



หุ่นยนต์เก็บฝุ่นอัตโนมัติ

Automatic dust-collecting robot

โดย

นางสาว สรินดา กันภัย รหัสนักศึกษา 67010927 สาขา 1 ปริญญา

นางสาว สุคนธ์ทิพย์ ตัวงวังหิน รหัสนักศึกษา 67010960 สาขา 1 ปริญญา

อาจารย์ที่ปรึกษา

อ. ไพศาล สิทธิโยภาสกุล

รศ. ดร. บุญยชนะ ภูระหงษ์

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา

01236249 FUNDAMENTAL OF DIGITAL SYSTEM DESIGN

หลักสูตร วิศวกรรมคอมพิวเตอร์ สาขา วิศวกรรมระบบไอโอทีและสารสนเทศ

ประจำภาคการศึกษาที่ 2 ปีการศึกษา 2567

ชื่อโครงการ	ภาษาไทย หุ่นยนต์เก็บฝุ่นอัตโนมัติ
ภาษาอังกฤษ	Automatic dust-collecting and floor-mopping robot
คณะจัดทำ	นางสาว สรินดา กันภัย รหัสนักศึกษา 67010927 สาขา 1 ปริญญา นางสาว สุคนธ์ทิพย์ ดั่งวงษ์ รหัสนักศึกษา 67010960 สาขา 1 ปริญญา
อาจารย์ที่ปรึกษา	อ. ไพศาล สิทธิโยภาสกุล รศ. ดร. บุญยชนะ ภูระหงษ์
ภาคการศึกษา	2/2567

บทคัดย่อ

โครงการ "หุ่นยนต์เก็บฝุ่นอัตโนมัติ" เป็นโครงการที่มุ่งเน้นการพัฒนาอุปกรณ์อัจฉริยะสำหรับทำความสะอาดพื้นโดยอัตโนมัติ โดยอาศัยเทคโนโลยี Internet of Things (IoT) และระบบคอมพิวเตอร์ฝังตัว (Embedded Systems) เพื่อเพิ่มประสิทธิภาพในการทำความสะอาดและลดภาระของผู้ใช้งาน หุ่นยนต์นี้ถูกออกแบบมาให้สามารถเคลื่อนที่ได้เอง ปรับเปลี่ยนทิศทางตามสภาพแวดล้อม และเชื่อมต่อกับระบบควบคุมผ่านอินเทอร์เน็ต ทำให้สามารถใช้งานได้ง่ายและสะดวกมากขึ้น

การพัฒนาโครงการนี้มีเป้าหมายเพื่อสร้างอุปกรณ์ทำความสะอาดอัตโนมัติที่สามารถทำงานได้อย่างมีประสิทธิภาพ ลดการใช้แรงงานมนุษย์ และเพิ่มความสะอาดสบายให้กับผู้ใช้งาน โดยเฉพาะผู้ที่มีข้อจำกัดทางร่างกายหรือที่ต้องการลดภาระในการทำความสะอาดบ้าน หุ่นยนต์นี้สามารถควบคุมและติดตามสถานะผ่านแอปพลิเคชันหรือเว็บอินเตอร์เฟซ นอกจากนี้ยังใช้เทคโนโลยีเซนเซอร์ในการตรวจจับสิ่งกีดขวางเพื่อหลบหลีกอัตโนมัติ

กิตติกรรมประกาศ

โครงการ "หุ่นยนต์ทำความสะอาดพื้นอัตโนมัติ" นี้สำเร็จลุล่วงตามวัตถุประสงค์ของผู้จัดทำได้ด้วยความช่วยเหลือและการสนับสนุนจากหลายฝ่าย ข้าพเจ้าขอขอบพระคุณอาจารย์ที่ปรึกษาที่ให้คำแนะนำ องค์กรความรู้ และข้อเสนอแนะที่เป็นประโยชน์ตลอดกระบวนการดำเนินโครงการ นอกจากนี้ ขอขอบคุณเพื่อนร่วมทีมที่ร่วมแรงร่วมใจและทุ่มเทความสามารถในการพัฒนาโครงการให้สำเร็จลุล่วง

คณะผู้จัดทำขอขอบพระคุณอาจารย์สำนักวิชาวิศวกรรมศาสตร์ทุกท่าน ที่ประสิทธิ์ประสาทความรู้แก่ศิษย์ ขอขอบคุณบิดามารดาที่คอยให้การสนับสนุนให้กำลังใจและขอขอบคุณ เพื่อนๆทุกคน รวมถึงบุคคลอื่นที่ไม่ได้กล่าวไว้ ณ ที่นี้ที่ให้ความช่วยเหลือในทุกๆ ด้านด้วยดีมาโดย ตลอด จนสำเร็จเป็นโครงงานฉบับนี้

นางสาว สรินดา กันภัย

นางสาว สุคนธ์ทิพย์ ดั่งวงษ์หิน

สารบัญ

รายละเอียดเกี่ยวกับโครงงาน	ก
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
บทที่1 บทนำ	1
1.1 วัตถุประสงค์ของโครงงาน	1
1.2 ขอบเขตของการดำเนินงาน	2
1.3 ขั้นตอนการดำเนินงาน	2
1.4 สถานที่ดำเนินงาน	4
1.5 ผลที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 โปรแกรม Arduino	5
2.2 Node – Red	10
2.3 MQTT (Message Queuing Telemetry Transport)	13
2.4 Blynk	14
2.5 L298N Motor Driver	16
2.6 Gear Motor	17
2.7 Ultrasonic Sensor	18
2.8 Micro Servo Tower Pro SG90	19
2.9 DHT11	21
2.10 Battery Alkaline 9 V	22
บทที่3 วิธีการดำเนินงาน	23
3.1 การพัฒนาระบบหุ่นยนต์เก็บฝุ่นอัตโนมัติ	23
3.2 วิธีการดำเนินงาน	24
3.3 อุปกรณ์	25
3.4 หลักการทำงาน	27
3.4 การต่อวงจร	28
บทที่ 4 ผลการดำเนินงาน	29
4.1 ผลการดำเนินงาน	29

4.2 รูปโมเดล	29
4.3 dashboard	30
บทที่ 5 สรุปผลการดำเนินงาน	31
5.1 สรุปผลการดำเนินงาน	31
5.2 ข้อเสนอแนะ	31
บรรณานุกรม	32

บทที่1

บทนำ

ในปัจจุบัน เทคโนโลยีหุ่นยนต์และระบบอัตโนมัติได้รับการพัฒนาอย่างต่อเนื่องและมีบทบาทสำคัญในชีวิตประจำวัน โดยเฉพาะในด้านการทำความสะอาดบ้านและสถานที่ทำงาน ด้วยไลฟ์สไตล์ของผู้คนที่เปลี่ยนไป และเวลาที่จำกัด ทำให้ความต้องการอุปกรณ์ช่วยทำความสะอาดที่มีประสิทธิภาพและใช้งานสะดวกเพิ่มสูงขึ้น หุ่นยนต์ทำความสะอาดพื้นอัตโนมัติจึงเป็นทางเลือกที่ได้รับความนิยม เนื่องจากช่วยลดภาระของผู้ใช้และเพิ่มคุณภาพชีวิตได้อย่างมีประสิทธิภาพ

โครงการ " หุ่นยนต์เก็บฝุ่นอัตโนมัติ " มีเป้าหมายในการพัฒนาหุ่นยนต์ทำความสะอาดที่สามารถทำงานได้อย่างอิสระ โดยใช้เทคโนโลยี Internet of Things (IoT) และระบบคอมพิวเตอร์ฝังตัว (Embedded Systems) เพื่อเพิ่มความสามารถในการควบคุมและตรวจสอบสถานะการทำงานแบบเรียลไทม์ หุ่นยนต์นี้ถูกออกแบบให้สามารถเคลื่อนที่ได้เอง หลีกสิ่งกีดขวาง และทำความสะอาดพื้นได้อย่างมีประสิทธิภาพ

โครงการนี้มุ่งเน้นการสร้างนวัตกรรมที่สามารถช่วยลดภาระของมนุษย์ในงานบ้าน อีกทั้งยังเป็นแนวทางในการศึกษาและพัฒนาาระบบอัจฉริยะที่สามารถนำไปประยุกต์ใช้ในงานด้านอื่น ๆ เช่น หุ่นยนต์ทำความสะอาดในสำนักงาน โรงพยาบาล หรือพื้นที่สาธารณะ ด้วยเหตุนี้ การพัฒนาหุ่นยนต์ทำความสะอาดอัตโนมัติในโครงการนี้จึงเป็นก้าวสำคัญในการนำเทคโนโลยีมาปรับใช้เพื่อเพิ่มความสะดวกสบายและคุณภาพชีวิตของผู้ใช้งาน

1.1 วัตถุประสงค์ของโครงการ

1.1.1 เพื่อลดภาระงานบ้านและเพิ่มความสะดวกสบายในชีวิตประจำวัน

- หุ่นยนต์สามารถทำความสะอาดพื้นได้โดยอัตโนมัติ ลดการใช้แรงงานคนในการกวาดฝุ่น
- ประหยัดเวลา เนื่องจากผู้ใช้งานเพียงแค่เปิดใช้งานหรือปล่อยให้หุ่นยนต์ทำงานโดยอัตโนมัติ หุ่นยนต์สามารถทำความสะอาดพื้นได้อย่างหมดจด

1.1.2 เพื่อเพิ่มประสิทธิภาพในการทำความสะอาดและความครอบคลุมของพื้นที่

- หุ่นยนต์สามารถหลบสิ่งกีดขวางได้อย่างชาญฉลาด

- สามารถทำความสะอาดได้ในจุดที่เข้าถึงยาก เช่น ใต้โซฟา ใต้เตียง หรือมุมแคบ

1.1.3 ศึกษาและพัฒนาการใช้ Blink ในการควบคุมหุ่นยนต์ผ่านระบบ IoT

1.2 ขอบเขตของการดำเนินงาน

1.2.1 การดูฝุ่น:

- หุ่นยนต์สามารถดูฝุ่นและสิ่งสกปรกจากพื้นผิวต่างๆ เช่น พื้นไม้ กระเบื้อง หรือพรมได้อย่างมีประสิทธิภาพ โดยมีเซ็นเซอร์ตรวจจับสิ่งสกปรกและกำหนดเส้นทางการทำความสะอาดเอง

1.2.2 การตรวจจับและหลบหลีกสิ่งกีดขวาง:

- ใช้เซ็นเซอร์ในการตรวจจับสิ่งกีดขวาง เช่น เฟร์นิเจอร์ หรือสายไฟ เพื่อหลีกเลี่ยงการชนและสามารถเดินทางไปยังพื้นที่ที่ต้องการทำความสะอาดได้อย่างมีประสิทธิภาพ

1.3 ขั้นตอนการดำเนินงาน

เป็นออกเป็นทั้งหมด 3 ขั้นตอนคือ

1.3.1 การศึกษาทฤษฎีและองค์ประกอบของระบบ

- Arduino UNO R4 Wi-Fi: ทำความเข้าใจกับโครงสร้างบอร์ด การใช้งาน Wi-Fi และการเชื่อมต่อกับเซ็นเซอร์
- L298N Drive Motor Module: ศึกษาการทำงานของมอเตอร์ไดรเวอร์สำหรับควบคุมมอเตอร์ของหุ่นยนต์
- Ultrasonic Sensor: ศึกษาการใช้เซ็นเซอร์อัลตราโซนิกเพื่อตรวจจับสิ่งกีดขวางและคำนวณระยะทาง
- มอเตอร์และล้อขับเคลื่อน: วิเคราะห์ประเภทของมอเตอร์ที่เหมาะสมสำหรับการเคลื่อนที่ของหุ่นยนต์
- ระบบทำความสะอาด (เก็บฝุ่น)
- DHT11 ศึกษาการทำงานการวัดอุณหภูมิและความชื้น

