



รายงาน
ถังขยะอัจฉริยะ
Automatic Trash Bin

จัดทำโดย
64200002 กัญทบวง เณรวงษ์
64200039 ชยากุล ธรรมสวัสดิ์
64200055 ณัฏฐ์ แจ่มจิตร
64200289 กฤษณ์ คงคา
64200358 วารินทร์ พรหมพิชัย

เสนอ
อาจารย์ศิลา ศิลิมาสกุล

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพร
เขตรอุดมศักดิ์จังหวัดชุมพร ปีการศึกษา 2566



รายงาน
ถังขยะอัจฉริยะ
Automatic Trash Bin

จัดทำโดย
64200002 กัญทบวง เณรวงษ์
64200039 ชยากุล ธรรมสวัสดิ์
64200055 ณัฐ์ แจ่มจิตร
64200289 กฤษณ์ คงคา
64200358 วารินทร์ พรหมพิชัย

เสนอ
อาจารย์ศิลา ศิธิมาสกุล

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพร
เขตอุดมศักดิ์จังหวัดชุมพร ปีการศึกษา 2566



COPYRIGHT 2023

DEPARTMENT OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

PRINCE OF CHUMPHON CAMPUS

รายงานปีการศึกษา 2566

สาขาวิชาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพรเขตอุดมศักดิ์ จังหวัดชุมพร
เรื่อง ถังขยะอัจฉริยะ

Automatic Trash Bin

ผู้จัดทำ

- | | | |
|----------------|-------------|-----------------------|
| 1. นายภูทบวง | ณรงษ์ | รหัสนักศึกษา 64200002 |
| 2. นายชยากุล | ธรรมสวัสดิ์ | รหัสนักศึกษา 64200039 |
| 3. นายณัฐ | แจ้งจิตร | รหัสนักศึกษา 64200055 |
| 4. นายกฤษณ์ | คงคา | รหัสนักศึกษา 64200289 |
| 5. นายวารินทร์ | พรหมพิชัย | รหัสนักศึกษา 64200358 |

.....อาจารย์ที่ปรึกษา
(ว่าที่ร้อยตรี ศิลา ศิริมาสกุล)

ชื่อปริญญาบัตร	ถึงขยะอัจฉริยะ		
นักศึกษา	นายภูทรวง	ณรงษ์	รหัสนักศึกษา 64200002
	นายชยากุล	ธรรมสวัสดิ์	รหัสนักศึกษา 64200039
	นายณัฐ	แจ้งจิตร	รหัสนักศึกษา 64200055
	นายกฤษณ์	คงคา	รหัสนักศึกษา 64200289
	นายวารินทร์	พรหมพิชัย	รหัสนักศึกษา 64200358
อาจารย์ที่ปรึกษา	อาจารย์ศิลา	ศิริมาสกุล	
หลักสูตร	วิศวกรรมคอมพิวเตอร์		
หลักสูตร	วิศวกรรมคอมพิวเตอร์		
ปีการศึกษา	2566		

บทคัดย่อ

โครงการ "ถึงขยะอัจฉริยะ" นำเสนอการนวัตกรรมในการจัดการขยะที่ทันสมัยและรวมระบบสำคัญสองรายการ การตรวจนับปริมาณขยะและการเปิดปิดฝาอัตโนมัติ.

ถึงขยะอัจฉริยะใช้เทคโนโลยีเซนเซอร์เพื่อตรวจนับปริมาณขยะที่มีอยู่ในถัง โดยประเมินค่าเรียลไทม์โดยอัตโนมัติ ทำให้ไม่ต้องเดาเมื่อถึงเวลาที่จะนำขยะออก ที่สำคัญคือเมื่อปริมาณขยะถึงระดับที่กำหนดล่วงหน้าจะแจ้งเตือนไปยังไลน์,ระบบเปิดปิดฝาโดยอัตโนมัติโดยไม่ต้องการสัมผัส

นวัตกรรมนี้ไม่เพียงเพิ่มความสะดวกในการจัดการขยะ แต่ยังช่วยลดความสกปรกและสะอาดของรอบขยะ ป้องกันขยะที่เต็มถังรั่วไหลออกมาและลดการแพร่ระบาดของเชื้อโรคที่มากับขยะ

ผู้ใช้งานสามารถตรวจสอบสถานะของถังขยะและตั้งค่าให้เปิดปิดโดยอัตโนมัติผ่านแอปพลิเคชันที่ใช้งานง่าย โครงการนี้ไม่เพียงเพียงทำให้ชีวิตเป็นสะดวกและคงสภาพแวดล้อมน่าอยู่มากขึ้น แต่ยังเป็นการแสดงการพิสูจน์เพื่อเป็นส่วนสำคัญในการลดปัญหาขยะและสร้างสุขภาพให้กับสิ่งแวดล้อม.

Project Title	Automatic Trash Bin	
Students	Mr. Kottaboung Nanvong	Student ID 64200002
	Mr. Chayakul Tumsawat	Student ID 64200039
	Mr. Nutt Jangjit	Student ID 64200055
	Mr. Krisn Kongkar	Student ID 64200289
	Mr. Warin Prompichai	Student ID 64200358
Advisor	Mr. Silar Sirimasakul	
Degree	Computer Engineering	
Program in	Computer Engineering	
Academic Year	2023	

ABSTRACT

The "Auto Trash Bin" project introduces innovation in contemporary waste management, integrating two vital systems waste quantity detection and automated lid opening.

The Auto Trash Bin utilizes sensor technology to detect the quantity of waste present in the bin, providing real-time assessment without the need for guesswork. Significantly, when the waste quantity reaches a predetermined threshold, an automatic alert is sent via notification, prompting the system to open and close the lid without requiring physical contact.

This innovation not only enhances waste management convenience but also contributes to reducing litter and promoting cleanliness in the vicinity of the trash bin. It prevents overflowing of waste, minimizing contamination and mitigating the spread of diseases associated with waste.

Users can monitor the status of the trash bin and configure automatic lid opening and closing through a user-friendly mobile application. This project not only simplifies daily life but also plays a crucial role in addressing waste management challenges and fostering environmental well-being.

คำนำ

การจัดทำโครงการสิ่งประดิษฐ์ “ถังขยะอัจฉริยะ” นี้สำเร็จได้ด้วยความร่วมมือจากสมาชิกทุกคนใน กลุ่ม รวมไปถึงการได้รับความช่วยเหลือ และสนับสนุนจากหลาย ๆ ท่าน อันดับแรกต้องขอขอบคุณอาจารย์ที่ปรึกษาโครงการ และสอนให้คณะผู้จัดทำมีความรู้ความสามารถ และท าให้เกิดโครงการนี้ขึ้น ที่ให้ความสนับสนุนสถานที่ในการทำโครงการ ให้ความช่วยเหลือในด้านอุปกรณ์ คณะผู้จัดทำจึงขอขอบพระคุณทุกท่านที่ได้สนับสนุนการทำงาน จนทำให้การจัดทำรายงานครั้งนี้

สำเร็จลุล่วงได้ด้วยดี การศึกษาค้นคว้าครั้งนี้ คณะผู้จัดทำขอขอบ แต่ผู้มีพระคุณทุกท่าน และขอกราบขอบพระคุณมา ณ โอกาสนี้

คณะผู้จัดทำ

ตุลาคม 2566

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
คำนำ.....	ค
สารบัญ.....	ง
บทที่ 1	1
บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของรายงาน.....	1
1.3 สมมติฐานการศึกษา.....	2
1.4 ขอบเขตของรายงาน.....	2
1.5 ประโยชน์ที่ได้รับจากรายงาน.....	2
1.6 ขั้นตอนการดำเนินงาน	2
บทที่ 2	4
เอกสารและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 Servo Motor	4
2.2 Ultrasonic sensor	5
2.3 ESP32.....	6
2.4 Infrared (IR) Sensor.....	7
2.5 LCD (Liquid Crystal Display).....	9
บทที่ 3	12
การออกแบบการทดลองและการสร้าง	12
3.1 เครื่องมือ.....	12
3.2 Block Diagram ของถังขยะอัจฉริยะ.....	12
บทที่ 4	17
วิธีการและการทดลอง.....	17
4.1 การทดลองเปิด-ปิดฝาถัง	17

4.2 การทดลองทดสอบเซนเซอร์ตรวจจับวัตถุ.....	19
4.3 การทดสอบระยะของเซนเซอร์ตรวจจับวัตถุ.....	20
บทที่ 5	22
บทสรุปและข้อเสนอแนะ.....	22
5.1 สรุปผลการทดลอง.....	22
5.2 ปัญหาและอุปสรรค.....	23
5.3 ข้อเสนอแนะและวิธีแก้ปัญหา.....	23
เอกสารอ้างอิง	24
ภาคผนวก ก.....	25
โปรแกรมของระบบควบคุมการทำงานของถังขยะอัจฉริยะ.....	25
โค้ดการทำงานของถังขยะ	26
การทำงานของ Servo-motor.....	26
การทำงานของ IR-Sensor.....	30
ภาคผนวก ข.....	33
โปรแกรมของระบบควบคุมการทำงานของถังขยะอัจฉริยะ.....	33
ภาคผนวก ค.....	41
คู่มือการใช้อุปกรณ์(Datasheets).....	41
ประวัติผู้เขียน.....	58

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ในปัจจุบันนี้ปัญหาขยะล้นเมืองในประเทศไทยยังคงเป็นปัญหาอย่างรุนแรงที่ต้องการการแก้ไขอย่างเร่งด่วน การเพิ่มปริมาณขยะที่สะสมอยู่ในถังขยะที่ถูกทิ้งอย่างไม่ถูกต้องเป็นปัญหาหลักที่ทำให้ถังขยะเต็ม เต็มไปด้วยขยะและเกือบเป็นไปไม่ได้เลยที่จะจัดการ นี่ทำให้เป็นแหล่งกำเนิดของปัญหาสาธารณสุขและเชื้อโรค แม้ว่าการจัดการขยะในหน่วยงานหรือองค์กรสาธารณสุข รัฐอาจจัดการได้อย่างเหมาะสม แต่ขยะในพื้นที่สาธารณะทั่วไปยังมีการจัดการไม่ถูกต้องและการควบคุมที่ไม่เพียงพอ เป็นภาพที่พบเห็นบ่อยในการใช้งานทั่วไป

ความสำคัญของโครงการนี้คือการเปิดทางให้ประชาชนเข้าใจและมีสติที่ดีในการจัดการขยะ การใช้เทคโนโลยีเซนเซอร์ในถังขยะช่วยเพิ่มควมมีสติในการจัดการขยะที่ทันสมัย ระบบตรวจจับปริมาณขยะและแจ้งเตือนไปยังไลน์ช่วยให้ผู้ใช้ทราบเมื่อถังขยะเต็มและต้องการการบริหารจัดการ ระบบเปิดปิดฝาอัตโนมัติจะทำให้ถังขยะเปิดหรือปิดโดยอัตโนมัติเพื่อรักษาความสะอาดและป้องกันการแพร่ระบาดของเชื้อโรคที่อาจมาจากขยะโครงการนี้สร้างแรงบันดาลใจในการฟื้นฟูและวิเคราะห์การจัดการขยะอย่างระมัดระวังและคำนึงถึงสิ่งแวดล้อมในสังคมของเรา

โครงการนี้ไม่เพียงเพิ่มความสะดวกในการจัดการขยะ แต่ยังเป็นการสร้างความตระหนักในการเป็นส่วนสำคัญในการลดปัญหาขยะและสร้างสุขภาพให้กับสิ่งแวดล้อมและสังคมของเรา

บทความนี้เน้นถึงปัญหาขยะล้นเมืองในประเทศไทยและเสนอโครงการที่มุ่งหน้าที่ในการแก้ไขปัญหา โดยเน้นความสำคัญของการเริ่มต้นที่ตัวเราเองในการจัดการขยะและการใช้เทคโนโลยีเพื่อช่วยในการแก้ปัญหาขยะในสังคม

1.2 วัตถุประสงค์ของรายงาน

- 1.2.1 เพื่อสร้างถังขยะที่สะดวกและใช้งานง่าย โครงการนี้มุ่งหมายเพื่อพัฒนาถังขยะที่มีความสะดวกและใช้งานง่ายเพื่อเพิ่มความสะดวกในการจัดการขยะ
- 1.2.2 เพื่อลดปริมาณขยะที่เก็บอยู่ในถังเป็นเวลานาน: โครงการนี้เป้าหมายในการลดปริมาณขยะที่สะสมอยู่ในถังขยะโดยการระบบเปิดปิดฝาทันทีอัตโนมัติ โดยส่งข้อมูลจากเซนเซอร์
- 1.2.3 เพื่อควบคุมปริมาณขยะอย่างมีประสิทธิภาพ โครงการนี้มุ่งหมายที่การตรวจจับปริมาณขยะในถังเพื่อตรวจสอบว่าถังเต็มหรือยัง และทำให้การบริหารจัดการขยะได้อย่างมีประสิทธิภาพ
- 1.2.4 เพื่อสร้างระบบการแจ้งเตือนเมื่อจำเป็น โครงการนี้จัดทำเพื่อสร้างระบบที่สามารถแจ้งเตือนการเปิดปิดถังหรือปริมาณขยะในขณะนั้นให้ผู้ดูแลได้อย่างรวดเร็วและมีประสิทธิภาพ

ดังนั้น รายงานนี้มีวัตถุประสงค์ที่นำเสนอโครงการ "ถึงขยะอัจฉริยะ" และเพิ่มความตระหนักเกี่ยวกับปัญหาขยะในสังคมเพื่อส่งเสริมการจัดการขยะที่ยั่งยืนและเป็นประโยชน์ต่อสิ่งแวดล้อมและสังคมในประเทศไทย

1.3 สมมติฐานการศึกษา

ระบบนี้มีความสามารถในการตรวจจับปริมาณขยะในถังขยะอย่างแม่นยำและแจ้งเตือนให้ทราบเมื่อถังขยะเริ่มเต็ม และระบบเปิดปิดฝาถังขยะโดยอัตโนมัติเพื่อรักษาความสะอาดและป้องกันการแพร่ระบาดของเชื้อโรคที่อาจมาจากขยะ การใช้เทคโนโลยีเซนเซอร์ในระบบจัดการขยะนี้เป็นการเข้ากันกับการพัฒนาทางเทคโนโลยีและการปรับปรุงระบบให้มีประสิทธิภาพและเชื่อมโยงกับเศรษฐกิจแห่งอนาคตที่ยั่งยืนและเป็นมิตรกับสิ่งแวดล้อม การสมมติฐานนี้เน้นความสำคัญของการใช้เทคโนโลยีเพื่อแก้ไขปัญหาทางสิ่งแวดล้อมและสังคมในระยะยาวและปรับปรุงระบบการจัดการขยะให้มีประสิทธิภาพมากขึ้นในอนาคต

1.4 ขอบเขตของรายงาน

- 1.4.1 ระบบจัดการขยะสามารถจัดกลุ่มประเภทของขยะอ้างอิงจากขยะที่พบบ่อยทั่วไป เพื่อความคล่องตัวและความสะดวกในการจัดการขยะที่แตกต่างกัน.
- 1.4.2 ถังขยะที่ได้รับการพัฒนาเหมาะสำหรับการใช้งานภายในอาคารหรือสถานที่ปิด เพื่อให้การจัดการขยะในพื้นที่ปิดใช้งานได้อย่างมีประสิทธิภาพและสะดวกสบาย.
- 1.4.3 โครงการนี้เน้นการออกแบบและพัฒนาระบบที่ใช้ในการตรวจจับปริมาณขยะและระบบเปิดปิดฝาอัตโนมัติในถังขยะ ระบบนี้จะช่วยให้ผู้ดูแลสามารถตรวจสอบสถานะของปริมาณขยะในปัจจุบันได้อย่างทันทั่วถึงและจัดการขยะได้อย่างมีประสิทธิภาพ.
- 1.4.4 ขอบเขตของเทคโนโลยี รายงานจะระบุขอบเขตของเทคโนโลยีที่ใช้ในโครงการ "ถึงขยะอัจฉริยะ" และเปรียบเทียบกับเทคโนโลยีที่ใช้อยู่ในปัจจุบัน

