



เครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน

AUTOMATIC LAWN MOWER USING GPS CONTROLLER VIA APPLICATION



นันทวัฒน์ เหมวงศ์

ปาณัท สุขเผือก

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

ภาควิชาวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วิทยาเขตชุมพรเขตรอุดมศักดิ์ จังหวัดชุมพร

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2021

DEPARTMENT OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

PRINCE OF CHUMPHON

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รับที่...../.....

งานทะเบียนและประมวลผล

ปริญญาโทปีการศึกษา 2563

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วิทยาเขตชุมพรเขตรอุดมศักดิ์ จังหวัดชุมพร

เรื่อง เครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน

Automatic lawn mower using GPS controlled via application

ผู้จัดทำ

1. นายนันท์วัฒน์ เหมวงศ์ รหัสนักศึกษา 60511056

2. นายปานท์ สุขเผือก รหัสนักศึกษา 60511057

.....อาจารย์ที่ปรึกษา

(ว่าที่ ร.ต.ศิลา ศิริมาสกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อปริญญาบัตร	เครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน	
นักศึกษา	นายณัฐวัฒน์ เหมวงศ์	รหัสประจำตัว 60511056
	นายปานัท สุขเผือก	รหัสประจำตัว 60511057
อาจารย์ที่ปรึกษา	ว่าที่ ร.ต.ศิวา ศิริมาสกุล	
หลักสูตร	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมอิเล็กทรอนิกส์	
ปีการศึกษา	2563	

## บทคัดย่อ

ปริญญาบัตรฉบับนี้นำเสนอเครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน สร้างขึ้นมาเพื่อใช้ตัดหญ้าในสนามที่มีขนาดใหญ่อย่าง สนามฟุตบอล เพื่อความสะดวกสบายต่อ ผู้ใช้งานมากขึ้น โดยจะศึกษาเกี่ยวกับการทำงานของจีพีเอส การสร้างเครื่องตัดหญ้าสนามที่ควบคุมด้วยไมโครคอนโทรลเลอร์ การนำมาประยุกต์ใช้งานในด้านการนำทาง และการนำมาใช้งานในด้านการเกษตร โดยการทำงานของเครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน จะต้องประกอบไปด้วย แอปพลิเคชัน Thunkable บนโทรศัพท์มือถือเพื่อสั่งการให้เครื่องตัดหญ้าสามารถทำงานได้ ภายในแอปพลิเคชันจะประกอบไปด้วย ระบบอัตโนมัติ เพอร์เซ็นต์แบตเตอรี่ ระบุตำแหน่งตำแหน่งของเครื่องตัดหญ้า การเปิดปิดมอเตอร์ตัดหญ้า และระบบบังคับ ผลการทดลองพบว่า การเคลื่อนที่ของเครื่องตัดหญ้าในระยะทาง 50 เมตร จำนวน 10 ครั้ง มีความเร็วเฉลี่ยที่ 0.411 เมตรต่อวินาที และในการตัดหญ้า 100 ตารางเมตร จำนวน 5 ครั้ง พบว่าเปอร์เซ็นต์แบตเตอรี่ที่ใช้ไปเฉลี่ย 52.5% และใช้เวลาเฉลี่ยอยู่ที่ 13.51 นาที สำหรับการทดลองความคลาดเคลื่อนของจีพีเอสในระยะ 10 เมตร 20 เมตร และ 30 เมตร มีความคลาดเคลื่อนที่ 3.2% 2.58% และ 1.34% ตามลำดับ พบว่าระยะทางที่ 30 เมตร มีความคลาดเคลื่อนของจีพีเอสน้อยกว่าระยะที่ 10 เมตร และ 20 เมตร

<b>Project</b>	Automatic lawn mower using GPS controlled via application	
<b>Student</b>	Mr. Nanthawat Hemwong	Student ID 60511056
	Mr. Panat Sukpuak	Student ID 60511057
<b>Advisor</b>	ACTING SUB LT. Silar Sirimasakul	
<b>Degree</b>	Bachelor of Engineering	
<b>Program in</b>	Electronics Engineering	
<b>Academic Year</b>	2020	

## ABSTRACT

This project presents an automatic lawn mower using GPS controlled via application. Built to mow large fields such as football fields for further convenience. More users will study the operation of GPS. Building a Microcontroller-Controlled Lawn Mower application in navigation and its use in agriculture by the operation of the automatic lawn mower using GPS controlled through the application. It must include the Thunkable mobile application in order to operate the mower. Within the application will consist of automation battery percentage Specify the location of the mower. Turning the mower motor on and off and a compulsory system, the results showed that the movement of the mower at a distance of 50 meters is 10 times, with average speed of 0.411 meters per second. And in 5 times of mowing 100 square meters, the average battery percentage was 52.5% and the average time was 13.51 minutes. The discrepancy was 3.2%, 2.58% and 1.34%, respectively. It was found that distances at 30 meters had fewer GPS errors than distances of 10 meters and 20 met

## กิตติกรรมประกาศ

คณะผู้จัดทำโครงการขอขอบพระคุณ ว่าที่ ร.ต.ศิลา ศิริมาสกุล อาจารย์ที่ปรึกษาปริญญาโทนิพนธ์ ภาควิชาวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพรเขตรอุดมศักดิ์ ที่ให้คำปรึกษาแนะนำ และแก้ไขข้อบกพร่องต่างๆ ด้วยความเอาใจใส่เป็นอย่างสูงทำให้เล่มปริญญาโทนิพนธ์ เล่มนี้สมบูรณ์ที่สุด

ทางคณะผู้จัดทำขอขอบคุณทางสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพรเขตรอุดมศักดิ์ ที่ได้ให้การสนับสนุนทางด้าน ทุนทรัพย์ อุปกรณ์ เครื่องมือต่างๆ และรวมไปถึงอาคารสถานที่ที่ใช้ในการดำเนินงานและโรงเรียนบ้านหินกบที่ให้ใช้สถานที่ปฏิบัติงานจนสำเร็จสุดท้ายนี้ทางคณะผู้จัดทำ

ขอขอบพระคุณบุคคลสำคัญที่สุด คือ บิดามารดาอันเป็นที่เคารพ โดยให้โอกาสในการศึกษาเล่าเรียน และสนับสนุนทุนทรัพย์ตลอดมาในการศึกษา เพื่อนร่วมชั้นเรียน ตลอดจนผู้ที่มีส่วนเกี่ยวข้องที่คอยสนับสนุน และให้กำลังใจเสมอมา

นันทวัฒน์ เหมวงศ์

ปานท์ สุขเผือก

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของงาน	2
1.3 สมมติฐานของการศึกษา	2
1.4 ขอบเขตของการศึกษา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 ขั้นตอนและวิธีการดำเนินงาน	3
1.7 โครงสร้างปริญญานิพนธ์	5
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	6
2.1 ความหมายของเครื่องตัดหญ้า	6
2.2 บอร์ดควบคุม (control board)	8
2.3 รีเลย์ (Relay)	10
2.4 โมดูลเรกูเลเตอร์ (DC-to-DC Step Down)	11
2.5 แบตเตอรี่ (Battery)	12
2.6 มอเตอร์กระแสตรง (DC Motor)	12
2.7 ไฟร์เบส (Firebase)	14
2.8 เซอร์โวมอเตอร์ (Servo Motor)	15
2.9 ทฤษฎีบทพีทาโกรัส (Pythagoras's Theorem)	17
2.10 จีพีเอส (GPS)	18
2.11 แอปพลิเคชัน (Thunkable)	21

## สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบ	24
3.1 การออกแบบ	24
3.2 การคำนวณหาแบตเตอรี่ที่ใช้ในรถเครื่องตัดหญ้า	25
3.3 วงจรควบคุมมอเตอร์ขับเคลื่อน	26
3.4 ออกแบบวงจรขับเคลื่อน	28
3.5 วงจรขับมอเตอร์ใบตัด	28
3.6 โครงสร้างของเครื่องตัดหญ้า	30
3.7 หาเส้นทางในการตัดโดยกรอกพิกัด 4 จุด	30
3.8 การลดความเร็วรอบของมอเตอร์ขับเคลื่อน	38
3.9 การคำนวณเปอร์เซ็นต์แบตเตอรี่จากแรงดันแบตเตอรี่	38
3.10 การออกแบบ แอปพลิเคชันแอนดรอยด์ (Application android) โดยใช้ Thunkable ในการเขียน	39
3.11 ตัวกลางในการรับส่งค่า	44
บทที่ 4 ผลการทดลอง	45
4.1 ทดสอบความเร็วของเครื่องตัดหญ้าที่ระยะทาง 50 เมตร	45
4.2 ทดสอบความเร็วในการเคลื่อนที่ 20 ตารางเมตร	46
4.3 ทดสอบการตัดหญ้า 20 ตารางเมตร	47
4.4 ทดสอบการตัดหญ้า 100 ตารางเมตร	48
4.5 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 10 เมตร	50
4.6 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 20 เมตร	51
4.7 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 30 เมตร	52
บทที่ 5 วิเคราะห์และสรุปผลการทดลอง	53
5.1 สรุปผลการทดลอง	53
5.2 ปัญหาและอุปสรรค	53
5.3 ข้อเสนอแนะ	53

## สารบัญ (ต่อ)

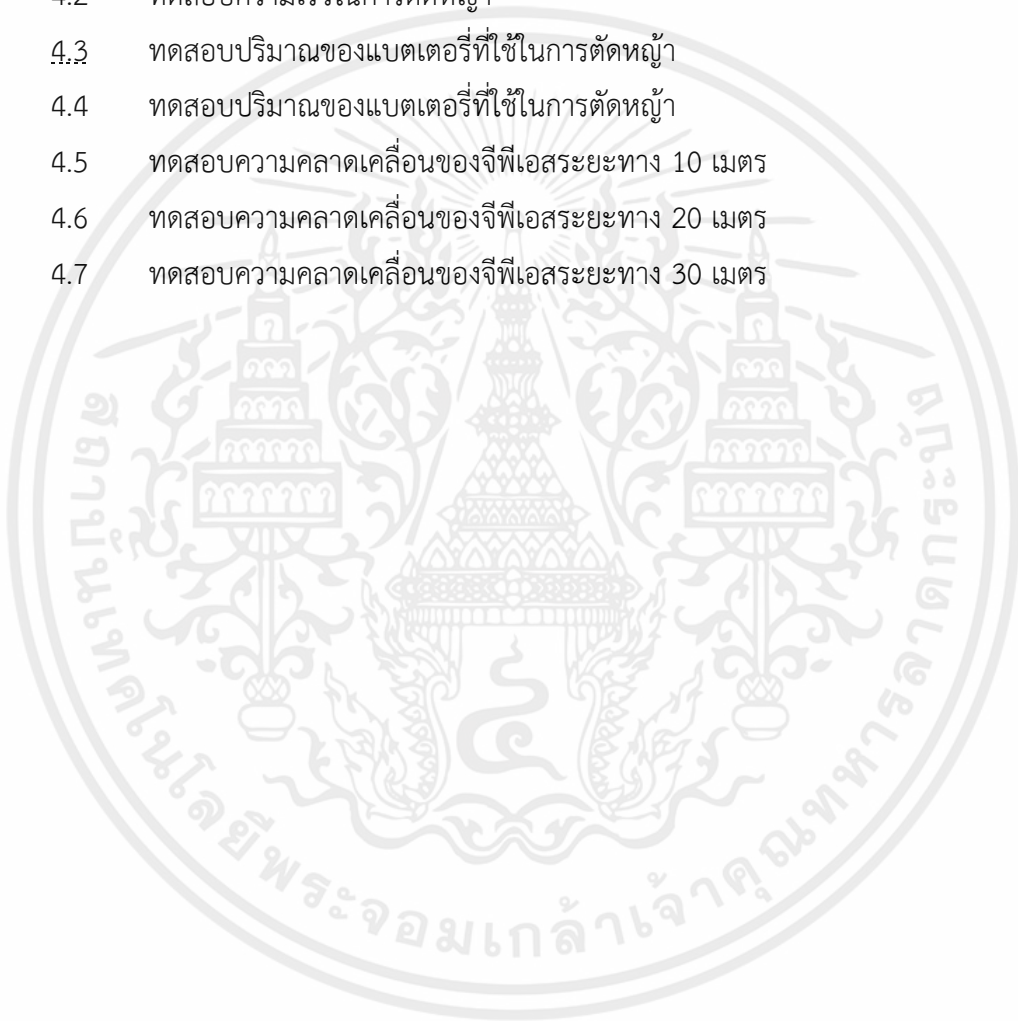
	หน้า
เอกสารอ้างอิง	54
ภาคผนวก ก	56
ภาคผนวก ข	70
ประวัติผู้เขียน	85



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่		หน้า
1.1	ขั้นตอนและวิธีการดำเนินงานภาคเรียนที่ 1	3
1.2	ขั้นตอนและวิธีการดำเนินงานภาคเรียนที่ 2	4
4.1	ทดสอบความเร็วในการเคลื่อนที่	51
4.2	ทดสอบความเร็วในการตัดหญ้า	52
4.3	ทดสอบปริมาณของแบตเตอรี่ที่ใช้ในการตัดหญ้า	52
4.4	ทดสอบปริมาณของแบตเตอรี่ที่ใช้ในการตัดหญ้า	53
4.5	ทดสอบความคลาดเคลื่อนของจีพีเอสระยะทาง 10 เมตร	53
4.6	ทดสอบความคลาดเคลื่อนของจีพีเอสระยะทาง 20 เมตร	54
4.7	ทดสอบความคลาดเคลื่อนของจีพีเอสระยะทาง 30 เมตร	54