1.3.2 การออกแบบและประกอบหุ่นยนต์เก็บฝุ่นอัตโนมัติ

- ออกแบบตัวหุ่นยนต์โดยใช้วัสดุที่เหมาะสม โดยได้เลือกใช้ไม้พลาสติกความหนา 3 มิล
- วางตำแหน่งของมอเตอร์, ล้อ, เซ็นเซอร์, และระบบดูดฝุ่นเพื่อให้มีประสิทธิภาพสูงสุด
- กำหนดพื้นที่สำหรับแบตเตอรี่และบอร์ดควบคุม
- ติดตั้งมอเตอร์และล้อ
- ติดตั้งเซ็นเซอร์อัลตราโซนิกบริเวณด้านหน้าของหุ่นยนต์เพื่อใช้ในการตรวจจับสิ่งกีดขวาง
- เชื่อมต่อบอร์ดควบคุม (Arduino UNO R4 Wi-Fi) กับอุปกรณ์อื่นๆ
- ติดตั้งลูกกลิ้งเก็บฝุ่น

1.3.3 การเขียนโค้ดและเชื่อมต่อบอร์ดใน Arduino UNO R4 Wi-Fi

1.3.3.1 การเขียนโค้ดควบคุมหุ่นยนต์

- เขียนโปรแกรมควบคุมการเคลื่อนที่ของมอเตอร์ผ่าน L298N
- เขียนโค้ดสำหรับเซ็นเซอร์อัลตราโซนิกเพื่อหลบหลีกสิ่งกีดขวาง
- เขียนโค้ดสำหรับควบคุมระบบดูดฝุ่น
- ตั้งค่าการเชื่อมต่อ Wi-Fi และส่งข้อมูลผ่านเครือข่าย

1.3.3.2 การเชื่อมต่อกับแอปพลิเคชัน Blynk

- เชื่อมต่อ Arduino UNO R4 Wi-Fi กับแพลตฟอร์ม Blynk
- ตั้งค่า Dashboard ใน Blynk เพื่อให้ผู้ใช้สามารถควบคุมหุ่นยนต์ผ่านสมาร์ตโฟน
- ทดสอบการส่งคำสั่งระยะไกล เช่น เริ่มทำความสะอาด หยุดการทำงาน หรือเปลี่ยนโหมดทำความสะอาด

โดยทั้ง 3 ขั้นตอนสามารถ แสดงด้วยตารางการดำเนินงานดัง ตารางที่ 1.1

ตารางที่ 1.1 แผนการดำเนินงานโครงการ

ขั้นตอนการดำเนินงาน	สัปดาห์ที่			
	1	2	3	4
1. เสนอโครงร่าง	✓			
2. เขียนโครงการเพื่อพิจารณางบประมาณ	✓			
3. เตรียมอุปกรณ์สำหรับการศึกษา		✓	✓	
4. ดำเนินการสร้างตัวชิ้นงานและดำเนินการทดลอง		✓	✓	
5. สรุปผล			✓	
6. สรุปอภิปรายผลและข้อเสนอแนะ				✓
7. จัดทำรายงานและนำเสนอผลงาน				✓

1.4 สถานที่ดำเนินงาน

1.4.1 อาคาร 12 ชั้น ห้อง 1008 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1.4.2 อาคาร 12 ชั้น ห้อง 1216 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1.5 ผลที่คาดว่าจะได้รับ

1.5.1 หุ่นยนต์สามารถเก็บฝุ่นได้อัตโนมัติ ควบคุมผ่านแอปพลิเคชันได้

1.5.2 ช่วยลดภาระการทำความสะอาดบ้าน ทำให้ชีวิตสะดวกขึ้น

1.5.3 พัฒนาทักษะด้านการเขียนโค้ด อิเล็กทรอนิกส์ และ IoT

1.5.4 เป็นพื้นฐานในการต่อยอดเทคโนโลยีหุ่นยนต์ทำความสะอาดที่ชาญฉลาดมากขึ้น

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ในบทที่ 2 นี้กล่าวถึงทฤษฎีที่เกี่ยวข้องกับโครงงานระบบไอโอทั้งหมด โดยแบ่งเป็นหัวข้อย่อย ดังนี้

2.1 โปรแกรม Arduino

2.1.1 Arduino

เป็นแพลตฟอร์มฮาร์ดแวร์และซอฟต์แวร์แบบโอเพนซอร์สที่ใช้งานง่าย เหมาะสำหรับผู้เริ่มต้น และผู้เชี่ยวชาญในการสร้างอุปกรณ์อิเล็กทรอนิกส์แบบโต้ตอบ

- บอร์ด Arduino: บอร์ดไมโครคอนโทรลเลอร์ที่สามารถรับอินพุตจากเซ็นเซอร์หรือสวิตช์ และควบคุมเอาต์พุต
- Arduino IDE (Integrated Development Environment) เป็นโปรแกรมคอมพิวเตอร์ที่ใช้ในการเขียน อัปโหลด และทดสอบโค้ดสำหรับบอร์ด Arduino แถบเมนูใน Arduino IDE มีหน้าที่หลักในการจัดการไฟล์ แก้ไขโค้ด อัปโหลดโค้ด และตั้งค่าโปรแกรม ต่อไปนี้เป็นคำอธิบายโดยละเอียด แสดงดังภาพที่ 2.1
- ฟังก์ชันหลักของ Arduino IDE :

void setup():

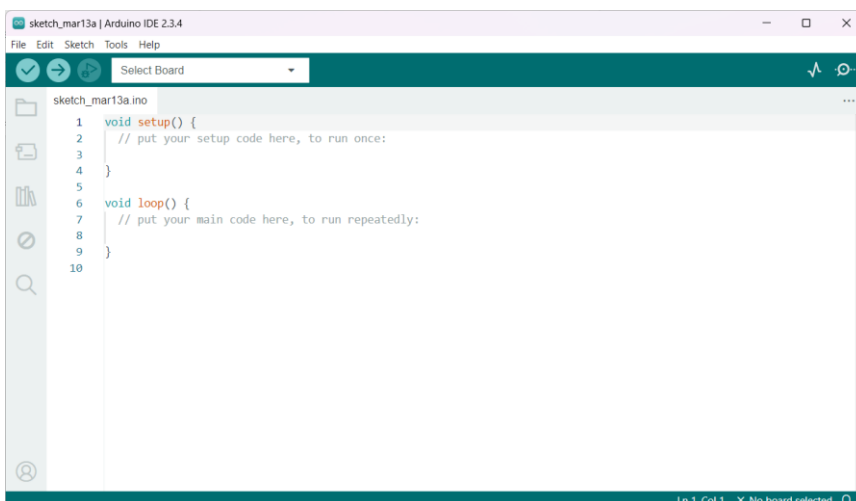
ฟังก์ชันนี้จะถูกเรียกใช้งานเพียงครั้งเดียวเมื่อบอร์ด Arduino เริ่มทำงาน

ใช้สำหรับกำหนดค่าเริ่มต้นของบอร์ด เช่น กำหนดโหมดของขา (pin) และเริ่มต้นการสื่อสารอนุกรม

void loop():

ฟังก์ชันนี้จะถูกเรียกใช้งานซ้ำๆ อย่างต่อเนื่องหลังจากฟังก์ชัน setup() ทำงานเสร็จ

ใช้สำหรับเขียนโปรแกรมหลักที่ต้องการให้บอร์ด Arduino ทำงานซ้ำๆ



ภาพที่ 2.1.1 ลักษณะโดยทั่วไปของโปรแกรม Arduino IDE

2.1.2 Serial Monitor

เป็นเครื่องมือวินิจฉัยและสื่อสารแบบอนุกรมใน Arduino IDE (Integrated Development Environment) ซึ่งทำหน้าที่เป็นอินเทอร์เฟซระหว่างบอร์ด Arduino และคอมพิวเตอร์ของผู้ใช้ โดยมีวัตถุประสงค์หลักดังนี้ ภาพที่ 2.1.2 หน้าต่าง Serial Monitor :

- การเปิดใช้งาน : สามารถเปิด Serial Monitor ได้โดยคลิกที่ไอคอนบนขวา "Serial Monitor" ใน Arduino IDE
- การแสดงผล: ข้อมูลที่ส่งมาจากบอร์ด Arduino จะปรากฏในหน้าต่าง Serial Monitor

การแสดงผลข้อมูลขาออก (Output Data Visualization) :

Serial Monitor ช่วยให้ผู้ใช้สามารถตรวจสอบข้อมูลที่ส่งมาจากบอร์ด Arduino ได้แบบเรียลไทม์ ซึ่งรวมถึง: ค่าเซ็นเซอร์ (Sensor Readings): แสดงค่าที่อ่านได้จากเซ็นเซอร์ต่างๆ เช่น อุณหภูมิ ความชื้น หรือแสง ข้อความวินิจฉัย (Diagnostic Messages): แสดงข้อความที่ใช้ในการตรวจสอบข้อผิดพลาดหรือสถานะการทำงานของโปรแกรม

การรับข้อมูลขาเข้า (Input Data Reception) :

Serial Monitor อนุญาตให้ผู้ใช้ส่งข้อมูลจากคอมพิวเตอร์ไปยังบอร์ด Arduino ผ่านการสื่อสารแบบอนุกรม ซึ่งสามารถใช้ในการ: ส่งคำสั่งควบคุม (Control Commands): ส่งคำสั่งเพื่อควบคุมการทำงานของอุปกรณ์ที่เชื่อมต่อกับบอร์ด Arduino ป้อนข้อมูล (Data Input): ป้อนข้อมูลให้กับโปรแกรม

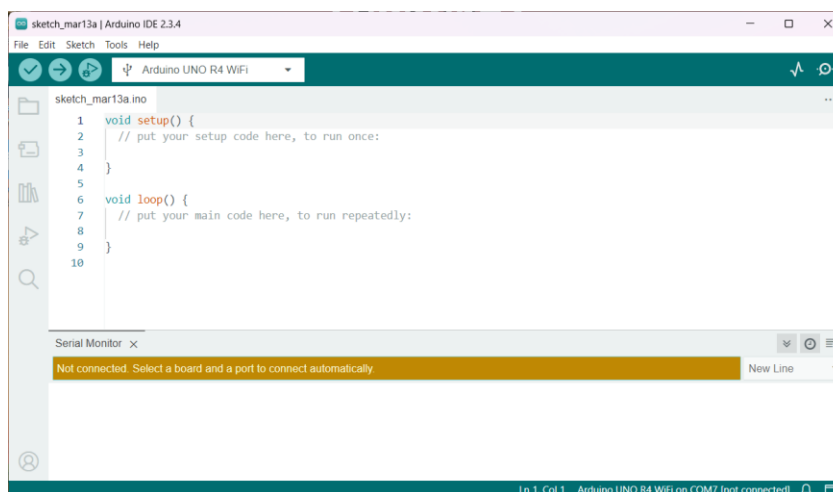
Arduino เพื่อทดสอบหรือปรับแต่งการทำงานการรับข้อมูลขาเข้าช่วยให้ผู้ใช้สามารถโต้ตอบกับบอร์ด Arduino และควบคุมการทำงานของโปรแกรมได้