1.5 ประโยชน์ที่ได้รับจากรายงาน

- 1.5.1 เพื่อสร้างถังขยะที่สะดวกและใช้งานง่าย โครงการนี้จัดทำเพื่อพัฒนาถังขยะที่มีความสะดวกและใช้งานง่าย เพื่อเพิ่มความสะดวกในการจัดการขยะ.
- 1.5.2 เพื่อลดปัญหาการทิ้งขยะที่เต็มถัง โครงการนี้มุ่งหมายเพื่อลดปัญหาการทิ้งขยะในถังที่เต็มและล้น โดยระบบตรวจจับปริมาณขยะและแจ้งเตือนเมื่อถังเต็มช่วยให้ป้องกันปัญหาการทิ้งขยะล้นออกจากถังขยะ

1.6 ขั้นตอนการดำเนินงาน

- 1.6.1 ค้นหาหาข้อมูลของอุปกรณ์รวมทั้งหลักการทำงานต่าง ๆ ของวงจรที่นำมาใช้งาน
- 1.6.2 ออกแบบโครงสร้างของชิ้นงาน
- 1.6.3 จัดหาอุปกรณ์ และสั่งซื้อวัสดุที่ใช้ในการทำโครงการ
- 1.6.4 ทดลองเขียนโปรแกรม (Program) เพื่อใช้ในการควบคุมการทำงานของชิ้นงาน
- 1.6.5 นำชิ้นงานมาประกอบร่วมกับวงจรอิเล็กทรอนิกส์
- 1.6.6 ทดลองใช้งานจริงรวมถึงการแก้ปัญหาและอุปสรรคที่เกิดจากการทำโครงการ
- 1.6.7 สรุปผลการทำโครงการ

ตารางที่ 1.1 แผนการดำเนินงาน

ขั้นตอนการดำเนินงาน	ระยะเวลา															
	กรกฎาคม				สิงหาคม				กันยายน				ตุลาคม			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1) ประชุมเพิ่มหาหัวข้อโครงงาน																
2) นำเสนอหัวข้อโครงงาน																
3) วางแผนการออกแบบโครงงาน																
4) รวบรวมและสั่งอุปกรณ์																
5) เขียน diagram เพื่อวางแผนการเขียน code																
7) นำเสนอความคืบหน้า																
8) ทดลองการใช้ IR sensor																
9) ทดลองการใช้ ultrasonic กับ servo																

บทที่ 2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ในกระบวนการรายงานเรื่อง "ถังขยะอัจฉริยะ" คณะผู้จัดทำได้ดำเนินการรวบรวมแนวคิด ทฤษฎี และหลักการจากเอกสารต่าง ๆ และงานวิจัยที่เกี่ยวข้อง เพื่อสนับสนุนและข้อมูลการพัฒนาของรายงาน ดังต่อไปนี้

2.1 Servo Motor

Servo Motor (SG90) เป็นอุปกรณ์ที่มีความหลากหลายในการควบคุมเครื่องจักรและระบบการทำงานต่าง ๆ ให้เป็นไปตามความต้องการ โดยสามารถควบคุมความเร็ว (Speed) แรงบิด (Torque) และตำแหน่ง (Position) ของหมุนได้ตามค่าที่กำหนดไว้ล่วงหน้า และสามารถทำให้หมุนไปยังตำแหน่งที่กำหนดได้อย่างแม่นยำด้วยการใช้ระบบควบคุมแบบป้อนกลับ (Feedback Control) ซึ่งให้ข้อมูลตัวเรียกคือความต้องการและตัวเรียกคือตำแหน่งปัจจุบันและจัดการให้ Servo Motor หมุนไปที่ตำแหน่งที่ต้องการอย่างแม่นยำและมีความแม่นยำสูง



รูปที่ 2.1 Servo motor

(ที่มา: <https://th.rs-online.com>)

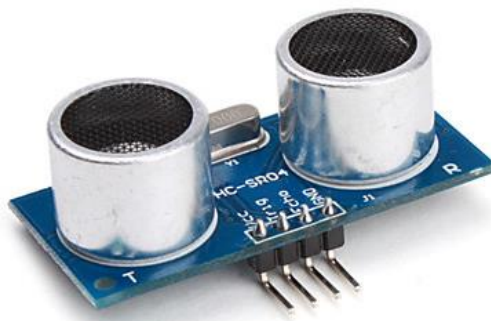
Servo Motor SG90 มีคุณสมบัติเด่นคือขนาดเล็กกะทัดรัดรูปแบบใช้งานง่าย มีแรงบิดในช่วง 1.2-1.4 kg/cm และใช้แรงดันไฟเข้าระหว่าง 4.8-7.2 V โดยสายกราวด์สีน้ำตาล สายไฟเข้าสีแดง และสายสัญญาณอินพุตสีส้ม และมีความสามารถในการหมุนไปยังตำแหน่ง 0-180 องศา หรือถ้าต้องการหมุนไปต่อเนื่อง 360 องศาสามารถใช้ระบบควบคุมพิเศษเช่น ใช้ตัวต้านทานขนาด 2.2 k/ohm.

หลักการทำงานของ Servo Motor เน้นควบคุมความแม่นยำของตำแหน่งโดยการส่งสัญญาณพัลส์เข้ามาที่ Servo Motor โดยภายใน Servo Motor จะมีวงจรอิเล็กทรอนิกส์ควบคุม (Electronic Control System) ซึ่งจะอ่านและประมวลผลความกว้างของสัญญาณพัลส์ที่ส่งเข้ามา และจะแปลค่านั้นเป็นตำแหน่งองศาที่ต้องการให้ Servo Motor หมุนไปยังตำแหน่งนั้น การทำงานนี้ใช้ Position Sensor เพื่อวัดค่ามุมที่ Servo Motor กำลังหมุนอยู่และส่ง Feedback กลับมาให้วงจรควบคุม จากนั้นวงจร

ควบคุมจะเปรียบเทียบค่า Feedback กับค่าอินพุตเพื่อควบคุมให้ Servo Motor หมุนไปยังตำแหน่งที่ต้องการอย่างแม่นยำและมีความแม่นยำสูง.

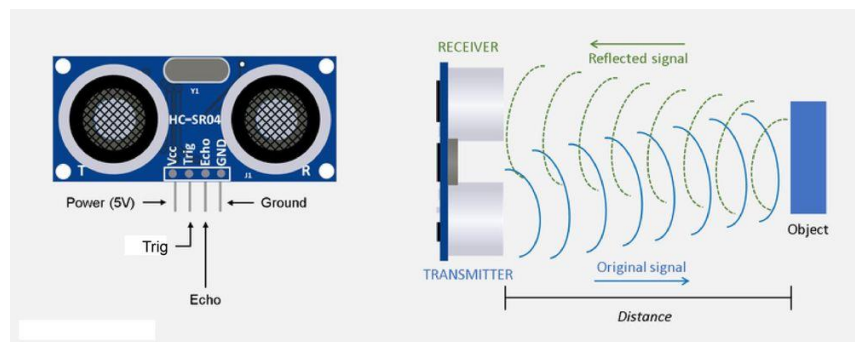
2.2 Ultrasonic sensor

Ultrasonic sensor คือ อุปกรณ์สำหรับวัดระดับหรือระยะทางชนิดหนึ่งโดยใช้คลื่น Ultrasonic ซึ่งอาศัยหลักการสะท้อนของคลื่นความถี่สูง Ultrasonic โดยอุปกรณ์จะปล่อยคลื่น Ultrasonic ให้กระทบกับวัตถุ จากนั้นรอคลื่น Ultrasonic สะท้อนกลับมาที่เซ็นเซอร์เพื่อคำนวณหา ระยะทางที่วัดได้ นอกเหนือจาก Ultrasonic sensor แล้ว ยังมีเซ็นเซอร์ชนิดอื่นๆอีกที่ใช้ในการวัด ระยะได้แก่ Radar sensor, Hydrostatic sensor เป็นต้น



รูปที่ 2.2 Ultrasonic sensor

(ที่มา: <https://store-usa.arduino.cc>)

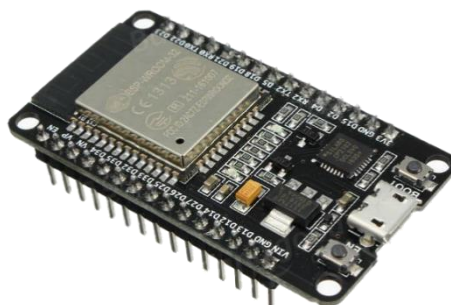


รูปที่ 2.2 การทำงานของ Ultrasonic Sensor

(ที่มา: <https://mall.factomart.com/>)

2.3 ESP32

ESP32 เป็น Micro Controller ที่รองรับการเชื่อมต่อ Wi-Fi, Bluetooth – BLE ในตัว ภาษาที่ใช้ในการพัฒนาโปรแกรมคือ ภาษา C หรือ Python ภาษา Python ต้องทำการอัปเดตเฟิร์มแวร์ให้รองรับ Python การพัฒนาโปรแกรมขึ้นอยู่กับผู้ที่พัฒนา โปรแกรม IDE ที่ใช้พัฒนาคือ Arduino IDE หรือ Visual Studio สำหรับ Visual Studio จำเป็นต้องติดตั้ง Plugin Espressif IDF หรือ Platform I/O IDE และต้อง Enable (Arduino)



รูปที่ 2.3 ESP32 NodeMCU

(ที่มา: <https://techtalk2apply.com>)

ESP32 คือ Wi-Fi microcontroller ที่ถูกพัฒนาต่อจาก ESP8266 โดยเพิ่ม CPU เป็น 2 core, Wi-Fi ที่เร็วขึ้น, มีขา GPIO ให้ใช้งานมากขึ้น และรองรับ Bluetooth อีกด้วย นอกจากนี้ ESP32 ยังมาพร้อมกับ touch-sensitive pins ที่สามารถใช้ปลุก ESP32 จากโหมด deep sleep และยังมี hall effect sensor และ temperature sensor ด้วย

ESP32 เป็นชื่อของไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi และ Bluetooth 4.2 BLE ในตัว ผลิตโดยบริษัท Espressif จากประเทศจีน โดยราคา ณ ที่เขียนบทความอยู่นี้ มีราคาไม่เกิน 500 บาท (บอร์ดพัฒนาสำเร็จรูป) โดยตัวไอซี ESP32 มีสเปคโดยละเอียด ดังนี้

ชิพที่ใช้สถาปัตยกรรม Tensilica LX6 แบบ 2 แกนสมอง สัญญาณนาฬิกา 240MHz มีแรมในตัว 512KB รองรับการเชื่อมต่อรอมภายนอกสูงสุด 16MB มาพร้อมกับ WiFi มาตรฐาน 802.11 b/g/n รองรับการใช้งานทั้งในโหมด Station softAP และ Wi-Fi direct มีบลูทูธในตัว รองรับการใช้งานในโหมด 2.0 และโหมด 4.0 BLE ใช้แรงดันไฟฟ้าในการทำงาน 2.6V ถึง 3V ทำงานได้ที่อุณหภูมิ -40°C ถึง 125°C นอกจากนี้ ESP32 ยังมีเซ็นเซอร์ต่าง ๆ มาในตัวด้วย ดังนี้

วงจรรองสัญญาณรบกวนในวงจรขยายสัญญาณเซ็นเซอร์แม่เหล็ก เซ็นเซอร์สัมผัส (Capacitive touch) รองรับ 10 ช่อง รองรับการเชื่อมต่อคลิสตอล 32.768kHz สำหรับใช้กับส่วนวงจรนับเวลาโดยเฉพาะ ขาใช้งานต่าง ๆ ของ ESP32 รองรับการเชื่อมต่อต่าง ๆ ดังนี้

มี GPIO จำนวน 32 ช่อง รองรับ UART จำนวน 3 ช่อง รองรับ SPI จำนวน 3 ช่อง รองรับ I2C จำนวน 2 ช่อง รองรับ ADC จำนวน 12 ช่อง รองรับ DAC จำนวน 2 ช่อง รองรับ I2S จำนวน 2

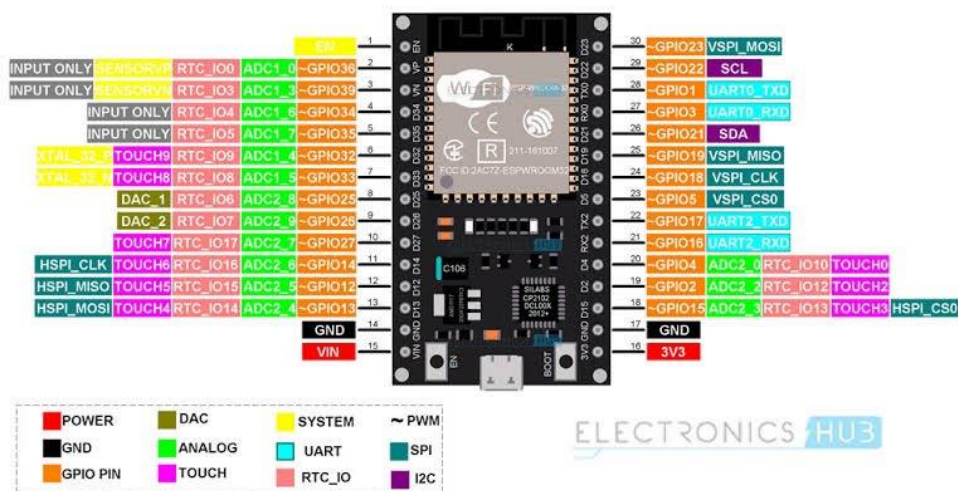
ช่อง รองรับ PWM / Timer ทุกช่อง รองรับการทำงานเชื่อมต่อกับ SD-Card นอกจากนี้ ESP32 ยังรองรับ ฟังก์ชันเกี่ยวกับความปลอดภัยต่าง ๆ ดังนี้

รองรับการเข้ารหัส WiFi แบบ WEP และ WPA/WPA2 PSK/Enterprise
มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว
ในด้านประสิทธิภาพการใช้งาน ตัว ESP32 สามารถทำงานได้ดี โดยรับ – ส่ง ข้อมูลได้ความเร็วสูงสุดที่ 150Mbps เมื่อเชื่อมต่อแบบ 11n HT40 ได้ความเร็วสูงสุด 72Mbps เมื่อเชื่อมต่อแบบ 11n HT20 ได้ความเร็วสูงสุดที่ 54Mbps เมื่อเชื่อมต่อแบบ 11g และได้ความเร็วสูงสุดที่ 11Mbps เมื่อเชื่อมต่อแบบ 11b เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ – ส่งข้อมูลได้ด้วยความเร็ว 135Mbps ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5uA

จะเห็นได้ว่า ในราคาไม่ถึง 500 บาท (บอร์ดพัฒนาสำเร็จรูป) และโมดูลเปล่าราคาไม่ถึง 400 บาท สามารถให้ประสิทธิภาพได้เกินราคา ด้วยเหตุนี้ ESP32 จึงเหมาะสำหรับนำมาใช้งานมาก ด้วยเหตุผลทางด้านราคา และประสิทธิภาพที่ได้

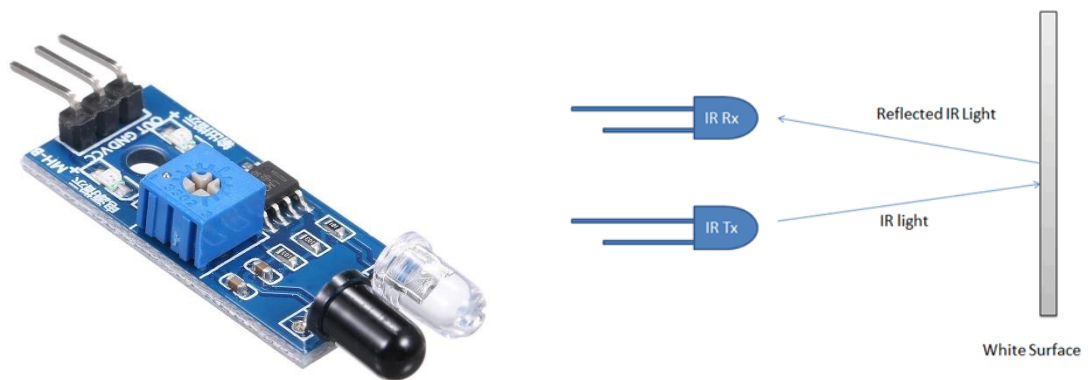
2.4 Infrared (IR) Sensor

Infrared (IR) Sensor เซ็นเซอร์แสงอินฟราเรดใช้ในการตรวจจับวัตถุโดยใช้การส่งและรับความเข้มของแสงอินฟราเรด แบ่งเป็น IR Reflective Sensor เซ็นเซอร์ที่ต้องการสิ่งที่จะสะท้อนแสงกลับมา และ IR light Sensor เซ็นเซอร์ที่ตรวจจับระยะห่างจากวัตถุตามการเปลี่ยนแปลงความเข้มของแสง