## สารบัญรูป

รูปที่		หน้า
2.1	เครื่องตัดหญ้าแบบนั่งขับ	7
2.2	เครื่องตัดหญ้าแบบเข็น	7
2.3	เครื่องตัดหญ้าแบบสะพายป่า	8
2.4	ESP32	9
2.5	การต่อรีเลย์	10
2.6	โมดูลเรกูเลเตอร์ปรับลดแรงดัน	11
2.7	แบตเตอรี่ขนาด 12 โวลต์ 12 แอมป์	12
2.8	มอเตอร์กระแสตรง	13
2.9	กฏมือซ้ายของเฟลมมิง	14
2.10	เซอร์โวมอเตอร์	15
2.11	สามเหลี่ยมพีทาโกรัส	17
2.12	เส้นแบ่งซีกโลก	20
2.13	แอปพลิเคชัน Thunkable	22
3.1	บล็อกไดอะแกรมของเครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน	24
3.2	บล็อกไดอะแกรมเปอร์เซ็นต์แบตเตอรี่	25
3.3	วงจรรีเลย์ควบคุมการเคลื่อนที่	26
3.4	การทำงานของวงจรรีเลย์ตัวที่ 1	27
3.5	การทำงานของวงจรรีเลย์ตัวที่ 2	27
3.6	วงจรรับมอเตอร์ขับเคลื่อน	28
3.7	วงจรรับมอเตอร์ใบตัด	29
3.8	วงจรรับเคลื่อนและวงจรมอเตอร์ตัดหญ้า	29
3.9	โครงสร้างเครื่องตัดหญ้า	30
3.10	รูปแบบการวิ่งของหุ่นยนต์ตัดหญ้าจีพีเอส	31
3.11	ตัวอย่างการแสดงการเซตพิกัดจุด 4 จุดจากภูเกิลแมพ	31
3.12	ทฤษฎีบทพีทาโกรัส เพื่อหาระยะทาง	32

## สารบัญรูป (ต่อ)

รูปที่		หน้า
3.13	แสดงการหา $n$ จำนวนครั้งที่จะวิ่ง	33
3.14	ละติจูด และ ลองจิจูด แต่ละจุดของ $n$ จากสมบัติของสามเหลี่ยม แต่ละจุด	33
3.15	Latitude <sub><math>n</math></sub> และ Longitude <sub><math>n</math></sub> แต่ละจุด	34
3.16	ภาพตัวอย่างการทดสอบพิกัด 4 จุด ที่สนามฟุตบอลโรงเรียนบ้านหินกบ	35
3.17	โครงสร้างของแอปพลิเคชัน	39
3.18	การกรอกพิกัด 4 จุดเพื่อเริ่มทำงาน	40
3.19	หน้าแสดงสถานะเปอร์เซ็นต์แบตเตอรี่	41
3.20	แสดงพิกัดปัจจุบันของเครื่องตัดหญ้า	42
3.21	ปุ่มสวิตช์ควบคุมการเปิด-ปิดใบตัดหญ้า	43
3.22	หน้าควบคุมแบบบังคับบังคับ	43
3.22	ไฟร์เบสเก็บข้อมูล	44
4.1	ทดสอบความเร็วในการเคลื่อนที่	46
4.2	ทดสอบความเร็วในการเคลื่อนที่พื้นที่ 20 ตาราง	47
4.3	ทดสอบการตัดหญ้า 20 ตารางเมตร	48
4.4	ทดสอบการตัดหญ้าที่พื้นที่ 100 ตารางเมตร	49
4.5	ทดสอบการตัดหญ้าที่พื้นที่ 100 ตารางเมตร	49
4.6	ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 10 เมตร	50
4.7	ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 20 เมตร	51
4.8	ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 30 เมตร	52

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญ

ในประเทศไทยมีลักษณะภูมิประเทศแบบร้อนชื้นเหมาะสำหรับการเกษตร พื้นที่ที่อุดมสมบูรณ์จะประกอบไปด้วยพืชต่างๆ เช่น กล้วย ซึ่งนิยมนำมาปลูกในสนามเพื่อตกแต่งหน้าดินตามสวนต่างๆ เช่น สนามหญ้าในสวนสาธารณะ สนามฟุตบอล และสนามเด็กเล่น เป็นต้น และยิ่งในช่วงฤดูฝนทำให้หญ้าในสนามนั้นขึ้นเร็วจนเกินไปสนามจึงรกและดูไม่สวยงาม อีกทั้งเป็นเหตุทำให้สัตว์ที่มีพิษมาอาศัยอยู่ในบริเวณสนามหญ้าได้ และอาจทำอันตรายแก่ผู้ที่ใช้พื้นที่ในสนาม เนื่องจากในปัจจุบันการพัฒนาทางด้านเทคโนโลยีต่างๆนั้นถูกพัฒนาขึ้นอย่างรวดเร็ว เครื่องตัดหญ้าที่ใช้งานในปัจจุบันนั้นมีทั้งแบบสะพาย และแบบเข็นที่ยังต้องใช้มนุษย์เป็นผู้ควบคุมการทำงานในการตัดหญ้า และค่อนข้างใช้เวลานานในการตัดหญ้าแต่ละครั้ง อีกทั้งเป็นเหตุทำให้ผู้ใช้งานเหนื่อยล้าอ่อนแรงง่าย พวกเราจึงเห็นว่าเทคโนโลยีของจีพีเอส (GPS) สามารถนำมาประยุกต์ใช้กับเครื่องตัดหญ้าโดยใช้ในการนำทางการตัดหญ้าที่มีพื้นที่ขนาดใหญ่ได้ เพื่อลดจำนวนแรงงานในการทำงาน และอำนวยความสะดวกให้กับมนุษย์มากยิ่งขึ้น รวมไปถึงสามารถนำไปประยุกต์ใช้งานในด้านการเกษตรกรรม

เครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสจะมุ่งเน้นการทำงานด้วยระบบอัตโนมัติอย่างเต็มรูปแบบ เช่น การเคลื่อนที่ตามระยะทางโดยกำหนด การหยุดทำงานเองอัตโนมัติเมื่อเคลื่อนที่สู่จุดหมายที่กำหนดเอาไว้ เป็นต้น โดยจะทำงานควบคู่กับแอปพลิเคชันที่สร้างขึ้นมา เพื่อทราบสถานะต่างๆ เช่น สถานะของเครื่องตัดหญ้าเริ่มทำงานหรือหยุดทำงาน การบอกตำแหน่งที่เครื่องตัดหญ้าอยู่ สถานะของแบตเตอรี่ในรูปแบบเปอร์เซ็นต์ และระบบบังคับ เป็นต้น

## 1.2 วัตถุประสงค์ของงาน

1. เพื่อศึกษาเกี่ยวกับการทำงานของจีพีเอส
2. เพื่อศึกษาและสร้างเครื่องตัดหญ้าสนามที่ควบคุมด้วยไมโครคอนโทรลเลอร์
3. สามารถนำมาประยุกต์ใช้งานในด้านการนำทาง
4. สามารถนำมาใช้งานในด้านการเกษตร

## 1.3 สมมติฐานของการศึกษา

1. เครื่องตัดหญ้าสามารถตัดหญ้าได้ทั้งแบบบังคับและแบบอัตโนมัติ
2. สามารถระบุตำแหน่งบนแอปพลิเคชัน (Application) มือถือ

## 1.4 ขอบเขตของการศึกษา

1. สามารถตัดหญ้าสนามเป็นรูปสี่เหลี่ยมจัตุรัสหรือสี่เหลี่ยมผืนผ้าได้
2. สามารถส่งงานผ่านแอปพลิเคชันแอนดรอยด์บนมือถือได้โดยรับค่าผ่านที่กำหนดจุดบนมือถือ
3. ความเร็วของการในการตัดหญ้า 6 ตารางเมตรต่อนาที
4. สามารถตัดหญ้า 100 ตารางเมตรต่อการชาร์จ 1 ครั้ง
5. ระบบแจ้งเตือนเปอร์เซ็นต์ของแบตเตอรี่

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถประยุกต์ใช้งานระบบจีพีเอสได้
2. สามารถใช้เครื่องตัดหญ้าทำงานเองด้วยระบบอัตโนมัติได้
3. ทำให้เกิดทักษะการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์
4. สามารถนำไปใช้งานได้จริงในการตัดหญ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.6 ขั้นตอนและวิธีการดำเนินงาน

ขั้นตอนและวิธีการดำเนินงานทางคณะผู้จัดทำได้ดำเนินการแบ่งออกเป็น 2 ภาคการศึกษา คือ แผนการดำเนินงานในภาคเรียนที่ 1 แผนการดำเนินงานในภาคเรียนที่ 2 ซึ่งได้แจกแจงไว้ในตารางที่ 1.1 และ ตารางที่ 1.2 ดังนี้

ตารางที่ 1.1 ขั้นตอนและวิธีการดำเนินงานภาคเรียนที่ 1

ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน																			
	ส.ค. 63				ก.ย. 63				ต.ค. 63				พ.ย. 63				ธ.ค. 63			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.คิดหัวข้อโครงการ	←————→																			
2. ศึกษาการทำงานของเครื่อง					←————→															
3.จัดทำโครงสร้างของเครื่อง									←————→											
4. จัดทำวงจรของเครื่อง									←————→											
5. จัดทำโปรแกรมของเครื่อง									←————→											
6. ทดสอบระบบและแก้ปัญหา													←————→							
7. จัดทำรายงานและนำเสนอ																	←————→			

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.2 ขั้นตอนและวิธีการดำเนินงานภาคเรียนที่ 2

ขั้นตอนการดำเนินงาน	ระยะเวลาการดำเนินงาน																			
	ม.ค. 64				ก.พ. 64				มี.ค. 64				เม.ย 64				พ.ค. 64			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. จัดซื้ออุปกรณ์เติม เพิ่มที่ใช้ในโครงการ	↔																			
2. ศึกษาการทำงานของ จีพีเอส			↔			↔														
3. ศึกษาเกี่ยวกับการ เขียนแอปพลิเคชัน						↔				↔										
4. แก้ไขวงจรเพิ่มเติม											↔									
5. เขียนโปรแกรมเพื่อ ทดสอบ														↔						
6. ทดสอบระบบและ แก้ปัญหา																		↔		
7. จัดทำรายงานและ นำเสนอ																				↔

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.7 โครงสร้างปริญญาโท

เครื่องตัดหญ้าสนามโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน ได้อธิบายขั้นตอนกระบวนการและวิธีการ ออกแบบต่างๆ โดยมีส่วนของเนื้อหาเป็นบทต่างๆดังต่อไปนี้

บทที่ 1 บทนำ ในส่วนของบทนี้จะกล่าวถึงที่มาและความสำคัญของโครงการ วัตถุประสงค์ของโครงการ สมมติฐานของการศึกษา ขอบเขตของโครงการ ประโยชน์ที่คาดว่าจะได้รับ และขั้นตอนการดำเนินงาน

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง บทนี้จะกล่าวถึง ชนิดของเครื่องตัดหญ้าแบบต่างๆ การทำงานของมอเตอร์ การทำงานของรีเลย์ ทฤษฎีบทพีทาโกรัส และทฤษฎีการทำงานของระบบจีพีเอส

บทที่ 3 การออกแบบ ในบทนี้จะกล่าวถึงการออกแบบโครงสร้างของเครื่องตัดหญ้า วงจรการขับเคลื่อน บล็อกไดแกรมการทำงานของเครื่องตัดหญ้า วงจรการขับมอเตอร์ใบตัด และการพื้นที่ในการตัดหญ้า

บทที่ 4 การทดลองและผลการทดลอง บทนี้จะกล่าวถึงวิธีการทดลอง การทดสอบการตัดหญ้าของเครื่องตัดหญ้าในพื้นที่ที่กำหนด

บทที่ 5 บทสรุปและข้อเสนอแนะ ในบทนี้จะกล่าวถึงการสรุปผลการทดลองการทำงานของเครื่องหญ้าอุปสรรคและข้อเสนอแนะต่างๆที่สามารถพัฒนาเครื่องตัดหญ้าให้มีประสิทธิภาพในการทำงานเพิ่มขึ้นในโอกาสต่อไป

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้อง

โครงการเครื่องตัดหญ้าอัตโนมัติโดยใช้จีพีเอส (GPS) ควบคุมผ่านแอปพลิเคชัน นั้นเป็นการนำเทคโนโลยีทางด้านจีพีเอสเข้ามาประยุกต์ใช้งาน ซึ่งเป็นการนำทางในการตัดหญ้า โดยการระบุตำแหน่งลงในแอปพลิเคชันบนมือถือเพื่อให้เครื่องตัดหญ้าสามารถทำงานเองแบบอัตโนมัติ โดยทฤษฎีที่ได้นำมาใช้ในโครงการมีหัวข้อดังต่อไปนี้