การทดสอบ (Testing):

Serial Monitor ช่วยให้ผู้ใช้สามารถทดสอบการทำงานของโปรแกรม Arduino และตรวจสอบว่าอุปกรณ์ต่างๆ ทำงานตามที่คาดหวังหรือไม่

หลักการทำงาน:

การสื่อสารแบบอนุกรม (Serial Communication): Serial Monitor ใช้การสื่อสารแบบอนุกรม ซึ่งเป็นการส่งข้อมูลทีละบิตผ่านสายสัญญาณเส้นเดียว อัตรา Baud (Baud Rate): อัตรา Baud คือความเร็วในการสื่อสาร ซึ่งต้องตรงกันทั้งใน Arduino IDE และบอร์ด Arduino เพื่อให้การสื่อสารเป็นไปอย่างถูกต้อง คำสั่ง Serial: Arduino IDE มีคำสั่ง Serial ที่ใช้ในการส่งและรับข้อมูลผ่าน Serial Monitor เช่น Serial.print(), Serial.println(), Serial.read() และ Serial.available()

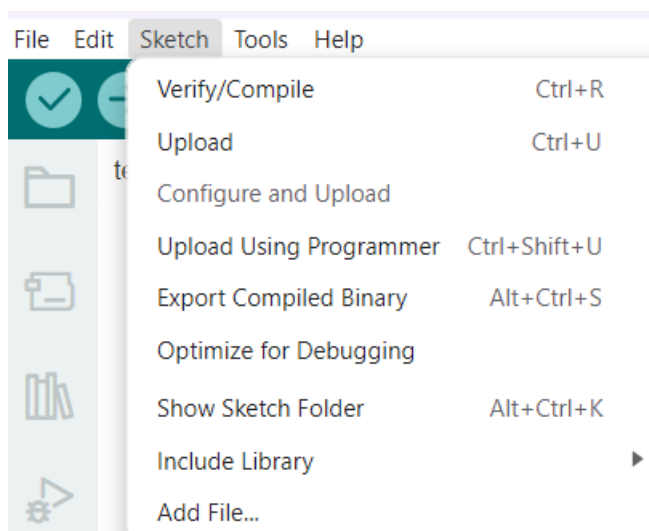


ภาพที่ 2.1.2 หน้าต่าง Serial Monitor

- เมนู "Sketch" ใน Arduino IDE เป็นส่วนที่ใช้จัดการและควบคุมโค้ดที่เขียน หรือที่เรียกว่า "Sketch" เมนูนี้มีตัวเลือกที่สำคัญหลายอย่าง แสดงดังภาพที่ 2.1.3 ดังนี้ :

- Verify/Compile (ตรวจสอบ/คอมไพล์): ตัวเลือกนี้ใช้เพื่อตรวจสอบว่าโค้ดที่เขียนมีข้อผิดพลาดทางไวยากรณ์หรือไม่ และแปลงโค้ดให้อยู่ในรูปแบบที่บอร์ด Arduino เข้าใจได้ (คอมไพล์) การตรวจสอบนี้สำคัญมากก่อนที่จะอัปโหลดโค้ดไปยังบอร์ด เพื่อป้องกันปัญหาที่อาจเกิดขึ้น
- Upload (อัปโหลด): ตัวเลือกนี้ใช้เพื่อคอมไพล์โค้ดและอัปโหลดไปยังบอร์ด Arduino ที่เชื่อมต่อกับคอมพิวเตอร์ เมื่ออัปโหลดสำเร็จ บอร์ด Arduino จะเริ่มทำงานตามโค้ดที่เขียน
- Include Library (รวมไลบรารี): ตัวเลือกนี้ใช้เพื่อเพิ่มไลบรารี (Library) ลงใน Sketch ไลบรารีคือชุดของโค้ดที่เขียนไว้แล้ว ซึ่งช่วยให้คุณใช้งานฟังก์ชันต่างๆ ได้ง่ายขึ้น เช่น การควบคุมเซ็นเซอร์ หรือการสื่อสารกับอุปกรณ์อื่นๆ
- Add File (เพิ่มไฟล์): ตัวเลือกนี้ใช้เพื่อเพิ่มไฟล์อื่นๆ ลงใน Sketch ของคุณ เช่น ไฟล์โค้ดเพิ่มเติมหรือไฟล์ข้อมูล
- Show Sketch Folder (แสดงโฟลเดอร์ Sketch): ใช้สำหรับเปิดโฟลเดอร์ที่เก็บ Sketch ที่กำลังใช้งาน

ความสำคัญของเมนู Sketch : เมนู Sketch เป็นส่วนสำคัญในการพัฒนาโปรแกรมสำหรับ Arduino ตัวเลือกต่างๆ ในเมนูนี้ช่วยให้สามารถตรวจสอบ แก้ไข และอัปโหลดโค้ดได้อย่างมีประสิทธิภาพ



ภาพที่ 2.1.3 เมนู Sketch

2.1.4 Arduino UNO R4 Wi-Fi

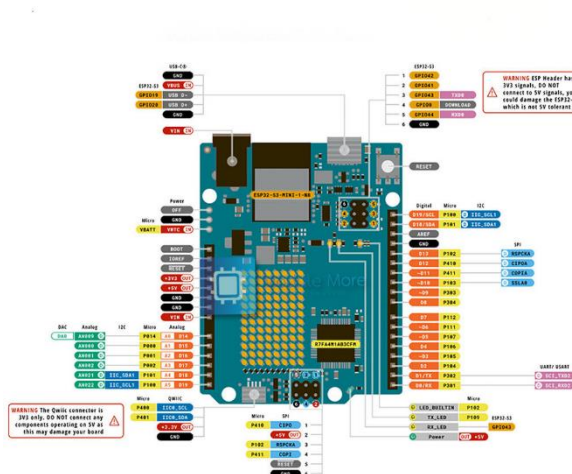
Arduino UNO R4 Wi-Fi เป็นบอร์ดไมโครคอนโทรลเลอร์รุ่นใหม่จากตระกูล Arduino UNO ที่ได้รับการอัปเดตจากรุ่น R3 โดยเพิ่มความสามารถด้านประสิทธิภาพและการเชื่อมต่อไร้สาย (Wi-Fi และ Bluetooth) บอร์ดนี้ใช้ชิปประมวลผล Renesas RA4M1 ซึ่งเป็นไมโครคอนโทรลเลอร์สถาปัตยกรรม ARM Cortex-M4 ที่มีประสิทธิภาพสูงกว่า ATmega328P ของรุ่นก่อน นอกจากนี้ ยังมาพร้อมกับโมดูล ESP32-S3 สำหรับการเชื่อมต่อ Wi-Fi และ Bluetooth

คุณสมบัติหลักของ Arduino UNO R4 Wi-Fi

1. ประสิทธิภาพสูงขึ้น – ใช้ไมโครคอนโทรลเลอร์ Renesas RA4M1 (ARM Cortex-M4) ความเร็ว 48 MHz และมีหน่วยความจำ RAM ขนาด 32 KB และแฟลช 256 KB
2. การเชื่อมต่อไร้สาย – มาพร้อมโมดูล ESP32-S3 ที่รองรับ Wi-Fi และ Bluetooth LE ทำให้สามารถเชื่อมต่อกับเครือข่ายไร้สายและอุปกรณ์ IoT ได้ง่าย
3. รองรับแรงดันไฟ 5V – แม้จะเปลี่ยนไปใช้สถาปัตยกรรม ARM แต่ยังคงรองรับอุปกรณ์เสริมและเซ็นเซอร์ที่ทำงานบนแรงดันไฟ 5V ได้
4. พิน I/O และรูปแบบบอร์ดเหมือน UNO R3 – สามารถใช้งานร่วมกับอุปกรณ์เสริม (shields) ของ Arduino UNO รุ่นก่อนหน้าได้
5. DAC และพิน PWM เพิ่มเติม – มี DAC 12-bit สำหรับการสร้างสัญญาณอะนาล็อก และเพิ่มพิน PWM ที่มากขึ้นเพื่อรองรับการควบคุมอุปกรณ์ได้หลากหลายขึ้น
6. RTC (Real-Time Clock) – มีวงจรมินิเจอร์ในตัว ทำให้สามารถจัดการเวลาได้โดยไม่ต้องใช้โมดูลเสริม
7. NeoPixel LED และ CAN Bus – บอร์ดมาพร้อมกับไฟ LED RGB NeoPixel และรองรับการสื่อสารแบบ CAN Bus ซึ่งเป็นมาตรฐานที่ใช้ในระบบอุตสาหกรรมและยานยนต์
8. หน่วยความจำมากขึ้น – มี SRAM และแฟลชเมมโมรีที่มากขึ้นเมื่อเทียบกับ Arduino UNO R3 ทำให้รองรับโปรแกรมที่ซับซ้อนกว่าได้

การใช้งาน Arduino UNO R4 Wi-Fi เหมาะสำหรับโครงการที่ต้องการการเชื่อมต่อไร้สาย เช่น

1. IoT (Internet of Things) – ใช้เชื่อมต่อเซ็นเซอร์และส่งข้อมูลผ่าน Wi-Fi หรือ Bluetooth
2. Home Automation – ควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านแอปพลิเคชันหรือเว็บเซิร์ฟเวอร์
3. Data Logging & Monitoring – เก็บข้อมูลจากเซ็นเซอร์และส่งไปยังคลาวด์
4. Robotics & Automation – ควบคุมมอเตอร์ เซอร์โวมอเตอร์ และอุปกรณ์อุตสาหกรรมผ่าน Wi-Fi



ภาพที่ 2.1.4 Arduino UNO R4 Wi-Fi

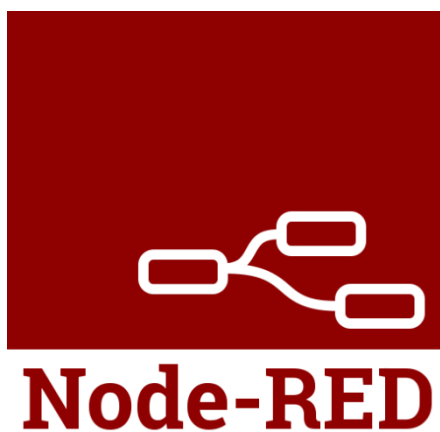
2.2 Node – Red

Node-RED เป็นเครื่องมือพัฒนาซอฟต์แวร์แบบโอเพ่นซอร์สที่ใช้หลักการโปรแกรมเชิงโฟลว์ (Flow-based Programming) ซึ่งช่วยให้ผู้ใช้สามารถเชื่อมโยงฮาร์ดแวร์ บริการออนไลน์ และ API ต่างๆ ได้โดยไม่ต้องเขียนโค้ดจำนวนมาก พัฒนาโดย IBM Emerging Technology และเผยแพร่ภายใต้สัญญาอนุญาตแบบ Apache 2.0 Node-RED ใช้ Node.js เป็นโครงสร้างหลัก ทำให้สามารถรันบนแพลตฟอร์มต่างๆ ได้ เช่น Windows, macOS, Linux, Raspberry Pi และระบบคลาวด์ต่างๆ โครงสร้างของ Node-RED ประกอบด้วย "โหนด" (Nodes) ซึ่งเป็นองค์ประกอบหลักในการประมวลผลข้อมูล โดยมีโหนดอินพุต (Input Nodes) ที่ใช้รับข้อมูลจากแหล่งต่างๆ เช่น HTTP Request, MQTT และ WebSockets โหนดประมวลผล (Processing Nodes) ที่ใช้จัดการและแปลงข้อมูล เช่น ฟังก์ชัน JavaScript, JSON และ

