

ทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ
AUTOMATIC BUOYANCY FISH FEEDER



โดย
นางสาวชนิษฐา
นางสาวชลธิชา
นางสาวรัชนิกร
บรรเทพ
สุดประเวศ
โมอ่อน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ
AUTOMATIC BUOYANCY FISH FEEDER

โดย

นางสาวชนิษฐา	บรรเทพ	63010098
นางสาวชลธิชา	สุดประเวศ	63010207
นางสาวรัชนิกร	โมอ่อน	63010811

อาจารย์ที่ปรึกษา

รศ.ดร. พิพัฒน์ พรหมมี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2566

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

AUTOMATIC BUOYANCY FISH FEEDER

ผู้จัดทำ

- | | | |
|------------------|-----------|----------|
| 1. นางสาวณิชฐา | บรรเทพ | 63010098 |
| 2. นางสาวชลธิชา | สุดประเวศ | 63010207 |
| 3. นางสาวรัชนิกร | โมอ่อน | 63010811 |



..... อาจารย์ที่ปรึกษา

(รศ.ดร. พิพัฒน์ พรหมมี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ
AUTOMATIC BUOYANCY FISH FEEDER

โดย นางสาวณิชฐา บรรเทพ 63010098
นางสาวชลธิชา สุดประเวศ 63010207
นางสาวรัชณีกร โมอ่อน 63010811

อาจารย์ที่ปรึกษา รศ.ดร. พิพัฒน์ พรหมมี

บทคัดย่อ

ปฏิญานิพนธ์นี้คณะผู้จัดทำมีแนวคิดที่จะทำทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ เนื่องจากในปัจจุบันการให้อาหารปลานั้นใช้วิธีตักหรือเทอาหารลงไปในบ่อเลี้ยงปลาเพื่อให้อาหารกับปลาโดยตรง ปัญหาที่เกิดกับเกษตรกรผู้เลี้ยงปลาเป็นการใช้แรงงานคนในการเลี้ยงปลา ในบางครั้งอาจให้อาหารปลาในปริมาณที่มากหรือน้อยจนเกินไป ทำให้ปลากินอาหารไม่พอดีหรือไม่หมด จึงทำให้สิ้นเปลืองและทำให้น้ำในบ่อเลี้ยงปลาเกิดการเน่าเสียได้ และตำแหน่งการให้อาหารในบ่อปลา เป็นปัจจัยหนึ่งที่สำคัญต่อการเลี้ยงปลา จึงได้ออกแบบและสร้างทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ ที่สามารถลดภาระการใช้แรงงานคนในการให้อาหารปลา โดยใช้บอร์ดไมโครคอนโทรลเลอร์ ESP-32 เพื่อส่งข้อมูลไปบน APPLICATION ผ่านเว็บ MIT และบอร์ด Arduino Uno ควบคุมการทำงานของทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ และควบคุมทิศทางการทำงานของทุ่นลอยน้ำให้อาหารปลาอัตโนมัติ ใช้แหล่งจ่ายไฟจากแบตเตอรี่และโซลาร์เซลล์จ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 และบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno ใช้โมดูล Gps ในการรับสัญญาณจาก Gps รับค่าพิกัดละติจูด ลองจิจูด เพื่อระบุตำแหน่งบน Application ผ่าน Smart Phone ให้ทุ่นลอยน้ำให้อาหารปลาอัตโนมัติเคลื่อนที่ไปข้างหน้าด้วยแรงดันน้ำจาก Dc Pump ใช้วาล์วควบคุมทิศทางการเลี้ยวของทุ่นลอยน้ำ แล้วทำการปล่อยอาหารปลาผ่านมอเตอร์ และสร้างออกซิเจนให้กับบ่อเลี้ยงปลาด้วยการทำงานของ DC Pump

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

This Project introduces an automatic buoyancy fish feeder based on microcontroller. The fish farmers are required human labor. Sometimes, they may feed too much or too less food to the fish. If the fish receive not enough food, the fish become not standard and the water will be wasted. Feeding location is also an important factor in fish farming, therefore an automatic buoyancy fish feeder has been designed and constructed. The automatic buoyancy fish feeder uses an esp-32 microcontroller and Arduino UNO as a main microcontroller. For controlling the direction. Battery and solar cells supplier the power to the esp-32 and Arduino UNO, a GPS module for received the latitude and longitude coordinates. The location of the automatic buoyancy fish feeder can be controlled by the application through a smartphone and also generating the oxygen into the water.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	IX
สารบัญตาราง	XII
บทที่ 1	
บทนำ	
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญาานิพนธ์	2
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	
2.1 อัตราการให้อาหารต่อวันของปลานิลและการคำนวณปริมาณอาหารที่ ให้ปลานิล	3
2.1.1 การคำนวณปริมาณอาหาร	4
2.2 บอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM	4
2.3 บอร์ดไมโครคอนโทรลเลอร์ ARDUINO	6
2.3.1 ส่วนประกอบของบอร์ดไมโครคอนโทรลเลอร์ ARDUINO	7
2.4 โมดูลระบุตำแหน่งพิกัด GPS GY-NEO6MV2 NEO-6M	9
2.5 โมดูลเข็มทิศ GY-273	10
2.6 แผงพลังงานแสงอาทิตย์	11
2.6.1 ประเภทของแผงพลังงานแสงอาทิตย์	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.2 คุณสมบัติและตัวแปรที่สำคัญของแผงพลังงานแสงอาทิตย์	13
2.6.3 ขั้นตอนการทำงานของระบบแผงพลังงานแสงอาทิตย์เพื่อนำพลังงานไปใช้	14
2.6.4 การคำนวณระบบแผงพลังงานแสงอาทิตย์	15
2.7 โซลาร์ชาร์จคอนโทรลเลอร์	15
2.8 แบตเตอรี่ลิเธียม	17
2.8.1 ชนิดของแบตเตอรี่	17
2.8.2 หลักการทำงานของแบตเตอรี่ลิเธียม	18
2.8.3 ข้อดีและข้อควรระวังการใช้แบตเตอรี่ลิเธียมไอออน	19
2.8.4 สูตรคำนวณ	19
2.9 IRF540	19
2.9.1 โครงสร้างของIRF 540	20
2.9.2 การทำงานของ IRF 540	20
2.9.3 การคำนวณหากระแสที่ผ่านโหลด	21
2.10 LM2596	21
2.11 โซลินอยด์วาล์ว	23
2.11.1 การทำงานของโซลินอยด์วาล์ว	23
2.11.2 ประเภทโซลินอยด์วาล์ว	23
2.12 DC PUMP	24
2.12.1 ประเภท DC PUMP	24
2.13 มอเตอร์ไฟฟ้ากระแสตรง	26
2.13.1 การทำงานของมอเตอร์ไฟฟ้ากระแสตรง	26
2.13.2 ควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.14 โปรแกรมที่ใช้ในปริญญาานิพนธ์	27
2.14.1 ARDUINO IDE	27
2.14.2 PROTEUS	28
2.14.3 AUTOCAD	30
2.14.4 MIT APP INVERTER	30
บทที่ 3 การออกแบบและการจัดทำปริญญาานิพนธ์	
3.1 การออกแบบ	32
3.1.1 การออกแบบและสร้างหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	33
3.1.2 การออกแบบระบบควบคุมขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	36
3.1.3 การออกแบบถังให้อาหารปลา	39
3.2 เครื่องมือที่ใช้ในการทดลอง	39
3.2.1 ไมโครคอนโทรลเลอร์	39
3.2.2 มอเตอร์ไฟฟ้ากระแสตรง	40
3.2.3 โมดูลระบุตำแหน่งพิกัด GPS	40
3.2.4 โมดูลเข็มทิศ	40
3.2.5 วงจรแปลงกระแสไฟ	40
3.2.6 แผงพลังงานแสงอาทิตย์	40
3.2.7 โซลาร์ชาร์จคอนโทรลเลอร์	40
3.2.8 แบตเตอรี่	40
3.2.9 DC PUMP	41
3.3 การจัดเก็บผลการทดลอง	41
3.3.1 ผลการทดสอบการทำงานของโมดูลระบุตำแหน่งพิกัด GPS	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.3.2 ผลการทดสอบการทำงานของไมโครเคมีทิส	41
3.3.3 ผลการทดสอบการชาร์จแผงพลังงานแสงอาทิตย์เข้ากับ แบตเตอรี่	41
3.3.4 ผลการการทำงานของ DC PUMP	41
3.3.5 ผลการทดสอบการปล่อยให้อาหารปลาและระบบสร้าง ออกซิเจนในบ่อ	41
3.3.6 ผลการทดสอบการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด	42
3.3.7 ผลการทดสอบการทำงานของฟุนลายน้ําให้อาหารปลา อัตโนมัติ	42
3.3.8 ผลการทดสอบการทำงานของแอปพลิเคชัน	42
บทที่ 4 ผลการทดลอง	
4.1 ผลการทดสอบการทำงานของไมโครระบุตำแหน่งพิกัด GPS	43
4.2 ผลการทดสอบการทำงานของไมโครเคมีทิส	45
4.3 ผลการทดสอบการชาร์จแผงพลังงานแสงอาทิตย์เข้ากับ แบตเตอรี่	47
4.4 ผลการการทำงานของ DC PUMP	49
4.5 ผลการทดสอบระบบปล่อยอาหารปลาและระบบสร้างออกซิเจน ในบ่อ	50
4.6 ผลการทดสอบการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด	52
4.7 ผลการทดสอบการทำงานของฟุนลายน้ําให้อาหารปลาอัตโนมัติ	54
4.8 ผลการทดสอบการทำงานของแอปพลิเคชัน	57
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 สรุปผล	59
5.2 ข้อเสนอแนะ	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บรรณานุกรม	61
ภาคผนวก	
ภาคผนวก ก คำสั่งการทดสอบการทำงานของโมดูลระบุตำแหน่งพิกัด GPS	64
ภาคผนวก ข คำสั่งการทดสอบการทำงานของโมดูลเข็มทิศ	66
ภาคผนวก ค คำสั่งการทดสอบการขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	68
ภาคผนวก ง คำสั่งรับค่าจากไมโครคอนโทรลเลอร์ ESP 32 และคำสั่งการทำงานของหุ่นลอยน้ำ	71
ภาคผนวก จ คำสั่งรับค่าจากแอปพลิเคชัน	84

สารบัญรูป

รูปที่	หน้า
2.1	5
2.2	5
2.3	7
2.4	9
2.5	9
2.6	11
2.7	12
2.8	13
2.9	13
2.10	16
2.11	16
2.12	17
2.13	18
2.14	20
2.15	20
2.16	22
2.17	23
2.18	25
2.19	26
2.20	27
2.21	28
2.22	29
2.23	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า	
2.24	สัญลักษณ์ของ MIT APP INVERTER	31
3.1	บล็อกไดอะแกรมและอุปกรณ์การทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	32
3.2	ภาพจำลอง 3 มิติโครงสร้างของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	33
3.3	ภาพจำลอง 3 มิติโครงสร้างด้านหน้าของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	34
3.4	ภาพจำลอง 3 มิติโครงสร้างด้านข้างของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	34
3.5	ภาพจำลอง 3 มิติโครงสร้างด้านบนของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	35
3.6	ภาพถ่ายโครงสร้างของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	35
3.7	บล็อกไดอะแกรมระบบควบคุมขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	36
3.8	บล็อกไดอะแกรมการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	37
3.9	ออกแบบควบคุมการทำงานของโซลินอยด์	38
3.10	ออกแบบการขับเคลื่อนของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ	39
3.11	อุปกรณ์ในการปล่อยอาหารปลา	39
4.1	เชื่อมต่อโมดูลระบุตำแหน่งพิกัด GPS รุ่น GY-NEO6MV2 เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32	43
4.2	แสดงค่าละติจูด ลองจิจูดบนหน้าจอ SERIAL MONITOR	44
4.3	การทดสอบความถูกต้องชุดรับสัญญาณ GPS	44
4.4	การทดสอบระยะเวลาในการรับสัญญาณจากดาวเทียม	44
4.5	เชื่อมต่อโมดูลวัดสนามแม่เหล็ก 3 แกน โมดูลเข็มทิศ GY-273 ใช้ชิป HMC5883L เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32	46
4.6	แสดงค่าองศาบนหน้าจอ SERIAL MONITOR	46
4.7	การทดสอบความถูกต้องของโมดูลเข็มทิศ	46
4.8	การชาร์จแผงพลังงานแสงอาทิตย์เข้ากับแบตเตอรี่	48

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.9 การเชื่อมต่อ DC PUMP เข้ากับแผงพลังงานแสงอาทิตย์ แบตเตอรี่ และโซลาร์ชาร์จคอนโทรลเลอร์	49
4.10 การปล่อยอาหารปลา โดยใช้มอเตอร์ 12 โวลต์ ในการหมุนสปริงเกลียว เพื่อปล่อยอาหารปลา	50
4.11 อาหารปลาที่มีเส้นผ่าศูนย์กลาง 0.7 เซนติเมตร ใช้ในการทดสอบการปล่อยอาหารปลา	50
4.12 การทดสอบการปล่อยอาหารปลา	51
4.13 การสร้างออกซิเจนในบ่อเลี้ยงปลา	51
4.14 ตำแหน่งจุดอ้างอิงที่ต้องการเคลื่อนที่	53
4.15 การทดสอบความคลาดเคลื่อนของการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด	53
4.16 ตัวอย่างการเคลื่อนที่ไปด้านหน้า	54
4.17 ผลจาก SERIAL MONITOR	54
4.18 ตัวอย่างการเคลื่อนที่ไปด้านซ้าย	55
4.19 ผลจาก SERIAL MONITOR	55
4.20 ตัวอย่างการเคลื่อนที่ไปด้านขวา	56
4.21 ผลจาก SERIAL MONITOR	56
4.22 หน้าแอปพลิเคชันการทำงาน	58

สารบัญตาราง

ตารางที่	หน้า
2.1	4
2.2	5
2.3	7
2.4	8
2.5	10
2.6	11
2.7	15
2.8	16
2.9	19
2.10	21
2.11	22
2.12	24
2.13	25
2.14	27
3.1	38
4.1	45
4.2	47

สารบัญตาราง (ต่อ)

ตารางที่	หน้า
4.3 ผลการทดสอบการชาร์จแผงพลังงานแสงอาทิตย์ 50 วัตต์ เข้ากับ แบตเตอรี่ 12 โวลต์	48
4.4 ผลการทดสอบอัตราการไหลของน้ำโดยใช้ท่อ PVC ที่มีขนาด เส้นผ่าศูนย์กลางภายนอก 1 นิ้ว 1 ทาง และผลการทดสอบอัตราการไหล ของน้ำโดยใช้ท่อ PVC ที่มีขนาดเส้นผ่าศูนย์กลางภายนอก 1 นิ้ว 2 ทาง	49
4.5 ผลการทดสอบการปล่อยอาหารปลา	52
4.6 ผลการทดสอบการวัดกระแสการทำงานของโซลีนอยด์เมื่อทำการสั่งการใช้ งาน 1 ตัว, 2 ตัว, 3 ตัว, 4 ตัว และ 5 ตัว แบบไม่รวมกระแส DC PUMP	56
4.7 ผลการทดสอบการวัดกระแสการทำงานของโซลีนอยด์เมื่อทำการสั่งการใช้ งาน 1 ตัว, 2 ตัว, 3 ตัว, 4 ตัว และ 5 ตัว แบบรวมกระแส DC PUMP	57
4.8 ผลการทำงานของฟุนลอยน้ำให้อาหารปลาอัตโนมัติ	57

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการเพาะเลี้ยงสัตว์น้ำเป็นเกษตรกรรมที่มีความสำคัญต่อเศรษฐกิจของประเทศไทยเป็นอย่างมากโดยเฉพาะการเพาะเลี้ยงปลา นิล ปัญหาที่พบบ่อยคือเกษตรกรขาดแคลนงบประมาณในการจ้างแรงงานคนในการเลี้ยงปลา ดูแลปลาและให้อาหารปลา ในบางครั้งอาจให้อาหารปลาในปริมาณที่มากหรือน้อยจนเกินไป ทำให้ปลากินอาหารไม่พอดีหรือไม่หมด จึงทำปลารับสารอาหารที่ไม่เพียงพอหรือสิ้นเปลืองงบประมาณและส่งผลกระทบต่อปลาในบ่อเลี้ยงปลาเกิดการเน่าเสียได้ ซึ่งเป็นปัญหาอย่างหนักของเกษตรกรที่เลี้ยงปลา

ผู้จัดทำจึงมีแนวคิดในการทำหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติขึ้นมา เพื่อมาแก้ปัญหาที่เกิดขึ้นโดยสิ่งประดิษฐ์นี้ยังสอดคล้องกับเทคโนโลยียุคปัจจุบัน เพราะเล็งเห็นว่าหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติ จะสามารถช่วยลดต้นทุนในการจ้างแรงงานและช่วยลดภาระงานในการให้อาหารปลาของเกษตรกร

โดยหลักการการทำงานของหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติ ใช้บอร์ดไมโครคอนโทรลเลอร์ ESP-32 และบอร์ดไมโครคอนโทรลเลอร์ Arduino UNO ควบคุมการทำงานของหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติ และควบคุมทิศทางการทำงานของหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติ โดยใช้แหล่งจ่ายไฟจากโซลาร์เซลล์และแบตเตอรี่ จ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 และบอร์ดไมโครคอนโทรลเลอร์ Arduino UNO ใช้ Module GPS ในการรับสัญญาณจาก GPS รับค่าพิกัดละติจูด ลองจิจูด เพื่อระบุตำแหน่งบน Application ผ่าน Smart Phone ให้หุ่นยนต์น้ำให้อาหารปลาอัตโนมัติเคลื่อนที่ไปข้างหน้าด้วยแรงดันน้ำจาก DC Pump ใช้โซลินอยด์วาล์วควบคุมทิศทางการเลี้ยงของหุ่นยนต์น้ำ และสร้างออกซิเจนให้กับบ่อเลี้ยงปลาด้วยการทำงานของ DC Pump

1.2 วัตถุประสงค์

- 1) เพื่อออกแบบและสร้างหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติตามจุดที่กำหนดไว้ในแผนที่
- 2) เพื่อลดภาระงานในการให้อาหารปลาของเกษตรกร
- 3) เพื่อสามารถเพิ่มออกซิเจนในบ่อปลาได้โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญาโท

ศึกษาการออกแบบและพัฒนาสิ่งประดิษฐ์หุ่นลอยน้ำให้อาหารปลาอัตโนมัติ สามารถกำหนดปริมาณการให้อาหารในแต่ละช่วงวัยของปลานิลได้ ควบคุมการเคลื่อนที่ของหุ่นโดยกำหนดจุดในการให้อาหารปลานิลแบบอัตโนมัติได้จากพิกัดละติจูดและลองจิจูดเมื่อไม่มีการให้อาหารปลานิลสามารถบำบัดน้ำเสียโดยการผลิตออกซิเจนได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์เรื่อง “ฟุนลอยน้ำให้อาหารปลาอัตโนมัติ” ได้ทำการออกแบบและสร้างฟุนลอยน้ำให้อาหารปลาอัตโนมัติที่สามารถกำหนดปริมาณการให้อาหารในแต่ละช่วงวัยของปลานิลได้ ควบคุมการเคลื่อนที่ของฟุนลอยน้ำให้อาหารปลาอัตโนมัติโดยกำหนดจุดในการให้อาหารปลานิลแบบอัตโนมัติได้จาก พิกัดละติจูด ลองจิจูดและเมื่อไม่มีการให้อาหารปลานิลสามารถบำบัดน้ำเสียโดยการผลิตออกซิเจนได้ ซึ่งมีทฤษฎีและหลักการดังต่อไปนี้

2.1 อัตราการให้อาหารต่อวันของปลานิลและการคำนวณปริมาณอาหารที่ให้ปลานิล

ปลานิลเป็นปลาที่กินอาหารได้ทุกชนิด ดังนั้นปลาชนิดนี้จึงเป็นปลาที่ให้ผลผลิตสูง โดยเฉพาะอาหารธรรมชาติที่มีอยู่ในบ่อ เช่น ไรน้ำ ตะไคร่น้ำ ตัวอ่อนของแมลงและสัตว์เล็กๆ ที่อยู่ในบ่อ ตลอดจนสาหร่ายและแพลง การให้อาหารแต่ละครั้งไม่ควรให้ปริมาณมากจนเกินไปควรประมาณ ให้มีปริมาณเพียงพอต่อความต้องการของปลา ควรให้อาหารในปริมาณน้ำหนักราว 5% ของน้ำหนักปลาที่เลี้ยง ถ้าให้อาหารมากเกินไป ปลาจะกินไม่หมด ทำให้เสียค่าอาหารไปโดยเปล่าประโยชน์ และยังทำให้น้ำเน่าเสีย ก่อให้เกิดอันตรายแก่ปลาได้ อาหารที่ใช้ในการเลี้ยงปลานิลต้องเป็นอาหารที่มีโปรตีนสูง สามารถเลือกให้เหมาะสมกับปัจจัยต่าง ๆ ได้ดังนี้

1. ปริมาณของโปรตีนที่เหมาะสมต่อการเจริญเติบโตของปลานิลในแต่ละช่วง ซึ่งต้องเลือกให้แตกต่างกันตามอายุวันของการเลี้ยง ในระดับที่ยังเป็นลูกปลาสามารถเลือกให้อาหารที่มีโปรตีนสูงประมาณ 32-40% แต่เมื่อลูกปลานิลมีขนาดใหญ่ขึ้นเจริญเติบโตมากขึ้นระดับโปรตีนที่เลือกใช้ในการให้อาหารจะลดลงมาอยู่ที่ประมาณ 27-34%

2. ความถี่ของการให้อาหารปลานิล ปลานิลจะสามารถกินอาหารได้ ทีละน้อย ๆ เนื่องจากไม่มีกระเพาะอาหารจริง รวมถึงมีการย่อยที่ค่อนข้างช้า จึงต้องเน้นเรื่องการให้อาหารในปริมาณที่น้อยแต่มีความถี่มากในการให้อาหารปลานิล

3. ช่วงเวลาในการให้อาหารปลานิล ควรเลือกเป็นเวลาเช้าและเย็น

4. อุณหภูมิที่เหมาะสมที่สุดและส่งผลให้ปลากินอาหารได้มากที่สุด ที่ 25-30 องศาเซลเซียส[1]

โดยอัตราการให้อาหารปลาชนิดต่อวันจะแตกต่างกันตามน้ำหนักของปลา แสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 อัตราการให้อาหารปลาชนิดต่อวัน

น้ำหนักปลา (กรัม)	อัตราการให้อาหารต่อวัน (% น้ำหนักปลา)	จำนวนครั้งที่ให้ต่อวัน
0-1	30-10	8
1-5	10-6	6
5-20	6-4	4
20-100	4-3	3-4
>100	3-2	2-3

2.1.1 การคำนวณปริมาณอาหาร

การคำนวณปริมาณอาหารที่ให้ปลาชนิดในแต่ละวันคำนวณได้จาก

1. การคำนวณน้ำหนักปลาทั้งหมด

$$\text{น้ำหนักปลาทั้งหมด} = \text{น้ำหนักปลาแต่ละตัว} \times \text{จำนวนปลาทั้งหมด} \quad (2.1)$$

ในการทดลองนี้สมมติให้น้ำหนักปลาชนิด 500 กรัม จำนวน 100 ตัว จะได้น้ำหนักปลาชนิดทั้งหมด $500 \times 100 = 50,000$ กรัม หรือ 50 กิโลกรัม

2. อัตราการให้อาหารปลาต่อวัน 3% ของน้ำหนักตัวปลาจากตาราง

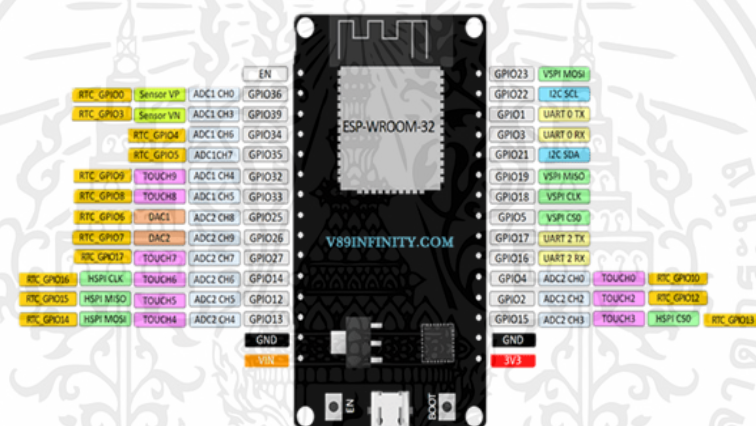
$$\text{ดังนั้นจะใช้อาหาร } 3 \times 50 / 100 = 1.5 \text{ กิโลกรัม/วัน [2]}$$