### 2.1 ความหมายของเครื่องตัดหญ้า

#### 2.1.1 ความรู้เรื่องเครื่องตัดหญ้า [1]

เครื่องตัดหญ้าเป็นเครื่องในการช่วยตัดและลดความสูงของหญ้า โดยที่มีมนุษย์เป็นผู้ย้ายตำแหน่ง แหล่งพลังงานเป็นเครื่องยนต์สันดาป แต่ในปัจจุบันเครื่องตัดหญ้าที่มีขนาดใหญ่ไม่ใช่เพียงแต่มนุษย์ผู้อยู่ข้างหลังเครื่องตัดหญ้า เพราะมีเครื่องตัดหญ้าที่เดินตามมนุษย์แล้ว ซึ่งเครื่องตัดหญ้ามีความเป็นมาที่น่าสนใจ มีประวัติการกำเนิดเครื่องตัดหญ้า ในปี 1830 วิศวกรที่ชื่อ Edwin เขาได้สร้างเครื่องตัดหญ้าขึ้น อันที่จริงแล้วเหตุผลของการสร้างก็คือต้องการเครื่องมือที่สามารถแทนเคียวได้เพราะเมื่อก่อนยังใช้เคียวตัดหญ้าอยู่ ถือได้ว่าช่วยทุ่นแรงได้มากทีเดียวกับการตัดหญ้าในสนามกีฬาและสวนกว้างๆ จากนั้นก็พัฒนามาเรื่อย ๆ ทำให้เครื่องตัดหญ้าเบาขึ้นและเงียบขึ้น เครื่องตัดหญ้าขณะนั้นทำจากเหล็กหล่อ กลไกไม่ซับซ้อน ถัดมาก็ใช้ไอน้ำเป็นแหล่งพลังงาน แล้วก็เครื่องยนต์เบนซิน จนมาเป็นการใช้ไฟฟ้า ล่าสุดจะเป็นไร้สายใช้พลังงานจากแบตเตอรี่แทน นวัตกรรมเกี่ยวกับเครื่องตัดหญ้านั้นพัฒนาไปอย่างต่อเนื่อง

#### 2.1.2 เครื่องตัดหญ้าแบบนั่งขับ [1]

เครื่องตัดหญ้าแบบนั่งขับ เป็นเครื่องตัดหญ้า ที่ใช้เครื่องยนต์ในขับเคลื่อนขนาดของใบมีดจะมีขนาด 30-48 นิ้ว จะมีใบจานหมุนแบบกลมและหมุนด้วยความเร็วรอบสูง 3,000 รอบ/นาที แต่จะต้องอาศัยกำลังจากเพลาส่งกำลังของตัวรถแทรกเตอร์ ซึ่งจะต้องมีการทดรอบการหมุนของเพลาลูก และส่งกำลังไปที่ชุดเฟืองก่อนเพื่อเปลี่ยนแปลงกำลังให้ใบมีด ดังแสดงรูปที่ 2.1



รูปที่ 2.1 เครื่องตัดหญ้าแบบนั่งขับ

ที่มา: <https://www.umacthai.com>

### 2.1.3 เครื่องตัดหญ้าแบบเข็น [1]

เครื่องตัดหญ้าแบบเข็น ใช้เครื่องยนต์เบนซินในการขับเคลื่อน มีขนาดใบมีดที่ 18-22 นิ้ว เหมาะกับการใช้งานในพื้นที่เรียบ เช่น สนามหญ้าหน้าบ้าน สนามเด็กเล่น และสนามฟุตบอลใช้งานง่ายสะดวกต่อการใช้งาน และใช้กำลังจากเครื่องยนต์เป็นหลัก ดังแสดงรูปที่ 2.2



รูปที่ 2.2 เครื่องตัดหญ้าแบบเข็น

ที่มา: <https://www.stintertrade.com>

### 2.1.4 เครื่องตัดหญ้าแบบสะพายบ่า [1]

เครื่องตัดหญ้าแบบสะพายบ่า จะนิยมใช้งานตามไร่นาหรือตามสวน และในส่วนของตัวเครื่องยนต์จะแบบ 2 จังหวะ และแบบ 4 จังหวะตามลักษณะการใช้งานเหมาะกับพื้นที่ลาดชัน พื้นที่สูงต่ำไม่เท่ากัน ดังแสดงรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 เครื่องตัดหญ้าแบบสะพายบ่า

ที่มา: <https://www.umacthai.com>

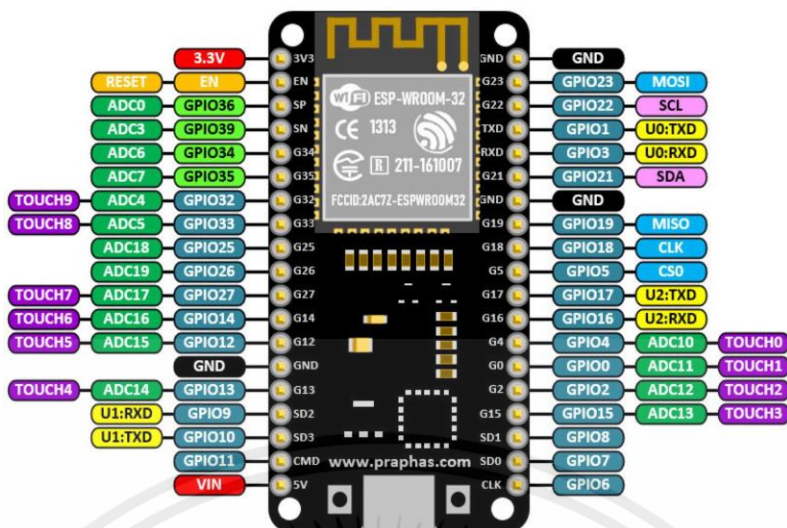
## 2.2 บอร์ดควบคุม (control board) [2]

### 2.2.1 โหนดเอ็มซียู (Node MCU ESP 32)

บอร์ด ESP 32 แสดงดังรูปที่ 2.4 เป็นบอร์ดไมโครคอนโทรลเลอร์อีกตัวที่ปัจจุบันกำลังได้รับความนิยมอย่างสูง เนื่องจากมีขนาดเล็กมาก และสามารถเป็นไมโครคอนโทรลเลอร์ที่สามารถเชื่อมต่อไวไฟ (WiFi) ได้และเชื่อมต่อบลูทูธได้ มาพร้อมไมโครคอนโทรลเลอร์สถาปัตยกรรม Tensilica LX6 สัญญาณนาฬิกา 240 MHz แบบ 2 แกนสมอง มาพร้อมกับ I/O ทั้งหมด 32 ขา รองรับ ADC จำนวน 12 ช่อง UART, SPI จำนวนอย่างละ 3 ช่อง และ I2S, I2C, DAC อย่างละ 2 ช่อง รองรับ PWM และมี Timer, RTC ภายใน

### 2.2.2 Esp-32 Module IntoRobotW32/W33

IntoRobotW32/W33 ผลิตโดยบริษัท Into Robot มีอยู่ด้วยกัน 2 รุ่น คือรุ่น W32 ที่ใช้เสาอากาศแบบ PCB ที่อยู่บนบอร์ดและรุ่น W33 ที่มีเสาชนิดเซรามิกเป็นเสาหลักในการรับสัญญาณและสามารถเปลี่ยนไปใช้คอนเนคเตอร์แบบ IPEX สำหรับต่อเสาอากาศภายนอกได้ตำแหน่งขาและขนาดของโมดูลจะมีขาต่อใช้งานทั้งหมด 31 ขา น้อยกว่า ESP-WROOM-32 / ESP32S แต่มีขามากกว่าโมดูล ESP32-Bit มีขนาดเล็กกว่า ESP-WROOM32/ESP32S แต่มีขนาดใหญ่กว่าโมดูล ESP32-Bit



รูปที่ 2.4 ESP 32

ที่มา: <https://www.praphas.com>

### 2.2.3 ขาใช้งานต่างๆ ของ ESP32

- มี GPIO จำนวน 32 ช่อง
- รองรับ UART จำนวน 3 ช่อง
- รองรับ SPI จำนวน 3 ช่อง
- รองรับ I2C จำนวน 2 ช่อง
- รองรับ ADC จำนวน 12 ช่อง
- รองรับ DAC จำนวน 2 ช่อง
- รองรับ I<sup>2</sup>S จำนวน 2 ช่อง
- รองรับ PWM / Timer ทุกช่อง
- รองรับการเชื่อมต่อกับ SD-Card
- รองรับการเข้ารหัส WIFI แบบ WEP และ WPA/WPA2 PSK/Enterprise
- มีวงจรเข้ารหัส AES / SHA2 / Elliptical Curve Cryptography / RSA-4096 ในตัว
- รับ - ส่ง ข้อมูลได้ความเร็วสูงสุดที่ 150Mbps เมื่อเชื่อมต่อแบบ 11n HT40 ได้ความเร็วสูงสุด 72Mbps เมื่อเชื่อมต่อแบบ 11n HT20 ได้ความเร็วสูงสุดที่ 54Mbps เมื่อเชื่อมต่อแบบ 11g และได้ความเร็วสูงสุดที่ 11Mbps เมื่อเชื่อมต่อแบบ 11b
- เมื่อใช้การเชื่อมต่อผ่านโปรโตคอล UDP จะสามารถรับ - ส่งข้อมูลได้ที่ความเร็ว 135Mbps ในโหมด Sleep ใช้กระแสไฟฟ้าเพียง 2.5u

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3 รีเลย์ (Relay) [3]

รีเลย์เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็กตั้งรีเลย์ใช้กระแสต่ำ ๆ เพื่อควบคุมตัดต่อโหลดที่มีกระแสสูง ๆ รีเลย์นั้นทำหน้าที่ปิดและเปิดทำหน้าที่คล้ายสวิตช์อิเล็กทรอนิกส์ ซึ่งสามารถนำรีเลย์ไปประยุกต์ใช้งานในการควบคุมวงจรต่างๆได้อีกมากมาย

### 2.3.1 หลักการทำงานของรีเลย์

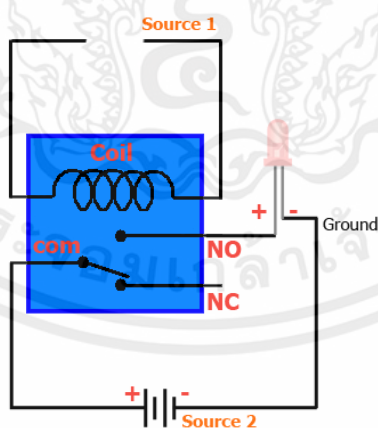
รีเลย์ประกอบไปด้วย 2 ส่วนหลักๆ

1. ส่วนของขดลวด (coil) เหนี่ยวนำกระแสต่ำทำหน้าที่สร้างสนามแม่เหล็กไฟฟ้าให้แก่โลหะไปตันให้หน้าสัมผัสต่อกัน ทำงานโดยการรับแรงดันจากภายนอกต่อคร่อมที่ ขดลวดเหนี่ยวนำ เมื่อขดลวดได้รับแรงดัน (ค่าแรงดันที่รีเลย์ต้องการขึ้นกับชนิด และรุ่นตามทีผลิต) จะเกิดสนามแม่เหล็กไฟฟ้าทำให้แกนโลหะด้านในไปตันให้แผ่นหน้าสัมผัสกัน

2. ส่วนของหน้าสัมผัส (Contact) ทำหน้าที่คล้ายกับสวิตช์จ่ายกระแสไฟฟ้าให้กับอุปกรณ์ที่เราต้องการใช้งานนั่นเอง

### 2.3.2 จุดการต่อใช้งานของรีเลย์

จุด NC (Normal close) คือ ปกติรีเลย์จะอยู่ที่ตำแหน่งนี้หากไม่มีการจ่ายกระแสไฟฟ้าให้กับขดลวดเหนี่ยวนำหน้าสัมผัสจะติดกัน โดยทั่วไปแล้วจะต้องต่ออุปกรณ์ไฟฟ้าที่ต้องการให้ทำงานไว้ที่จุดนี้ จุด NO (Normal open) คือ ปกติเปิดหรือหากยังไม่มีการจ่ายกระแสไฟฟ้าให้กับขดลวดเหนี่ยวนำหน้าสัมผัสจะไม่ติดกัน และจุด C (Common) คือ จุดรวมที่ต่อจากแหล่งจ่ายไฟฟ้า ดังแสดงรูปที่ 2.5



รูปที่ 2.5 การต่อรีเลย์

ที่มา: <https://commandronestore.com>

### 2.3.3 ประเภทของรีเลย์

รีเลย์เป็นอุปกรณ์ที่ทำหน้าที่คล้ายกับสวิตช์มีหน้าที่ปิดเปิด รีเลย์สามารถใช้ในการควบคุมวงจรไฟฟ้าได้หลายอย่างมาก รีเลย์จะทำงานโดยการจ่ายกระแสไฟฟ้าให้กับขดลวด สามารถแบ่งออกตามลักษณะการใช้งานได้ 2 ประเภทคือ

รีเลย์กำลัง (Power Relay) โดยทั่วไปแล้วจะเรียกว่าคอนแทคเตอร์ (Contactor or Magnetic contactor) ใช้ในการควบคุมไฟฟ้ากำลัง และมีขนาดใหญ่กว่ารีเลย์ธรรมดา รีเลย์ควบคุม (Control Relay) ใช้ในการควบคุมในงานทั่วไปที่มีกำลังไฟฟ้าต่ำ โดยรีเลย์ควบคุมประเภทนี้มักเรียกว่า รีเลย์

## 2.4 โมดูลเรกูเลเตอร์ปรับลดแรงดัน (DC-to-DC Step Down LM2593)



รูปที่ 2.6 โมดูลเรกูเลเตอร์ปรับลดแรงดัน

ที่มา: <https://www.arduitronics.com>

### 2.4.1 โมดูลเรกูเลเตอร์ LM2593 [4]

เป็นโมดูลสวิตช์ซึ่งเรกูเลเตอร์แบบลดแรงดัน (Step-down) ดังแสดงรูปที่ 2.6 แปลงไฟกระแสตรงที่มีแรงดันมากให้มีแรงดันที่น้อยลง โมดูลเรกูเลเตอร์ LM2593 มีประสิทธิภาพในการจ่ายไฟที่สูง อีกทั้งยังราคาถูกอีกด้วย สามารถปรับตั้งแรงดันได้โดยการหมุนตัวต้านทานปรับค่าได้