Switch และโหนดเอาต์พุต (Output Nodes) ที่ใช้ส่งข้อมูลออกไปยังปลายทาง เช่น MQTT Broker, Database หรือ API เมื่อโหนดเหล่านี้ถูกเชื่อมต่อกันจะเกิดเป็น "โฟลว์" (Flow) ซึ่งเป็นกระบวนการทำงานอัตโนมัติ Node-RED รองรับการขยายฟังก์ชันการทำงานเพิ่มเติมผ่าน Node-RED Library ซึ่งมีปลั๊กอินและโหนดสำเร็จรูปให้ใช้งานมากมาย นอกจากนี้ ยังสามารถสร้างแดชบอร์ด (Dashboard) เพื่อนำเสนอข้อมูลในรูปแบบกราฟ แผนภูมิ และปุ่มควบคุมสำหรับการโต้ตอบกับผู้ใช้ได้ ภาพที่ 2.2 Node-red

2.2.1 คุณสมบัติหลักของ Node-RED

การพัฒนาแบบ Low-Code: ใช้หลักการลากและวาง (Drag-and-Drop) ในการออกแบบโฟลว์การทำงาน รองรับโหนดหลากหลายประเภท: มีโหนดมาตรฐานที่รองรับการเชื่อมต่อ MQTT, HTTP, WebSockets, Database และอื่นๆ สามารถขยายเพิ่มเติมได้: ผู้ใช้สามารถสร้างโหนดของตนเองหรือใช้โหนดเสริมจาก Node-RED Library รองรับ IoT และ Edge Computing: เหมาะสำหรับการพัฒนาโซลูชัน IoT โดยสามารถรันบน Raspberry Pi, Docker, Kubernetes และแพลตฟอร์มคลาวด์ต่างๆ ใช้ JavaScript และ Node.js: พัฒนาโดยใช้ Node.js และสามารถเขียนโค้ดเพิ่มเติมใน JavaScript ได้



ภาพที่ 2.2 Node-red

2.2.2 องค์ประกอบหลักของ Node-RED

- โหนด (Nodes) มี โหนดอินพุต (Input Nodes): รับข้อมูลจากแหล่งภายนอก เช่น HTTP Request, MQTT, WebSockets โหนดประมวลผล (Processing Nodes): ใช้สำหรับแปลง

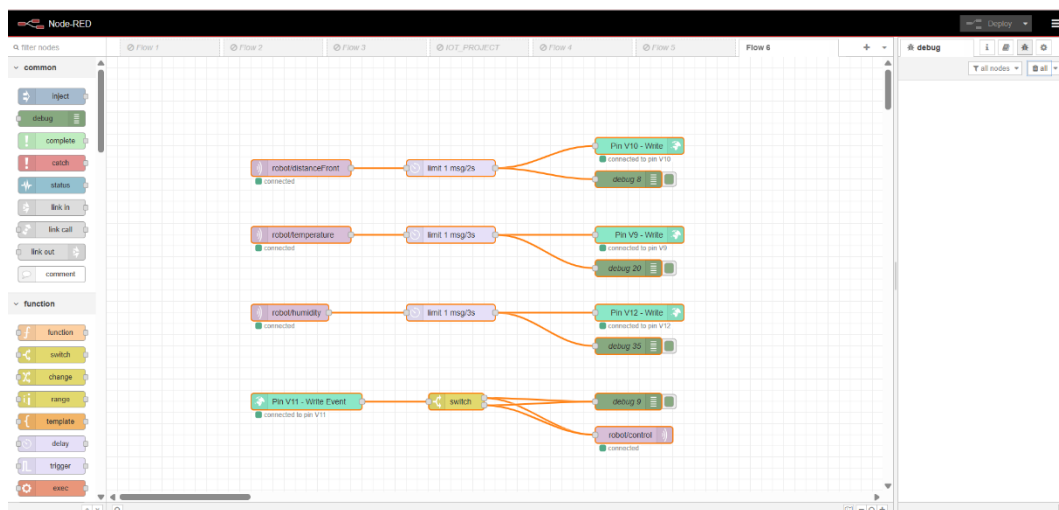
ข้อมูล เช่น ฟังก์ชัน JavaScript, JSON, Switch โหนดเอาต์พุต (Output Nodes): ส่งข้อมูลออกไปยังปลายทาง เช่น MQTT Broker, Database, API

- โฟลว์ (Flows) เป็นชุดของโหนดที่เชื่อมต่อกันเพื่อสร้างกระบวนการทำงาน
- แดชบอร์ด (Dashboard) ใช้สำหรับสร้าง UI แสดงผลข้อมูล เช่น กราฟ แผนภูมิ และปุ่มควบคุม

2.2.3 การใช้งาน Node-RED กับ IoT และระบบอัตโนมัติ

Node-RED นิยมใช้ในงานด้าน IoT, การเชื่อมต่อเซนเซอร์ และระบบอัตโนมัติ เช่น:

- Home Automation: ควบคุมอุปกรณ์อัจฉริยะผ่าน MQTT และ Home Assistant
- Industrial IoT (IIoT): ใช้ในการเชื่อมต่อ SCADA, OPC-UA, Modbus
- API Integration: เชื่อมต่อ REST API, WebSockets และฐานข้อมูล
- Machine Learning & AI: ประมวลผลข้อมูลร่วมกับ TensorFlow.js และ AI API



ภาพที่ 2.2.2 องค์ประกอบหลักของ Node-RED

2.3 MQTT (Message Queuing Telemetry Transport)

MQTT เป็นโพรโทคอลการสื่อสารแบบ publish/subscribe ที่ออกแบบมาสำหรับการส่งข้อมูลที่มีขนาดเล็กผ่านเครือข่ายที่มีข้อจำกัดด้านแบนด์วิดท์หรือพลังงาน เช่น ระบบ Internet of Things (IoT) และเครือข่ายที่มีการเชื่อมต่อแบบไร้สายที่ไม่เสถียร โพรโทคอลนี้ถูกพัฒนาโดย IBM ในปี 1999 และได้รับความนิยมอย่างแพร่หลายเนื่องจากการใช้ทรัพยากรต่ำ มีโครงสร้างที่เรียบง่าย และสามารถใช้งานได้ในระบบที่มีข้อจำกัดด้านพลังงาน เช่น อุปกรณ์ IoT, ระบบเซ็นเซอร์ไร้สาย และอุปกรณ์สมาร์ตโฮม

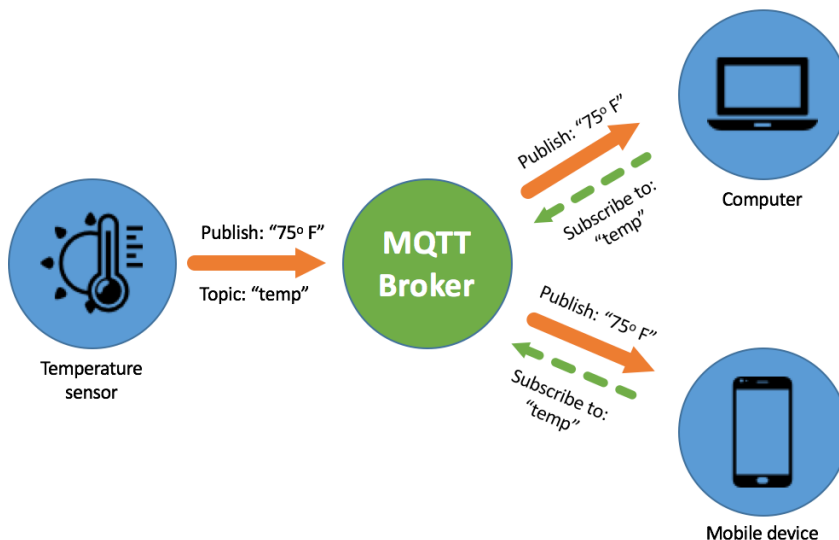
หลักการทำงานของ MQTT

1. MQTT ใช้โมเดล publish/subscribe (การเผยแพร่และการสมัครรับข้อมูล) ซึ่งแตกต่างจากการสื่อสารแบบไคลเอนต์-เซิร์ฟเวอร์ทั่วไป เช่น HTTP โดยมีองค์ประกอบหลัก 3 ส่วน ดังนี้
2. Publisher (ผู้เผยแพร่ข้อมูล) เป็นอุปกรณ์ที่ส่งข้อมูลไปยัง Broker ไม่ต้องรู้ว่ามีใครจะรับข้อมูล เพียงแค่ส่งไปยังหัวข้อ (Topic) ที่กำหนด ตัวอย่าง: เซ็นเซอร์อุณหภูมิที่ส่งค่ามายัง MQTT Broker
3. Broker (ตัวกลางจัดการข้อมูล) ทำหน้าที่รับข้อมูลจาก Publisher และกระจายไปยัง Subscriber ที่สมัครรับข้อมูลอยู่ เป็นหัวใจหลักของ MQTT ซึ่งช่วยให้ระบบมีความยืดหยุ่นสูง ตัวอย่าง: Mosquitto หรือ EMQX เป็น MQTT Broker ยอดนิยม
4. Subscriber (ผู้รับข้อมูล) เป็นอุปกรณ์ที่สมัครรับข้อมูลจากหัวข้อที่ต้องการจะได้รับข้อมูลเมื่อ Publisher ส่งข้อมูลมายัง Broker ตัวอย่าง: แอปพลิเคชันบนมือถือที่สมัครรับค่าจากเซ็นเซอร์อุณหภูมิ

ตัวอย่างการใช้งาน MQTT

1. สมาร์ตโฮม (Smart Home) ใช้ MQTT เพื่อควบคุมอุปกรณ์ภายในบ้าน เช่น เปิด-ปิดไฟ แสดงค่าจากเซ็นเซอร์อุณหภูมิ หรือควบคุมเครื่องปรับอากาศผ่านมือถือ
2. อุตสาหกรรม (Industrial IoT - IIoT) ใช้ MQTT ในการตรวจสอบสถานะเครื่องจักรและการแจ้งเตือนเมื่อพบปัญหา
3. เกษตรอัจฉริยะ (Smart Farming) ใช้ MQTT ส่งข้อมูลอุณหภูมิ ความชื้น หรือค่าฝนตกจากเซ็นเซอร์ไปยังเซิร์ฟเวอร์เพื่อช่วยวิเคราะห์สภาพแวดล้อม

4. ระบบติดตามตำแหน่ง (GPS Tracking) ใช้ MQTT เพื่อส่งข้อมูลตำแหน่ง GPS ของรถยนต์หรือยานพาหนะไปยังศูนย์ควบคุม



