หน่วยการเรียนรู้ที่ 2 การเขียนโปรแกรมโดยใช้ Library และการเชื่อมต่อพอร์ตอนุกรม I2C

นายธงชัย ชาบุดศรี
แผนกวิชาเทคโนโลยีสารสนเทศ

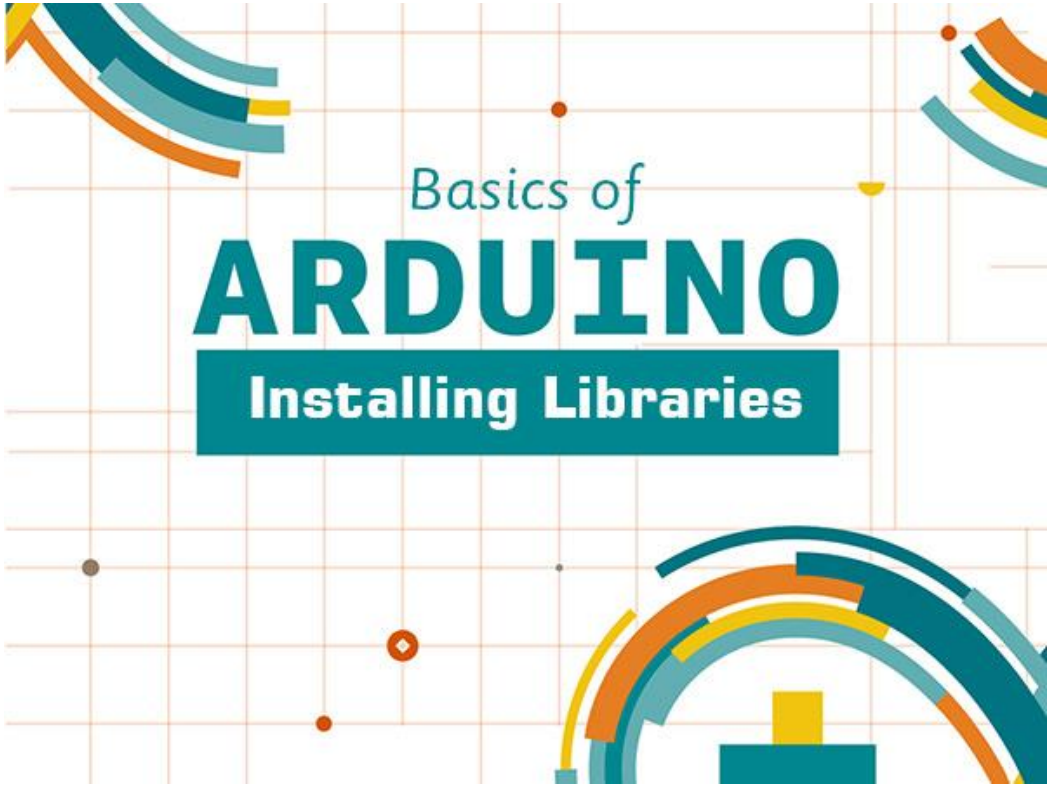
วิทยาลัยเทคนิคชลบุรี

จุดประสงค์การเรียนรู้

1. เรียนรู้โครงสร้าง และหลักการทำงานของไลบรารี่ (Library)

2. เรียนรู้รูปแบบการสื่อสารประเภท I2C

ใบงานที่ 2.1 การค้นหา และติดตั้งไลบรารี่ (Library)
ทดสอบโดยใช้เซนเซอร์วัดอุณหภูมิ และความชื้น DHT22



ใบงานที่ 2.1 ทฤษฎีเบื้องต้น

Library in Arduino IDE

ในการเขียนโปรแกรม Arduino นั้นจำเป็นต้องใช้เซ็นเซอร์และอุปกรณ์ต่อพ่วงต่างๆ ซึ่งอาจต้องเขียนโค้ดยาวและซับซ้อนในการเรียกใช้งานอุปกรณ์ **เพื่อความสะดวกและรวดเร็ว** และ**เขียนโปรแกรมได้ง่าย**ขึ้น จึงมีการสร้าง Library ขึ้นมาใช้งาน **การจะเรียกใช้งาน Library ของแต่ละอุปกรณ์ได้นั้น ต้องทำการติดตั้งบน Arduino IDE ก่อน**

Library คือ ชุดโค้ดโปรแกรมที่มีคนอื่นเขียนไว้แล้ว เช่น การใช้งานเซนเซอร์หรือโมดูลต่างๆ โดยแค่เพิ่ม Library บน Arduino IDE ก็สามารถนำมาใช้งานได้ทันที

วิธีการเรียกใช้ Library

```
#include <Library_name.h>
```

ตัวอย่างการเรียกใช้ Library ของ DHT Sensor

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

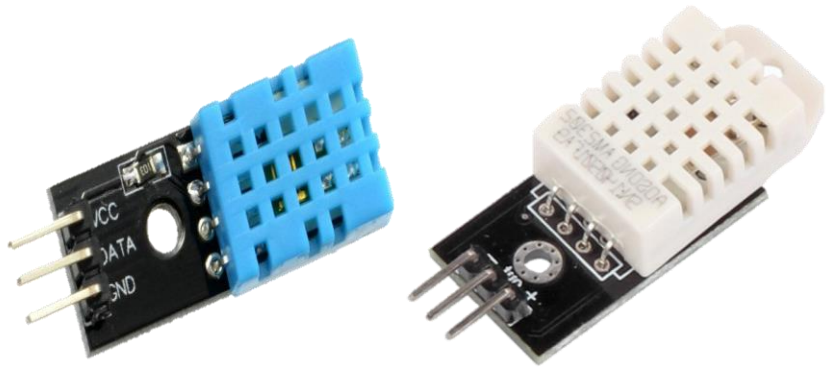
ใบงานที่ 2.1 ทฤษฎีเบื้องต้น

วิธีการติดตั้ง Library บน Arduino IDE

วิธีการติดตั้ง Library บน Arduino IDE นั้นสามารถทำได้ 3 วิธี

1. ติดตั้ง Library ใน Arduino IDE โดยใช้ Library Manager
2. การเพิ่ม Library ด้วยไฟล์ .ZIP
3. Manual Installation

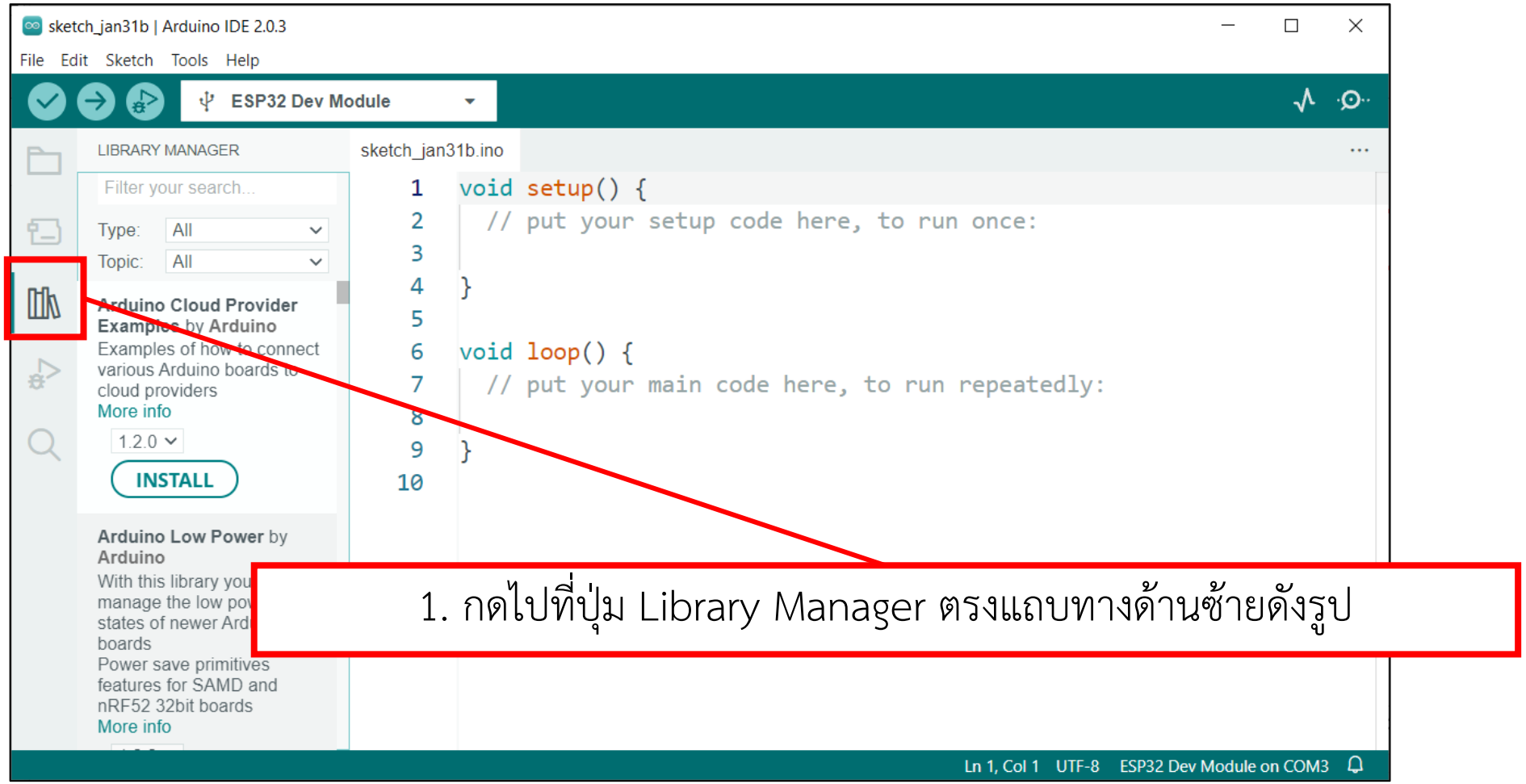
โดยเราจะยกตัวอย่างการ Download Library ของ DHT Sensor



DHT11 และ DHT22

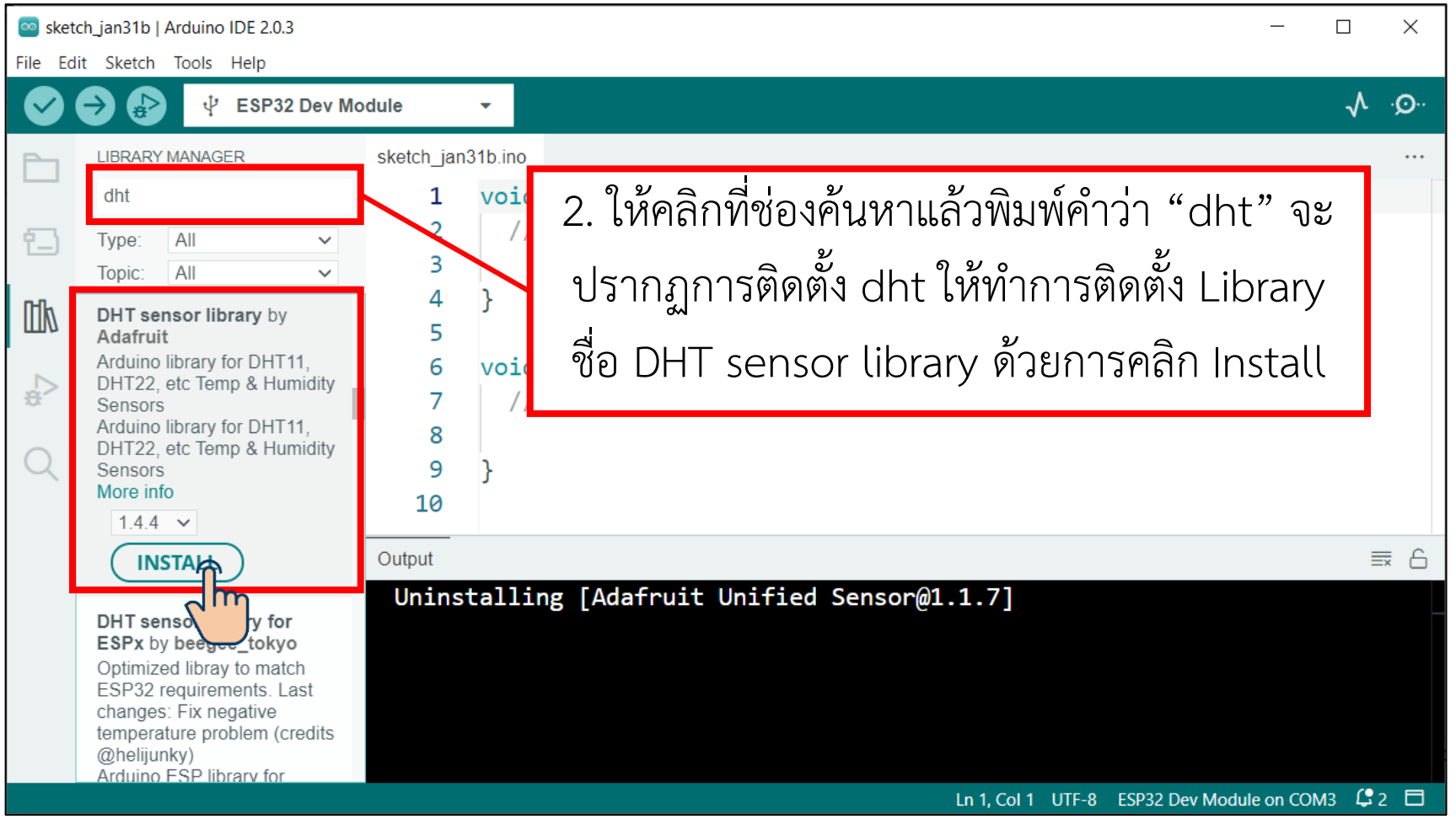
ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 1 การติดตั้ง ผ่าน Library Manager