2.2 บอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM

บอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM เป็นบอร์ดที่พัฒนามาจาก ESP8266 โดยมีทั้ง Wi-Fi และ Bluetooth ในตัวเดียวกัน มีเซนเซอร์สัมผัสในตัวและเซนเซอร์อุณหภูมิในตัว มีความละเอียดในการแปลงข้อมูล Analog Digital Converter (ADC) และหน่วยความจำเพิ่มขึ้น ทำให้การต่อพ่วงกับอุปกรณ์ต่างๆ น้อยลง จึงเหมาะสำหรับเป็นบอร์ดสำหรับทดลอง และใช้งานจริง มีขาใช้งานถึง 38 ขา ดังรูปที่ 2.2 และตัวอย่างของบอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM แสดงดังรูปที่ 2.1 ข้อมูลทางเทคนิคของบอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM แสดงได้ดังตารางที่ 2.2 [5]



รูปที่ 2.1 บอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM [3]



รูปที่ 2.2 ขาการใช้งานของบอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM [3]

ตารางที่ 2.2 ข้อมูลทางเทคนิคของบอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM

ข้อมูลทางเทคนิค	รายละเอียด
สถาปัตยกรรมซีพียู	Xtensa Dual-Core 32 บิต LX6 600DMIPS
แรงดันไฟฟ้า	3.3 โวลต์ – 5 โวลต์
Clock Speed	160 MHz (Default), 240MHz
SRAM	512kB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ข้อมูลทางเทคนิคของบอร์ดไมโครคอนโทรลเลอร์ ESP32-WROOM (ต่อ)

ข้อมูลทางเทคนิค	รายละเอียด
Flash Memory	16MB
ขา GPIO	36 ขา
Hardware PWM	1
Software PWM	16
SPI	4
I2C	2
I2S	2
UART	2
ADC	12 บิต
CAN	1
มาตรฐาน Wi-Fi	802.11 b/g/n (HT40)
มาตรฐาน Bluetooth	Bluetooth 4.2
Ethernet MAC Interface	1
อุณหภูมิการทำงาน	-40 to 125 °C
Touch Sensor & Temperature Sensor	

2.3 บอร์ดไมโครคอนโทรลเลอร์ Arduino

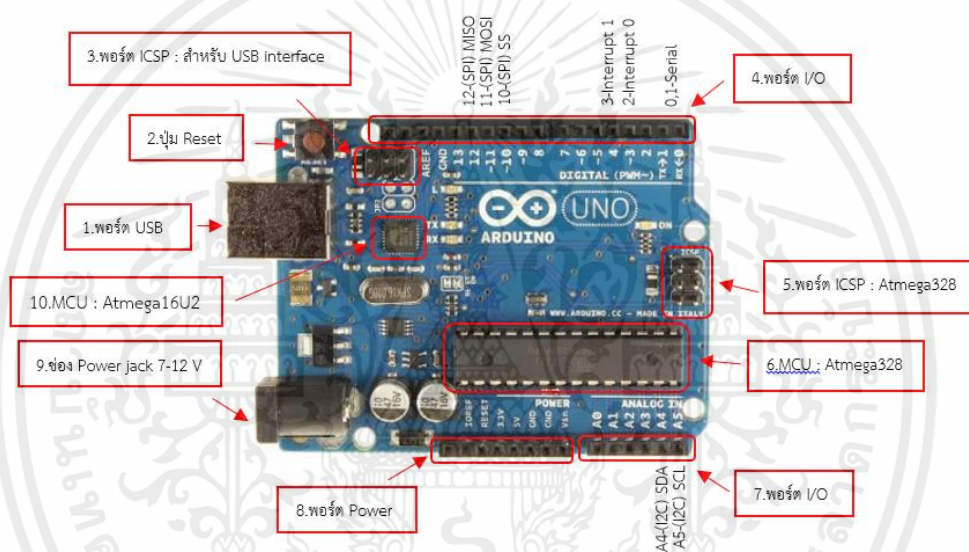
เป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Source คือมีการเปิดเผยข้อมูลทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ ตัวบอร์ด Arduino ถูกออกแบบมาให้ใช้งานได้ง่าย ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดแล้วเขียนโปรแกรมพัฒนาต่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 เป็นบอร์ดมาตรฐานที่นิยมใช้งานมากที่สุด เนื่องจากเป็นขนาดที่เหมาะสมสำหรับการเริ่มต้นเรียนรู้ Arduino และมี Shields ให้เลือกใช้งานได้มากกว่าบอร์ดไมโครคอนโทรลเลอร์ Arduino รุ่นอื่นๆ [6]

2.3.1 ส่วนประกอบของบอร์ดไมโครคอนโทรลเลอร์ Arduino

จากรูปที่ 2.3 แสดงส่วนประกอบของบอร์ดไมโครคอนโทรลเลอร์ Arduino และตารางที่ 2.3 แสดงส่วนประกอบและหน้าที่ของบอร์ดไมโครคอนโทรลเลอร์ Arduino



รูปที่ 2.3 ส่วนประกอบของบอร์ดไมโครคอนโทรลเลอร์ Arduino [7]

ตารางที่ 2.3 รายละเอียดของบอร์ดไมโครคอนโทรลเลอร์ Arduino

ส่วนประกอบของบอร์ด Arduino	หน้าที่
1. USB Port	ใช้สำหรับต่อกับ Computer เพื่ออัปโหลดโปรแกรมเข้า MCU และจ่ายไฟให้กับบอร์ด
2. Reset Button	เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทำงานใหม่
3. ICSP Port	ของ Atmega16U2 เป็นพอร์ตที่ใช้โปรแกรม Visual Com port บน Atmega16U2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 รายละเอียดของบอร์ดไมโครคอนโทรลเลอร์ Arduino (ต่อ)

4. I/O Port	Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx, Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
5. ICSP Port	Atmega328 เป็นพอร์ตที่ใช้โปรแกรม Bootloader
6. MCU	Atmega 328 เป็น MCU ที่ใช้บนบอร์ด Arduino
7. I/O Port	นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
8. Power Port	ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายนอก ประกอบด้วย ขาไฟเลี้ยง +3.3 V, +5V, GND, Vin
9. Power Jack	รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
10. MCU	ของ Atmega16U2 เป็น MCU ที่ทำหน้าที่เป็น USB to Serial โดย Atmega328 จะติดต่อกับ Computer ผ่าน Atmega16U2

ในปฏิญานพนธ์นี้ ผู้จัดทำใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 มาใช้ในการควบคุมการเปิด-ปิดโซลินอยด์ ทำให้หุ่นลอยน้ำให้อาหารปลาอัตโนมัติเคลื่อนที่ และควบคุมมอเตอร์ในการให้อาหารปลา โดยตารางที่ 2.4 แสดงข้อมูลทางเทคนิคของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 [7]

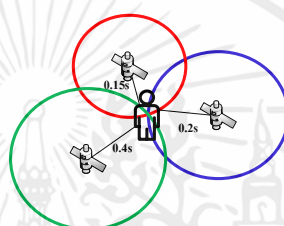
ตารางที่ 2.4 ข้อมูลทางเทคนิคของบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3

ข้อมูลทางเทคนิค	รายละเอียด
ชิปไอซีไมโครคอนโทรลเลอร์	ATmega328
ใช้แรงดันไฟฟ้า	5 โวลต์
รองรับการจ่ายแรงดันไฟฟ้า	7 – 12 โวลต์
รองรับการจ่ายแรงดันไฟฟ้า	6 – 20 โวลต์
พอร์ต Digital I/O	14 พอร์ต (มี 6 พอร์ต PWM output)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โมดูลระบุตำแหน่งพิกัด GPS GY-NEO6MV2 NEO-6M

GPS (Global Positioning System) เป็นระบบ Radionavigation-satellite service คือการส่งคลื่นความถี่จากดาวเทียมลงมาสู่อุปกรณ์บนพื้นโลกที่สามารถทำงานได้ทุกพื้นที่บนโลกและทุกสภาพอากาศ การทำงานของ GPS คือ ดาวเทียมจะส่งสัญญาณคลื่นความถี่ออกมาไปยังเครื่องรับสัญญาณที่อยู่บนพื้นโลก เครื่องรับจะทำการหาสัญญาณดาวเทียมอย่างน้อย 4 ดวง เมื่อจับสัญญาณได้ครบก็จะหาระยะห่างจากเครื่องรับถึงดาวเทียมแต่ละดวงโดยใช้ระยะเวลาในการเดินทางของสัญญาณมาคำนวณ [8] หลักการทำงานแสดงดังรูปที่ 2.4



รูปที่ 2.4 หลักการทำงานของ GPS

โมดูล GPS GY-NEO6MV2 แสดงดังรูปที่ 2.5 สามารถติดตามดาวเทียมได้สูงสุด 22 ดวงใน 50 ช่องสัญญาณ และมีความไวในการติดตามระดับสูงสุดในอุตสาหกรรม เช่น -161 dB ในขณะที่ใช้กระแสไฟเพียง 45 mA สถาปัตยกรรมที่กะทัดรัด ในการออกแบบโมดูล ทำให้โมดูล NEO6MV2 เหมาะสำหรับอุปกรณ์เคลื่อนที่ที่ใช้แบตเตอรี่ ซึ่งมีข้อจำกัดด้านต้นทุนและพื้นที่ โดยข้อมูลทางเทคนิคของโมดูลระบุตำแหน่งพิกัด GPS GY-NEO6M แสดงได้ดังตารางที่ 2.5 [6]

ในปฏิญานินพจน์นี้ ผู้จัดทำใช้โมดูลระบุตำแหน่งพิกัด GPS GY-NEO6MV2 NEO-6M เชื่อมต่อกับ ESP32-WROOM เพื่อใช้ระบุตำแหน่งที่ต้องการให้หุ่นลอยน้ำให้อาหารปลาอัตโนมัติ เคลื่อนที่ไปตามตำแหน่งที่กำหนด



รูปที่ 2.5 โมดูล GPS GY-NEO6MV2 [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 ข้อมูลทางเทคนิคของโมดูลระบุตำแหน่งพิกัด GPS GY-NEO6MV2 NEO-6M

ข้อมูลทางเทคนิค	รายละเอียด
แรงดันไฟฟ้า	3.3 โวลต์ – 5 โวลต์ไฟตรง
Communication interface	UART TTL, 9600bps.
กระแสไฟฟ้าที่ใช้งาน	45 มิลลิแอม
TXD/RXD Impedance	510 โอห์ม
Navigation Sensitivity	-161 dBm
โพรโตคอลในการสื่อสาร	NMEA, UBX Binary, RTCM
อุณหภูมิการทำงาน	-40°C ~ 85°C
อัตราการอัปเดตข้อมูลในการนำทาง	1 เฮิร์ต (สูงสุด 5 เฮิร์ต)

2.5 โมดูลเข็มทิศ GY-273 ใช้ชิป HMC5883L

โมดูล GY-273 เป็นเซนเซอร์แม่เหล็กแบบหลายชิปที่มีสามแกน ใช้สำหรับวัดทิศทางของการหมุนเบา ๆ หรือการเคลื่อนไหวของอุปกรณ์อิเล็กทรอนิกส์ ออกแบบมาเพื่อใช้ในการประยุกต์ใช้งานที่ต้องการความแม่นยำสูง เช่น ในการใช้เป็นเครื่องวัดทิศทาง (compassing), การนำทาง (navigation), อุปกรณ์โดรน, หุ่นยนต์, โทรศัพท์มือถือ, และอุปกรณ์พกพาส่วนตัว

โมดูลเข็มทิศ GY-273 แสดงดังรูปที่ 2.6 สามารถให้ความแม่นยำในการหาทิศทางของเข็มทิศอยู่ในช่วง 1° ถึง 2° และมีการเชื่อมต่อผ่านบัสซีเรียล I2C (Inter-Integrated Circuit) ซึ่งเป็นส่วนหนึ่งของโปรโตคอลการสื่อสารระหว่างอุปกรณ์อิเล็กทรอนิกส์ ทำให้สามารถเชื่อมต่อได้สะดวก ข้อมูลทางเทคนิคของโมดูลเข็มทิศ GY-273 แสดงได้ดังตารางที่ 2.6 [10]

ในปริญญานิพนธ์นี้ ผู้จัดทำใช้โมดูลเข็มทิศ QMC5883L เพื่อใช้ระบุองศาที่ต้องการให้หุ่นยนต์นำอาหารปลาอัตโนมัติเคลื่อนที่ไปตามตำแหน่งที่กำหนด



รูปที่ 2.6 โมดูลเข็มทิศ GY-273 [11]

ตารางที่ 2.6 ข้อมูลทางเทคนิคของโมดูลเข็มทิศ GY-273

ข้อมูลทางเทคนิค	รายละเอียด
ADC (Analog-to-Digital Converter)	16 บิต ร่วมกับเซนเซอร์แม่เหล็ก Anisotropic Magneto-Resistive (AMR) ที่มีสัญญาณรบกวนต่ำ สามารถให้ความละเอียดในการวัดฟิลด์แม่เหล็กได้ถึง 2 มิลลิแกส (Milli-Gauss)
อุณหภูมิในการทำงาน	-40 °C ถึง +85 °C
ช่วงการวัดฟิลด์แม่เหล็ก	ระหว่าง ± 8 กัสส์ (Gauss) คือสามารถวัดแรงดันแม่เหล็กที่อยู่ในช่วงระหว่าง -8 กัสส์ ถึง +8 กัสส์ โดยรอบแกนแม่เหล็กทั้งหมดของเซนเซอร์ได้
อินเทอร์เฟซ I2C	ใช้งานในโหมดมาตรฐาน (Standard Mode) และโหมดเร็ว (Fast Mode)
แรงดันในการทำงาน	ระหว่าง 2.16 โวลต์ ถึง 3.6 โวลต์ และใช้พลังงาน 75 ไมโครแอมป์ (μA)

2.6 แผงพลังงานแสงอาทิตย์

แผงพลังงานแสงอาทิตย์หรือโซลาร์เซลล์ (Solar cell) หรือ เซลล์โฟโตวอลเทอิก (Photovoltaic cell) เป็นอุปกรณ์อิเล็กทรอนิกส์ที่ทำจากสารกึ่งตัวนำ ที่มีความสามารถในการแปลงพลังงานแสงอาทิตย์ที่ตกกระทบตัวสารกึ่งตัวนำเพื่อเปลี่ยนมาเป็นพลังงานไฟฟ้ากระแสตรง (DC current) โดยหลักการทำงานของแผงพลังงานแสงอาทิตย์ เริ่มต้นจากการที่แผงพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

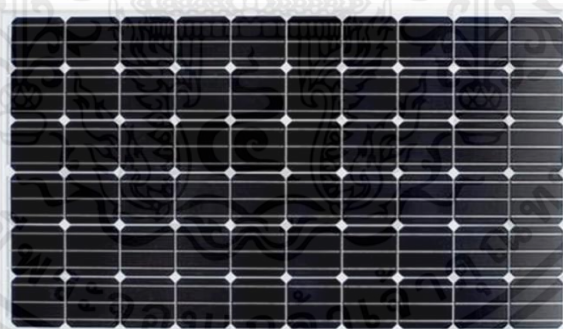
แสงอาทิตย์ได้รับพลังงานแสงอาทิตย์ในช่วงเวลากลางวัน ซึ่งเป็นคลื่นแม่เหล็กไฟฟ้าไปกระทบกับสารกึ่งตัวนำของแผงพลังงานแสงอาทิตย์ โดยพลังงานจากแสงอาทิตย์จะทำการถ่ายเทพลังงานให้กับอิเล็กตรอนเคลื่อนที่ไปรวมตัวกันที่ขั้วลบของแผงพลังงานแสงอาทิตย์และโฮลจะถูกเติมเต็มด้วยอิเล็กตรอนจากขั้วบวกของแผงพลังงานแสงอาทิตย์ ซึ่งจากกระบวนการนี้ทำให้เกิดกระแสไฟฟ้าและสามารถไปใช้งานได้

2.6.1 ประเภทของแผงพลังงานแสงอาทิตย์

แผงพลังงานแสงอาทิตย์ที่นิยมใช้กันอยู่ในปัจจุบันแบ่งเป็น 3 แบบคือ

2.6.1.1 โมโนคริสตัลไลน์

โมโนคริสตัลไลน์ (Mono Crystalline Silicon Solar Cell) ทำจากผลึกซิลิคอนเชิงเดี่ยวที่มีคุณภาพและมีความบริสุทธิ์สูง ลักษณะของเซลล์เป็นรูปสี่เหลี่ยมที่ลบมุมทั้งสี่ออก โดยนำมาวางเรียงต่อ ๆ กันเพื่อให้การใช้งานมีประสิทธิภาพสูงสุด มีสีเข้มที่ดูสวยงาม และสามารถผลิตกระแสไฟฟ้าได้สูงกว่า แผงพลังงานแสงอาทิตย์ชนิดอื่น แม้อยู่ในภาวะแสงแดดน้อยก็ตาม ส่วนอายุการใช้งานยาวนานถึง 25 ปีขึ้นไป ตัวอย่างของแผงพลังงานแสงอาทิตย์ชนิดโมโนคริสตัลไลน์แสดงดังรูปที่ 2.7

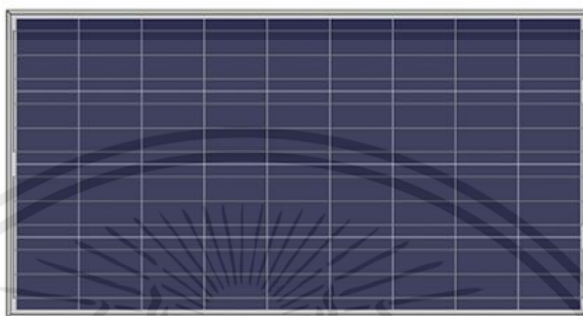


รูปที่ 2.7 แผงพลังงานแสงอาทิตย์ชนิดโมโนคริสตัลไลน์ [13]

2.6.1.2 พอลิคริสตัลไลน์

พอลิคริสตัลไลน์ (Poly Crystalline Silicon Solar Cell) ทำมาจากผลึกซิลิคอนเหมือนแผงพลังงานแสงอาทิตย์แบบโมโนคริสตัลไลน์ แต่มีขั้นตอนการผลิตแตกต่างกันใช้ปริมาณซิลิคอนในการผลิตน้อยกว่า แผงเป็นสีน้ำเงินแต่ไม่เข้มมาก มีประสิทธิภาพในการผลิตกระแสไฟฟ้าได้ดี แต่น้อยกว่าแผงแบบโมโนคริสตัลไลน์ (ยกเว้นการใช้งานในที่อุณหภูมิสูง)

แผงแบบพอลิคริสตัลไลน์จะผลิตกระแสไฟฟ้าได้ดีกว่าแผงแบบโมโนคริสตัลไลน์เล็กน้อย) จึงมีราคาถูกลงกว่า อายุการใช้งานประมาณ 20-25 ปี ตัวอย่างของแผงพลังงานแสงอาทิตย์ชนิดพอลิคริสตัลไลน์ แสดงดังรูปที่ 2.8



รูปที่ 2.8 แผงพลังงานแสงอาทิตย์ชนิดพอลิคริสตัลไลน์ [13]

2.6.1.3 แบบฟิล์มบาง

แผงพลังงานแสงอาทิตย์แบบฟิล์มบาง หรือ อะมอร์ฟัสโซลาร์เซลล์ (Amorphous Solar Cell) แผงพลังงานแสงอาทิตย์แบบฟิล์มบาง (Thin Film Solar Cell) เกิดจากการนำสารที่มีคุณสมบัติในการเปลี่ยนพลังงานแสงอาทิตย์ไปเป็นพลังงานไฟฟ้ามาฉาบเป็นชั้นฟิล์มบาง ๆ (ซ้อนกันหลาย ๆ ชั้น) บางคนจึงเรียกว่าแผงพลังงานแสงอาทิตย์ชนิดนี้ว่า ฟิล์มบาง เป็นแผงพลังงานแสงอาทิตย์ที่มีราคาถูกที่สุด การผลิตกระแสไฟฟ้า และอายุการใช้งานมีน้อยกว่าแผงชนิดอื่น ๆ ตัวอย่างของแผงพลังงานแสงอาทิตย์ชนิดแบบฟิล์มบาง แสดงดังรูปที่ 2.9



รูปที่ 2.9 แผงพลังงานแสงอาทิตย์ชนิดโซลาร์เซลล์แบบฟิล์มบาง [13]

2.6.2 คุณสมบัติและตัวแปรที่สำคัญของแผงพลังงานแสงอาทิตย์

ประสิทธิภาพการทำงานในแต่ละพื้นที่ที่มีความแตกต่างกัน ตลอดจนการนำไปคำนวณระบบหรือคำนวณจำนวนเซลล์แผงแสงอาทิตย์ที่ต้องใช้ในแต่ละพื้นที่ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ความเข้มของแสง

เมื่อความเข้มของแสงสูงกระแสที่ได้จากเซลล์แสงอาทิตย์ก็จะสูงขึ้น ในขณะที่แรงดันไฟฟ้าหรือโวลต์ แทบจะไม่แปรไปตามความเข้มของแสง ความเข้มของแสงที่ใช้วัดเป็นมาตรฐานคือ ความเข้มของแสงที่วัดบน พื้นโลกในสภาพอากาศปลอดโปร่ง ปราศจากเมฆหมอกและวัดที่ระดับน้ำทะเล ในสภาพที่แสงอาทิตย์ตั้งฉากกับ พื้นโลก ซึ่งความเข้มของแสงจะมีค่าเท่ากับ 100 mW ต่อ ตร.ซม. หรือ 1,000 W ต่อตารางเมตร ซึ่งมีค่า เท่ากับ AM 1.5 (Air Mass 1.5)

2. อุณหภูมิ

การเปลี่ยนแปลงของอุณหภูมิจะไม่ส่งผลต่อกระแสไฟฟ้า (Current) ในขณะที่แรงดันไฟฟ้า จะลดลงเมื่ออุณหภูมิสูงขึ้น ซึ่งโดยเฉลี่ยแล้วทุก ๆ 1 องศาที่เพิ่มขึ้น จะทำให้แรงดันไฟฟ้าลดลง 0.5% และใน กรณีของแผงเซลล์แสงอาทิตย์มาตรฐานที่ใช้กำหนดประสิทธิภาพของแผงแสงอาทิตย์คือ ณ อุณหภูมิ 25 °C

3. การปรับทิศทางในการรับแสง

ปริมาณพลังงานแสงอาทิตย์ (Insolation) ที่ตกกระทบบนแผงเซลล์แสงอาทิตย์ในรอบปี (Annual insolation) จะขึ้นอยู่กับตำแหน่งที่ตั้งทางภูมิศาสตร์ของพื้นที่มุมเอียงของแผงและทิศทางที่เปียงเบน จากทิศใต้ตั้งนั้น ในการติดตั้งแผงเซลล์แสง อาทิตย์ ในประเทศไทยควรหันไปทางทิศใต้ทำมุมเอียง 10-45 องศา กับแนวนอน และควรห่างจากพื้นราบ หรือพื้นหลังคา ไม่น้อย 10 เซนติเมตร เพื่อช่วยระบาย ความร้อน รวมถึงยืดระยะเวลาการใช้งาน และเป็นการป้องกันการลัดวงจรอันเนื่องมาจากความร้อน

4. ประสิทธิภาพของระบบ (Electrical conversion efficiency)

ระบบเซลล์แสงอาทิตย์ส่วนใหญ่ประกอบไปด้วยชุดแผงเซลล์แสงอาทิตย์อุปกรณ์ต่อเชื่อมและเครื่องแปลงกระแส ดังนั้น ประสิทธิภาพของระบบจึงขึ้นอยู่กับคุณภาพของอุปกรณ์

