คู่มือการใช้งาน Farm1

- เอาต์พุต Contect 5A/NO 4 ช่อง
- อินพุตดิจิทัล 2 ช่อง
- อินพุตแอนะล็อก 2 ช่อง
- ต่ออุปกรณ์เสริมผ่าน RS485 และ I^2C
- ใช้แรงดันไฟฟ้าได้ 5V ถึง 24V
- ใช้ ESP32 เป็นตัวประมวลผล
- เชื่อมต่ออินเตอร์เน็ตผ่าน WiFi
- มีวงจรนับเวลา (RTC) ในตัว
- ขนาด 110 x 120 x 40 มิลลิเมตร

Farm1 กล่องควบคุมฟาร์มระบบ IoT ใช้ไมโครคอนโทรลเลอร์ ESP32 มาพร้อม WiFi และบลูทูธ ติดตั้งบนบอร์ด Farm1 รองรับการต่ออุปกรณ์ภายนอกผ่านช่องเอาต์พุต 5A/220V จำนวน 4 ช่อง ต่ออินพุตดิจิทัล 2 ช่อง อินพุตแอนะล็อก 2 ช่อง และ I/O จำนวน 1 ช่อง มีช่องต่อ I2C และ RS485 ใช้งานได้กับเซ็นเซอร์ทุกประเภท มีวงจรนับเวลา (RTC) มาในตัว มาพร้อมกล่องติดตั้งบนรางปีกนก

เวอร์ชั่น	รายละเอียดการแก้ไข	เผยแพร่
1.0	เพิ่มข้อมูลทั่วไปเกี่ยวกับ Farm1 และการเขียนโปรแกรมสั่งงานเบื้องต้น	18/11/2564
1.1	เพิ่มวิธีการใช้งานกับ HandySense	25/11/2564
1.2	เปลี่ยนเซ็นเซอร์หัวข้อ HandySense จาก SHT30 เป็น SHT20	8/12/2564
1.3	เพิ่มหัวข้อ ตั้งค่าเชื่อมต่อ WiFi ด้วยแอพฯ EspTouch	10/3/2565

สารบัญ

สร้างระบบควบคุมฟาร์ม IoT อย่างรวดเร็วด้วย HandySense 4

การต่อเซ็นเซอร์เข้ากับบอร์ด Farm1 4

การอัพเดทเฟิร์มแวร์ HandySense 5

การเพิ่มอุปกรณ์เข้าระบบ HandySense 7

ข้อมูลด้านฮาร์ดแวร์และวงจร 15

คุณสมบัติทางไฟฟ้า 15

ตำแหน่งขาต่อใช้งาน 16

การจ่ายไฟ 16

ช่องเอาต์พุต 18

ช่องอินพุตดิจิทัล 20

ช่องอินพุตแอนะล็อก 21

ช่อง I/O 22

ช่อง I^2C 24

ช่อง RS485 25

วงจรนับเวลา (RTC) 26

การเขียนโปรแกรมสั่งงานด้วย Arduino IDE 28

ติดตั้งโปรแกรม Arduino IDE 28

ติดตั้งไดร์เวอร์ FT231X 30

การเพิ่มบอร์ด ESP32 ลงโปรแกรม Arduino IDE 33

การเชื่อมต่อ Farm1 เข้ากับเครื่องคอมพิวเตอร์ 35

ทดสอบอัพโหลดโปรแกรมไฟกระพริบ 35

เขียนโปรแกรมสั่งงานช่องเอาต์พุต 38

เขียนโปรแกรมอ่านช่องอินพุตดิจิทัล 39

เขียนโปรแกรมอ่านช่องอินพุตแอนะล็อก 40

เขียนโปรแกรมช่อง I/O อ่านค่าอุณหภูมิจาก DS18B20 42

เขียนโปรแกรมช่อง I^2C อ่านค่าอุณหภูมิและความชื้นจาก SHT30 45

เขียนโปรแกรมช่อง RS485 อ่านค่าอุณหภูมิและความชื้นจาก XY-MD02 46

เขียนโปรแกรมตั้งเวลาและอ่านค่าเวลาจากวงจรนับเวลา (RTC) 47

คำแนะนำการติดตั้ง 47

แนวทางการแก้ไขปัญหา 48

สร้างระบบควบคุมฟาร์ม IoT อย่างรวดเร็วด้วย HandySense

HandySense เป็นแพลตฟอร์มระบบควบคุมฟาร์มประกอบด้วยฮาร์ดแวร์และซอฟแวร์ เปิดเป็น Open Source เปิดให้ผู้สนใจนำไปพัฒนาต่อได้ บอร์ด Farm1 สามารถใช้งานร่วมกับระบบ Cloud ของ HandySense ผ่านเฟิร์มแวร์ที่ปรับแต่งมาโดยเฉพาะ โดยมีข้อกำหนดสำคัญที่ต้องใช้ฮาร์ดแวร์ เซ็นเซอร์ และการต่อสาย ในรูปแบบที่ผู้ผลิตกำหนดเท่านั้น

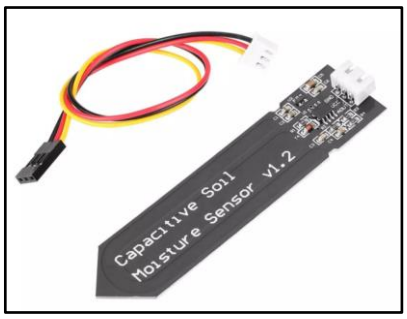
การต่อเซ็นเซอร์เข้ากับบอร์ด Farm1

อุปกรณ์ด้านอิเล็กทรอนิกส์ที่จำเป็นต้องใช้ มีดังนี้

1. บอร์ด Farm1

2. เซ็นเซอร์วัดอุณหภูมิและความชื้น SHT20

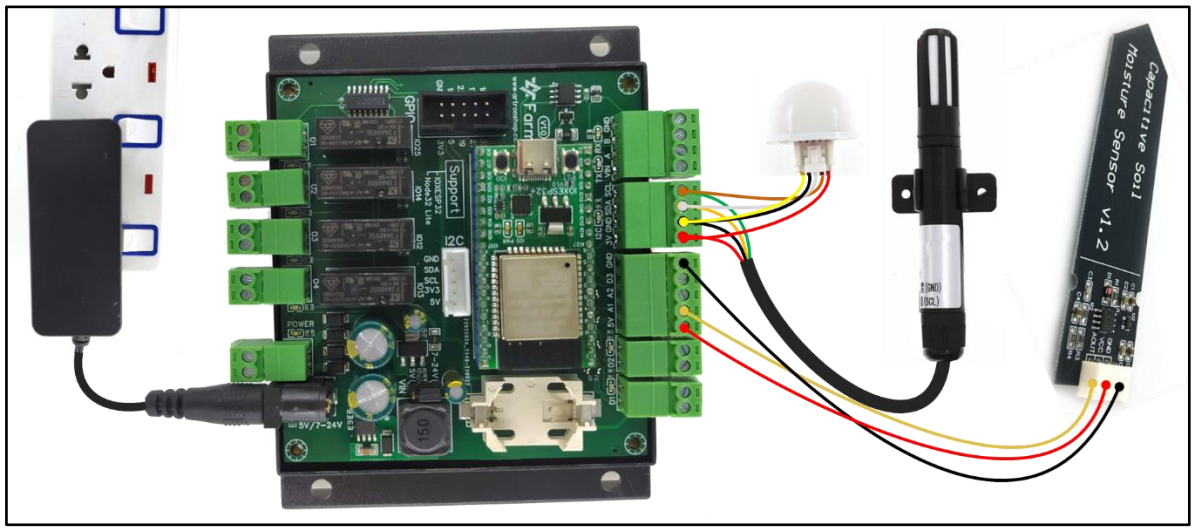
3. เซ็นเซอร์วัดความชื้นในดิน



4. เซ็นเซอร์วัดแสง (BH1750)

5.อะแดปเตอร์ 12V 1A หัว DC Jack

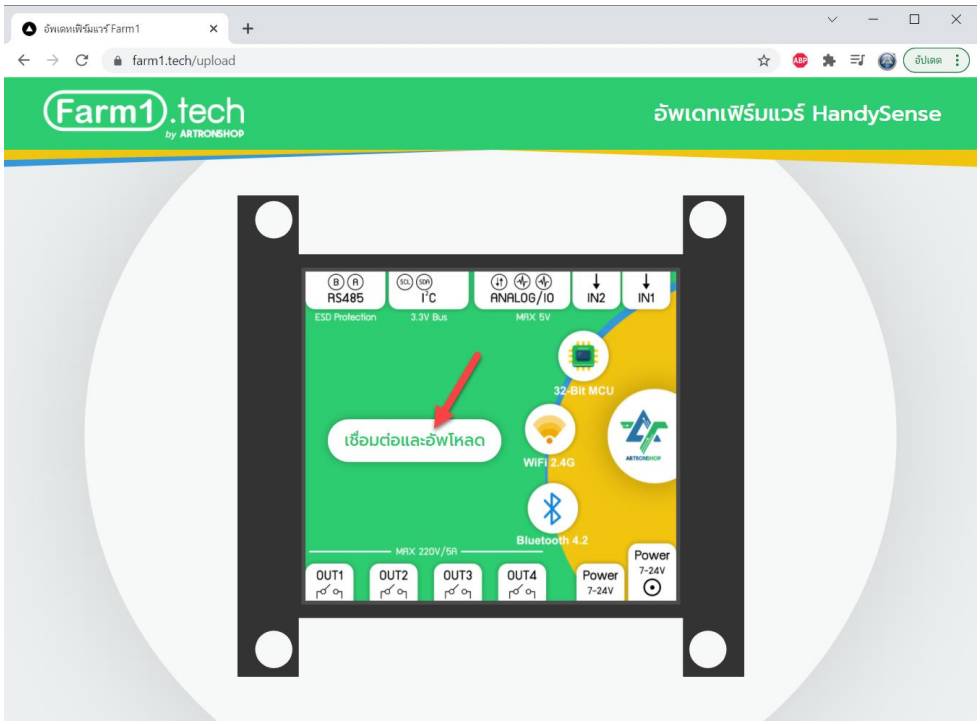
ต่ออุปกรณ์เข้าด้วยกันดังนี้



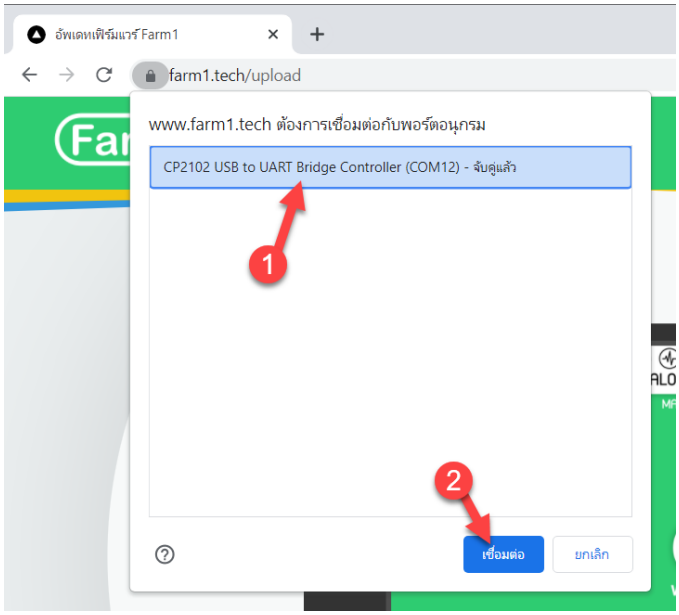
การอัพเดทเฟิร์มแวร์ HandySense

เปิดฝากล่อง Farm1 ออกมา แล้วใช้สาย USB-C เชื่อมต่อเข้ากับคอมพิวเตอร์ตามหัวข้อ การเชื่อมต่อ Farm1 เข้ากับเครื่องคอมพิวเตอร์ ติดตั้งไดร์เวอร์ตามหัวข้อ ติดตั้งไดร์เวอร์ FT231X

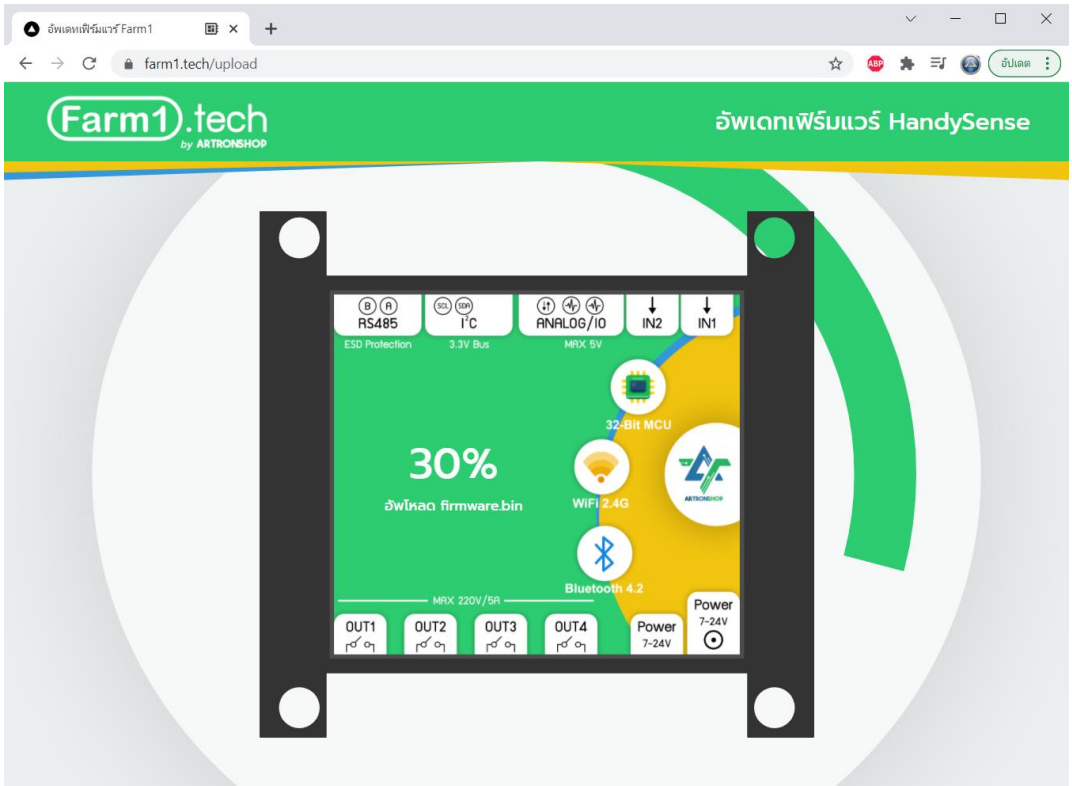
อัพเดทเฟิร์มแวร์ HandySense ให้เป็นเวอร์ชั่นล่าสุด โดยใช้โปรแกรม Google Chome หรือ Microsoft Edge เวอร์ชั่นล่าสุด เข้าไปที่ https://www.farm1.tech/upload กดปุ่ม เชื่อมต่อและอัพโหลด



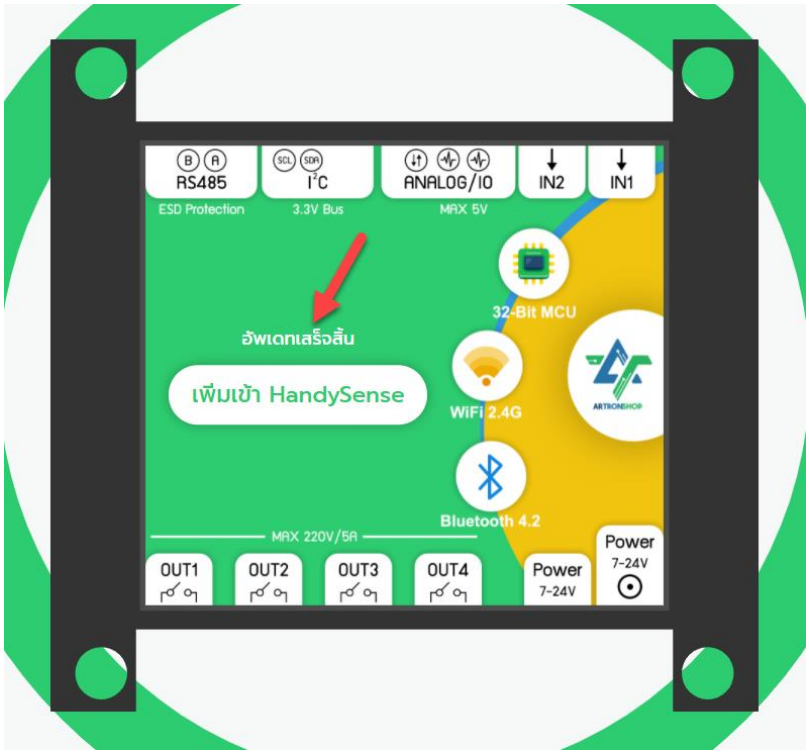
เลือกพอร์ตของบอร์ด แล้วกดปุ่ม เชื่อมต่อ



รออัพเดทเฟิร์มแวร์ซักครู่

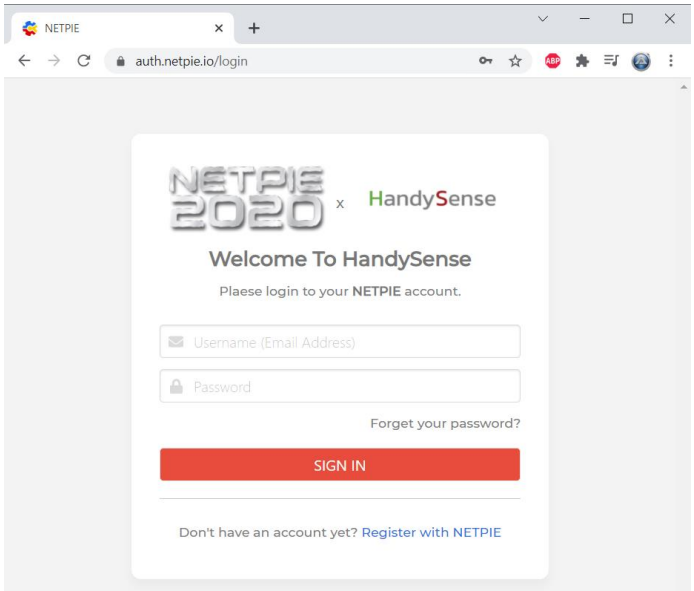


เมื่อเสร็จแล้วจะมีข้อความแจ้ง อัพเดทเสร็จสิ้น (ดังรูป)