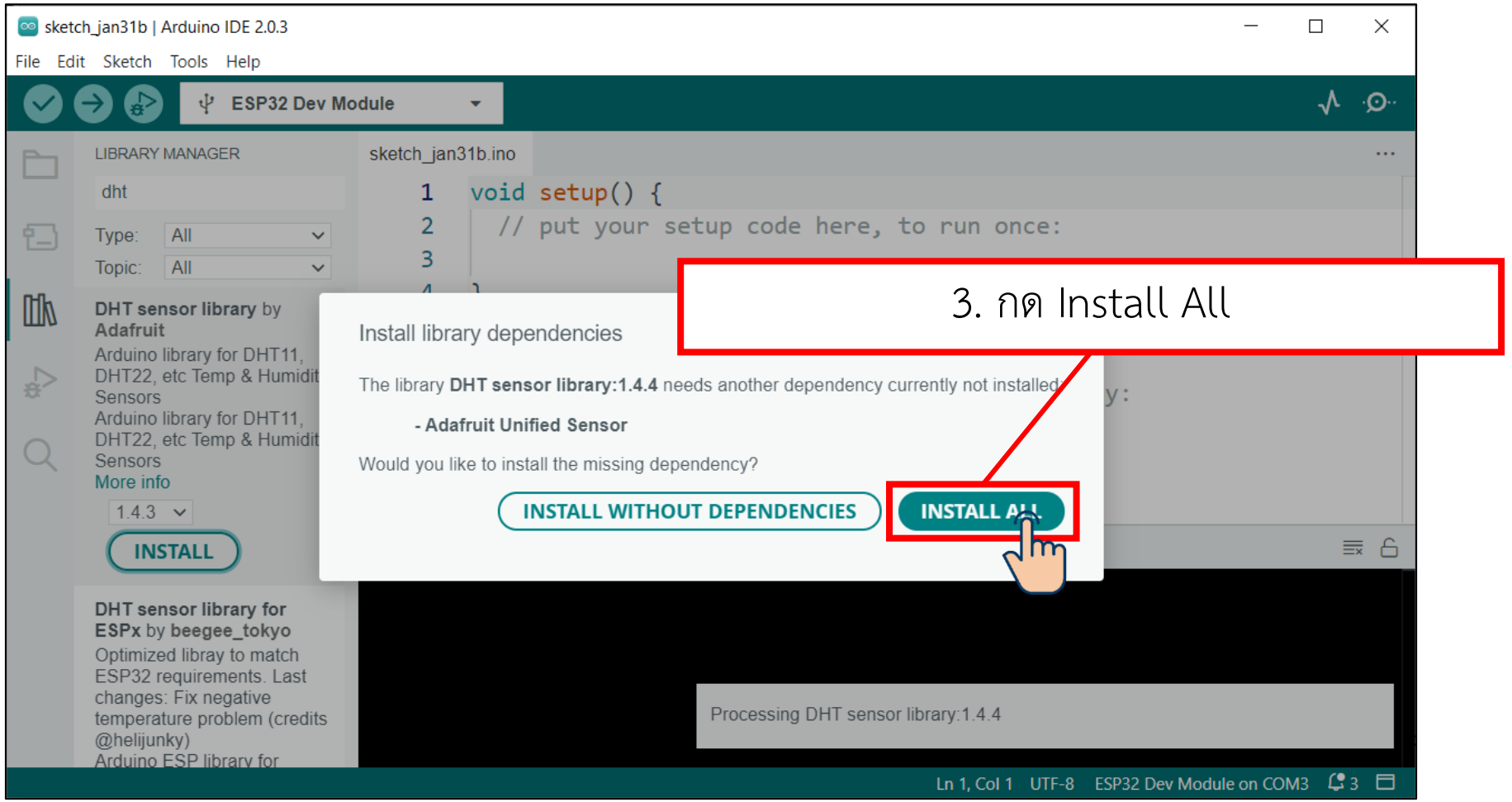
ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 1 การติดตั้ง ผ่าน Library Manager



ใบงานที่ 2.1 ขั้นตอนการทดลอง

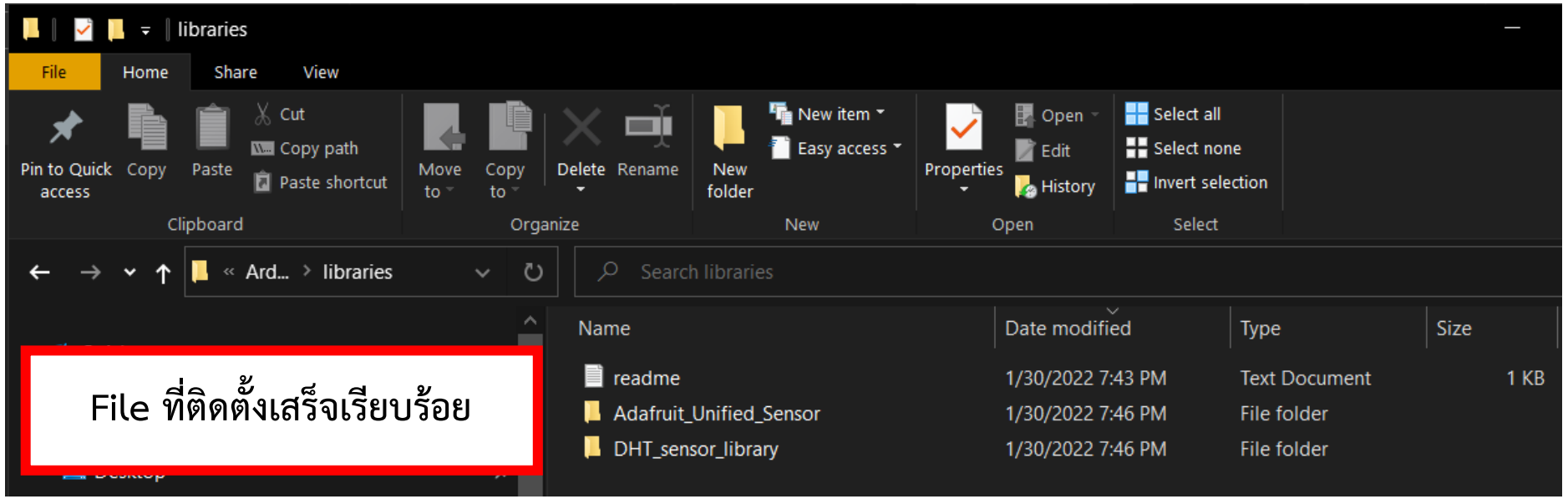
การทดลองที่ 1 การติดตั้ง ผ่าน Library Manager



ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 1 การติดตั้ง ผ่าน Library Manager

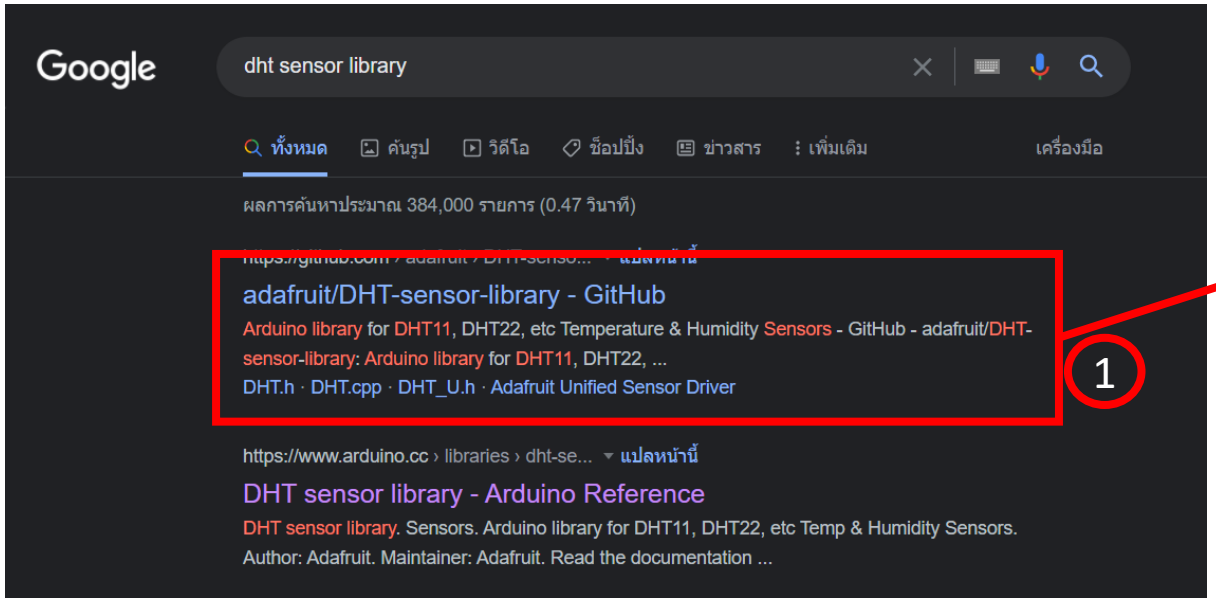
4. เมื่อติดตั้งเสร็จแล้ว ไฟล์ Library จะถูกสร้างขึ้นที่ ~/Documents/Arduino/libraries/. บน Windows จะอยู่ที่ My Documents\Arduino\libraries

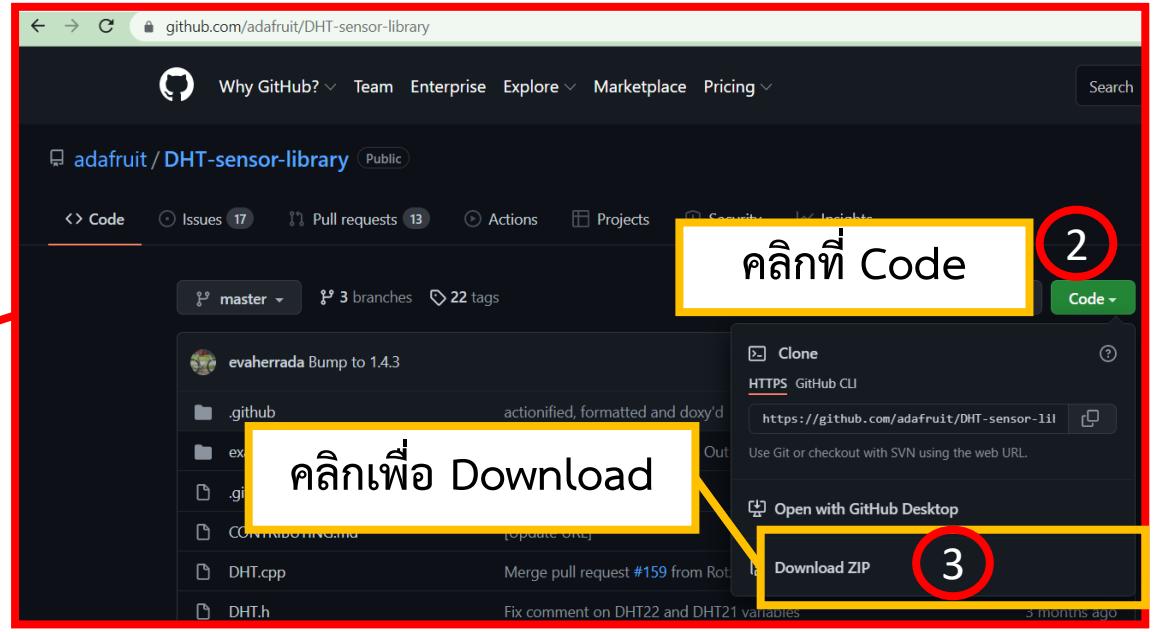


ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 2 การติดตั้ง ผ่าน การเพิ่มด้วย .zip