รูปที่ 2.4 ESP32 Pinout | ESP-WROOM-32 Pinout

(ที่มา: <https://www.electronicshub.org/esp32-pinout/>)



รูปที่ 2.5 Infrared (IR) Sensor และ ภาพแสดงการทำงาน

ที่มา: <https://www.artronshop.co.th>)

Compare sensor

Sensors	เหมาะสมกับ	ระยะตรวจจับวัตถุ จากลำตัวชี้	ระยะตรวจจับวัตถุ จากการทดลอง	หมายเหตุ
FC-51	การวัดระยะใกล้ มีการกระจายตัวน้อย	2 - 30 CM	2 - 27 CM 22 CM เริ่มจับไม่ค่อยได้	Infrared
E18-D80NK	การวัดระยะใกล้ มีการกระจายตัวน้อย	3 - 80 CM	2 - 70 CM 60 CM เริ่มจับไม่ค่อยได้	Infrared
US-100	การวัดระยะไกล มีการกระจายตัวมาก	2 - 450 CM	2-420 CM 420 CM เริ่มจับระยะไม่นิ่ง	Ultrasonic มีการวัด อุณหภูมิในตัวด้วย



รูปที่ 2.6 Compare sensor

(ที่มา: <https://medium.com/@projectid030>)

2.5 LCD (Liquid Crystal Display)

เทคโนโลยีมอนิเตอร์ LCD ย่อมาจาก Liquid Crystal Display ซึ่งเป็นจอแสดงผลแบบ (Digital) โดยภาพที่ปรากฏขึ้นเกิดจากแสงที่ถูกปล่อยออกมาจากหลอดไฟด้านหลังของจอภาพ (Black Light) ผ่านชั้นกรองแสง (Polarized filter) แล้ววิ่งไปยัง คริสตัลเหลวที่เรียงตัวด้วยกัน 3 เซลล์คือ แสงสีแดง แสงสีเขียวและแสงสีน้ำเงิน กลายเป็นพิกเซล (Pixel) ที่สว่างสดใสเกิดขึ้น

เทคโนโลยีที่พัฒนามาใช้กับ LCD นั้นแบ่งออกเป็น 2 ประเภทคือ

Passive Matrix หรือที่เรียกว่า Super-Twisted Nematic (STN) เป็นเทคโนโลยีแบบเก่าที่ให้ความคมชัดและความสว่างน้อยกว่า ใช้ในจอโทรศัพท์มือถือทั่วไปหรือจอ Palm ขาวดำเป็นส่วนใหญ่

Active Matrix หรือที่เรียกว่า Thin Film Transistors (TFT) สามารถแสดงภาพได้คมชัดและสว่างกว่าแบบแรก ใช้ในจอมอนิเตอร์หรือนิตบุ๊ก

เทคโนโลยี TFT LCD Monitor

TN + Film (Twisted Nematic + Film)

Twisted Nematic (TN) คือสารประเภทนี้จะมีการจัดโครงสร้างโมเลกุลเป็นเกลียว แต่ถ้าเราผ่านกระแสไฟฟ้าเข้าไปมันก็จะคลายตัวออกเป็นเส้นตรง เราใช้ปรากฏการณ์นี้เป็นตัวกำหนดว่าจะให้แสงผ่านได้หรือไม่ได้ Twisted Nematic (TN) ผลึกเหลวชนิดนี้จะให้เราสามารถเปลี่ยนทิศทางการสั่นของคลื่นแสงได้ 90° ถึง 180° คือเปลี่ยนจากแนวตั้งให้กลายเป็นแนวนอน หรือเปลี่ยนกลับกันจากแนวนอนให้เป็นแนวตั้งก็ได้ ด้วยจุดนี้เองทำให้การค่า Response Time (ค่าตอบสนองสัญญาณเทียบกับเวลา) มีค่าสูง

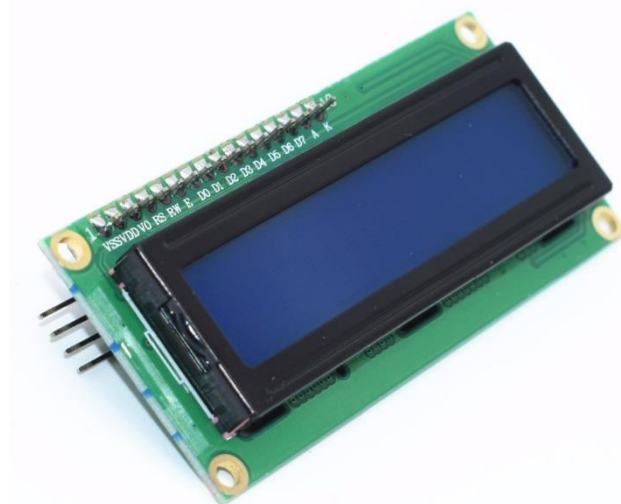
IPS (In-Plane Switching or Super-TFT)

การจัดโครงสร้างของผลึกจากเดิมที่วางไว้ตามแนวขนานกับแนวตั้ง (เทียบกับระนาบ) เปลี่ยนมาเป็นวางตามแนวขนานกับระนาบ เรียกจอชนิดนี้ว่า IPS (In-Plane Switching or Super-TFT) จากเดิมขั้วไฟฟ้าจะอยู่คนละด้านของผลึกเหลวแต่แบบนี้จะอยู่ด้านเดียวกันปะหัวท้าย เพราะย้ายแนวของผลึกให้ตั้งขึ้น (เมื่อมองจากมุมมองของคนดูจอ) เป้าหมายเพื่อออกแบบมาแก้ไขการที่มุมของผลึกเหลวจะเปลี่ยนไปเมื่อมันอยู่ห่างจากขั้วไฟฟ้าออกไป ปัญหานี้ทำให้จอมีมุมมองที่แคบมาก จอชนิด IPS จึงทำให้สามารถมีมุมมองที่กว้างขึ้น แต่ข้อเสียของจอชนิดนี้ก็คือต้องใช้ทรานซิสเตอร์สองตัวต่อหนึ่งจุดทำให้เปลืองมาก นอกจากนั้นการที่มีทรานซิสเตอร์เยอะกว่าเดิมทำให้แสงจากด้านหลังผ่านได้น้อยลง ทำให้ต้องมี Backlight ที่สว่างกว่าเดิม ความสิ้นเปลืองก็มากขึ้นอีกด้วย

MVA (Multi-Domain Vertical Alignment)

บริษัท Fujitsu ค้นพบผลึกเหลวชนิดใหม่ที่คุณสมบัติ คือทำงานในแนวระนาบโดยธรรมชาติ และต้องการทรานซิสเตอร์เพียงตัวเดียวก็สามารถให้ผลลัพธ์เหมือน IPS เลยเรียกว่าว่าชนิด VA (Vertical Align) จอชนิดนี้จะไม่ใช้ผลึกเหลวที่ทำงานเป็นเกลียวอีกต่อไป แต่จะมีผลึกเป็นแท่ง ซึ่งปกติถ้าไม่มีไฟป้อนเข้าไปหา ก็จะขวางจอเอาไว้ทำให้เป็นสีดำ และเมื่อได้รับกระแสไฟฟ้า ก็จะตั้งฉากกับจอให้แสงผ่านเป็นสีขาว ทำให้จอชนิดนี้มีความเร็วสูงมาก เพราะไม่ได้เคลือบเกลียว แต่ปรับทิศทางของผลึกเท่านั้น จอชนิดนี้จะมีมุมมองได้กว้างราว 160 องศา

ปัจจุบันบริษัท Fujitsu ได้ออกจอชนิดใหม่คือ MVA (Multi-Domain Vertical Alignment) ออกมาแก้ปัญหานี้เอง คือจากรูจะเห็นว่าด้วยความที่เป็นผลึกแท่ง และองศาของมันใช้กำหนดความสว่างของจุด ดังนั้นเมื่อมองจากมุมมองอื่น ความสว่างของภาพก็จะเปลี่ยนไปเลย เพราะถูกผสมในอีกรูปแบบหนึ่ง จอ Multidomain ก็จะพยายามกระจายมุมมองให้แต่ละ Pixel นั้นมีผลึกหลายมุมเอียงกันไป ทำให้ผลกระทบจากการกระมองมุมที่ต่างออกไปหักล้างกันเอง



รูปที่ 2.7 LCD (Liquid Crystal Display)

(ที่มา: <https://www.robotsiam.com/>)

การต่อ LCD แบบ I2C (Inter-Integrated Circuit)

เป็นวิธีที่ใช้ในการเชื่อมต่อหน้าจอลiquid crystal display (LCD) กับไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์ในรูปแบบดิจิทัล โดยใช้ระบบ I2C สัญญาณแสดงผลและควบคุมของ LCD ถูกส่งผ่านสายสัญญาณข้อมูลแบบสองสายเพียงแค่อสองสายเท่านั้น แบบ I2C มีข้อได้เปรียบอื่น ๆ เช่น ใช้สายน้อยกว่าและจำเป็นต้องใช้ขาการทำงานน้อยกว่าในการควบคุมหน้าจอลiquid crystal display โดยเฉพาะในสถานะที่ทรานซิสเตอร์ระบบไม่เพียงพอ.

ขั้นตอนการต่อหน้าจอ LCD แบบ I2C โดยละเอียด:

เลือกหน้าจอ LCD I2C: เริ่มต้นโดยการเลือกหน้าจอ LCD ที่รองรับการเชื่อมต่อแบบ I2C โดยรายชื่อหรือรหัสผลิตภัณฑ์.

สายสัญญาณ: จัดเตรียมสายสัญญาณสำหรับการเชื่อมต่อ ในกรณีนี้คุณจะต้องมีสายสายที่มีสายสัญญาณสองสาย คือ SDA (Serial Data) และ SCL (Serial Clock) เพื่อการสื่อสารระหว่างไมโครคอนโทรลเลอร์และ LCD.

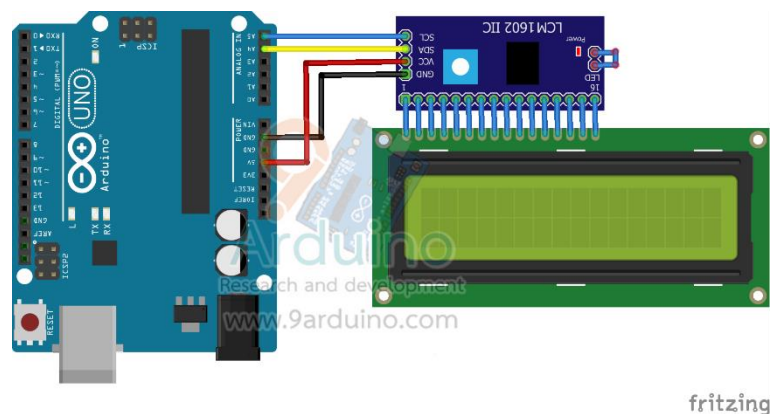
ต่อสายสัญญาณ: ต่อสายสัญญาณ SDA และ SCL จากไมโครคอนโทรลเลอร์ไปยังหน้าจอ LCD โดยใช้ขา GPIO หรือขา I2C บนไมโครคอนโทรลเลอร์ของคุณ.

ไฟเลี้ยง: ตรวจสอบความต้องการในการจ่ายไฟเลี้ยงและต่อสายไฟเลี้ยงจากแหล่งไฟที่เหมาะสม (เช่น 5V) ไปยังหน้าจอ LCD.

โค้ดควบคุม: เขียนโค้ดควบคุมที่จำเป็นเพื่อส่งข้อมูลและคำสั่งไปยังหน้าจอ LCD ผ่านทาง I2C. คุณสามารถใช้ไลบรารี I2C ที่มีให้แกไมโครคอนโทรลเลอร์ของคุณเพื่อควบคุม LCD.

อินิเชียลไต์ LCD: เมื่อทุกอย่างถูกตั้งค่าและเชื่อมต่ออย่างถูกต้อง คุณสามารถใช้หน้าจอ LCD แสดงผลข้อมูลและข้อความตามที่กำหนดในโค้ดควบคุมของคุณ.

การเชื่อมต่อหน้าจอ LCD แบบ I2C มีประสิทธิภาพสูงและเป็นวิธีที่คุณสามารถควบคุมหน้าจอ LCD ในโปรเจกต์ต่าง ๆ อย่างรวดเร็วและสะดวก โดยไม่ต้องใช้สายสัญญาณมากเกินไป.



รูปที่ 2.8 การต่อ LCD I2C

(ที่มา: <https://www.9arduino.com/>)

บทที่ 3

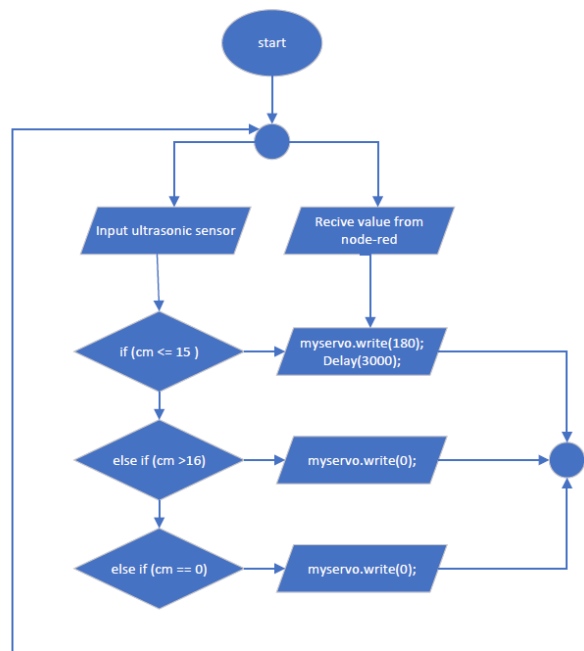
การออกแบบการทดลองและและการสร้าง

ในบทนี้จะกล่าวถึงการออกแบบระบบควบคุมของถังขยะอัจฉริยะ ซึ่งมีบล็อกไดอะแกรม (Block diagrams) แสดงหลักการทำงานของส่วนต่าง ๆ ในโครงงานโฟลวชาร์ต (Flowchart) การทำงานระบบควบคุมของถังขยะอัจฉริยะ การออกแบบวงจร และการออกแบบโครงสร้างการทำงานของระบบที่เกี่ยวข้องกับถังขยะอัจฉริยะ

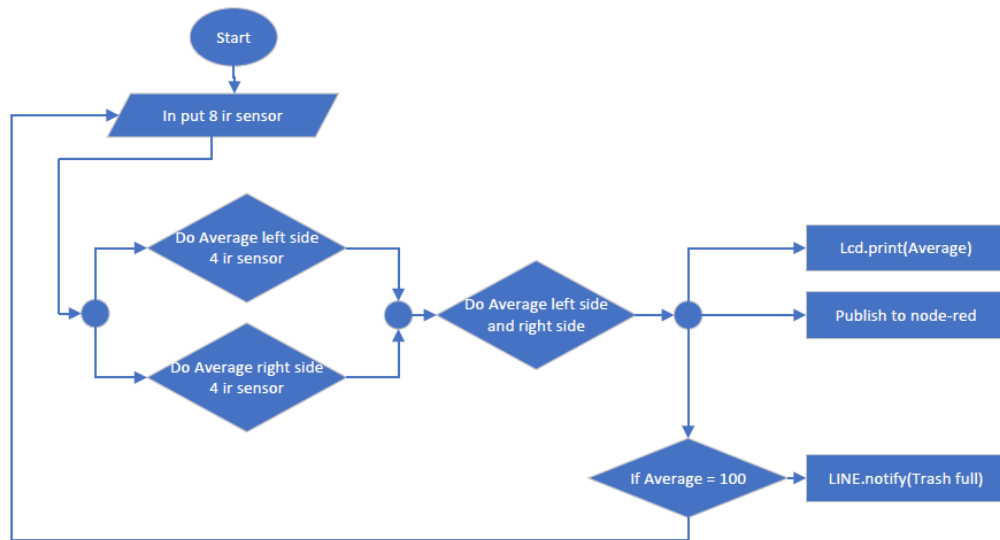
3.1 เครื่องมือ

- 3.1.1 อุปกรณ์
- 3.1.2 Arduino board
- 3.1.3 Servo Motor Arduino SG90
- 3.1.4 Ultrasonic sensor
- 3.1.5 ESP32
- 3.1.6 1602 LCD 16 x 2
- 3.1.7 Infrared (IR) Sensor
- 3.1.8 สายไฟจัมเปอร์