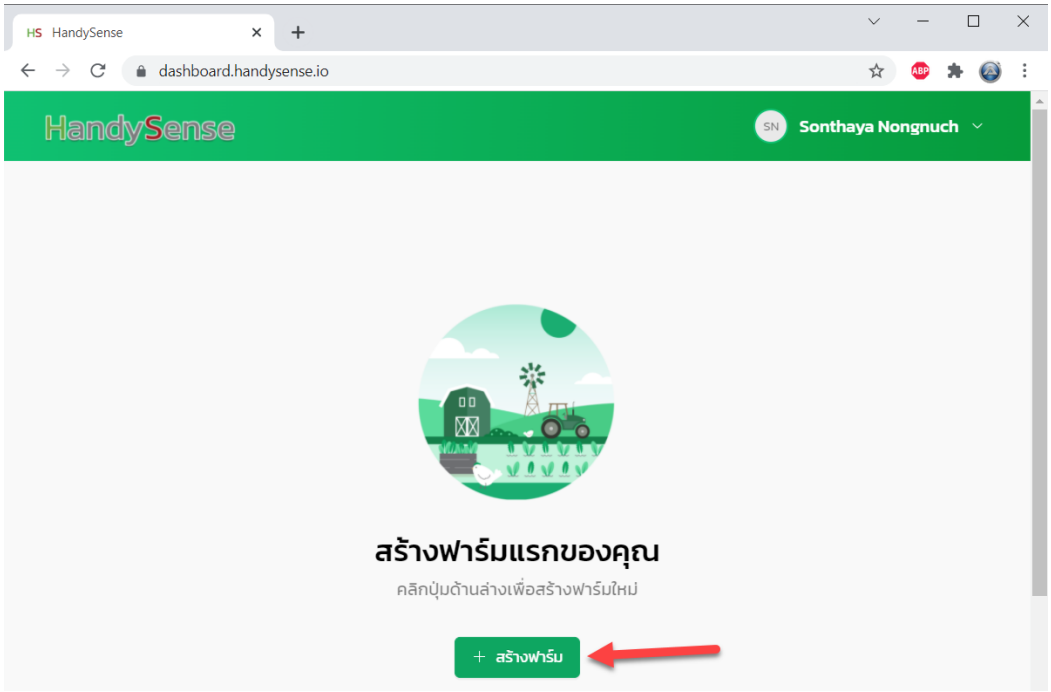
การเพิ่มอุปกรณ์เข้าระบบ HandySense

เข้าไปที่ dashboard.handysense.io จากนั้นสมัครสมาชิก หรือล็อกอินตามขั้นตอน

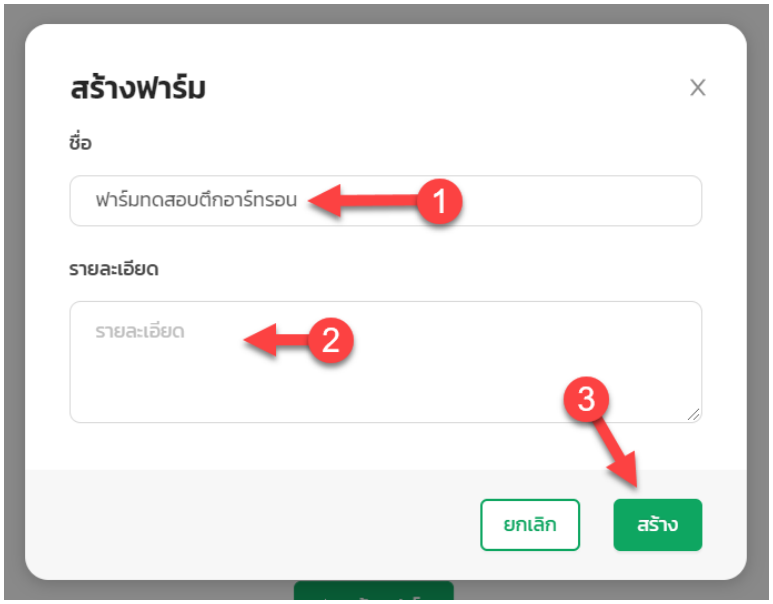


จากนั้นระบบจะพาเข้ามาหน้าแรก ซึ่งเป็นหน้าจัดการฟาร์ม กดปุ่ม สร้างฟาร์ม

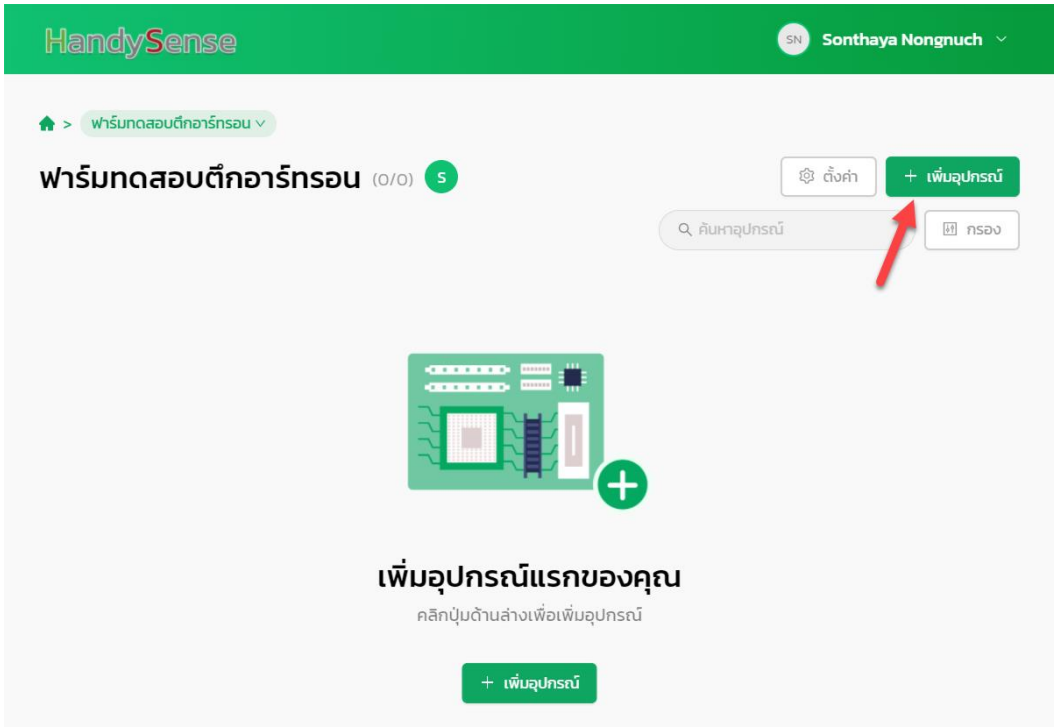
■ HandySense ออกแบบระบบให้ต้องมีฟาร์มก่อน จากนั้นจึงจะเพิ่มอุปกรณ์ลงในฟาร์มได้ ใน 1 ฟาร์ม เพิ่มได้หลายอุปกรณ์



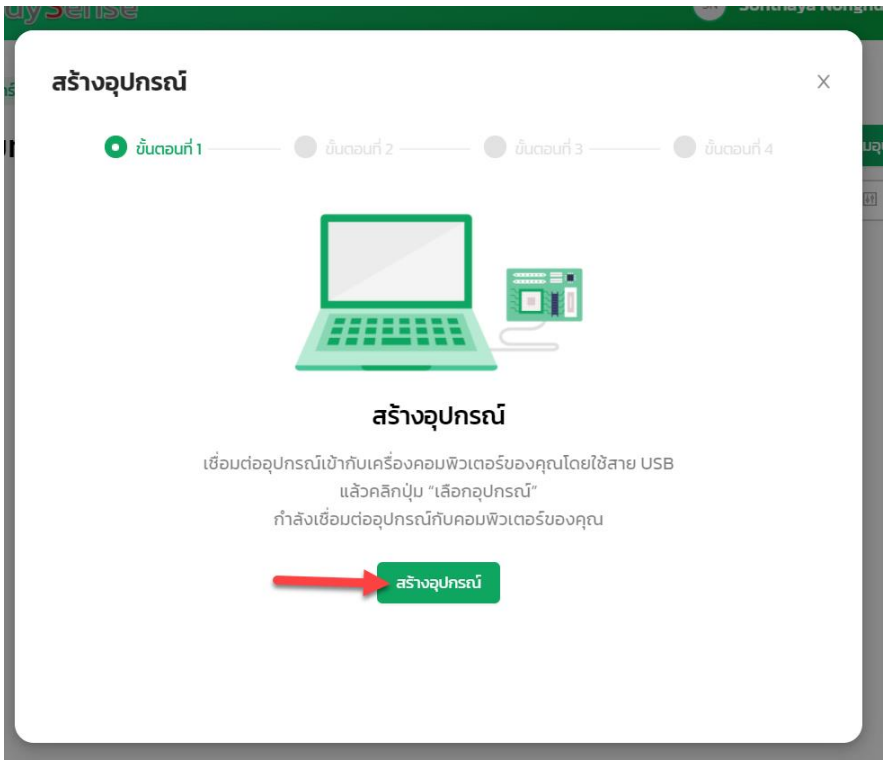
ในหน้าต่างสร้างฟาร์ม ให้ตั้งชื่อฟาร์ม (ห้ามมีเว้นวรรค) ใส่รายละเอียด (เว้นว่างได้) จากนั้นกดปุ่ม สร้าง



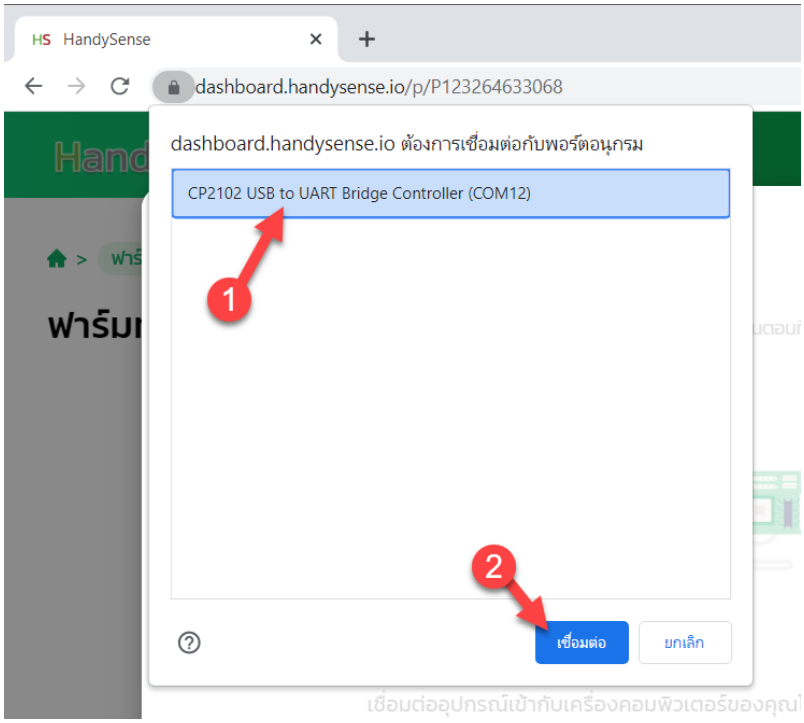
ระบบจะพาเข้ามาหน้าฟาร์มที่สร้างอัตโนมัติ ให้กดปุ่ม เพิ่มอุปกรณ์ เพื่อเริ่มเพิ่มอุปกรณ์ทันที



หน้าต่างสร้างอุปกรณ์จะแสดงขึ้นมา ในขั้นตอนที่ 1 เป็นแจ้งเตือนให้เชื่อมต่ออุปกรณ์กับเครื่องคอมพิวเตอร์ ให้กดปุ่ม สร้างอุปกรณ์

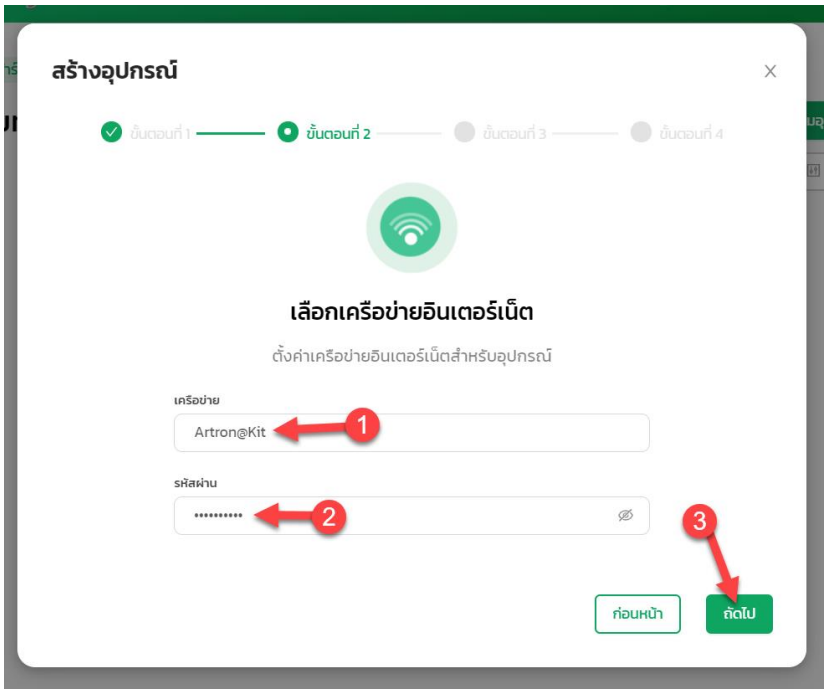


เลือกพอร์ตของอุปกรณ์ แล้วกดปุ่ม เชื่อมต่อ

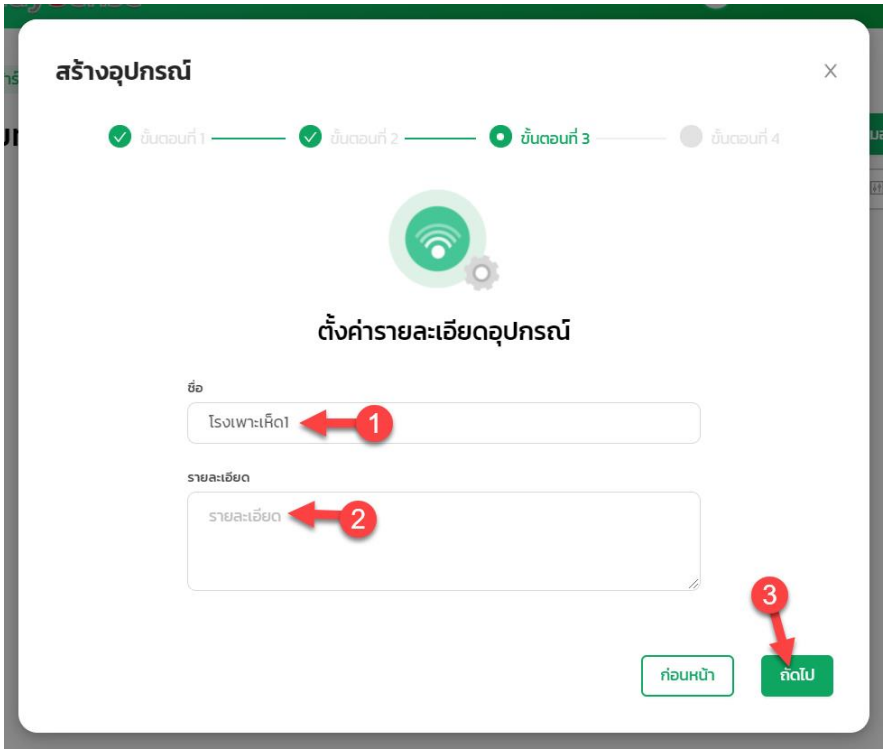


ขั้นตอนที่ 2 ตั้งค่าการเชื่อมต่อ WiFi ให้อุปกรณ์ แล้วกดปุ่ม ถัดไป

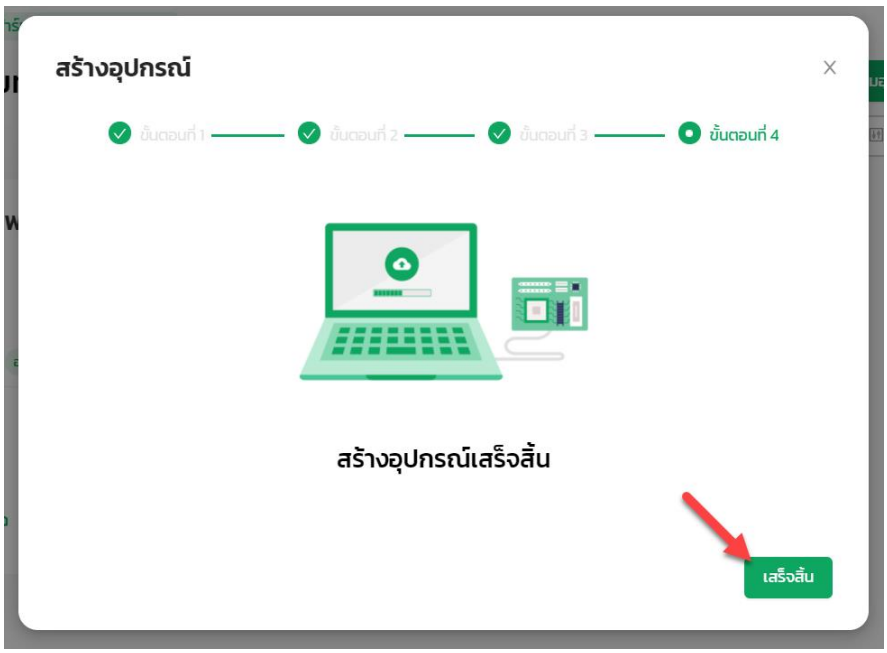
- Farm1 รองรับการเชื่อมต่อ WiFi 2.4GHz เท่านั้น
- ถ้า WiFi ไม่มีรหัสผ่าน ช่องรหัสผ่านให้เว้นว่างไว้



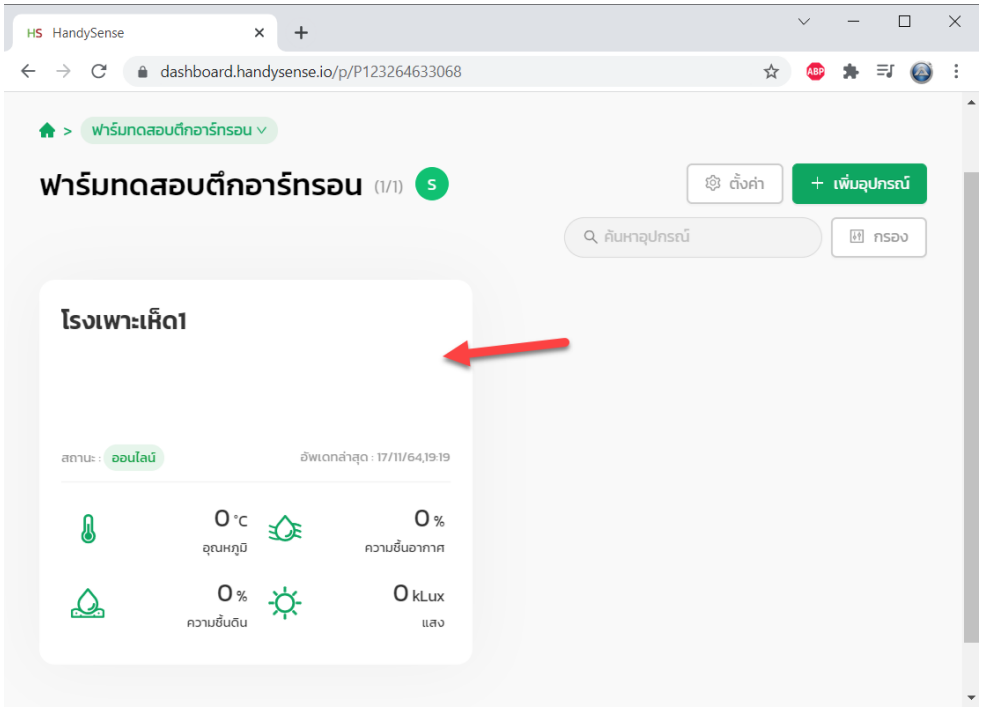
ขั้นตอนที่ 3 ตั้งชื่อชุดอุปกรณ์ ใส่รายละเอียด (เว้นว่างได้) แล้วกดปุ่ม ถัดไป



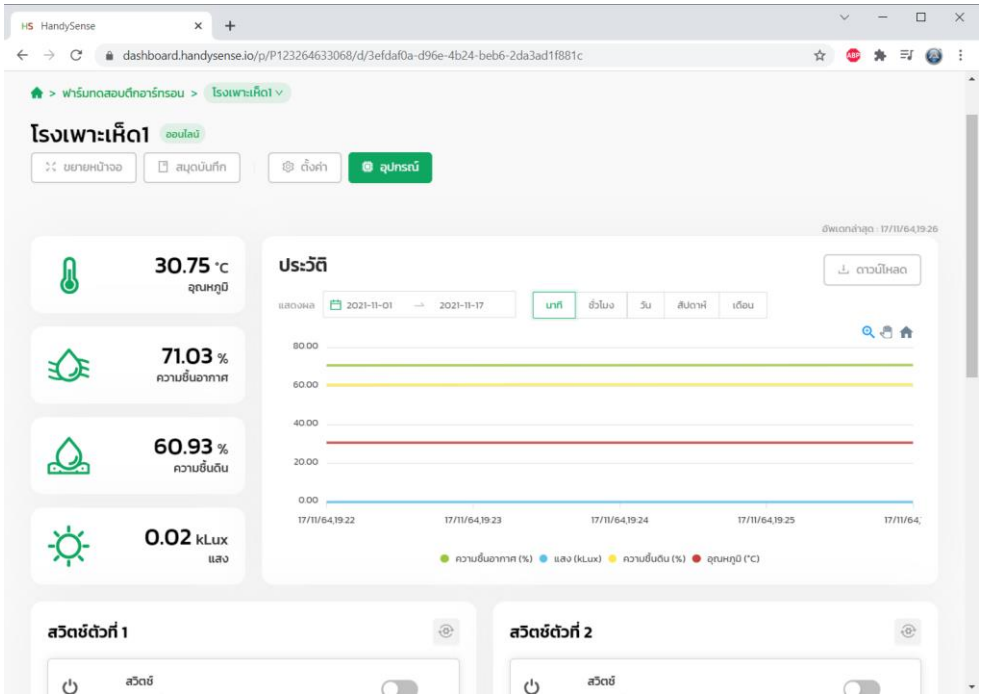
รอประมวลผล ใช้เวลา 2 – 5 วินาที จะแสดงหน้าขั้นตอนที่ 4 สร้างอุปกรณ์เสร็จสิ้น กดปุ่ม เสร็จสิ้น เพื่อปิดหน้าต่างนี้