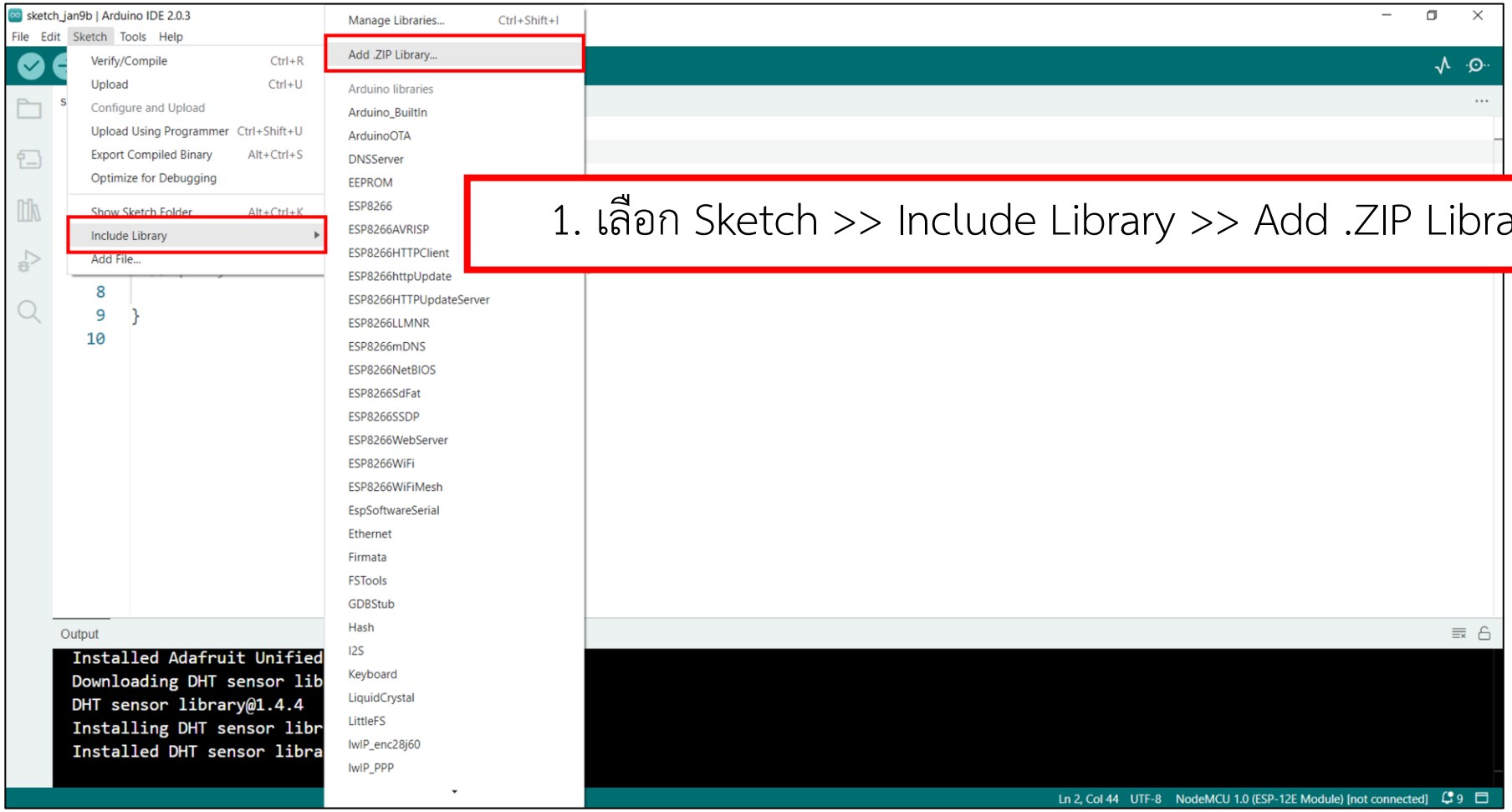
การติดตั้ง Library ด้วยวิธีนี้ เราจะต้องมีไฟล์ Library ที่เราต้องการไว้เตรียมพร้อมอยู่แล้ว โดยเราสามารถ Download Library ของอุปกรณ์ต่างๆ ได้ที่ https://www.arduino.cc/reference/en/libraries/ หรือ สามารถพิมพ์ชื่อ Sensor library เพื่อค้นหาใน Google ได้เลยดังรูปด้านล่าง





ใบงานที่ 2.1 ขั้นตอนการทดลอง

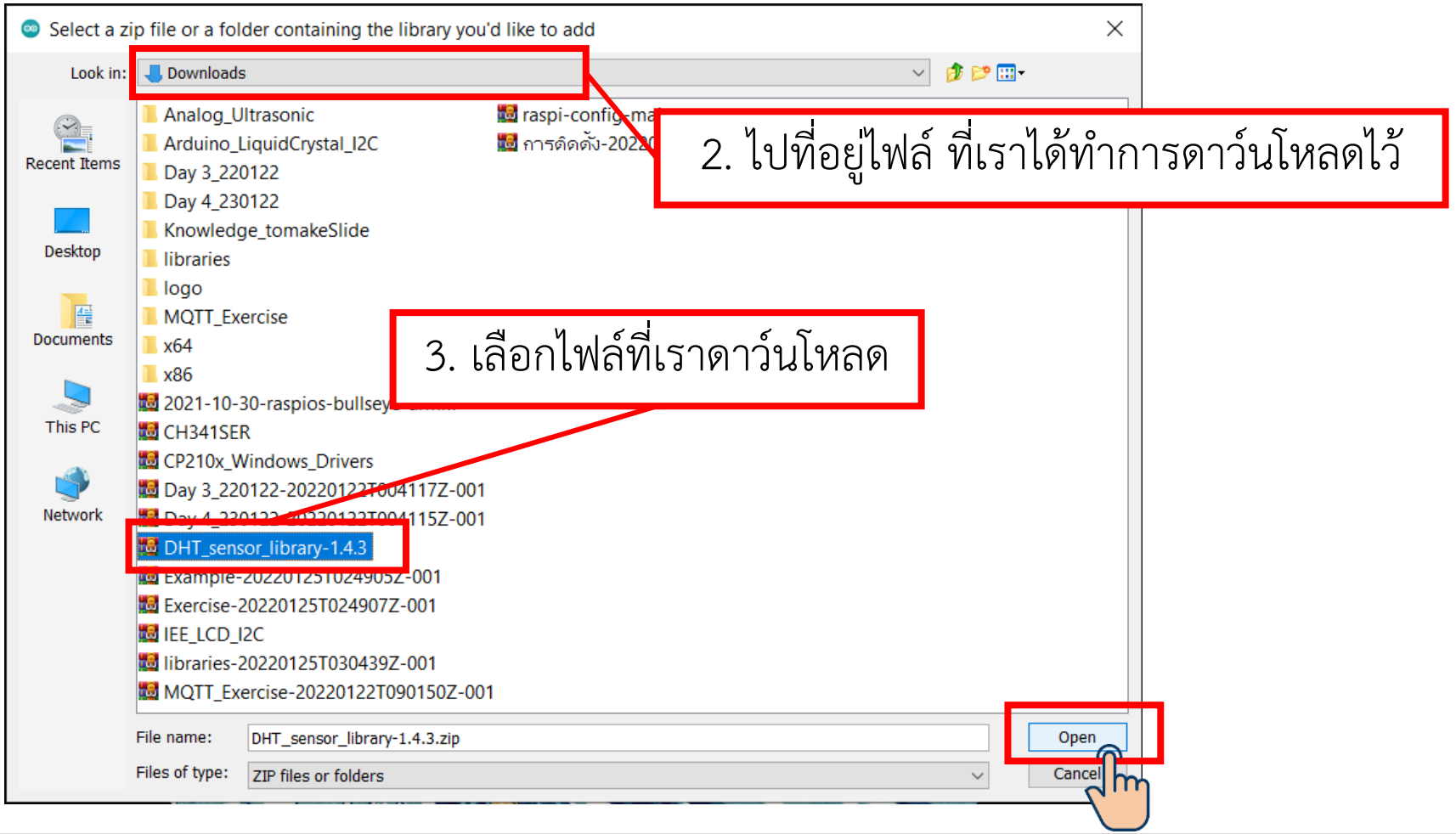
การทดลองที่ 2 การติดตั้ง ผ่าน การเพิ่มด้วย .zip



1. เลือก Sketch >> Include Library >> Add .ZIP Library.... ดังรูป

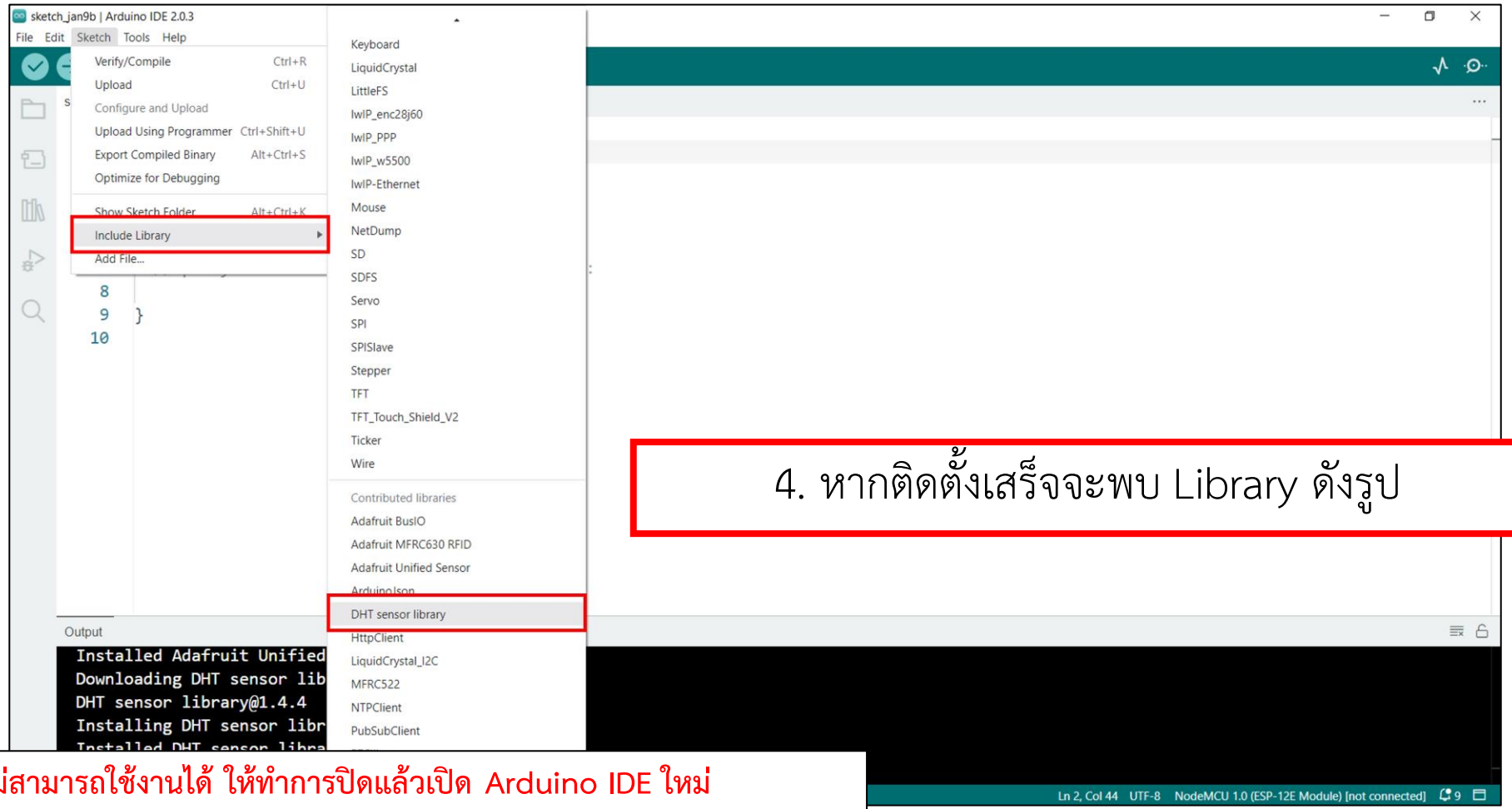
ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 2 การติดตั้ง ผ่าน การเพิ่มด้วย .zip



ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 2 การติดตั้ง ผ่าน การเพิ่มด้วย .zip