3.2 Block Diagram ของถังขยะอัจฉริยะ



รูปที่ 3.1 Block Diagram เปิดฝาอัตโนมัติถังขยะอัจฉริยะ

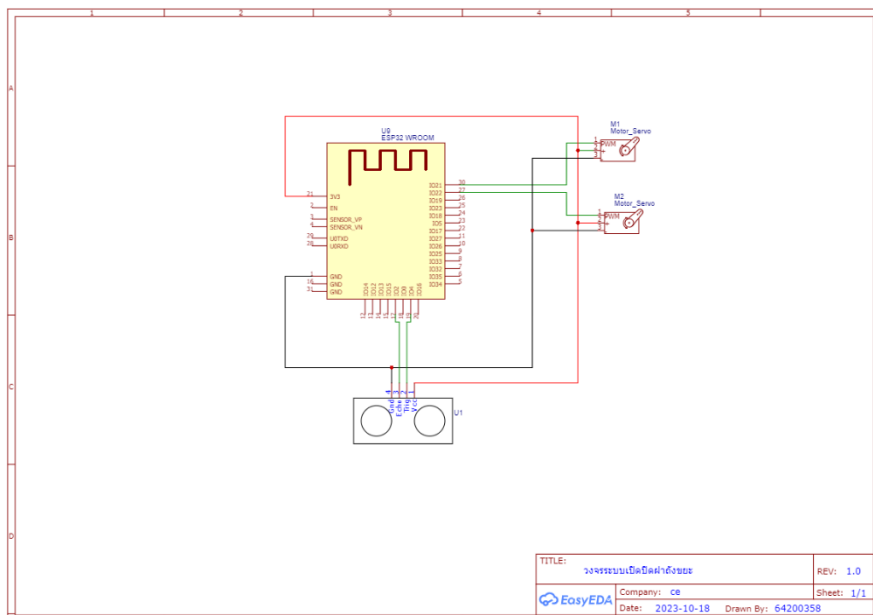


รูปที่ 3.2 Block Diagram วัดระดับขยะ

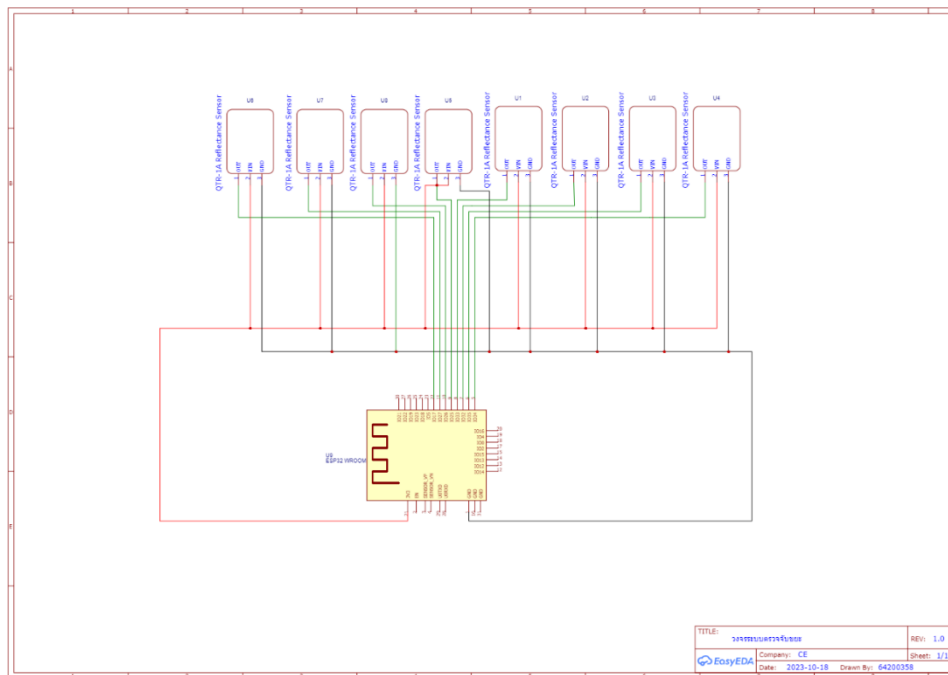
3.3 การออกแบบวงจร

ในการออกแบบวงจรจะขอกล่าวถึง การออกแบบวงจรรวมของระบบถังขยะอัจฉริยะ โดยแบ่งออกเป็นทั้งหมด 2 ส่วน ดังต่อไปนี้

1. วงจรของระบบเปิดปิดฝาถังขยะ (Ultrasonic sensor และ Servo motor)
2. วงจรของระบบตรวจจับขยะ (IR-sensor และ LCD I2C)



รูปที่ 3.3 วงจรของระบบเปิดปิดฝาถังขยะ



รูปที่ 3.4 วงจรของระบบตรวจจับขยะ

3.4 การออกแบบโครงสร้าง

การออกแบบโครงสร้างของระบบถังขยะอัจฉริยะนั้น มีความสำคัญมาก หากโครงสร้างของระบบถังขยะอัจฉริยะไม่มีความแข็งแรง หรือตัวถังมีน้ำหนักไม่เหมาะสม จะส่งผลให้การทำงานของระบบเกิดการผิดพลาดและไม่สามารถทำงานได้ตรงตามจุดประสงค์ที่วางไว้

3.4.1 การออกแบบโครงสร้างของระบบถังขยะอัจฉริยะ

โครงสร้างหลักของระบบถังขยะอัจฉริยะใช้วัสดุพลาสติก ที่มีความสามารถในการรองรับถังขยะทั่วไป โครงสร้างนี้จัดเตรียมให้เหมาะสมกับขนาดและน้ำหนักของถังขยะ ซึ่งถังขยะมีขนาด 20 ลิตร และมีขนาดที่กว้าง 31.5 เซนติเมตร x 23 เซนติเมตร x 42 เซนติเมตร ดังรูปภาพที่ 3.5 โดยโครงสร้างนี้ใช้เหล็กหรือวัสดุแข็งในการสร้างคานเพื่อยึดกับด้านหลังของถังขยะเพื่อใช้ในกระบวนการดึงถังขยะขึ้น ในกรณีนี้เราใช้ตะขอแขวนรอก (hanker hook) ดังรูปภาพที่ 3.6 และใช้เส้นเอ็นเพื่อให้มีความยืดหยุ่นเพียงพอในกระบวนการผูกติดกับตัว Servo-motor ดังรูปภาพที่ 3.7 นอกจากนี้ เรายังใช้กล่องอุปกรณ์หรือกล่องเพื่อใส่วงจรต่างๆ ของระบบถังขยะอัจฉริยะ ตามรูปภาพที่ 3.8 และทำให้การจัดการและการบริหารจัดการระบบมีความสะดวกมากขึ้นเช่นกัน ในทางปฏิบัติแล้วนี่คือรายละเอียดของโครงสร้างหลักของระบบถังขยะอัจฉริยะที่เราได้ออกแบบไว้ ซึ่งมีความสำคัญในการให้บริการอย่างมีประสิทธิภาพและความสะดวกสบายในการใช้งาน.



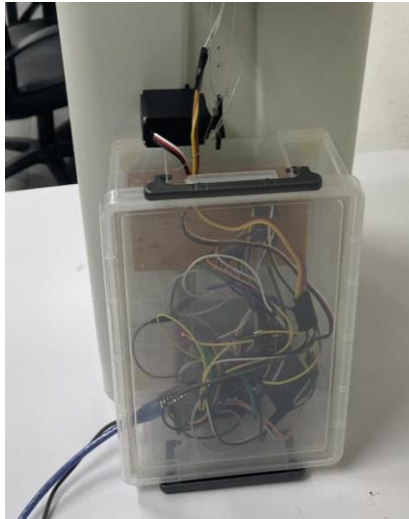
รูปที่ 3.5 ถังขยะขนาด 20 ลิตร



รูปที่ 3.6 ตะขอแขวนรอก (hanker hook)



รูปที่ 3.7 เส้นเอ็นผูกระหว่างตะขอและ Servo-motor

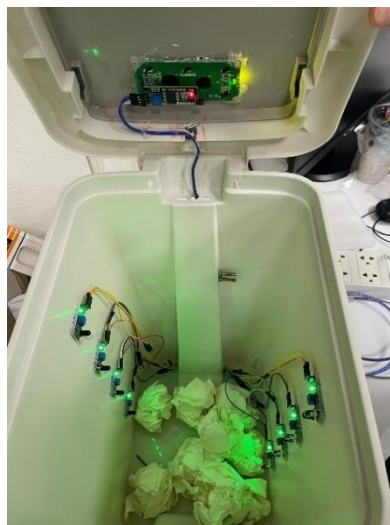


รูปที่ 3.8 กล่องเก็บสายและวงจรทั้งหมดของถังขยะอัจฉริยะ

3.4.2 การออกแบบโครงสร้างภายในของระบบถังขยะอัจฉริยะ

โดยโครงสร้างภายในของถังพยายามออกแบบให้ใช้พื้นที่ภายในน้อยที่สุด และจะติดตั้งเซนเซอร์ระบบตรวจจับระบบอินฟราเรด (IR-sensor) อยู่ทางด้านภายในถัง ทั้งสองด้าน ด้วยจำนวน 4 ตัวสำหรับแต่ละด้าน โดยระยะห่างนี้เท่ากันทุกด้าน เท่ากับ 7 เซนติเมตร ระยะเริ่มต้นจากระยะห่างจากฝาถัง 3 เซนติเมตรนอกจากนี้ เราจะติดตั้งสายเชื่อมต่อไปยังจอแสดงผล LCD ซึ่งจะถูกติดตั้งบนฝาถัง ตามภาพที่ 3.9

การออกแบบนี้ทำให้เราสามารถใช้พื้นที่ในถังอย่างมีประสิทธิภาพและใช้ระบบระดับเซนเซอร์ IR ในการตรวจจับสถานะของถังขยะ โดยจอ LCD จะถูกติดตั้งบนฝาถังเพื่อแสดงผลข้อมูลและสถานะของถังขยะอัจฉริยะเพื่อความสะดวกและการบริหารจัดการที่เรียบง่ายขึ้นในการใช้งาน.



รูปที่ 3.8 โครงสร้างภายในของระบบถังขยะอัจฉริยะ

บทที่ 4

วิธีการและการทดลอง

การทดลองประสิทธิภาพการใช้งานถึงขยะมี

4.1 การทดลองเปิด-ปิดฝาล้าง

การเปิด-ปิดโดยการกำหนดระยะที่เซ็นเซอร์อัลตราโซนิก (ultrasonic sensor) ตรวจจับได้ในระยะที่กำหนดคือ 1 เซนติเมตร – 15 เซนติเมตร โดยจะทำการให้เซอร์โวมอเตอร์ (servo motor) หมุน 180 องศาเพื่อเปิดฝาล้างขยะขึ้นค้างไว้เป็นเวลา 3 วินาที และปิดลงเองอัตโนมัติ หากมีการตรวจจับวัตถุในระยะเซ็นเซอร์อัลตราโซนิก (ultrasonic sensor) ค้างไว้ฝาล้างก็จะเปิดไว้เรื่อยๆ ตามที่เซ็นเซอร์อัลตราโซนิก (ultrasonic sensor) จับได้

4.1.1 วิธีการทดลองเปิด-ปิดฝาล้าง

- 1) เสียบแหล่งจ่ายไฟเลี้ยงให้บอร์ดอีเอสพี 32
- 2) ต่อสายเซ็นเซอร์อัลตราโซนิกกับบอร์ดอีเอสพี 32
- 3) ต่อสายเซอร์โวมอเตอร์กับบอร์ดอีเอสพี 32
- 4) ทดลองใช้มือวางมือในระยะเซ็นเซอร์อัลตราโซนิก 5 ครั้ง
- 5) ตรวจสอบว่าแต่ละครั้งใช้เวลาในการเปิด-ปิด ฝาล้างได้เท่าใด
- 6) บันทึกผลการทดลองในตาราง 4.1 และ 4.2

4.1.2 ผลการทดลองการเปิดฝาล้าง

ตารางที่ 4.1.2 ตารางผลการทดลองดีเลย์การเปิดฝาล้าง(วินาที)

เปิดด้วย	การเปิดฝา					เวลาเฉลี่ย
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5	
Ultrasonic	0	0	0	0	0	0
Node-red	0.5	1	0.2	0.2	0.2	0.42

4.1.3 ผลการทดลองการปิดฝาลัง

ตารางที่ 4.1.3 ตารางผลการทดลองดีเลย์การปิดฝาลัง(วินาที)

เปิด-ปิดด้วย	การปิดฝา					เวลาเฉลี่ย
	ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5	
Ultrasonic	3	3	3	3	3	3
Node-red	4	3.5	3.2	3	3.1	3.36

ตารางผลการทดลองดีเลย์การรับค่า IR sensor (วินาที)

ครั้งที่	ดีเลย์รับค่าแสดงบน terminal	ดีเลย์รับค่าแสดงบน debug node-red	ดีเลย์รับค่าแสดงบน dashboard	ดีเลย์รับค่าแสดงบน LINE notify
1	3	4	4	6
2	3	3	3	5
3	3	4	4	6
4	3	3	3	4
5	3	3	3	4



รูปที่ 4.1 รูปการทดลอง ultrasonic เปิดปิดฝาลัง

4.2 การทดลองทดสอบเซนเซอร์ตรวจจับวัตถุ

ในการทดลองนี้จะใช้เซนเซอร์ตรวจจับวัตถุ 8 ตัว ทำการวัดการตรวจจับวัตถุในถังขยะที่ทำทดลอง แบ่งเป็น 2 ฝั่งๆละ 4 ตัว จะทำการทดลองเซนเซอร์ตรวจจับวัตถุ โดยใช้วัตถุผ่านตัวเซนเซอร์แต่ละเซนเซอร์ โดยแสดงค่าของการตรวจจับวัตถุของแต่ละเซนเซอร์ไปยัง serial monitor เพื่อดูว่าเซนเซอร์ตรวจจับวัตถุแต่ละตัวสามารถตรวจจับวัตถุได้

4.2.1 วิธีการทดลองเซนเซอร์ตรวจจับวัตถุ

- 1) ต่อแหล่งจ่ายไฟไปยังบอร์ดอีเอสพี 32
- 2) ต่อแหล่งจ่ายไฟไปยังเซนเซอร์ตรวจจับวัตถุ
- 3) ต่อขาของเซนเซอร์ตรวจจับวัตถุตามที่กำหนดเอาไว้
- 4) วางวัตถุไว้ด้านหน้าของเซนเซอร์
- 5) ทดลองเซนเซอร์ตรวจจับวัตถุ 5 ครั้ง
- 6) บันทึกผลการทดลองในตารางที่ 4.1

4.2.2 ผลการทดลองทดสอบเซนเซอร์ตรวจจับวัตถุ

ตารางที่ 4.2.1 ผลการทดลองทดสอบเซนเซอร์ตรวจจับวัตถุ

เซนเซอร์ที่	การทดลองเซนเซอร์ตรวจจับวัตถุ				
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5
เซนเซอร์ที่ 1	✓	✓	✓	✓	✓
เซนเซอร์ที่ 2	✓	✓	✓	✓	✓
เซนเซอร์ที่ 3	✓	✓	✓	✓	✓
เซนเซอร์ที่ 4	✓	✓	✓	✓	✓
เซนเซอร์ที่ 5	✓	✓	✓	✓	✓
เซนเซอร์ที่ 6	✓	✓	✓	✓	✓
เซนเซอร์ที่ 7	✓	✓	✓	✓	✓
เซนเซอร์ที่ 8	✓	✓	✓	✓	✓