จะมีอุปกรณ์เพิ่มเข้ามาในฟาร์มแล้ว กดที่กล่องอุปกรณ์เพื่อดูรายละเอียด และตั้งค่าต่าง ๆ



จากนั้นรอประมาณ 2 นาที (ค่าจากเซ็นเซอร์ส่งขึ้น HandySense ทุก ๆ 2 นาที) ค่าที่วัดได้จากเซ็นเซอร์ก็จะแสดงขึ้นมาแล้ว



รายละเอียดการใช้งานระบบ ศึกษาเพิ่มเติมได้จากเอกสาร [คู่มือการใช้งาน Web Application: HandySense ระบบเกษตรแม่นยำ ฟาร์มอัจฉริยะ]

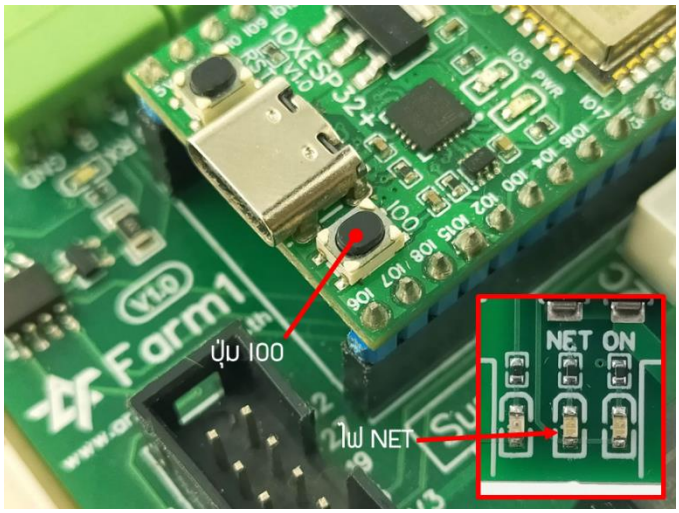
ตั้งค่าเชื่อมต่อ WiFi ด้วยแอพฯ EspTouch

การเพิ่มอุปกรณ์ลง HandySense ที่หน้างานบางครั้งอาจจะไม่สะดวก วิธีการที่สะดวกกว่าคือการเพิ่มอุปกรณ์ลง HandySense และทดสอบให้เรียบร้อย เมื่อนำไปติดตั้งหน้างานจริงจึงค่อยใช้แอพพลิเคชั่น EspTouch เปลี่ยนการตั้งค่าการเชื่อมต่อ WiFi เป็นของหน้างาน โดยมีขั้นตอนการเปลี่ยนการตั้งค่าดังนี้

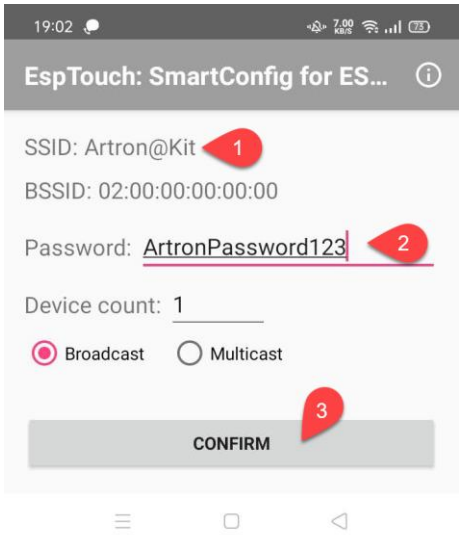
■ แอพฯ EspTouch ใช้ได้กับเฟิร์มแวร์ HandySense for Farm1 เวอร์ชั่นล่าสุดเท่านั้น หากใช้เฟิร์มแวร์เวอร์ชั่นเก่าแนะนำให้ติดตั้งเฟิร์มแวร์ใหม่ตามหัวข้อ การอัพเดทเฟิร์มแวร์ HandySense ก่อน

เข้าไปที่ Google Play ค้นหา และติดตั้งแอพพลิเคชั่น EspTouch ให้เรียบร้อย

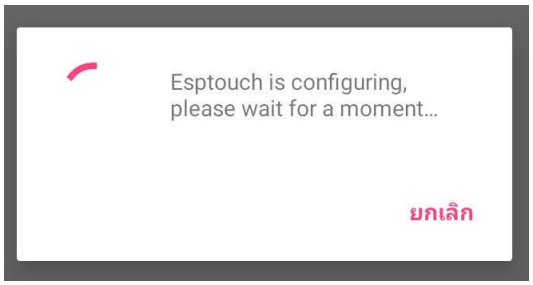
ที่กล่อง Farm1 ให้จ่ายไฟ แล้วกดปุ่ม IO0 ค้างไว้ 5 วินาที แล้วปล่อย สังเกตไฟ NET กระพริบเร็วขึ้น แสดงว่าเข้าโหมด SmartConfig ซึ่งสามารถใช้แอพพลิเคชั่น EspTouch ตั้งค่าได้แล้ว



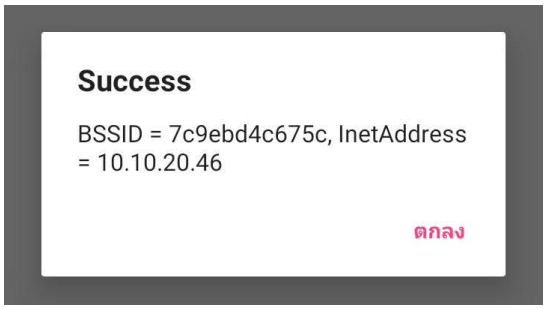
เปิดแอพฯ EspTouch ขึ้นมา ชื่อ WiFi เป็นชื่อเดียวกับชื่อที่โทรศัพท์มือถือเชื่อมต่ออยู่ (1) (หากต้องการเปลี่ยน WiFi ให้เปลี่ยนการเชื่อมต่อที่โทรศัพท์ก่อน) จากนั้นใส่รหัสผ่านของ WiFi (2) แล้วกดปุ่ม CONFIRM (3)



รอแอพฯ ส่งข้อมูลการตั้งค่า WiFi ไปที่กล่อง Farm1 ประมาณ 20 วินาที ถึง 1 นาที



เมื่อตั้งค่าสำเร็จ จะแสดงหมายเลข IP ที่กล่อง Farm1 ได้รับ (ดังรูป) กล่อง Farm1 จะรีบูต และพยามเชื่อมต่อ WiFi ใหม่ที่ได้ตั้งค่าไว้ โดยสังเกตไฟ NET กระพริบช้า ๆ แสดงว่ากำลังเชื่อมต่อ WiFi และหากติดค้างแสดงว่าเชื่อมต่อ WiFi ได้แล้ว



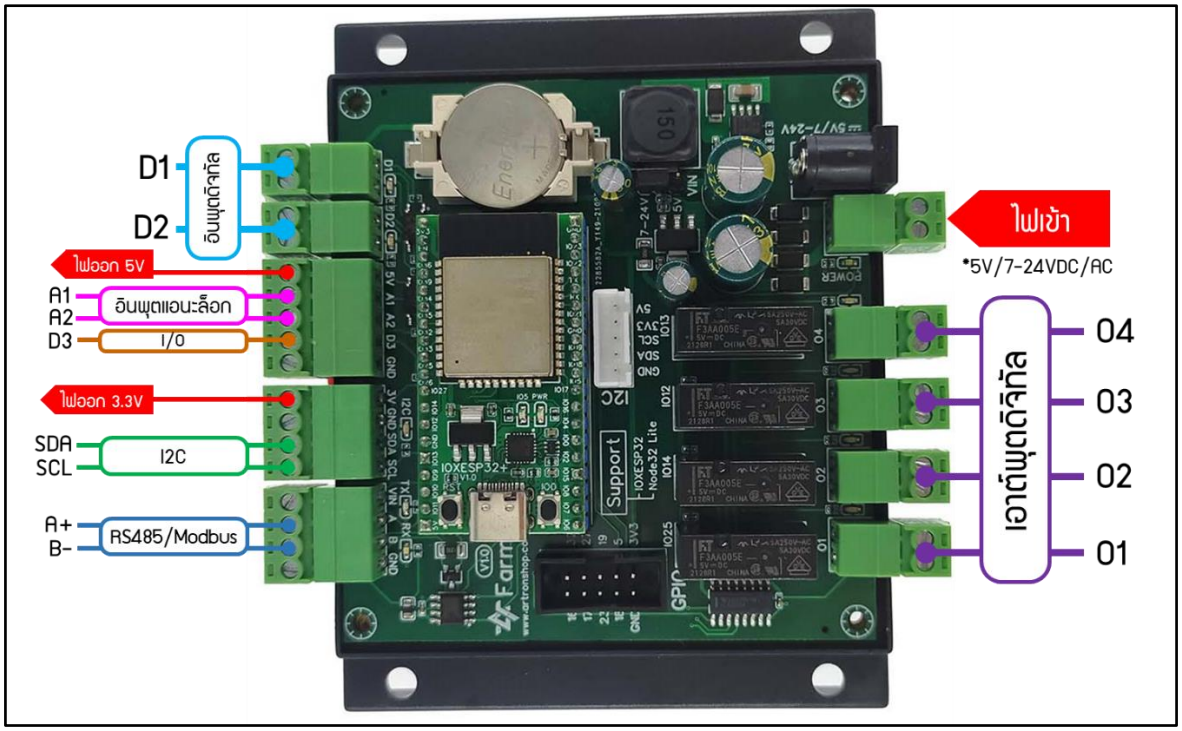
เป็นอันจบขั้นตอนการเปลี่ยนการตั้งค่า WiFi ด้วยแอพฯ EspTouch

ข้อมูลด้านฮาร์ดแวร์และวงจร

คุณสมบัติทางไฟฟ้า

คุณสมบัติ	ต่ำสุด	ปกติ	สูงสุด
แรงดันไฟฟ้า (DC/AC)	5V	12V	24V
กระแสไฟฟ้า	-	-	1A
การทนกระแสไฟฟ้าช่องเอาต์พุต	-	5A/220V	-
แรงดันช่องอินพุตดิจิทัล	3V	-	12V
กระแสช่องอินพุตดิจิทัล	0A (LOW)	-	3mA (HIGH)
แรงดันช่องอินพุตแอนะล็อก	0V	-	9.9V
กระแสช่องอินพุตแอนะล็อก	-	-	<1mA
แรงดันช่อง I/O (D3)	0V	-	3.3V
กระแสช่อง I/O ขณะเป็นอินพุต	0A (LOW)	-	1mA (HIGH)
กระแสช่อง I/O ขณะเป็นเอาต์พุต	0A	-	10mA
แรงดันช่อง I^2C	-	3.3V	-
แรงดันช่อง VIN	แรงดันไฟเข้า - 0.6V		
แบตเตอรี่วงจรนับเวลา (RTC)	3V (CR2032)		

ตำแหน่งขาต่อใช้งาน



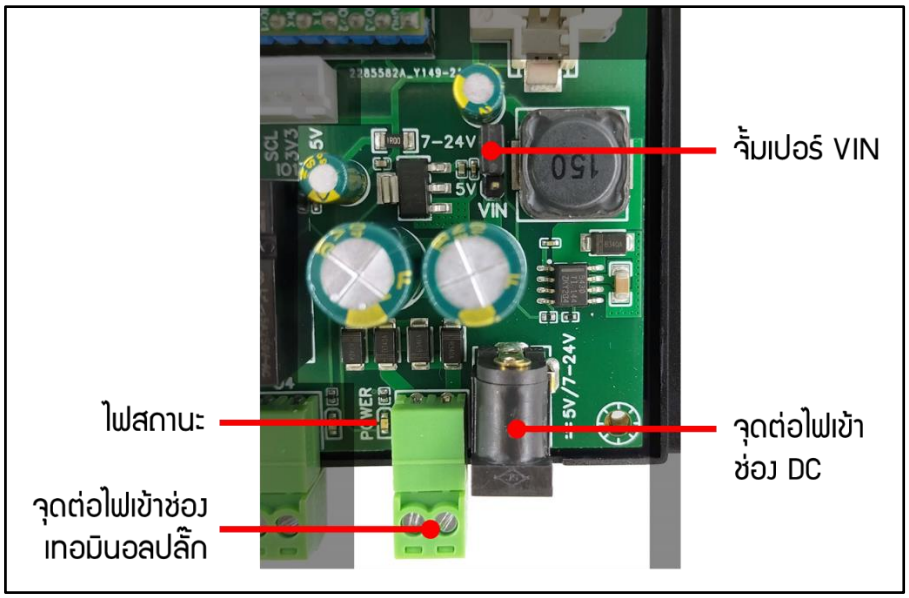
รูปที่ 1 ตำแหน่งขาต่อใช้งาน Farm1

การจ่ายไฟ

รองรับการจ่ายไฟผ่านช่อง DC / เทอมินอลปลั๊ก รองรับไฟเข้าได้ 5V และ 7V ถึง 24V (ขึ้นอยู่กับจั้มเปอร์) ทั้งกระแสตรง (DC) และกระแสสลับ (AC) กระแสไฟฟ้าที่ต้องการต่ำสุด 1A

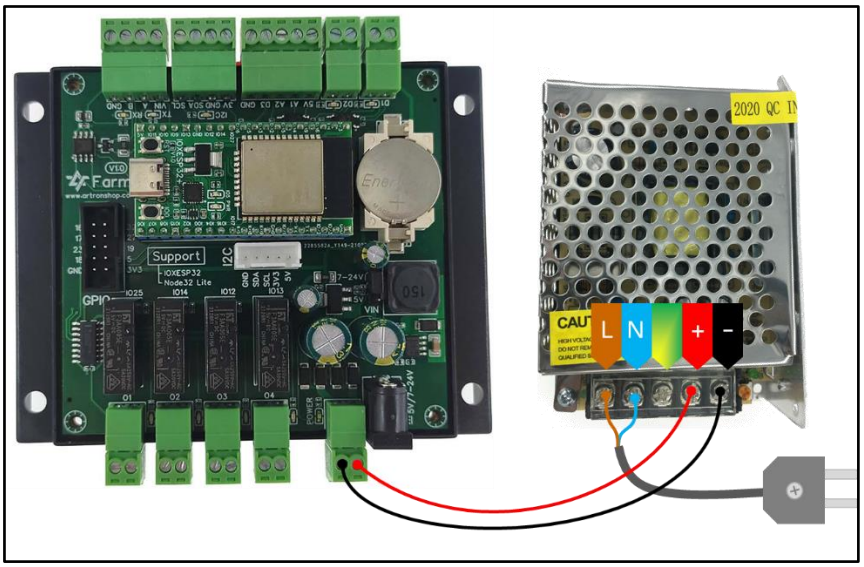
จั้มเปอร VIN ใช้เลือกระดับแรงดันไฟเข้า โดยหากจั้มที่ 5V ต้องจ่ายไฟเข้า 5V เท่านั้น (DC) หากเลือกเป็น 7-24V เลือกจ่ายไฟด้วยหม้อแปลง แบตเตอรี่ โซล่าเซลล์ หรือแหล่งจ่ายไฟอื่นที่อยู่ในช่วง 7V ถึง 24V ได้ รองรับทั้งไฟฟ้ากระแสตรงและกระแสสลับ

หลอดแอลอีดี POWER ติดสว่างเป็นสีเขียวเมื่อได้รับไฟฟ้าจากแหล่งจ่ายไฟ



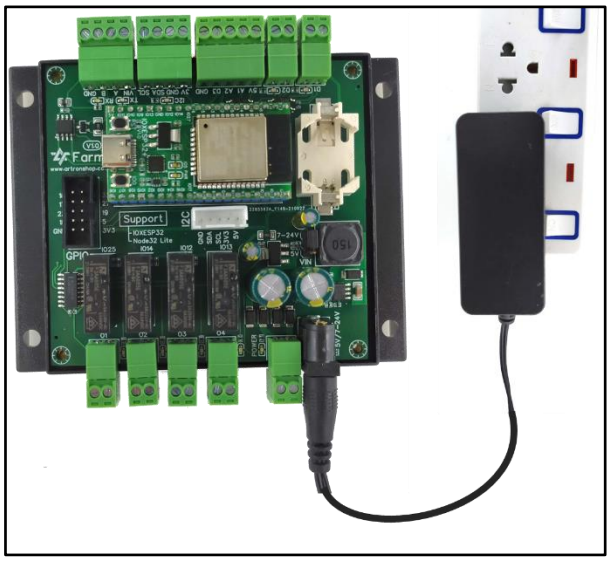
รูปที่ 2 วงจรส่วนภาคจ่ายไฟ

ตัวอย่างการจ่ายไฟให้บอร์ด Farm1 ด้วยสวิทชิ่งเพาเวอร์ซัพพลาย 12V แสดงดังรูปที่ 3



รูปที่ 3 การจ่ายไฟให้บอร์ด Farm1 ด้วยสวิทชิ่งเพาเวอร์ซัพพลาย 12V

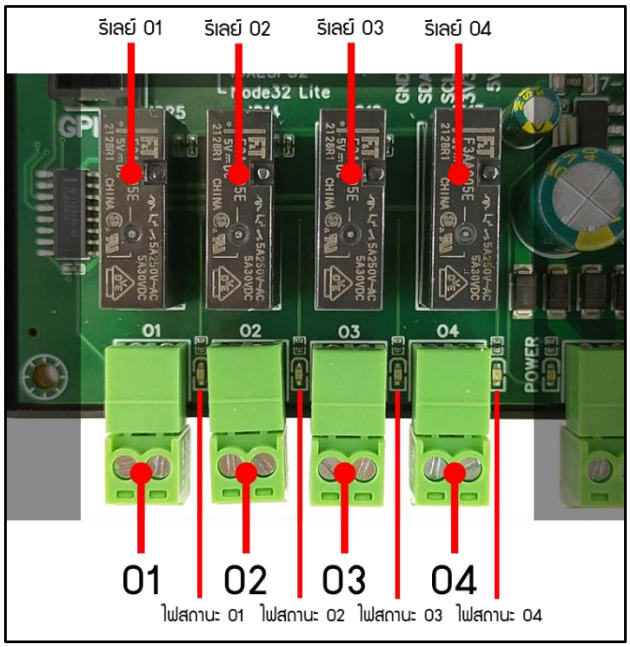
ตัวอย่างการจ่ายไฟให้บอร์ด Farm1 ด้วยอะแดปเตอร์ 9V แสดงดังรูปที่ 4



รูปที่ 4 การจ่ายไฟให้บอร์ด Farm1 ด้วยอะแดปเตอร์ 9V

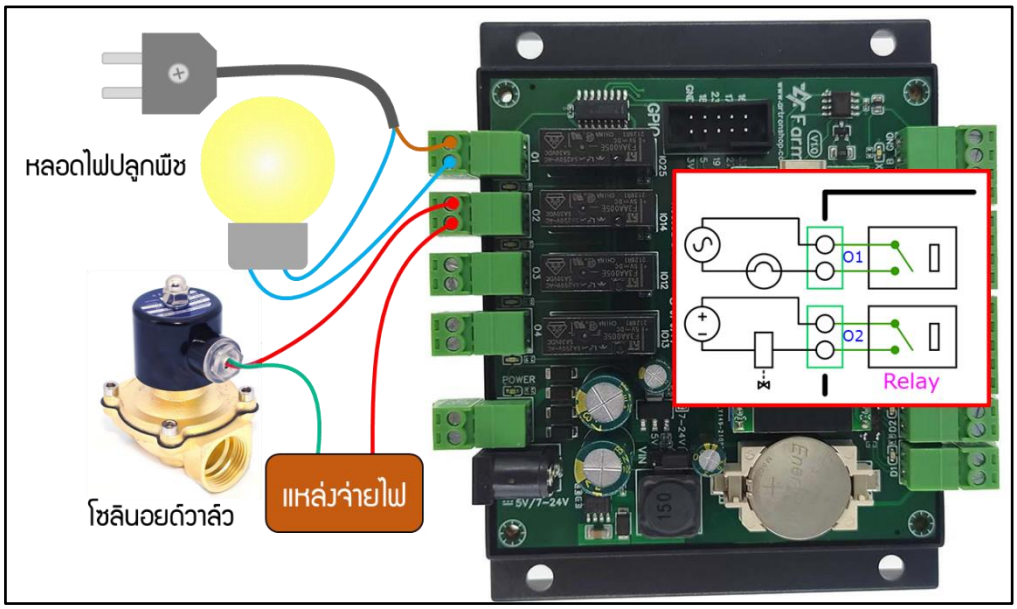
ช่องเอาต์พุต

ช่องเอาต์พุตประกอบด้วย 01 ถึง 04 เป็นเอาต์พุตแบบหน้าคอนเทครีเลย์ NO ทนกระแสได้สูงสุด 5A ที่ 220V มีหลอดแอลอีดีแสดงสถานการณ์ทำงาน โดยจะติดสว่างเป็นสีส้มเมื่อหน้าคอนเทคต่อ (รีเลย์ทำงาน)