### 2.4.2 รายละเอียด LM2593

- รองรับแรงดันเอาต์พุตสูงสุด 1.23-30V
- ปรับตั้งแรงดันอินพุตได้ในช่วง 4-35 V
- จ่ายกระแสต่อเนื่องได้สูงสุด 3A

### 2.4.3 วิธีการใช้งานโมดูลเรกูเลเตอร์

โมดูลเรกูเลเตอร์สามารถปรับแรงดันตั้งแต่ 1.23-30V โดยหมุนที่ตัวต้านทานปรับค่าได้ หลายนรอบแรงดันจะค่อยๆเพิ่มและลดตามที่เราปรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. หมุนตามเข็มนาฬิกา คือ ปรับแรงดันขึ้น
2. หมุนทวนเข็มนาฬิกา คือ ปรับแรงดันลง

## 2.5 แบตเตอรี่ (Battery) [5]

แบตเตอรี่เป็นอุปกรณ์ที่ทำหน้าที่จัดเก็บพลังงานไฟฟ้าแสดงดังรูปที่ 2.6 และจ่ายพลังงานให้กับอุปกรณ์ไฟฟ้าและอิเล็กทรอนิกส์ต่างๆ แบตเตอรี่เป็นอุปกรณ์ที่สามารถแปลงพลังงานเคมีให้เป็นพลังงานไฟฟ้าได้โดยตรงด้วยการใช้เซลล์กัลวานิก (galvanic cell) ที่ประกอบไปด้วยขั้วบวกและขั้วลบ พร้อมกับสารละลายอีกชนิดหนึ่งคือ สารละลายอิเล็กโทรไลต์ (electrolyte solution) แบตเตอรี่เป็นอุปกรณ์สำหรับจัดเก็บพลังงานไฟฟ้าเท่านั้น ไม่ได้ผลิตไฟฟ้า สามารถอัดประจุเข้าไปได้อีกหลายๆ ครั้งแต่ประสิทธิภาพของแบตเตอรี่นั้นจะไม่เต็ม 100% ในการอัดประจุเข้าไปใหม่ในแต่ละครั้งนั้นประสิทธิภาพของแบตเตอรี่ จะอยู่ที่ประมาณ 85% เพราะเสียพลังงานไปบางส่วนในรูปแบบความร้อนและปฏิกิริยาเคมีจากการประจุและคายประจุ



รูปที่ 2.7 แบตเตอรี่ขนาด 12 โวลต์ 12 แอมป์

## 2.6 มอเตอร์กระแสตรง (DC Motor) [6]

มอเตอร์ไฟฟ้า ดังแสดงรูปที่ 2.8 เป็นอุปกรณ์ที่นิยมใช้งานกันอย่างแพร่หลายเป็นอุปกรณ์ที่ใช้ควบคุมเครื่องจักรกลต่างๆในงานอุตสาหกรรมมอเตอร์มีหลายแบบหลายชนิดที่ใช้ให้เหมาะสมกับงานดั่งนั้น เราจึงควรทราบถึงความหมายและชนิดของมอเตอร์ไฟฟ้าตลอดคุณสมบัติการใช้งานของมอเตอร์แต่ละชนิดเพื่อให้เกิดประสิทธิภาพสูงสุดในการใช้งานของตัวมอเตอร์นั้น และสามารถเลือกใช้งานให้เหมาะสมกับงาน

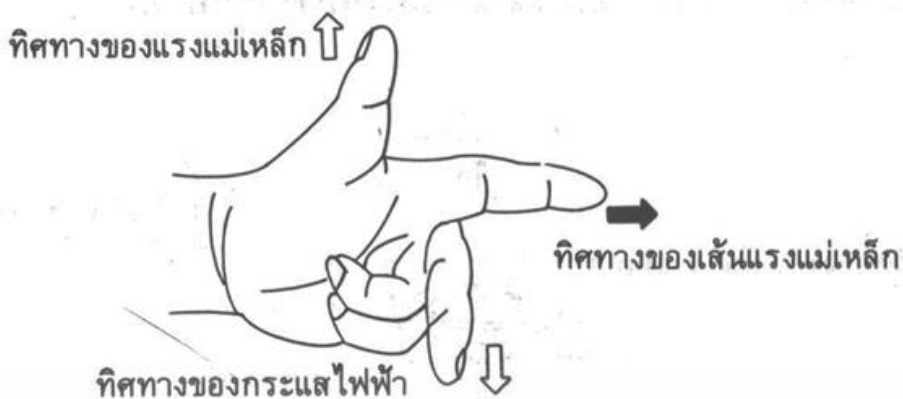


รูปที่ 2.8 มอเตอร์กระแสตรง

ที่มา: <https://th.aliexpress.com>

### 2.6.1 หลักการทำงานของมอเตอร์กระแสตรง [6]

ในอดีตมีนักวิทยาศาสตร์ 2 ท่าน คือ ไฮน์ริช เฮอร์เชล และเซอร์จอห์น อัมโบรสเฟลมมิง ได้ทำการทดลองและพบหลักการในการทำงานของมอเตอร์ไฟฟ้ากระแสตรง ดังนี้ เมื่อมีขดลวดตัวนำ หมุนตัดกับสนามแม่เหล็กไฟฟ้าจะทำให้เกิดแรงดันไฟฟ้าเหนี่ยวนำขึ้นในขดลวดตัวนำ และทิศทางในการไหลของกระแสไฟฟ้าในขดลวดตัวนำจะไหลในทิศทางที่ต้านกับการหมุนของขดลวดตัวนำซึ่งกฎทิศทางกระแสไฟฟ้าในขดลวดตัวนำจะไหลในทิศทางที่ต้านกับการหมุนของขดลวดตัวนำซึ่งกฎทิศทางกระแสไฟฟ้า เรียกว่า กฎของเลนซ์นักวิทยาศาสตร์ เซอร์จอห์น อัมโบรสเฟลมมิง พบการหมุนของมอเตอร์ว่าจะหมุนไปในทิศทางใดถ้าหากว่าทราบทิศทางกระแสไฟฟ้าก็จะพบความสัมพันธ์ ถ้าทราบว่าทิศทางของเส้นแรงแม่เหล็กและทิศทางกระแสไฟฟ้า จะทราบทิศทางของการหมุนของขดลวดตัวนำได้ โดยกฎมือซ้ายของเฟลมมิง ดังแสดงรูปที่ 2.9



รูปที่ 2.9 กฏมือซ้ายของเฟลมมิง

ที่มา: <https://www.auto2drive.com>

### 2.6.2 การกลับทิศทางของมอเตอร์กระแสตรง [7]

การทำงานของมอเตอร์และการกลับทิศทางของมอเตอร์กระแสตรง [3] ในการทำงานที่จะควบคุมการหมุนและทิศทางของมอเตอร์นั้นจะต้องมีวงจรในการช่วยทำงาน การใช้รีเลย์เข้ามาควบคุมการเปลี่ยนทิศทางของมอเตอร์ โดยการควบคุมการปิด-เปิดรีเลย์ 2 ตัว ซึ่งจะทำงานกลับทิศทางกันโดยสลับการทำงานของรีเลย์ทีละตัว เช่น การให้รีเลย์ตัวที่ 1 ทำงาน และรีเลย์ตัวที่ 2 หยุดทำงานมอเตอร์จะหมุนตามเข็มนาฬิกา และถ้าให้รีเลย์ตัวที่ 1 หยุดทำงาน และรีเลย์ตัวที่ 2 ทำงานมอเตอร์จะหมุนทวนเข็มนาฬิกา

### 2.7 ไฟร์เบส (Firebase) [8]

ไฟร์เบสเป็นหนึ่งในผลิตภัณฑ์ของกูเกิล (google) ปัจจุบันถูกนำมาใช้งานอย่างแพร่หลายไฟร์เบส คือแพลตฟอร์ม (Platform) ที่รวบรวมข้อมูลต่างๆ สำหรับการจัดการในส่วน Backend หรือ Server side ซึ่งสามารถพัฒนาแอปพลิเคชัน (Application) ได้อย่างมีประสิทธิภาพ โดยมีเครื่องมือที่ฟรี และเสียค่าใช้จ่าย

ไฟร์เบส ก่อตั้งขึ้นในปี 2011 โดยแอนดรูลีและเจมส์ เทมปลิน ในส่วนของบริการหลักของไฟร์เบส คือระบบฐานข้อมูลแบบเรียลไทม์ ซึ่งมี 2 รูปแบบดังนี้

1. คลาวด์ไฟร์สโตร์ (Cloud Firestore) หรือเรียกอีกอย่างว่าไฟร์สโตร์เป็นบริการของฐานข้อมูลแบบใหม่ ใช้สำหรับในการจัดเก็บข้อมูลและอัปเดตข้อมูลในรูปแบบเรียลไทม์

2. ข้อมูลแบบเรียลไทม์ (Realtime Database) เป็นบริการฐานข้อมูลแบบเดิมของไฟร์เบสที่ใช้วิธีเก็บข้อมูลในแบบ JSON รองรับการทำงานในแบบของเรียลไทม์ แต่เป็นฐานข้อมูลที่เก็บอยู่ในโซนเดียว

## 2.8 เซอร์โวมอเตอร์ (Servo Motor)

เซอร์โวมอเตอร์ ดังแสดงรูปที่ 2.10 เป็นคำศัพท์ที่ใช้ในการควบคุมอัตโนมัติ ในเชิงความหมายของ เซอร์โวมอเตอร์คือมอเตอร์ที่สามารถสั่งงานหรือตั้งตามที่ต้องการ องศา ในการใช้งาน โดยใช้การควบคุมแบบป้อนกลับ (Feedback control) การควบคุมแบบป้อนกลับ คือ ระบบควบคุมที่มีการวัดค่าเอาต์พุตของระบบนำมาเปรียบเทียบกับค่าอินพุต เพื่อควบคุมและปรับแต่งให้ค่าเอาต์พุตของระบบมีค่าเท่ากับหรือใกล้เคียงกับค่าอินพุต



รูปที่ 2.10 เซอร์โวมอเตอร์

ที่มา: <http://robo-circuit.com/product/>

### 2.8.1 หลักการทำงานของเซอร์โวมอเตอร์

เมื่อจ่ายสัญญาณพัลส์เข้ามายังเซอร์โวมอเตอร์ส่วนวงจรควบคุม ภายในเซอร์โวมอเตอร์จะทำการอ่านและประมวลผลค่าความกว้างของสัญญาณพัลส์ที่ส่งเข้ามาเพื่อแปลค่าเป็นตำแหน่งองศาที่ต้องการให้ เซอร์โวมอเตอร์หมุนเคลื่อนที่ไปยังตำแหน่งนั้นแล้วส่งคำสั่งไปทำการควบคุมให้เซอร์โวมอเตอร์หมุนไปยังตำแหน่งที่ต้องการ โดยมีตัวเซ็นเซอร์คอยวัดค่ามุมที่มอเตอร์กำลังหมุน เป็น การควบคุมแบบป้อนกลับ กลับมาให้วงจรควบคุมเปรียบเทียบกับค่าอินพุตเพื่อควบคุมให้ได้ตำแหน่งที่ต้องการอย่างถูกต้องและแม่นยำ เซอร์โวมอเตอร์มีการควบคุมโดยความกว้างของสัญญาณพัลส์ (Pulse Width Modulation : PWM) ตัวเซอร์โวมอเตอร์ ออกแบบสำหรับการเคลื่อนที่แบบเชิงมุม เช่น การควบคุมขดพืดของแขนหุ่นยนต์ การเลี้ยวของรถบังคับ เป็นต้น

### 2.8.2 การควบคุมเซอร์โวมอเตอร์ด้วยอาดูโน่

อาดูโน่ (Arduino) มีไลบรารีสำหรับสั่งงาน เซอร์โวมอเตอร์มาให้อยู่แล้วเป็นฟังก์ชันที่สำเร็จรูปไลบรารีเซอร์โว (Servo Library) สามารถสั่งงานให้เซอร์โวมอเตอร์หมุนไปและหมุนกลับได้ 0-180 องศา

ฟังก์ชันภายในสำหรับการสั่งงานเซอร์โวมอเตอร์

- attach ()
- write ()
- writeMicroseconds ()
- read ()
- attached ()
- detach ()
- attach () คือฟังก์ชันที่ใช้ในการกำหนดขาสัญญาณที่เซอร์โวมอเตอร์กับอาดูโน่และ

กำหนดความกว้างของสัญญาณพัลส์ที่ 0 องศาและ 180 องศา

การนำฟังก์ชันไปใช้งาน (Syntax)

Servo.attach(pin)

Servo.attach(pin,min,max)

Pin คือขาสัญญาณของอาดูโน่ที่ใช้ในการเชื่อมต่อกับบอร์ดอาดูโน่

Min คือความกว้างของสัญญาณพัลส์ที่ 0 องศาของเซอร์โวมอเตอร์ที่ใช้ในหน่วยไมโครวินาที (us) โดยปกติ แล้วหากไม่มีการตั้งค่าโปรแกรมจะกำหนดค่าไว้ที่ 544 ไมโครวินาที

