบฝึกหัด

1. จงต่อวงจรในการตั้งเวลาเพื่อวัดอุณหภูมิแล้วความชื้น ทุก ๆ 1 นาที
2. เขียนโปรแกรมเพื่อการตั้งเวลาเพื่อวัดอุณหภูมิแล้วความชื้น ทุก ๆ 1 นาที ผ่าน serial monitor
3. เขียนโปรแกรมเพื่อการตั้งเวลาเพื่อวัดอุณหภูมิแล้วความชื้น ทุก ๆ 1 นาที โดยแสดงผลบนหน้าจอ LCD

แบบฝึกหัด

แบบฝึกหัด

1. ต้องการให้แสดงอุณหภูมิ และ ความชื้น ด้วย DHT22 บนจอ LCD
2. ต้องการให้แสดงระยะห่าง ด้วย Ultrasonic บนจอ LCD
3. ให้วัดความชื้นในดิน แล้วแสดงบน LCD
4. ให้วัดระยะห่าง ถ้าเกิน 10 CM ให้หลอด LED ติดสว่าง ถ้าน้อยกว่าให้ดับ
5. ----- ให้ 5 คน ต่อกลุ่ม -----

อ้างอิง

อ้างอิง

- <http://www.siam2dev.com>
- www.myarduino.net

กรณีศึกษา

ป๊มน้ำสร้างรอยยิ้ม WeWy





สาขาวิชาวิศวกรรมดิจิทัล ม.รังสิต

19 ชม. · 🌐

👉 **ขอแสดงความยินดี** กับนักศึกษาสาขาวิศวกรรมดิจิทัล นำเสนอแผนธุรกิจกับคณะกรรมการจากธนาคารออมสิน ในโครงการ SMART STARTUP COMPANY การพัฒนาไปสู่การเป็นผู้ประกอบการ SME Startup ในอนาคตได้รับทุนสนับสนุนการจัดตั้งธุรกิจนวัตกรรมเริ่มต้นจำนวน 200,000 บาท 🙌👍

📍 เรียนครบรอบใจหายด้านไอที และนวัตกรรมทันสมัย ที่ทุกองค์กรต้องการ

📅 สมัครและลงทะเบียนเรียนตั้งแต่วันที่ - 31 พ.ค. 66 รับทุน RSU Early Bird 5,000 บาท

🔥 เปิดรับสมัครนักศึกษาใหม่ ระดับปริญญาตรี ภาคการศึกษา S/2566

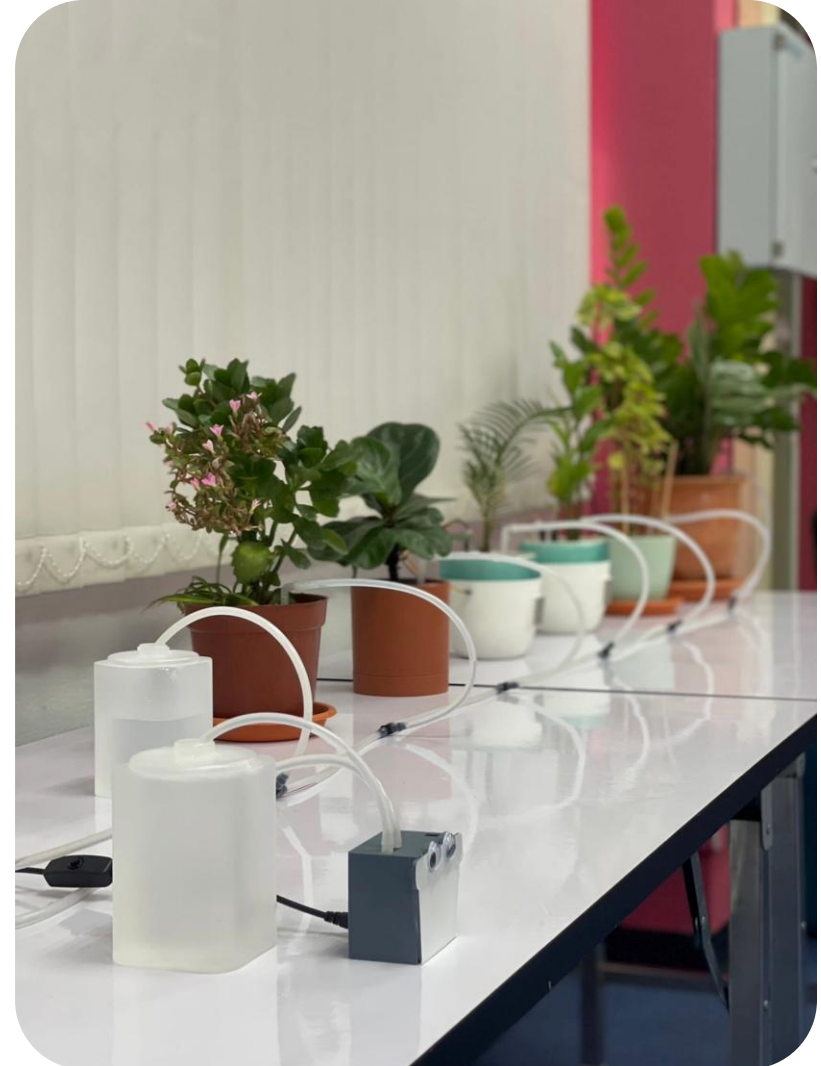
📞 สมัครผ่านระบบออนไลน์ : <https://admissiononline.rsu.ac.th>

📍 อย่าลืมนะสาขาวิชาที่สมัครเรียนเป็น "1402-สาขาวิชาวิศวกรรมดิจิทัล"

"1407-สาขาวิชาวิศวกรรมดิจิทัล" (ภาคพิเศษ)

📱 ทักไลน์ มาสอบถามข้อมูลเพิ่มเติมได้ที่ : @ITranssit

#นวัตกรรมดิจิทัล #DITE



ป๊มน้ำสร้างรอยยิ้ม WeWy



Thank you