รูปที่ 5 ส่วนประกอบภาคเอาต์พุต

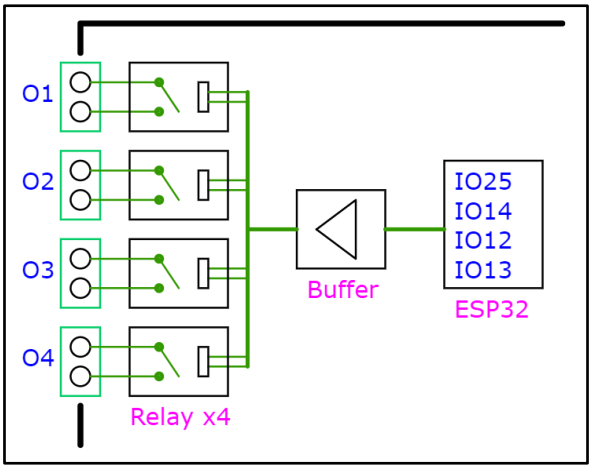
ตัวอย่างการต่อช่องเอาต์พุตเข้ากับหลอดไฟและโซลินอยด์วาล์ว แสดงดังรูปที่ 6



รูปที่ 6 การต่อช่องเอาต์พุตเข้ากับหลอดไฟและโซลินอยด์วาล์ว

■ การต่อช่องเอาต์พุตเข้ากับอุปกรณ์กำลังสูง ต้องต่อผ่านแมกเนติก (Magnetic Contactor) เท่านั้น

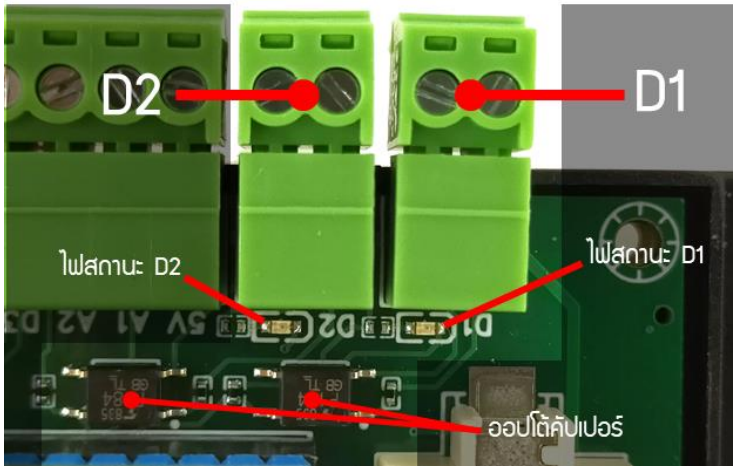
รีเลย์ของช่องเอาต์พุตต่อเข้ากับ ESP32 ผ่านขา 25, 14, 12 และ 13 ตามลำดับ บล็อกไดอะแกรมส่วนเอาต์พุตดิจิทัลแสดงดังรูปที่ 7



รูปที่ 7 บล็อกไดอะแกรมส่วนเอาต์พุตดิจิทัล

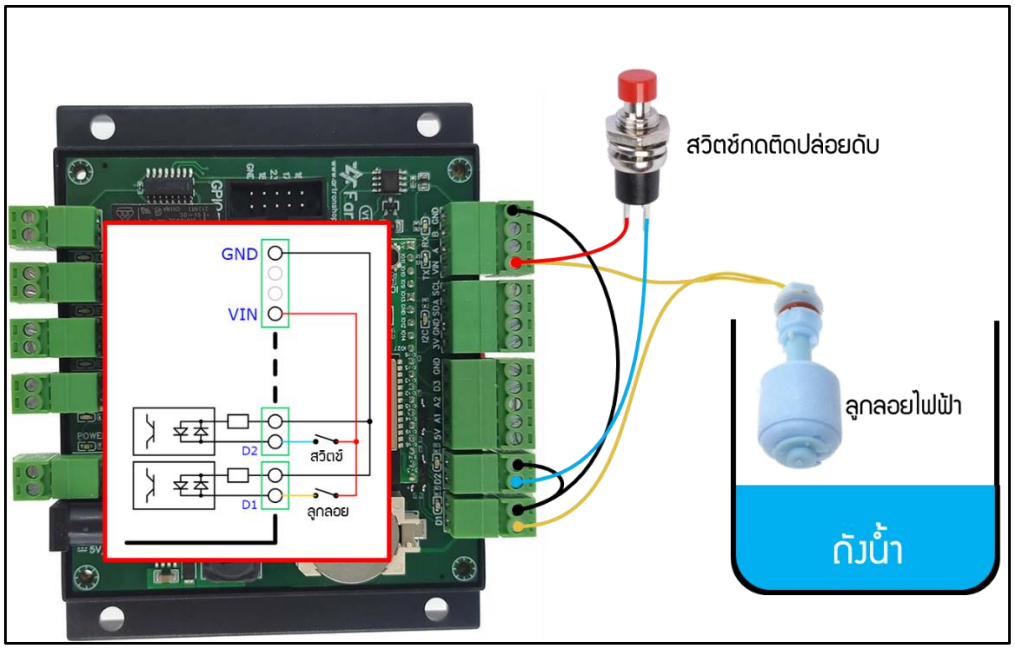
ช่องอินพุตดิจิทัล

ช่องอินพุตดิจิทัลประกอบด้วย D1 และ D2 รองรับไฟเข้าได้ 3V ถึง 12V คั่นระหว่างสัญญาณเข้ากับ ESP32 ด้วยออปโต้คัปเปอร์ มีหลอดแอลอีดีสีส้ม ติดเมื่อได้รับลอจิก 1



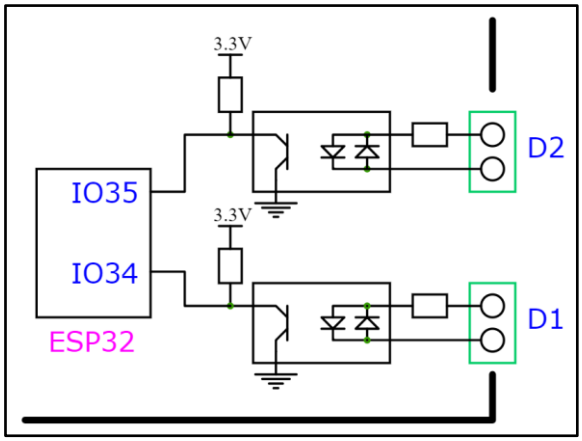
รูปที่ 8 ส่วนประกอบภาคอินพุตดิจิทัล

ตัวอย่างการต่อสวิตช์ และลูกลอยเข้ากับช่องอินพุต แสดงดังรูปที่ 9



รูปที่ 9 การต่อสวิตช์ และลูกลอยเข้ากับช่องอินพุต

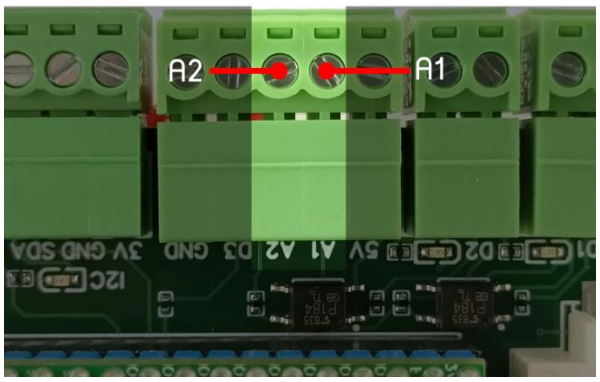
ช่องอินพุตดิจิทัล เชื่อมต่อกับ ESP32 ผ่านออปโต้คอปเปอร์ที่ขา 34, 35 ตามลำดับ ทำงานแบบ Active LOW เมื่อมีสัญญาณอินพุตเข้ามา สัญญาณเข้า ESP32 จะเป็น LOW และหากไม่มีสัญญาณอินพุตเข้ามา สัญญาณเข้า ESP32 จะเป็น HIGH บล็อกไดอะแกรมส่วนอินพุตดิจิทัล แสดงดังรูปที่ 10



รูปที่ 10 บล็อกไดอะแกรมส่วนอินพุตดิจิทัล

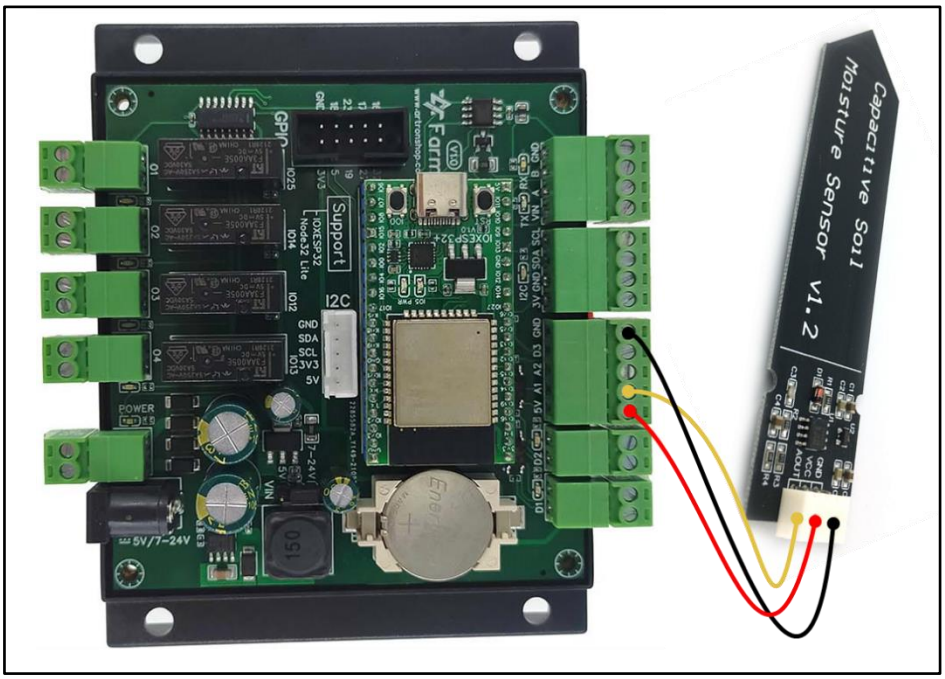
ช่องอินพุตแอนะล็อก

ช่องอินพุตแอนะล็อกประกอบด้วย A1 และ A2 รองรับไฟเข้าได้ 0V ถึง 9.9V ใช้ต่อเข้ากับเซ็นเซอร์ที่ให้เอาต์พุตเป็นแรงดันไฟฟ้า เช่น เซ็นเซอร์วัดความชื้นในดิน, เซ็นเซอร์วัดระดับน้ำ, เซ็นเซอร์วัดความขุ่นของน้ำ, เซ็นเซอร์วัดความเป็นกรด-ด่าง (PH) เป็นต้น



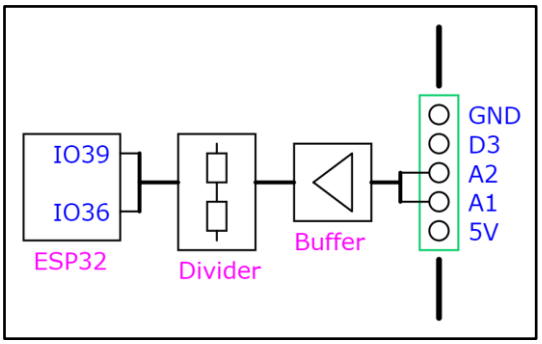
รูปที่ 11 ส่วนประกอบภาคอินพุตแอนะล็อก

ตัวอย่างการต่อเซ็นเซอร์วัดความชื้นในดินเข้ากับช่องอินพุตแอนะล็อก แสดงดังรูปที่ 12



รูปที่ 12 การต่อเซ็นเซอร์วัดความชื้นในดินเข้ากับช่องอินพุตแอนะล็อก

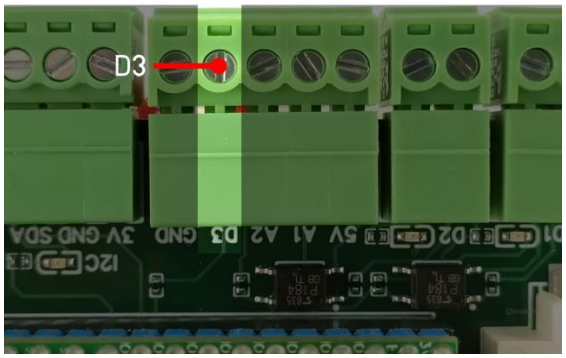
อินพุตแอนะล็อกผ่านวงจรบัฟเฟอร์จากนั้นเข้าวงจรแบ่งแรงดันเพื่อให้แรงดันไม่เกิน 3.3V ทำให้ ESP32 สามารถรับแรงดันสูงได้ ช่องแอนะล็อกต่อกับ ESP32 ขา 36, 39 ตามลำดับ บล็อกไดอะแกรมส่วนอินพุตแอนะล็อก แสดงดังรูปที่ 12



รูปที่ 13 บล็อกไดอะแกรมส่วนอินพุตแอนะล็อก

ช่อง I/O

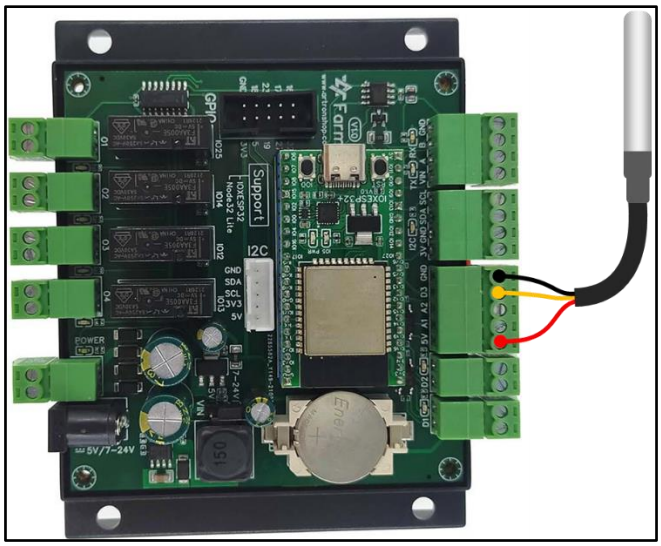
ช่อง I/O คือ D3 รองรับแรงดันไฟเข้าสูงสุด 3.3V ใช้ต่อกับกับเซ็นเซอร์/อุปกรณ์ภายนอกสื่อสารด้วยสายเส้นเดียว เช่น เซ็นเซอร์วัดอุณหภูมิและความชื้น DHT11/DHT22, เซ็นเซอร์วัดอุณหภูมิแบบกันน้ำ DS18B20, หลอดแอลอีดี RGB เบอร์ WS2812, เซอร์โวมอเตอร์ เป็นต้น



รูปที่ 14 ส่วนประกอบภาค I/O

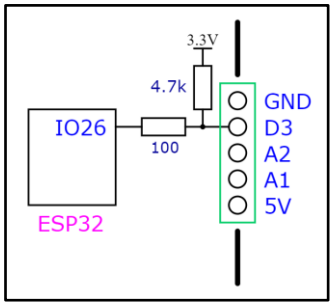
ตัวอย่างการต่อเซ็นเซอร์วัดอุณหภูมิหัวกันน้ำ DS18B20 เข้ากับช่อง I/O แสดงดังรูปที่ 15

■ DS18B20 ทำงานแบบ 1-wire ตามข้อกำหนด ต้องต่อตัวต้านทาน Pull-up ด้วย



รูปที่ 15 การต่อเซ็นเซอร์วัดอุณหภูมิหัวกันน้ำ DS18B20 เข้ากับช่อง I/O

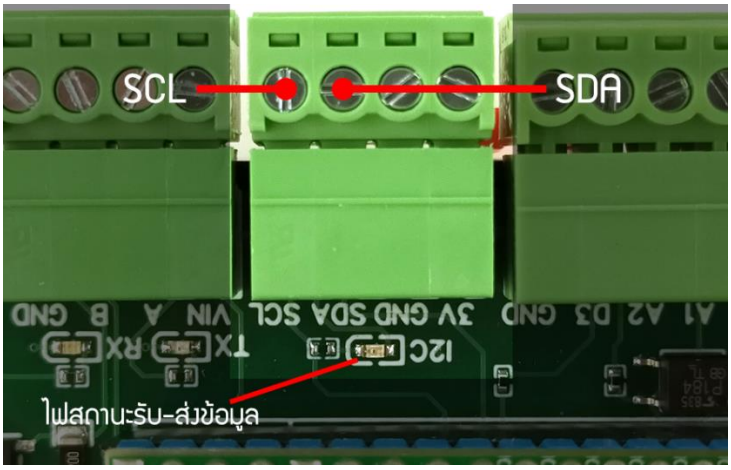
ช่อง I/O ต่อตรงกับ ESP32 ขา 26 คั่นด้วยตัวต้านทาน 100Ω เพื่อป้องกันกระแสไหลเกิน พร้อมตัวต้านทาน 4.7 kΩ เป็น Pull-up บล็อกไดอะแกรมส่วน I/O แสดงดังรูปที่ 16



รูปที่ 16 บล็อกไดอะแกรมส่วน I/O

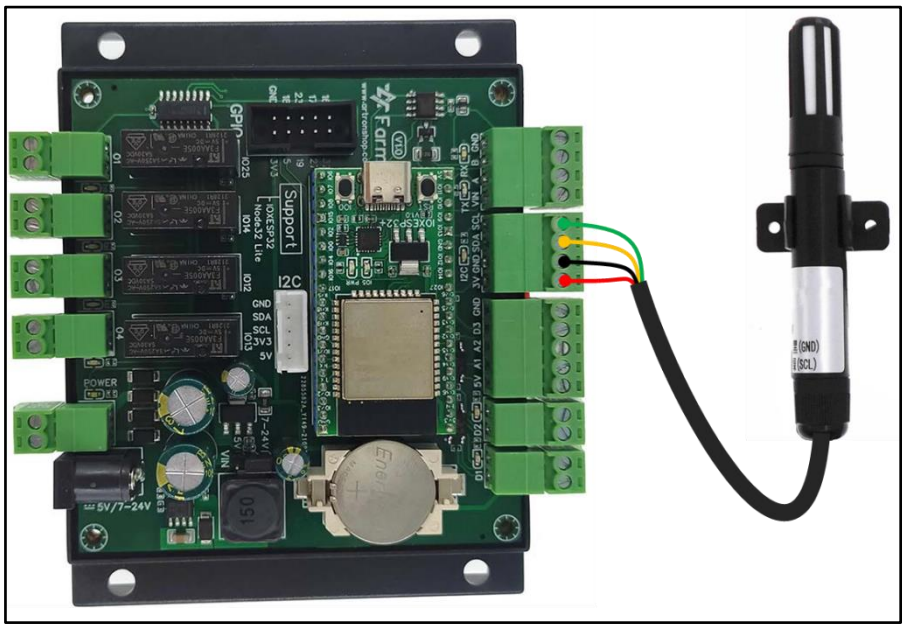
ช่อง I²C

ช่อง I²C ประกอบด้วย SCL และ SDA ใช้ต่อกับอุปกรณ์ I²C เช่น เซ็นเซอร์วัดอุณหภูมิและความชื้น SHT30, เซ็นเซอร์วัดความกดอากาศ BMP280, เซ็นเซอร์แสง BH1750 เป็นต้น ใช้งานได้กับอุปกรณ์ที่ใช้แรงดันไฟฟ้า 3.3V มีหลอดแอลอีดีกระพริบบอกสถานะรับ-ส่งข้อมูล