หมายเหตุ : ✓ หมายถึง ทำงานได้ปกติ

จากผลการทดลองตารางที่ 4.1 พบว่าการทดลองเซนเซอร์ตรวจจับวัตถุทั้ง 8 ตัว และทำการทดสอบ 5 ครั้ง จากการทดลองเซนเซอร์ตรวจจับวัตถุสามารถตรวจจับวัตถุได้ทุกตัว

4.3 การทดสอบระยะของเซนเซอร์ตรวจจับวัตถุ

ในการทดลองนี้จะใช้เซนเซอร์ตรวจจับวัตถุ 8 ตัว ทำการตรวจจับวัตถุในระยะที่ 20 และมากกว่า 20 เซนติเมตร โดยที่การทดลอง จะทำการทดสอบระยะเซนเซอร์ตรวจจับวัตถุ โดยวางวัตถุไว้ด้านหน้าของเซนเซอร์ตามระยะที่กำหนด และแสดงค่าของบน LCD 16x2 โมดูลจอแสดงผล LCD พร้อม I2C Interface เพื่อดูว่าเซนเซอร์ตรวจจับวัตถุแต่ละตัวสามารถตรวจจับวัตถุได้ตามระยะที่กำหนด

4.3.1 วิธีการทดลองเซนเซอร์ตรวจจับวัตถุ

- 1) ต่อแหล่งจ่ายไฟไปยังบอร์ดอีเอสพี 32
- 2) ต่อแหล่งจ่ายไฟไปยังเซนเซอร์ตรวจจับวัตถุ
- 3) ต่อแหล่งจ่ายไฟไปยัง บน LCD 16x2 โมดูลจอแสดงผล LCD พร้อม I2C Interface
- 4) ต่อขาของเซนเซอร์ตรวจจับวัตถุตามที่กำหนดเอาไว้
- 5) วางวัตถุไว้ด้านหน้าของเซนเซอร์ตามระยะที่กำหนด
- 6) ทดลองเซนเซอร์ตรวจจับวัตถุ
- 7) บันทึกผลการทดลองในตารางที่ 4.2

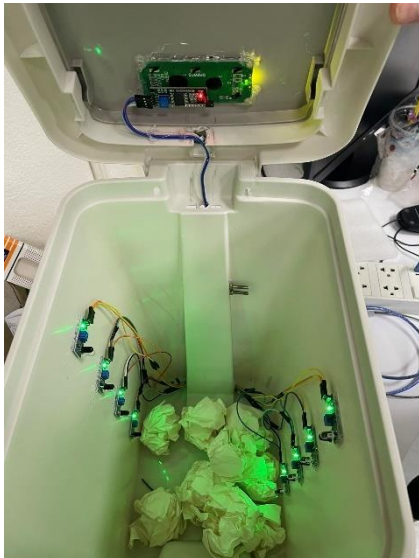
4.3.2 ผลการทดสอบระยะเซนเซอร์ตรวจจับวัตถุ

ตารางที่ 4.3.1 ผลการทดสอบระยะเซนเซอร์ตรวจจับวัตถุ

เซนเซอร์ที่	การทดลองเซนเซอร์ตรวจจับวัตถุ									
	ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3		ครั้งที่ 4		ครั้งที่ 5	
	20 cm	>20 cm	20 cm	>20 cm	20 cm	>20 cm	20 cm	>20 cm	20 cm	>20 cm
เซนเซอร์ที่ 1	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 2	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 3	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 4	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 5	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 6	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 7	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
เซนเซอร์ที่ 8	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗

หมายเหตุ : ✓ หมายถึง ทำงานได้ปกติ

จากผลการทดลองตารางที่ 4.2.1 และ 4.3.1 พบว่าการทดลองเซนเซอร์ตรวจจับวัตถุทั้ง 8 ตัว และทำการทดสอบ 5 ครั้ง เซนเซอร์สามารถตรวจจับวัตถุได้ในระยะที่กำหนดได้ทุกเซนเซอร์



รูปที่ 4.2 รูปการทดลองการทำงานของ IR-Sensor และหน้าจอแสดงผล LCD

บทที่ 5

บทสรุปและข้อเสนอแนะ

ในบทนี้กล่าวถึง บทสรุปในการทดลองจากบทที่ 4 ในการทดสอบเซนเซอร์ตรวจจับวัตถุ การทดสอบระยะของเซนเซอร์ตรวจจับวัตถุ การทดลองดีเลย์การเปิดฝาถัง(วินาที) การทดลองดีเลย์การปิดฝาถัง(วินาที) การทดลองดีเลย์การรับค่าจากเซ็นเซอร์อินฟราเรด ปัญหาและอุปสรรคในการทำงาน ข้อเสนอแนะ และวิธีการแก้ไขปัญหา

5.1 สรุปผลการทดลอง

5.1.1 การทดลองดีเลย์การเปิดฝาถัง(วินาที)

จากการทดลองการดีเลย์การเปิดฝาถัง โดยเปิดด้วย Ultrasonic Sensor Module (HC-SR04) และ Node-red จำนวน 5 ครั้ง ผลปรากฏว่า Ultrasonic Sensor Module (HC-SR04) มีความหน่วงของการเปิดฝาถัง เฉลี่ยอยู่ที่ 0 วินาที และ Node-red มีความหน่วงของการเปิดฝาถัง เฉลี่ยอยู่ที่ 0.42 วินาที ดังนั้น Ultrasonic Sensor Module (HC-SR04) มีความหน่วงในการเปิดน้อยกว่าการเปิดจาก Node-red

5.1.2 การทดลองดีเลย์การปิดฝาถัง(วินาที)

จากการทดลองการดีเลย์การปิดฝาถัง โดยปิดด้วย Ultrasonic Sensor Module (HC-SR04) และ Node-red จำนวน 5 ครั้ง ผลปรากฏว่า Ultrasonic Sensor Module (HC-SR04) มีความหน่วงของการปิดฝาถัง เฉลี่ยอยู่ที่ 3 วินาที และ Node-red มีความหน่วงของการปิดฝาถัง เฉลี่ยอยู่ที่ 3.36 วินาที ด้วย Ultrasonic Sensor Module (HC-SR04) มีความหน่วงในการปิดน้อยกว่าการปิดจาก Node-red

5.1.3 การทดลองดีเลย์การรับค่าจากเซ็นเซอร์อินฟราเรด

จากการทดลองการดีเลย์การรับค่าจากเซนเซอร์ตรวจจับวัตถุ โดยแสดงค่าผ่าน Terminal , Dashboard Node-red และ Line Notify จำนวน 5 ครั้ง ผลปรากฏว่า **Terminal** มีความหน่วงในการรับค่าจากเซนเซอร์ตรวจจับวัตถุอินฟราเรด เวลาเฉลี่ย 1 วินาที **Dashboard Node-red** มีความหน่วงในการรับค่าจากเซนเซอร์ตรวจจับวัตถุอินฟราเรด เวลาเฉลี่ย 2.42 วินาที **Line Notify** มีความหน่วงในการรับค่าจากเซนเซอร์ตรวจจับวัตถุอินฟราเรด เวลาเฉลี่ย 2.4 วินาที ดังนั้นสรุปได้ว่า Terminal รับค่าจากเซนเซอร์ได้เร็วกว่า Dashboard Node-red และ Line Notify

5.1.4 การทดสอบเซนเซอร์ตรวจจับวัตถุ

จากการทดลองทดสอบเซนเซอร์ตรวจจับวัตถุ จำนวน 8 เซนเซอร์ ทั้งหมด 5 ครั้ง ผลปรากฏว่าเมื่อวางวัตถุเอาไว้ด้านหน้าเซนเซอร์ที่ 1 – 8 เซนเซอร์สามารถตรวจจับวัตถุและแสดงค่าใน Serial Monitor

5.1.5 การทดสอบระยะของเซนเซอร์ตรวจจับวัตถุ

จากการทดลองทดสอบระยะของเซนเซอร์ตรวจจับวัตถุ จำนวน 8 เซนเซอร์ ทั้งหมด 5 ครั้ง โดยกำหนดระยะที่ 20 เซนติเมตร และ มากกว่า 20 เซนติเมตร ผลปรากฏว่าวางวัตถุในระยะ 20 เซนติเมตร เซนเซอร์ตรวจจับวัตถุสามารถตรวจจับวัตถุได้ และแสดงค่าของบน LCD 16x2 โมดูล จอแสดงผล LCD พร้อม I2C Interface ในระยะมากกว่า 20 เซนติเมตร เซนเซอร์ไม่สามารถตรวจจับวัตถุได้

5.2 ปัญหาและอุปสรรค

- 1) โมดูลที่ใช้มีปัญหาตอนต่ออุปกรณ์และทดลองโมดูล เพราะต้องมีการถอดเข้า-ออก บ่อยและสายไฟจัมเปอร์มีการหลุดออกจากกันง่าย ทำให้มีปัญหาในภายหลัง
- 2) บอร์ดที่ใช้ในการทำชิ้นงาน ไม่สามารถอัปโหลดโค้ดลงบอร์ดได้ในช่วงแรก และช่วงหลังต้องซื้อบอร์ดตัวใหม่ เพราะบอร์ดตัวเก่ามีปัญหา
- 3) ปัญหาด้าน internet ในช่วงที่สัญญาณมีปัญหา ไม่สามารถเชื่อมกับ Dashboard Node-red ได้
- 4) ฝาถังขยะอาจมีน้ำหนักที่มากเกินไป จนทำให้มอเตอร์ขัดข้อง จึงต้องเปลี่ยนและทดลองมอเตอร์หลายตัว
- 5) เซนเซอร์ตรวจจับวัตถุภายในถังขยะ ไม่สามารถป้องกันวัตถุที่มีความแข็งมากหรือของเหลวได้

5.3 ข้อเสนอแนะและวิธีแก้ปัญหา

- 1) เพิ่มงบประมาณในการทำชิ้นงาน
- 2) เพิ่มอุปกรณ์ป้องกันตัวเซนเซอร์
- 3) เปลี่ยนหรือเพิ่ม Servo Motor ให้มีแรงขับเพิ่มมากขึ้น
- 4) เปลี่ยนเซนเซอร์ตรวจจับวัตถุภายในถังให้มีคุณภาพมากกว่านี้

เอกสารอ้างอิง

- [1] “ESP 32” [ออนไลน์] แหล่งที่มา: <https://techtalk2apply.com/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 19 กรกฎาคม 2566
- [2] “Servo Motor คืออะไร ???” [ออนไลน์] แหล่งที่มา: <https://www.sangchaimeter.com/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 20 กรกฎาคม 2566
- [3] “Ultrasonic Sensor คืออะไร” [ออนไลน์] แหล่งที่มา: <https://mall.factomart.com/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 20 กรกฎาคม 2566
- [4] “การทำงานของ Ultrasonic Sensor” [ออนไลน์] แหล่งที่มา: <https://mall.factomart.com/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 20 กรกฎาคม 2566
- [5] “ESP32 Pinout” [ออนไลน์] แหล่งที่มา: <https://www.electronicshub.org/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 21 กรกฎาคม 2566
- [6] “เซนเซอร์อินฟราเรด” [ออนไลน์] แหล่งที่มา: <https://medium.com/@projectid030/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 24 กรกฎาคม 2566
- [7] “Infrared (IR) Sensor” [ออนไลน์] แหล่งที่มา: <https://www.artronshop.co.th/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 25 กรกฎาคม 2566
- [9] “ESP32 ทำงานอย่างไร” [ออนไลน์] แหล่งที่มา: <https://v89infinity.com/>
เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 28 กรกฎาคม 2566
- [10] “สอนใช้งาน ESP32 เซ็นเซอร์วัดระยะทาง” [ออนไลน์] แหล่งที่มา:
<https://www.cybertice.com/> เข้าถึงครั้งสุดท้ายเมื่อ วันที่ 5 สิงหาคม 2566
- [11] “สอนใช้งาน ESP32 เซ็นเซอร์ตรวจจับ” [ออนไลน์] แหล่งที่มา:
<https://www.cybertice.com/>
เข้าถึงครั้งสุดท้ายเมื่อ 21 สิงหาคม 2566
- [12] “IR Module with ESP32” [ออนไลน์] แหล่งที่มา: <https://linuxhint.com/>
เข้าถึงครั้งสุดท้ายเมื่อ 21 สิงหาคม 2566
- [13] “จอ LCD คืออะไร ?” [ออนไลน์] แหล่งที่มา: <https://www.gotoknow.org/>
เข้าถึงครั้งสุดท้ายเมื่อ 21 สิงหาคม 2566

ภาคผนวก ก

โปรแกรมของระบบควบคุมการทำงานของถังขยะอัจฉริยะ

โค้ดการทำงานของถังขยะ

การทำงานของ Servo-motor

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <TridentTD_LineNotify.h>
#include <WiFi.h>
#include <PubSubClient.h>

#define SSID      "Bin1"
#define PASSWORD  "12345678"
#define LINE_TOKEN "HrmkzTI4mgOEpVH3eNLK41cEn1v7yZ61f5Fi4oGcq7"
#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT  1883
const char *topic = "status/test";
unsigned long lastMsg = 0;

// Define the I2C address for the LCD
LiquidCrystal_I2C lcd(0x27, 16, 2);

WiFiClient espClient;
PubSubClient client(espClient);

// Define IR sensor pins for each side
const int leftIrSensorPins[] = {2, 0, 4, 16}; // Pins for the left side sensors
const int rightIrSensorPins[] = {17, 5, 18, 19}; // Pins for the right side sensors

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

  Serial.println();
  Serial.println(LINE.getVersion());

  WiFi.begin(SSID, PASSWORD);
```

```

LINE.setToken(LINE_TOKEN);

Serial.printf("WiFi connecting to %s\n", SSID);

while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(400);
}

client.setServer(MQTT_SERVER, MQTT_PORT);
client.setCallback(callback);

Serial.printf("\nWiFi connected\nIP : ");
Serial.println(WiFi.localIP());
// Initialize LCD
lcd.begin();
lcd.setCursor(0, 0);
lcd.backlight();

// Initialize IR sensors as inputs
for (int i = 0; i < 4; i++) {
    pinMode(leftIrSensorPins[i], INPUT);
    pinMode(rightIrSensorPins[i], INPUT);
    digitalWrite(leftIrSensorPins[i], LOW);
    digitalWrite(rightIrSensorPins[i], LOW);
}
}

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
}

void reconnect() {
    while (!client.connected()) {

```

```

Serial.print("Attempting MQTT connection...");
String clientId = "ESP32Client-";
clientId += String(random(0xffff), HEX);
if (client.connect(clientId.c_str())) {
    Serial.println("Connected");
    client.publish("status/test", "Welcome");
    client.subscribe("status/test");
} else {
    Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 1 seconds");
    delay(1000);
}
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
    // Read values from IR sensors on the left side
    int leftSensorValues[4];
    int leftSum = 0;
    for (int i = 0; i < 4; i++) {
        leftSensorValues[i] = digitalRead(leftIrSensorPins[i]);
        leftSum += leftSensorValues[i];
    }
    int leftAverageValue = (leftSum * 100) / 4; // Calculate average as a percentage

    // Read values from IR sensors on the right side
    int rightSensorValues[4];
    int rightSum = 0;
    for (int i = 0; i < 4; i++) {
        rightSensorValues[i] = digitalRead(rightIrSensorPins[i]);
        rightSum += rightSensorValues[i];
    }
}

```

```

}
int rightAverageValue = (rightSum * 100) / 4; // Calculate average as a percentage

// Calculate overall average
int overallAverage = (leftAverageValue + rightAverageValue) / 2;

// Invert the overallAverage value to represent it as 0% when full and 100% when
not full
overallAverage = 100 - overallAverage;

unsigned long now = millis();
if (now - lastMsg > 2000) { //perintah publish data
    lastMsg = now;

    String overallaverage = String(overallAverage);
    client.publish("status/test", overallaverage.c_str()); // publish temp topic
/ThinkIOT/temp

}
// Display overall average on LCD
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Status: ");
lcd.print(overallAverage);
lcd.print("%");