2.6.3 ขั้นตอนการทำงานของแผงพลังงานแสงอาทิตย์เพื่อนำพลังงานไปใช้

1. แผงพลังงานแสงอาทิตย์รับแสงแดด เปลี่ยนเป็นพลังงานไฟฟ้า
2. ปรับแรงดันไฟฟ้าที่ได้ให้เป็น 12V ด้วยเครื่องควบคุมประจุ (Solar Charge Controller) เพื่อควบคุมแรงดันไฟฟ้าให้คงที่ กระแสไฟฟ้าที่ได้เป็นกระแสตรง (DC)
3. เก็บสำรองพลังงานไฟฟ้าไว้ในแบตเตอรี่

2.6.4 การคำนวณระบบแผงพลังงานแสงอาทิตย์

$$\text{ขนาดของแผง} = \frac{\text{ค่าการใช้พลังงานรวมทั้งหมด}}{5 \text{ ชั่วโมง}} \quad (2.2)$$

5 ชั่วโมงคือปริมาณแสงอาทิตย์ที่นำจะได้ใน 1 วัน

ในปฏิญานิพนธ์นี้ ผู้จัดทำใช้แผงพลังงานแสงอาทิตย์ชนิดพอลิคริสตัลไลน์ รุ่น 50 วัตต์ เพื่อนำพลังงานไฟฟ้าที่ได้จากพลังงานแสงอาทิตย์ไปเก็บไว้ในแบตเตอรี่ ใช้เป็นแหล่งจ่ายพลังงานให้แก่หุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยตารางที่ 2.7 แสดงข้อมูลทางเทคนิคของแผงพลังงานแสงอาทิตย์ [12]-[13]

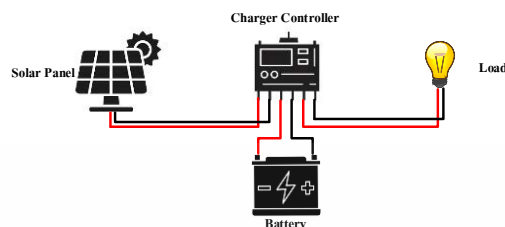
ตารางที่ 2.7 ข้อมูลทางเทคนิคของแผงพลังงานแสงอาทิตย์

ข้อมูลทางเทคนิค	รายละเอียด
กำลังไฟฟ้าสูงสุด	50Wp (0/+5%)
ค่าแรงดันไฟเปิดวงจร	21.6 โวลต์
ค่ากระแสลัดวงจร	3.23 แอมแปร์
ค่าแรงดันที่ กำลังไฟฟ้าสูงสุด	17.2 โวลต์
ค่ากระแสที่ กำลังไฟฟ้าสูงสุด	2.91 แอมแปร์
ค่าแรงดันระบบสูงสุด	1000 โวลต์ไฟตรง
อุณหภูมิในการทำงาน	-40~+85°C
ขนาด	675*534*28 มิลลิเมตร

2.7 โซลาร์ชาร์จคอนโทรลเลอร์

เป็นอุปกรณ์อิเล็กทรอนิกส์ตัวหนึ่งที่มีคุณสมบัติคอยควบคุมการชาร์จไฟฟ้าจากแผงพลังงานแสงอาทิตย์เข้าสู่แบตเตอรี่ ของระบบของแผงพลังงานแสงอาทิตย์ทำการเก็บกระแสไฟเพื่อนำมาใช้งานตามที่เรากำหนดไว้ จะมีหลักการทำงานหรือหน้าที่ จ่ายกระแสไฟเมื่อแรงดันแบตเตอรี่อยู่ในระดับต่ำตามที่ตั้งค่า และทำการตัดการจ่ายกระแสไฟเพื่อไปประจุยังแบตเตอรี่เมื่อแรงดันของแบตเตอรี่อยู่ในระดับที่สูงตามที่ได้กำหนดไว้ เพื่อป้องกันการ Over Charge ซึ่งจะทำให้แบตเตอรี่เกิดความเสียหายและเสื่อมอายุ รูปที่ 2.10 แสดงตัวอย่างการทำงานของโซลาร์

ชาร์จคอนโทรลเลอร์ ตัวอย่างโซลาร์ชาร์จคอนโทรลเลอร์แสดงดังรูปที่ 2.11 และคุณสมบัติของโซลาร์ชาร์จคอนโทรลเลอร์แสดงได้ดังตารางที่ 2.8 [14]



รูปที่ 2.10 การทำงานของโซลาร์ชาร์จคอนโทรลเลอร์



รูปที่ 2.11 โซลาร์ชาร์จคอนโทรลเลอร์[14]

ตารางที่ 2.8 คุณสมบัติของโซลาร์ชาร์จคอนโทรลเลอร์

ข้อมูลทางเทคนิค	รายละเอียด
การควบคุมสัญญาณ	Pulse Width Modulation (PWM)
แรงดันที่ใช้งาน	12 โวลต์ / 14 โวลต์
กระแสไฟที่ใช้งาน	30 แอมป์
กระแสการชาร์จแบตเตอรี่ / กระแสโหลด	สูงสุด 30 แอมป์ / สูงสุด 10 แอมป์
กระแสการชาร์จแบตเตอรี่ / กระแสโหลด	สูงสุด 30 แอมป์ / สูงสุด 10 แอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 คุณสมบัติของโซลาร์ชาร์จคอนโทรลเลอร์ (ต่อ)

ข้อมูลทางเทคนิค	รายละเอียด
ช่วงแรงดันที่รองรับ	<ul style="list-style-type: none"> • ≤ 23 โวลต์สำหรับแบตเตอรี่ 12 โวลต์ช่วงของแรงดันการทำงานสูงสุด • ≤ 46 โวลต์สำหรับแบตเตอรี่ 24 โวลต์ช่วงของแรงดันการทำงานสูงสุด
ขนาด	134 * 70 * 30 มิลลิเมตร
คุณสมบัติหลัก	<ul style="list-style-type: none"> • 2 พอร์ต USB สำหรับการจ่ายไฟ: 5 โวลต์ / 2 แอมป์ • จอ Liquid Crystal Display (LCD)

2.8 แบตเตอรี่ลิเธียม

2.8.1 ชนิดของแบตเตอรี่

แบตเตอรี่ลิเธียมแบ่งประเภทออกเป็น 2 แบบ คือ

2.8.1.1 แบตเตอรี่ชนิดลิเธียม-ไอออน

แบตเตอรี่ชนิดลิเธียม-ไอออน (Lithium Ion Batteries) ดังรูปที่ 2.12 มีความหนาแน่นพลังงานสูง ไม่จำเป็นต้องกระตุ้นก่อนที่จะใช้หลังจากเก็บมาเป็นเวลานาน มีอัตราการคายประจุต่ำ และไม่จำเป็นต้องดูแลรักษามาก มีข้อจำกัดคือจำเป็นต้องใช้วงจรป้องกันแรงดันและกระแสให้อยู่ในเขตปลอดภัย มีการเสื่อมอายุตามเวลาแม้ว่าจะไม่มีการใช้งาน และมีอัตราการจ่ายกระแสไม่สูง ไม่เหมาะสมกับงานที่ใช้โหลดหนักๆ



รูปที่ 2.12 แบตเตอรี่ชนิดลิเธียม-ไอออน [15]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.1.2 แบตเตอรี่ชนิดลิเทียมโพลีเมอร์

แบตเตอรี่ลิเทียมโพลีเมอร์ (Lithium Polymer Batteries) ดังรูปที่ 2.13 ชนิดของสาร ลิเทียมโพลีเมอร์ ใช้ฟิล์มคล้ายพลาสติกเจล แทนการใช้แผ่นเมมเบรนที่มีรูพรุน เป็นตัวส่งผ่านลิเทียมโพลีเมอร์ เป็นการผลิตที่ง่าย สามารถทำให้บางและไม่จำกัดรูปแบบ สามารถทำได้หลายขนาด มีน้ำหนักเบา ไม่ใช้ตัวถังโลหะ และมีความปลอดภัย รับการชาร์จไฟเกินได้มากกว่า โอกาสของการรั่วของอิเล็กโทรไลต์ลดลง ซีตจำกัดของแบตเตอรี่ชนิดลิเทียมโพลีเมอร์ มีความหนาแน่นของพลังงานต่ำกว่า และจำนวนรอบการใช้งานที่ต่ำกว่าแบตเตอรี่ชนิดลิเทียม [15]



รูปที่ 2.13 แบตเตอรี่ชนิดลิเทียมโพลีเมอร์ [15]

2.8.2 หลักการทำงานของแบตเตอรี่ลิเทียม

แบตเตอรี่ลิเทียม ทำงานโดยอาศัยหลักการทางไฟฟ้าเคมี โดยพลังงานที่ประจุไฟเข้าไป จะเข้าไปทำปฏิกิริยาเคมีภายในแบตเตอรี่ ทำให้เกิดการบังคับให้ลิเทียมไอออนไหลออก โดยไหลผ่านอิเล็กโทรไลต์ หลังจากนั้นจะผ่าน Separator และเข้าไปสอดตัวอยู่ในโครงสร้างของวัสดุที่ใช้ทำขั้วแอโนด ผลของปฏิกิริยาจะทำให้วัสดุที่ใช้ทำขั้วแคโทด และแอโนดอยู่ในสภาวะที่ไม่เสถียร และในขณะที่ใช้งานหรือการคายประจุ ปฏิกิริยาเคมีของแบตเตอรี่จะสามารถเกิดขึ้นได้เอง และจะทำให้ระบบเสถียรอีกครั้ง [16]

ในปริญญานิพนธ์นี้ ผู้จัดทำใช้แบตเตอรี่ชนิด แบตเตอรี่ชนิดลิเทียม-ไอออน ขนาด 12 โวลต์ 20 แอมป์ เป็นตัวเก็บพลังงานไฟฟ้าที่ได้จากแผงพลังงานแสงอาทิตย์และจ่ายพลังงานไฟฟ้าให้แก่อุปกรณ์

2.8.3 ข้อดีและข้อควรระวังการใช้แบตเตอรี่ลิเธียมไอออน

ตารางที่ 2.9 ข้อดีและข้อควรระวังของแบตเตอรี่ลิเธียม

ข้อดี	ข้อควรระวัง
สามารถใช้งานได้นานกว่าแบตเตอรี่แบบเดิม	ไม่ควรให้แบตเตอรี่ลิเธียมโดนน้ำ
ให้กำลังสม่ำเสมอ ให้กำลังเท่าเดิมตั้งแต่เริ่มใช้งานจนทำงานเสร็จ	ควรชาร์จแบตเตอรี่ทุกๆ 3 เดือน
สามารถชาร์จแบตเตอรี่ลิเธียม ได้ตลอดเวลาโดยที่ไม่มีผลเสียต่อความสามารถในการเก็บประจุไฟฟ้า	โปรดใช้ที่ชาร์จแบบพิเศษสำหรับแบตเตอรี่ลิเธียม
อัตราการคายประจุน้อยกว่าแบตเตอรี่นิกเกิล-แคดเมียม	ห้ามนำขั้วบวกขั้วลบมาช็อตกันโดยตรง
มีน้ำหนักเบา	อย่าปล่อยให้กระแสเกินที่กำหนดเมื่อใช้งานแบตเตอรี่

2.8.4 สูตรคำนวณ

1. หาปริมาณความจุที่ใช้ไป

$$A \cdot h \cdot 1000 = \text{mAh} \quad (2.3)$$

2. หาเวลาที่เหลือที่แบตเตอรี่จ่ายได้

$$\text{mAh} / \text{Amps} \cdot 1000 = \text{hours} \quad (2.4)$$

3. หาเวลาที่เหลือที่แบตเตอรี่จ่ายได้

$$(\text{mAh} \cdot \text{Volts}) / (\text{Watts} \cdot 1000) = \text{hours} \quad (2.5)$$

2.9 IRF540

เป็น MOSFET หรือ Metal Oxide Semiconductor Field Effect Transistor ออกแบบมาเพื่อลดขนาดความจุของอินพุตและกระแสเกต เพื่อให้เหมาะสำหรับการใช้เป็นสวิตช์ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

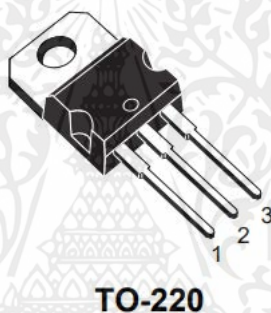
ตัวแปลงไฟฟ้ากระแสตรง มีประสิทธิภาพสูงและทำงานในความถี่สูง เหมาะสำหรับการใช้งานในระบบโทรคมนาคมและคอมพิวเตอร์ แอปพลิเคชันทุกรูปแบบที่ต้องการการขับเคลื่อนขา Gate ต่ำ

2.9.1 โครงสร้าง IRF 540

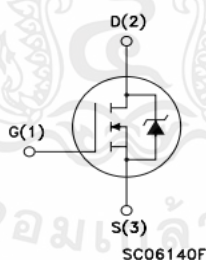
IRF 540 ประกอบด้วยสามส่วน ดังรูปที่ 2.14 และสัญลักษณ์แสดงดังรูปที่ 2.15

ส่วนประกอบของ IRF เป็นดังนี้

1. ขา G (Gate) เป็นขาที่ต่อกับแรงดันไฟฟ้าขาเข้า
2. ขา D (Drain) เป็นขาที่เชื่อมต่อกับแรงดันไฟฟ้าและโหลด
3. ขา S (Source) เป็นขาที่ต่อกับโฟลบ [17]



รูปที่ 2.14 ส่วนประกอบของ IRF 540 [17]



รูปที่ 2.15 สัญลักษณ์ของ IRF 540 [17]

2.9.2 การทำงานของ IRF 540

ขั้นตอนการทำงานของ IRF 540 ที่ทำหน้าที่เป็นสวิตช์

1. ตั้งแต่สถานะปิด (Cut-off) เมื่อมีแรงดันเกตที่ต่ำ (VIN), MOSFET จะอยู่ในสถานะปิด ซึ่งหมายความว่าไม่มีกระแสไฟฟ้าไหลผ่านทางช่องทาง (ไม่ conducting) ทำให้ MOSFET เข้าสู่

สถานะทำงาน, จะต้องมีการป้อนแรงดันเกตที่เพียงพอที่จะเปิดช่องทางไฟฟ้า (Channel). เมื่อมีแรงดันที่สูงขึ้นที่เกต (VIN), ช่องนำไฟฟ้าเปิดและ MOSFET จะเริ่ม conducting.

2. สถานะทำงาน (Saturation) เมื่อ MOSFET เปิดและมีกระแสไฟฟ้าไหลผ่าน จะอยู่ในสถานะทำงาน (saturation) นอกจากนี้ MOSFET มักจะมีค่าความต้านทานต่ำในช่องทางทำให้สามารถส่งผ่านกระแสไฟฟ้าได้ดี และเพื่อให้ MOSFET ปิด แรงดันเกตต้องถูกลดลง เมื่อมีการลดแรงดันที่เกต (VIN) ช่องนำไฟฟ้าปิดและ MOSFET จะไม่ทำงาน (non-conducting) [18]

2.9.3 การคำนวณหากระแสที่ไหลผ่านโหลด

$$I = \frac{P}{V} \quad (2.6)$$

I คือ กระแสไฟฟ้า

P คือ กำลังไฟฟ้า

V คือ แรงดันไฟฟ้า

ในปฏิยานิพนธ์นี้ ผู้จัดทำใช้ IRF 540 นำมาใช้เป็นสวิตซ์ในการควบคุมการเปิด-ปิด โหลดโดยแสดงข้อมูลทางเทคนิคดังตารางที่ 2.10

ตารางที่ 2.10 ข้อมูลทางเทคนิคของ IRF 540

ข้อมูลทางเทคนิค	รายละเอียด
แรงดัน Vds	100 โวลต์
กระแสไฟฟ้ตรง	22 แอมป์
ค่าความต้านทานในช่องทางตอนเปิด (RDS)	น้อยกว่า 0.077 โอห์ม

2.10 LM2596

LM2596 เป็นวงจรรวมที่ออกแบบมาเพื่อให้เหมาะสมสำหรับการออกแบบเครื่องแปลงไฟแบบลดระดับ (buck converter) ง่ายและสะดวกต่อการใช้งาน สามารถขับโหลดได้สูงสุดถึง 3.0 แอมป์ สามารถควบคุมและควบคุมโหลดที่ดี สามารถปรับได้และมีการคอมเพนเซตภายในเพื่อลดจำนวนของอุปกรณ์ภายนอกเพื่อให้การออกแบบเครื่องจ่ายไฟง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ LM2596 เป็นแหล่งจ่ายไฟแบบสวิตช์โหมด (switch-mode power supply) ความหน่วงเวลาของมันมีประสิทธิภาพสูงมากเมื่อเทียบกับรีเลย์กลไกเตอร์เชิงเส้นสายสามขาที่ได้รับ ความนิยม เมื่อมีแรงดันนำเข้าไปสูงขึ้น LM2596 ทำงานที่ความถี่สวิตช์ที่ 150 กิโลเฮิร์ต ทำให้สามารถใช้คอมโพเนนต์ตัวกรองขนาดเล็กกว่าที่จะต้องใช้กับรีเลย์กลไกเตอร์แบบที่มีความถี่ต่ำ มีให้เลือกในแพ็คเกจมาตรฐาน TO-220 โดยรูปที่ 2.16 แสดงตัวอย่างของ LM2596 และข้อมูลทางเทคนิคของ LM2596 แสดงได้ในตารางที่ 2.11 [19]–[20]



รูปที่ 2.16 LM2596 [21]

ตารางที่ 2.11 ข้อมูลทางเทคนิคของ LM2596

ข้อมูลทางเทคนิค	รายละเอียด
แรงดันเอาต์พุต	3.3 โวลต์, 5 โวลต์, 12 โวลต์ และรุ่นที่สามารถปรับได้
วงของแรงดันเอาต์พุตในรุ่นที่สามารถปรับได้	1.2 โวลต์ ถึง 37 โวลต์สูงสุดคลาดเคลื่อนไม่เกิน +4% ภายใต้เงื่อนไขของแรงดันนำเข้าไปและโหลด
Output Load Current	3 แอมป์
แรงดันนำเข้าไปสูงสุด	40 โวลต์
การป้องกันด้วยการตัดการทำงานเมื่อมีอุณหภูมิสูงเกินไปและการจำกัดกระแส	

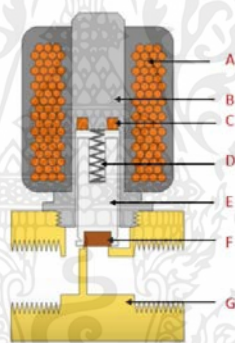
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 โซลินอยด์วาล์ว

โซลินอยด์วาล์ว คือ วาล์วควบคุมการเปิด-ปิดตัวกลางที่เป็นของเหลวและก๊าซ ด้วยไฟฟ้ามีทั้งชนิด $1/2$, $2/2$, $3/2$, $4/2$, $5/2$ และ $5/3$ ซึ่งตัวเลขหน้าบอกถึงจำนวนทางเข้าออกของวาล์ว ส่วนตัวเลขที่ตามหลังเครื่องหมายทับ (/) นั้นบอกถึงจำนวนสถานะ หรือ จำนวนตำแหน่ง (position) ของวาล์ว เช่น วาล์ว $2/2$ ก็คือ วาล์วที่มี 2 ทาง และมี 2 สถานะ คือ ปิด และ เปิด [22]

2.11.1 การทำงานของโซลินอยด์วาล์ว

โซลินอยด์วาล์วที่ทำหน้าที่โดยมีหลักการทำงานที่ง่าย สำหรับวาล์วปิดปกติโดยไม่มีกำลังลูกสูบ (E) จะบล็อกปากด้วยซีลวาล์ว (F) สปริง (D) กำลังบังคับให้ปิดนี้ เมื่อพลังงานถูกนำไปใช้กับขดลวด (A) จะสร้างสนามแม่เหล็กไฟฟ้าดึงดูดลูกสูบขึ้นเอาชนะแรงสปริง สิ่งนี้จะเปิดปากและช่วยให้สื่อไหลผ่านได้ วาล์วเปิดปกติมีส่วนประกอบเดียวกัน แต่ทำงานในลักษณะตรงกันข้ามแสดงดังรูปที่ 2.17



รูปที่ 2.17 ส่วนประกอบของโซลินอยด์ [23]

2.11.2 ประเภทโซลินอยด์วาล์ว

โซลินอยด์วาล์วมี 3 ประเภท

2.11.2.1 Normally closed Solenoid Valve

สำหรับโซลินอยด์วาล์วที่ปิดตามปกติวาล์วจะถูกปิดเมื่อไม่มีพลังงานและน้ำหรือแก๊สไม่สามารถไหลผ่านได้ เมื่อกระแสไฟฟ้าถูกส่งไปยังขดลวดมันจะสร้างสนามแม่เหล็กไฟฟ้าที่บังคับให้ลูกสูบขึ้นไปเอาชนะแรงสปริง คลายซีลและเปิดปากเพื่อให้สื่อไหลผ่านวาล์ว

โซลินอยด์วาล์วแบบเปิดตามปกติเหมาะสำหรับการใช้งานที่ต้องเปิดวาล์วเป็นเวลานานเนื่องจากประหยัดพลังงานมากขึ้น

2.11.2.3 Bi-stable Solenoid Valve

โซลินอยด์วาล์วแบบ Bi-stable หรือล๊อคสามารถเปลี่ยนได้ด้วยแหล่งจ่ายไฟชั่วขณะ จากนั้นจะอยู่ในตำแหน่งนั้น ดังนั้นจึงไม่เปิดหรือปิดตามปกติเนื่องจากอยู่ในตำแหน่งปัจจุบันเมื่อไม่มีการใช้พลังงาน [20]

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้โซลินอยด์วาล์วชนิด 1/2 แบบ Normally closed Solenoid Valve โดยถูกควบคุมการทำงานจาก MOSFET

ตารางที่ 2.12 ข้อมูลทางเทคนิคของโซลินอยด์วาล์ว

ข้อมูลทางเทคนิค	รายละเอียด
แรงดันไฟฟ้า	12 โวลต์ไฟตรง
กระแสไฟฟ้าที่ใช้	0.5 แอมป์
ขนาด	4 ทุน
ความดัน	0.02 – 0.80 เมกะปาสกาล

2.12 DC Pump

ปั๊มน้ำที่ใช้ไฟฟ้ากระแสตรง (DC) คือ ขั้วบวกต่อกับขั้วบวกและขั้วลบต่อกับขั้วลบ มาต่อเข้ากับระบบแผงพลังงานแสงอาทิตย์หรือเป็นแหล่งจ่ายไฟที่มาจากแบตเตอรี่ สามารถใช้งานได้หลายประเภท เช่น สูบน้ำทั่วไป สูบน้ำมัน ล้างรถ อัดฉีดรถ เป็นต้น

2.12.1 ประเภท DC Pump

2.12.1.1 ปั๊มแรงดันน้ำ

ปั๊มที่เพิ่มแรงดันน้ำปลายทาง เช่น ล้างรถ พ่นน้ำ ซึ่งควรต่อพ่วงกับแบตเตอรี่เพื่อรักษาระดับแรงดันไฟฟ้า

2.12.1.2 ปั๊มน้ำตีซีแบบจุ่ม

ต้องจุ่มตัวปั๊มลงไปใต้น้ำเพื่อจะดูดน้ำไปยังอีกที่หนึ่ง โดยใช้กับปริมาณน้ำและความสูงไม่มากนัก ควรเลือกให้มีแรงดันไฟฟ้ากระแสตรงให้ตรงกับระบบที่ออกแบบไว้ และควรต่อไฟฟ้ากระแสตรงฝั่งที่ออกจากแบตเตอรี่เพื่อจะได้มีแรงดันที่คงที่ในการใช้งาน อีกทั้งยังช่วยป้องกันความเสียหายที่จะเกิดขึ้นกับตัวปั๊มด้วย โดยรูปที่ 2.18 แสดงตัวอย่างของปั๊มน้ำตีซีแบบจุ่ม



รูปที่ 2.18 ปั้มน้ำดีซีแบบจุ่ม [25]