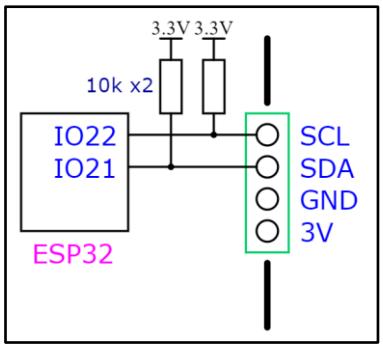
รูปที่ 17 ส่วนประกอบภาค I²C

การต่อกับเซ็นเซอร์ จำเป็นจะต้องต่อขา SCL และ SDA ให้ถูกต้อง ตัวอย่างการต่อเซ็นเซอร์วัดอุณหภูมิและความชื้น SHT20 แสดงดังรูปที่ 18



รูปที่ 18 การต่อเซ็นเซอร์วัดอุณหภูมิและความชื้น SHT30 เข้ากับช่อง I²C

ช่อง I2C ต่อตรงกับ ESP32 ขา 22 (SCL), 21 (SDA) โดยมีตัวต้านทาน Pull-up 10k□ ต่อไว้ให้แล้ว บล็อกไดอะแกรมส่วน I^2C แสดงดังรูปที่ 19

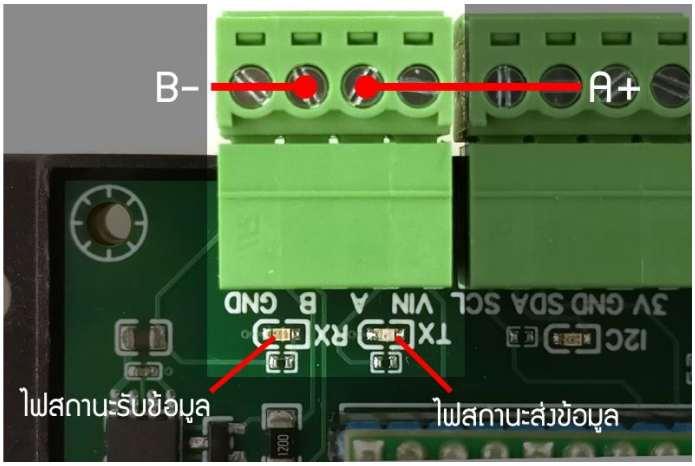


รูปที่ 19 บล็อกไดอะแกรมส่วน I2C

ช่อง RS485

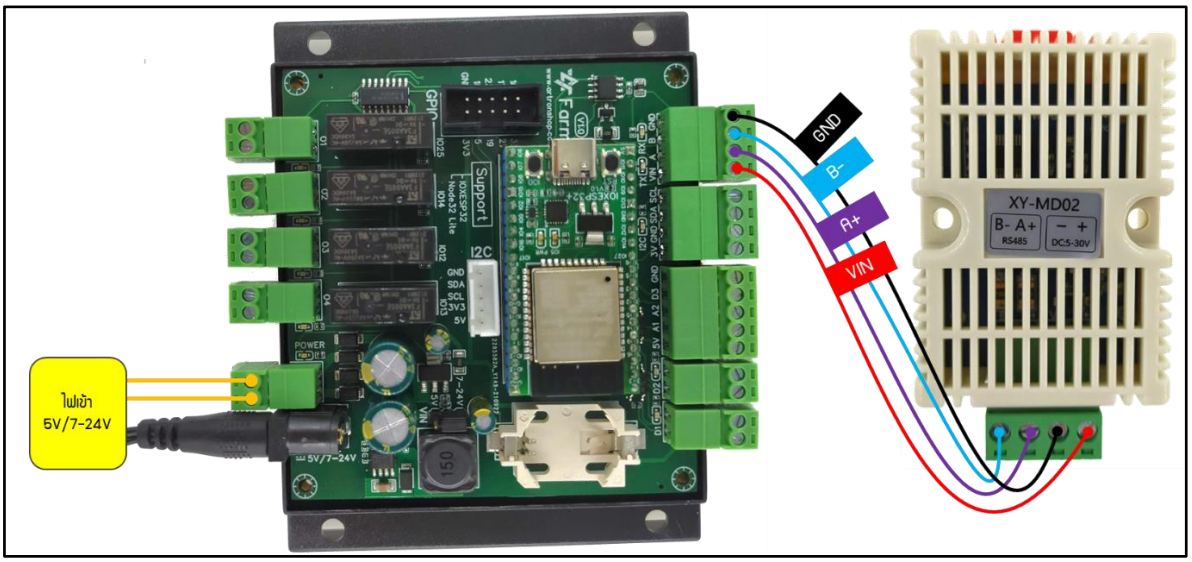
ช่อง RS485 ประกอบด้วย A+ และ B- ใช้ต่อกับเซ็นเซอร์ RS485 หรือ Modbus RTU เช่น เซ็นเซอร์วัดความชื้นในดิน เซ็นเซอร์อุณหภูมิ เซ็นเซอร์วัดระดับน้ำฝน เซ็นเซอร์วัดความเร็วลม เป็นต้น

ช่อง RS485 บนบอร์ดมี Resistor Termination ค่าความต้านทาน 120□ มาให้แล้ว พร้อม TVS ป้องกันแรงดันไฟฟ้าในบัสสูงเกินกำหนด



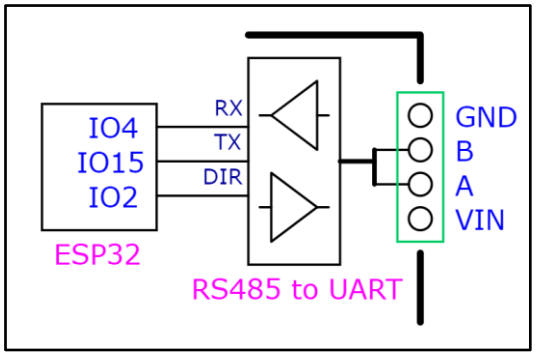
รูปที่ 20 ส่วนประกอบภาค RS485

ตัวอย่างการต่อเซ็นเซอร์วัดอุณหภูมิและความชื้น XY-MD02 เข้ากับช่อง RS485 แสดงดังรูปที่ 21



รูปที่ 21 การต่อเซ็นเซอร์วัดอุณหภูมิและความชื้น XY-MD02 เข้ากับ Fame1

ช่อง RS485 ต่อกับ ESP32 ผ่านวงจรแปลง RS485 เป็น UART เข้าที่ขา 4 (RX), 15 (TX) และ 2 (DIR) บล็อกไดอะแกรมส่วน RS485 แสดงดังรูปที่ 22



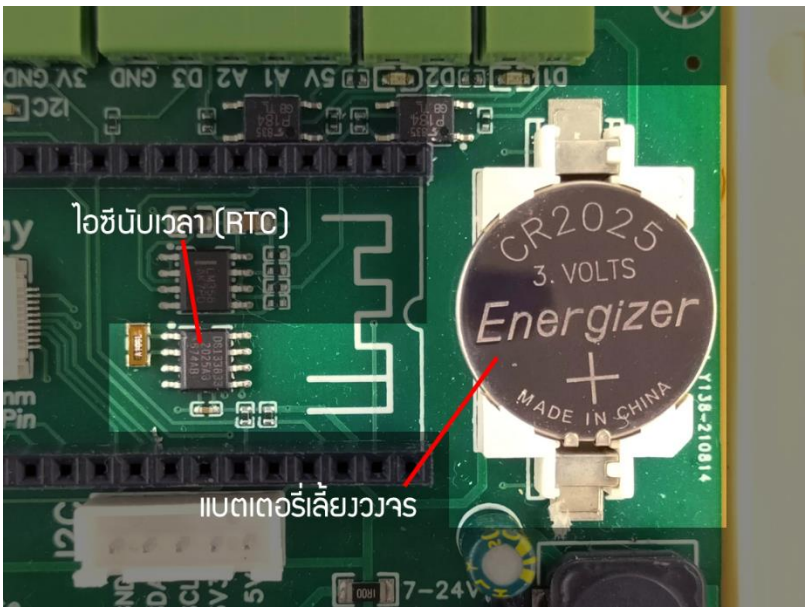
รูปที่ 22 บล็อกไดอะแกรมส่วน RS485

วงจรนับเวลา (RTC)

วงจรนับเวลาต่อตรงกับ ESP32 ผ่านช่อง I^2C ใช้นับเวลาเมื่อไม่มีอินเตอร์เน็ต หรือเมื่อไม่มีไฟเลี้ยง ใช้ประกอบการเขียนโปรแกรมที่ต้องการให้ทำงานตามเวลา เช่น ส่งค่าที่วัดได้ในฟาร์มขึ้นระบบคลาวด์ตามเวลา เปิด-ปิดไฟตามเวลา เปิด-ปิดปั้มน้ำตามเวลา เป็นต้น

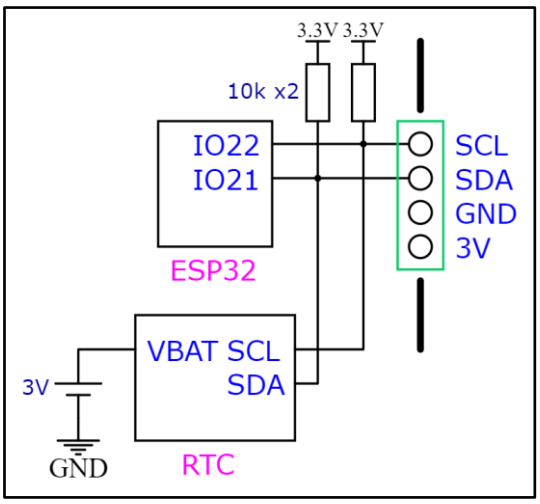
วงจรนับเวลา (RTC) จำเป็นต้องใช้ถ่าน CR2032 จ่ายไฟเลี้ยงให้วงจร โดยไฟจากถ่านจะถูกใช้เมื่อไม่มีแรงดันจากแหล่งจ่ายไฟจ่ายเข้ามา ทำให้วงจรส่วนนับเวลา (RTC) ยังมีไฟเลี้ยงอยู่ตลอด ทำให้วงจรยัง

ทำงาน นับเวลาอยู่เรื่อย ๆ จดจำค่าเวลาได้ และพร้อมให้เรียกใช้ค่าเวลาเมื่อวงจรส่วนประมวลผลกลับมาทำงานอีกครั้ง



รูปที่ 23 ส่วนประกอบวงจรนับเวลา (RTC)

บล็อกไดอะแกรมส่วนวงจรนับเวลา (RTC) แสดงดังรูปที่ 24



รูปที่ 24 บล็อกไดอะแกรมส่วนวงจรนับเวลา (RTC)

การเขียนโปรแกรมสั่งงานด้วย Arduino IDE

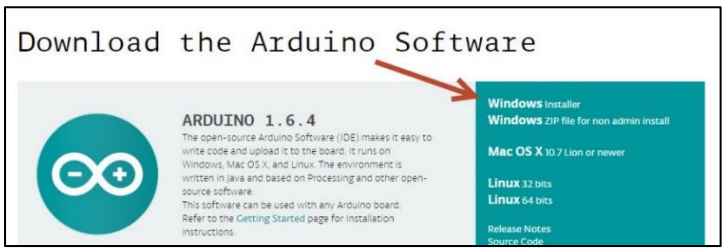
Farm1 ใช้ไมโครคอนโทรลเลอร์ ESP32 ในการประมวลผล รองรับเครื่องมือพัฒนาทั้ง ESP-IDF และ Arduino IDE ในเอกสารนี้จะแนะนำตัวอย่างการเขียนโปรแกรมแต่ละส่วนด้วย Arduino IDE

ติดตั้งโปรแกรม Arduino IDE

โปรแกรม Arduino IDE เป็นหนึ่งในเครื่องมือที่ใช้พัฒนาโปรแกรมให้ Farm1 ซึ่งประกอบไปด้วย Text Editor ใช้แก้ไขโค้ดโปรแกรม, Compiler และ Uploader

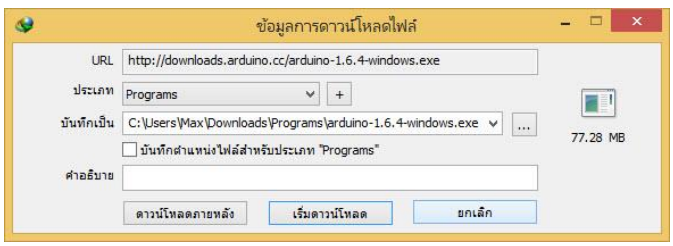
ขั้นตอนการติดตั้งโปรแกรม Arduino IDE มีดังนี้

เข้าไปที่ http://www.arduino.cc/en/Main/Software เลือก OS ให้ตรงกับเครื่องคอมพิวเตอร์



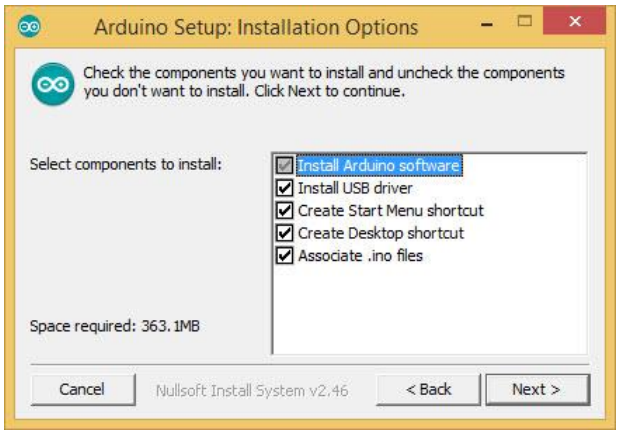
กดปุ่ม JUST DOWNLOAD เพื่อเริ่มดาว์โหลดโปรแกรม

รอจนกว่าการดาว์โหลดจะเสร็จสิ้น แล้วจึงทำขั้นตอนถัดไป

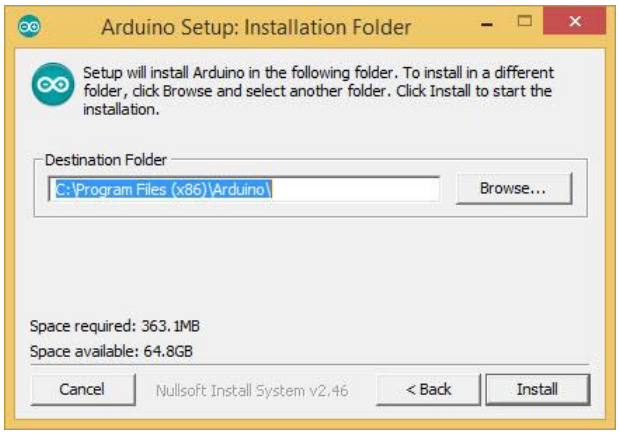


เมื่อดาว์โหลดเสร็จแล้วให้เปิดไฟล์ติดตั้งขึ้นมา แล้วกดปุ่ม I Agree เพื่อยืนยันยอมรับเงื่อนไขการใช้งานโปรแกรม

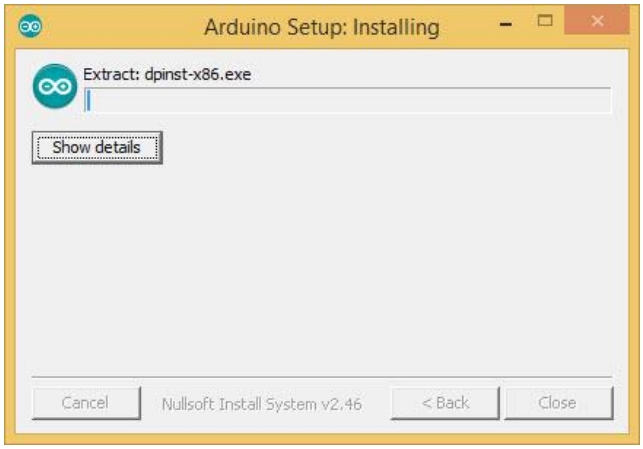
คลิกปุ่ม Next > เพื่อติดตั้งทั้งหมด



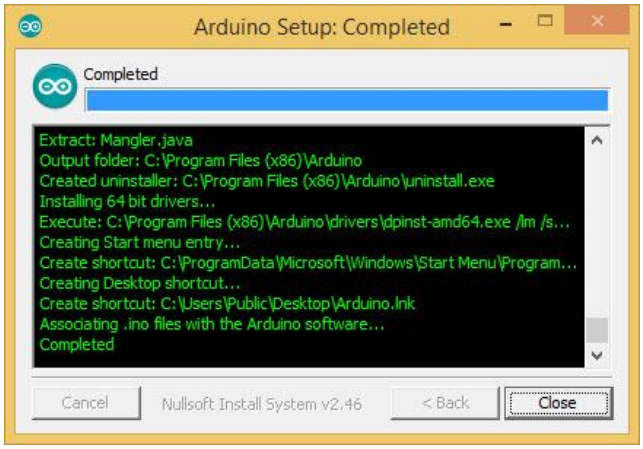
เลือกโฟลเดอร์ติดตั้งโปรแกรม หากไม่ต้องการแก้ไขคลิกปุ่ม Install



รอจนกว่าโปรแกรมจะติดตั้งเสร็จสิ้น



เมื่อขึ้นคำว่า Completed ให้กดปุ่ม Close เพื่อปิดหน้าต่างติดตั้งโปรแกรมได้เลย

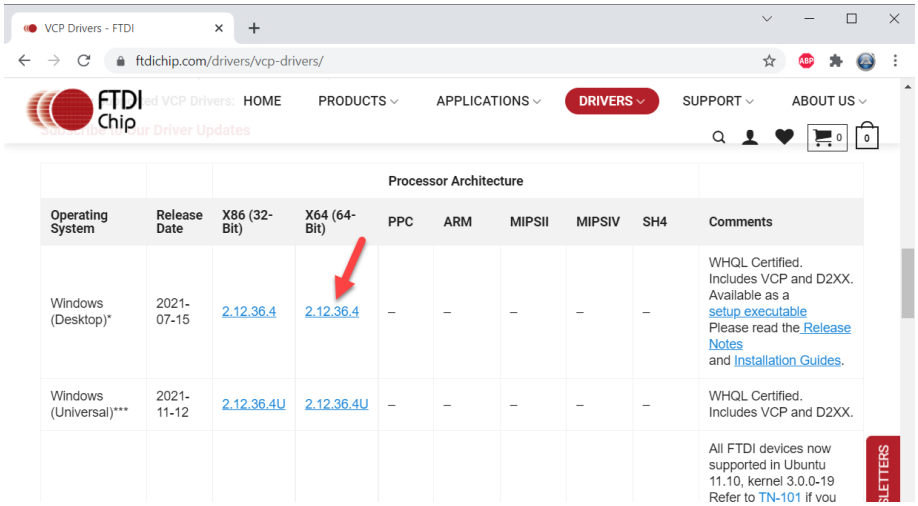


หน้าเดสท็อปก็จะมีไอค่อนโปรแกรม Arduino ขึ้นมาแล้ว

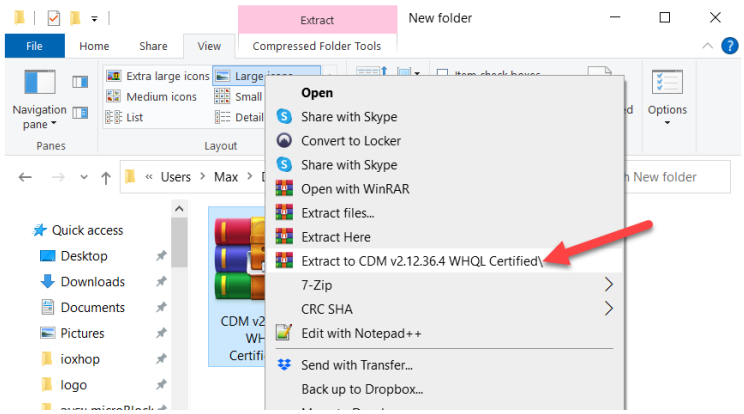
ติดตั้งไดร์เวอร์ FT231X

บอร์ด IOXESP32+ เป็นบอร์ดที่ใช้พัฒนา ESP32 เสียบอยู่บน Farm1 มีวงจรอัพโหลดโปรแกรมบนบอร์ดผ่านชิปแปลง USB เป็น UART เบอร์ FT231X ซึ่งจำเป็นต้องติดตั้งไดร์เวอร์ก่อนจึงจะอัพโหลดโปรแกรมเข้าได้