// Check if overall average is 0%, if yes, send notification
if (overallAverage == 100) {
    // Send notification to LINE Notify
    //TridentTD_LineNotify::send(LINE_TOKEN, "Trash Full");
    LINE.notify("Trash Full");
    delay(1000);
}
delay(500); // Update the display every 1 second (adjust as needed)
}

```

การทำงานของ IR-Sensor

```
#include <ESP32Servo.h>
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "Bin1";
const char* password = "12345678";
const char* mqtt_server = "broker.hivemq.com";
int inPin = 14;
const int pingPin = 27;

Servo myservo;
int servoPin = 12; // กำหนดขาที่เชื่อมกับ Servo
int servoAngle = 90; // ตำแหน่งเริ่มต้นของ Servo

WiFiClient espClient;
PubSubClient client(espClient);

void setup() {
  pinMode(pingPin, OUTPUT);
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  myservo.attach(servoPin);
  myservo.write(servoAngle);
  delay(1000);
}

void callback(char* topic, byte* payload, unsigned int length) {
```

```

String string;
Serial.print("Message arrived [");
Serial.print(topic);
Serial.print("] ");
Serial.print("Received message on mqttTopic2: ");

for (int i = 0; i < length; i++) {
    string += (char)payload[i];
}

if (topic= "servo/control") {
    Serial.print(" ");
    myservo.write(0);
    int resultado = string.toInt();
    int angle = map(resultado, 1, 100, 0, 180);

    Serial.println(angle);
    myservo.write(angle);
    delay(2000);
}

}

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("esp32Client")) {
            Serial.println("connected");
            client.subscribe("servo/control");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}
}

```



```

void loop() {
    long duration, cm;
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    pinMode(inPin, INPUT);
    duration = pulseIn(inPin, HIGH);

    cm = microsecondsToCentimeters(duration);

    Serial.print("cm");
    Serial.println(cm);
    delay(1000);

    if (cm <= 15 ) {
        myservo.write(180);
        delay(3000);
    }
    else if (cm >16){
        myservo.write(0);
    }
    else if (cm <= 0){
        myservo.write(0);
    }
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
}