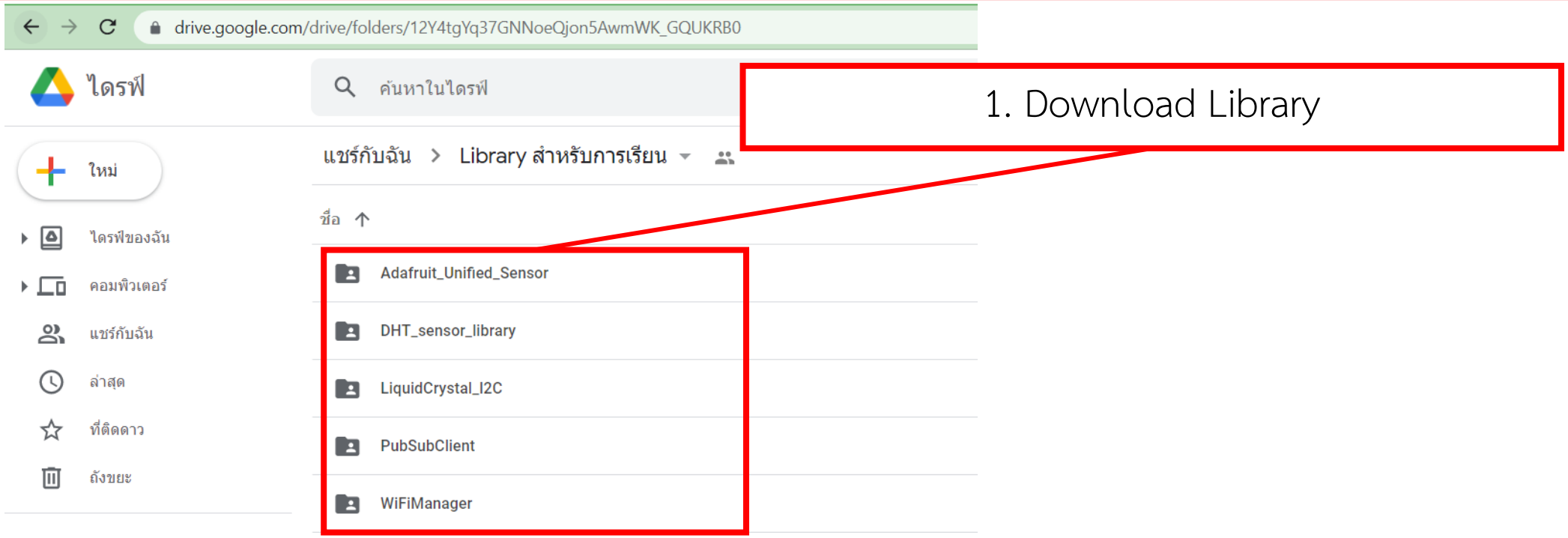
4. หากติดตั้งเสร็จจะพบ Library ดังรูป

หมายเหตุ: หากไม่สามารถใช้งานได้ ให้ทำการปิดแล้วเปิด Arduino IDE ใหม่

ใบงานที่ 2.1 ขั้นตอนการทดลอง

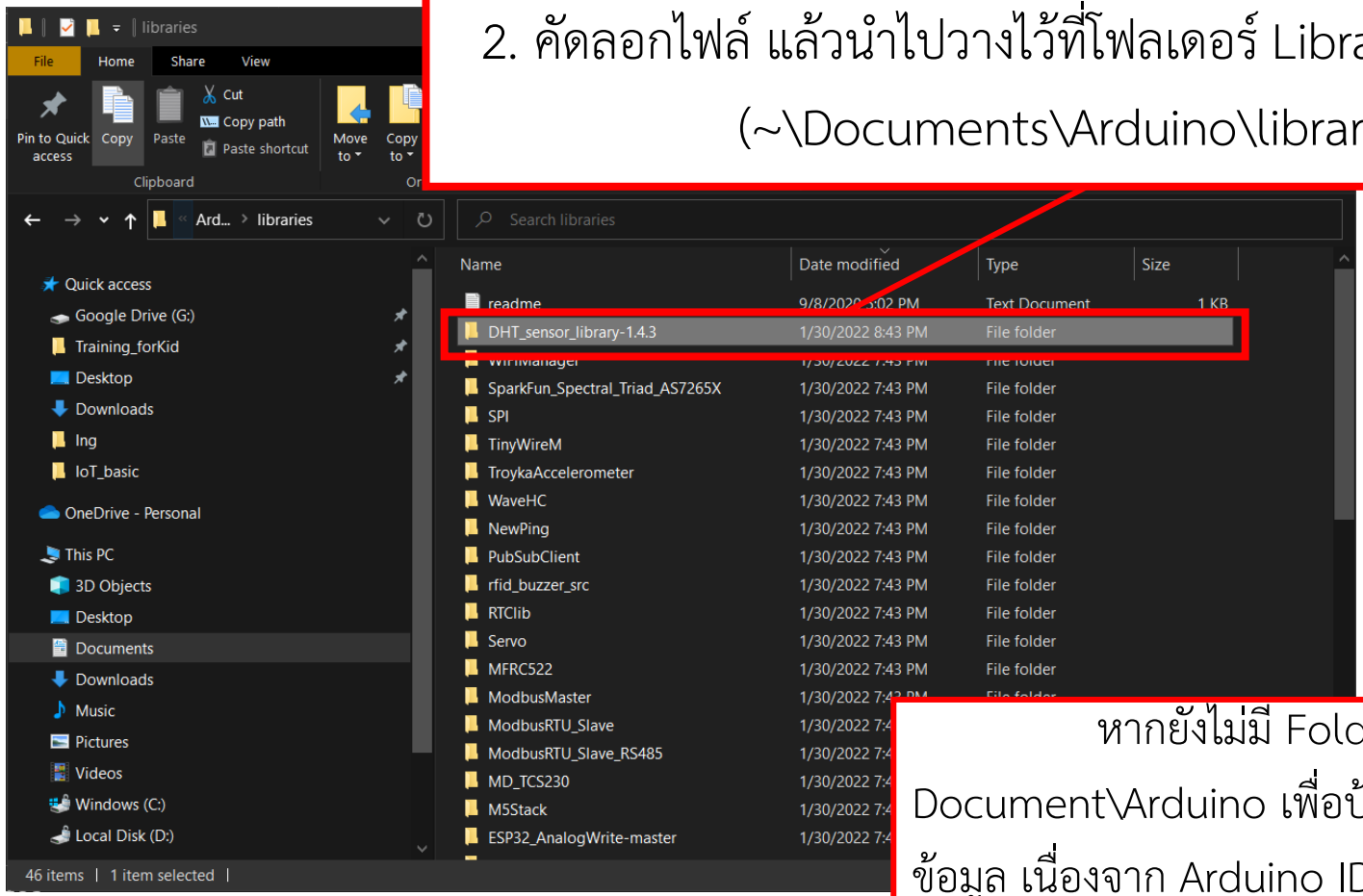
การทดลองที่ 3 การติดตั้ง ผ่าน Manual Installation

หากต้องการเพิ่มไลบรารีด้วยตนเอง ให้ดาวน์โหลดไฟล์ ZIP และแตกไฟล์ หลังจากนั้นนำไปใส่ไดเร็กทอรีของ Arduino IDE โดยในที่นี้ ให้ผู้เรียนทำการดาวน์โหลด Library ที่ link : https://bit.ly/3ueTYS4 **เพื่อใช้ในการเรียนต่อไป**



ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 3 การติดตั้ง ผ่าน Manual Installation



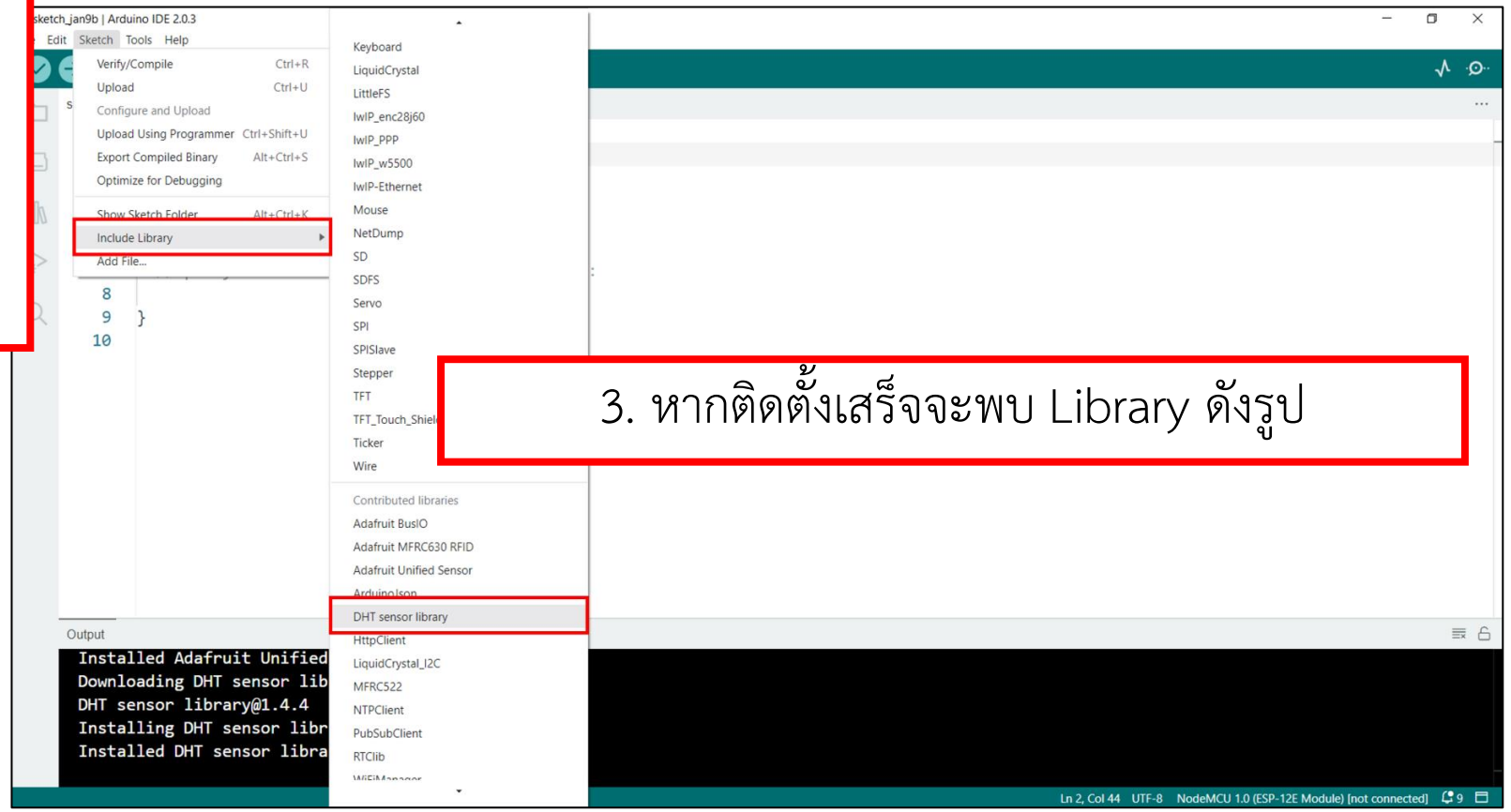
2. คัดลอกไฟล์ แล้วนำไปวางไว้ที่โฟลเดอร์ Library ของ Arduino (~\Documents\Arduino\libraries)

หากยังไม่มี Folder Library แนะนำให้สร้างไว้ที่ Document\Arduino เพื่อป้องกันการสูญหายของ Library และข้อมูล เนื่องจาก Arduino IDE มีการ Update Version อยู่ตลอด

ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 3 การติดตั้ง ผ่าน Manual Installation

เปิดโปแกรม Arduino (IDE) ไปที่ Sketch> Include Library > ตรวจสอบว่าไลบรารีที่เราเพิ่งเพิ่มเข้าไปมีอยู่ในรายการหรือไม่



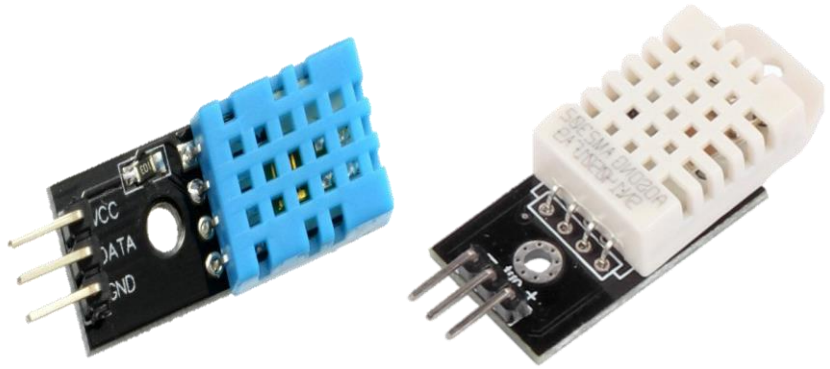
ใบงานที่ 2.1 ทฤษฎีเบื้องต้น

การใช้งาน Library

อุปกรณ์

DHT22 Sensor Module

เป็นเซนเซอร์วัดความชื้นและอุณหภูมิ ซึ่งการอ่านค่าจะเป็นสัญญาณแอนะล็อก ซึ่งเซนเซอร์จะแปลงเป็นสัญญาณดิจิทัล ดังนั้นจึงมีการนำ Library เข้ามาเพื่อทำให้ใช้งานได้มากขึ้น