ภาพที่ 2.3 หลักการทำงาน MQTT

2.4 Blynk

Blynk เป็นแพลตฟอร์มสำหรับ Internet of Things (IoT) ที่ช่วยให้ผู้ใช้สามารถควบคุมและตรวจสอบอุปกรณ์ Arduino, ESP8266, ESP32, Raspberry Pi และไมโครคอนโทรลเลอร์อื่น ๆ ผ่านอินเทอร์เน็ตได้ง่าย ๆ โดยใช้แอปพลิเคชันบนมือถือหรือเว็บแดชบอร์ด ช่วยให้สามารถพัฒนา Smart Home, ระบบควบคุมอัตโนมัติ, การติดตามเซ็นเซอร์ และการแจ้งเตือนแบบเรียลไทม์ ได้โดยไม่ต้องเขียนโค้ดที่ซับซ้อน

คุณสมบัติของ Blynk

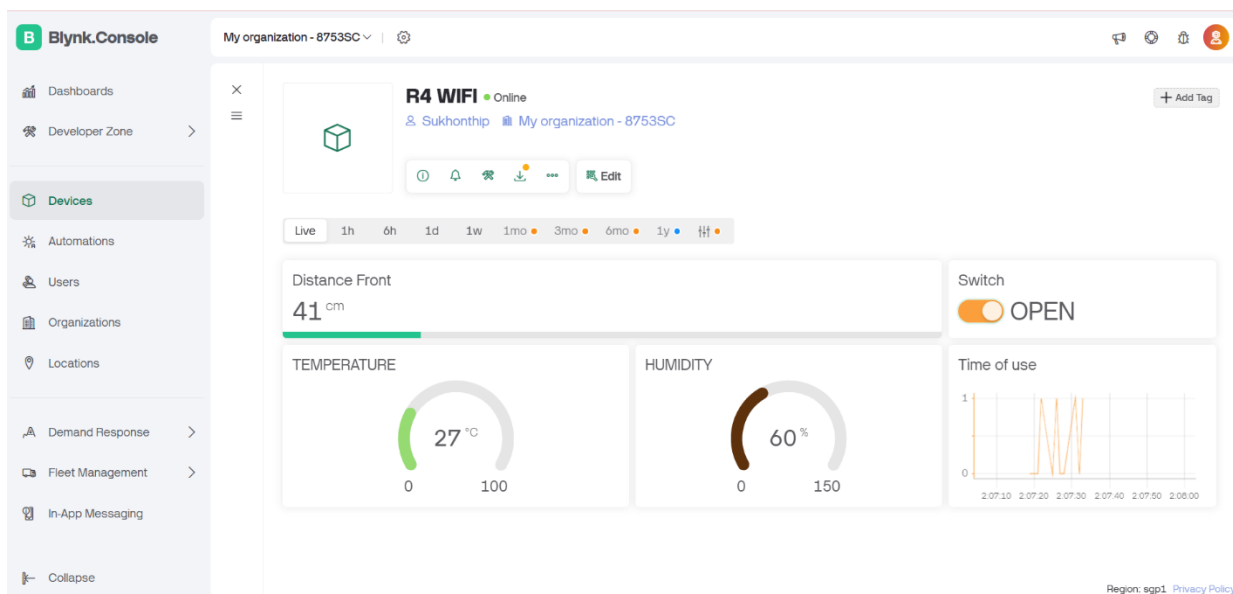
1. รองรับหลายแพลตฟอร์ม – ใช้ได้กับ Arduino, ESP8266, ESP32, Raspberry Pi, NodeMCU ฯลฯ
2. มีแอปพลิเคชันสำหรับมือถือ – สามารถสร้างอินเทอร์เฟซสำหรับควบคุมอุปกรณ์ได้ผ่าน iOS และ Android
3. รองรับ WiFi, Bluetooth, GSM, Ethernet และ MQTT – ใช้งานได้กับเครือข่ายที่หลากหลาย
4. มีระบบ Blynk Cloud – ไม่ต้องตั้งค่าเซิร์ฟเวอร์เอง
5. รองรับ Web Dashboard – ควบคุมอุปกรณ์ผ่านเว็บไซต์ได้

หลักการทำงานของ Blynk ใช้ Blynk Cloud เป็นตัวกลางในการสื่อสารระหว่างอุปกรณ์กับแอปพลิเคชัน โดยมีองค์ประกอบหลักดังนี้

1. Blynk App – แอปพลิเคชันบนมือถือที่ใช้ควบคุมและแสดงผลข้อมูลจากอุปกรณ์
2. Blynk Cloud – เซิร์ฟเวอร์กลางที่ช่วยให้การสื่อสารระหว่างอุปกรณ์กับแอปพลิเคชันเป็นไปอย่างราบรื่น
3. Blynk Library – ไลบรารีสำหรับใช้ในไมโครคอนโทรลเลอร์ เช่น Arduino, ESP8266, ESP32 เพื่อให้สามารถเชื่อมต่อกับ Blynk Cloud ได้

การใช้งาน Blynk ในโปรเจก IoT

1. ระบบบ้านอัจฉริยะ (Smart Home) เปิด-ปิดไฟ หรือเครื่องใช้ไฟฟ้าผ่านมือถือ ตรวจสอบอุณหภูมิและความชื้นในบ้าน สั่งเปิด-ปิดแอร์อัตโนมัติ
2. ระบบแจ้งเตือนแบบเรียลไทม์ แจ้งเตือนเมื่อมีการเคลื่อนไหวผ่านเซ็นเซอร์ PIR แจ้งเตือนเมื่ออุณหภูมิสูงเกินกำหนด ส่งข้อมูลไปยัง Line หรือ Telegram
3. ระบบควบคุมอุปกรณ์จากระยะไกล เปิด-ปิดปั๊มน้ำผ่านแอป Blynk ควบคุมหุ่นยนต์หรือโดรนผ่านเครือข่ายอินเทอร์เน็ต
4. ระบบมอนิเตอร์ข้อมูลเซ็นเซอร์ แสดงค่าจากเซ็นเซอร์อุณหภูมิ ความชื้น หรือ PM2.5 บนมือถือ บันทึกข้อมูลลง Blynk Cloud และดูย้อนหลังได้



ภาพที่ 2.4 Web Dashboard

2.5 L298N Motor Driver

L298N เป็นโมดูล motor driver ที่ใช้ในการควบคุมมอเตอร์ DC และมอเตอร์แบบ Stepper ผ่านการใช้งานไมโครคอนโทรลเลอร์ เช่น Arduino หรือ ESP32/ESP8266 โมดูลนี้มีคุณสมบัติพิเศษคือ สามารถขับมอเตอร์ได้สองตัวพร้อมกัน และรองรับการควบคุมทิศทางและความเร็วของมอเตอร์โดยใช้ PWM (Pulse Width Modulation)

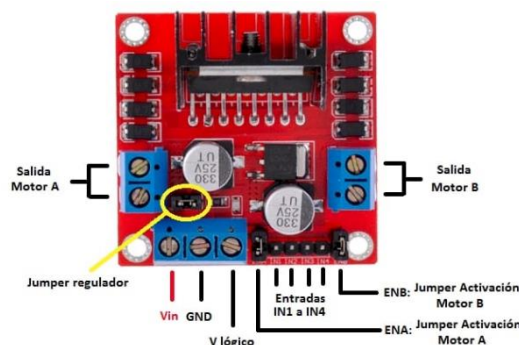
ถูกออกแบบให้สามารถขับมอเตอร์ DC หรือ Stepper ด้วยแรงดันไฟฟ้า จาก 4.5V ถึง 46V และสามารถขับเคลื่อนสูงสุดถึง 2A ต่อช่องสัญญาณ ซึ่งทำให้มันเหมาะสำหรับการใช้งานในระบบหุ่นยนต์หรือการควบคุมมอเตอร์ในโปรเจกต์ต่าง ๆ

หลักการทำงานของ L298N

โมดูล L298N ใช้ H-Bridge ในการควบคุมมอเตอร์ DC ซึ่งช่วยให้สามารถเปลี่ยนทิศทางการหมุนของมอเตอร์ได้ โดยการสลับขั้วไฟฟ้า

1. ทิศทางการหมุนมอเตอร์ โดยการควบคุมขาของอินพุต 2 ขา (Input1, Input2 สำหรับมอเตอร์ 1 หรือ Input3, Input4 สำหรับมอเตอร์ 2) สามารถเปลี่ยนทิศทางการหมุนของมอเตอร์ได้ ถ้าทั้งสอง

- ขามีค่า LOW มอเตอร์จะหยุด ถ้ามีค่า HIGH-LOW หรือ LOW-HIGH มอเตอร์จะหมุนในทิศทางหนึ่ง ถ้ามีค่า HIGH-HIGH หรือ LOW-LOW มอเตอร์จะหยุดหรือหมุนได้ขึ้นอยู่กับ การเชื่อมต่อของ
2. ควบคุมความเร็ว (PWM) ใช้สัญญาณ PWM ที่มีความถี่และความกว้างต่างกันในการควบคุมความเร็วของมอเตอร์ โดยการส่งสัญญาณ PWM เข้าไปที่ Enable Pin (สำหรับแต่ละมอเตอร์) จะทำให้มอเตอร์หมุนเร็วหรือ ช้าตามค่าที่กำหนด



ภาพที่ 2.5 L298N Motor Driver

2.6 Gear Motor

เกียร์มอเตอร์ (Gear Motor) คือ มอเตอร์ไฟฟ้า ที่มีชุดเกียร์ติดตั้งอยู่ภายในหรือเชื่อมต่อกับมอเตอร์เพื่อปรับเปลี่ยนอัตราการหมุนและเพิ่มแรงบิด (Torque) โดยใช้เกียร์ในการเปลี่ยนความเร็วและแรงดันจากมอเตอร์ให้เหมาะสมกับการใช้งานที่ต้องการ เช่น การควบคุมมอเตอร์ในหุ่นยนต์หรือเครื่องจักรที่ต้องการแรงบิดสูงแต่หมุนช้าลง เกียร์มอเตอร์สามารถทำงานได้ในช่วงแรงดันไฟฟ้าหลายระดับ เช่น 3V, 6V, 9V, 12V ขึ้นอยู่กับการใช้งานและประเภทของมอเตอร์ที่เลือก

หลักการทำงานของเกียร์มอเตอร์

การใช้เกียร์ช่วยให้สามารถควบคุมแรงบิดและความเร็วได้ตามต้องการในหลายๆ การใช้งาน

1. มอเตอร์ไฟฟ้า: เกียร์มอเตอร์ประกอบด้วย มอเตอร์ไฟฟ้า ที่รับพลังงานจากแหล่งจ่ายไฟฟ้าและทำการหมุนแกนของมอเตอร์
2. ชุดเกียร์: ชุดเกียร์ที่เชื่อมต่อกับแกนของมอเตอร์จะทำการลดความเร็วในการหมุนของมอเตอร์ แต่เพิ่มแรงบิด (Torque) ขึ้น ทำให้สามารถใช้มอเตอร์ในการขับเคลื่อนโหลดที่มีน้ำหนักมากหรือมีแรงต้านสูง
3. การปรับอัตราส่วนของเกียร์: เกียร์ที่มีอัตราส่วนต่าง ๆ จะให้ผลลัพธ์ที่แตกต่างกัน เช่น เกียร์อัตราส่วนต่ำจะให้แรงบิดสูงและความเร็วต่ำ ขณะที่เกียร์อัตราส่วนสูงจะให้ความเร็วสูงแต่แรงบิดต่ำ