Max คือ ความกว้างของสัญญาณพัลส์ที่ 180 องศาของเซอร์โวมอเตอร์ที่ใช้ในหน่วยไมโครวินาที (us) โดยปกติ แล้วหากไม่มีการตั้งค่าโปรแกรมจะกำหนดค่าไว้ที่ 2400 ไมโครวินาที

Write () ฟังก์ชันที่ใช้ในการกำหนดให้เซอร์โวมอเตอร์หมุนไปตามที่ต้องการ คือ 0-180 องศา

Angle คือ คือมุมที่ต้องการให้ เซอร์โวมอเตอร์แบบ 0-180 องศาหมุนไป

writeMicroseconds() คือ ฟังก์ชันที่ใช้ควบคุมตำแหน่งที่ให้เซอร์โวมอเตอร์ไปยังตำแหน่งองศาที่กำหนดโดยกำหนดเป็นค่าความกว้างของพัลส์ในหน่วยไมโครวินาทีปกติแล้วเซอร์โวมอเตอร์จะใช้ความกว้างของสัญญาณพัลส์อยู่ที่ 1000-2000 ไมโครวินาที

การนำเอาฟังก์ชันไปใช้งาน

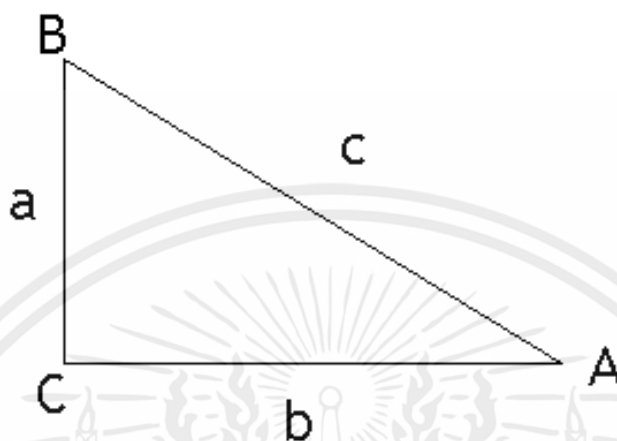
servo.writeMicroseconds(uS)

uS: คือค่าความกว้างของสัญญาณพัลส์ที่จำเป็นต้องกำหนดในหน่วยไมโครวินาที โดยชนิดของตัวแปรเป็นชนิด int

read () คือฟังก์ชันอ่านค่าองศาที่สั่งเข้าไปทำงานด้วยฟังก์ชัน write() เพื่อให้ทราบว่าตำแหน่งองศาสุดท้าย ที่สั่งเข้าไป นั้นมีค่าเท่าไร ซึ่งค่าที่อ่านออกมานั้นจะมีค่าอยู่ในช่วง 0 – 180 องศา

## 2.9 ทฤษฎีบทพีทาโกรัส (Pythagoras's Theorem)

ถ้ามีสามเหลี่ยม ABC เป็นสามเหลี่ยมมุมฉาก ซึ่ง ACB เป็นมุมฉากให้ a, b และ c เป็นความยาวของด้านตรงข้ามมุม A, B และ C ตามลำดับ และจะได้ความสัมพันธ์ระหว่างความยาวทั้งสามของสามเหลี่ยมมุมฉาก ดังแสดงดังรูปที่ 2.11



รูปที่ 2.11 สามเหลี่ยมพีทาโกรัส

ที่มา : <http://www.kr.ac.th>

จากรูป ABC เป็นรูปสามเหลี่ยมมุมฉาก ซึ่งมี C เป็นมุมฉาก

- AB เป็นด้านตรงข้ามมุมฉาก อยู่ตรงข้ามกับมุม C แทนด้วยความยาว c
- AC เป็นด้านประกอบมุมฉาก อยู่ตรงข้ามกับมุม B แทนด้วยความยาว b
- BC เป็นด้านประกอบมุมฉาก อยู่ตรงข้ามกับมุม A แทนด้วยความยาว a

### 2.9.1 ทฤษฎีบทของพีทาโกรัส [9]

แบบที่ 1 ในรูปสามเหลี่ยมมุมฉากใด ๆ ความยาวกำลังสองของด้านตรงข้ามมุมฉากมีค่าเท่ากับผลบวกของความยาวกำลังสองของด้านประกอบมุมฉากทั้งสองด้านหรือพิจารณาจากพื้นที่รูปสี่เหลี่ยมจัตุรัส

แบบที่ 2 ในรูปสี่เหลี่ยมมุมฉากใด ๆ พื้นที่ของรูปสี่เหลี่ยมจัตุรัสบนด้านตรงข้ามมุมฉากมีค่าเท่ากับผลบวกของพื้นที่รูปสี่เหลี่ยมจัตุรัสบนด้านประกอบมุมฉากสองด้าน

จากทฤษฎีบททั้ง 2 แบบ จะได้ความสัมพันธ์ระหว่างความยาวของด้านทั้งสามของรูปสามเหลี่ยมมุมฉาก ABC สามารถคำนวณได้จากสมการที่ 2.1

$$c^2 = a^2 + b^2 \quad (2.1)$$

โดยที่ c เป็นความยาวด้านตรงข้ามมุมฉาก และ a และ b เป็นความยาวของด้านประกอบมุมฉากสามารถคำนวณได้จากสมการที่ 2.2

$$a^2 = b^2 - c^2 \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

$$b^2 = c^2 - a^2 \quad (2.3)$$

หาก  $c$  นั้นแทนด้วยความยาวของด้านตรงข้ามมุมฉาก และ  $a$  และ  $b$  แทนความยาวของด้านประกอบมุมฉากแล้ว ทฤษฎีบทพีทาโกรัสจะสามารถเขียนในรูปสมการพีทาโกรัสได้ดังนี้

ถ้าทราบความยาวของทั้ง  $a$  และ  $b$  ค่า  $c$  จะสามารถคำนวณได้จากสมการที่ 2.4

$$c = \sqrt{a^2 + b^2} \quad (2.4)$$

ถ้าทราบความยาวด้านตรงข้ามมุมฉาก  $c$  และด้านประกอบมุมฉากด้านใดด้านหนึ่ง ( $a$  หรือ  $b$ ) แล้วความยาวด้านที่เหลือสามารถคำนวณได้ดังนี้ จะสามารถคำนวณได้จากสมการที่ 2.4 และ 2.5

$$a = \sqrt{c^2 - b^2} \quad (2.5)$$

หรือ

$$b = \sqrt{c^2 - a^2} \quad (2.6)$$

## 2.10 จีพีเอส (GPS) [10]

จีพีเอส มีชื่อเต็มว่า Global Positioning System คือ ระบบระบุตำแหน่งบนพื้นโลก เป็นระบบดาวเทียมประมาณ 24 ดวง จะโคจรรอบโลกที่ความสูง 10,600 ไมล์ จากพื้นผิวโลก ดาวเทียมจะเว้นพื้นที่เท่าๆกันจากทุกมุมบนโลก และจะมีดาวเทียม 4 ดวง ที่อยู่เหนือเส้นขอบฟ้า ระบบจีพีเอสนี้ทำให้คนบนพื้นโลกที่มีเครื่องรับสัญญาณจีพีเอสสามารถทราบถึงพิกัดตำแหน่งที่ตนเองยืนอยู่ได้โดยความแม่นยำของการระบุตำแหน่งนั้นอยู่ระหว่าง 10 ถึง 100 เมตร

### 2.10.1 ส่วนประกอบของจีพีเอส

จีพีเอส ประกอบไปด้วย 3 ส่วนหลักๆ

1. ส่วนอวกาศ ประกอบไปด้วยดาวเทียม 3 ค่าย คือ อเมริกา รัสเซีย ยุโรป ของอเมริกาชื่อ NAVSTAR (Navigation Satellite Timing and Ranging GPS) มีดาวเทียมทั้งหมด 28 ดวงและที่ใช้งานจริง 24 ดวง อีก 4 ดวงเป็นดาวเทียมสำรอง มีรัศมีวงโคจรจากพื้นโลกประมาณ 20,162.81 กิโลเมตร หรือ 12,600 ไมล์ ดาวเทียมแต่ละดวงนั้นใช้เวลาในการโคจรรอบโลกประมาณ 12 ชั่วโมง

2. ส่วนควบคุม ประกอบไปด้วยสถานีภาคพื้นดิน สถานีใหญ่ที่ Falcon Air Force Base ประเทศอเมริกา และศูนย์ควบคุมย่อยอีก 5 จุด กระจายไปยังภูมิภาคอื่นๆทั่วโลก

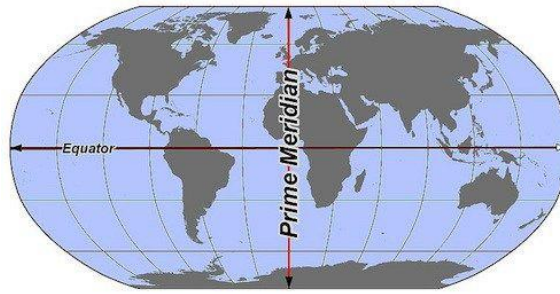
3. ส่วนผู้ใช้งาน ในส่วนของผู้ใช้งานต้องมีเครื่องรับสัญญาณที่สามารถรับคลื่นและแปรรหัสจากดาวเทียมเพื่อนำมาประมวลผลให้เหมาะสมกับการใช้งานในรูปแบบต่างๆ

### 2.10.2 การทำงานของจีพีเอส[11]

1. ดาวเทียมหลักของจีพีเอสทั้งหมด 21 ดวง และอีก 3 ดวงเป็นดาวเทียมสำรอง
2. ดาวเทียมแต่ละตัวนั้นจะประกอบไปด้วยระบบคอมพิวเตอร์ที่เวลาในระบบจะไม่หยุดเดินและส่งสัญญาณวิทยุได้ตลอดจนกว่าดาวเทียมดวงนั้นจะหมดอายุและร่วงลงสู่พื้นโลก
3. ดาวเทียมทุกดวงจะโคจรตามวงโคจรของตนเองและเวลาที่ใช้ในดาวเทียมดวงนั้นๆก็จะถูกตั้งค่าไว้แล้ว ซึ่งการถ่ายทอดตำแหน่งและเวลาของดาวเทียมแต่ละดวงยังสามารถเปลี่ยนแปลงได้อย่างต่อเนื่อง
4. อุปกรณ์รับสัญญาณจีพีเอสทุกตัวจะประกอบไปด้วยจอที่สามารถแสดงแผนที่โลกได้ โดยตำแหน่งทางภูมิศาสตร์ที่ได้ก็จะไปแสดงให้สอดคล้องกับตำแหน่งบนแผนที่
5. อุปกรณ์รับสัญญาณจีพีเอสที่มีความสามารถเพิ่มเติม สามารถรับสัญญาณจากดาวเทียมได้ครบทั้ง 4 ดวงโดยข้อมูลที่ได้จะแสดงถึงความสูงของตำแหน่งนั้นด้วย
6. ถ้าเกิดการเคลื่อนที่ขึ้นอุปกรณ์รับสัญญาณจีพีเอสสามารถที่จะคำนวณความเร็วและทิศทางในการเคลื่อนที่ได้
7. ระบบจีพีเอสที่มีความสามารถสูงจะสามารถระบุสถานที่เป้าหมายที่ต้องการไปและคำนวณเวลาในการเดินทางได้

### 2.10.3 ระบบทางภูมิศาสตร์ (Geographic coordinate system) [12]

- ระบบภูมิศาสตร์เป็นระบบที่ใช้กำหนดจุดต่างๆบนพื้นโลกด้วยตัวเลข 3 ชุด คือ ความสูงจากระดับน้ำทะเลปานกลาง (elevation) และตัวเลขอีก 2 ชุดเป็นพิกัดแนวระนาบเป็นระยะเชิงมุมจากเส้นศูนย์สูตรละติจูด (Latitude) และค่าระยะเชิงมุมจากเส้น Prime Meridian เป็น เส้นลองจิจูด
- เส้นศูนย์สูตร (Equator) กำหนดขึ้นจากแนวระดับที่ตัดผ่านศูนย์กลางของโลกและตั้งฉากกับแกนหมุนซึ่งจะแบ่งโลกออกเป็น 2 ส่วน คือ ซีกโลกเหนือ กับ ซีกโลกใต้ เส้นละติจูด (Latitude) จะเริ่มจากเส้นศูนย์สูตรเริ่มจาก 0 องศา ไปจนถึงขั้วโลกเป็น 90 องศาเหนือ และองศาใต้ ดังแสดงรูปที่ 2.12
- เส้นพรมเมริเดียน (Prime Meridian) เป็นเส้นที่กำหนดขึ้นโดยลากผ่านเมืองGreenwich ประเทศอังกฤษจากขั้วโลกเหนือไปยังขั้วโลกใต้ โดยจะแบ่งเป็นซีกตะวันออก และซีกตะวันตกเส้น ละติจูดจะเริ่มจากเส้นพรมเมริเดียน เป็น 0 องศา และไปสุดที่ 180 องศาตะวันออกและตะวันตก



รูปที่ 2.12 เส้นแบ่งซีกโลก

ที่มา: <https://www.pinterest.com>

#### 2.10.4 รูปแบบพิกัด

hddd.dddddd คือ องศาและจุดทศนิยมขององศา (Decimal)

hdd mm.mmm คือ องศา ลิปดา (arcsemiunte) และทศนิยมของลิปดา

hdd mm.ss.s คือ องศา ลิปดา ฟลิปดา (arcsecond) ทศนิยมของฟลิปดา

การแปลงหน่วยจาก hdd mm.ss.s ไปเป็น hddd.dddddd

ทำได้ตามสูตร ดังสมการที่ 2.7

$$d + ( m + ( s / 60 ) ) / 60 )$$

(2.7)