DHT11 และ DHT22

คุณสมบัติทางเทคนิคที่สำคัญของ DHT11

1. ใช้แรงดันไฟเลี้ยง +3V ถึง +5.5V และกระแสไฟฟ้า 2.5 mA
2. วัดความชื้นสัมพัทธ์ได้ 20 ถึง 80%RH ความผิดพลาด ±5%RH และมีความละเอียดในการวัด 1% ขนาดของข้อมูล 8 บิต
3. วัดอุณหภูมิได้ 0 – 50 ℃ ความผิดพลาด ±2 ℃ ความละเอียดในการวัด 1 ℃ ขนาดของข้อมูล 8 บิต
4. อัตราการสุ่มวัด 1 ครั้งต่อวินาที ความเร็วในการตอบสนองต่อการเปลี่ยนแปลงในการวัด 6 – 30 วินาที

ใบงานที่ 2.1 ทฤษฎีเบื้องต้น

Function และคำสั่งใน Library DHT22

ส่วนของการประกาศต่างๆ

DHT.h

#include <DHT.h> เป็นการเรียกใช้ไลบรารี่ DHT ซึ่งเป็นไลบรารี่ที่สามารถเรียกอ่านค่าอุณหภูมิ และความชื้นจากโมดูล DHT11 และ DHT22

DHT NAME(pin, DHTTYPE)

DHT NAME(pin, DHTYPE) เป็นการกำหนดชื่อฟังก์ชันที่จะเรียกใช้คำสั่งชุด DHT และกำหนดขา และประเภทของ DHT ตัวอย่างเช่น DHT dht(D4,DHT11);

ส่วนของเริ่มต้นการใช้งาน (Setup)

NAME.begin()

เป็นคำสั่งเริ่มต้นเรียกใช้งาน Library DHT โดยปกติมักนำไปไว้ใน setup ตัวอย่างเช่น dht.begin();

ส่วนของของการนำไปใช้งาน

NAME.readHumidity()

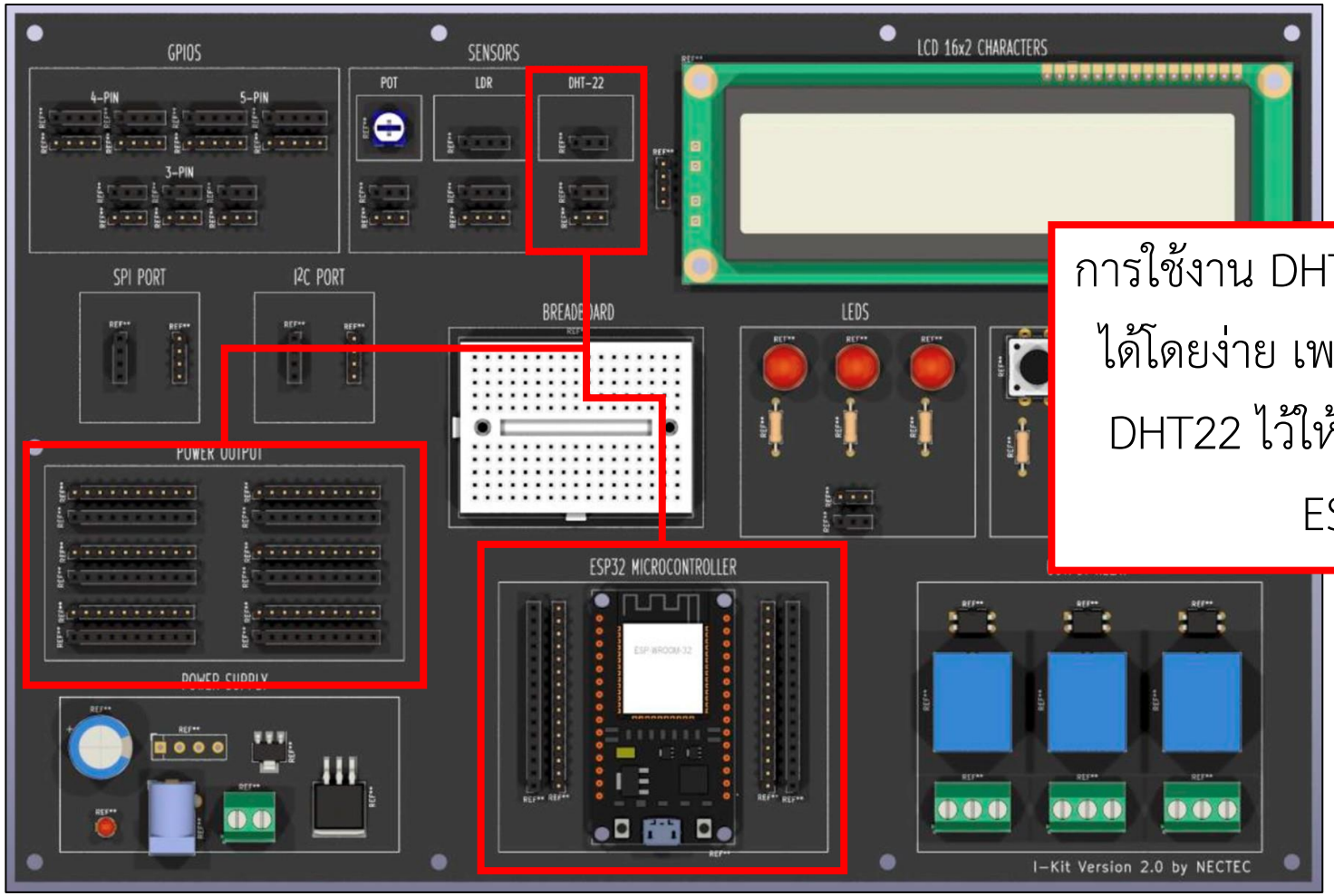
เป็นคำสั่งเรียกอ่านค่าความชื้นจากโมดูล DHT โดยหน่วยที่ได้จากการอ่านค่าความชื้นคือ %RH ตัวอย่างการเรียกใช้ เช่น dht.readHumidity();

NAME.readTemperature()

เป็นคำสั่งเรียกอ่านค่าอุณหภูมิจากโมดูล DHT โดยหน่วยที่ได้จากการอ่านค่าความชื้นคือ °C ตัวอย่างการเรียกใช้ เช่น dht.readTemperature();

ใบงานที่ 2.1 ทฤษฎีเบื้องต้น

การใช้งาน DHT22



การใช้งาน DHT22 บน I-KIT V.2 นั้น สามารถใช้งานได้โดยง่าย เพราะในช่อง Sensors มีพอร์ทสำหรับ DHT22 ไว้ให้แล้ว เพียงต่อสายเข้ากับ GPIO ของ ESP32 และต่อ Vcc, GND

ใบงานที่ 2.1 ขั้นตอนการทดลอง

การทดลองที่ 4 การอ่านค่าอุณหภูมิ และความชื้นของเซนเซอร์ DHT22 ผ่าน DHT Sensor Library

```
#include <DHT.h>

#define DHTPIN 34
#define DHTTYPE DHT22

DHT dht(DHTPIN, DHTTYPE);

void setup()
{
  Serial.begin(9600);
  dht.begin();
}

void loop()
{
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  float f = dht.readTemperature(true);

  if (isnan(h) || isnan(t) || isnan(f)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }

  Serial.print("DHT11 Humidity: ");
  Serial.print(h);
  Serial.print(" %\t");
  Serial.print("DHT11 Temperature: ");
  Serial.print(t);
  Serial.print(" °C ");
  Serial.print(f);
  Serial.println(" °F\t\n");
  delay(1000);
}
```

ใบงานที่ 2.1 คำถามท้ายหน่วยการเรียนรู้

คำถามท้ายหน่วยการเรียนรู้ที่ 1

จงเขียนโปรแกรมควบคุมอุปกรณ์ต่างๆที่เชื่อมต่อกับ ESP32 โดยมีลักษณะการทำงานดังนี้

LED1 จะติดเมื่อมีค่าของแสงสว่างจากเซนเซอร์ LDR มากกว่าเท่ากับ 150 ขึ้นไป

LED1 จะดับเมื่อมีค่าของแสงสว่างจากเซนเซอร์ LDR น้อยกว่า 150 ลงไป

LED บน ESP32 จะกระพริบเมื่ออุณหภูมิมากกว่า 25 องศาเซสเซียส

LED บน ESP32 จะดับเมื่ออุณหภูมิต่ำกว่า 25 องศาเซสเซียส

คำถามท้ายหน่วยการเรียนรู้ที่ 2

จงเขียนโปรแกรมควบคุมอุปกรณ์ต่างๆที่เชื่อมต่อกับ ESP32 โดยมี SW1 เป็นอุปกรณ์ควบคุมการทำงานคือ

เมื่อกด SW1 ครั้งที่ 1 ให้แสดงค่าแสงสว่างจาก LDR บน Serial Monitor โดยมีข้อความคือ

“Light Value in the Room = xxx lux”

เมื่อกด SW1 ครั้งที่ 2 ให้แสดงค่าความชื้นจาก DHT บน Serial Monitor โดยมีข้อความคือ

“Humidity Value in the Room = xxx %H”

เมื่อกด SW1 ครั้งที่ 3 ให้แสดงค่าความชื้นจาก DHT บน Serial Monitor โดยมีข้อความคือ

“Temperature Value in the Room = xxx °C”

เมื่อกด SW1 ครั้งที่ 4 ให้แสดงค่าความชื้นจาก DHT บน Serial Monitor โดยมีข้อความคือ

“Temperature Value in the Room = xxx °F”

เมื่อกด SW1 ครั้งที่ 5 ให้เริ่มกลับไปทำงานเหมือนครั้งที่ 1

LED จะติดเมื่อมีค่าของแสงสว่างจากเซนเซอร์ LDR มากกว่าเท่ากับ 150 ขึ้นไป, LED จะดับเมื่อมีค่าของแสงสว่างจากเซนเซอร์ LDR น้อยกว่า 150 ลงไป

LED บน ESP32 จะกระพริบเมื่ออุณหภูมิมากกว่า 25 องศาเซสเซียส, LED บน ESP32 จะดับเมื่ออุณหภูมิต่ำกว่า 25 องศาเซสเซียส

ใบงานที่ 2.1 คำถามท้ายหน่วยการเรียนรู้

คำถามท้ายหน่วยการเรียนรู้ที่ 3

จงเขียนโปรแกรมควบคุมอุปกรณ์ต่างๆที่เชื่อมต่อกับ ESP32 โดยมี SW1 เป็นอุปกรณ์ควบคุมการทำงานคือ

ทุกครั้งที่มีการกด SW1 LED จะต้องทำการติดค้างไว้เสมอจะดับก็ต่อเมื่อมีการปล่อย

เมื่อกด SW1 ครั้งที่ 1 ให้แสดงค่าแสงสว่างจาก LDR บน Serial Monitor โดยมีข้อความคือ

"Light Value in the Room = xxx lux"

เมื่อกด SW1 ครั้งที่ 2 ให้แสดงค่าความชื้นจาก DHT บน Serial Monitor โดยมีข้อความคือ

"Humidity Value in the Room = xxx %H"

เมื่อกด SW1 ครั้งที่ 3 ให้แสดงค่าความชื้นจาก DHT บน Serial Monitor โดยมีข้อความคือ

"Temperature Value in the Room = xxx °C"

เมื่อกด SW1 ครั้งที่ 4 ให้แสดงค่าความชื้นจาก DHT บน Serial Monitor โดยมีข้อความคือ

"Temperature Value in the Room = xxx °F"

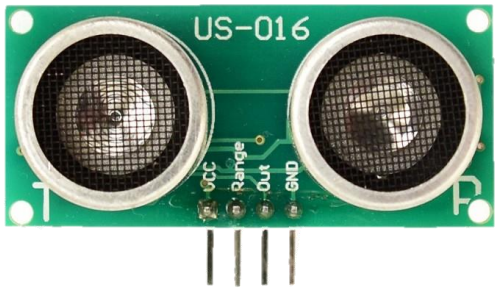
เมื่อกด SW1 ครั้งที่ 7 ให้เริ่มกลับไปทำงานเหมือนครั้งที่ 1

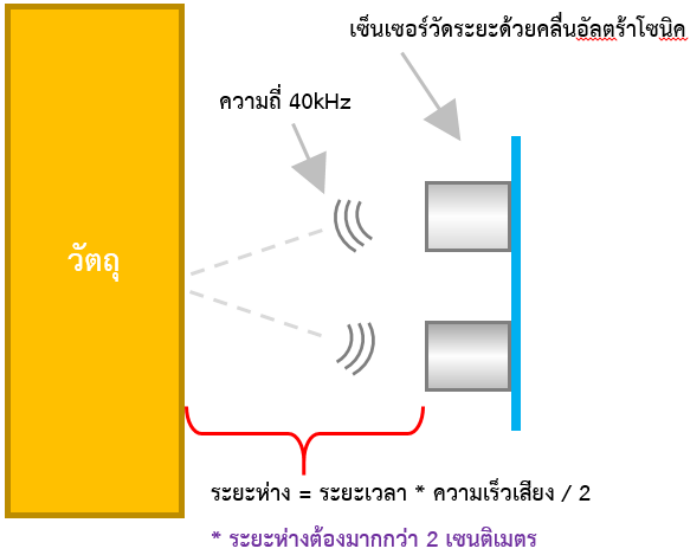
ใบงานที่ 2.1 ทฤษฎีเบื้องต้น (เสริม)