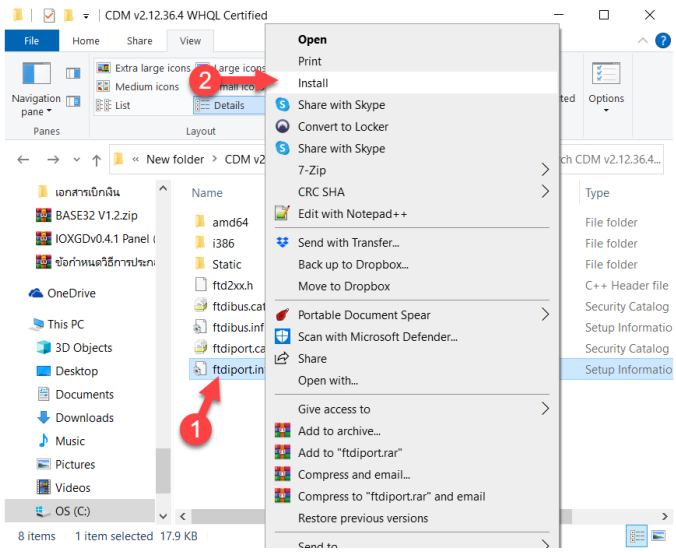
เข้าไปที่ https://ftdichip.com/drivers/vcp-drivers/ กดดาวน์โหลดไฟล์ตาม OS ของเครื่องคอมพิวเตอร์



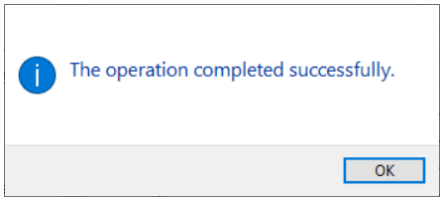
จะได้ไฟล์มา CDM vxxxxx WHQL Certified.zip ให้แตกไฟล์ด้วยโปรแกรม WinRAR, 7-Zip



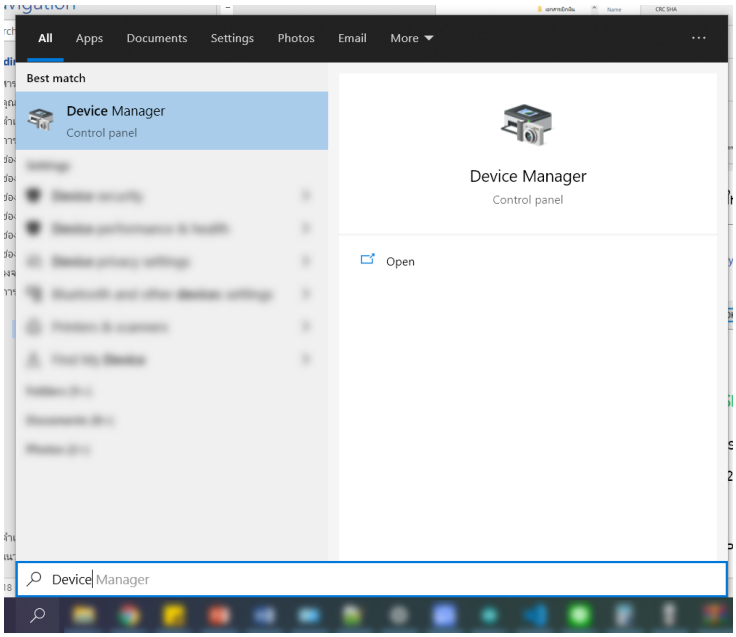
เข้าไปในโฟลเดอร์ที่แตกไฟล์มา แล้วคลิกขวาที่ไฟล์ ftdiport.inf กด Install



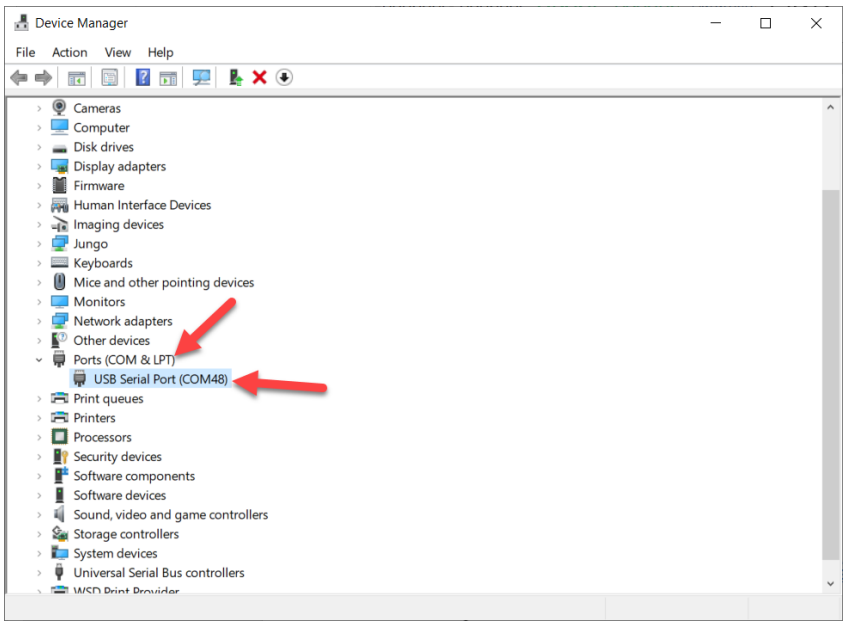
จากนั้นหน้าต่างให้ยืนยันจะแสดงขึ้นมา ให้กดปุ่ม Yes หากติดตั้งสำเร็จ จะมีข้อความแจ้ง (ดังรูป)



เปิด Device Manager ขึ้นมา โดยพิมพ์ในช่องค้นหา



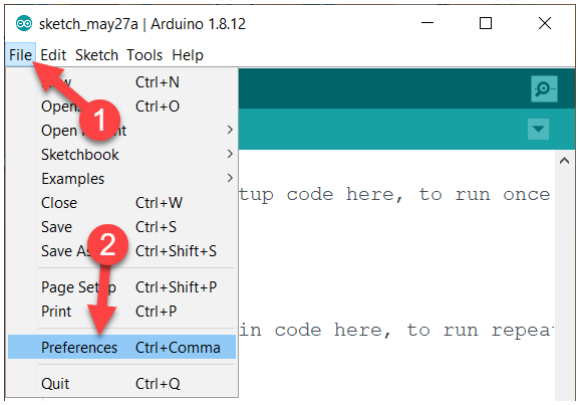
เสียบสาย USB ตามหัวข้อ [การเชื่อมต่อ Farm1 เข้ากับเครื่องคอมพิวเตอร์] จากนั้นสังเกตในหน้าต่าง Device Manager จะต้องมีหัวข้อ Ports และมี USB Serial Port ขึ้น เป็นอันเสร็จสิ้นการติดตั้งไดร์เวอร์



การเพิ่มบอร์ด ESP32 ลงโปรแกรม Arduino IDE

โปรแกรม Arduino IDE เดิมรองรับการเขียนโปรแกรมสั่งงานเฉพาะบอร์ดที่ผลิตโดย Arduino เท่านั้น หากต้องการใช้เขียนโปรแกรมให้ ESP32 จำเป็นต้องติดตั้งบอร์ด ESP32 เพิ่ม โดยติดตั้ง Arduino core for the ESP32

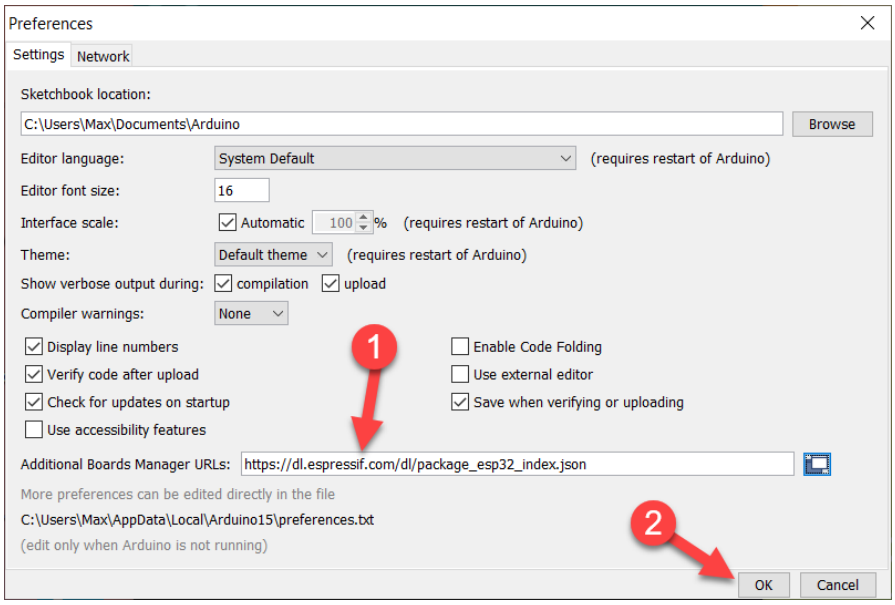
เปิดโปรแกรม ArduinoIDE ขึ้นมา โดยดับเบิลคลิกที่ไอค่อน Arduino บนหน้าเดสท็อป กด File เลือก Preferences



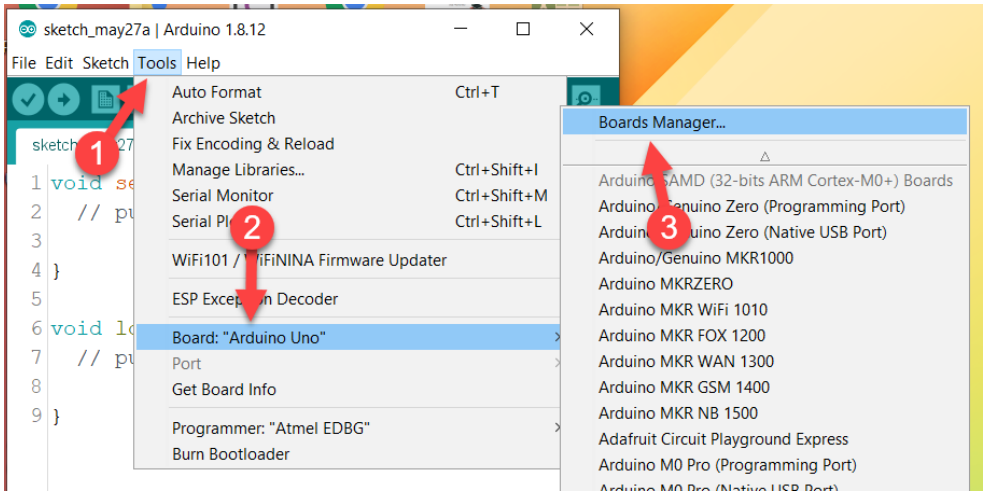
ในช่อง Additional Boards Manager URLs ใส่

https: / / raw. githubusercontent. com/ espressif/ arduino- esp32/ gh-pages/package_esp32_index.json

แล้วกดปุ่ม OK (ในรูปเป็นลิ้งเก่า ให้ใส่ลิ้งใหม่ด้านบน)

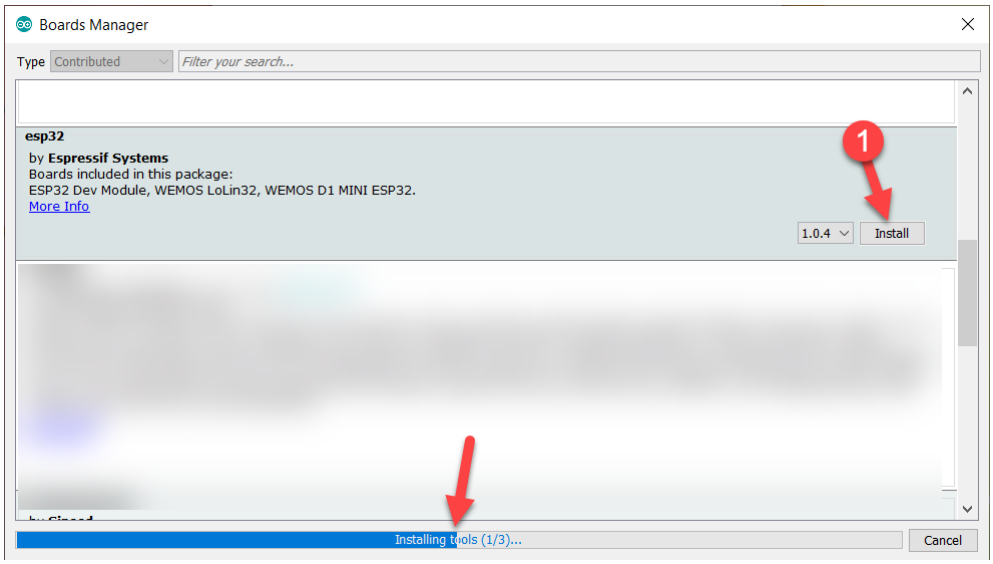


จากนั้นกด Tools > Board เลือก Boards Manager...



จากนั้นกด Type เลือก Contributed

เลื่อนหา esp32 แล้วกดปุ่ม Install

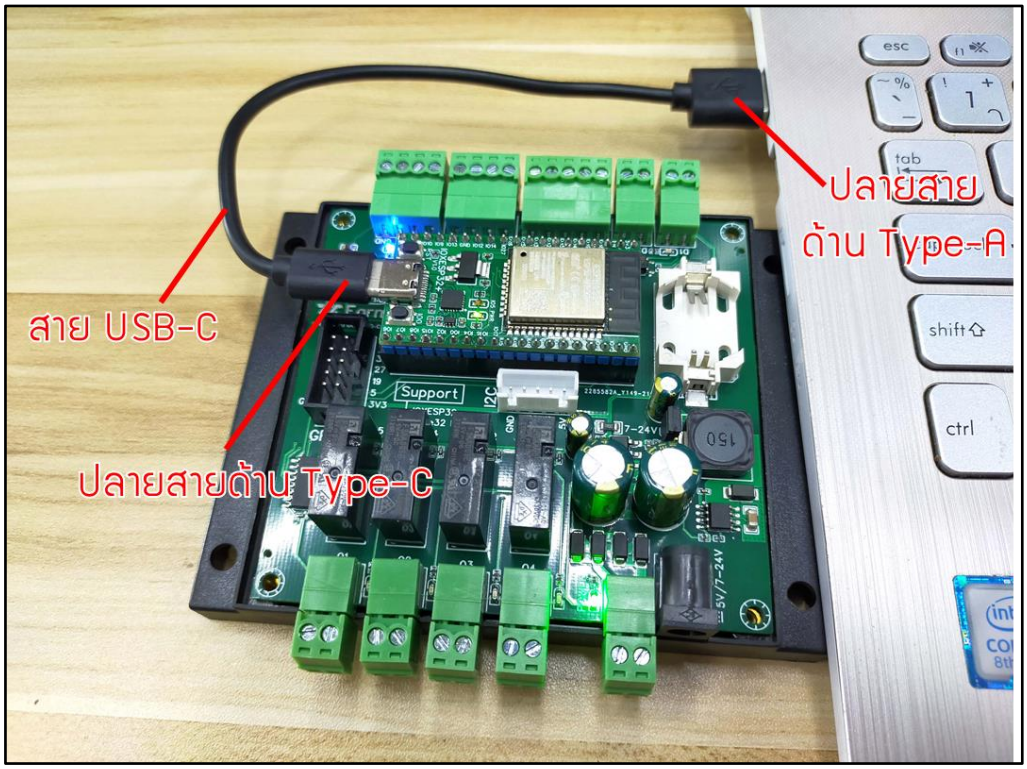


เมื่อติดตั้งเสร็จแล้ว จะแสดงคำว่า INSTALLED กดปุ่ม Close เพื่อปิดหน้าต่าง เป็นอันเสร็จสิ้นการเพิ่มบอร์ด ESP32 ลงโปรแกรม Arduino IDE

การเชื่อมต่อ Farm1 เข้ากับเครื่องคอมพิวเตอร์

ใช้สาย USB-C เสียบช่อง USB-C บนบอร์ด IOXESP32+ ปลายอีกด้านของสายเสียบเข้ากับเครื่องคอมพิวเตอร์

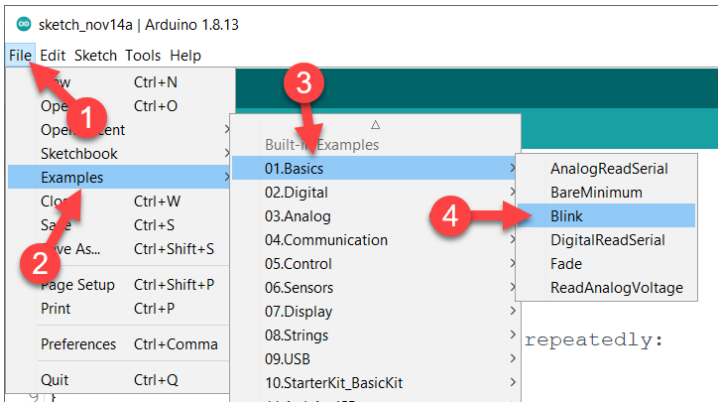
■ สาย USB-C หาซื้อได้ทั่วไปตามร้านขายอุปกรณ์ IT, ร้านโทรศัพท์, ร้านสะดวกซื้อ, ห้างสรรพสินค้า



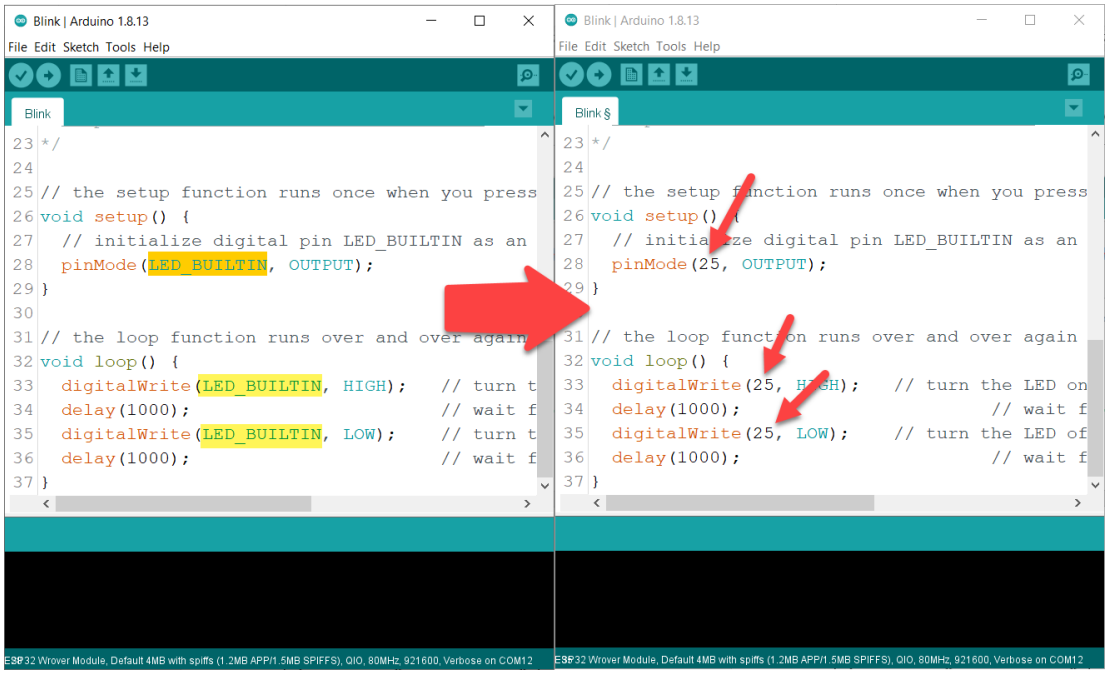
รูปที่ 24 การเชื่อมต่อ Farm1 เข้ากับเครื่องคอมพิวเตอร์ด้วยสาย USB-C