ประเภทของเกียร์มอเตอร์

เกียร์มอเตอร์มีหลายประเภทที่สามารถเลือกใช้ได้ตามลักษณะการใช้งาน ได้แก่:

1. DC Gear Motor (มอเตอร์เกียร์ DC): ใช้พลังงานจากแหล่งจ่ายไฟฟ้า DC เช่น แบตเตอรี่ 3V, 6V, 12V โดยเหมาะกับการใช้งานที่ต้องการแรงบิดสูง เช่น หุ่นยนต์เล็กๆ หรืออุปกรณ์ที่ต้องการกำลังขับเคลื่อนสูง
2. Stepper Gear Motor (มอเตอร์เกียร์สเต็ปเปอร์): ใช้ในงานที่ต้องการการควบคุมตำแหน่งที่แม่นยำ เช่น เครื่องพิมพ์ 3D หรือหุ่นยนต์ที่ต้องการการหมุนที่แม่นยำ
3. Planetary Gear Motor (มอเตอร์เกียร์พลาแนรี): ใช้เกียร์หลายตัวที่อยู่ในลักษณะวงกลม เพื่อเพิ่มแรงบิดและลดขนาดของมอเตอร์ เหมาะกับการใช้งานที่ต้องการขนาดเล็กและแรงบิดสูง



ภาพที่ 2.6 Gear Motor

2.7 Ultrasonic Sensor

Ultrasonic Sensor หรือ เซ็นเซอร์อัลตราโซนิก คืออุปกรณ์ที่ใช้เทคโนโลยีคลื่นเสียงความถี่สูง (อัลตราโซนิก) ในการวัดระยะทางหรือหามุมของวัตถุที่อยู่ในเส้นทางการส่งสัญญาณเสียง เซ็นเซอร์นี้ทำงานโดยการส่งคลื่นเสียงความถี่สูงไปยังวัตถุ แล้วรอรับการสะท้อนกลับของคลื่นเสียง (Echo) เพื่อคำนวณระยะห่างจากเซ็นเซอร์ถึงวัตถุที่สะท้อนกลับคลื่นเสียง

หลักการทำงานของ Ultrasonic Sensor

หลักการทำงานของเซ็นเซอร์อัลตราโซนิกประกอบด้วย 2 ส่วนหลัก คือ ส่ง (Transmit) และ รับ (Receive) คลื่นเสียง

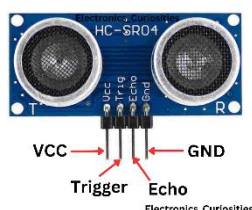
1. การส่งคลื่นเสียง : เซ็นเซอร์อัลตราโซนิกจะส่งคลื่นเสียงความถี่สูง (ประมาณ 40kHz) ออกไปจากเซ็นเซอร์ในทิศทางที่ต้องการตรวจวัดระยะทาง

2. การสะท้อนของคลื่นเสียง : เมื่อคลื่นเสียงไปถึงวัตถุใด ๆ มันจะสะท้อนกลับมายังเซ็นเซอร์
 3. การรับคลื่นเสียง : เซ็นเซอร์จะรับคลื่นเสียงที่สะท้อนกลับและคำนวณระยะห่างจากเวลาที่คลื่นเสียงเดินทางไปถึงวัตถุและกลับมา โดยใช้สูตรการคำนวณจากระยะทาง = (ความเร็วของเสียง × เวลา) ÷ 2
- ความเร็วของเสียง : โดยทั่วไปจะอยู่ที่ประมาณ 343 เมตร/วินาทีในอากาศที่อุณหภูมิ 20°C
- เวลา : เป็นเวลาที่คลื่นเสียงใช้ในการเดินทางไปถึงวัตถุและกลับมา
- ดังนั้น เซ็นเซอร์จะใช้เวลาในการรับคลื่นเสียงที่สะท้อนกลับเพื่อคำนวณระยะทางที่เกิดขึ้น

ส่วนประกอบของ Ultrasonic Sensor

เซ็นเซอร์อัลตราโซนิกส่วนใหญ่ที่ใช้ในการตรวจวัดระยะทางมักจะประกอบด้วย 3 ส่วนหลัก ๆ คือ:

1. Transmitter (ตัวส่งคลื่นเสียง) : ส่งคลื่นเสียงอัลตราโซนิกไปยังวัตถุ
2. Receiver (ตัวรับคลื่นเสียง) : รับคลื่นเสียงที่สะท้อนกลับจากวัตถุ
3. Control Circuit (วงจรควบคุม) : ใช้ในการควบคุมการทำงานของตัวส่งและตัวรับคลื่นเสียง พร้อมคำนวณระยะทาง



ภาพที่ 2.7 Ultrasonic Sensor

2.8 Micro Servo Tower Pro SG90

Micro Servo Tower Pro SG90 เป็น เซอร์โวมอเตอร์ขนาดเล็ก ที่ใช้สำหรับการควบคุมการหมุนในมุมที่กำหนด โดยมีขนาดกะทัดรัดและน้ำหนักเบา เหมาะสำหรับใช้งานในโปรเจกต์หุ่นยนต์, โดรน, หรืออุปกรณ์อิเล็กทรอนิกส์ที่ต้องการการเคลื่อนไหวที่มีความแม่นยำในการหมุน เช่น การเปิดปิดประตู, การควบคุมการหมุนของกลไกต่างๆ ในระบบอัตโนมัติ

SG90 ถือเป็นเซอร์โวมอเตอร์ที่นิยมใช้ในวงการอิเล็กทรอนิกส์และหุ่นยนต์ เนื่องจากมีราคาที่ถูกและสามารถใช้งานได้ง่ายผ่านการควบคุมสัญญาณ PWM (Pulse Width Modulation)

ลักษณะของ Micro Servo Tower Pro SG90

1. ขนาด: เซอร์โวมอเตอร์ SG90 มีขนาดเล็กและเบา ทำให้สะดวกในการติดตั้งในโปรเจกต์ที่ต้องการอุปกรณ์ที่ไม่ใช้พื้นที่มาก
2. แรงบิด (Torque): SG90 มีแรงบิดประมาณ 1.8 kg·cm (ที่ 4.8V) ซึ่งถือว่าเหมาะสมสำหรับการใช้งานทั่วไปที่ไม่ต้องการแรงบิดสูงมาก
3. ช่วงการหมุน: สามารถหมุนได้ประมาณ 180 องศา (จาก -90 องศาถึง +90 องศา)
4. พลังงาน: เซอร์โวมอเตอร์ SG90 ใช้แรงดันไฟฟ้าในช่วง 4.8V ถึง 6V ซึ่งเป็นช่วงแรงดันที่สามารถใช้กับแหล่งจ่ายไฟทั่วไปได้

การทำงานของ Micro Servo Tower Pro SG90

การทำงานของเซอร์โวมอเตอร์ SG90 ใช้ สัญญาณ PWM ในการควบคุมการหมุนของมอเตอร์ โดยการปรับค่า Pulse Width ในสัญญาณ PWM จะกำหนดมุมการหมุนของเซอร์โว:

1. สัญญาณ PWM (Pulse Width Modulation) เซอร์โวมอเตอร์จะได้รับสัญญาณที่มีความกว้างของ Pulse (ความยาวของสัญญาณ High) ซึ่งสามารถควบคุมการหมุนของมอเตอร์ได้ตามมุมที่ต้องการ
 - มุมการหมุน 0 องศา: จะได้รับ Pulse กว้างประมาณ 1 ms
 - มุมการหมุน 90 องศา: จะได้รับ Pulse กว้างประมาณ 1.5 ms
 - มุมการหมุน 180 องศา: จะได้รับ Pulse กว้างประมาณ 2 ms
2. การหมุน : เมื่อเซอร์โวได้รับสัญญาณ PWM ที่มีความกว้างของ Pulse ต่างกัน มันจะหมุนไปยังมุมที่สอดคล้องกับสัญญาณนั้น เช่น หาก Pulse ยาวไปถึง 2 ms เซอร์โวจะหมุนไปที่ 180 องศา หรือหาก Pulse ยาวแค่ 1 ms จะหมุนที่ 0 องศา



ภาพที่ 2.8 Micro Servo Tower Pro SG90

2.9 DHT11

DHT11 เป็นเซ็นเซอร์สำหรับวัด อุณหภูมิ (Temperature) และ ความชื้น (Humidity) ซึ่งใช้ในการเก็บข้อมูลสภาพอากาศในระบบอิเล็กทรอนิกส์ต่าง ๆ โดยมีความสามารถในการให้ข้อมูลทั้งอุณหภูมิและความชื้นในเวลาเดียวกัน เซ็นเซอร์ DHT11 นิยมใช้ในโปรเจกต์ที่เกี่ยวข้องกับการควบคุมสภาพแวดล้อม เช่น ระบบทำความเย็นอัตโนมัติ, ระบบควบคุมอุณหภูมิในโรงเรือน, และโปรเจกต์ IoT (Internet of Things)

ลักษณะของ DHT11

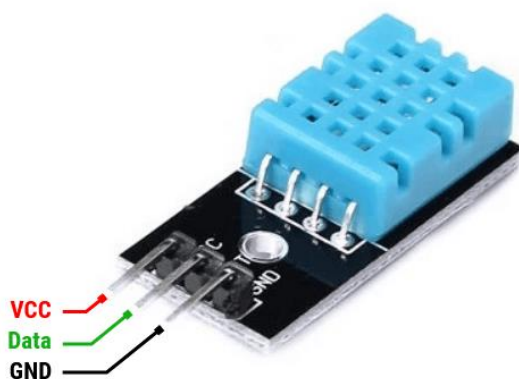
1. ขนาด : ขนาดของเซ็นเซอร์ DHT11 ค่อนข้างเล็กและกะทัดรัด ทำให้สามารถใช้งานได้โปรเจกต์ที่ต้องการพื้นที่จำกัด
2. ช่วงการวัด : อุณหภูมิ: สามารถวัดอุณหภูมิได้ตั้งแต่ 0°C ถึง 50°C ด้วยความแม่นยำประมาณ $\pm 2^{\circ}\text{C}$
3. ความชื้น: สามารถวัดความชื้นได้ตั้งแต่ 20% ถึง 90% RH (Relative Humidity) ด้วยความแม่นยำ $\pm 5\% \text{ RH}$
4. ไฟเลี้ยง: เซ็นเซอร์ DHT11 ต้องการแรงดันไฟฟ้าที่ 3.3V ถึง 5V ซึ่งสามารถใช้ร่วมกับไมโครคอนโทรลเลอร์เช่น Arduino หรือ Raspberry Pi ได้
5. สัญญาณที่ใช้ : ใช้การส่งข้อมูลแบบ Digital ผ่านสาย Data Pin ซึ่งส่งข้อมูลแบบ One-Wire ซึ่งหมายความว่า การสื่อสารระหว่างเซ็นเซอร์และไมโครคอนโทรลเลอร์ใช้เพียงเส้นสัญญาณเดียวในการส่งข้อมูล