ตัวอย่างเช่น 1 องศา 23 ลิปดา 45.6 ฟลิปดา 1 23' 45.6"

$$= 1 + ( 23 + ( 45.6 / 60 ) ) / 60$$

$$= 1.396$$

#### 2.10.5 การคำนวณหาตำแหน่งจุดบนพื้นโลก [13]

ตำแหน่งที่ต้องการทราบ	X,Y,Z
ตำแหน่งของดาวเทียมดวงที่ 1	X <sub>1</sub> ,Y <sub>1</sub> ,Z <sub>1</sub>
ตำแหน่งของดาวเทียมดวงที่ 1	X <sub>2</sub> ,Y <sub>2</sub> ,Z <sub>2</sub>
ตำแหน่งของดาวเทียมดวงที่ 1	X <sub>3</sub> ,Y <sub>3</sub> ,Z <sub>3</sub>
ตำแหน่งของดาวเทียมดวงที่ 1	X <sub>4</sub> ,Y <sub>4</sub> ,Z <sub>4</sub>
เวลาความผิดพลาดบนดาวเทียมกับพื้นโลก	t <sub>0</sub>
เวลาจากดาวเทียมดวงที่ 1 เดินทาง	t <sub>1</sub>
เวลาจากดาวเทียมดวงที่ 2 เดินทาง	t <sub>2</sub>
เวลาจากดาวเทียมดวงที่ 3 เดินทาง	t <sub>3</sub>
เวลาจากดาวเทียมดวงที่ 4 เดินทาง	t <sub>4</sub>
ความเร็วของสัญญาณ	C

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะแสดงสมการการระยะทางระหว่างดาวเทียม 4 ดวงกับจุดที่ต้องการทราบ ดังสมการที่ 2.8 สมการที่ 2.9 สมการที่ 2.10 และ สมการที่ 2.11

$$(X - X_1)^2 + (Y - Y_1)^2 + (Z - Z_1)^2 = (C^*(T_1 - T_0))^2 \quad (2.8)$$

$$(X - X_2)^2 + (Y - Y_2)^2 + (Z - Z_2)^2 = (C^*(T_2 - T_0))^2 \quad (2.9)$$

$$(X - X_3)^2 + (Y - Y_3)^2 + (Z - Z_3)^2 = (C^*(T_3 - T_0))^2 \quad (2.10)$$

$$(X - X_4)^2 + (Y - Y_4)^2 + (Z - Z_4)^2 = (C^*(T_4 - T_0))^2 \quad (2.11)$$

### 2.10.6 ปัจจัยที่ส่งผลต่อสัญญาณจีพีเอส [10]

1. สัญญาณกระจายหลายเส้นทาง สัญญาณจากจีพีเอสอาจสะท้อนจากวัตถุต่างๆ เช่น ตึกสูง ภูเขาสูงใหญ่ที่อยู่ระหว่างตัวส่งสัญญาณและตัวรับสัญญาณ ซึ่งจะเพิ่มเวลาในการเดินทางของสัญญาณ และก่อให้เกิดความผิดพลาดของสัญญาณได้
2. ความล่าช้าของชั้นบรรยากาศไอโอโนสเฟียร์และโทรโพสเฟียร์ เป็นชั้นบรรยากาศที่บางที่ประกอบไปด้วยแก๊สที่แตกตัวเป็นประจุบวกและลบ ทำให้การส่งสัญญาณจีพีเอสล่าช้ากว่าปกติได้
3. ความผิดพลาดของการโคจร ตำแหน่งของดาวเทียมที่รายงานไปอาจไม่แม่นยำ
4. ความผิดพลาดจากนาฬิกาตัวรับสัญญาณ นาฬิกาในตัวรับสัญญาณอาจมีการระบุนเวลาผิดพลาดเล็กน้อยเนื่องจากแม่นยำน้อยกว่านาฬิกาในดาวเทียมจีพีเอส

### 2.10.7 โพรโทคอล NMEA – 0183 [14]

NMEA ย่อมาจาก Nation Maritime Electronics Association เป็นโพรโทคอลมาตรฐานที่นำมาใช้งานโดยเครื่องรับสัญญาณ GPS โพรโทคอล ค่อนข้างมีข้อจำกัดและจะเน้นทางการสื่อสารระหว่าง Loran C กับ Autopilot ดังนั้นดั่งนั้น ได้มีการพัฒนาให้สามารถใช้งานได้อย่างกว้างขวางขึ้น โดยครอบคลุมอุปกรณ์อิเล็กทรอนิกส์ในการเดินเรือจนกลายมาเป็น NMEA-0183 ใช้อักษร ASCII และการสื่อสารข้อมูลแบบอนุกรมในการส่งข้อมูล จากอุปกรณ์ตัวหนึ่งไปยังอุปกรณ์รับตัวหนึ่งหรือหลายๆตัว โพรโทคอล NMEA-0183 จะใช้มาตรฐาน ตัวอักษรที่ใช้คือ ASCII Text โดยโพรโทคอล NMEA นั้นจะส่งข้อมูลด้วยอัตราเร็วประมาณ 4800 bps

### 2.11 แอปพลิเคชัน Thinkable [16]

Thinkable เป็นเครื่องมือสร้างโมบายแอปพลิเคชัน เพื่อติดตั้งบนสมาร์ตโฟนที่ใช้ระบบปฏิบัติการ Android เครื่องมือที่ใช้ในการสร้าง Thinkable นั้นนอกจากเครื่องมือพื้นฐานแล้วยังมีการเชื่อมต่อไปยังผลิตภัณฑ์จาก กูเกิล (Google), ทวิตเตอร์ (Twitter) และ ไมโครซอฟท์ (Microsoft) โดยชุดคำสั่งหลังจากที่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบหน้าจอด้วยเครื่องมือต่าง ๆ Thinkable คือเว็บไซต์ที่ให้เราสามารถสร้างโมบายแอปพลิเคชัน สวยๆ ใช้งานได้ ดังแสดงรูปที่ 2.14



thinkable

### รูปที่ 2.13 แอปพลิเคชัน Thinkable

ที่มา: <https://www.scimath.org>

#### 2.11.1 เครื่องมือที่น่าสนใจใน Thinkable

##### 1. Media

- Speech Recognizer เป็นการรู้จำเสียงพูด จาก กูเกิล (Google)
- Text-to-Speech การแสดงเสียงจากข้อความ

##### 2. Sensor

- Accelerometer ใช้งานด้านความเร่ง
- Barcode Scanner ใช้งานด้านเครื่องอ่านบาร์โค้ด
- Clock ใช้งานด้านเวลา
- Gyroscope ใช้งานด้านวัดการหมุน
- Location Sensor ใช้งานด้านตำแหน่ง
- NFC Sensor ใช้งานด้านเทคโนโลยีสื่อสารไร้สายระยะใกล้
- Orientation Sensor ใช้งานด้านการปรับของสมาร์ทโฟน
- Pedometer ใช้งานด้านการนับก้าว
- Proximity Sensor ใช้งานด้านวัดความใกล้ชิด

##### 3. Social

- Twitter เป็นการใช้บริการสังคมออนไลน์ของทวิตเตอร์

##### 4. Visualization

- Google Maps เป็นการใช้บริการด้านแผนที่ของกูเกิล

##### 5. Artificial Intelligence

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Emotion Recognizer เป็นการวิเคราะห์ ของ Microsoft
- Image Recognizer เป็นการวิเคราะห์ภาพ ของ Microsoft
- 6. LEGO MINDSTORMS สำหรับควบคุมอุปกรณ์หุ่นยนต์ LEGO
- 7. Experimental
- Firebase DB เป็นการติดต่อกับฐานข้อมูล Firebase ของ Google
- Spreadsheets เป็นการติดต่อไฟล์ตารางทำการออนไลน์ ของ Google

### 2.11.2 การนำเอาแอปพลิเคชัน Thinkable ไปใช้งาน

- โปรแกรมคำนวณตัวเลขอย่างง่าย
- โปรแกรมสุ่มตัวเลข
- โปรแกรมฝึกเขียนตัวอักษร
- โปรแกรมฝึกวาดภาพ
- โปรแกรมประยุกต์ด้านงานต่างๆ ที่ใช้ฐานข้อมูล
- โปรแกรมนับการเดิน



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





รูปที่ 3.2 บล็อกไดอะแกรมเปอร์เซ็นต์แบตเตอรี่

### 3.2 การคำนวณหาแบตเตอรี่ที่ใช้ในรถเครื่องตัดหญ้า

เลือกใช้มอเตอร์ที่ใช้ในการตัดหญ้าแบบเป็นจำนวน 3 ตัว 12 โวลต์กำลังไฟฟ้าขนาด 50 วัตต์ โดยมีความเร็ว 8000 รอบต่อนาที และมีความเร็วพอนี่เหวี่ยงเอ็นตัดหญ้าที่มีความยาว 20 เซนติเมตร ให้สามารถตัดหญ้าได้ จะคำนวณหากระแสได้ดังสมการที่ 3.1

$$P=IV \quad (3.1)$$

$$I = \frac{P}{V} \quad (3.2)$$

จะได้

$$I_{m1} = \frac{50}{12} \quad (3.3)$$

$$I_{m1} = 4.167 \text{ A}$$

เนื่องจากมอเตอร์ตัดหญ้ามีจำนวน 3 ตัว จึงนำไปคูณ 3 จะได้ดังสมการต่อไปนี้

$$I_{m123} = I_{m1} \times 3$$

$$I_{m123} = 4.167 \times 3$$

จะได้กระแสทั้งหมดของมอเตอร์ตัดหญ้า ดังนี้

$$I_{m123} = 12.501 \text{ A}$$

มอเตอร์ขับเคลื่อน ขนาด 120 w จะคำนวณหา กระแสได้ดังนี้ (I)

จากสมการที่ (3.2) จะได้

$$I = \frac{120}{12}$$

$$I_2 = 10 \text{ A}$$

ดังนั้น

$$I_1 + I_2 = 10 + 12.501 \text{ A} = 20.501 \text{ A}$$

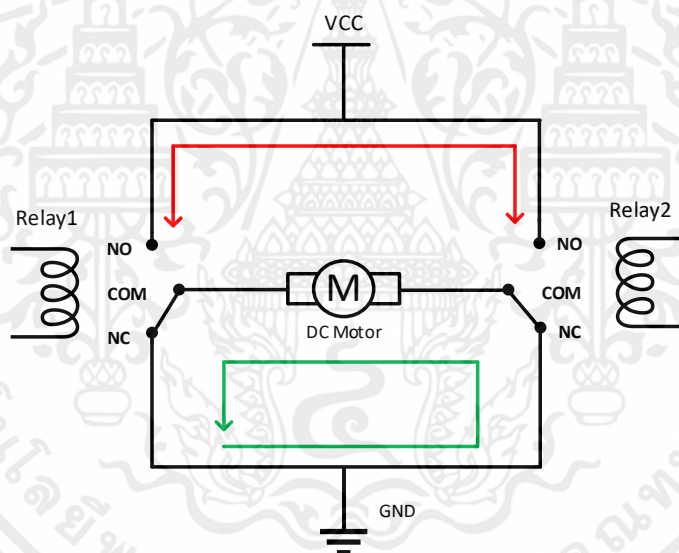
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 วงจรควบคุมมอเตอร์ขับเคลื่อน

วิธีการควบคุมทิศทางมอเตอร์อย่างง่ายด้วยการใช้รีเลย์ (Relay) 2 ชุด กลับทิศทางของมอเตอร์เราสามารถควบคุมทิศทางมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) ได้อย่างง่ายดาย โดยการใช้รีเลย์ที่มีหน้าสัมผัส NO NC COM มาต่อกันจำนวน 2 ตัว ดังแสดงในรูปที่ 3.3 เป็นที่ทราบกันดีอยู่แล้วว่ามอเตอร์ไฟฟ้ากระแสตรงนั้นจะสามารถกลับทิศทางการหมุนด้วยการกลับขั้วแหล่งจ่าย ซึ่งเป็นพื้นฐานการ ควบคุมมอเตอร์แบบไฮบริดจ์ (H-Bridge) นั่นเอง

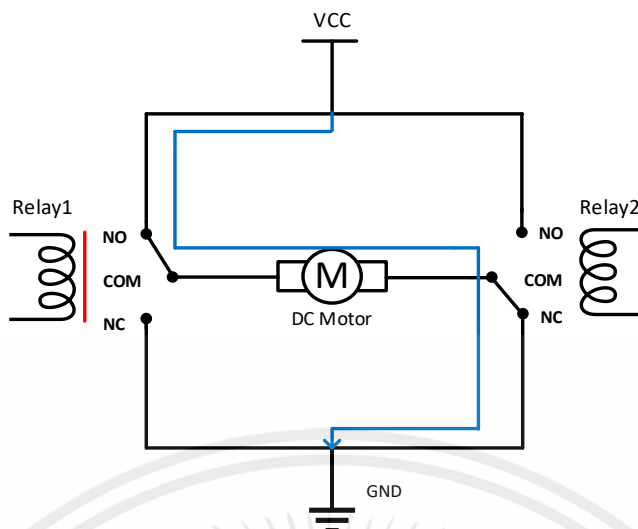
#### 3.3.1 หลักการทำงานของมอเตอร์

เมื่อไม่ได้ทำการจ่ายไฟเข้าไปขั้วรีเลย์ทั้งสองตัว ขั้วมอเตอร์จะถูกต่อกับหน้าสัมผัส NC ลงกราวด์ไปทั้งสองสาย ซึ่งจะทำให้มอเตอร์อยู่ในสภาวะเบรก (Brake) โดยเมื่อมอเตอร์มีแรงหมุนจากภายในมอเตอร์จะเปลี่ยนสถานะในตัวเองของมอเตอร์เป็นสภาวะไดนาโมเกิดกระแสไหลครบวงจร ในตัวเอง ทำให้มอเตอร์เกิดการหน่วงกว่าปกติเมื่อเปรียบเทียบกับ การปล่อยสายลอยไว้