ทดสอบอัพโหลดโปรแกรมไฟกระพริบ

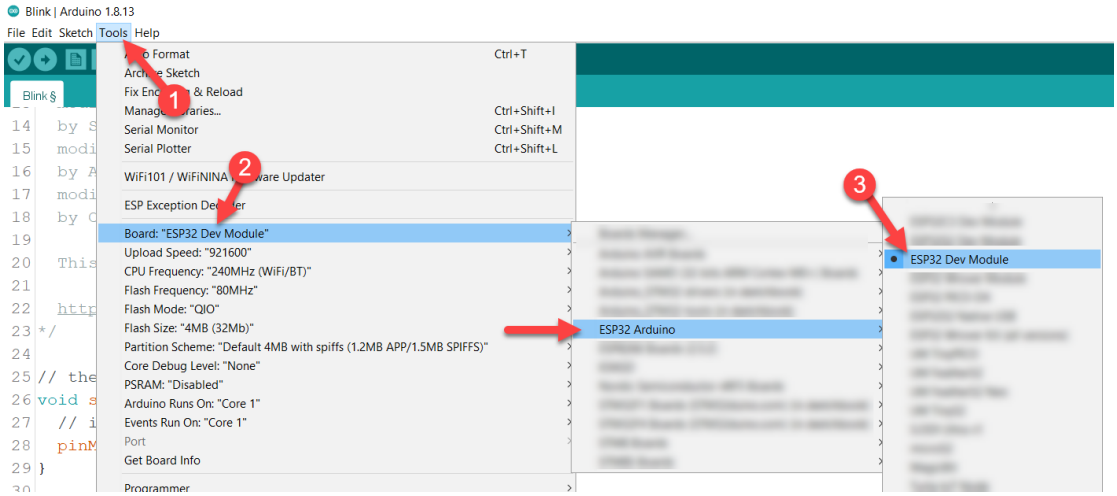
ที่โปรแกรม Arduino IDE กดไปที่ File > Examples > 01.Basics เลือก Blink



แก้ไข LED_BUILIN ให้เป็น 25 (ขา 01)

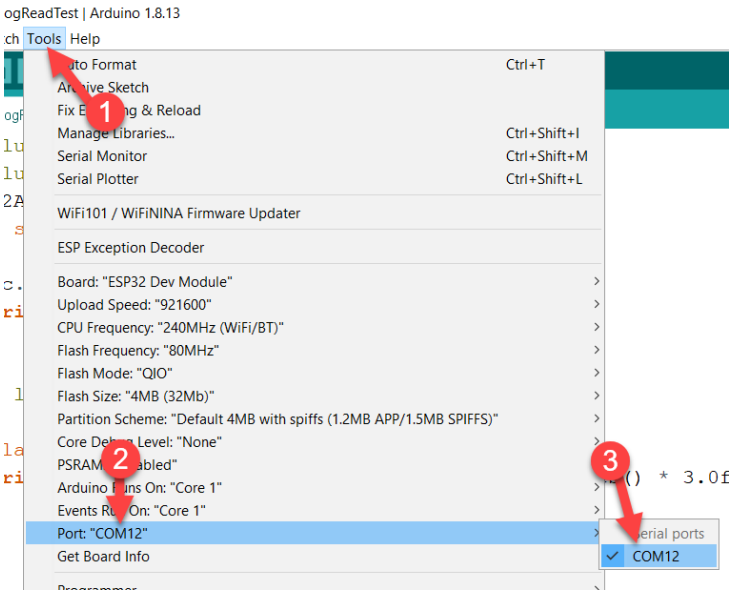


เลือกบอร์ดให้ถูกต้อง โดยกด Tool > Board > ESP32 Arduino เลือกบอร์ด ESP32 Dev Module

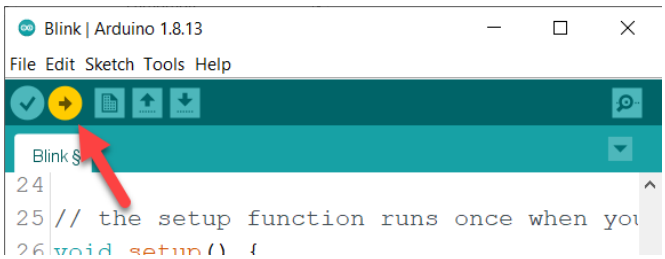


เลือกพอร์ตให้ถูกต้อง โดยกดไปที่ Tool > Port เลือก COM... ที่มีให้เลือก

■ หากไม่ขึ้นพอร์ตให้เลือก ให้ตรวจสอบสาย USB-C ว่าเสียบแน่นแล้วหรือยัง โดยสังเกตบอร์ด IOXESP32+ ไฟ PWR สีเขียวต้องติดสว่าง หากคาดว่าบอร์ดทำงานปกติ ให้ตรวจสอบการติดตั้งไดร์เวอร์ FT231Xอีกครั้ง



กดปุ่ม เพื่อคอมไพล์และอัพโหลดโปรแกรมลงบอร์ด (เวลาที่ใช้ในการคอมไพล์และอัพโหลดอาจถึง 2 นาที หากไม่มีข้อความแจ้ง Error หรือ Done uploading ให้รอก่อน)



หากคอมไพล์และอัพโหลดโปรแกรมสำเร็จ จะมีข้อความ Done uploading แสดงขึ้นมา (ดังรูป)

ผลลัพธ์ที่ได้ ไฟที่ช่อง 01 จะกระพริบทุก ๆ 1 วินาที พร้อมมีเสียงรีเลย์ตัด-ต่อ

เขียนโปรแกรมสั่งงานช่องเอาต์พุต

ใช้คำสั่ง pinMode() และ digitalWrite() ในการเขียนโปรแกรมสั่งงานช่องเอาต์พุต

กำหนดแทนที่ O1_PIN, ..., O4_PIN ด้วยหมายเลขขาที่ต่อกับ ESP32 ดังนี้

```
#define O1_PIN 25
#define O2_PIN 14
#define O3_PIN 12
#define O4_PIN 13
```

ใน void setup() ใช้สั่ง pinMode() กำหนดขา O1 ถึง O4 เป็นเอาต์พุต

```
void setup(){
  pinMode(O1_PIN, OUTPUT);
  pinMode(O2_PIN, OUTPUT);
  pinMode(O3_PIN, OUTPUT);
  pinMode(O4_PIN, OUTPUT);
}
```

ใน void loop() ใช้คำสั่ง digitalWrite() สั่งให้รีเลย์ช่อง O1 ตัด-ต่อทุก ๆ 1 นาที

```
void loop(){
  digitalWrite(O1_PIN, HIGH);
  delay(1000);
  digitalWrite(O1_PIN, LOW);
  delay(1000);
}
```

ได้โค้ดโปรแกรมที่สมบูรณ์ดังนี้

```
#define O1_PIN 25
#define O2_PIN 14
#define O3_PIN 12
#define O4_PIN 13

void setup(){
  pinMode(O1_PIN, OUTPUT);
  pinMode(O2_PIN, OUTPUT);
  pinMode(O3_PIN, OUTPUT);
  pinMode(O4_PIN, OUTPUT);
}

void loop(){
  digitalWrite(O1_PIN, HIGH);
  delay(1000);
  digitalWrite(O1_PIN, LOW);
  delay(1000);
}
```

อัพโหลดโปรแกรมลงบอร์ด จะได้ยินเสียงรีเลย์ตัด-ต่อ และไฟช่องเอาต์พุตติด-ดับ ทุก ๆ 2 วินาที

เขียนโปรแกรมอ่านช่องอินพุตดิจิทัล

ใช้คำสั่ง pinMode() และ digitalRead() ในการอ่านค่าช่องอินพุตดิจิทัล

กำหนดแทนที่ O1_PIN, ..., O4_PIN และ D1_PIN, D2_PIN ด้วยหมายเลขขาที่ต่อกับ ESP32 ดังนี้

```
#define O1_PIN 25
#define O2_PIN 14
#define O3_PIN 12
#define O4_PIN 13

#define D1_PIN 34
#define D2_PIN 34
```

ใน void setup() ใช้สั่ง pinMode() กำหนดขา O1 ถึง O4 เป็นเอาต์พุต และกำหนดขา D1 และ D4 เป็นอินพุตดิจิทัล

```
void setup() {
  pinMode(O1_PIN, OUTPUT);
  pinMode(O2_PIN, OUTPUT);
  pinMode(O3_PIN, OUTPUT);
  pinMode(O4_PIN, OUTPUT);

  pinMode(D1_PIN, INPUT);
  pinMode(D2_PIN, INPUT);
}
```

ใน void loop() ใช้คำสั่ง digitalRead() อ่านค่าเก็บลงตัวแปร value ชนิด int

```
 int value = digitalRead(D1_PIN);
```

ใช้คำสั่ง if ตรวจสอบค่าในตัวแปร value หากได้ 0 ให้สั่งรีเลย์ช่อง O1 ทำงาน ใช้คำสั่ง else หากเงื่อนไขใน if ไม่เป็นจริง ให้รีเลย์ช่อง O1 หยุดทำงาน

```
 if(value == 0){
    digitalWrite(O1_PIN, HIGH);
  } else {
    digitalWrite(O1_PIN, LOW);
  }
```

ได้โค้ดโปรแกรมที่สมบูรณ์ดังนี้ (ดาวน์โหลดโค้ดโปรแกรมได้ที่)

```
#define O1_PIN 25
#define O2_PIN 14
#define O3_PIN 12
#define O4_PIN 13

#define D1_PIN 34
#define D2_PIN 34

void setup() {
  pinMode(O1_PIN, OUTPUT);
  pinMode(O2_PIN, OUTPUT);
  pinMode(O3_PIN, OUTPUT);
  pinMode(O4_PIN, OUTPUT);

  pinMode(D1_PIN, INPUT);
  pinMode(D2_PIN, INPUT);
}

void loop() {
 int value = digitalRead(D1_PIN);
 if(value == 0) {
    digitalWrite(O1_PIN, HIGH);
  } else {
    digitalWrite(O1_PIN, LOW);
  }
  delay(10);
}
```

อัพโหลดโปรแกรมลงบอร์ด ที่ช่อง D1 ด้านนึงต่อเข้ากับ GND อีกด้านใช้สายลอย (ดังรูปที่ ...) แตะกับ 5V หรือ VIN สังเกตไฟสีส้มของช่อง D1 จะติดเมื่อแตะโดน และจะดับเมื่อนำออก ส่วนรีเลย์ช่อง O1 จะทำงานเมื่อแตะสายไฟ และหยุดทำงานเมื่อนำสายไฟออก

เขียนโปรแกรมอ่านช่องอินพุตแอนะล็อก

ใช้คำสั่ง analogRead() อ่านค่าช่องอินพุตแอนะล็อก ได้ค่า 0 ถึง 4095 เมื่อ 4095 เทียบได้กับแรงดัน 9.9V (ตามวงจรในหัวข้อ ช่องอินพุตแอนะล็อก)

กำหนดแทนที่ A1 และ A2 ด้วย 36 และ 39 ตามลำดับ

```
#define A1_PIN 36
#define A2_PIN 39
```

ใน void loop() ใช้คำสั่ง analogRead() อ่านค่าแอนะล็อกจาก A0 แล้วนำค่าที่ได้เทียบบัญญัติไตรยางค์เพื่อแปลงเป็นแรงดัน เก็บลงตัวแปร volt ชนิด float

```
1.  float volt=analogRead(A1_PIN)/4095.0*9.9;
```

ใช้คำสั่ง Serial.println() ให้แสดงค่าในตัวแปร volt ออกมาทาง Serial Monitor

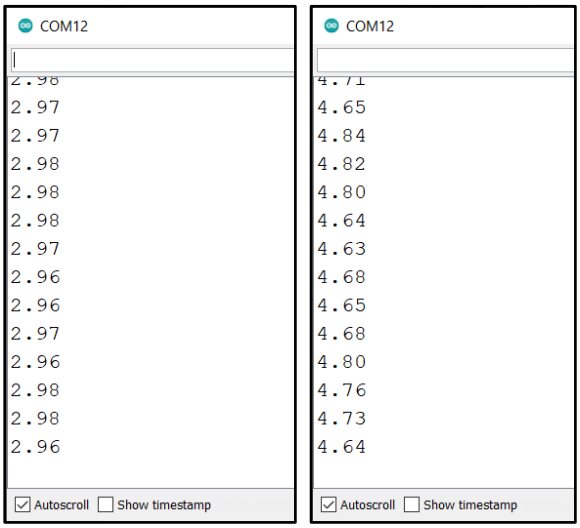
```
2.  Serial.println(volt);
```

ได้โค้ดโปรแกรมที่สมบูรณ์ดังนี้

```
3.  #define A1_PIN 36
4.  #define A2_PIN 39
5.
6.  void setup(){
7.   Serial.begin(115200);
8.  }
9.
10. void loop(){
11.  float volt=analogRead(A1_PIN)/4095.0*9.9;
12.  Serial.println(volt);
13.    delay(10);
14. }
```

อัพโหลดโค้ดโปรแกรมลงบอร์ด เปิด Serial Monitor โดยกด ตั้ง band เป็น 115200 แล้วทดลองจั้มระหว่าง A1 กับ 5V จากนั้นทดสอบจั้มระหว่าง A1 กับ 3.3V สังเกตผลที่ได้ใน Serial Monitor จะต่างกัน

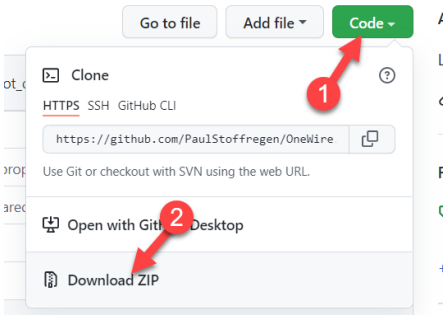
■ วงจรส่วนแอนะล็อกใช้ไฟจาก VIN จึงจำเป็นต้องต่อไฟเข้า VIN เพื่อให้วงจรแอนะล็อกทำงานได้ถูกต้อง



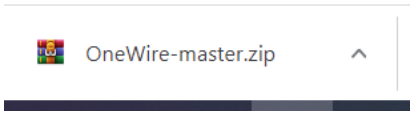
เขียนโปรแกรมช่อง I/O อ่านค่าอุณหภูมิจาก DS18B20

ต่อวงจรตามหัวข้อ ช่อง I/O ติดตั้งไลบารี่ OneWire ตามขั้นตอนดังนี้

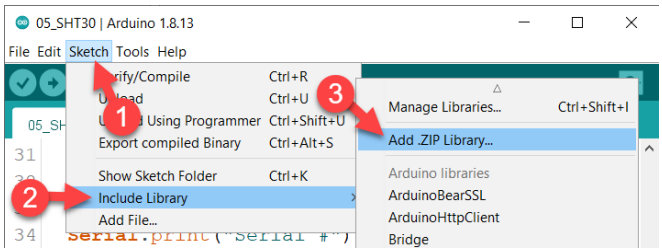
เข้าไปที่ https://github.com/PaulStoffregen/OneWire กด Code แล้วกด Download ZIP



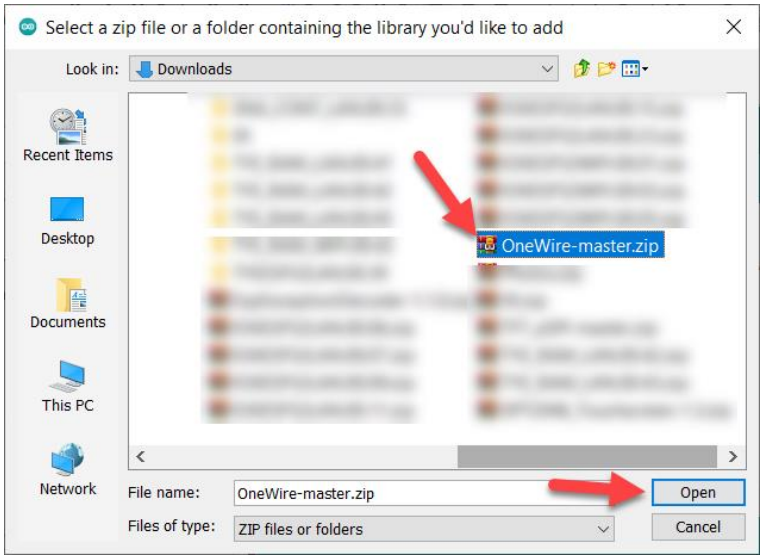
จะได้ไฟล์ OneWire-master.zip มา



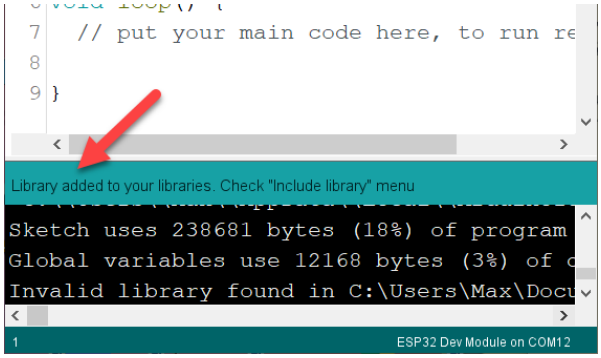
ที่โปรแกรม Arduino IDE กด Sketch > Include Library เลือก Add .ZIP Library



เลือกไฟล์ OneWire-master.zip ที่ดาวน์โหลดมา กดปุ่ม Open

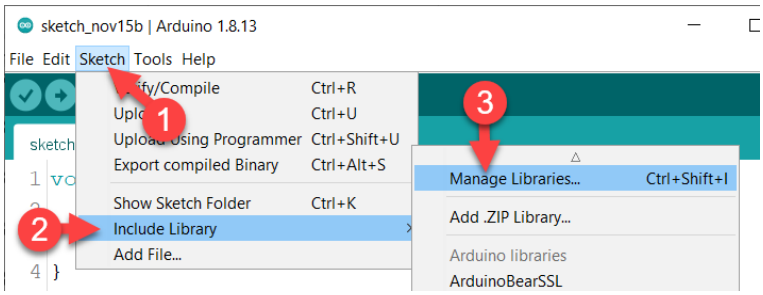


หากติดตั้งสำเร็จ จะมีข้อความ Library added ขึ้น (ดังรูป)

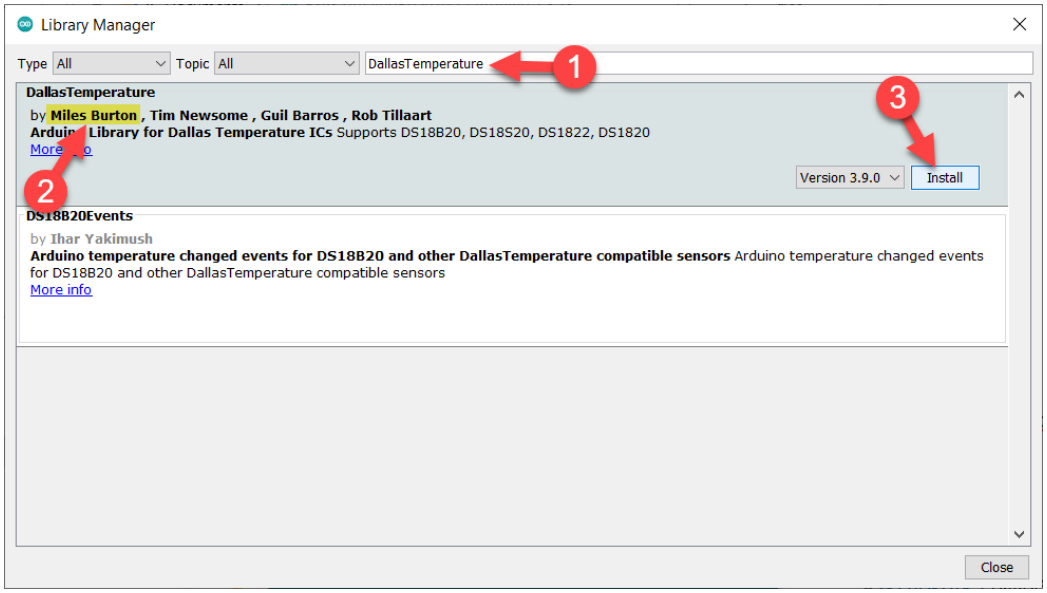


ติดตั้งไลบารี่ DallasTemperature ด้วย Library Manager ตามขั้นตอนดังนี้

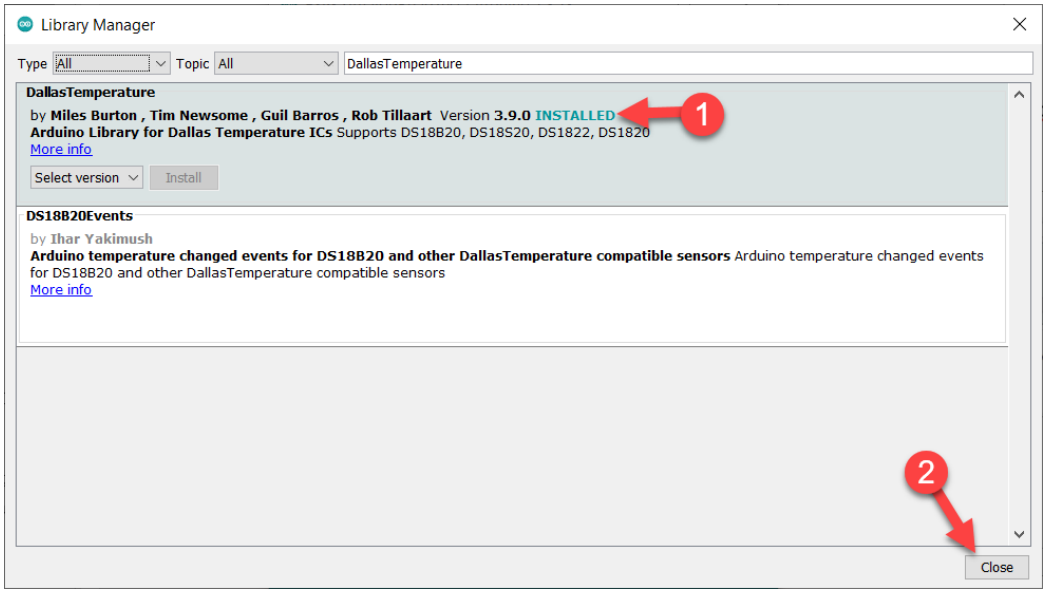
กด Sketch > Include Library เลือก Manage Libraries