การทำงานของ DHT11

การทำงานของเซ็นเซอร์ DHT11 เป็นการวัดข้อมูลอุณหภูมิและความชื้นในอากาศและส่งข้อมูลผ่านสาย Data Pin ไปยังไมโครคอนโทรลเลอร์ โดยใช้โปรโตคอลการสื่อสาร One-Wire ซึ่งทำให้การใช้งานไม่ซับซ้อนและใช้สายเชื่อมต่อเพียงเส้นเดียวสำหรับการส่งข้อมูล

1. การวัดอุณหภูมิและความชื้น : เซ็นเซอร์ DHT11 ใช้เทคโนโลยีเซ็นเซอร์ดิจิทัลในการตรวจจับและแปลงค่าความชื้นและอุณหภูมิให้เป็นสัญญาณดิจิทัล

2. การส่งข้อมูล : ข้อมูลที่ได้จากเซ็นเซอร์จะถูกส่งไปยังไมโครคอนโทรลเลอร์ผ่านสัญญาณ One-Wire โดยใช้เวลาไม่นานในการส่งข้อมูล 40 บิต (16 บิตสำหรับอุณหภูมิ, 16 บิตสำหรับความชื้น, และ 8 บิตสำหรับการตรวจสอบความถูกต้อง)
3. ความแม่นยำในการวัด : เซ็นเซอร์จะให้ค่าความชื้นและอุณหภูมิที่ค่อนข้างแม่นยำสำหรับงานทั่วไป โดยความแม่นยำสำหรับอุณหภูมิอยู่ที่ประมาณ $\pm 2^{\circ}\text{C}$ และสำหรับความชื้นอยู่ที่ $\pm 5\% \text{ RH}$



ภาพที่ 2.9 DHT11

2.10 Battery Alkaline 9 V

ถ่าน 9V เป็นแหล่งจ่ายไฟฟ้าที่มีแรงดันไฟฟ้า 9 โวลต์ มีลักษณะเป็นแท่งสี่เหลี่ยมขนาดเล็ก ถูกใช้กันอย่างแพร่หลายในการจ่ายไฟฟ้าให้กับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก เช่น รีโมทคอนโทรล ไมโครโฟน และเครื่องตรวจจับควัน

ขั้วถ่าน 9V เป็นอุปกรณ์ที่ใช้เชื่อมต่อถ่าน 9V กับวงจรหรืออุปกรณ์ที่ต้องการใช้ไฟจากถ่านนี้ ขั้วถ่าน ประกอบด้วยขั้วบวกและขั้วลบที่เชื่อมต่อกับสายไฟ



ภาพที่ 2.10 Battery Alkaline 9 V

บทที่ 3

วิธีการดำเนินงาน

3.1 การพัฒนาระบบหุ่นยนต์เก็บฝุ่นอัตโนมัติ

จากการดำเนินการพัฒนาระบบหุ่นยนต์เก็บฝุ่นอัตโนมัติ สามารถแบ่งการทำงานออกได้เป็น

2 ส่วน คือ Hardware และ Software

3.1.1. การพัฒนาระบบหุ่นยนต์เก็บฝุ่นอัตโนมัติส่วน Hardware

- Microcontroller Arduino ร่วมกับโปรแกรม Arduino IDE เพื่อสร้างการทำงานของหุ่นยนต์เก็บฝุ่นอัตโนมัติ
- L298N Motor Driver Module ควบคุมการหมุนของล้อผ่านมอเตอร์ไฟฟ้า
- DC Ultrasonic Sensor จะวัดระยะห่างของสิ่งกีดขวางด้านหน้าเพื่อไม่ให้ชน หากพบสิ่งกีดขวางที่ระยะใกล้กว่าค่าที่กำหนด หุ่นยนต์จะหยุดและพิจารณาการเลี้ยว
- Micro Servo SG90 Motor 180-degree หมุนเซ็นเซอร์เพื่อตรวจสอบเส้นทางซ้ายและขวา ระบบจะเลือกทิศทางที่โล่งที่สุดและหมุนล้อไปในทิศทางนั้น
- DHT11 วัดอุณหภูมิและความชื้นภายในห้องที่หุ่นยนต์อยู่

3.1.2. การพัฒนาระบบหุ่นยนต์เก็บฝุ่นอัตโนมัติส่วน Software

- ใช้ Node-RED เป็นแพลตฟอร์มสำหรับการออกแบบและควบคุมระบบผ่านอินเทอร์เฟซแบบ Flow-Based Programming จะช่วยให้สามารถสร้าง Dashboard สำหรับแสดงผลข้อมูลจากเซ็นเซอร์ต่างๆ ได้แบบเรียลไทม์ รวมถึงควบคุมการทำงานของหุ่นยนต์จากระยะไกลผ่านอินเทอร์เน็ต
- ใช้ MQTT (Message Queuing Telemetry Transport) จาก EMQX broker เป็นโปรโตคอลสื่อสารสำหรับการส่งข้อมูลระหว่างตัวหุ่นยนต์และเซิร์ฟเวอร์หรืออุปกรณ์ IoT อื่นๆ ทำให้สามารถรับ-ส่งข้อมูลสถานะของหุ่นยนต์ เช่น ความชื้น การเปิดปิด

3.2 วิธีการดำเนินงาน

3.2.1 การออกแบบและการสร้างหุ่นยนต์เก็บฝุ่นอัตโนมัติ

- ออกแบบแผนผังการทำงานของระบบ (Flowchart)
- ออกแบบฮาร์ดแวร์ เช่น วงจรควบคุม (Circuit Diagram)
- ออกแบบซอฟต์แวร์ เช่น การประมวลผลข้อมูลและส่งข้อมูลผ่าน MQTT และ NODE-RED
- ออกแบบโมเดลหุ่นยนต์เก็บฝุ่นอัตโนมัติให้เหมาะสมกับการทำงานและพื้นที่ใช้งาน

3.2.2 การพัฒนาโปรแกรมและโมเดล


- เขียนโค้ดสำหรับหลบสิ่งกีดขวางโดยใช้ข้อมูลจาก Ultrasonic Sensor
- เขียนโค้ดสำหรับเชื่อมต่อกับ Wi-Fi และส่งข้อมูลไปยัง MQTT Broker
- พัฒนาโปรแกรมให้สามารถแสดงผลผ่าน Node-RED Dashboard
- สร้างและพัฒนาโมเดลหุ่นยนต์เก็บฝุ่นอัตโนมัติทั้งในส่วนของโครงสร้างและระบบควบคุม

3.2.3 การทดสอบระบบ

- ทดสอบการทำงานของ Ultrasonic Sensor ในการตรวจจับสิ่งกีดขวาง
- ทดสอบการเชื่อมต่อระหว่าง NODE-RED กับ MQTT เพื่อรับ-ส่งข้อมูลได้อย่างถูกต้อง
- ทดสอบความแม่นยำในการตรวจจับและการตัดสินใจเปลี่ยนเส้นทางของหุ่นยนต์

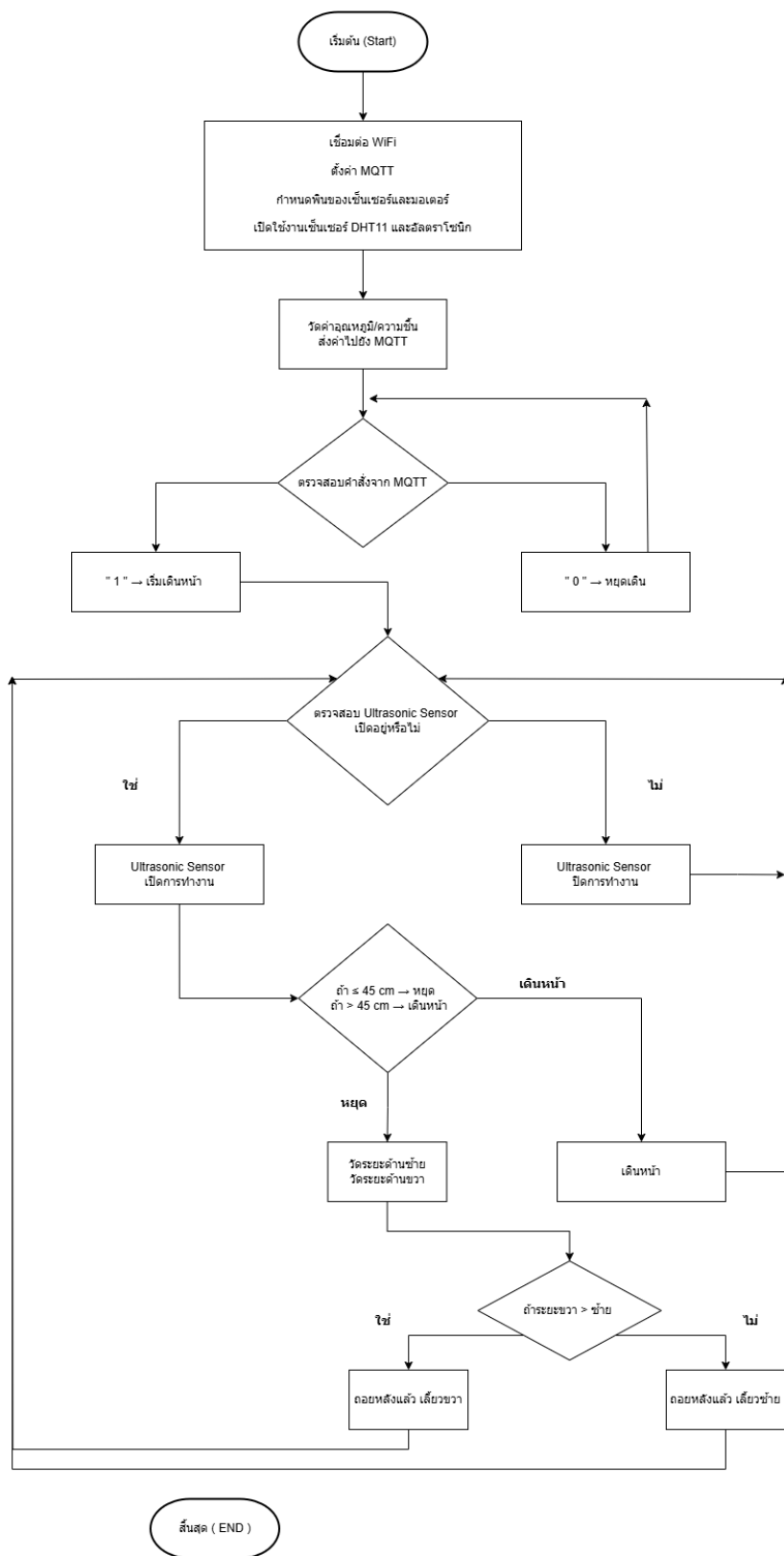
3.3 อุปกรณ์

รูปอุปกรณ์	ชื่อ	ราคา
	Arduino UNO R4	ฟรี
	L298N Drive Motor Module	65 บาท
	ถ่าน 9V	89 บาท
 เมีย-เมีย คู่-เมีย คู่-คู่	Jumper	ชุดละ 28 บาท ทั้งหมด 3 ชุด รวม 84 บาท
	Ultrasonic Sensor	37 บาท
	Micro Servo SG90 Motor 180-degree	86 บาท