2.12.1.3 ปั้มน้ำดีซีบาดาล

จุ่มตัวปั้มลงไปใต้น้ำเพื่อจะดูดน้ำไปยังอีกที่หนึ่งเช่นเดียวกับปั้มดีซีแบบจุ่ม แต่ตัวนี้สามารถต่อวงจรตัวปั้มโดยตรงกับตัวแผงพลังงานแสงอาทิตย์ได้และไม่ทำให้ตัวปั้มชนิดนี้เสียหาย ส่วนการที่แผงพลังงานแสงอาทิตย์จะผลิตแรงดันได้มากหรือน้อยขึ้นอยู่กับความเข้มแสงที่ตกกระทบแผง ตัวปั้มจะเปลี่ยนอัตราการหมุนของมอเตอร์เมื่อนำไฟฟ้ากระแสตรงที่ผลิตไปจ่ายปั้มบาดาล ทำให้ไม่ต้องใช้แบตเตอรี่ต่อระบบหรือหากต้องการให้มีประสิทธิภาพมากขึ้นก็สามารถพ่วงต่อแบตเตอรี่ได้ ซึ่งปั้มชนิดนี้ส่วนใหญ่จะใช้สูบบ่อขนาดลึก [24]–[25]

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้ DC Pump แบบ ปั้มน้ำดีซีแบบจุ่ม โดยใช้แรงดันกระแสไฟฟ้า 12 โวลต์

ตารางที่ 2.13 ข้อมูลทางเทคนิคของ Dc Pump

ข้อมูลทางเทคนิค	รายละเอียด
แรงดันไฟฟ้า	12 โวลต์
กำลังไฟฟ้า	180 วัตต์
ระยะดูดสูงสุด	6 – 8 เมตร
อัตราการไหลสูงสุด	4 ลูกบาศก์เมตรต่อชั่วโมง
น้ำหนัก	2.65 กิโลกรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 มอเตอร์ไฟฟ้ากระแสตรง

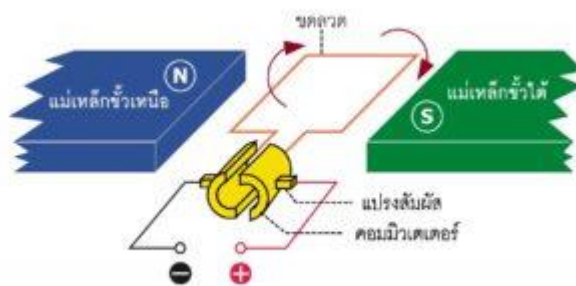
เป็นอุปกรณ์ที่ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล โดยเมื่อจ่ายไฟให้แก่มอเตอร์ จะทำให้แกนของมอเตอร์หมุน จึงสามารถนำการหมุนของแกนมอเตอร์ไปใช้ในการขับเคลื่อนวัตถุให้เกิดการเคลื่อนที่ โดยปกติมอเตอร์ไฟฟ้ากระแสตรงจะถูกสร้างขึ้นให้สามารถหมุนแกนด้วยความเร็วสูงมาก ตั้งแต่ 1,000 รอบขึ้นไป แต่แรงบิดที่ความเร็วรอบสูงมีน้อยมาก จนไม่สามารถนำไปขับเคลื่อนกลไกเคลื่อนที่ไหวได้ จึงต้องมีการทดจำนวนรอบด้วยการใช้เฟือง เพื่อให้เกิดแรงบิดมากขึ้น นั่นคือ ยิ่งมีอัตราทดสูงเท่าใด ความเร็วรอบของแกนมอเตอร์จะลดลง แต่จะมีแรงบิดมากขึ้น ดังนั้นการกำหนดอัตราทดที่เหมาะสมจะทำให้สามารถใช้งานมอเตอร์ไฟตรงเพื่อขับเคลื่อนกลไกเคลื่อนที่ได้มีประสิทธิภาพ รูปที่ 2.19 แสดงตัวอย่างของมอเตอร์ไฟฟ้ากระแสตรง



รูปที่ 2.19 มอเตอร์ไฟฟ้ากระแสตรง [26]

2.13.1 การทำงานของมอเตอร์ไฟฟ้ากระแสตรง

การขับหรือทำให้มอเตอร์ไฟฟ้ากระแสตรงทำงานเพื่อหมุนแกน ทำได้โดยจ่ายไฟเข้าที่ขั้วของมอเตอร์เท่านั้น และเมื่อกลับขั้วของการจ่ายไฟมอเตอร์ก็จะหมุนกลับทิศทาง รูปที่ 2./จอธิบายการทำงานของมอเตอร์แบบ 2 ขั้ว เมื่อจ่ายไฟให้แก่มอเตอร์ผ่านทางแปรงสัมผัสซึ่งต่ออยู่กับคอมมิวเตเตอร์และขดลวด เมื่อกระแสไฟฟ้าไหลผ่านขดลวดทำให้เกิดสนามแม่เหล็กขึ้น และเกิดแรงดูดจากแม่เหล็กถาวร ทำให้ขดลวดสามารถหมุนได้



รูปที่ 2.20 การทำงานของมอเตอร์ไฟฟ้ากระแสตรง [26]

2.13.2 ควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง

ในการขับมอเตอร์โดยปกติจะป้อนแรงดันไฟตรงให้โดยตรง มอเตอร์จะทำงานเต็มกำลัง ซึ่งอาจมีความเร็วมากเกินไป ดังนั้นการปรับความเร็วของมอเตอร์จึงใช้วิธีลดแรงดันไฟฟ้าที่ป้อนให้กับมอเตอร์ วิธีที่นิยมคือ การป้อนพัลส์ไปขับมอเตอร์แทน แล้วปรับความกว้างพัลส์ช่วงบวก เพื่อให้ได้ค่าแรงดันเฉลี่ยตามต้องการ วิธีการนี้เรียกว่า พัลส์วิดท์มอดูเลเตอร์ (PWM)

โดยความกว้างพัลส์ช่วงบวกเมื่อเทียบกับความกว้างพัลส์ทั้งหมดเรียกว่า ดิวตี้ไซเคิล (duty cycle) โดยจะคิดค่าดิวตี้ไซเคิลเป็นเปอร์เซ็นต์ของค่าความกว้างพัลส์ทั้งหมด [27]

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้มอเตอร์ขนาด 12 โวลต์ 5000r/min

ตารางที่ 2.14 ข้อมูลทางเทคนิคของมอเตอร์ไฟฟ้ากระแสตรง

ข้อมูลทางเทคนิค	รายละเอียด
แรงดันไฟฟ้า	12 โวลต์
ความเร็วสูงสุด	5000 รอบต่อนาที
กระแสไฟฟ้าที่ใช้งาน	0.175 แอมป์

2.14 โปรแกรมที่ใช้ในปฏิญานิพนธ์

2.14.1 Arduino IDE

Arduino IDE ส่วน IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น การใช้งานลักษณะ Open source ซึ่ง Arduino IDE จะทำหน้าที่ ติดต่อระหว่างคอมพิวเตอร์ ไม่ว่าจะเป็นระบบ Windows, Mac OS X หรือ Linux กับ บอร์ด Arduino ซึ่งโปรแกรมนี้ออกแบบให้ง่ายต่อการเขียนโค้ดและอัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino ส่วนในการเขียนโปรแกรมและคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชันเพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR หรือเวอร์ชันอื่นๆ ของ Arduino โดยรูปที่ 2.21 แสดงตัวอย่างของสัญลักษณ์ Arduino IDE [28]



รูปที่ 2.21 สัญลักษณ์ Arduino IDE

2.14.2 Proteus

Proteus Professional 8 เป็นโปรแกรมสำหรับการจำลองวงจรและการออกแบบ PCB เป็นเครื่องมือที่มีประโยชน์มากสำหรับโปรแกรมเมอร์ที่ใช้ไมโครคอนโทรลเลอร์ในการประยุกต์ใช้งาน ทำให้ไม่ต้องต่อวงจรจริงเพื่อพิสูจน์ว่าโปรแกรมที่เขียนขึ้นใช้งานได้หรือไม่ เพียงแต่ใช้ Proteus ต่่วงจรเพื่อจำลองพฤติกรรม (Simulator) ก็สามารถรู้ผลได้ทันที Proteus เป็นซอฟต์แวร์ Computer Aided Design (CAD) ที่มีประสิทธิภาพซึ่งออกแบบมาโดยเฉพาะเพื่อช่วยให้นักออกแบบสามารถสร้างและตรวจสอบแผงวงจรได้อย่างง่ายดาย ด้วยการใช้ซอฟต์แวร์นี้คุณสามารถสร้างการออกแบบวงจรไฟฟ้าที่มีความซับซ้อนสูง ช่วยให้คุณสามารถออกแบบ PCB ซึ่งมีส่วนประกอบทางไฟฟ้าที่มีรายละเอียดมาก เช่น ไมโครคอนโทรลเลอร์, ไมโครโปรเซสเซอร์, และส่วนประกอบอื่นๆ สัญลักษณ์ของ Proteus แสดงดังรูปที่ 2.22



รูปที่ 2.22 สัญลักษณ์ของ Proteus [29]

2.14.2.1 คุณสมบัติของโปรแกรม Proteus Professional

- สามารถสร้างวงจรไฟฟ้าแบบ Schematic ได้อย่างง่ายดาย
- สามารถสร้าง PCB Layout ได้อย่างละเอียด
- สามารถจำลองการทำงานของวงจรไฟฟ้าได้อย่างเสมือนจริง
- มีอุปกรณ์ให้เลือกใช้งานมากมาย
- รองรับการทำงานแบบ 3 มิติ
- รองรับการทำงานร่วมกับ Arduino

2.14.2.2 การใช้งานโปรแกรม Proteus Professional

โปรแกรม Proteus Professional แบ่งออกเป็น 2 ส่วนหลักๆ

1. ส่วนการออกแบบวงจรไฟฟ้า (Schematic Design)

ในส่วนการออกแบบวงจรไฟฟ้า ผู้ใช้สามารถออกแบบวงจรไฟฟ้าได้ตามต้องการ โดยโปรแกรมมีอุปกรณ์ให้เลือกใช้งานมากมาย เช่น ทรานซิสเตอร์ ไดโอด ไอซี ฯลฯ เมื่อออกแบบวงจรไฟฟ้าเสร็จแล้ว ผู้ใช้สามารถจำลองการทำงานของวงจรไฟฟ้าได้

2. ส่วนการออกแบบ PCB (PCB Layout)

ในส่วนการออกแบบ PCB ผู้ใช้สามารถออกแบบ PCB Layout ได้ตามต้องการ โดยโปรแกรมมีเครื่องมือช่วยออกแบบ PCB Layout มากมาย เช่น เครื่องมือวาง

อุปกรณ์ เครื่องมือลากสาย ฯลฯ เมื่อออกแบบ PCB Layout เสร็จแล้ว ผู้ใช้สามารถส่งออกไฟล์ PCB เพื่อนำไปผลิตเป็น PCB จริง [29]

2.14.3 AutoCAD

เป็นซอฟต์แวร์คอมพิวเตอร์ช่วยพัฒนาโดย Autodesk ที่ช่วยให้ การเขียนแบบได้ รวดเร็วและมีประสิทธิภาพมากยิ่งขึ้น ช่างเขียนแบบ สถาปนิก วิศวกร และผู้เชี่ยวชาญด้านอื่น ๆ มักจะใช้โปรแกรมเขียนแบบ CAD เพื่อสร้างงานสองมิติ (2D) และสามมิติ (3D) ช่วยให้มองเห็นการ ออกแบบและแนวความคิดในการออกแบบ นักออกแบบสามารถปรับเปลี่ยนแนวความคิดในการ ออกแบบได้โดยใช้เครื่องมือออกแบบ 3 มิติ (3D) แสดงผลและแปลงเป็นภาพเคลื่อนไหวได้อย่างน่า ทึ่ง รวมไปถึงการทำแบบเพื่อยื่นขออนุญาตก่อสร้าง สัญลักษณ์ของ Proteus แสดงดังรูปที่ 2.23

AutoCAD ทำงานร่วมกับไฟล์งานหลายแบบ ทำงานร่วมกับรูปแบบไฟล์หลายรูปแบบ ตัวหลักคือ DWG และ DWT [29]



รูปที่ 2.23 สัญลักษณ์ของ AutoCAD [30]

2.14.4 MIT APP Inverter

MIT App Inventor เป็นเครื่องมือที่ใช้สร้างแอปพลิเคชันที่ทำงานบน ระบบปฏิบัติการแอนดรอยด์ (Android) โดย MIT App Inventor ใช้หลักการพัฒนาซอฟต์แวร์เชิง คอมโพเนนต์ (Component-based Software Development) ทำให้ผู้ใช้สามารถพัฒนาแอป พลิเคชันได้ง่าย โดยไม่ต้องเขียนรหัสคำสั่ง (Source code) ภาษาจาวา โดยมีการแบ่งคอมโพเนนต์ พื้นฐานเป็นกลุ่ม ดังนี้

1. User Interface components เป็นกลุ่มของ คอมโพเนนต์ที่ใช้ติดต่อกับผู้ใช้ เช่น ส่วนรับข้อความ หรือ ปุ่มคำสั่ง เป็นต้น

2. Layout components เป็นกลุ่มของคอมโพเนนต์ ที่ควบคุมการวางตำแหน่งของคอมโพเนนต์ที่ใช้ติดต่อกับ ผู้ใช้บนหน้าจอ Media components เป็นกลุ่มของคอมโพเนนต์ ที่ใช้แสดงสื่อในรูปแบบต่างๆ เช่น รูปภาพ วิดีโอ เสียง เป็นต้น

3. Drawing and Animation components เป็น กลุ่มของคอมโพเนนต์ที่ใช้วาดรูปและแสดงภาพเคลื่อนไหว Sensor components เป็นกลุ่มของคอมโพเนนต์ ที่ใช้ทำงานร่วมกับเซ็นเซอร์ของอุปกรณ์ต่างๆ เช่น จีพีเอส (GPS) และ เข็มทิศ

4. Social components เป็นกลุ่มของคอมโพเนนต์ ที่ใช้สื่อสารกับสังคมออนไลน์ เช่นการส่งเอสเอ็มเอส (SMS) การเรียกดูสมุดโทรศัพท์ หรือ การแชร์กับสังคมออนไลน์ เป็นต้น

5. Storage components เป็นกลุ่มของคอมโพเนนต์ ที่ใช้จัดการกับข้อมูลในฐานข้อมูลขนาดเล็ก หรือข้อมูลใน ไฟล์

6. Connectivity components เป็นกลุ่มของ คอมโพเนนต์ที่ใช้สร้างการเชื่อมต่อกับอุปกรณ์อื่นผ่าน บลูทูธ และเชื่อมต่อกับแอปพลิเคชันที่พัฒนาโดยใช้ ภาษาจาวา (Java Application)

ประโยชน์ของคอมโพเนนต์เหล่านี้คือ ทำให้การ พัฒนาแอปพลิเคชันเร็วขึ้น เพราะแต่ละคอมโพเนนต์มี หน้าที่เฉพาะ มีการจัดการกับข้อมูลของตนเอง ทำให้ นักพัฒนาไม่ต้องเสียเวลาในการทำงานพื้นฐานเหล่านั้น แต่จะใช้เวลาในการเลือกคอมโพเนนต์ที่เหมาะสมเพื่อนำมาประกอบกันเป็นแอปพลิเคชันแทน ซึ่งการพัฒนา แอปพลิเคชันในลักษณะนี้ จะเป็นการใช้ประโยชน์จาก แนวคิดของการนำกลับมาใช้ใหม่ (Reuse) [31] โดยมีสัญลักษณ์ดังรูปที่ 2.24



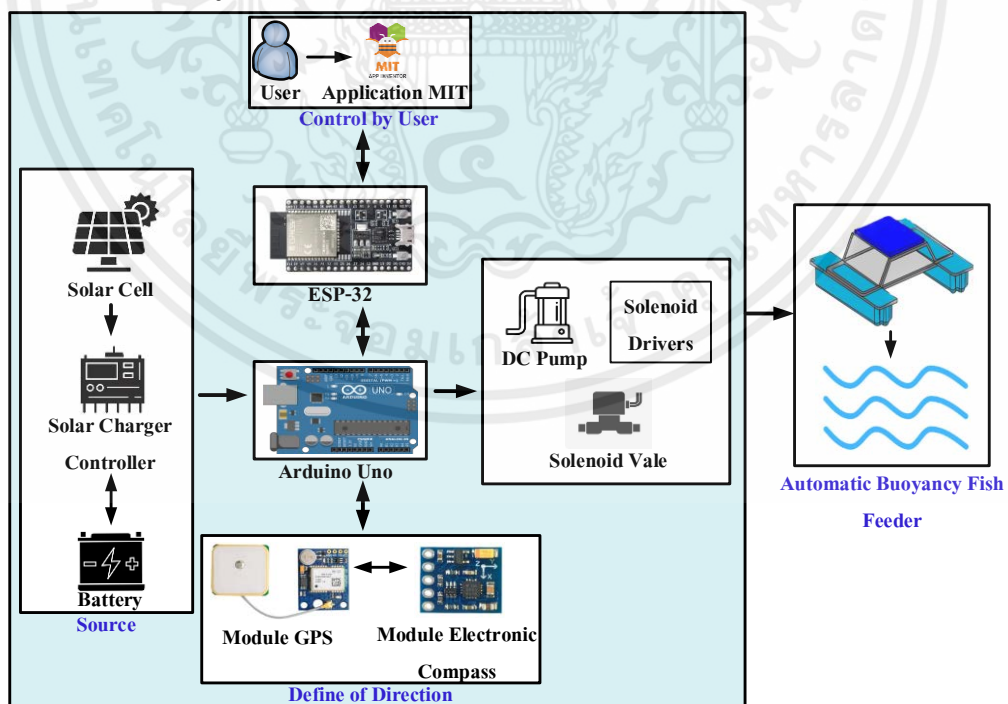
รูปที่ 2.24 สัญลักษณ์ของ MIT APP Inverter

บทที่ 3

การออกแบบและการจัดทำปฏิญญาพันธ

3.1 การออกแบบ

ปฏิญญาพันธฉบับนี้แนะนำให้เสนอสิ่งประดิษฐ์หุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno ควบคุมการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ และควบคุมทิศทางการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติโดยใช้แหล่งจ่ายไฟจากแบตเตอรี่และแผงพลังงานแสงอาทิตย์จ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno และบอร์ดไมโครคอนโทรลเลอร์ ESP-32 ใช้โมดูลระบุตำแหน่งพิกัด GPS ในการรับสัญญาณจาก GPS เพื่อรับค่าพิกัดละติจูด ลองจิจูด ในการระบุตำแหน่ง และใช้บอร์ดไมโครคอนโทรลเลอร์ ESP-32 เพื่อส่งข้อมูลไปบน Application ผ่าน MIT App Inventor ให้หุ่นลอยน้ำให้อาหารปลาอัตโนมัติเคลื่อนที่ไปข้างหน้าด้วยแรงดันน้ำจาก DC Pump ใช้โซลินอยด์วาล์วควบคุมทิศทางการเลี้ยวของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ แล้วทำการปล่อยอาหารปลาโดยใช้สปริงเกลียวติดกับมอเตอร์ในการหมุนเพื่อปล่อยอาหารปลา และสร้างออกซิเจนในบ่อเลี้ยงปลาด้วยการทำงานของโซลินอยด์วาล์ว แสดงได้ดังบล็อกไดอะแกรมรูปที่ 3.1

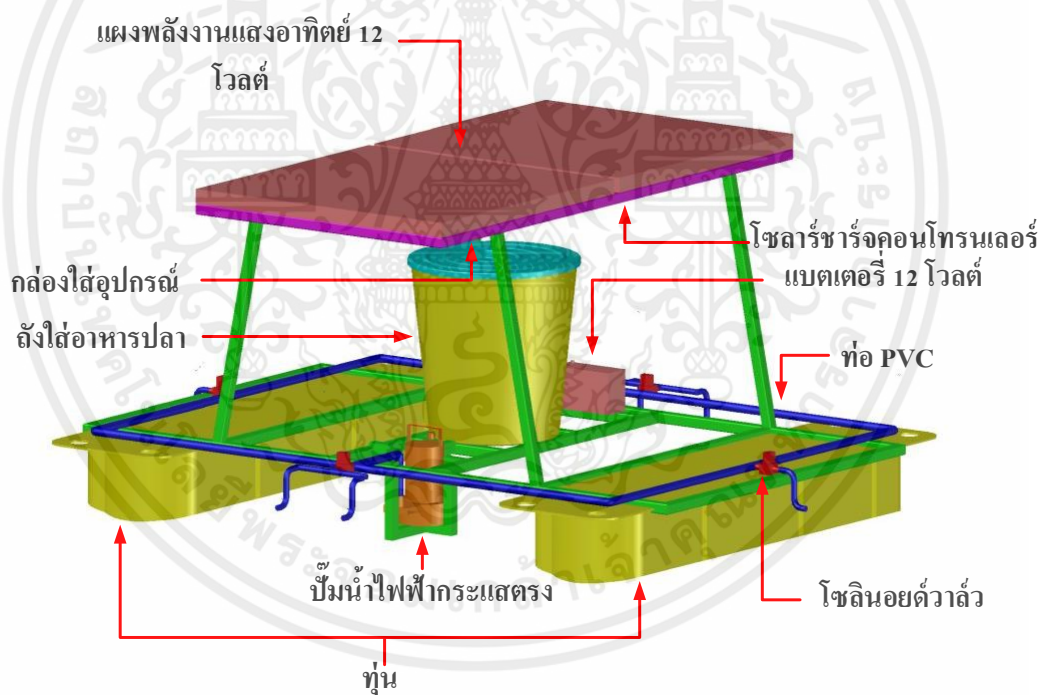


รูปที่ 3.1 บล็อกไดอะแกรมและอุปกรณ์การทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

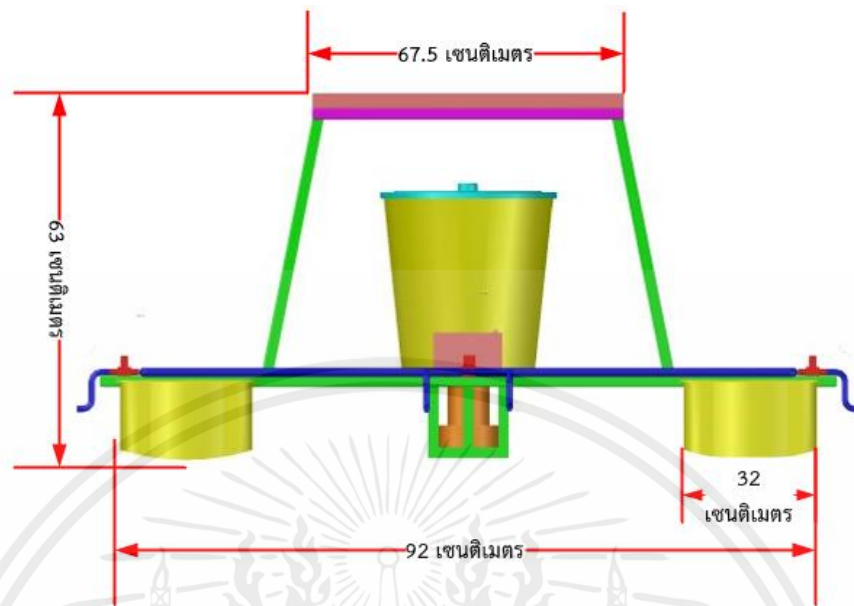
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การออกแบบและสร้างท่อนลอยน้ำให้อาหารปลาอัตโนมัติ