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

```

ภาคผนวก ข

โปรแกรมของระบบควบคุมการทำงานของถังขยะอัจฉริยะ



คู่มือการใช้งาน
Automatic Dustbin



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วิทยาเขตอุดมศักดิ์ จังหวัดชุมพร
ปีการศึกษา 2566

ระบบที่ใช้

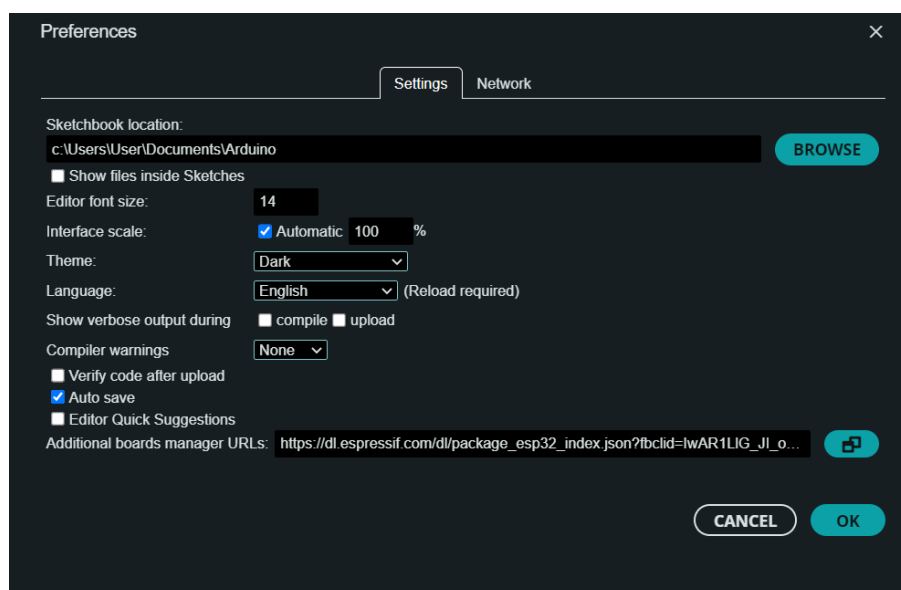
1) การเตรียมโปรแกรมอาดูโน้ (Arduino IDE) สำหรับประมวลผลบนอีเอสพี32

1.1) ดาวน์โหลดอาดูโน้ไอดีอีจากเว็บไซต์ <https://www.arduino.cc/en/software> จากนั้นติดตั้งโปรแกรมลงบนคอมพิวเตอร์ดังรูปที่



รูปที่ ข.1 หน้าดาวน์โหลด Arduino IDE 2.2.1

1.2) เปิดโปรแกรมขึ้นมา คลิกไปที่เมนู File แล้วเลือกที่ Preferences เพื่อติดตั้งบอร์ดอีเอสพี 32 ดังรูป

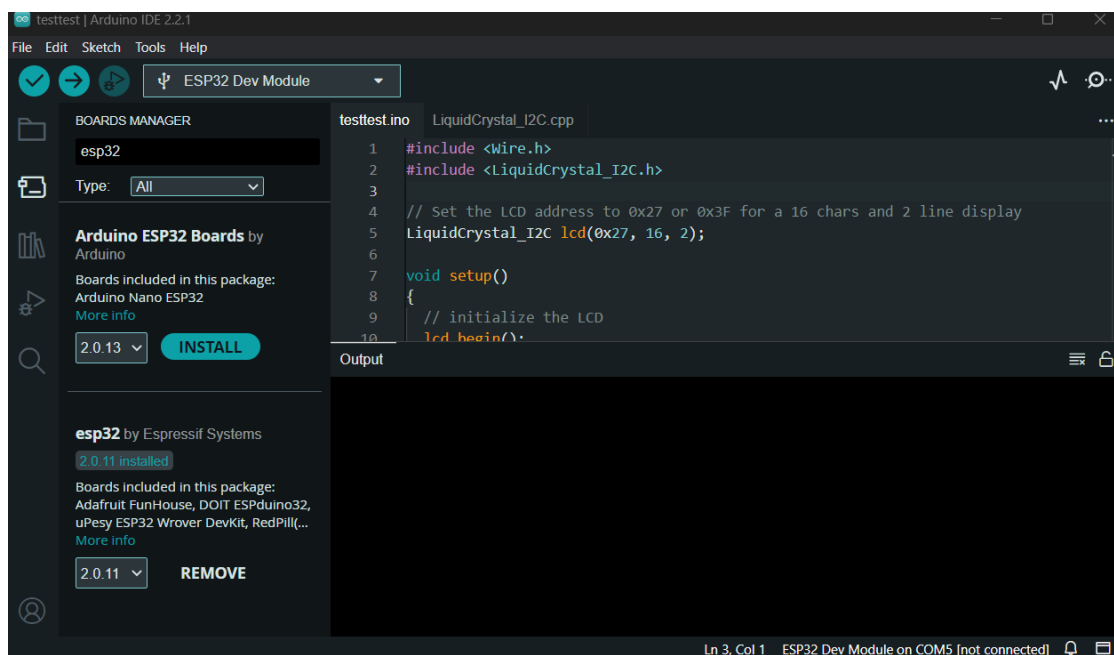


รูปที่ ข.2 ติดตั้งบอร์ด ESP32

1.3) ใส่ลิงค์เพื่อที่จะติดตั้งไดรเวอร์

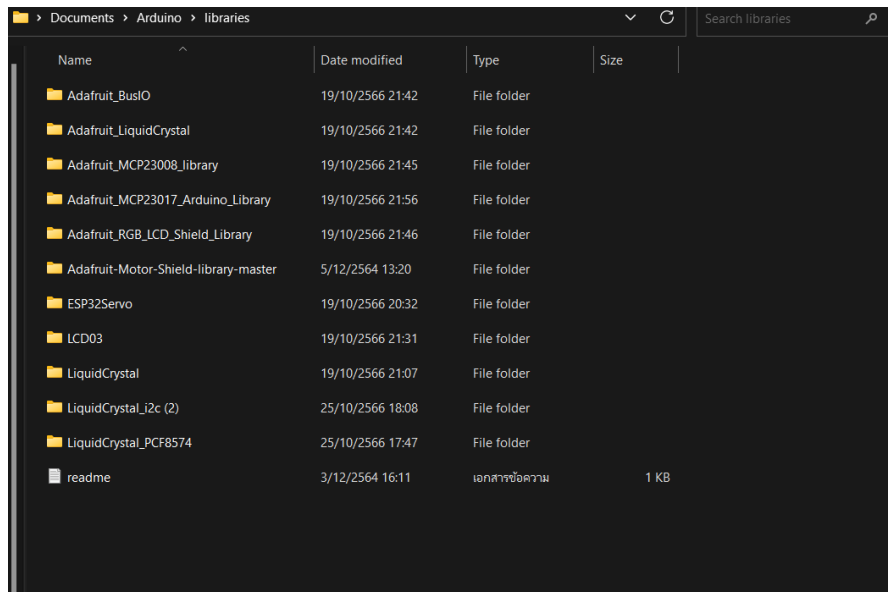
https://dl.espressif.com/dl/package_esp32_index.json?fbclid=IwAR1LLG_JI_oK6b72d6A8fH_LOpGPoojpRsKJx1LoVONfpyuOhnOaFjJnxcI

1.4) เลือกเมนูชื่อ Tool แล้วไปที่ Board Manager ค้นหา ESP32 Dev Module ในช่อง
การค้นหา ดังรูป



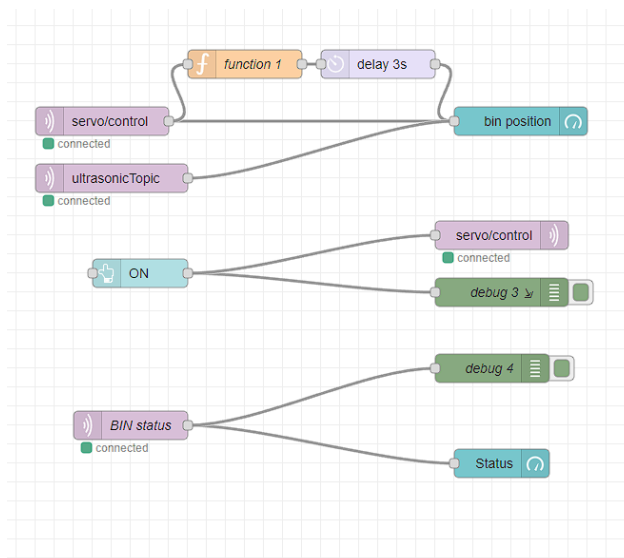
รูปที่ ข.3 เลือกใช้ ESP32 Dev Module

1.5) ค้นหา libraries ESP32 จากนั้นกดติดตั้งใน Arduino > libraries ดังรูป



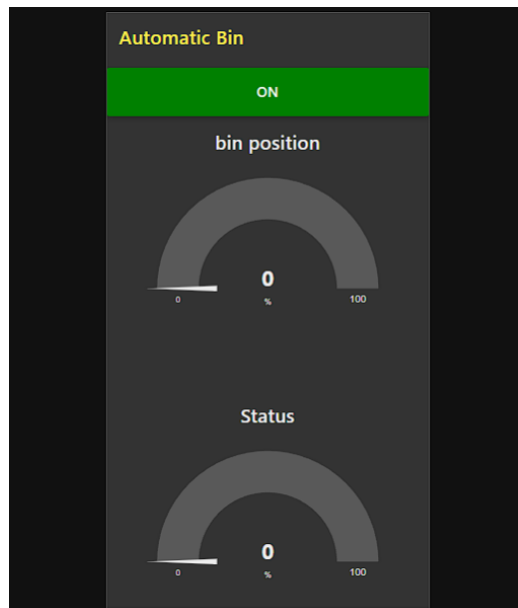
รูปที่ ข.4 ติดตั้งไลบรารีอีเอสพี 32

1.6) หน้าจอโหมด ดังรูป



รูปที่ ข.5 หน้าจอโหมด

1.7) ตัวอย่างหน้าแดสบอร์ด ดังรูป



รูปที่ ข.6 หน้าแดสบอร์ด

1.7) ตัวอย่างหน้าการทำงานหน้า Line notify ดังรูป



รูปที่ ข.7 หน้าการทำงานหน้า Line notify

การใช้งานทั่วไป

- 1) เสียบไฟเข้าตัวช่องถังขยะ เพื่อให้อุปกรณ์เริ่มการทำงาน ดังรูป



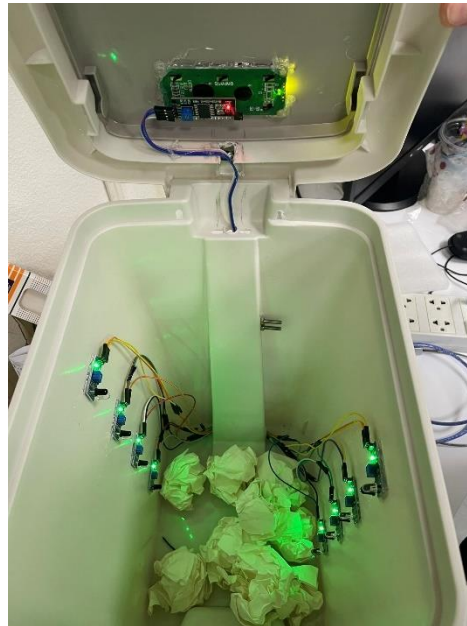
รูปที่ ข.8 เสียบไฟเสียบเข้าตัวช่องถังขยะ

- 2) เปิดฝาถังขยะโดยใช้ ultrasonic sensor บังเป็นเวลา 1 วินาที เพื่อเปิดฝาถัง ดังรูป



รูปที่ ข.9 เปิดฝาถังขยะ

3) ใส่ขยะลงไปเพื่อทดสอบการตรวจวัดระดับปริมาณขยะภายในถังขยะ โดยใช้ ir sensor ดังรูป



รูปที่ ข.10 การตรวจวัดระดับปริมาณขยะภายในถังขยะ

4) ตรวจสอบการแสดงค่าของปริมาณขยะเป็นเปอร์เซ็นต์ผ่านหน้าจอ LCD ดังรูป



รูปที่ ข.11 การแสดงผลผ่านหน้าจอ LCD ดังรูป

ภาคผนวก ค

คู่มือการใช้อุปกรณ์(Datasheets)

HC-SR04 Ultrasonic Sensor

Elijah J.
Morgan
Nov. 16
2014

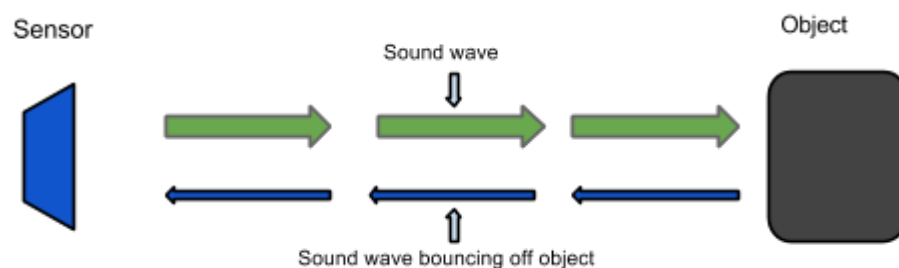
The purpose of this file is to explain how the HC-SR04 works. It will give a brief explanation of how ultrasonic sensors work in general. It will also explain how to wire the sensor up to a microcontroller and how to take/interpret readings. It will also discuss some sources of errors and bad readings.

1. How Ultrasonic Sensors Work
2. HC-SR04 Specifications
3. Timing chart, Pin explanations and Taking Distance Measurements
4. Wiring HC-SR04 with a microcontroller
5. Errors and Bad Readings



1. How Ultrasonic Sensors Work

Ultrasonic sensors use sound to determine the distance between the sensor and the closest object in its path. How do ultrasonic sensors do this? Ultrasonic sensors are essentially sound sensors, but they operate at a frequency above human hearing.

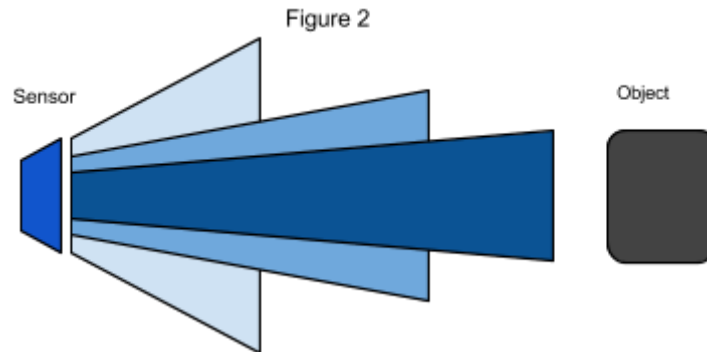


The sensor sends out a sound wave at a specific frequency. It then listens for that specific sound wave to bounce off of an object and come back (Figure 1). The sensor keeps track of the time between sending the sound wave and the sound wave returning. If you know how fast something is going and how long it is traveling you can find the distance traveled with equation 1.

Equation 1. $d = v \times t$

The speed of sound can be calculated based on the a variety of atmospheric conditions, including temperature, humidity and pressure. Actually calculating the distance will be shown later on in this document.

It should be noted that ultrasonic sensors have a cone of detection, the angle of this cone varies with distance, Figure 2 show this relation. The ability of a sensor to detect an object also depends on the objects orientation to the sensor. If an object doesn't present a flat surface to the sensor then it is possible the sound wave will bounce off the object in a way that it does not return to the sensor.



2. HC-SR04 Specifications

The sensor chosen for the Firefighting Drone Project was the HC-SR04. This section contains the specifications and why they are important to the sensor module. The sensor modules requirements are as follows.

- Cost
- Weight
- Community of hobbyists and support
- Accuracy of object detection
- Probability of working in a smoky environment
- Ease of use

The HC-SR04 Specifications are listed below. These specifications are from the Cytron Technologies HC-SR04 User's Manual (source 1).

- Power Supply: +5V DC
- Quiescent Current: <2mA
- Working current: 15mA
- Effectual Angle: <15°
- Ranging Distance: 2-400 cm
- Resolution: 0.3 cm
- Measuring Angle: 30°
- Trigger Input Pulse width: 10uS
- Dimension: 45mm x 20mm x 15mm
- Weight: approx. 10 g

The HC-SR04's best selling point is its price; it can be purchased at around \$2 per unit.

3. Timing Chart and Pin Explanations

The HC-SR04 has four pins, VCC, GND, TRIG and ECHO; these pins all have different functions. The VCC and GND pins are the simplest -- they power the HC-SR04. These pins need to be attached to a +5 volt source and ground respectively. There is a single control pin: the TRIG pin. The TRIG pin is responsible for sending the ultrasonic burst. This pin should be set to HIGH for 10 μ s, at which point the HC-SR04 will send out an eight cycle sonic burst at 40 kHz. After a sonic burst has been sent the ECHO pin will go HIGH. The ECHO pin is the data pin -- it is used in taking distance measurements. After an ultrasonic burst is sent the pin will go HIGH, it will stay high

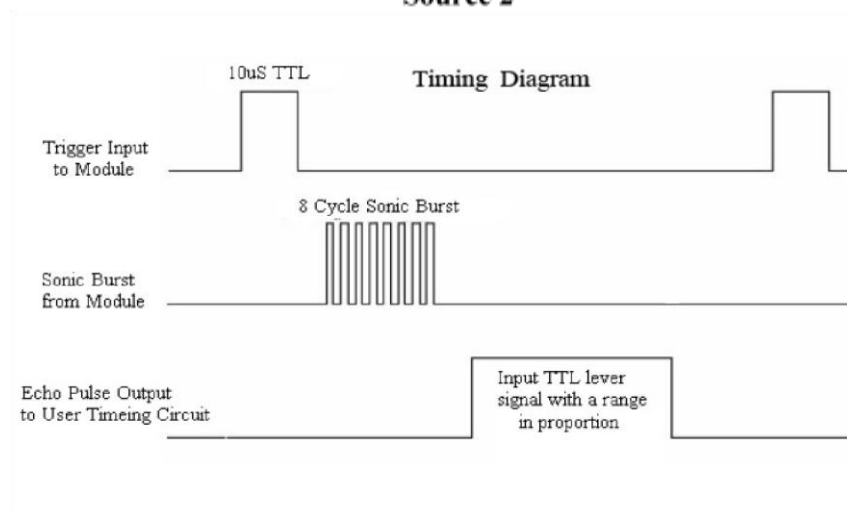
until an ultrasonic burst is detected back, at which point it will go LOW.

Taking Distance Measurements

The HC-SR04 can be triggered to send out an ultrasonic burst by setting the TRIG pin to HIGH. Once the burst is sent the ECHO pin will automatically go HIGH. This pin will remain HIGH until the the burst hits the sensor again. You can calculate the distance to the object by keeping track of how long the ECHO pin stays HIGH. The time ECHO stays HIGH is the time the burst spent traveling. Using this measurement in equation 1 along with the speed of sound will yield the distance travelled. A summary of this is listed below, along with a visual representation in Figure 2.

1. Set TRIG to HIGH
2. Set a timer when ECHO goes to HIGH
3. Keep the timer running until ECHO goes to LOW
4. Save that time
5. Use equation 1 to determine the distance travelled

Figure 3
Source 2



Source 2

To interpret the time reading into a distance you need to change equation 1. The clock on the device you are using will probably count in microseconds or smaller. To use equation 1 the speed of sound needs to be determined, which is 343 meters per second at standard temperature and pressure. To convert this into more useful form use equation 2

to change from meters per second to microseconds per centimeter. Then equation 3 can be used to easily compute the distance in centimeters.

Equation 2. $Distance = 1750 \frac{m}{s} \times 10^{-6} \frac{s}{\mu s} \times 117.06 \frac{\mu s}{m} \times 58.7 \frac{m}{\mu s}$

Equation 3. $Distance = \frac{t}{58.7} = \frac{\mu s}{cm} = cm$

4. Wiring the HC-SR04 to a Microcontroller

This section only covers the hardware side. For information on how to integrate the software side, look at one of the links below or look into the specific microcontroller you are using.

The HC-SR04 has 4 pins: VCC, GND, TRIG and ECHO.

1. VCC is a 5v power supply. This should come from the microcontroller
2. GND is a ground pin. Attach to ground on the microcontroller.
3. TRIG should be attached to a GPIO pin that can be set to HIGH
4. ECHO is a little more difficult. The HC-SR04 outputs 5v, which could destroy many microcontroller GPIO pins (the maximum allowed voltage varies). In order to step down the voltage use a single resistor or a voltage divider circuit. Once again this depends on the specific microcontroller you are using, you will need to find out its GPIO maximum voltage and make sure you are below that.

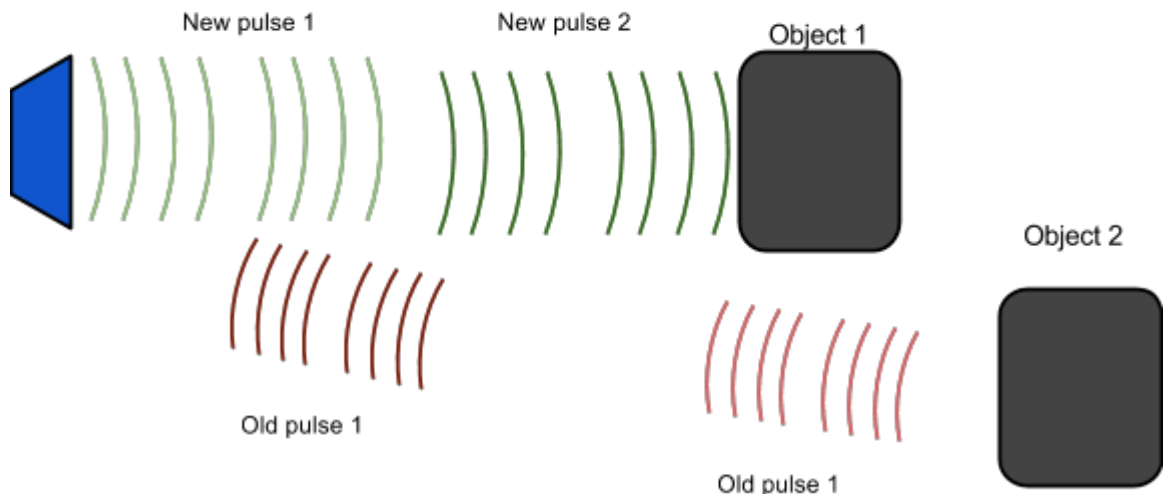


Figure 4

5. Errors and Bad Readings

Ultrasonic sensors are great sensors -- they work well for many applications where other types of sensors fall short. Unfortunately, they do have weaknesses. These

weaknesses can be mitigated and worked around, but first they must be understood. The first weakness is that they use sound. There is a limit to how fast ultrasonic sensors can get distance measurements. The longer the distance, the slower they are at reporting the distance. The second weakness comes from the way sound bounces off of objects. In enclosed spaces it is possible, if not probable that there will be unintended echos. The echos can very easily cause false short readings. In Figure 2 a pulse was sent out. It bounced off of object 1 and returned to the sensor. The distance was recorded and then a new pulse was sent. There was another object farther away, so that when the new pulse reaches object 1, the first signal will reach the sensor. This will cause the sensor to think that there is an object closer than is actually true. The old pulse is smaller than the new pulse because it has grown weaker. The longer the pulse exists the weaker it grows until it is negligible. If multiple sensors are being used, the number of echos will increase along with the number of errors. There are two main ways to reduce the number of errors. The first is to provide shielding around the sensor. This prevents echos coming in from angle outside what the sensor should actually pick up. The second is to reduce the frequency at which pulses are sent out. This gives more time for the echos to dissipate.



Works Cited

Source 1.

"HC-SR04 User's_Manual." *docs.google*. Cytron Technologies, May 2013 Web. 5 Dec. 2009.

<https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit>

Source 2.

"Attiny2313 Ultrasonic distance (HR-SR04) example." *CircuitDB*. n.a. 7 Sept. 2014

Web. 5 Dec. 2014. <<http://www.circuitdb.com/?p=1162>>

Links

These are not formatted; you will need to copy and paste them into your web browser.

Want to learn about Ultrasonic Sensors in general?

<http://www.sensorsmag.com/sensors/acoustic-ultrasound/choosing-ultrasonic-sensor-proximity-or-distance-measurement-825>

All about the HC-SR04

- <http://www.circuitdb.com/?p=1162>
- <http://www.micropik.com/PDF/HCSR04.pdf>
- <http://randomnerdtutorials.com/complete-guide-for-ultrasonic-sensor-hc-sr04/>
- <http://www.ezdenki.com/ultrasonic.php>
(^fantastic tutorial, explains a lot of stuff)
- <http://www.elecrow.com/hcsr04-ultrasonic-ranging-sensor-p-316.html> (^this one has some cool charts)



ESP32 Series

Datasheet

2.4 GHz Wi-Fi + Bluetooth® + Bluetooth LE SoC

Including:

ESP32-D0WD-V3

ESP32-D0WDR2-V3

ESP32-U4WDH

ESP32-S0WD – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WD – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6 – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6-V3 – [Not Recommended for New Designs \(NRND\)](#)



Version 4.3
Espressif Systems
Copyright © 2023



Product Overview

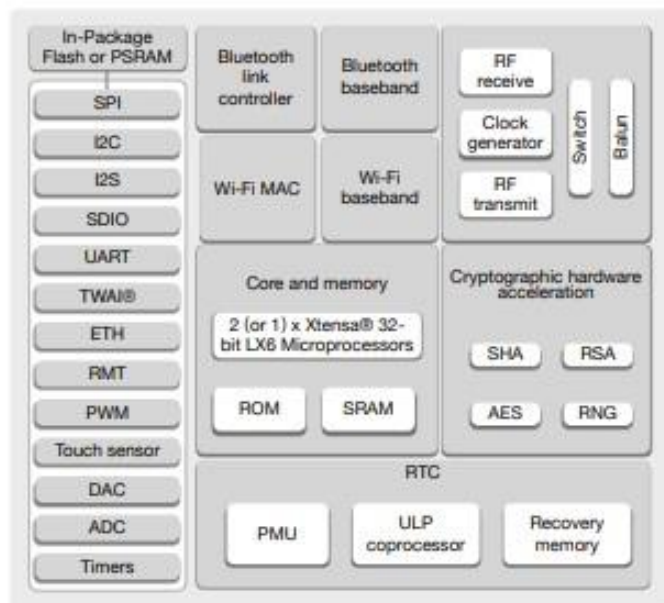
ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

The ESP32 series of chips includes ESP32-D0WD-V3, ESP32-D0WDR2-V3, ESP32-U4WDH, ESP32-S0WD [\(NRND\)](#), ESP32-D0WDQ6-V3 [\(NRND\)](#), ESP32-D0WD [\(NRND\)](#), and ESP32-D0WDQ6 [\(NRND\)](#), among which,