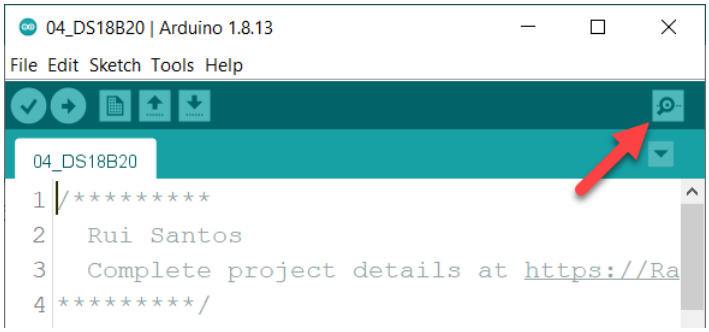
หน้าต่าง Library Manager จะแสดงขึ้นมา ให้ค้นหา DallasTemperature แล้วเลือกกดปุ่ม Install ไลบารี่ by Miles Burton



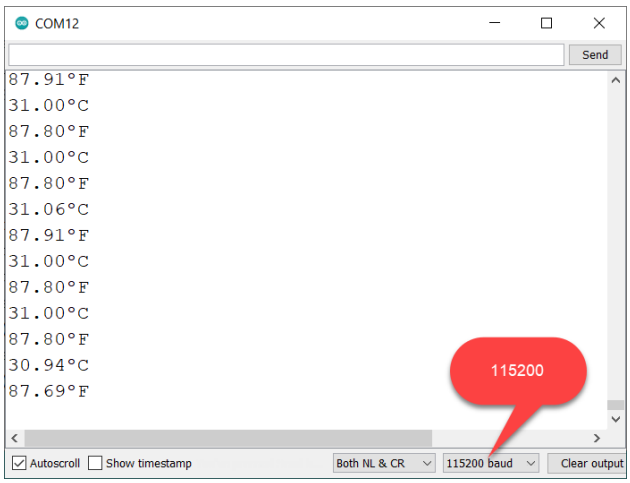
เมื่อติดตั้งเสร็จแล้ว จะมีข้อความ INSTALLED แสดง (ดังรูป) กดปุ่ม Close เพื่อปิดได้เลย



ดาวน์โหลดโปรแกรมตัวอย่างจาก https://bit.ly/3HkpJx5 อัพโหลดโปรแกรมลงบอร์ด แล้วเปิด Serial Monitor ขึ้นมา โดยกด

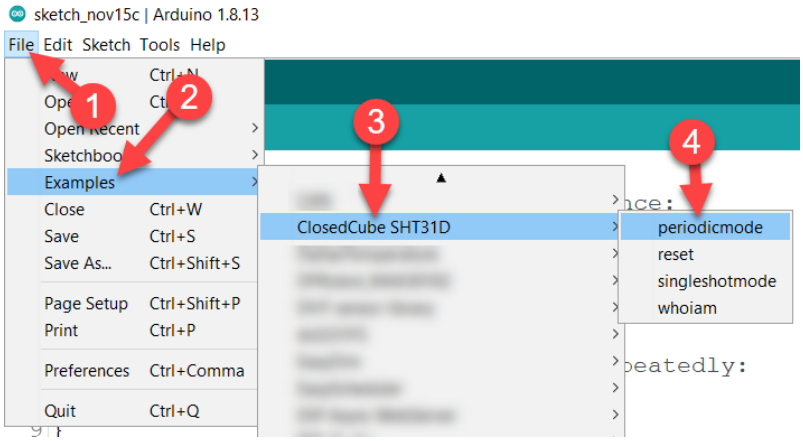


เลือก 115200 band หากต่อวงจรถูกต้อง เซ็นเซอร์ทำงานได้ จะมีอุณหภูมิในหน่วยองศาเซลเซียส และองศาฟาเรนไฮต์แสดงขึ้นมา



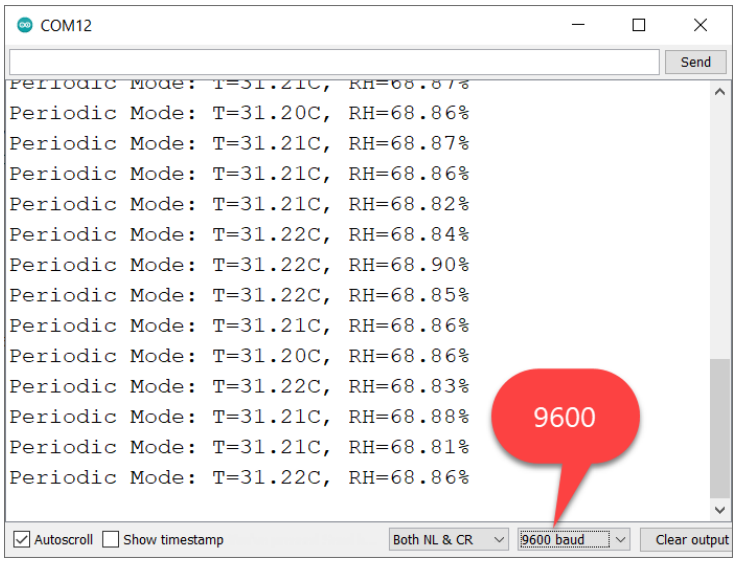
เขียนโปรแกรมช่อง I²C อ่านค่าอุณหภูมิและความชื้นจาก SHT30

ต่อวงจรตามหัวข้อ ช่อง I2C ติดตั้งไลบารี่ ClosedCube SHT31D ด้วย Library Manager จากนั้นเปิดโค้ดโปรแกรมตัวอย่าง periodicmode ขึ้นมา



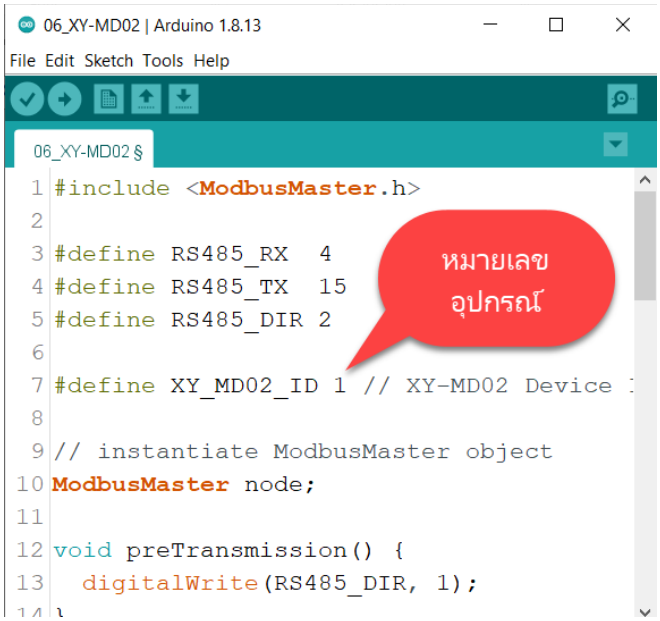
อัพโหลดโปรแกรมลงบอร์ด เปิด Serial Monitor ขึ้นมา ปรับ band เป็น 9600 ค่าอุณหภูมิและความชื้นที่ SHT30 วัดได้จะแสดงขึ้นมา

■ หากมีข้อความ Periodic Mode: [ERROR] ให้แก้ไขหมายเลขของอุปกรณ์บรรทัดที่ 32 จาก 0x44 เป็น 0x45 อัพโหลดโปรแกรมแล้วดูผลอีกครั้ง

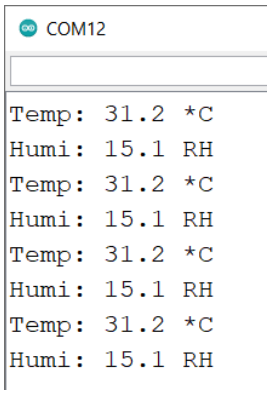


เขียนโปรแกรมช่อง RS485 อ่านค่าอุณหภูมิและความชื้นจาก XY-MD02

ต่อวงจรตามหัวข้อ ช่อง RS485 ติดตั้งไลบารี่ ModbusMaster ด้วย Library Manager จากนั้นดาวน์โหลดโค้ดตัวอย่างจาก https://bit.ly/3ccUxRV แก้ไขหมายเลขอุปกรณ์ (Device ID) ของ XY-MD02 ให้ถูกต้อง (ค่าจากโรงงาน เป็น ID: 1) อัพโหลดโปรแกรมลงบอร์ด

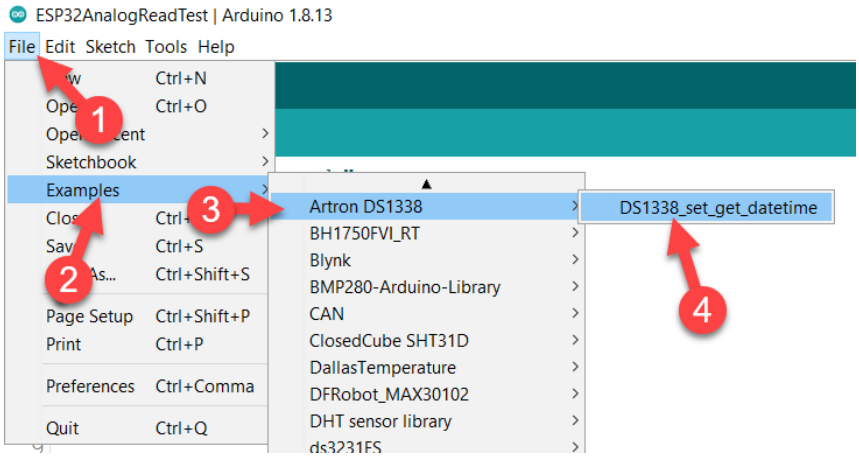


เปิด Serial Monitor ขึ้นมา ปรับ Band เป็น 115200 หากต่อวงจรถูกต้อง เซ็นเซอร์ทำงานได้ ค่าอุณหภูมิและความชื้นที่วัดได้จะแสดงขึ้นมา

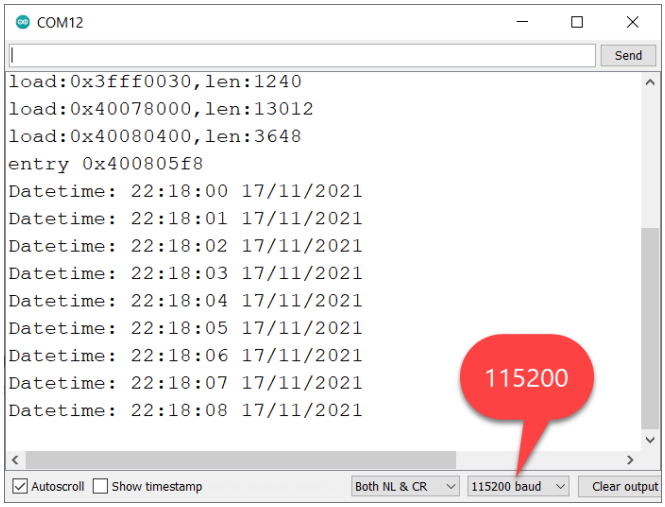


เขียนโปรแกรมตั้งเวลาและอ่านค่าเวลาจากวงจรนับเวลา (RTC)

ติดตั้งไลบารี่ Artron DS1338 ด้วย Library Manager เปิดตัวอย่างโค้ดจากเมนู File > Examples > Artron DS1338 > DS1338_set_get_datetime



อัพโหลดโปรแกรมตัวอย่างลงบอร์ด เปิด Serial Monitor ปรับ 115200 band ผลที่ได้คือ เวลาใน RTC จะถูกตั้ง และแสดงเวลาที่นับขึ้นเรื่อย ๆ บน Serial Monitor



คำแนะนำการติดตั้ง

อุปกรณ์รองรับการติดตั้งบนรางปีกนก ไม่กันน้ำ การใช้งานจริงจึงควรติดตั้งบนรางปีกนกในกล่องกันน้ำ รูร้อยสายไฟที่เข้ากล่องควรใช้คอนดนคเตอร์แบบกันน้ำ หรือใช้เคเบิ้ลแกลนด์ เพื่อป้องกันน้ำเข้าภายในกล่อง ควรรักษาอุณหภูมิให้ไม่เกิน 80 องศาเซลเซียส ซึ่งอาจทำให้อุปกรณ์อิเล็กทรอนิกส์เสียหาย การเก็บรักษาควรเก็บในที่แห้ง ไม่โดนแดดโดยตรง

พาวเวอร์ซัพพลายควรเลือกใช้รุ่นที่สามารถทนไฟกระชากได้ มีระบบตัดไฟเมื่อกระแสเกิน มีอุปกรณ์ป้องกันฟ้าผ่า โดยอาจสังเกตและเลือกใช้พาวเวอร์ซัพพลายที่มีสัญลักษณ์ CE หรือ มอก. พร้อมติดตั้งเบรกเกอร์เพื่อใช้ปิดอุปกรณ์เมื่อต้องบำรุงรักษา

แนวทางการแก้ไขปัญหา

ปัญหา	แนวทางแก้ไข
ปัญหาเอาต์พุต	
ใช้คำสั่ง digitalWrite() แต่รีเลย์ไม่ทำงาน ไฟเอาต์พุตไม่ติด	เป็นได้ทั้งปัญหาซอฟแวร์และปัญหาด้านฮาร์ดแวร์ • อาจลืมใช้คำสั่ง pinMode() หรือใช้ไม่ถูกต้อง ก่อนเรียกใช้ digitalWrite() • ใช้คำสั่ง digitalWrite() พิดขา • ESP32 รีเซ็ตก่อนทำงานถึงคำสั่ง digitalWrite() (ให้เปิด Serial Monitor ขึ้นมาดู) • บอร์ด IOXESP32+ หลวม ทำให้ไฟฟ้าไม่สามารถไหลได้ • กระแสไฟฟ้าจากแหล่งจ่ายไม่เพียงพอให้อุปกรณ์ทำงานได้ปกติ
ไฟติด-ดับ แต่ไม่มีเสียงรีเลย์ตัด-ต่อ	• รีเลย์เสีย
ปัญหาอินพุตดิจิทัล	
ใช้คำสั่ง digitalRead() ได้ค่าเป็น 1 หรือเป็น 0 ตลอดเวลา	• ให้สังเกตไฟตรงจุดเอาต์พุต หากต่อวงจรให้ต้องเป็น 1/0 แต่ไฟไม่ติด-ไม่ดับ เป็นปัญหาด้านฮาร์ดแวร์หรือระบบไฟฟ้า ให้ใช้มัลติมิเตอร์ตรวจสอบแรงดันแต่ละจุด • หากไฟติด-ดับปกติ ให้ตรวจสอบโค้ดโปรแกรมว่าอ่านออกมาถูกขา • ตรวจสอบว่าใช้คำสั่ง pinMode() ก่อนใช้คำสั่ง digitalRead() แล้ว

ปัญหาอินพุตแอนะล็อก	
วัดแรงดันได้สูงสุด 2V กว่า ๆ แม้จะป้อนแรงดันมากกว่านี้	เกิดจากไม่ได้จ่ายไฟเข้า VIN 7V ถึง 12V วงจรอินพุตแอนะล็อกใช้ไฟจาก VIN ในการทำงาน หากไม่จ่ายไฟ ค่าที่วัดได้จะไม่ถูกต้อง
วัดแรงดันได้ไม่ตรง	เป็นปัญหาในระดับชิป ให้ใช้ไลบารี่ ESP32AnalogRead อ่านแรงดันแทน จะให้ค่าที่ตรงความเป็นจริงมากขึ้น
ช่อง I^2C	
ติดต่อกับอุปกรณ์ I^2C ไม่ได้	สังเกตไฟช่อง I^2C ว่ากระพริบหรือไม่ หากกระพริบแสดงว่ามีการส่งข้อมูลออกจากบอร์ดแล้ว แต่อุปกรณ์ปลายทางอาจไม่ได้รับ หรือไม่ทำงาน หากไฟไม่กระพริบ แสดงว่าไม่มีข้อมูลส่งออกจากบอร์ด ให้ตรวจสอบโค้ดโปรแกรม
ช่อง RS485	
ติดต่อกับอุปกรณ์ RS485 ไม่ได้	สังเกตไฟช่อง RS485 ตรง TX ว่ากระพริบหรือไม่ หากกระพริบแสดงว่ามีการส่งข้อมูลออกจากบอร์ดแล้ว จากนั้นให้สังเกตไฟ RX ว่าทำงานอย่างไร • ไฟ RX กระพริบ แสดงว่าอุปกรณ์ปลายทางส่งข้อมูลเข้ามาแล้ว แต่ฟั่ง Farm1 รับข้อมูลไม่ได้ ให้ตรวจสอบโค้ดโปรแกรม • ไฟ RX ไม่กระพริบ แสดงว่าอุปกรณ์ปลายทางไม่ตอบสนอง อาจเป็นไปได้ว่าหมายเลข ID ของอุปกรณ์ที่ร้องขอข้อมูลไม่มีอยู่ หรือ ID ของอุปกรณ์ผิด ให้ใช้โปรแกรม CAS Modbus Scanner ลองสแกนหา ID ดูก่อน รวมทั้งอาจตรวจสอบการตั้งค่า Band Rate และ

	Prarity ด้วย ว่าตั้งค่าไว้ตรงกับที่อุปกรณ์ต้องการหรือไม่ หากไฟ TX ไม่กระพริบ • ยังส่งข้อมูลออกไม่ได้ ให้ตรวจสอบโค้ดโปรแกรม
ปัญหาวงจรนับเวลา	
เวลาไม่ถูกต้องหลังตัดไฟ	ไม่ได้ใส่แบตเตอรี่ หรือแบตเตอรี่หมด

บริษัท อาร์ทรอน ชอป จำกัด รับพัฒนาสินค้าอิเล็กทรอนิกส์ บอร์ดอิเล็กทรอนิกส์ ด้านระบบ IoT พัฒนาเว็บไซต์ระบบ IoT ด้วย ReactJS / Next.js รับทำระบบหลังบ้านให้อุปกรณ์ IoT รับเขียนเฟิร์มแวร์อุปกรณ์ IoT ด้วย Arduino IDE / PlatformIO / Atmel Studio / MPLAB IDE / STM32CubeIDE / IAR / ESP-IDF / Arm Keil รับจัดหาชิ้นส่วนอิเล็กทรอนิกส์ ออกใบเสนอราคาและใบกำกับภาษีได้

ติดต่อสอบถามข้อมูลเพิ่มเติม

- เว็บไซต์ www.artronshop.co.th
- โทร: 02 003 3688 (สำนักงาน)
- อีเมล: contact@atronshop.co.th
- LINE: @artronshop
- Facebook: ArtronShop
- Youtube: ArtronShop