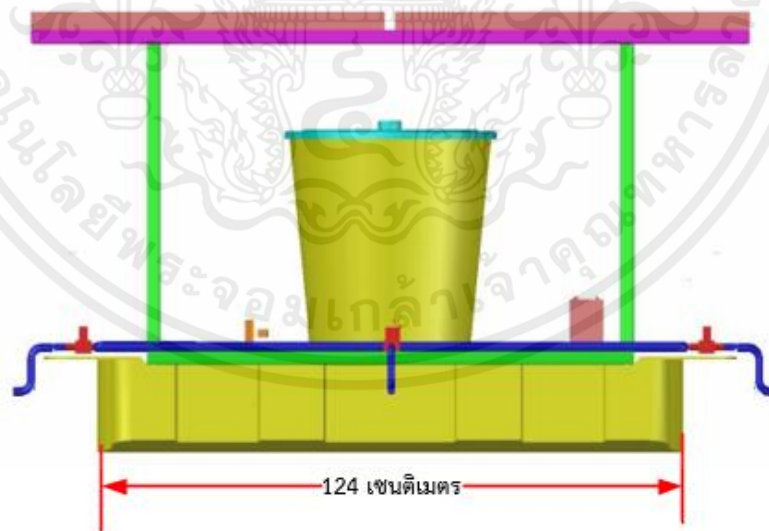
ในการออกแบบโครงสร้างท่อนลอยน้ำให้อาหารปลาอัตโนมัติ จะใช้โปรแกรม AutoCAD ในการออกแบบและจำลองโครงสร้างของท่อนลอยน้ำให้อาหารปลาอัตโนมัติ 3 มิติ แสดงดังรูปที่ 3.2 ถึงรูปที่ 3.5 ซึ่งรายละเอียดโครงสร้างด้านบนของท่อนลอยน้ำให้อาหารปลาอัตโนมัติจะติดตั้งแผงพลังงานแสงอาทิตย์ขนาด 12 โวลต์ 50 วัตต์ เพื่อเป็นแหล่งจ่ายไฟให้กับแบตเตอรี่ ด้านล่างแผงพลังงานแสงอาทิตย์จะติดกล่องกันน้ำสำหรับใส่อุปกรณ์เพื่อป้องกันไม่ให้อุปกรณ์เสียหายจากน้ำ โดยตรงกลางจะติดตั้งเพื่อบรรจุอาหารปลา ข้างบนท่อนจะติดท่อ PVC ทำเป็นวงรูป ที่มีเส้นผ่าศูนย์กลางภายนอก 1.25 นิ้ว ด้านซ้ายและด้านขวาติดโซลินอยด์วาล์ว 12 โวลต์ หลังจากโซลินอยด์วาล์วใช้ท่อ PVC ที่มีเส้นผ่าศูนย์กลาง 1 นิ้ว ออกมา 1 ทาง ด้านหน้าและด้านหลังติดโซลินอยด์วาล์ว 12 โวลต์ หลังจากโซลินอยด์วาล์วใช้ท่อ PVC ที่มีเส้นผ่าศูนย์กลาง 1 นิ้ว ออกมา 2 ทาง และด้านหน้าท่อนลอยน้ำให้อาหารปลาอัตโนมัติจะติด DC Pump แสดงดังรูปที่ 3.6



รูปที่ 3.2 ภาพจำลอง 3 มิติโครงสร้างของท่อนลอยน้ำให้อาหารปลาอัตโนมัติ

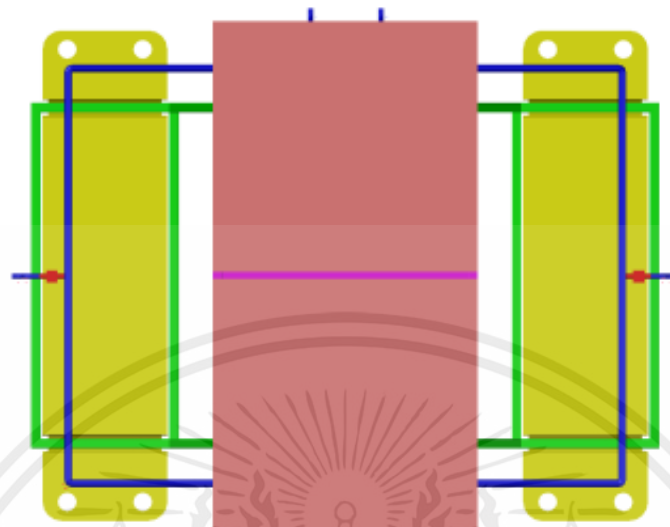


รูปที่ 3.3 ภาพจำลอง 3 มิติโครงสร้างด้านหน้าของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

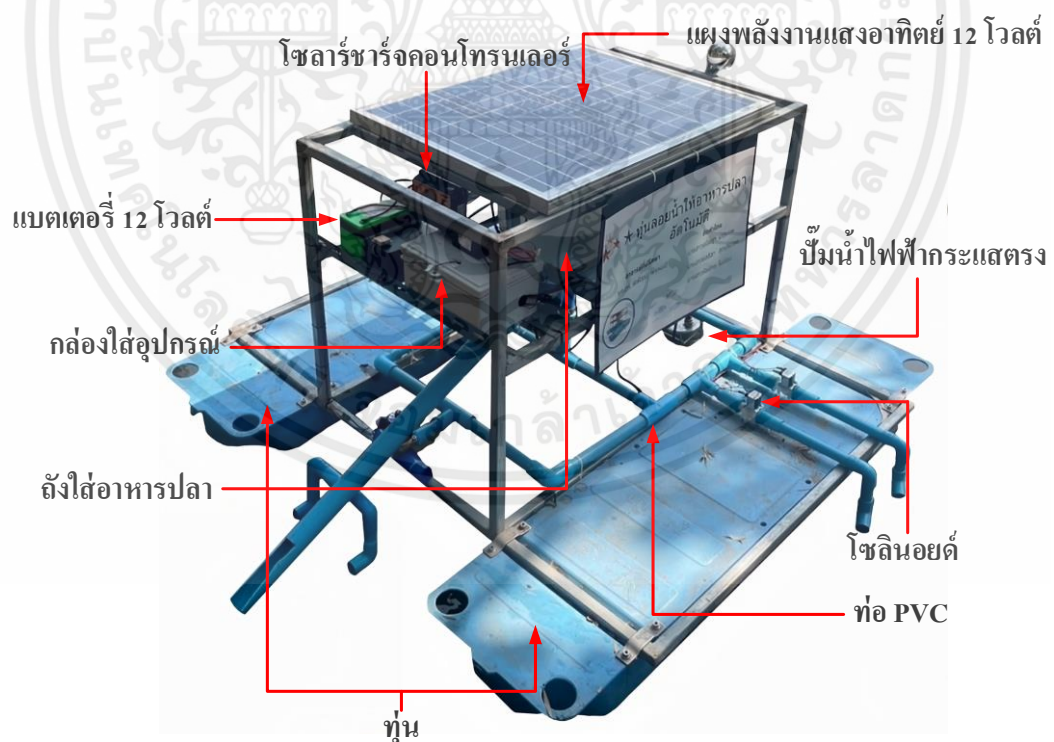


รูปที่ 3.4 ภาพจำลอง 3 มิติโครงสร้างด้านข้างของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ภาพจำลอง 3 มิติโครงสร้างด้านบนของฟูลอยน้ำให้อาหารปลาอัตโนมัติ



รูปที่ 3.6 ภาพถ่ายโครงสร้างของฟูลอยน้ำให้อาหารปลาอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบระบบควบคุมขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

หลักการการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยเมื่อแผงพลังงานแสงอาทิตย์ได้รับแสงแดด โซลาร์ชาร์จคอนโทรลเลอร์จะทำหน้าที่ควบคุมการชาร์จไฟฟ้าที่ได้รับจากแผงพลังงานแสงอาทิตย์ มาเก็บในแบตเตอรี่อย่างเหมาะสมและทำหน้าที่ควบคุมการจ่ายไฟให้กับโหลดไฟฟ้า เมื่อแบตเตอรี่จ่ายไฟให้กับโหลดส่งผลให้โซลินอยด์วาล์วและ DC Pump ทำงาน และใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno ควบคุมการทำงาน โดยใช้ MOSFET มาควบคุมแรงดันและกระแสไฟระหว่างแหล่งจ่าย ทำงานเหมือนเป็นตัวสวิตช์เปิดปิด บล็อกไดอะแกรมแสดงดังรูปที่ 3.7

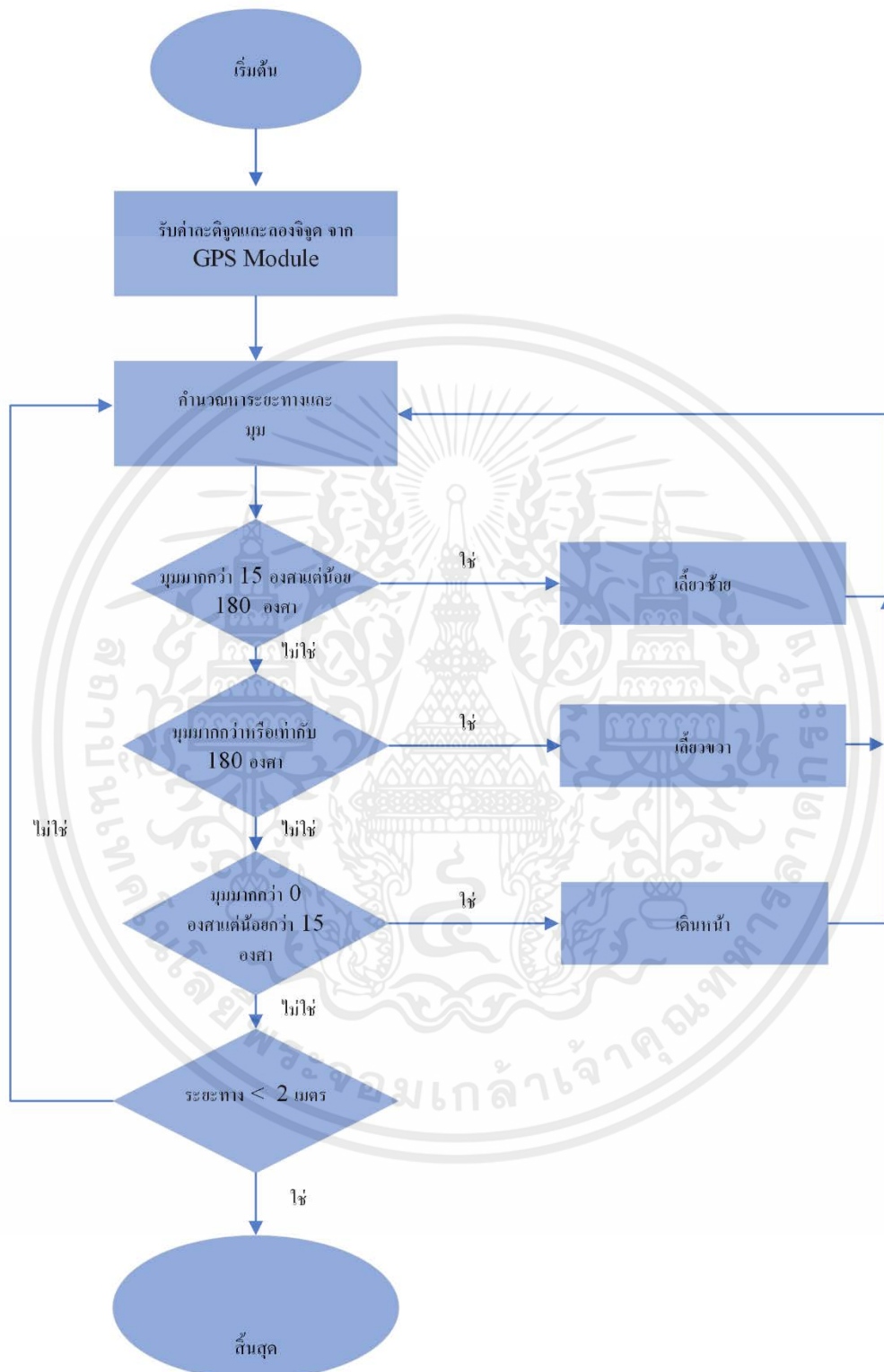


รูปที่ 3.7 บล็อกไดอะแกรมระบบควบคุมขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

3.1.2.1 การออกแบบการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

รูปแบบการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ แสดงดังรูปที่ 3.8 จะใช้โซลินอยด์วาล์ว 5 อัน คือ ด้านหลัง 1 อัน ด้านซ้าย 2 อัน และด้านขวา โดยแบ่งการเคลื่อนที่ออกเป็น 3 รูปแบบ ดังนี้

1. เคลื่อนที่ไปด้านหน้า จะกำหนดให้โซลินอยด์วาล์วด้านหลังเปิด ส่วนโซลินอยด์วาล์วด้านหน้า ด้านซ้าย และด้านขวาปิด
2. เคลื่อนที่ไปทางซ้าย จะกำหนดให้โซลินอยด์วาล์วด้านขวาเปิด ส่วนโซลินอยด์วาล์วด้านหน้า ด้านซ้าย และด้านหลังปิด
3. เคลื่อนที่ไปทางขวา จะกำหนดให้โซลินอยด์วาล์วด้านซ้ายเปิด ส่วนโซลินอยด์วาล์วด้านหน้า ด้านหลัง และด้านขวาปิด



รูปที่ 3.8 บล็อกไดอะแกรมการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

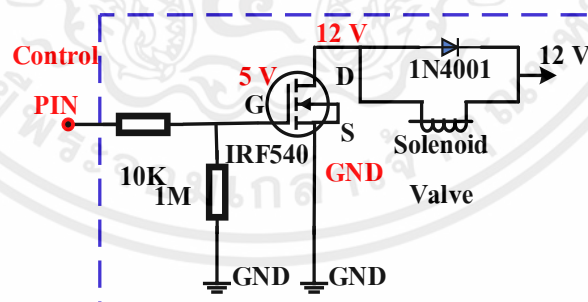
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.2 การออกแบบการขับเคลื่อนของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

ทำการออกแบบให้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno ทำงานร่วมกับ วงจรขับโซลินอยด์และวงจรแปลงไฟ แสดงดังรูปที่ 3.9 เพื่อควบคุมการขับเคลื่อนของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ แสดงดังรูปที่ 3.10 และตารางที่ 3.1

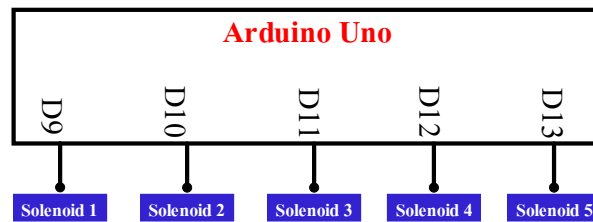
ตารางที่ 3.1 การเชื่อมต่อวงจรระหว่างบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno กับวงจร Solenoid drivers

บอร์ดไมโครคอนโทรลเลอร์ ESP-32	วงจร Solenoid drivers
GND	GND
VIN	5V
PIN 12	Solenoid drivers 1 (ขับเคลื่อนไปด้านหน้า)
PIN 11	Solenoid drivers 3 (ขับเคลื่อนไปทางซ้าย)
PIN 10	Solenoid drivers 4 (ขับเคลื่อนไปทางขวา)
PIN 9	Solenoid drivers 5 (ควบคุมการเปิดปิด DC Pump)



รูปที่ 3.9 ออกแบบควบคุมการทำงานของโซลินอยด์

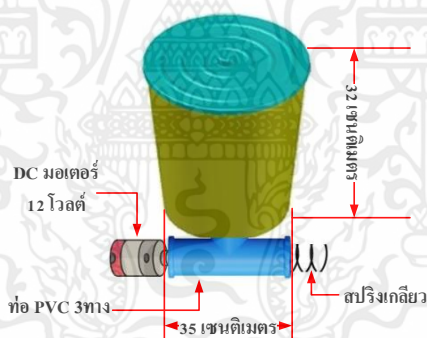
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ออกแบบการขับเคลื่อนของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

3.1.3 การออกแบบถังให้อาหารปลา

ออกแบบถังให้อาหารปลา โดยใช้ถังที่เป็นรูปทรงกระบอก สูง 32 เซนติเมตร มีขนาดเส้นผ่าศูนย์กลาง 35 เซนติเมตร ที่สามารถบรรจุอาหารปลาหนัก 5 กิโลกรัม ได้ และนำเอาสปริงต่อเข้ากับมอเตอร์ และท่อ PVC 3 ทาง ขนาดเส้นผ่าศูนย์กลาง 4.80 เซนติเมตร ซึ่งทำหน้าที่ควบคุมการจ่ายอาหารปลา กำหนดให้ขดลวดมีทั้งหมด 9 ขดลวด ความยาวสปริงเท่ากับ 30 เซนติเมตร แสดงดังรูปที่ 3.11



รูปที่ 3.11 อุปกรณ์ในการปล่อยอาหารปลา

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 ไมโครคอนโทรลเลอร์

ในปฏิญานพนธ์นี้ ผู้จัดทำเลือกใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno เป็นบอร์ดไมโครคอนโทรลเลอร์ open-source มีขา Analog pin 6 ขา และขา Digital pin 14 ขา โดยมี 6 ขา สามารถใช้เป็น output แบบ PWM ได้ รองรับการเขียนโปรแกรมโดยใช้โปรแกรม Arduino IDE สามารถเชื่อมต่อกับคอมพิวเตอร์ด้วยสายเคเบิล USB หรือใช้ adaptor AC-to-DC เพื่อเริ่มต้นใช้

งานและเลือกใช้บอร์ดไมโครคอนโทรลเลอร์ ESP-32 NodeMCU ESP-WROOM-32 สามารถรองรับทั้ง Wi-Fi และ Bluetooth ใช้งานได้ 30 ขา รองรับการเขียนโปรแกรมโดยใช้ Arduino IDE

3.2.2 มอเตอร์ไฟฟ้ากระแสตรง

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้มอเตอร์ไฟฟ้ากระแสตรง แรงดันไฟฟ้า 12 โวลต์ เนื่องจากต้องการให้สปริงเกลียวที่ติดกับมอเตอร์หมุนเพื่อใช้ในการจ่ายอาหารปลา

3.2.3 โมดูลระบุตำแหน่งพิกัด GPS

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้โมดูลระบุตำแหน่งพิกัด GPS รุ่น GY-NEO6MV2 เชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 เพื่อระบุตำแหน่งให้กับหุ่นลอยน้ำให้อาหารปลาอัตโนมัติในการจ่ายอาหารปลา ณ จุดนั้น

3.2.4 โมดูลเข็มทิศ

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้โมดูลวัดสนามแม่เหล็ก 3 แกน โมดูลเข็มทิศ GY-273 ใช้ชิป QMC5883L เชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 เพื่อระบุทิศทางในการขับเคลื่อนให้กับหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

3.2.5 วงจรแปลงกระแสไฟ

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้วงจรแปลงกระแสไฟ เพื่อแปลงกระแสไฟจาก 12 โวลต์ ที่จ่ายกระแสไฟให้กับ DC Pump, มอเตอร์ และโซลินอยด์วาล์ว เป็น 5 โวลต์ เพื่อจ่ายกระแสไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno, บอร์ดไมโครคอนโทรลเลอร์ ESP-32, โมดูลระบุตำแหน่งพิกัด GPS และโมดูลเข็มทิศ

3.2.6 แผงพลังงานแสงอาทิตย์

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้แผงพลังงานแสงอาทิตย์ 50 วัตต์ เพื่อนำพลังงานไฟฟ้าไปเก็บไว้ในแบตเตอรี่และนำไปใช้กับหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

3.2.7 โซลาร์ชาร์จคอนโทรลเลอร์

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้โซลาร์ชาร์จคอนโทรลเลอร์ เพื่อควบคุมการชาร์จไฟฟ้าที่ได้รับจากแผงพลังงานแสงอาทิตย์ มาเก็บในแบตเตอรี่อย่างเหมาะสม

3.2.8 แบตเตอรี่

ในปฏิญานิพนธ์นี้ ผู้จัดทำเลือกใช้แบตเตอรี่ 12 โวลต์ 20 แอมป์ เพื่อเป็นแหล่งจ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno, บอร์ดไมโครคอนโทรลเลอร์ ESP-32, โมดูลระบุตำแหน่งพิกัด GPS, โมดูลเข็มทิศ DC Pump มอเตอร์, และโซลินอยด์วาล์ว

3.2.9 DC Pump

ในปฏิญญาฉบับนี้ ผู้จัดทำเลือกใช้ DC Pump แรงดันไฟฟ้า 12 โวลต์ เพื่อดูน้ำเข้าไปให้กับท่อ PVC ที่ทำเป็นวนรูป ใช้ในการขับเคลื่อนของหุ่นยนต์น้ำให้อาหารปลาอัตโนมัติ

3.3 การจัดเก็บผลการทดลอง

3.3.1 ผลการทดสอบการทำงานของโมดูลระบุตำแหน่งพิกัด GPS

ทดสอบโดยการเชื่อมต่อโมดูลระบุตำแหน่งพิกัด GPS รุ่น GY-NEO6MV2 เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 แล้วรับค่าละติจูด ลองติจูดเพื่อแสดงบนหน้าจอ Serial Monitor

3.3.2 ผลการทดสอบการทำงานของโมดูลเข็มทิศ

ทดสอบโดยการเชื่อมต่อโมดูลวัดสนามแม่เหล็ก 3 แกน โมดูลเข็มทิศ GY-273 ใช้ชิป HMC5883L เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 แล้วรับค่าองศาเพื่อแสดงบนหน้าจอ Serial Monitor

3.3.3 ผลการทดสอบการชาร์จแผงพลังงานแสงอาทิตย์เข้ากับแบตเตอรี่

ทดสอบโดยการเชื่อมต่อแผงพลังงานแสงอาทิตย์ เข้ากับแบตเตอรี่ โดยที่โซลาร์ชาร์จคอนโทรลเลอร์จะต่อระหว่างแผงพลังงานแสงอาทิตย์กับแบตเตอรี่ จากนั้นบันทึกค่าระยะเวลาในการชาร์จ จนกระทั่งแบตเตอรี่มีระดับแรงดันไฟฟ้าเต็ม 12 โวลต์

3.3.4 ผลการการทำงานของ DC Pump

ทดสอบโดยการเชื่อมต่อ DC Pump 12 เข้ากับแผงพลังงานแสงอาทิตย์, แบตเตอรี่ และโซลาร์ชาร์จคอนโทรลเลอร์ โดยทำการทดสอบ 2 รูปแบบ คือ ทดสอบให้กระแสน้ำไหลออก 1 ทาง และให้กระแสน้ำไหลออก 2 ทาง จากนั้นบันทึกระยะเวลาและปริมาณของกระแสน้ำที่ไหลออกมา

3.3.5 ผลการทดสอบระบบปล่อยอาหารปลาและระบบสร้างออกซิเจนในบ่อ

ทดสอบโดยการปล่อยอาหารปลาครั้งละ 187.5 กรัม ผ่านการหมุนของสปริงที่ติดอยู่กับมอเตอร์ จากนั้นทำการบันทึกระยะเวลาในการปล่อยจนหมดตามจำนวนที่กำหนดและระบบสร้างออกซิเจนในบ่อ

3.3.6 ผลการทดสอบการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด

ทดสอบโดยการหาค่าความคลาดเคลื่อนโดยเปรียบเทียบจากค่าระยะทางที่วัดจริง เทียบกับค่าระยะทางที่หุ่นลอยน้ำให้อาหารปลาอัตโนมัติเคลื่อนที่ไปตามจุดพิกัดละติจูด ลองจิจูด ที่กำหนดไว้

3.3.7 ผลการทดสอบการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

ทดสอบโดยการเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno เข้ากับวงจรขับโซลินอยด์ โดยใช้แบตเตอรี่เป็นแหล่งจ่ายพลังงานไฟฟ้าให้กับ DC Pump, โซลินอยด์ และบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno

3.3.8 ผลการทดสอบการทำงานของแอปพลิเคชัน

ทดสอบโดยการสั่งการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติให้ทำงานตามพิกัดละติจูด ลองจิจูด ที่กำหนดไว้ รวมถึงดูสถานะของแบตเตอรี่ ปริมาณอาหารปลาที่เหลือ

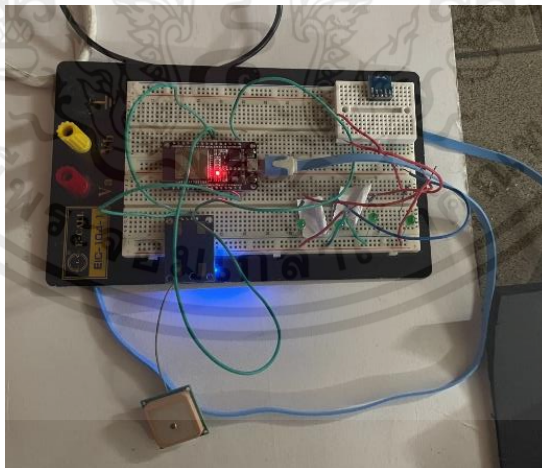
บทที่ 4

ผลการทดลอง

ผู้จัดทำได้ทำการเก็บผลการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยแบ่งการทดลองและจัดเก็บผลการทดลองเป็นส่วนๆ ดังต่อไปนี้