- ESP32-S0WD [\(NRND\)](#), ESP32-D0WD [\(NRND\)](#), and ESP32-D0WDQ6 [\(NRND\)](#) are based on chip revision v1 or chip revision v1.1.
- ESP32-D0WD-V3, ESP32-D0WDR2-V3, ESP32-U4WDH, and ESP32-D0WDQ6-V3 [\(NRND\)](#) are based on chip revision v3.0 or chip revision v3.1.

For details on part numbers and ordering information, please refer to Section 1 [ESP32 Series Comparison](#). For details on chip revisions, please refer to [ESP32 Chip Revision v3.0 User Guide](#) and [ESP32 Series SoC Errata](#).

The functional block diagram of the SoC is shown below.



ESP32 Functional Block Diagram

Features

Wi-Fi

- 802.11b/g/n
- 802.11n (2.4 GHz), up to 150 Mbps
- WMM
- TX/RX A-MPDU, RX A-MSDU
- Immediate Block ACK
- Defragmentation
- Automatic Beacon monitoring (hardware TSF)
- 4 × virtual Wi-Fi interfaces
- Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes
Note that when ESP32 is in Station mode, performing a scan, the SoftAP channel will be changed.
- Antenna diversity

Bluetooth®

- Compliant with Bluetooth v4.2 BR/EDR and Bluetooth LE specifications
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced Power Control
- +9 dBm transmitting power
- NZIF receiver with -94 dBm Bluetooth LE sensitivity
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- Bluetooth 4.2 BR/EDR and Bluetooth LE dual mode controller
- Synchronous Connection-Oriented/Extended (SCO/eSCO)
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet
- Multi-connections in Classic Bluetooth and Bluetooth LE
- Simultaneous advertising and scanning

CPU and Memory

- Xtensa® single-/dual-core 32-bit LX6 microprocessor(s)
- CoreMark® score:
 - 1 core at 240 MHz: 504.85 CoreMark; 2.10 CoreMark/MHz

- 2 cores at 240 MHz: 994.26 CoreMark; 4.14 CoreMark/MHz
- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC
- QSPI supports multiple flash/SRAM chips

Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz ~ 60 MHz crystal oscillator (40 MHz only for Wi-Fi/Bluetooth functionality)
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 × 64-bit timers and 1 × main watchdog in each group
- One RTC timer
- RTC watchdog

Advanced Peripheral Interfaces

- 34 × programmable GPIOs
 - 5 strapping GPIOs
 - 6 input-only GPIOs
 - 6 GPIOs needed for in-package flash/PSRAM (ESP32-D0WDR2-V3, ESP32-U4WDH)
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DAC
- 10 × touch sensors
- 4 × SPI
- 2 × I2S
- 2 × I2C
- 3 × UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- TWAI[®], compatible with ISO 11898-1 (CAN Specification 2.0)
- RMT (TX/RX)
- Motor PWM
- LED PWM up to 16 channels

Power Management

- Fine-resolution power control through a selection of clock frequency, duty cycle, Wi-Fi operating modes, and individual power control of internal components
- Five power modes designed for typical scenarios: Active, Modem-sleep, Light-sleep, Deep-sleep, Hibernation
- Power consumption in Deep-sleep mode is 10 μ A
- Ultra-Low-Power (ULP) coprocessors
- RTC memory remains powered on in Deep-sleep mode

Security

- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration:
 - AES
 - Hash (SHA-2)
 - RSA
 - ECC
 - Random Number Generator (RNG)

Applications

With low power consumption, ESP32 is an ideal choice for IoT devices in the following areas:

- | | |
|-------------------------|--|
| • Smart Home | • Generic Low-power IoT Sensor Hubs |
| • Industrial Automation | • Generic Low-power IoT Data Loggers |
| • Health Care | • Cameras for Video Streaming |
| • Consumer Electronics | • Speech Recognition |
| • Smart Agriculture | • Image Recognition |
| • POS machines | • SDIO Wi-Fi + Bluetooth Networking Card |
| • Service robot | • Touch and Proximity Sensing |
| • Audio Devices | |

2 Pins

2.1 Pin Layout

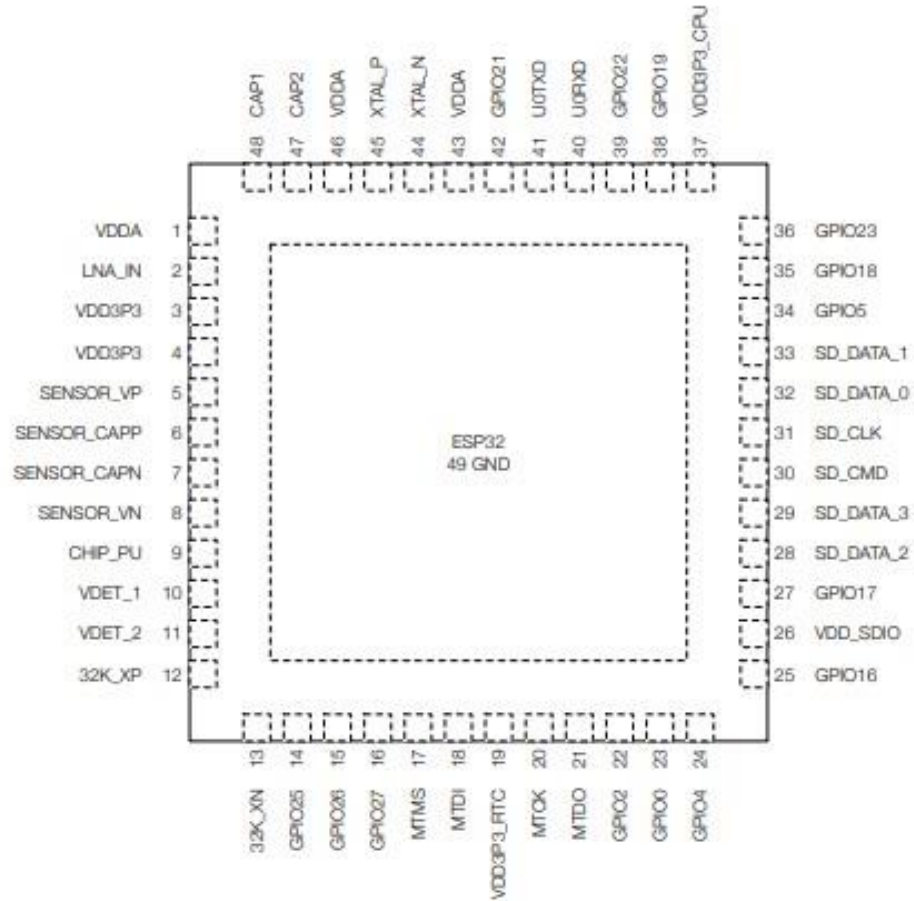


Figure 2-1. ESP32 Pin Layout (QFN 6*6, Top View)

2.2 Pin Overview

Table 2-1. Pin Overview

Name	No.	Type	Function
Analog			
VDDA	1	P	Analog power supply (2.3 V ~ 3.6 V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3	4	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_CAPP	6	I	GPIO37, ADC1_CH1, RTC_GPIO1
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, RTC_GPIO2
SENSOR_VN	8	I	GPIO39, ADC1_CH3, RTC_GPIO3
CHIP_PU	9	I	High: On; enables the chip Low: Off; the chip shuts down Note: Do not leave the CHIP_PU pin floating.
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	I/O	GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	I/O	GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0
GPIO26	15	I/O	GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS
MTDI	18	I/O	GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIQ, HS2_DATA2, SD_DATA2, MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V ~ 3.6 V)
MTCK	20	I/O	GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK
MTDO	21	I/O	GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPIC0, HS2_CMD, SD_CMD, MTDO

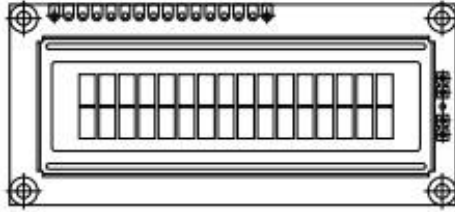
Name	No.	Type	Function
GPIO2	22	I/O	GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, RTC_GPIO11, TOUCH1, EMAC_TX_CLK, CLK_OUT1
GPIO4	24	I/O	GPIO4, ADC2_CH0, RTC_GPIO10, TOUCH0, EMAC_TX_ER, HSPiHD, HS2_DATA1, SD_DATA1
VDD_SDIO			
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	Output power supply: 1.8 V or the same voltage as VDD3P3_RTC
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPIHD
SD_DATA_3	29	I/O	GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPIWP
SD_CMD	30	I/O	GPIO11, HS1_CMD, U1RTS, SD_CMD, SPIC0
SD_CLK	31	I/O	GPIO6, HS1_CLK, U1CTS, SD_CLK, SPICLK
SD_DATA_0	32	I/O	GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPIQ
SD_DATA_1	33	I/O	GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPID
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, HS1_DATA6, VSPIC0, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, HS1_DATA7, VSPICLK
GPIO23	36	I/O	GPIO23, HS1_STROBE, VSPID
VDD3P3_CPU	37	P	Input power supply for CPU IO (1.8 V ~ 3.6 V)
GPIO19	38	I/O	GPIO19, U0CTS, VSPiQ, EMAC_TXD0
GPIO22	39	I/O	GPIO22, U0RTS, VSPWP, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPiHD, EMAC_TX_EN
Analog			
VDDA	43	P	Analog power supply (2.3 V ~ 3.6 V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Analog power supply (2.3 V ~ 3.6 V)
CAP2	47	I	Connects to a 3.3 nF (10%) capacitor and 20 kΩ resistor in parallel to CAP1

Name	No.	Type	Function
CAP1	48	I	Connects to a 10 nF series capacitor to ground
GND	49	P	Ground

Regarding highlighted cells, see Section 2.2.1 Restrictions for GPIOs and RTC_GPIOs.

For a quick reference guide to using the IO_MUX, Ethernet MAC, and GPIO Matrix pins of ESP32, please refer to Appendix ESP32 Pin Lists.

16 x 2 Character LCD



FEATURES

- Type: Character
- Display format: 16 x 2 characters
- Built-in controller: ST 7066 (or equivalent)
- Duty cycle: 1/16
- 5 x 8 dots includes cursor
- + 5 V power supply
- LED can be driven by pin 1, pin 2, or A and K
- N.V. optional for + 3 V power supply
- Optional: Smaller character size (2.95 mm x 4.35 mm)
- Material categorization: For definitions of compliance please see www.vishay.com/doc799912



RoHS
COMPLIANT

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0 x 13.2 (max.)	mm
Viewing Area	66.0 x 16.0	
Dot Size	0.55 x 0.65	
Dot Pitch	0.60 x 0.70	
Mounting Hole	75.0 x 31.0	
Character Size	2.95 x 5.55	

ABSOLUTE MAXIMUM RATINGS					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	V_{DD} to V_{SS}	- 0.3	-	13	V
Input Voltage	V_I	V_{SS}	-	V_{DD}	

Note

- $V_{SS} = 0\text{ V}$, $V_{DD} = 5.0\text{ V}$

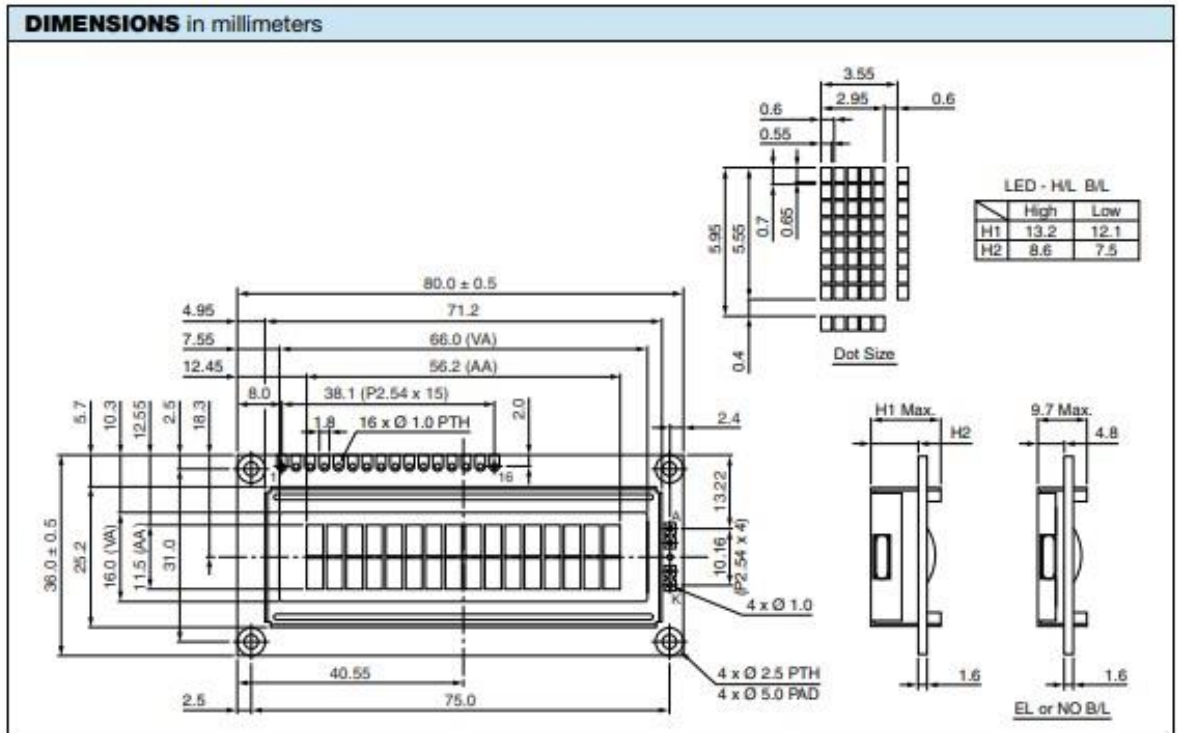
ELECTRICAL CHARACTERISTICS						
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TYP.	MAX.	
Input Voltage	V_{DD}	$V_{DD} = + 5\text{ V}$	4.5	5.0	5.5	V
Supply Current	I_{DD}	$V_{DD} = + 5\text{ V}$	1.0	1.2	1.5	mA
Recommended LC Driving Voltage for Normal Temperature Version Module	V_{DD} to V_0	- 20 °C	-	-	5.2	V
		0 °C	-	-	-	
		25 °C	-	3.7	-	
		50 °C	-	-	-	
		70 °C	3.1	-	-	
LED Forward Voltage	V_F	25 °C	-	4.2	4.6	V
LED Forward Current - Array	I_F	25 °C	-	100	-	mA
LED Forward Current - Edge			-	20	40	
EL Power Supply Current	I_{EL}	$V_{EL} = 110\text{ V}_{AC}$, 400 Hz	-	-	5.0	mA

DISPLAY CHARACTER ADDRESS CODE

Display Position

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
DD RAM Address	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

INTERFACE PIN FUNCTION		
PIN NO.	SYMBOL	FUNCTION
1	V _{SS}	Ground
2	V _{DD}	Supply voltage for logic
3	V ₀	Operating voltage for LCD
4	RS	H: Data/L: Instruction code
5	R/ \bar{W}	H: Read (MPU → Module)/L: Write (MPU → Module)
6	E	H → L chip enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Supply power for LED+
16	R	Supply power for Red-
17	G	Supply power for Green-
18	B	Supply power for Blue-





S3003 FUTABA SERVO



...S3003 FUTABA SERVO...

Detailed Specifications

Control System:	+Pulse Width Control 1520usec Neutral	Current Drain (4.8V):	7.2mA/idle
Required Pulse:	3-5 Volt Peak to Peak Square Wave	Current Drain (6.0V):	8mA/idle
Operating Voltage:	4.8-6.0 Volts	Direction:	Counter Clockwise/Pulse Traveling 15201900usec
Operating Temperature Range:	-20 to +60 Degree C	Motor Type:	3 Pole Ferrite
Operating Speed (4.8V):	0.23sec/60 degrees at no load	Potentiometer Drive:	Indirect Drive
Operating Speed (6.0V):	0.19sec/60 degrees at no load	Bearing Type:	Plastic Bearing
Stall Torque (4.8V):	44 oz/in. (3.2kg.cm)	Gear Type:	All Nylon Gears
Stall Torque (6.0V):	56.8 oz/in. (4.1kg.cm)	Connector Wire Length:	12"
Operating Angle:	45 Deg. one side pulse traveling 400usec	Dimensions:	1.6" x 0.8" x 1.4" (41 x 20 x 36mm)
360 Modifiable:	Yes	Weight:	1.3oz. (37.2g)

ประวัติผู้เขียน



ชื่อ-นามสกุล

วัน เดือน ปีเกิด

ที่ปัจจุบัน

ประวัติการศึกษา

ช่องทางการติดต่อ

นายกฤษทบวง เณรวงษ์

29 กรกฎาคม 2545

53/6 หมู่ 1 ต.นาชะอัง อ.เมือง จ.ชุมพร

พ.ศ. 2561 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น

จากโรงเรียนศรีราษฎร์ จังหวัดชุมพร

พ.ศ. 2564 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย

จากโรงเรียนศรีราษฎร์ จังหวัดชุมพร

Tel. 0980169099

Email: 64200002@kmitl.ac.th



ชื่อ-นามสกุล

วัน เดือน ปีเกิด

ที่ปัจจุบัน

ประวัติการศึกษา

ช่องทางการติดต่อ

นายชยากุล ธรรมสวัสดิ์

3 เมษายน 2545

ที่อยู่ 141/2 หมู่ 4 ต.วิสัยใต้ อ.สวี จ.ชุมพร 86130

พ.ศ. 2561 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น

จากโรงเรียนศรีวิทยาภัย จังหวัดชุมพร

พ.ศ. 2564 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย

จากโรงเรียนศรีวิทยาภัย จังหวัดชุมพร

Tel. 0653494262

Email: 64200039@kmitlac.th



ชื่อ-นามสกุล

วัน เดือน ปีเกิด

ที่ปัจจุบัน

ประวัติการศึกษา

ช่องทางการติดต่อ

นายณัฐ แจ่มจิตร

3 ตุลาคม 2545

ห้อง 206 หอพรรณิ เรสซิเดนซ์ 201 หมู่ที่ 6 ตำบลชุมโค

อำเภอปะทิว จังหวัดชุมพร 86160

พ.ศ. 2564 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น

จากโรงเรียนวัดราชพิธ จังหวัดกรุงเทพ

Tel. 0918503440

Email: 64200055@kmitLac.th



ชื่อ-นามสกุล

วัน เดือน ปีเกิด

ที่ปัจจุบัน

ประวัติการศึกษา

ช่องทางการติดต่อ

นายกฤษณ์ คงคา

23 สิงหาคม 2545

ห้อง 206 หอพรรณิ เรสซิเดนซ์ 201 หมู่ที่ 6 ตำบลชุมโค
อำเภอปะทิว จังหวัดชุมพร 86160

พ.ศ. 2564 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย
จากโรงเรียนเซนต์คาเบรียล จังหวัดกรุงเทพ

Tel. 0936525491

Email: 64200289@kmitl.ac.th



ชื่อ-นามสกุล

วัน เดือน ปีเกิด

ที่ปัจจุบัน

ประวัติการศึกษา

ช่องทางการติดต่อ

นายวารินทร์ พรหมพิชัย

25 พฤษภาคม 2543

143/15 หมู่ 11 ซอยปริมินทร์มรรคา 12 ต.บางหมาก
อ.เมือง จ.ชุมพร

พ.ศ. 2558 สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้น
จากโรงเรียนนิรมลชุมพร จังหวัดชุมพร

พ.ศ. 2561 สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย
จากโรงเรียนสอาดเผดิมวิทยา จังหวัดชุมพร

Tel. 0980615223

Email: 64200358@kmitl.ac.th