อุปกรณ์

Ultrasonic sensor module

โมดูล Ultrasonic ตรวจจับวัตถุ คำนวนระยะทางโดยใช้คลื่น มีลักษณะเป็นกรวยและไม่ใช่เส้นตรง จึงเหมาะสำหรับใช้ตรวจจับสิ่งกีดขวางด้วย





คุณสมบัติทางเทคนิคที่สำคัญของ Ultrasonic

1. ใช้แรงดันไฟเลี้ยง +5V และกระแสไฟฟ้า 3.8 mA
2. ความแม่นยำในการตรวจจับ 0.3 cm ความคลาดเคลื่อน 1%
3. ทำงานในอุณหภูมิ 0 ถึง 70 ℃ ความละเอียดในการวัด
4. ระยะการวัด 2 – 300 cm, ความละเอียด 1 mm และระยะการตรวจจับ 15 องศา

ใบงานที่ 2.1 ทฤษฎีเบื้องต้น (เสริม)

Ultrasonic sensor module

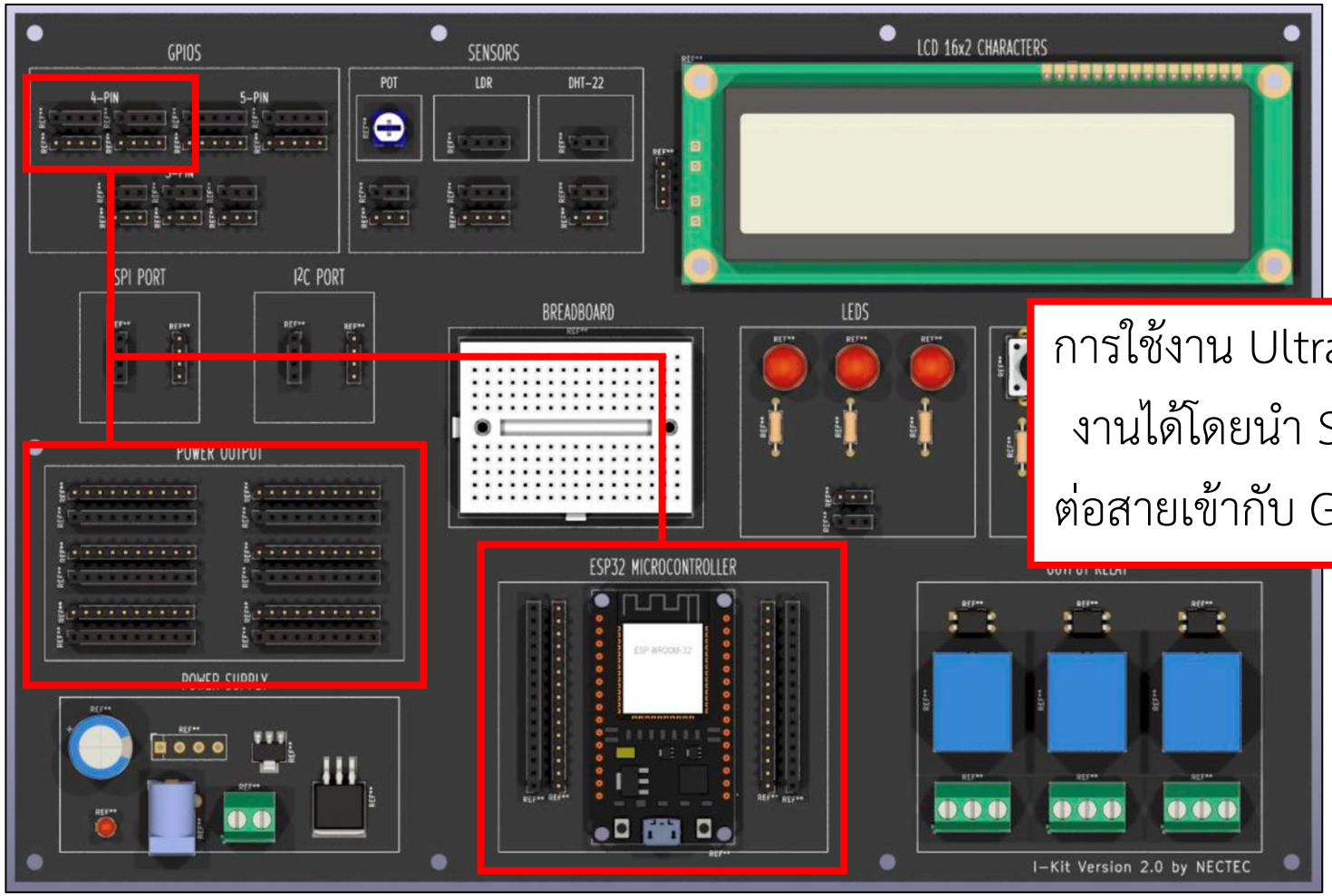
ในชุด I-Kit เราได้ใช้ Ultrasonic sensor รุ่น US-016 ซึ่งส่งค่าเป็น Analog อยู่แล้ว จึงไม่ต้องทำการติดตั้ง Library เพิ่มเติม

คุณสมบัติ	รุ่น HC-SR04	รุ่น HY-SRF05	รุ่น US-100	รุ่น US-016
แรงดันไฟฟ้า	5V		3 - 5V	5V
ใช้กระแสไฟฟ้า	<2mA	30mA ?	<2mA	3.8mA
ความถี่	40kHz			
ระยะในการวัด	2 cm - 4.5m	3 cm - 3 m	2 cm - 4.5m	2cm - 3m
การสื่อสาร	การทริกสัญญาณ		การทริกสัญญาณ / UART	สัญญาณอนาล็อก
องศาในการวัด	<15°	???		<15°

กรณีใช้ Ultrasonic sensor รุ่นอื่นๆ แนะนำให้ติดตั้ง Library เสริม เนื่องจากการสื่อสารเป็นแบบทริกสัญญาณ โดยแนะนำ Library ที่ชื่อ Newping เนื่องจากใช้งานง่าย

ใบงานที่ 2.1 ทฤษฎีเบื้องต้น (เสริม)

การใช้งาน Ultrasonic Sensor



การใช้งาน Ultrasonic บน I-KIT V.2 นั้น สามารถใช้งานได้โดยนำ Sensor ไปเสียบที่ GPIO 4 Pin แล้วต่อสายเข้ากับ GPIO ของ ESP32 และต่อ Vcc, GND

ใบงานที่ 2.1 ขั้นตอนการทดลอง (เสริม)

การทดลองเสริม

```
int ultra_out = 13;
int ultra_range = 12;

void setup() {
  Serial.begin(9600);
  pinMode(ultra_range, OUTPUT);
  digitalWrite(ultra_range, LOW);
  pinMode(ultra_out, INPUT);
}
void loop() {
  unsigned int val_ultra;
  val_ultra = analogRead(ultra_out);
  long cm;
  cm = microsecondsToCentimeters(long(val_ultra));
  Serial.print("Ultrasonic Distance : ");
  Serial.print(cm);
  Serial.println(" cm");
  delay(1000);
}

long microsecondsToCentimeters(long microseconds)
{
  return microseconds / 29 / 2;
}
```

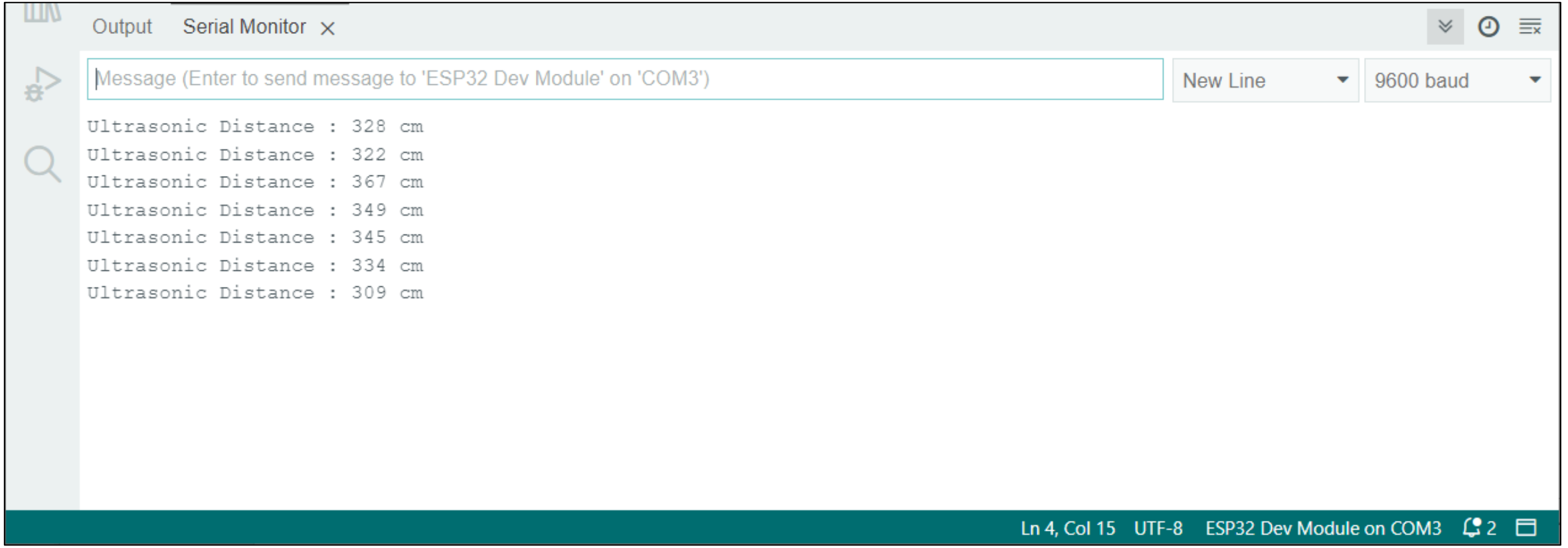
ทดสอบการใช้งานโมดูล Ultrasonic (US-016) วัดระยะ

ใบงานที่ 2.1 ขั้นตอนการทดลอง (เสริม)