4.1 ผลการทดสอบการทำงานของโมดูลระบุตำแหน่งพิกัด GPS

จากการเชื่อมต่อโมดูลระบุตำแหน่งพิกัด GPS รุ่น GY-NEO6MV2 เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 แสดงดังรูปที่ 4.1 โดยการทำงานของโมดูลระบุตำแหน่งพิกัด GPS เซนเซอร์จะทำการค้นหาและรับสัญญาณพิกัดที่ส่งจากดาวเทียม ในการทดสอบต้องทดสอบในพื้นที่กลางแจ้งที่มีมุมมองสูงขึ้นไปเพื่อให้สามารถรับสัญญาณจากดาวเทียมได้ดีที่สุด เมื่อโมดูลรับสัญญาณได้แล้วจะแสดงไฟ LED สีฟ้ากระพริบ และแสดงค่าละติจูด ลองจิจูดบนหน้าจอ Serial Monitor แสดงดังรูปที่ 4.2 การทดสอบความถูกต้องชุดรับสัญญาณ GPS ในการทดสอบได้ทำการทดสอบทั้งหมด 10 ครั้ง แสดงได้ดังรูปที่ 4.3 ซึ่งผลการรับสัญญาณจากดาวเทียมสามารถแสดงได้ดังตารางที่ 4.1 และทดสอบระยะเวลาในการรับสัญญาณจากดาวเทียม ในการทดสอบได้ทำการทดสอบทั้งหมด 10 ครั้ง แสดงได้ดังรูปที่ 4.4



รูปที่ 4.1 เชื่อมต่อโมดูลระบุตำแหน่งพิกัด GPS รุ่น GY-NEO6MV2 เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32

Output Serial Monitor ×

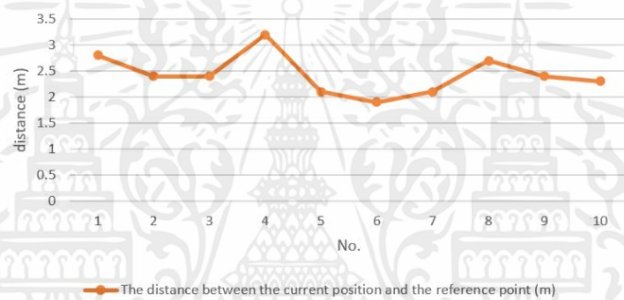
Message (Enter to send message to 'ESP32-WROOM

```

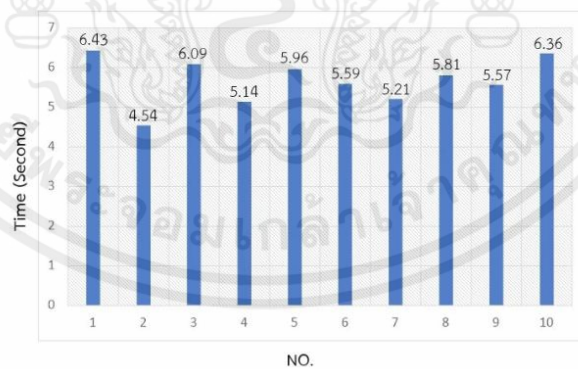
17:12:29.140 ->
17:12:29.235 -> 13.644953,100.620276
17:12:29.280 ->
17:12:30.114 -> 13.644953,100.620276
17:12:30.159 ->
17:12:30.239 -> 13.644953,100.620276
17:12:30.285 ->
17:12:31.115 -> 13.644952,100.620277
17:12:31.161 ->
17:12:31.272 -> 13.644952,100.620277
17:12:31.272 ->

```

รูปที่ 4.2 ค่าละติจูด ลองจิจูดบนหน้าจอ Serial Monitor



รูปที่ 4.3 การทดสอบความถูกต้องจุดรับสัญญาณ GPS



รูปที่ 4.4 การทดสอบระยะเวลาในการรับสัญญาณจากดาวเทียม

จากการทดสอบพบว่า ระยะเวลาในการรับสัญญาณจากดาวเทียมมีค่าระยะเวลารับสัญญาณสูงสุดเท่ากับ 6.43 วินาที และต่ำสุดเท่ากับ 4.54 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ผลการทดสอบความถูกต้องชุดรับสัญญาณ GPS รุ่น GY-NEO6MV2 เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 ในการทดสอบได้ทำการทดสอบทั้งหมด 10 ครั้งนั้นจะแสดงในตารางดังนี้

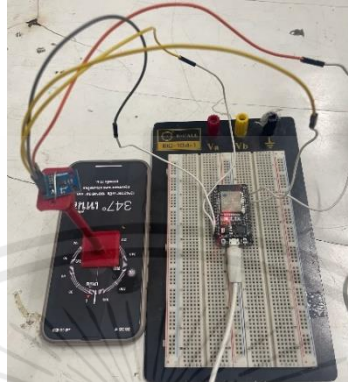
ครั้งที่	ค่าละติจูดและลองจิจูดจาก Application Google Maps ในโทรศัพท์	ระยะห่างระหว่างตำแหน่งปัจจุบันกับตำแหน่งอ้างอิง (เมตร)
1	13.7271935, 100.7758280	2.80
2	13.7271531, 100.7752289	2.40
3	13.7271580, 100.7748681	2.40
4	13.7270020, 100.7746686	3.20
5	13.7267518, 100.7746757	2.10
6	13.7263949, 100.7747950	1.90
7	13.7264076, 100.7754398	2.10
8	13.7264131, 100.7760245	2.70
9	13.7266391, 100.7768645	2.40
10	13.7264112, 100.7750198	2.30
เฉลี่ย		2.43 เมตร

จากตารางการทดสอบเมื่อทำการทดสอบความถูกต้องของโมดูลระบุตำแหน่งพิกัด GPS โดยอิงค่าละติจูดและลองจิจูดจาก Application Google Maps ในโทรศัพท์ พบว่า มีค่าระยะห่างระหว่างตำแหน่งปัจจุบันกับตำแหน่งอ้างอิงกับเท่า 2.43 เมตร

4.2 ผลการทดสอบการทำงานของโมดูลเข็มทิศ

จากการเชื่อมต่อโมดูลวัดสนามแม่เหล็ก 3 แกน โมดูลเข็มทิศ GY-273 ใช้ชิป QMC5883L เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 แสดงดังรูปที่ 4.5 โดยการทำงานของโมดูลเข็มทิศ จะต้องจัดวางโมดูลให้แกน x ของโมดูลตรงกับทิศที่ต้องการทดสอบ แล้วทำการหมุนเซนเซอร์ไปยังทิศต่างๆ ที่ต้องการทดสอบ จากนั้นจะแสดงค่าองศาบนหน้าจอ Serial Monitor

แสดงดังรูปที่ 4.6 การทดสอบความถูกต้องของโมดูลเข็มทิศ ในการทดสอบได้ทำการทดสอบทั้งหมด 10 ครั้ง แสดงได้ดังรูปที่ 4.7 ซึ่งผลของค่าองศาสามารถแสดงได้ดังตารางที่ 4.2

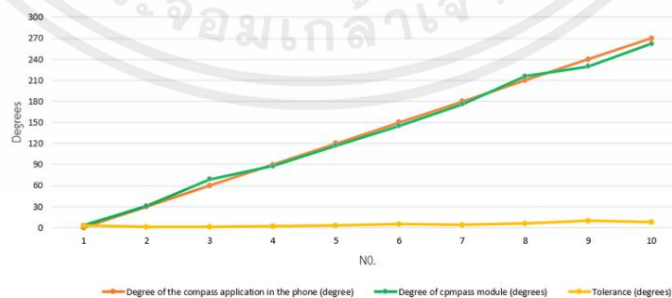


รูปที่ 4.5 เชื่อมต่อโมดูลวัดสนามแม่เหล็ก 3 แกน โมดูลเข็มทิศ GY-273 ใช้ชิป QMC5883L เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32

Output	Serial Monitor
Message (Enter to send message to 'Arduino Uno' on 'COM9')	
20:45:06.44	-> Heading = 6.21 Degree = 355.76
20:45:06.541	-> Heading = 6.21 Degree = 355.70
20:45:06.636	-> Heading = 6.21 Degree = 355.70
20:45:06.760	-> Heading = 6.21 Degree = 355.71
20:45:06.824	-> Heading = 6.21 Degree = 355.60
20:45:06.964	-> Heading = 6.21 Degree = 355.61
20:45:07.056	-> Heading = 6.21 Degree = 355.65
20:45:07.148	-> Heading = 6.21 Degree = 355.65
20:45:07.273	-> Heading = 6.21 Degree = 355.69
20:45:07.367	-> Heading = 6.21 Degree = 355.74
20:45:07.460	-> Heading = 6.21 Degree = 355.74

รูปที่ 4.6 ค่าองศาบนหน้าจอ Serial Monitor

The graph illustrates the relationship between the degrees information from the phone application and the degrees information from the compass module.



รูปที่ 4.7 การทดสอบความถูกต้องของโมดูลเข็มทิศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดสอบความถูกต้องของโมดูลเข็มทิศ GY-273 ใช้ชิป QMC5883L เข้ากับบอร์ดไมโครคอนโทรลเลอร์ ESP-32 ในการทดสอบได้ทำการทดสอบทั้งหมด 10 ครั้งนั้นจะแสดงในตารางดังนี้

ครั้งที่	ค่าองศาจาก Application เข็มทิศ ในโทรศัพท์ (องศา)	ค่าองศาจาก โมดูล เข็มทิศ (องศา)	ค่าความคลาดเคลื่อน (องศา)
1	0	3	3
2	30	31	1
3	60	69	1
4	90	88	2
5	120	117	3
6	150	145	5
7	180	176	4
8	210	216	6
9	240	230	10
10	270	262	8
เฉลี่ย		4.3 องศา	

จากตารางการทดลองเมื่อทำการทดสอบความถูกต้องของโมดูลเข็มทิศ โดยอิงค่าองศาของโมดูลเข็มทิศ กับ Application เข็มทิศในโทรศัพท์ พบว่า มีค่าความคลาดเคลื่อน 4.3 องศา

4.3 ผลการทดสอบการชาร์จแผงพลังงานแสงอาทิตย์เข้ากับแบตเตอรี่

จากการจากเชื่อมต่อแผงพลังงานแสงอาทิตย์ 50 วัตต์ เข้ากับแบตเตอรี่ 12 โวลต์ โดยที่โซลาร์ชาร์จคอนโทรลเลอร์จะต่อระหว่างแผงพลังงานแสงอาทิตย์กับแบตเตอรี่ แสดงดังรูปที่ 4.8 เพื่อนำพลังงานที่ได้มาใช้ในระบบขับเคลื่อนหุ่นยนต์ให้อาหารปลาอัตโนมัติและมาใช้ในระบบการหมุนของมอเตอร์เพื่อจ่ายอาหารปลา ระยะเวลาในการชาร์จแผงพลังงานแสงอาทิตย์เข้ากับแบตเตอรี่ ในการทดลองได้ทดลองทั้งหมด 5 ครั้ง ในช่วงเวลา 10:00-14:00 น. ผลการชาร์จแสดงดังตารางที่ 4.3



รูปที่ 4.8 การชาร์จแผงพลังงานแสงอาทิตย์เข้ากับแบตเตอรี่

จากรูปที่ 4.8 ที่ได้ทำการเชื่อมต่อแผงพลังงานแสงอาทิตย์ 50 วัตต์ เข้ากับแบตเตอรี่ 12 โวลต์ โดยที่โซลาร์ชาร์จคอนโทรลเลอร์จะต่อระหว่างแผงพลังงานแสงอาทิตย์กับแบตเตอรี่ โดยทำการวัดค่ากระแสไฟฟ้าที่แผงพลังงานแสงอาทิตย์ต่อชาร์จเข้ากับแบตเตอรี่ แผงพลังงานแสงอาทิตย์สามารถจ่ายกระแสไฟฟ้าได้ 1.540 แอมป์

ตารางที่ 4.3 ผลการทดสอบการชาร์จแผงพลังงานแสงอาทิตย์ 50 วัตต์ เข้ากับแบตเตอรี่ 12 โวลต์ โดยเริ่มทำการชาร์จจากปริมาณแบตเตอรี่ที่เหลือ 30 % ระยะเวลาในการชาร์จแผงพลังงานแสงอาทิตย์เข้ากับแบตเตอรี่ ในการทดลองได้ทดลอง 5 ครั้ง ในช่วงเวลา 10:00-14:00 น. ซึ่งผลการชาร์จนั้นจะแสดงในตารางดังนี้

ครั้งที่	เวลาชาร์จโซลาร์เซลล์เข้ากับ แบตเตอรี่ 12 โวลต์ (V) / ชม.	แบตเตอรี่ 12 โวลต์ (V)
1	10:00 ชม.	เต็ม 12 V
2	11:30 ชม.	เต็ม 12 V
3	11:00 ชม.	เต็ม 12 V
4	12:00 ชม.	เต็ม 12 V
5	11:00 ชม.	เต็ม 12 V
เฉลี่ย	11.06 ชั่วโมง/แบตเตอรี่ 12V	

จากตารางการทดลองพบว่าแผงพลังงานแสงอาทิตย์ 50 วัตต์ โดยเริ่มทำการชาร์จจากปริมาณแบตเตอรี่ที่เหลือ 30 % สามารถชาร์จเข้ากับแบตเตอรี่ 12 โวลต์ ได้ในเวลาเฉลี่ย 11.06

ชั่วโมง ซึ่งเวลาในการชาร์จขึ้นอยู่กับสภาพแสงแดดในวันนั้นๆ ด้วย ถึงสามารถชาร์จแบตเตอรี่ได้เต็ม และสามารถจ่ายพลังงานไฟฟ้าให้กับท่อนลอยน้ำให้อาหารปลาอัตโนมัติได้

4.4 ผลการการทำงานของ DC Pump

จากการเชื่อมต่อ DC Pump 12 โวลต์ เข้ากับแผงพลังงานแสงอาทิตย์, แบตเตอรี่ 12 โวลต์ และโซลาร์ชาร์จคอนโทรลเลอร์ แสดงดังรูปที่ 4.9 โดยแบ่งการทดสอบเป็น 2 รูปแบบ คือ ทดสอบอัตราการไหลของน้ำโดยใช้ท่อ PVC ที่มีขนาดเส้นผ่าศูนย์กลางภายนอก 1 นิ้ว 1 ทาง และทดสอบอัตราการไหลของน้ำในกรณีต่อท่อ PVC 2 ทาง ในการทดสอบได้ทำการทดสอบทั้งหมด 3 ครั้ง ซึ่งผลของกระแสไฟฟ้าที่ไหลสามารถแสดงได้ดังตารางที่ 4.4



รูปที่ 4.9 การเชื่อมต่อ DC Pump เข้ากับแผงพลังงานแสงอาทิตย์ แบตเตอรี่ และโซลาร์ชาร์จคอนโทรลเลอร์

ตารางที่ 4.4 ผลการทดสอบอัตราการไหลของน้ำโดยใช้ท่อ PVC ที่มีขนาดเส้นผ่าศูนย์กลางภายนอก 1 นิ้ว 1 ทาง และผลการทดสอบอัตราการไหลของน้ำโดยใช้ท่อ PVC ที่มีขนาดเส้นผ่าศูนย์กลางภายนอก 1 นิ้ว 2 ทาง ซึ่งในการทดสอบใช้น้ำปริมาณ 7 ลิตร ในการทดสอบได้ทดสอบทั้งหมด 3 ครั้ง ซึ่งผลอัตราการไหลของน้ำนั้นจะแสดงในตารางดังนี้

ครั้งที่	อัตราการไหลของน้ำในกรณีต่อท่อ PVC 1 ทาง (ลิตร/1 นาที)	อัตราการไหลของน้ำในกรณีต่อท่อ PVC 2 ทาง (ลิตร/1 นาที)
1	37.50	33.54
2	38.07	35.59
3	34.11	30.81
เฉลี่ย	36.56	33.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางการทดสอบพบว่า DC Pump ใช้กระแสไฟจากแผงพลังงานแสงอาทิตย์ และแบตเตอรี่โดยต่อเข้ากับโซลาร์ชาร์จเจอร์ จากที่ทำการทดสอบกับอุปกรณ์จริง และทำการวัดค่ากระแสไฟฟ้าที่ DC Pump ใช้งาน จ่ายกระแสไฟฟ้าได้ 2.580 แอมป์

4.5 ผลการทดสอบระบบปล่อยอาหารปลาและระบบสร้างออกซิเจนในบ่อ

จากการทดสอบการปล่อยอาหารปลา โดยใช้มอเตอร์ 12 โวลต์ ในการหมุนสปริงเกลียวเพื่อปล่อยอาหารปลา แสดงดังรูปที่ 4.10 จากการคำนวณปลานิลควรได้รับอาหาร 1.5 กิโลกรัม/1 วัน ใน 1 วันจะให้อาหารปลา 2 ครั้ง ครั้งละ 750 กรัม โดยจำลองการทดสอบปล่อยอาหารปลา 4 จุด จุดละ 187.5 กรัม ในการทดสอบการปล่อยอาหารปลาใช้ขนาดอาหารปลา ที่มีเส้นผ่าศูนย์กลาง 0.7 เซนติเมตร แสดงดังรูปที่ 4.11 ในการทดสอบได้ทดสอบทั้งหมด 10 ครั้ง แสดงดังรูปที่ 4.12 ผลการทดสอบการปล่อยอาหารปลาสามารถแสดงได้ดังตารางที่ 4.5 และมีระบบสร้างออกซิเจนในบ่อ แสดงดังรูปที่ 4.13

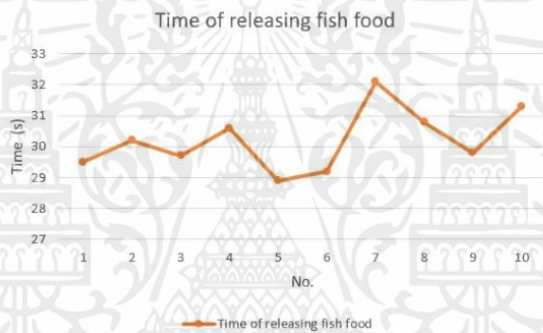


รูปที่ 4.10 การปล่อยอาหารปลา โดยใช้มอเตอร์ 12 โวลต์ ในการหมุนสปริงเกลียวเพื่อปล่อยอาหารปลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 อาหารปลา ที่มีเส้นผ่าศูนย์กลาง 0.7 เซนติเมตร ใช้ในการทดสอบการปล่อยอาหารปลา



รูปที่ 4.12 การทดสอบการปล่อยอาหารปลา



รูปที่ 4.13 การสร้างออกซิเจนในบ่อเลี้ยงปลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

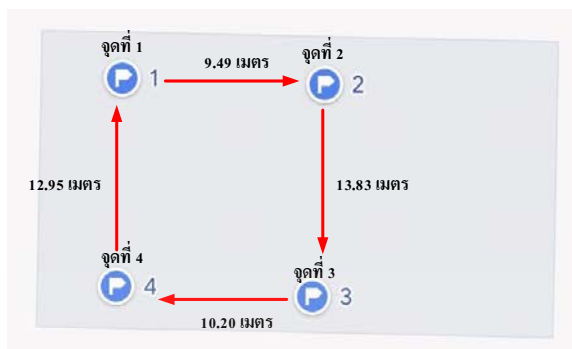
ตารางที่ 4.5 ผลการทดสอบการปล่อยอาหารปลา ได้ทดสอบทั้งหมด 10 ครั้ง ซึ่งผลของการปล่อยอาหารปลานั้นจะแสดงในตารางดังนี้

ครั้งที่	เวลาที่ปล่อยอาหารปลา 187.5 กรัม (วินาที)
1	44:50
2	44:10
3	50:48
4	49:15
5	46:33
6	44:25
7	43:30
8	40:13
9	50:55
10	50:88
เฉลี่ย	46.37

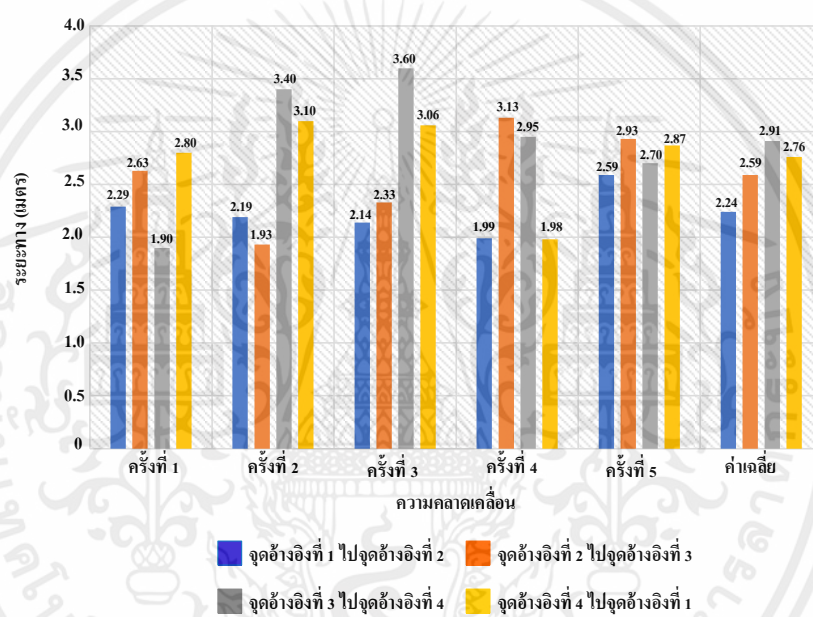
จากตารางการทดสอบพบว่า การปล่อยอาหารปลาใช้ระยะเวลาเฉลี่ยในการปล่อยอาหารปลา 187.5 กรัม เท่ากับ 46.37 วินาที

4.6 ผลการทดสอบการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด

จากการทดสอบการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด ซึ่งได้กำหนดจุดอ้างอิงทั้งหมด 4 จุด แสดงดังรูปที่ 4.14 แล้วทำการวัดระยะทางการเคลื่อนที่ของจุดอ้างอิงที่ 1-2 จุดอ้างอิงที่ 2 - 3 จุดอ้างอิงที่ 3 - 4 และจุดอ้างอิงที่ 4 - 1 จากนั้นหาค่าความคลาดเคลื่อนโดยเปรียบเทียบจากค่าระยะทางที่วัดจริงเทียบกับค่าระยะทางที่หุ้่นลยอนน้ำให้อาหารปลาอัตโนมัติเคลื่อนที่ไปตามจุดพิกัดละติจูด ลองจิจูด ที่กำหนดไว้ ได้ทำการทดสอบในแต่ละจุดอ้างอิงทั้งหมด 5 ครั้ง



รูปที่ 4.14 ตำแหน่งจุดอ้างอิงที่ต้องการเคลื่อนที่



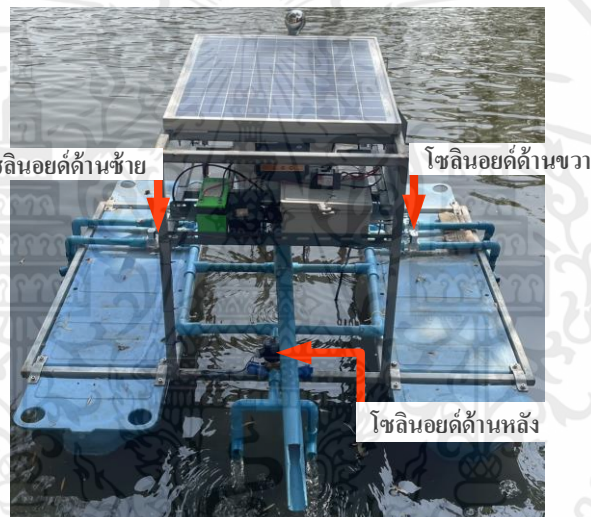
รูปที่ 4.15 การทดสอบความคลาดเคลื่อนของการเคลื่อนที่ตามจุดพิกัดละติจูด ลองจิจูด

จากการทดสอบพบว่า ค่าระยะทางที่เคลื่อนที่ได้มีความคลาดเคลื่อนไปจากตำแหน่งจริงอยู่บ้างจากคุณสมบัติทางเทคนิคของระบบ GPS ทั่วไป จากตำแหน่งจุดพิกัดละติจูด ลองจิจูด ที่กำหนดไว้ทั้ง 4 จุด พบว่าระยะความคลาดเคลื่อนสูงสุดเท่ากับ 2.90 เมตร และต่ำสุดเท่ากับ 2.24 ดังแสดงในรูปที่ 4.15

4.7 ผลการทดสอบการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

จากการทดสอบการขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยการเชื่อมต่อบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno เข้ากับวงจรขับโซลินอยด์ โดยใช้พลังงานไฟฟ้า 12 โวลต์ จ่ายให้กับโซลินอยด์วาล์วและ DC Pump แล้วทำการเชื่อมต่อแบตเตอรี่เข้ากับวงจรแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์ เพื่อเป็นแหล่งจ่ายพลังงานไฟฟ้าให้กับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno ซึ่งกำหนดให้ Arduino Uno ควบคุมการทำงานของโซลินอยด์วาล์วเพื่อขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ 3 รูปแบบ ดังนี้