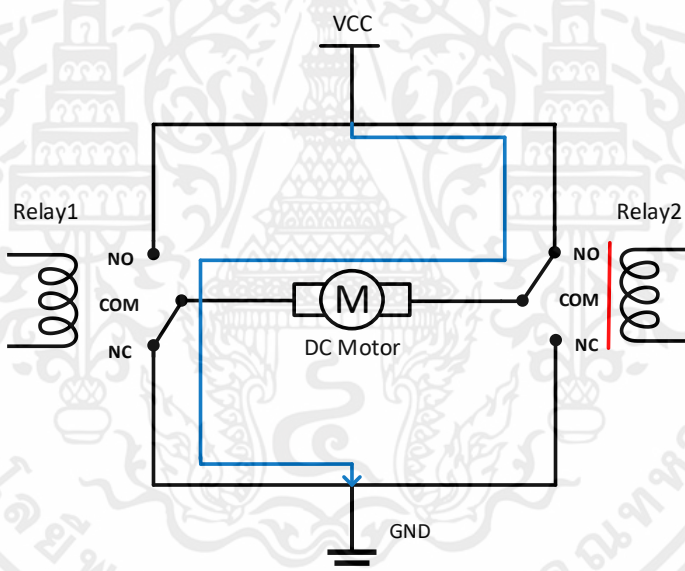
รูปที่ 3.3 วงจรรีเลย์ควบคุมการเคลื่อนที่

เมื่อมีการขับรีเลย์ 1 จะทำให้กระแสวิ่งจากแหล่งจ่าย VCC ไปตามทิศทางดั่งเส้นสีน้ำเงิน แสดงดังรูปที่ 3.4 ทำให้มอเตอร์หมุนไปยังทิศทางหนึ่ง



รูปที่ 3.4 การทำงานของวงจรรีเลย์ตัวที่ 1

เมื่อทำการเปิดรีเลย์ 2 และ ปิดรีเลย์ 1 ทำให้กระแสวิ่งผ่านมอเตอร์ผ่านเส้นสีน้ำเงิน แสดงดังรูปที่ 3.5 จะทำให้กระแสไหลกลับทิศทางในมอเตอร์ ส่งผลให้มอเตอร์หมุนสลับทิศทาง

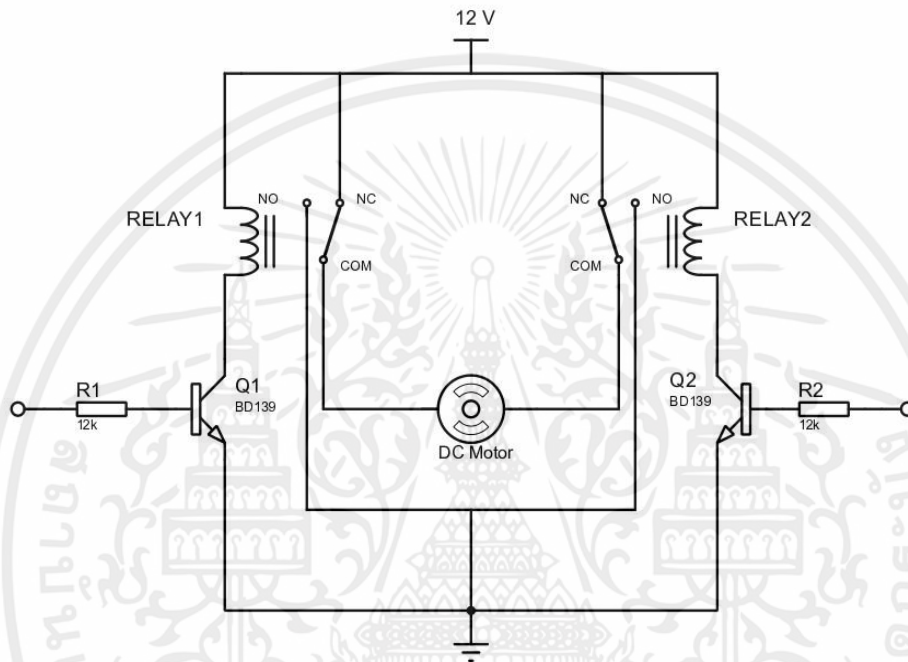


รูปที่ 3.5 การทำงานของวงจรรีเลย์ตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ออกแบบวงจรขับเคลื่อน

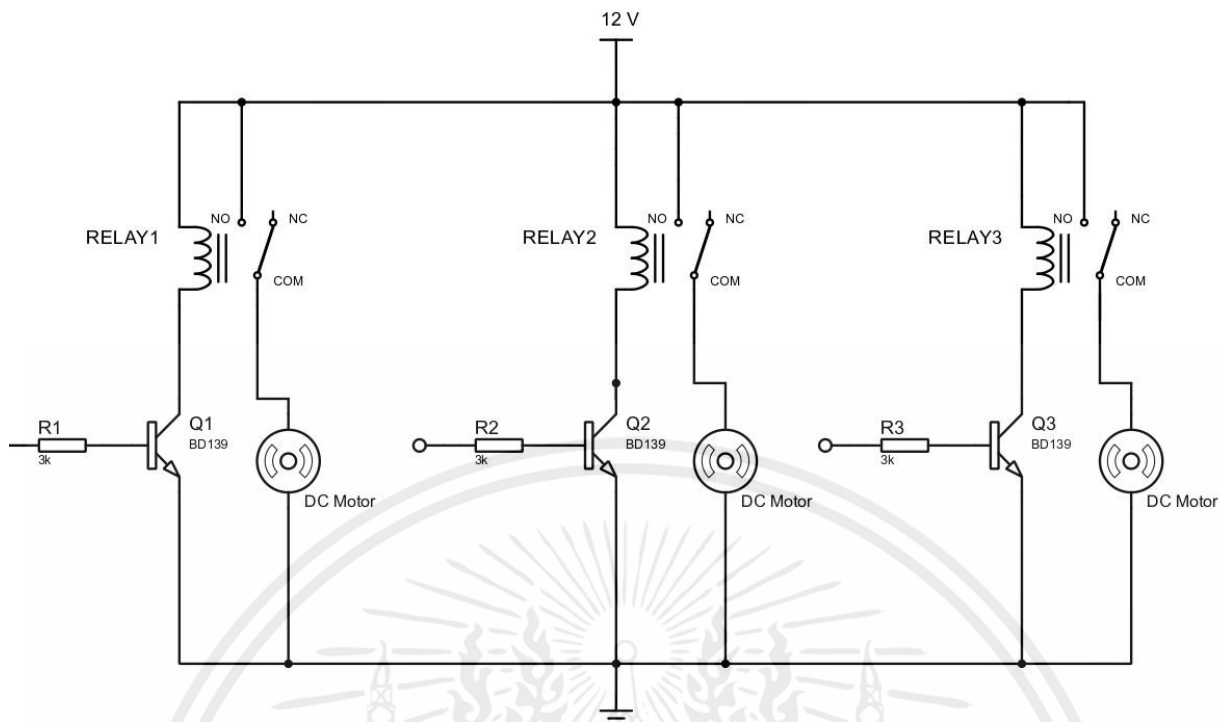
การทำงานของมอเตอร์ขับเคลื่อน เมื่อมีสัญญาณจาก Esp 32 ปล่อยลอจิกมาไบอัสที่ขาเบส (Base) ของทรานซิสเตอร์ทำให้ทรานซิสเตอร์ทำงานดังนั้น กระแสจึงไหลผ่านรีเลย์ตัวที่ 1 ทำให้รีเลย์เปลี่ยนสถานะจาก NC มาเป็น NO ทำให้รีเลย์หมุนไปข้างหน้า และหากมีการไบอัสที่ขาเบสของรีเลย์ตัวที่ 2 จะทำให้มอเตอร์หมุนกลับทิศไปข้างหลัง แสดงดังรูปที่ 3.6



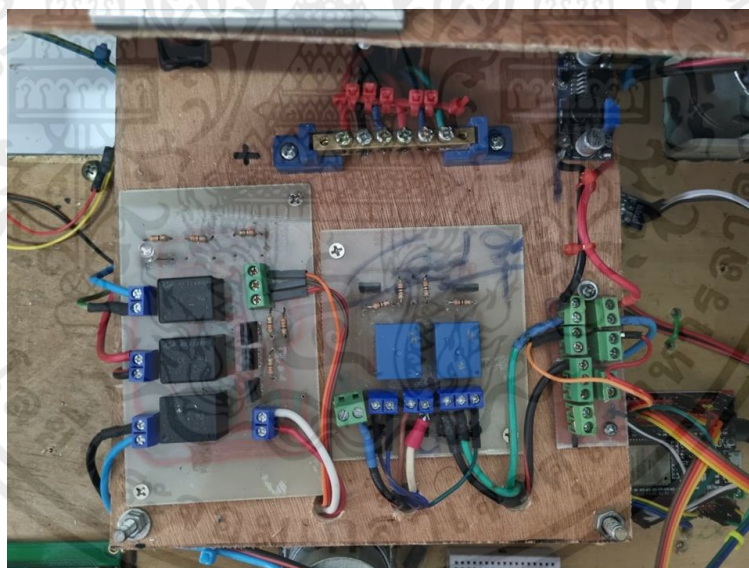
รูปที่ 3.6 วงจรขับมอเตอร์ขับเคลื่อน

### 3.5 วงจรขับมอเตอร์เบรค

เมื่อมีสัญญาณจาก ESP 32 ปล่อยลอจิกมาไบอัสที่ขาเบส (Base) ทรานซิสเตอร์ทำให้ทรานซิสเตอร์ทำงานดังนั้น กระแสจึงไหลผ่านรีเลย์ ทำให้รีเลย์เปลี่ยนสถานะจาก NC มาเป็น NO ทำให้มอเตอร์ตัดทำงาน ดังรูปที่ 3.7



รูปที่ 3.7 วงจรขับมอเตอร์ตัดหญ้า



รูปที่ 3.8 วงจรขับเคลื่อนและวงจรมอเตอร์ตัดหญ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 โครงสร้างของเครื่องตัดหญ้า



รูปที่ 3.9 โครงสร้างเครื่องตัดหญ้า

การออกแบบโดยให้เครื่องตัดหญ้าขับเคลื่อนด้วยล้อหลังโดยใช้มอเตอร์กระแสตรง 12V และในส่วนของ การเลี้ยวจะใช้เป็นเซอร์โวมอเตอร์ในการเลี้ยวโดยปรับไปที่องศาที่ต้องการจะให้เลี้ยว แสดงดังรูปที่ 3.9

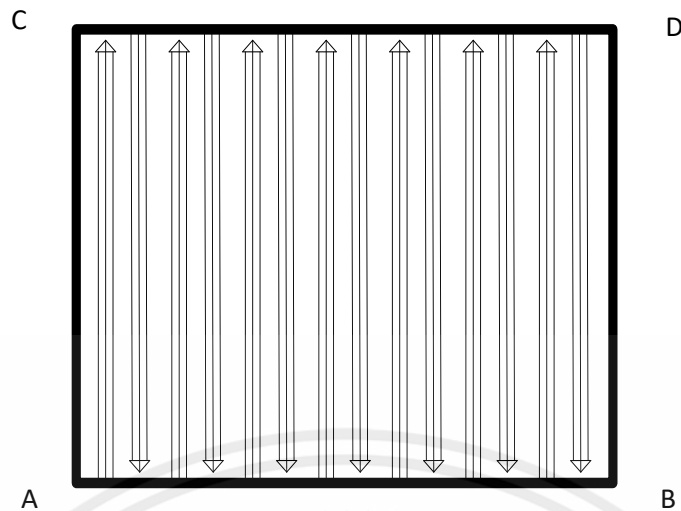
#### 3.6.1 ส่วนประกอบของเครื่องตัดหญ้า

- Esp32
- GPS module NE0-8m
- Module เซ็มทิส Gy-271
- Module วัดโวลต์ 0 - 25 v
- โซ่ t8f
- มอเตอร์ตัดหญ้า 3 ตัว Rs775 ขนาดกำลังไฟฟ้า 50 W ความเร็วรอบ 8,000 รอบต่อนาที
- เซอร์โวมอเตอร์ ขนาดแรงบิด 20 kg ใช้ไฟเลี้ยง 4.8-6.8V แรงบิดสูงสุด 20 kg.cm ที่ไฟเลี้ยง 6.8v หมุนได้ 0-180 องศา
- มอเตอร์ขับเคลื่อน ขนาดกำลังไฟฟ้า 120 W ความเร็วรอบ 1500 รอบต่อนาที- แบตเตอรี่ 12v 12Ah