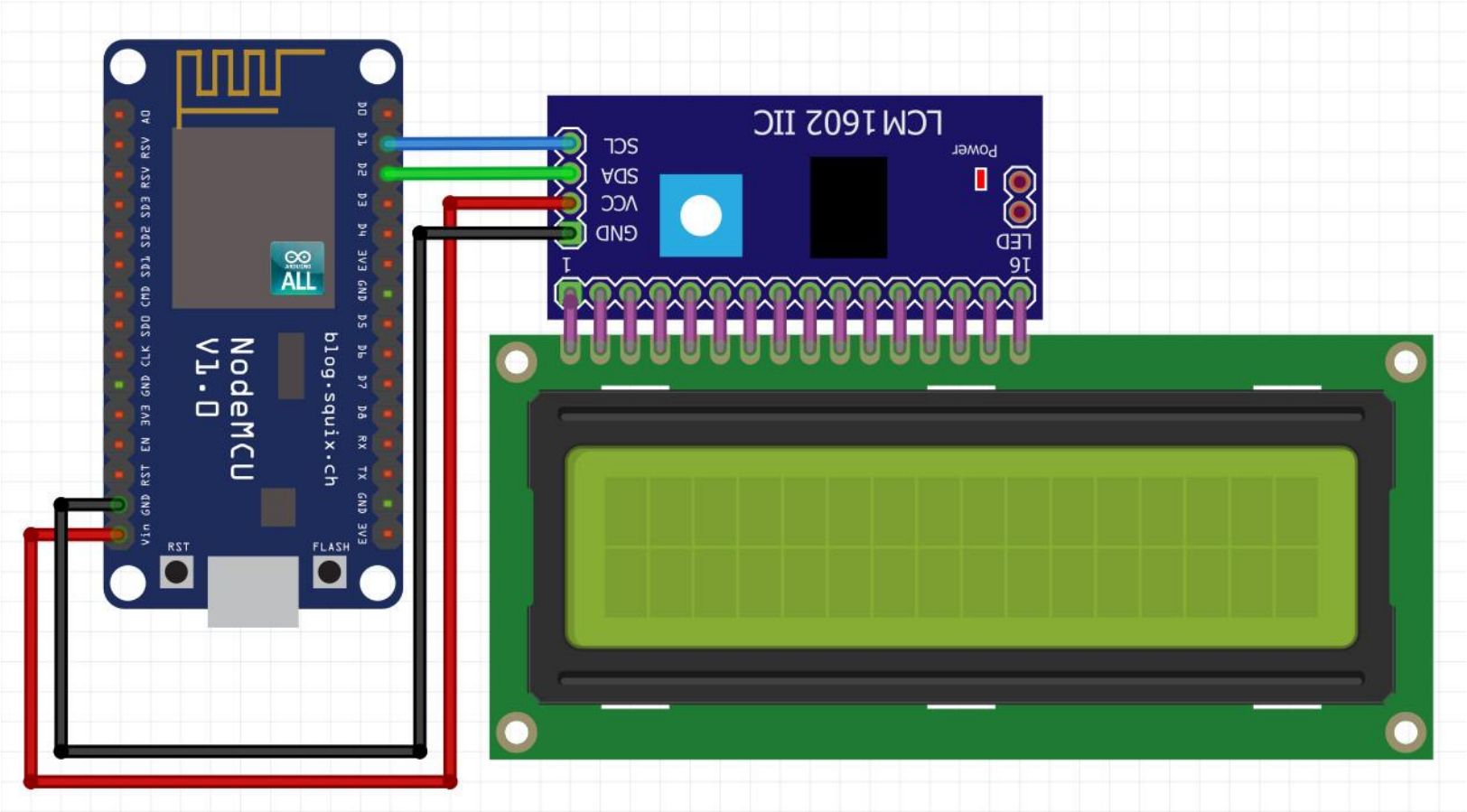
การทดลองเสริม

ทดสอบการใช้งานโมดูล Ultrasonic (US-016) วัดระยะ

ผลลัพธ์ของโปรแกรม



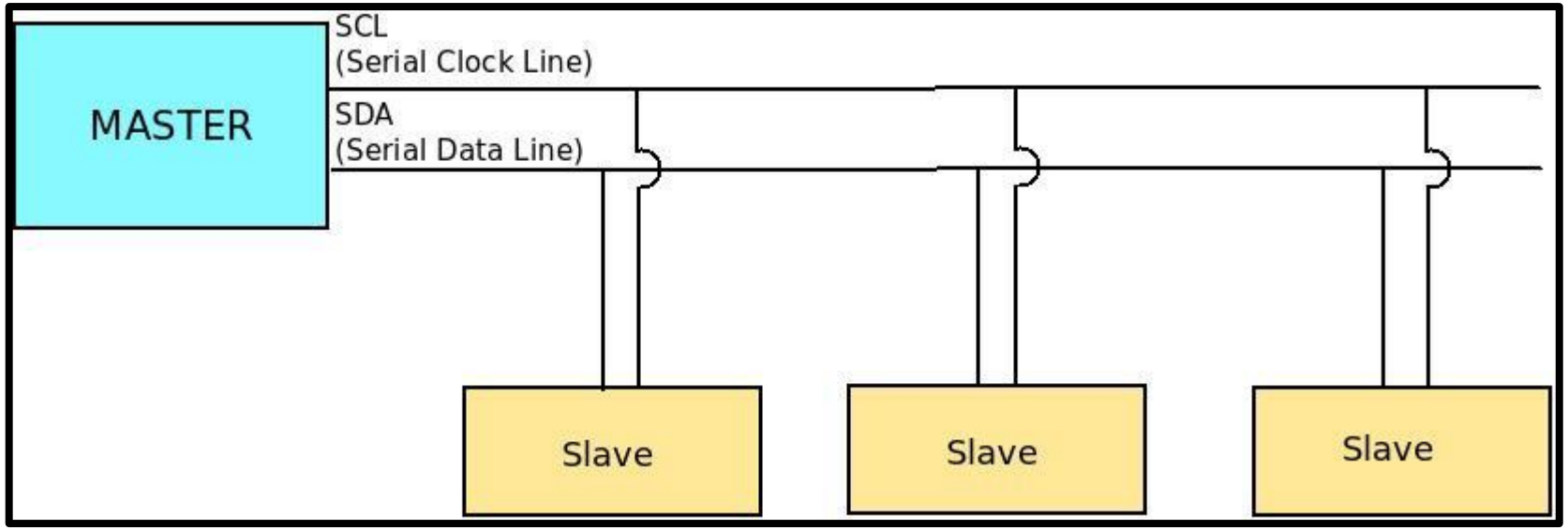
ใบงานที่ 2.2 การสื่อสารและใช้งาน I2C ผ่านหน้าจอแสดงผล LCD



ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

การสื่อสารรูปแบบ I²C

I²C ย่อมาจาก Inter Integrate Circuit เป็นการสื่อสารอนุกรมแบบ Synchronous เพื่อใช้สื่อสารข้อมูลระหว่างอุปกรณ์ความเร็วต่ำ โดย I²C มีความโดดเด่นคือ การเชื่อมต่อกันเป็นระบบบัสแบบรับและส่งข้อมูล ซึ่งสามารถเชื่อมต่อกับอุปกรณ์จำนวนมากโดยใช้สายเพียงแค่ 2 เส้น ทำให้ลดจำนวนของสายไฟ และขนาดของอุปกรณ์ไปได้ จึงทำให้เป็นการสื่อสารที่นิยมสำหรับระบบฝังตัว ซึ่งสายเพียง 2 เส้นที่พูดถึงคือ SCL (ส่งสัญญาณนาฬิกา เพื่อบอกว่าข้อมูลที่จะรับ/ส่ง จะทำที่ความถี่เท่าใด) และ SDA (ใช้ส่งสัญญาณเพียงอย่างเดียว)



ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

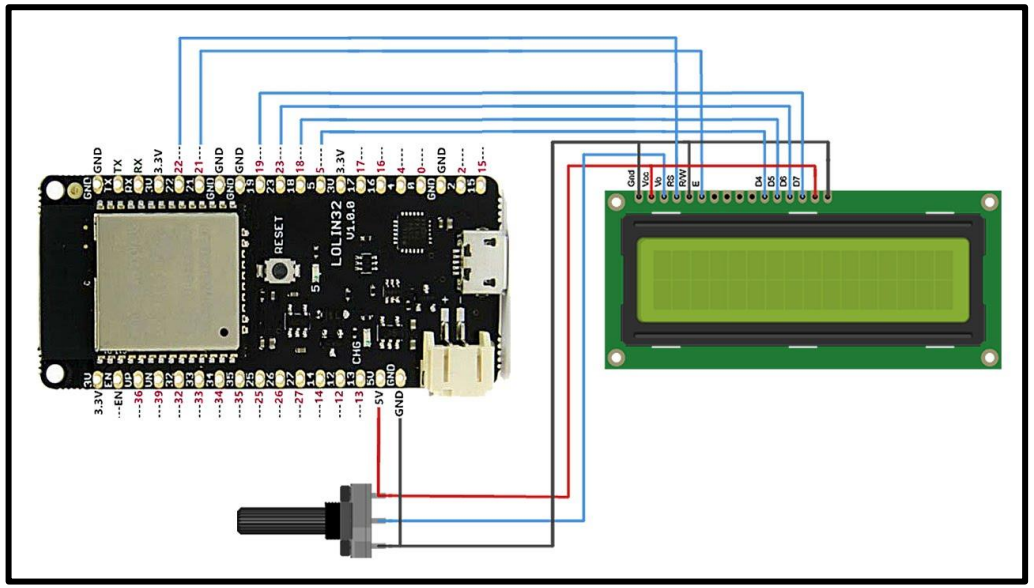
Character LCD Display

lcd ย่อมาจาก Liquid Crystal Display ซึ่งเป็นจอแสดงผล โดยภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพผ่านชั้นกรองแสง แล้ววิ่งไปยังคริสตัลเหลวที่เรียงตัวกัน โดย LCD โมดูล จะมี Address สำหรับใช้ในการเขียน Code ทุกครั้งที่เราทำการใช้งาน ต้องเช็ค Address ของ LCD ก่อนเสมอ

LCD ขนาด 16x2

คุณสมบัติทางเทคนิคที่สำคัญของ LCD

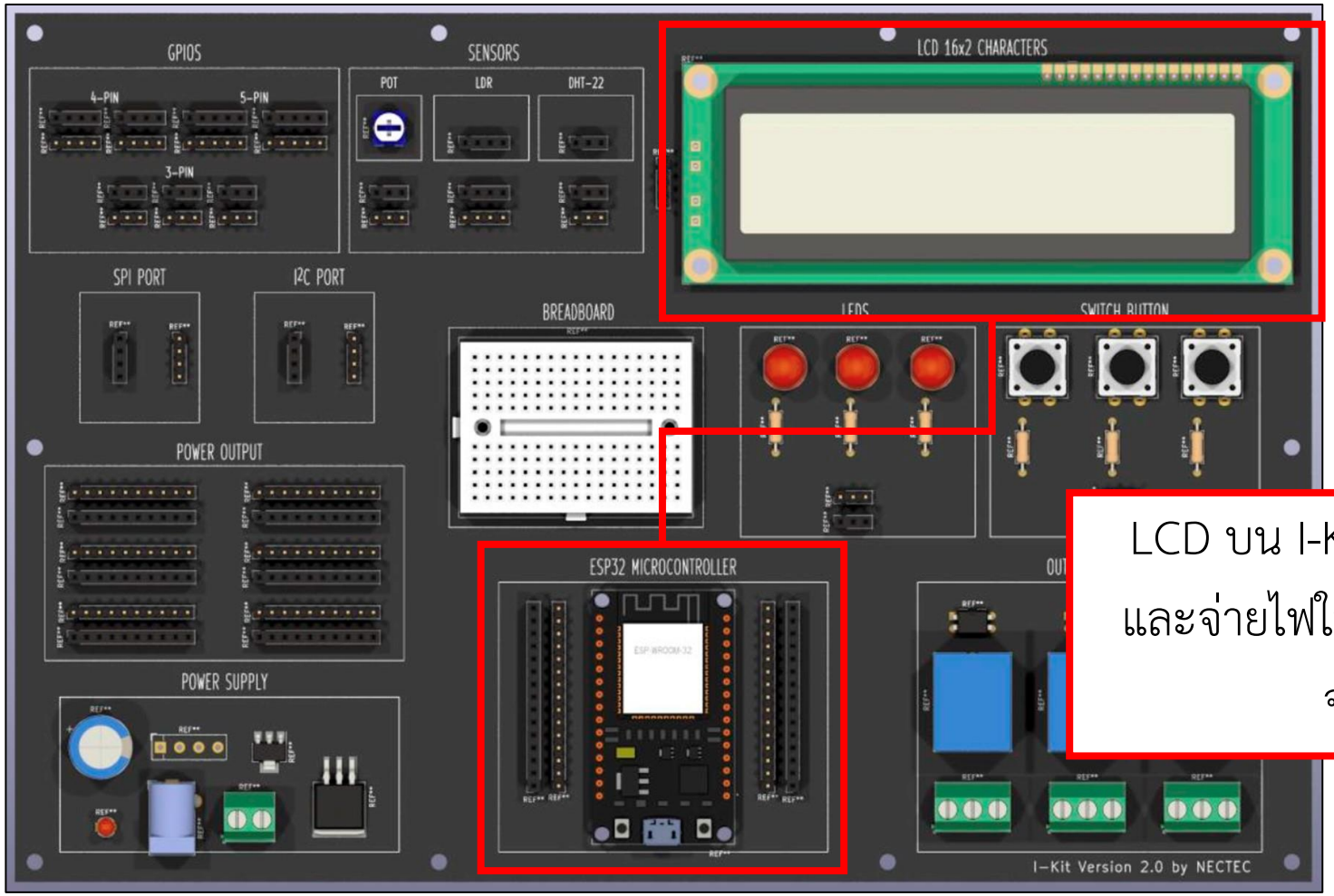
1. ใช้แรงดันไฟเลี้ยง +5V
2. Blacklight และ Contrast สามารถปรับค่าได้ด้วยตัวต้านทานปรับค่าได้
3. มีแบบขนาด 16 ตัวอักษร 2 บรรทัด และ 20 ตัวอักษร 4 บรรทัด
4. สามารถสื่อสารผ่าน I²C ได้ด้วยการใช้โมดูล I²C



ภาพการต่อ LCD กับ ESP32 หากไม่ใช้โมดูล I²C

ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

การใช้งาน LCD



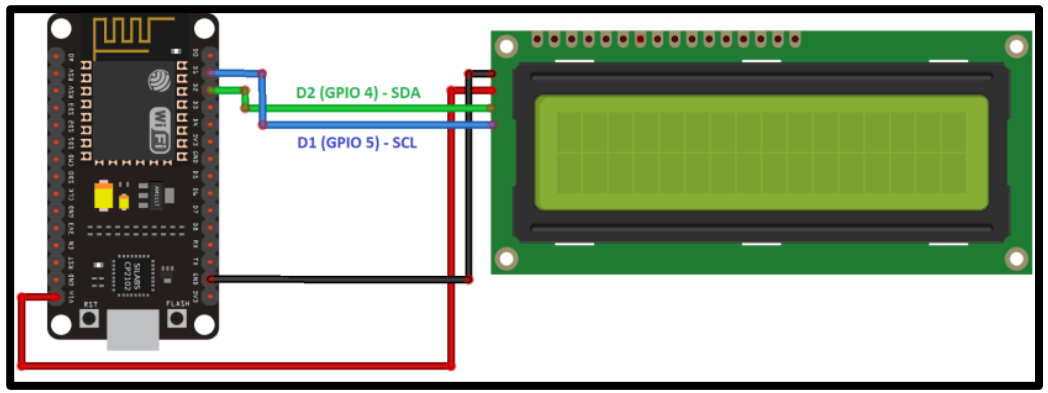
LCD บน I-KIT V.2 นั้นถูกเชื่อมเข้ากับSDA, SCL และจ่ายไฟให้เรียบร้อยแล้ว สามารถใช้งานโดยไม่จำเป็นต้องต่อสายเพิ่มเติม

ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

Library ที่ใช้สำหรับ LCD

1. Wire.h

Wire.h ใช้สำหรับเป็นการประกาศให้ NodeMCU และ ESP32 DevKit สามารถสื่อสารกับอุปกรณ์ต่างๆผ่านการสื่อสาร I²C โดยลักษณะการประกาศใช้งานคือ #include <Wire.h>



2. LiquidCrystal_I2C

เป็น library ของจอ LCD โดยมีฟังก์ชันต่างๆที่สำคัญสำหรับการใช้งาน LCD โดยการประกาศใช้งานคือ #include <LiquidCrystal_I2C.h>

ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

คำสั่งต่างๆใน library LiquidCrystal_I2C

lcd.init(จำนวนตัวอักษรที่ใส่ได้ในแถว,จำนวนแถวทั้งหมด)

> ใช้กำหนดลักษณะหน้าจอ เช่น ถ้าจอเรามี 2แถว 16ตัวอักษร ให้ใส่ lcd.init(16,2);

Lcd.backlight()

> ใช้เมื่อต้องการ เปิดไฟแบล็กไลค์

lcd.clear()

> ใช้ล้างหน้าจอ เมื่อมีตัวอักษรใดๆอยู่บนหน้าจอ จะถูกล้างออกทั้งหมด

lcd.home()

> ใช้ปรับให้เคเซอร์กลับไปอยู่ที่ตำแหน่งแรกด้านซ้าย เมื่อใช้คำสั่ง lcd.print() จะไปเริ่มแสดงผลทางด้านบนซ้าย

lcd.setCursor(ลำดับตัวอักษรนับจากทางซ้าย, บรรทัด)

> ใช้ตั้งค่าเคเซอร์ เช่น lcd.setCursor(2, 0); หมายถึงเซ็ตเคเซอร์ไปตัวอักษรที่ 2 นับจากทางซ้าย และอยู่บรรทัดแรก เมื่อใช้คำสั่ง lcd.print() ตัวอักษรตัวแรกจะอยู่ลำดับที่ 3 นับจากทางซ้าย

lcd.write(ข้อมูลที่ต้องการเขียนออกไป)

> ใช้สำหรับเขียนข้อมูลออกไปทีละตัวอักษร

lcd.print(ข้อมูลที่ต้องการให้เขียนออกไป [, รูปแบบข้อมูล])

> ใช้เขียนข้อมูลออกไปทั้งข้อความ

lcd.cursor()

> ใช้สั่งให้แสดงเคเซอร์บนหน้าจอ

ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

คำสั่งต่างๆใน library LiquidCrystal_I2C

lcd.noCursor()

> ใช้สั่งให้ไม่แสดงเคเซอร์บนหน้าจอ

lcd.display()

> แสดงตัวอักษรบนหน้าจอ

lcd.noDisplay()

> ปิดการแสดงตัวอักษรในหน้าจอ

lcd.scrollDisplayLeft()

> เลื่อนตัวอักษรไปทางซ้าย 1 ตัว

lcd.scrollDisplayRight()

> เลื่อนตัวอักษรไปทางขวา 1 ตัว

lcd.autoscroll()

> เลื่อนตัวอักษรไปทางขวาอัตโนมัติหากใช้คำสั่ง lcd.print() หรือ lcd.write() เมื่อตัวอักษรเต็มหน้าจอ

lcd.noAutoscroll()

> ปิดการเลื่อนตัวอักษรอัตโนมัติ

lcd.leftToRight()

> เมื่อใช้คำสั่ง lcd.print() หรือ lcd.write() ตัวอักษรจะเขียนจากซ้ายไปขวา

lcd.rightToLeft()

> เมื่อใช้คำสั่ง lcd.print() หรือ lcd.write() ตัวอักษรจะเขียนจากขวาไปซ้าย

ใบงานที่ 2.2 ทฤษฎีเบื้องต้น

Function และคำสั่งใน Library LCD

ส่วนของการประกาศต่างๆ

LiquidCrystal_I2C NAME(Address, columns, rows)

เป็นการกำหนดชื่อฟังก์ชันที่จะเรียกใช้คำสั่งชุด lcd และกำหนดขา และประเภทของ lcd ตัวอย่างเช่น LiquidCrystal_I2C lcd(0x27, 16, 2)

LCD Address โดยทั่วไปมีอยู่ 2 Address โดย address ขึ้นอยู่กับ IC (integrated circuit) ที่แปลงสัญญาณ I2C ได้แก่

1. PCF8574 ตำแหน่งเริ่มต้นคือ 0x27
2. PCF8574A ตำแหน่งเริ่มต้นคือ 0x3F

ส่วนของเริ่มต้นการใช้งาน (Setup)

NAME.init()

เป็นคำสั่งเริ่มต้นเรียกใช้งาน Library LCD โดยปกติมักนำไปไว้ใน setup ตัวอย่างเช่น lcd.init();

NAME.backlight()

เป็นคำสั่งเริ่มต้นใช้เมื่อต้องการ เปิดไฟแบล็กไลค์ มักนิยมไปไว้ที่ setup ตัวอย่างเช่น lcd.backlight();

ส่วนของของการนำไปใช้งาน

NAME.setCursor()

ในการกำหนดตำแหน่งของ Cursor เช่น lcd.setCursor(0,0) คือให้เริ่มต้นตัวอักษรที่ตำแหน่ง 0 บรรทัดที่ 1 ของจอ LCD

NAME.print()

ใช้ในการกำหนดข้อความที่ต้องการแสดง นิยมใช้หลังจากคำสั่ง setCursor โดยตัวอย่างการใช้งานคือ lcd.print(“I-Kit”)

NAME.clear()

เป็นคำสั่งที่ใช้สำหรับเคลียร์หน้าจอทั้งหมด

ใบงานที่ 2.2 ขั้นตอนการทดลอง

การทดลองที่ 1 การแสดงผลข้อความบนหน้าจอแสดงผล LCD

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 16, 2);

void setup()
{
    lcd.init();
    lcd.backlight();
}

void loop()
{
    lcd.setCursor(0, 0);
    lcd.print("   LCD 16 x 2   ");
    lcd.setCursor(0, 1);
    lcd.print("  I-Kit  V.2.0 ");
}
```

การทดลองเสริม

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

#define ldr 36

LiquidCrystal_I2C lcd(0x27, 16, 2);

float ADC_value = 0.0048828125;

void setup()
{
  Serial.begin(9600);
  pinMode(ldr, INPUT);
  lcd.init();
  lcd.backlight();
  lcd.clear();
}
void loop()
{
  float ldr_data = analogRead(ldr);
  int lux = int((250.000000/(ADC_value*ldr_data))-
50.000000);
  lcd.setCursor(0, 0);
  lcd.print("Data on I-Kit");
  lcd.setCursor(0, 1);
  lcd.print("light = ");
  lcd.setCursor(8, 1);
  lcd.print(lux);
  lcd.setCursor(11, 1);
  lcd.print(" lux");
  delay(1000);
}
```

ทดสอบการใช้งาน LCD ร่วมกับ LDR

ใบงานที่ 2.2 ขั้นตอนการทดลอง

การทดลองที่ 2 การแสดงผลค่าจากเซนเซอร์ LDR และ DHT22 บนหน้าจอแสดงผล LCD

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

#define DHTPIN 34
#define DHTTYPE DHT22
#define ldr 36

LiquidCrystal_I2C lcd(0x27, 16, 2);
DHT dht(DHTPIN, DHTTYPE);

float ADC_value = 0.0048828125;

void setup()
{
  Serial.begin(9600);
  pinMode(ldr, INPUT);
  lcd.init();
  lcd.backlight();
  dht.begin();
  lcd.clear();
}

void loop()
{
  float h = dht.readHumidity();
  int t = dht.readTemperature();
  float ldr_data = analogRead(ldr);
  int lux = int((250.000000/(ADC_value*ldr_data))-50.000000);
  lcd.setCursor(0, 0);
  lcd.print("humi=");
  lcd.setCursor(5, 0);
  lcd.print(h);
  lcd.setCursor(7, 0);
  lcd.print(",temp=");
  lcd.setCursor(13, 0);
  lcd.print(t);
  lcd.setCursor(0, 1);
  lcd.print("light = ");
  lcd.setCursor(8, 1);
  lcd.print(lux);
  lcd.setCursor(11, 1);
  lcd.print(" lux");
  delay(1000);
}
```

ทดสอบการใช้งาน LCD ร่วมกับ LDR และ DHT

ใบงานที่ 2.2 คำถามท้ายหน่วยการเรียนรู้

คำถามท้ายหน่วยการเรียนรู้ที่ 4

จงเขียนโปรแกรมควบคุมอุปกรณ์ต่างๆที่เชื่อมต่อกับ ESP32 โดยมี SW1 เป็นอุปกรณ์ควบคุมการทำงานคือ

ทุกครั้งที่มีการกด SW1 LED จะต้องทำการติดค้างไว้เสมอจะดับก็ต่อเมื่อมีการปล่อย

เมื่อกด SW1 ครั้งที่ 1 ให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Temperature=xx°C"

เมื่อกด SW1 ครั้งที่ 2 ให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Humidity = xx %H"

เมื่อกด SW1 ครั้งที่ 3 ให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Temperature=xx°F"

เมื่อกด SW1 ครั้งที่ 4 ให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Light = xxx lux"

เมื่อกด SW1 ครั้งที่ 5 ให้เริ่มกลับไปทำงานเหมือนครั้งที่ 1

โดยบรรทัดที่ 1 ของโปรแกรมคือ "Train by I-Kit"

ใบงานที่ 2.2 คำถามท้ายหน่วยการเรียนรู้

คำถามท้ายหน่วยการเรียนรู้ที่ 5

จงเขียนโปรแกรมควบคุมอุปกรณ์ต่างๆที่เชื่อมต่อกับ ESP32 โดยมี เวลา เป็นสิ่งควบคุมการทำงานโดยมีการทำงานดังนี้

10 วินาทีแรกให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

“Temperature=xx°C”

10 วินาทีถัดไปให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

“Humidity = xx %H”

10 วินาทีถัดไปให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

“Temperature=xx°F”

10 วินาทีถัดไปให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

“Light = xxx lux”

10 วินาทีถัดไปให้เริ่มกลับไปทำงานเหมือน 10 วินาทีแรก

โดยบรรทัดที่ 1 ของโปรแกรมคือ “IoT Training”

ใบงานที่ 2.2 คำถามท้ายหน่วยการเรียนรู้

คำถามท้ายหน่วยการเรียนรู้ที่ 6

จงเขียนโปรแกรมควบคุมอุปกรณ์ต่างๆที่เชื่อมต่อกับ ESP32 โดยมี เวลา เป็นสิ่งควบคุมการทำงานโดยมีการทำงานดังนี้

10 วินาทีแรกให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Temperature=xx°C"

10 วินาทีถัดไปให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Humidity = xx %H"

10 วินาทีถัดไปให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Temperature=xx°F"

10 วินาทีถัดไปให้แสดงค่าบน LCD โดยมีข้อความที่บรรทัดที่ 2 คือ

"Light = xxx lux"

10 วินาทีถัดไปให้เริ่มกลับไปทำงานเหมือน 10 วินาทีแรก

โดยบรรทัดที่ 1 ของโปรแกรมคือ "IoT Training"

โดยโปรแกรมมีเงื่อนไขเพิ่มเติมดังนี้

- ทุกครั้งที่มีการเข้าเงื่อนไขให้ LED1 ติดค้างไว้เสมอ แต่ถ้าไม่มีการเข้าเงื่อนไขให้ LED1 ดับ
- เมื่ออุณหภูมิมากกว่า 25 °C ให้บรรทัดที่ 1 ของ LCD แสดงว่า "Temperature High"
- เมื่อความชื้นมากกว่า 80 %H ให้บรรทัดที่ 1 ของ LCD แสดงว่า "Humidity High"
- เมื่อแสงมากกว่า 500 lux ให้บรรทัดที่ 1 ของ LCD แสดงว่า "Light High"
- เมื่อมีพารามิเตอร์ที่เข้าเงื่อนไขมากกว่า 2 พารามิเตอร์ ให้บรรทัดที่ 1 แสดงว่า ทั้ง 2 พารามิเตอร์มีค่าสูง เช่น "Temp, Humi High"
- เมื่อมีทุกพารามิเตอร์ที่เข้าเงื่อนไขให้บรรทัดที่ 1 แสดงว่า "All Data High"