1. เคลื่อนที่ไปด้านหน้า กำหนดให้โซลินอยด์วาล์วด้านหลังทำงาน แสดงดังรูปที่ 4.16 และผลการทำงานจาก Serial Monitor แสดงดังรูปที่ 4.17



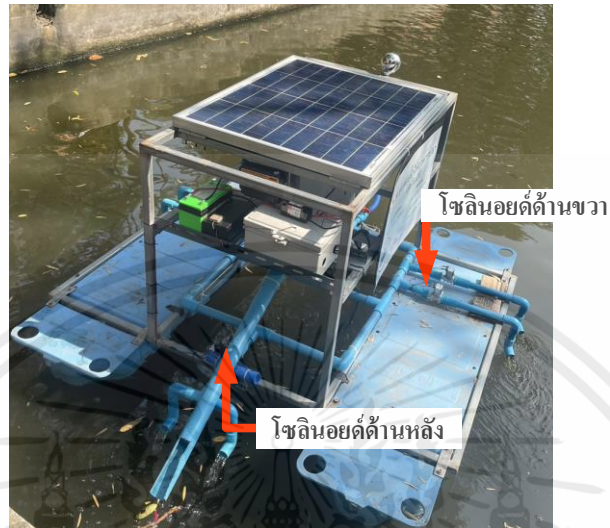
รูปที่ 4.16 ตัวอย่างการเคลื่อนที่ไปด้านหน้า

```
Output Serial Monitor ×
Message (Enter to send message to 'Arduino Uno' on 'COM9
16:07:47.197 -> Forward : Back Solenoid On
```

รูปที่ 4.17 ผลจาก Serial Monitor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เคลื่อนที่ไปทางซ้าย กำหนดให้โซลินอยด์วาล์วด้านหลังและด้านขวาทำงาน แสดงดังรูปที่ 4.18 และผลการทำงานจาก Serial Monitor แสดงดังรูปที่ 4.19



รูปที่ 4.18 ตัวอย่างการเคลื่อนที่ไปด้านซ้าย

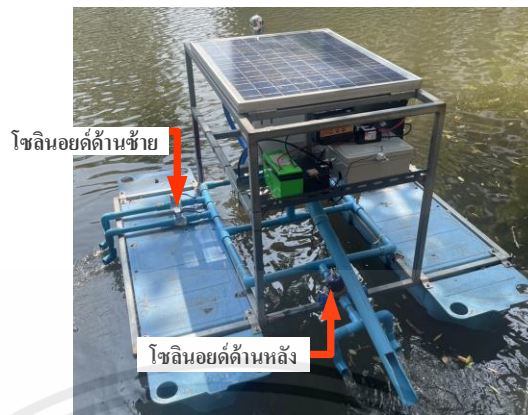
```

Output Serial Monitor x
Message (Enter to send message to 'Arduino Uno' on 'COM9')
16:09:49.671 -> Turn left : Back Solenoid On
16:09:49.671 ->
16:09:49.708 -> Right Solenoid On
  
```

รูปที่ 4.19 ผลจาก Serial Monitor

3. เคลื่อนที่ไปทางขวา กำหนดให้โซลินอยด์วาล์วด้านหลังและด้านซ้ายทำงาน แสดงดังรูปที่ 4.20 และผลการทำงานจาก Serial Monitor แสดงดังรูปที่ 4.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 ตัวอย่างการเคลื่อนที่ไปด้านขวา

```

Output Serial Monitor ×
Message (Enter to send message to 'Arduino Uno' on 'COM9')
16:11:23.424 -> Turn Right : Back Solenoid On
16:11:23.462 ->                               &&
16:11:23.462 ->                               Left Solenoid On
  
```

รูปที่ 4.21 ผลจาก Serial Monitor

ในการทดสอบได้ทำการวัดกระแสไฟเลี้ยงทั้งระบบซึ่งมีค่าเท่ากับ 0.07 แอมป์ และทำการวัดกระแสการทำงานของโซลินอยด์วาล์วเมื่อทำการสั่งการใช้งาน 1 ตัว, 2 ตัว, 3 ตัว และ 4 ตัว แบบไม่รวมกระแส DC Pump และรวมกระแส DC Pump แสดงดังตารางที่ 4.6 และ ตารางที่ 4.7

ตารางที่ 4.6 ผลการทดสอบการวัดกระแสการทำงานของโซลินอยด์วาล์วเมื่อทำการสั่งการใช้งาน 1 ตัว, 2 ตัว, 3 ตัว และ 4 ตัว แบบไม่รวมกระแส DC Pump จะแสดงในตารางดังนี้

จำนวนโซลินอยด์วาล์ว (ตัว)	กระแสไฟฟ้า (แอมป์)
1	0.55
2	0.99
3	1.46
4	1.99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 ผลการทดสอบการวัดกระแสการทำงานของโซลีนอยด์วาล์วเมื่อทำการสั่งการใช้งาน 1 ตัว, 2 ตัว, 3 ตัว และ 4 ตัว แบบรวมกระแส DC Pump จะแสดงในตารางดังนี้

จำนวนโซลีนอยด์วาล์ว (ตัว)	กระแสไฟฟ้า (แอมป์)
1	3.13
2	3.57
3	4.04
4	4.57

จากการทดสอบการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยใช้แบตเตอรี่ขนาด 12 V ความจุ 20 Ah จากนั้นทำการคำนวณระยะเวลาในการทำงาน โดยกำหนดให้ I_{bat} คือ ความจุของแบตเตอรี่ (Ah) I_{used} คือ กระแสรวมของอุปกรณ์ที่ใช้งานในหนึ่งชั่วโมง แสดงได้ดังตารางที่ 4.8 และ I_{sc} คือ กระแสไฟฟ้าที่แผงพลังงานแสงอาทิตย์ชาร์จเข้ากับแบตเตอรี่ (1.5A)

ตารางที่ 4.8 ผลการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

ลักษณะการทำงาน	เวลาในการทำงาน (h)	กระแสของอุปกรณ์ที่ใช้งาน i (A)
การทำงานของระบบรวม	1.00	4.50
การทำงานเฉพาะออกซิเจน	1.00	3.00
รวม	2.00	7.50

ระยะเวลาในการชาร์จมีค่าเท่ากับ

$$\frac{i_{bat} - i_{used}}{i_{sc}} = \frac{20 - (4.5 + 3)}{1.50} = 8.33 \text{ Hours} \quad (4.1)$$

จากสมการที่ (4.1) ต้องใช้ระยะเวลาในการชาร์จอีก 8.33 ชั่วโมง เพื่อให้แบตเตอรี่กลับมาเต็ม

4.8 ผลการทดสอบการทำงานของแอปพลิเคชัน

แอปพลิเคชัน MIT App Inventor เป็นแพลตฟอร์มที่ทางผู้จัดทำใช้ในการออกแบบแอปพลิเคชันบนระบบสมาร์ตโฟนในระบบปฏิบัติการแอนดรอยด์ (Android) โดยส่งข้อมูลผ่าน Firebase ไปยังบอร์ดไมโครคอนโทรลเลอร์ ESP-32 จากนั้นทำการส่งข้อมูลจากบอร์ดไมโครคอนโทรลเลอร์ ESP-32 ไปยังบอร์ดไมโครคอนโทรลเลอร์ Arduino ฟังก์ชันการทำงานจะแสดง

เมื่อทำการเชื่อมต่อกับระบบแอนดรอยด์ (Android) เพื่อสามารถโปรแกรมการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติรวมถึงดูสถานะของแบตเตอรี่และปริมาณอาหารปลาอีกด้วย โดยแสดงดังตัวอย่าง เช่น หน้าเลือกจุดการให้อาหารปลาจาก Google map ดังรูปที่ 4.22(ก) กำหนดเวลาการให้อาหารปลาในแต่ละจุดแสดงดังรูปที่ 4.22(ข) และสถานะคงเหลือของแบตเตอรี่ และอาหารปลาแสดงดังรูปที่ 4.22(ค)



รูปที่ 4.22 หน้าแอปพลิเคชันการทำงาน

(ก) การกำหนดจุดการให้อาหารปลาจากแผนที่ Google map (ข) การกำหนดเวลาการให้อาหารปลา (ค) แสดงสถานะของแบตเตอรี่และปริมาณอาหารปลาที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปฏิญานิพนธ์นี้มีจุดประสงค์เพื่อออกแบบและสร้างหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ โดยใช้แผงพลังงานแสงอาทิตย์และแบตเตอรี่เป็นแหล่งจ่ายพลังงานไฟฟ้าให้กับระบบการขับเคลื่อน และระบบการปล่อยอาหารปลา โดยระบบการทำงานของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ ประกอบไปด้วย 2 ส่วน คือ ส่วนของระบบควบคุมการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ และส่วนของระบบการปล่อยอาหารปลา

ส่วนของระบบควบคุมการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติจะทำงาน โดยใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino Uno เป็นตัวสั่งการควบคุมการเปิด-ปิด DC Pump และโซลินอยด์วาล์ว เพื่อให้หุ่นลอยน้ำให้อาหารปลาอัตโนมัติสามารถเคลื่อนที่ไปข้างหน้า เลี้ยวซ้าย และเลี้ยวขวาได้ โดยใช้แหล่งจ่ายไฟจากแผงพลังงานแสงอาทิตย์และแบตเตอรี่จ่ายไฟให้กับบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno, DC Pump และโซลินอยด์วาล์ว ซึ่งบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno ใช้แรงดันไฟฟ้า 5 โวลต์ ส่วน DC Pump และโซลินอยด์วาล์ว ใช้แรงดันไฟฟ้า 12 โวลต์ จากการทดสอบการทำงานของ DC Pump พบว่า เมื่อทำการวัดค่ากระแสไฟฟ้าที่ DC Pump มีค่ากระแสไฟฟ้าเท่ากับ 2.580 แอมป์ และทำการวัดค่ากระแสไฟฟ้าที่โซลินอยด์วาล์ว มีค่ากระแสไฟฟ้าเท่ากับ 0.500 แอมป์

ส่วนของระบบการปล่อยอาหารปลาใช้มอเตอร์แรงดัน 12 โวลต์ ในการหมุนสปริงเกลียวที่ติดอยู่กับมอเตอร์ เพื่อปล่อยอาหารปลา จากการคำนวณปลาชนิดควรได้รับอาหาร 1.5 กิโลกรัม/1 วัน ใน 1 วันจะให้อาหารปลา 2 ครั้ง ครั้งละ 750 กรัม โดยจะจำลองการทดสอบปล่อยอาหารปลา 4 จุด จุดละ 187.5 กรัม ซึ่งผลการทดสอบการหมุนของมอเตอร์ พบว่า เมื่อทำการวัดค่ากระแสไฟฟ้าได้เท่ากับ 0.150 แอมป์ และมีระบบจ่ายกำลังไฟฟ้าจากแผงพลังงานแสงอาทิตย์สามารถส่งงานผ่านแอปพลิเคชันบนมือถือ รวมถึงดูสถานะแบตเตอรี่ ปริมาณอาหารปลาที่เหลือ และสร้างออกซิเจนในบ่อเลี้ยงปลาได้

5.2 ข้อเสนอแนะ

1) จากการทดสอบการเคลื่อนที่ของหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ พบว่าโซลินอยด์วาล์วใช้กระแสไฟฟ้า 0.5 แอมป์ ซึ่งหุ่นลอยน้ำให้อาหารปลาอัตโนมัติใช้โซลินอยด์วาล์วทั้งหมด 6 ตัวส่งผลให้แบตเตอรี่ที่ใช้งานหมดเร็วกว่าที่คำนวณไว้เบื้องต้น

2) จากการทดสอบการทำงานของโมดูลเข็มทิศ พบว่าเมื่อนำโมดูลเข็มทิศวางไว้ใกล้กับวัสดุอุปกรณ์ที่เป็นแม่เหล็ก จะส่งผลให้ค่าองศาที่แสดงบนหน้าจอ Serial Monitor มีความคลาดเคลื่อน

3) จากการทดสอบการทำงานของโมดูลระบุตำแหน่งพิกัด GPS พบว่ามีความคลาดเคลื่อนไปจากตำแหน่งจริงอยู่บ้างจากคุณสมบัติทางเทคนิคของระบบ GPS ทั่วไป

บรรณานุกรม

- [1] nbdcthailand. “เคล็ดลับการให้อาหารปลานิล.” [ออนไลน์].
เข้าถึงได้จาก : <https://nbdcthailand.com/2021/05/31>.
- [2] กลุ่มโครงการและกิจกรรมพิเศษ กองโครงการอันเนื่องมาจากพระราชดำริและกิจกรรมพิเศษ กรมประมง. “การเลี้ยงปลานิลในกระชัง.” [ออนไลน์].
เข้าถึงได้จาก : https://www4.fisheries.go.th/local/file_document/2021_file
- [3] BYJU'S. “Archimedes Principle.” [ออนไลน์].
เข้าถึงได้จาก : <https://byjus.com/physics/archimedes-principle/>
- [4] pDragon. “ESP32 ทำงานอย่างไร Pin I/O พื้นฐาน.” [ออนไลน์].
เข้าถึงได้จาก : <https://v89infinity.com/esp32/>
- [5] ThaiEasyElec. “บทความ Arduino คืออะไร.” [ออนไลน์].
เข้าถึงได้จาก : <https://blog.thaieasyelec.com/what-is-arduino-ch1/>
- [6] The Invention. “ทำความเข้าใจบอร์ดไมโครคอนโทรลเลอร์ ARDUINO UNO R3.” [ออนไลน์].
เข้าถึงได้จาก <https://www.ai-corporation.net/2021/11/19/arduino-uno-r3/>
- [7] Fitrox Electronics. “การใช้งาน GPS Module (Ublox NEO-6M).” [ออนไลน์].
เข้าถึงได้จาก : <http://fitrox.lnwshop.com/article/59/tutorial-gps-module-ublox-neo-6m>
- [8] DatasheetHub. “GY-NEO6MV2 Flight Control GPS Module.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.datasheethub.com/gy-neo6mv2-flight-controlgpsmodule/>
- [9] OSOYOO. “3-Axis Magnetic Sensor QMC5883L.” [ออนไลน์].
เข้าถึงได้จาก : <http://wiki.sunfounder.cc/images/7/72/QMC5883L-Datasheet-1.0.pdf>
- [10] arduitrronics. “3-Axis Digital Compass Module (QMC5883L) GY-273.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.arduitronics.com/product/273/3-axis-digitalcompass-module-qmc5883l-gy-273>

- [11] สัญญา ลักษณะ. “เซลล์แสงอาทิตย์ (Solar Cell).” [ออนไลน์].
เข้าถึงได้จาก :
<https://www.prd.go.th/th/file/get/file/202103082fff21210bb97eb2ea51c14d30624891133438.pdf>
- [12] อิมเม้นซ์ คอร์ปอเรชั่น “มาทำความรู้จักชนิดต่างๆ ของแผงโซลาร์เซลล์กัน.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.gump.in.th/article/535/>
- [13] DELTA-OPTI Monika Matysiak. “SOLAR CHARGE CONTROLLER SCC-30A-PWM-LCD.” [ออนไลน์].
เข้าถึงได้จาก : [Delta-Opti Datasheet-SCC-30A-PWM-LCD \(shopdelta.eu\)](https://shopdelta.eu/Delta-Opti-Datasheet-SCC-30A-PWM-LCD)
- [14] SUNNERGY. “ทำความรู้จักกับ แบตเตอรี่ลิเธียม นวัตกรรมพลังงานแห่งอนาคต.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.sunnergysolar.com/article/362/>
- [15] เออีซีเอ็นจีเนียร์. “หลักการการทำงานของแบตเตอรี่ลิเธียมไอออน.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.changfi.com/fix/2021/10/26/lithium-ion-battery3/>
- [16] STMicroelectronics. “IRF540.” [ออนไลน์].
เข้าถึงได้จาก : [IRF540_03 STMICELECTRONICS | Alldatasheet](https://www.st.com/sg/asset/IRF540_03_STMICELECTRONICS|Alldatasheet)
- [17] Electronics Tutorials. “MOSFET as a Switch.” [ออนไลน์].
เข้าถึงได้จาก : https://www.electronics-tutorials.ws/transistor/trans_7.html
- [18] Texas Instruments Incorporated. “LM2596.” [ออนไลน์].
เข้าถึงได้จาก : <https://pdf1.alldatasheet.com/datasheet-pdf/download/524001/TI/LM2596.html>
- [19] Semiconductor Components Industries. “3.0 A, Step-Down Switching Regulator LM2596.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.onsemi.com/pdf/datasheet/lm2596-d.pdf>
- [20] arduitrronics. “DC-to-DC Step Down LM2596.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.arduitronics.com/product/570/dc-to-dc-step-down-lm2596-lm2596s-module-3a>

- [21] แฟ็คโตมาร์ท. “หลักการการทำงานของ Solenoid Valve.” [ออนไลน์].
เข้าถึงได้จาก : <https://mall.factomart.com/principle-of-solenoid-valve/>
- [22] ฟลูเทค. “โซลินอยด์วาล์ว (Solenoid Valve) คืออะไร ทำงานอย่างไร.” [ออนไลน์].
เข้าถึงได้จาก : <https://flutech.co.th/solenoid-valve-how-they-work/>
- [23] เออีซี เอ็นจิเนียริง. “ปั๊ม DC คือ.” [ออนไลน์].
เข้าถึงได้จาก : <https://www.pumpban.com/archives/9036>
- [24] Thai-AgroTech. “ปั๊มแช่ ปั๊มจุ่ม ปั๊มจุ่ม ปั๊มแช่ ไฟกระแสตรง DC submersible pump 12V/ 24V 180W ใช้กับแบตเตอรี่ 12V / ต่อตรงแผงโซลาร์เซลล์.” [ออนไลน์].
เข้าถึงได้จาก : <http://www.thaiagrotech.com/product/66-dc-submersible-pump-12v-24v-180w12v>
- [25] Candid Themes. “การทำงานของมอเตอร์ไฟตรงและการใช้งาน.” [Online]
เข้าถึงได้จาก : <https://www.inventor.in.th/home/>
- [26] Yantai Bonway. “อะไรคือความแตกต่างระหว่างมอเตอร์ dc และมอเตอร์เกียร์ dc.” [Online]
เข้าถึงได้จาก : <https://th.sogears.com/>
- [27] The Invention. “Arduino IDE.” [Online]
เข้าถึงได้จาก : <https://www.ai-corporation.net/2021/11/18/what-is-arduino-ide/#1648794579978-4ac5b2e6-420a>
- [28] i-LOADZONE. “Proteus Pro 8.16 SP3 Build 36097 โปรแกรมออกแบบ PCB.” [Online] เข้าถึงได้จาก : <https://i-loadzone.com/proteus-professional/>
- [29] Suthisa. “ทำความรู้จักกับโปรแกรมเขียนแบบ Autocad.” [Online]
เข้าถึงได้จาก : <https://news.trueid.net/detail/eLkZWONl3ON8>
- [30] Autodesk Inc. “AutoCAD - DWG Viewer & Editor.” [Online] เข้าถึงได้จาก : https://play.google.com/store/apps/details?id=com.autodesk.autocadw&hl=en_US&pli=1
- [31] ภูมินทร์ ดวงหาค้าง, พัฒน์นรี ศรีสมพันธ์, โรเบิร์ต แบทซิงเจอร์. “MIT App Inventor กับการพัฒนาเพื่อประยุกต์ใช้จริง” [Online] เข้าถึงได้จาก : [ชุมชน MIT App Inventor \(bu.ac.th\)](http://ชุมชน.MIT.App.Inventor(bu.ac.th))



ภาคผนวก ก
คำสั่งการทดสอบการทำงานของโมดูลระบุตำแหน่งพิกัด GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "TinyGPS++.h"
#include "SoftwareSerial.h"

SoftwareSerial serial_connection(3,2); //RX=pin
50, TX=pin 51

TinyGPSPlus gps;
void setup()
{
  Serial.begin(9600);
  serial_connection.begin(9600);
}

void loop()
{
  while(serial_connection.available())
  {
    gps.encode(serial_connection.read());
  }
  if(gps.location.isUpdated())
  {
    Serial.print(gps.location.lat(), 6);
    Serial.print(",");
    Serial.println(gps.location.lng(), 6);
    Serial.println("");
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <QMC5883LCompass.h>

QMC5883LCompass compass;

void setup() {
  Serial.begin(9600);
  compass.init();
}

void loop() {
  int a;

  // Read compass values
  compass.read();

  // Return Azimuth reading
  a = compass.getAzimuth();
  Serial.print("A: ");
  Serial.print(a);
  Serial.println();
  delay(250);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค

คำสั่งการทดสอบการขับเคลื่อนหุ่นลอยน้ำให้อาหารปลาอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// กำหนดขาที่จะใช้ควบคุม MOSFET
int mosfetPin1 = 13; //หน้า
int mosfetPin2 = 12; //หลัง
int mosfetPin3 = 11; //ซ้าย
int mosfetPin4 = 10; //ขวา
int mosfetPin6 = 9; //pump

void setup() {
  // กำหนดขาเป็น OUTPUT
  pinMode(mosfetPin1, OUTPUT);
  pinMode(mosfetPin2, OUTPUT);
  pinMode(mosfetPin3, OUTPUT);
  pinMode(mosfetPin4, OUTPUT);
  pinMode(mosfetPin6, OUTPUT);
}

void loop() {
  digitalWrite(mosfetPin1, HIGH);
  digitalWrite(mosfetPin2, HIGH);
  digitalWrite(mosfetPin3, HIGH);
  digitalWrite(mosfetPin4, HIGH);
  digitalWrite(mosfetPin6, HIGH);
  delay(5000);

  // ถอยหลัง
  digitalWrite(mosfetPin1, HIGH);
  digitalWrite(mosfetPin2, LOW);
  digitalWrite(mosfetPin3, LOW);
  digitalWrite(mosfetPin4, LOW);
  digitalWrite(mosfetPin6, HIGH);
  delay(10000);

  // // เดินหน้า
  digitalWrite(mosfetPin1, LOW);
  digitalWrite(mosfetPin2, HIGH);
  digitalWrite(mosfetPin3, LOW);
  digitalWrite(mosfetPin4, LOW);
  digitalWrite(mosfetPin6, HIGH);
  delay(10000);

  // // เลี้ยวขวา
  digitalWrite(mosfetPin1, LOW);
  digitalWrite(mosfetPin2, LOW);
  digitalWrite(mosfetPin3, HIGH);
  digitalWrite(mosfetPin4, LOW);
  digitalWrite(mosfetPin6, HIGH);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(10000);  
  
// // เลี้ยวซ้าย  
digitalWrite(mosfetPin1, LOW);  
digitalWrite(mosfetPin2, LOW);  
digitalWrite(mosfetPin3, LOW);  
digitalWrite(mosfetPin4, HIGH);  
digitalWrite(mosfetPin6, HIGH);  
delay(10000);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คำสั่งรับค่าจากไมโครคอนโทรลเลอร์ esp 32 และคำสั่งการทำงานของหุ่นลอยน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <Wire.h>
#include <TinyGPS++.h>
#include <QMC5883LCompass.h>
#include <SoftwareSerial.h>

int mosfetPin1 = 8; // หน้า
int mosfetPin2 = 6; // หลัง
int mosfetPin3 = 5; // ซ้าย
int mosfetPin4 = 7; // ขวา
int mosfetPin6 = 2; // Pump
int mosfetPin7 = 4; // feed
int mosfetPin7 = 3; // feed
SoftwareSerial gpsSerial(13, 12);

//StaticJsonDocument<1024> doc;
TinyGPSPlus gps;
QMC5883LCompass compass;
String currentPoint;
double LAT1;
double LONG1;
double LAT2;
double LONG2;
double LAT3;
double LONG3;
double LAT4;
double LONG4;
double LAT21;
double LONG21;
double LAT22;
double LONG22;
double LAT31;
double LONG31;
double LAT32;
double LONG32;
double LAT33;
double LONG33;
double LAT41;
double LONG41;
double LAT42;
double LONG42;
double LAT43;
double LONG43;
double LAT44;
double LONG44;
enum State {
    INIT,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    MOVE_TO_POINT1,
    MOVE_TO_POINT2,
    MOVE_TO_POINT3,
    MOVE_TO_POINT4,
    STOP
};
State currentState = INIT;
void transitionTo(State nextState) {
    currentState = nextState;
    stopMotors();
}
void updateMotors(double distance, double targetCourse,
double heading) {
    if (heading < 0) {
        heading += 360;
    }