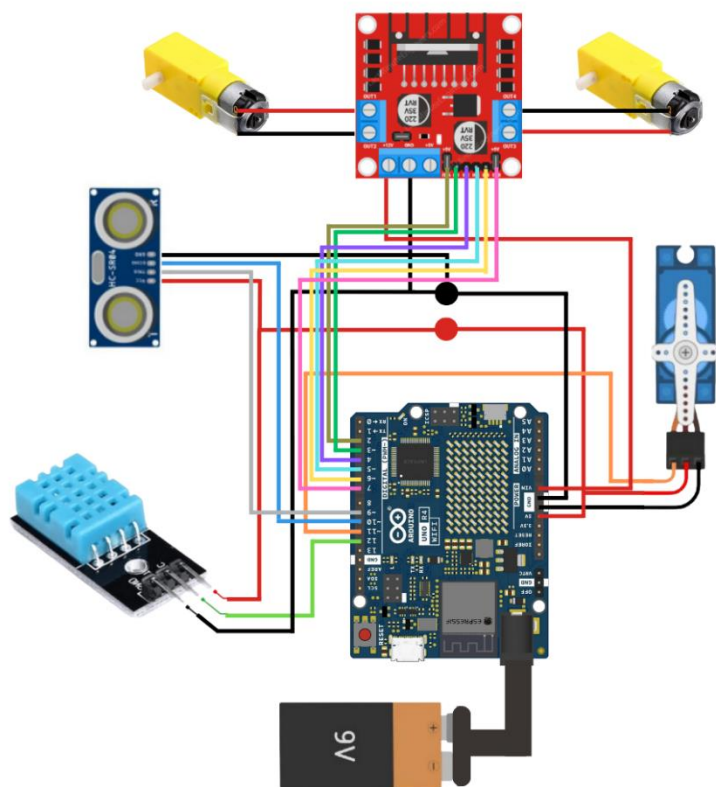
	Gear Motor	ตัวละ 28 บาท ทั้งหมด 2 ตัว รวม 56 บาท
	ล้อยาง	ตัวละ 40 บาท ทั้งหมด 2 ตัว รวม 80 บาท
	ไม้พลาสติก	120 บาท
	บานพับ	5 บาท
	ที่เก็บฝุ่น ลูกกลิ้งทำความสะอาด	36 บาท
	DHT 11	30 บาท

รวมงบประมาณที่ได้ใช้ในการทำระบบและโมเดล รวมทั้งหมด 688 บาท

3.4 หลักการทำงาน



3.5 การต่อวงจร



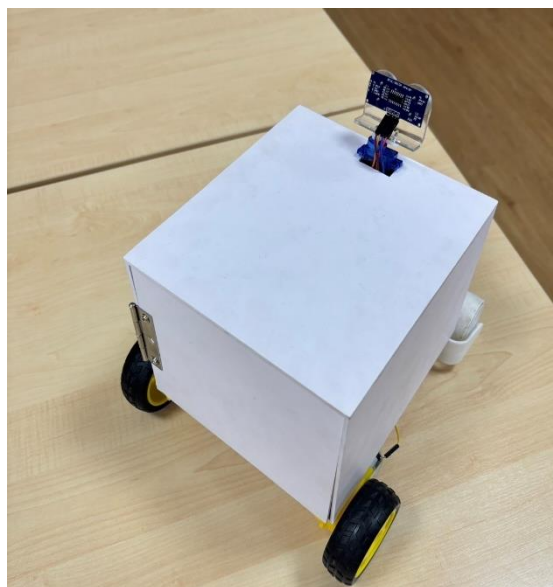
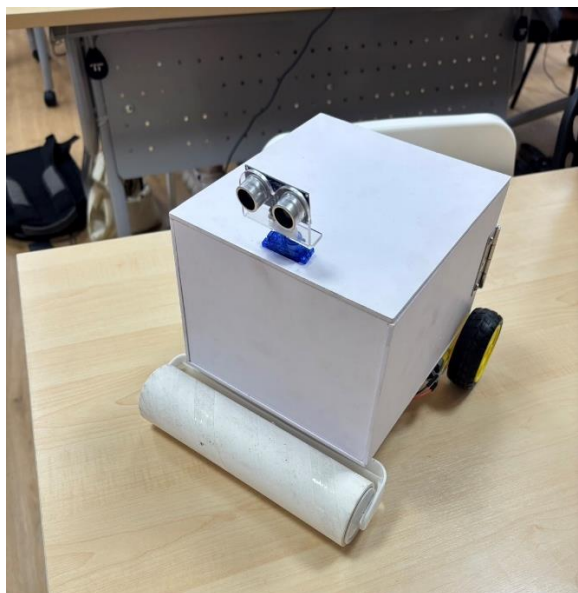
บทที่ 4

ผลการดำเนินงาน

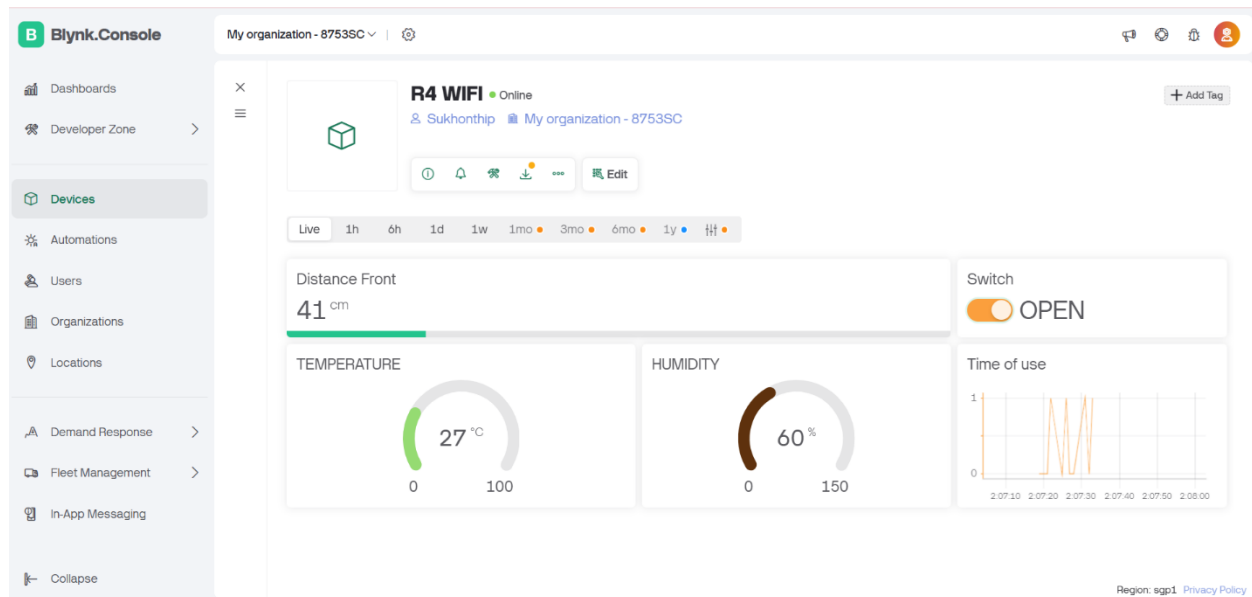
4.1 ผลการดำเนินงาน

ในการทดลอง หุ่นยนต์เก็บฝุ่นอัตโนมัติที่พัฒนาขึ้น เมื่อทดสอบการเดิน สามารถหลีกเลี่ยงสิ่งกีดขวาง โดยใช้ Ultrasonic Sensor และตรวจสอบการเชื่อมต่อกับระบบ IoT ผ่าน MQTT แสดงผลอุณหภูมิห้อง ได้อย่างแม่นยำ รวมถึงสามารถควบคุมการเปิดปิดหุ่นยนต์ผ่านทาง Node-RED นอกจากนี้ ยังมีโอกาสในการพัฒนาเพิ่มเติมเพื่อให้หุ่นยนต์มีความสามารถที่หลากหลายยิ่งขึ้นและตอบโจทย์การใช้งานในชีวิตประจำวันได้ดียิ่งขึ้น ในอนาคตอีกด้วย

4.2 รูปโมเดล



4.3 dashboard



บทที่ 5

สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

จากการพัฒนาระบบหุ่นยนต์เก็บฝุ่นอัตโนมัติ พบว่าการทำงานของระบบทั้งในส่วนของ Hardware และ Software สามารถทำงานได้อย่างมีประสิทธิภาพ โดยในส่วนของ Hardware การใช้งาน Microcontroller Arduino ร่วมกับ L298N Motor Driver และ Ultrasonic Sensor สามารถควบคุมการเคลื่อนที่ของหุ่นยนต์และตรวจจับสิ่งกีดขวางได้ดี รวมถึงการใช้งาน Micro Servo SG90 Motor ในการหมุนเพื่อเลือกเส้นทางที่เหมาะสม ส่วน DHT11 ใช้วัดอุณหภูมิและความชื้น ซึ่งเป็นข้อมูลเสริมสำหรับการทำงานของหุ่นยนต์ ในด้าน Software ระบบสามารถควบคุมหุ่นยนต์ผ่าน Node-RED และส่งข้อมูลผ่าน MQTT ได้อย่างราบรื่น ช่วยให้การสื่อสารระหว่างหุ่นยนต์และระบบเซิร์ฟเวอร์หรืออุปกรณ์ IoT อื่นๆ ทำงานได้อย่างมีประสิทธิภาพ นอกจากนี้ยังสามารถแสดงข้อมูลต่างๆ แบบเรียลไทม์ผ่านแดชบอร์ดของ Node-RED ได้อย่างสะดวกและรวดเร็ว

5.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับการพัฒนาในอนาคตคือ การเพิ่มความแม่นยำในการหลบหลีกสิ่งกีดขวางในสภาพแวดล้อมที่มีความซับซ้อนมากขึ้น เช่น การใช้เซ็นเซอร์ที่มีความละเอียดสูงกว่าเดิม เพื่อให้หุ่นยนต์สามารถหลีกเลี่ยงสิ่งกีดขวางได้แม่นยำยิ่งขึ้น นอกจากนี้ ควรพัฒนาระบบการเชื่อมต่อ Wi-Fi ให้มีความเสถียรในการใช้งานในพื้นที่ที่มีสัญญาณอินเทอร์เน็ตไม่ดี รวมถึงการพัฒนาแอปพลิเคชันมือถือสำหรับการควบคุมหุ่นยนต์จากระยะไกล ซึ่งจะช่วยเพิ่มความสะดวกในการใช้งานและควบคุมหุ่นยนต์ได้ทุกที่ทุกเวลา

บรรณานุกรม

- <https://www.tido.tech/index.php/product/dht11-temperature-humidity-sensor/>
- OASIS, (2019). *MQTT Version 5.0 Specification*. OASIS Standard. <https://mqtt.org/>
- https://www.researchgate.net/publication/337842014_Obstacle_detection_using_ultrasonic_sensor_for_a_mobile_robot
- *DHT11 Temperature and Humidity Sensor – User Manual*. (2019). Datasheet DHT11
- Node-RED Documentation: Node-RED (2025). *Flow-based Programming for the Internet of Things*. <https://nodered.org/docs/>
- Temperature and Humidity Sensing with DHT11. (2018). *Arduino Project Hub*. <https://create.arduino.cc/>
- <https://aws.amazon.com/what-is/mqtt/>
- <https://iot.jpnet.co.th/blynk/>
- <https://www.youtube.com/watch?v=kxojYz9kpt8>
- <https://www.arduino.cc/>
- <https://projecthub.arduino.cc/>
- <https://www.ros.org/>