    // คำนวณความต่างของทิศทางระหว่างจุดปัจจุบันและจุดตามทิศทางที่ต้องการไป
    double angleDiff = targetCourse - heading;

    // ปรับความต่างทิศทางให้อยู่ในช่วง -180 ถึง 180
    if (angleDiff > 180) {
        angleDiff -= 360;
    } else if (angleDiff < -180) {
        angleDiff += 360;
    } // ... แบบเดียวกับเดิม ...

    if (distance <= 1) {
        stopMotorsafeed(); // หยุดเมื่อระยะห่างเป็น 0
    } else if (angleDiff < -15) {
        turnLeft();
    } else if (angleDiff > 15) {
        turnRight();
    } else {
        goForward();
    }
}

void setup() {
    Serial.begin(9600);
    pinMode(mosfetPin1, OUTPUT);
    pinMode(mosfetPin2, OUTPUT);
    pinMode(mosfetPin3, OUTPUT);
    pinMode(mosfetPin4, OUTPUT);
    pinMode(mosfetPin6, OUTPUT);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pinMode(mosfetPin7, OUTPUT);
pinMode(mosfetPin8, OUTPUT);
compass.init();
compass.setCalibrationOffsets(-757.00, 685.00, 1133.00);
compass.setCalibrationScales(0.88, 1.24, 0.95);

gpsSerial.begin(9600); // เริ่มต้นการสื่อสารที่ความเร็ว 9600 bps
}
void goForward() {
digitalWrite(mosfetPin1, LOW);
digitalWrite(mosfetPin2, HIGH);
digitalWrite(mosfetPin3, LOW);
digitalWrite(mosfetPin4, LOW);
digitalWrite(mosfetPin6, HIGH);
digitalWrite(mosfetPin7, LOW);
digitalWrite(mosfetPin8, LOW);
}

void turnLeft() {
digitalWrite(mosfetPin1, LOW);
digitalWrite(mosfetPin2, HIGH);
digitalWrite(mosfetPin3, LOW);
digitalWrite(mosfetPin4, HIGH);
digitalWrite(mosfetPin6, HIGH);
digitalWrite(mosfetPin7, LOW);
digitalWrite(mosfetPin8, LOW);
}

void turnRight() {
digitalWrite(mosfetPin1, LOW);
digitalWrite(mosfetPin2, HIGH);
digitalWrite(mosfetPin3, HIGH);
digitalWrite(mosfetPin4, LOW);
digitalWrite(mosfetPin6, HIGH);
digitalWrite(mosfetPin7, LOW);
digitalWrite(mosfetPin8, LOW);
}

void stopMotors() {
digitalWrite(mosfetPin1, LOW);
digitalWrite(mosfetPin2, LOW);
digitalWrite(mosfetPin3, LOW);
digitalWrite(mosfetPin4, LOW);
digitalWrite(mosfetPin6, LOW);
digitalWrite(mosfetPin7, LOW);
digitalWrite(mosfetPin8, LOW);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay(5000);
}

void stopMotorsafeed() {
    digitalWrite(mosfetPin1, LOW);
    digitalWrite(mosfetPin2, LOW);
    digitalWrite(mosfetPin3, LOW);
    digitalWrite(mosfetPin4, LOW);
    digitalWrite(mosfetPin6, LOW);
    digitalWrite(mosfetPin7, HIGH);
    digitalWrite(mosfetPin8, HIGH);
    delay(30000);
}

void loop() {
    compass.read();
    if (Serial.available() > 0) {
        String data = Serial.readStringUntil('\n');
        int index = 0;
        int start = 0;
        for (int i = 0; i < data.length(); i++) {
            if (data.charAt(i) == ',') {
                if (index == 0) {
                    currentPoint = data.substring(start, i);
                } else if (index == 1) {
                    LAT1 = data.substring(start, i).toDouble();
                } else if (index == 2) {
                    LONG1 = data.substring(start, i).toDouble();
                } else if (index == 3) {
                    LAT2 = data.substring(start, i).toDouble();
                } else if (index == 4) {
                    LONG2 = data.substring(start, i).toDouble();
                } else if (index == 5) {
                    LAT3 = data.substring(start, i).toDouble();
                } else if (index == 6) {
                    LONG3 = data.substring(start, i).toDouble();
                } else if (index == 7) {
                    LAT4 = data.substring(start, i).toDouble();
                } else if (index == 8) {
                    LONG4 = data.substring(start, i).toDouble();
                }
                start = i + 1;
                index++;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// เก็บข้อมูลที่อยู่หลังตัวคั่นสุดท้าย
if (index == 0) {
    currentPoint = data.substring(start);
} else if (index == 1) {
    LAT1 = data.substring(start).toDouble();
} else if (index == 2) {
    LONG1 = data.substring(start).toDouble();
} else if (index == 3) {
    LAT2 = data.substring(start).toDouble();
} else if (index == 4) {
    LONG2 = data.substring(start).toDouble();
} else if (index == 5) {
    LAT3 = data.substring(start).toDouble();
} else if (index == 6) {
    LONG3 = data.substring(start).toDouble();
} else if (index == 7) {
    LAT4 = data.substring(start).toDouble();
} else if (index == 8) {
    LONG4 = data.substring(start).toDouble();
}

if (currentPoint == "P1") {
    Serial.println("case1");
while (gpsSerial.available() > 0) {
    if (gps.encode(gpsSerial.read())) {
        Serial.print("LAT1: ");
        Serial.print(LAT1, 6);
        Serial.print(", LONG1: ");
        Serial.println(LONG1, 6);

switch (currentState) {
case INIT:
    transitionTo(MOVE_TO_POINT1);
    break;

case MOVE_TO_POINT1:
    double currentLat = gps.location.lat();
    double currentLong = gps.location.lng();
    Serial.println("รับค่าgpsเรียบร้อยแล้ว");
    double distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT1, LONG1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        double courseToTarget =
TinyGPSPlus::courseTo(currentLat, currentLong, LAT1,
LONG1);
        double heading = compass.getAzimuth();
        Serial.println("พร้อมเคลื่อนที่");

        if (distanceToTarget <= 1.0) {
            stopMotorsafeed();
            transitionTo(STOP);
        } else {
            updateMotors(distanceToTarget, courseToTarget,
heading);
        }
        break;

case STOP:
    stopMotors();
    break;
}
}
}

else if (currentPoint == "P2") {
    while (gpsSerial.available() > 0) {
        if (gps.encode(gpsSerial.read())) {
            Serial.print("LAT21: ");
            Serial.print(LAT1, 6);
            Serial.print(", LONG21: ");
            Serial.println(LONG1, 6);
            Serial.print("LAT22: ");
            Serial.print(LAT2, 6);
            Serial.print(", LONG22: ");
            Serial.println(LONG2, 6);
        }
    }

switch (currentState) {
case INIT:
    transitionTo(MOVE_TO_POINT1);
    break;

case MOVE_TO_POINT1:
    double currentLat = gps.location.lat();
    double currentLong = gps.location.lng();
    Serial.println("รับค่าgpsเรียบร้อยแล้ว");
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        double distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT21, LONG21);
        double courseToTarget =
TinyGPSPlus::courseTo(currentLat, currentLong, LAT21,
LONG21);
        double heading = compass.getAzimuth();
        Serial.println("พร้อมเคลื่อนที่");

        if (distanceToTarget <= 1.0) {
            stopMotorsafeed();
            transitionTo(MOVE_TO_POINT2);
        } else {
            updateMotors(distanceToTarget, courseToTarget,
heading);
        }
        break;

        case MOVE_TO_POINT2:
            currentLat = gps.location.lat();
            currentLong = gps.location.lng();
            Serial.println("รับค่าgpsเรียบร้อยแล้ว");
            distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT22, LONG22);
            courseToTarget = TinyGPSPlus::courseTo(currentLat,
currentLong, LAT22, LONG22);
            heading = compass.getAzimuth();
            Serial.println("พร้อมเคลื่อนที่");

            if (distanceToTarget <= 1.0) {
                stopMotorsafeed();
                transitionTo(STOP);
            } else {
                updateMotors(distanceToTarget, courseToTarget,
heading);
            }
            break;

        case STOP:
            stopMotors();
            break;
    }
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if (currentPoint == "P3") {
    while (gpsSerial.available() > 0) {
        if (gps.encode(gpsSerial.read())) {
            Serial.print("LAT31: ");
            Serial.print(LAT1, 6);
            Serial.print(", LONG31: ");
            Serial.println(LONG1, 6);
            Serial.print("LAT32: ");
            Serial.print(LAT2, 6);
            Serial.print(", LONG32: ");
            Serial.println(LONG2, 6);
            Serial.print("LAT33: ");
            Serial.print(LAT3, 6);
            Serial.print(", LONG33: ");
            Serial.println(LONG3, 6);
            Serial.println("CASE3");
        }
    }

    switch (currentState) {
        case INIT:
            transitionTo(MOVE_TO_POINT1);
            break;

        case MOVE_TO_POINT1:
            double currentLat = gps.location.lat();
            double currentLong = gps.location.lng();
            Serial.println("รับค่าgpsเรียบร้อยแล้ว");
            double distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT31, LONG31);
            double courseToTarget =
TinyGPSPlus::courseTo(currentLat, currentLong, LAT31,
LONG31);
            double heading = compass.getAzimuth();
            Serial.println("พร้อมเคลื่อนที่");

            if (distanceToTarget <= 1.0) {
                stopMotorsafeed();
                transitionTo(MOVE_TO_POINT2);
            } else {
                updateMotors(distanceToTarget, courseToTarget,
heading);
            }
            break;

        case MOVE_TO_POINT2:
            currentLat = gps.location.lat();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        currentLong = gps.location.lng();
        Serial.println("รับค่าgpsเรียบร้อยแล้ว");
        distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT32, LONG32);
        courseToTarget = TinyGPSPlus::courseTo(currentLat,
currentLong, LAT32, LONG32);
        heading = compass.getAzimuth();
        Serial.println("พร้อมเคลื่อนที่");

        if (distanceToTarget <= 1.0) {
            stopMotorsafeed();
            transitionTo(MOVE_TO_POINT3);
        } else {
            updateMotors(distanceToTarget, courseToTarget,
heading);
        }
        break;

    case MOVE_TO_POINT3:
        currentLat = gps.location.lat();
        currentLong = gps.location.lng();
        Serial.println("รับค่าgpsเรียบร้อยแล้ว");
        distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT33, LONG33);
        courseToTarget = TinyGPSPlus::courseTo(currentLat,
currentLong, LAT33, LONG33);
        heading = compass.getAzimuth();
        Serial.println("พร้อมเคลื่อนที่");

        if (distanceToTarget <= 1.0) {
            stopMotorsafeed();
            transitionTo(STOP);
        } else {
            updateMotors(distanceToTarget, courseToTarget,
heading);
        }
        break;

    case STOP:
        stopMotors();
        break;
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else if (currentPoint == "P4") {
    while (gpsSerial.available() > 0) {
        if (gps.encode(gpsSerial.read())) {
Serial.print("LAT41: ");
        Serial.print(LAT1, 6);
        Serial.print(", LONG41: ");
        Serial.println(LONG1, 6);
        Serial.print("LAT42: ");
        Serial.print(LAT2, 6);
        Serial.print(", LONG42: ");
        Serial.println(LONG2, 6);
        Serial.print("LAT43: ");
        Serial.print(LAT3 , 6);
        Serial.print(", LONG43: ");
        Serial.println(LONG3, 6);
        Serial.print("LAT44: ");
        Serial.print(LAT4 , 6);
        Serial.print(", LONG44: ");
        Serial.println(LONG4, 6);
switch (currentState) {
    case INIT:
        transitionTo(MOVE_TO_POINT1);
        break;

    case MOVE_TO_POINT1:
        double currentLat = gps.location.lat();
        double currentLong = gps.location.lng();
        Serial.println("รับค่าgpsเรียบร้อยแล้ว");
        double distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT41, LONG41);
        double courseToTarget =
TinyGPSPlus::courseTo(currentLat, currentLong, LAT41,
LONG41);
        double heading = compass.getAzimuth();
        Serial.println("พร้อมเคลื่อนที่");

        if (distanceToTarget <= 1.0) {
            stopMotorsafeed();
            transitionTo(MOVE_TO_POINT2);
        } else {
            updateMotors(distanceToTarget, courseToTarget,
heading);
        }
        break;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case MOVE_TO_POINT2:
    currentLat = gps.location.lat();
    currentLong = gps.location.lng();
    Serial.println("รับค่าgpsเรียบร้อยแล้ว");
    distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT42, LONG42);
    courseToTarget = TinyGPSPlus::courseTo(currentLat,
currentLong, LAT42, LONG42);
    heading = compass.getAzimuth();
    Serial.println("พร้อมเคลื่อนที่");

    if (distanceToTarget <= 1.0) {
        stopMotorsafeed();
        transitionTo(MOVE_TO_POINT3);
    } else {
        updateMotors(distanceToTarget, courseToTarget,
heading);
    }
    break;

case MOVE_TO_POINT3:
    currentLat = gps.location.lat();
    currentLong = gps.location.lng();
    Serial.println("รับค่าgpsเรียบร้อยแล้ว");
    distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT43, LONG43);
    courseToTarget = TinyGPSPlus::courseTo(currentLat,
currentLong, LAT43, LONG43);
    heading = compass.getAzimuth();
    Serial.println("พร้อมเคลื่อนที่");

    if (distanceToTarget <= 1.0) {
        stopMotorsafeed();
        transitionTo(MOVE_TO_POINT4);
    } else {
        updateMotors(distanceToTarget, courseToTarget,
heading);
    }
    break;

case MOVE_TO_POINT4:
    currentLat = gps.location.lat();
    currentLong = gps.location.lng();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println("รับค่าgpsเรียบร้อยแล้ว");
distanceToTarget =
TinyGPSPlus::distanceBetween(currentLat, currentLong,
LAT44, LONG44);
courseToTarget = TinyGPSPlus::courseTo(currentLat,
currentLong, LAT44, LONG44);
heading = compass.getAzimuth();
Serial.println("พร้อมเคลื่อนที่");

if (distanceToTarget <= 1.0) {
stopMotorsafeed();
transitionTo(STOP);
} else {
updateMotors(distanceToTarget, courseToTarget,
heading);
}
break;

case STOP:
stopMotors();
break;
}
}
}

}

delay(1000);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <Arduino.h>

#if defined(ESP32)
  #include <WiFi.h>
#elif defined(ESP8266)
  #include <ESP8266WiFi.h>
#endif
#include <Firebase_ESP_Client.h>

//Provide the token generation process info.
#include "addons/TokenHelper.h"
//Provide the RTDB payload printing info and other helper
functions.
#include "addons/RTDBHelper.h"

// Insert your network credentials
#define WIFI_SSID "vivo 1819"
#define WIFI_PASSWORD "1212312121"
// Insert Firebase project API Key
#define API_KEY "AIzaSyAbI4-PmALZXOMMsZWWrRSmIEZUt9nqfHg"
// Insert RTDB URLdefine the RTDB URL */
#define DATABASE_URL "https://esp32-40254-default-
rtdb.firebaseio.com/"

//Define Firebase Data object
FirebaseData fbdo;

FirebaseAuth auth;
FirebaseConfig config;

unsigned long sendDataPrevMillis = 0;
String LAT1;
String point;
String LONG1;
String LAT21;
String LONG21;
String LAT22;
String LONG22;
String LAT31;
String LONG31;
String LAT32;
String LONG32;
String LAT33;
String LONG33;
String LAT41;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

String LONG41;
String LAT42;
String LONG42;
String LAT43;
String LONG43;
String LAT44;
String LONG44;
bool signupOK = false;
String cleanedLAT1;
String cleanedLONG1;
String cleanedpoint;
String cleanedLAT21;
String cleanedLONG21;
String cleanedLAT22;
String cleanedLONG22;
String cleanedLAT31;
String cleanedLONG31;
String cleanedLAT32;
String cleanedLONG32;
String cleanedLAT33;
String cleanedLONG33;
String cleanedLAT41;
String cleanedLONG41;
String cleanedLAT42;
String cleanedLONG42;
String cleanedLAT43;
String cleanedLONG43;
String cleanedLAT44;
String cleanedLONG44;
void setup() {
  Serial.begin(115200);
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("Connecting to Wi-Fi");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(300);
  }
  Serial.println();
  Serial.print("Connected with IP: ");
  Serial.println(WiFi.localIP());
  Serial.println();

  /* Assign the api key (required) */
  config.api_key = API_KEY;

  /* Assign the RTDB URL (required) */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

config.database_url = DATABASE_URL;

/* Sign up */
if (Firebase.signUp(&config, &auth, "", "")) {
  Serial.println("ok");
  signupOK = true;
}
else {
  Serial.printf("%s\n",
config.signer.signupError.message.c_str());
}

/* Assign the callback function for the long running
token generation task */
config.token_status_callback = tokenStatusCallback;
//see addons/TokenHelper.h

Firebase.begin(&config, &auth);
Firebase.reconnectWiFi(true);
}

void loop() {
  if (Firebase.ready() && signupOK && (millis() -
sendDataPrevMillis > 15000 || sendDataPrevMillis == 0)) {
    sendDataPrevMillis = millis();
    if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/point"))
    {
      if (fbdo.dataType() == "string") {
        point = fbdo.stringData();
        cleanedpoint = point.substring(2, point.length() -
2);
        //Serial.println(cleanedpoint);
        // StaticJsonDocument<1024> doc;
        // doc["point"] = cleanedpoint;
        // String output;
        // serializeJson(doc, output);

        // // ส่งข้อมูล JSON ไปยัง Arduino
        // Serial.println(output);

        delay(1000); // หนึ่งเ
      }
    }
    else {
      Serial.println(fbdo.errorReason());
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (cleanedpoint == "P1") {
    if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LAT1"))
    {
        if (fbdo.dataType() == "string") {
            LAT1 = fbdo.stringData();
            cleanedLAT1 = LAT1.substring(2, LAT1.length() -
2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LONG1")) {
        if (fbdo.dataType() == "string") {
            LONG1 = fbdo.stringData();
            cleanedLONG1 = LONG1.substring(2, LONG1.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }
    Serial.print(cleanedpoint);
    Serial.print(",");
    //delay(1000);
    Serial.print(cleanedLAT1);
    Serial.print(",");
    //delay(1000);
    Serial.println(cleanedLONG1);
    //delay(1000);

    delay(1000); // หน่วงเวลา 1 วินาที
}
else if (cleanedpoint == "P2") {
    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LAT21")) {
        if (fbdo.dataType() == "string") {
            LAT21 = fbdo.stringData();
            cleanedLAT21 = LAT21.substring(2, LAT21.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LONG21")) {
        if (fbdo.dataType() == "string") {
            LONG21 = fbdo.stringData();
            cleanedLONG21 = LONG21.substring(2,
LONG21.length() - 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LAT22")) {
        if (fbdo.dataType() == "string") {
            LAT22 = fbdo.stringData();
            cleanedLAT22 = LAT22.substring(2, LAT22.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LONG22")) {
        if (fbdo.dataType() == "string") {
            LONG22 = fbdo.stringData();
            cleanedLONG22 = LONG22.substring(2,
LONG22.length() - 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    Serial.print(cleanedpoint);
    Serial.print(",");
    delay(1000);
    Serial.print(cleanedLAT21);
    Serial.print(",");
    Serial.print(cleanedLONG21);
    Serial.print(",");
    delay(1000);
    Serial.print(cleanedLAT22);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.print(",");
Serial.print(cleanedLONG22);
delay(1000);

    delay(1000); // หน่วงเวลา 1 วินาที
}
else if (cleanedpoint == "P3") {
    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LAT31")) {
        if (fbdo.dataType() == "string") {
            LAT31 = fbdo.stringData();
            cleanedLAT31 = LAT31.substring(2, LAT31.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LONG31")) {
        if (fbdo.dataType() == "string") {
            LONG31 = fbdo.stringData();
            cleanedLONG31 = LONG31.substring(2,
LONG31.length() - 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LAT32")) {
        if (fbdo.dataType() == "string") {
            LAT32 = fbdo.stringData();
            cleanedLAT32 = LAT32.substring(2, LAT32.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LONG32")) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (fbdo.dataType() == "string") {
            LONG32 = fbdo.stringData();
            cleanedLONG32 = LONG32.substring(2,
LONG32.length() - 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LAT33")) {
        if (fbdo.dataType() == "string") {
            LAT33 = fbdo.stringData();
            cleanedLAT33 = LAT33.substring(2, LAT33.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo,
"/MIT_APP/LONG33")) {
        if (fbdo.dataType() == "string") {
            LONG33 = fbdo.stringData();
            cleanedLONG33 = LONG33.substring(2,
LONG33.length() - 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }
    Serial.print(cleanedpoint);
    Serial.print(",");
    delay(1500);
    Serial.print(cleanedLAT31);
    Serial.print(",");
    Serial.print(cleanedLONG31);
    Serial.print(",");
    delay(1500);
    Serial.print(cleanedLAT32);
    Serial.print(",");
    Serial.print(cleanedLONG32);
    Serial.print(",");
    delay(1500);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Serial.print(cleanedLAT33);
        Serial.print(",");
        Serial.print(cleanedLONG33);
        delay(1500);
        delay(1000); // หน่วงเวลา 1 วินาที
    }
    else if (cleanedpoint == "P4") {
    if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LAT41")) {
        if (fbdo.dataType() == "string") {
            LAT41 = fbdo.stringData();
            cleanedLAT41 = LAT41.substring(2, LAT41.length() -
2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LONG41")) {
        if (fbdo.dataType() == "string") {
            LONG41 = fbdo.stringData();
            cleanedLONG41 = LONG41.substring(2, LONG41.length()
- 2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LAT42")) {
        if (fbdo.dataType() == "string") {
            LAT42 = fbdo.stringData();
            cleanedLAT42 = LAT42.substring(2, LAT42.length() -
2);
        }
    }
    else {
        Serial.println(fbdo.errorReason());
    }

    if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LONG42")) {
        if (fbdo.dataType() == "string") {
            LONG42 = fbdo.stringData();
            cleanedLONG42 = LONG42.substring(2, LONG42.length()
- 2);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else {
    Serial.println(fbdo.errorReason());
}

if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LAT43")) {
    if (fbdo.dataType() == "string") {
        LAT43 = fbdo.stringData();
        cleanedLAT43 = LAT43.substring(2, LAT43.length() -
2);
    }
}
else {
    Serial.println(fbdo.errorReason());
}

if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LONG43")) {
    if (fbdo.dataType() == "string") {
        LONG43 = fbdo.stringData();
        cleanedLONG43 = LONG43.substring(2, LONG43.length()
- 2);
    }
}
else {
    Serial.println(fbdo.errorReason());
}

if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LAT44")) {
    if (fbdo.dataType() == "string") {
        LAT44 = fbdo.stringData();
        cleanedLAT44 = LAT44.substring(2, LAT44.length() -
2);
    }
}
else {
    Serial.println(fbdo.errorReason());
}

if (Firebase.RTDB.getString(&fbdo, "/MIT_APP/LONG44")) {
    if (fbdo.dataType() == "string") {
        LONG44 = fbdo.stringData();
        cleanedLONG44 = LONG44.substring(2, LONG44.length()
- 2);
    }
}
else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Serial.println(fbdo.errorReason());
}
    Serial.print(cleanedpoint);
    Serial.print(",");
    delay(1500);
    Serial.print(cleanedLAT41);
    Serial.print(",");
    Serial.print(cleanedLONG41);
    Serial.print(",");
    delay(1500);
    Serial.print(cleanedLAT42);
    Serial.print(",");
    Serial.print(cleanedLONG42);
    Serial.print(",");
    delay(1500);
    Serial.print(cleanedLAT43);
    Serial.print(",");
    Serial.print(cleanedLONG43);
    Serial.print(",");
    delay(1500);
    Serial.print(cleanedLAT44);
    Serial.print(",");
    Serial.print(cleanedLONG44);
    delay(1500); // หน่วงเวลา 1 วินาที
    delay(1000); // หน่วงเวลา 1 วินาที
}
}
delay(100);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้