### 3.7 หาเส้นทางในการตัด โดยกรอกพิกัด 4 จุด

โดยเครื่องตัดหญ้าจะเริ่มที่จุด A ดังรูปที่ 3.10 คอนโทรลเลอร์จะคำนวณจากจุดที่เริ่มไประยะทางต่อไป หน่วยเป็นเมตรและคำนวณองศา แล้วเลี้ยวไปยังปลายทาง เมื่อถึงจุดเป้าหมายจะคำนวณหาจุดต่อไป แล้วถอยหลังรูปแบบการวิ่งของหุ่นยนต์ตัดหญ้าจีพีเอสโดยรับค่า ละติจูด (Latitude) และ ลองจิจูด (Longitude) มาจาก แอปพลิเคชัน 4 จุด ABCD ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

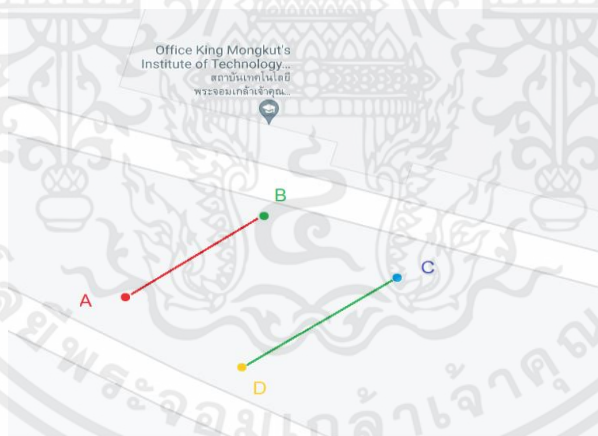


รูปที่ 3.10 รูปแบบการวิ่งของหุ่นยนต์ตัดหญ้าจีพีเอส

มีการประมวลผลโดย Esp32 เพื่อหาจุดเลี้ยวและหาระยะทางการวิ่งจากละติจูดและลองจิจูด ที่รับมาจาก GPS module NE0-8m แล้วนำมาประมวลผลหาเส้นทาง

### 3.7.1 รับค่าจากแอปพลิเคชัน

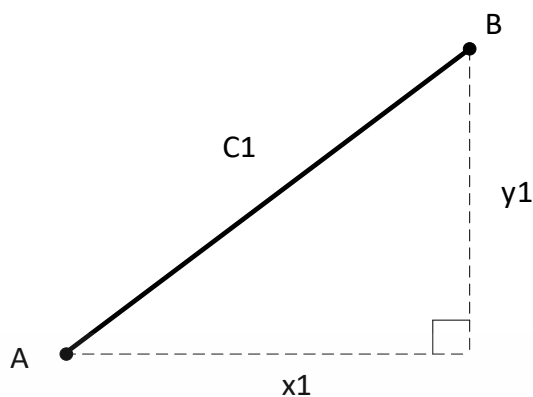
รับค่ามาจากแอปพลิเคชัน Control Car เป็นค่าละติจูด และลองจิจูด เป็นรูปสี่เหลี่ยมจัตุรัส หรือสี่เหลี่ยมผืนผ้า โดยจะกำหนดเป็น ABCD จากนั้นจะนำพิกัดไปคำนวณหาเส้นทาง ดังรูปที่ 3.11



รูปที่ 3.11 ตัวอย่างการแสดงผลการเซตพิกัดจุด 4 จุดจากกูเกิลแมพ

### 3.7.2 การหาระยะทาง

ในการหาระยะทาง (เมตร) ระหว่างจุดสองจุด คือ AB และ DC โดยนำค่า และ ละติจูดมาหาค่าระยะทางโดยใช้ ทฤษฎีบทพีทาโกรัส ดังรูปที่ 3.12



รูปที่ 3.12 ทฤษฎีบทพีทาโกรัส เพื่อหาระยะทาง (เมตร)

โดยจุด A จะได้ Latitude1 และ Longitude1 จุด B จะได้ Latitude2 และ Longitude2 จากนั้นนำ Latitude2 มาลบกับ Latitude1 และ Longitude2 มาลบกับ Longitude1 ได้ตั้งสมการที่ 3.4 และ สมการที่ 3.5

$$x_1 = \text{Latitude2} - \text{Latitude1} \quad (3.4)$$

$$y_1 = \text{Longitude2} - \text{Longitude1} \quad (3.5)$$

หาค่า  $C_1$  ได้จากสมการที่ 3.6

$$C_1 = \sqrt{(x_1)^2 + (y_1)^2} \quad (3.6)$$

หาระยะทางได้จากสมการที่ 3.7

$$D = \left( \frac{C_1}{P} \right) \times 1000 \quad (3.7)$$

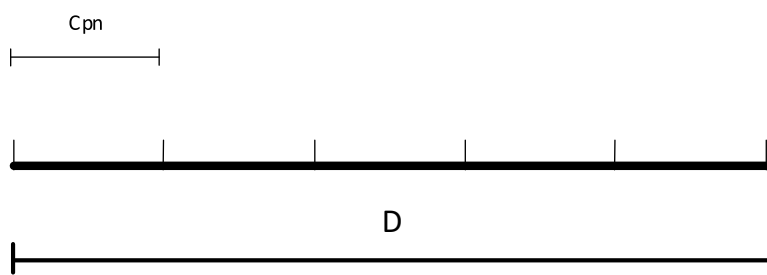
โดย

D คือ ระยะทาง มีหน่วยเป็นเมตร (m)

P มีค่าเท่ากับ  $9.259259 \times 10^{-3}$

### 3.7.3 การหาการวิ่งของรถ

หาจำนวนครั้งที่รถจะวิ่ง  $n$  และคำนวณหาองศาที่จะเลี้ยวดังรูปที่ 3.13



รูปที่ 3.13 แสดงการหา  $n$  จำนวนครั้งที่จะวิ่ง

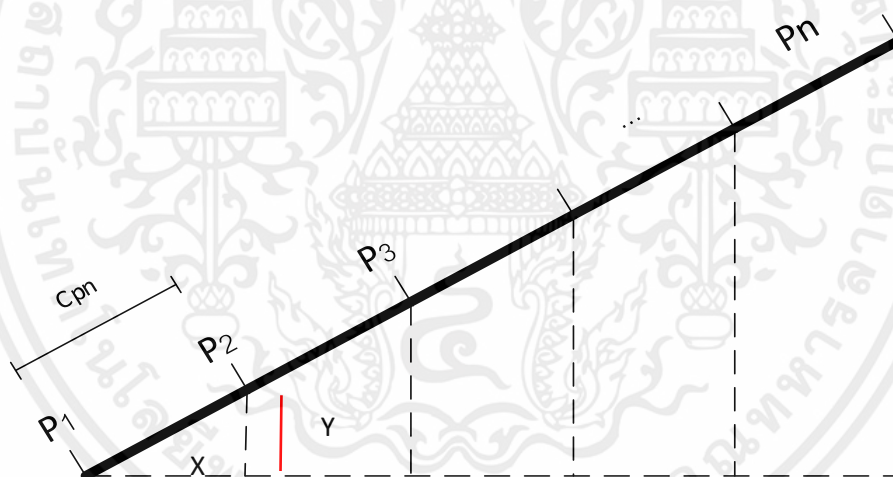
$$\text{channel} = \frac{D}{w} \tag{3.8}$$

โดย

Channel คือ จำนวนครั้งที่วิ่ง

$w$  คือความกว้างของเครื่องตัดหญ้า = 0.6 m

หา ละติจูด และ ลองจิจูด แต่ละจุดของ  $p_n$  จากคุณสมบัติของสามเหลี่ยมดังรูปที่ 3.14



รูปที่ 3.14 ละติจูด และ ลองจิจูด แต่ละจุดของ  $n$  จากสมบัติของสามเหลี่ยมแต่ละจุด

หา  $C_{pn}$  ได้จากสมการที่ 3.9

$$C_{pn} = \frac{C_1}{w} \times n \tag{3.9}$$

โดยที่  $n$  มีค่า 1,2,3.....ช่อง (Channel)

หาความชัน ได้จากสมการที่ 3.10

$$m = \frac{\text{Longitude2} - \text{Longitude1}}{\text{Latitude2} - \text{Latitude1}} \tag{3.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้มุม จากสมการที่ 3.11

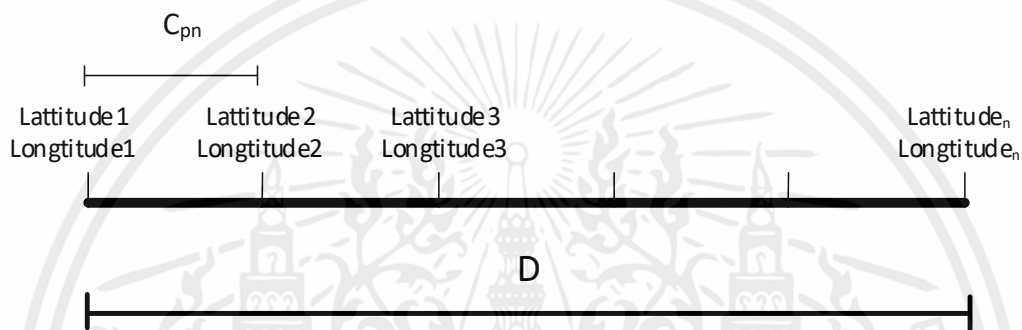
$$\tan^{-1} m = \theta \quad (3.11)$$

หา  $P_n$  ของแต่ละจุดจะได้ จากสมการที่ 3.12 และ สมการที่ 3.13

$$\text{Latitude}_{pn} = (\cos \theta)(C_{pn}) \quad (3.12)$$

$$\text{Longitude}_{pn} = (\sin \theta)(C_{pn}) \quad (3.13)$$

หา  $\text{Latitude}_n$  และ  $\text{Longitude}_n$  ได้จากสมการที่ 3.14 และ สมการที่ 3.15



รูปที่ 3.15 Latitude<sub>n</sub> และ Longitude<sub>n</sub> แต่ละจุด

$$\text{Latitude}_n = \text{Latitude}_{pn} + \text{Latitude}_1 \quad (3.14)$$

$$\text{Longitude}_n = \text{Longitude}_{pn} + \text{Longitude}_1 \quad (3.15)$$

โดย

$C_{pn}$  คือ ความห่างระหว่างพิกัดจุด

$\text{Latitude}_{pn}$  คือ ความยาวของ x แต่ละจุด

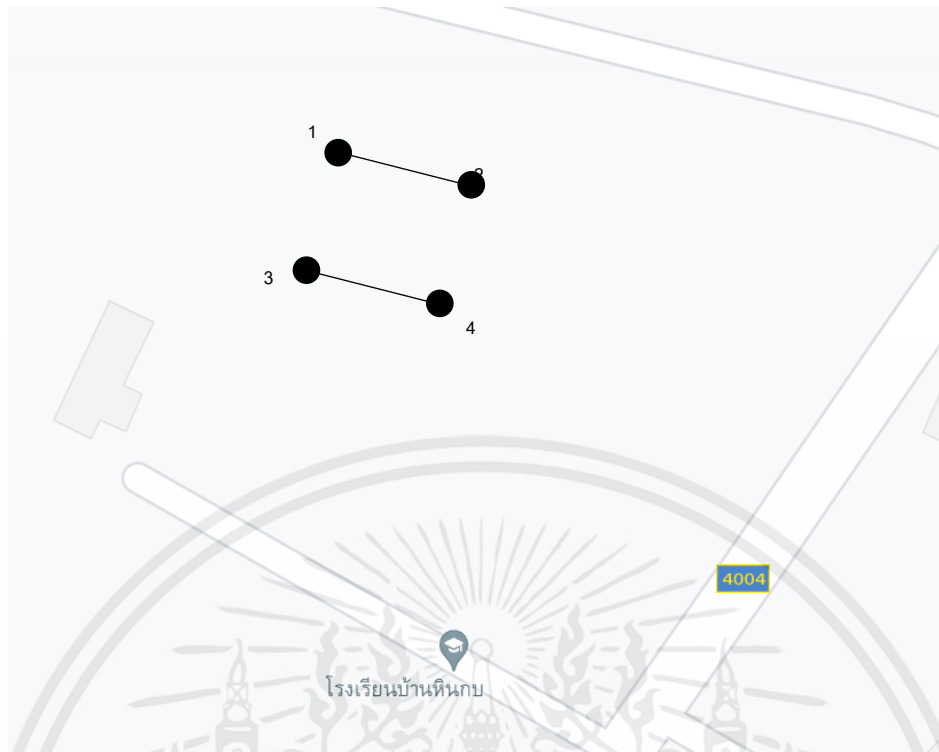
$\text{Longitude}_{pn}$  คือ ความยาวของ y แต่ละจุด

$\text{Latitude}_n$  และ  $\text{Longitude}_n$  คือ จุดแต่ละจุดที่จะต้องวิ่งไป

n คือ จำนวนเต็ม 1,2,3, ...

**ตัวอย่าง** การหาจุด  $\text{Latitude}_n$  และ  $\text{Longitude}_n$

เมื่อ Set ค่า ละติจูด และ ลองจิจูด ทั้ง 4 ที่จุด จากแมพ แสดงดังรูปที่ 3.16



รูปที่ 3.16 ภาพตัวอย่างการทดสอบพิกัด 4 จุด ที่สนามฟุตบอลโรงเรียนบ้านหินกบ

จุดที่ 1 Latitude 10.715668 Longitude 99.379141

จุดที่ 2 Latitude 10.715766 Longitude 99.379159

จุดที่ 3 Latitude 10.715705 Longitude 99.379263

จุดที่ 4 Latitude 10.715611 Longitude 99.379212

คำนวณระยะทาง จากจุดที่ 1 ไปยังจุดที่ 2 แทนค่าในสมการที่ 3.4 และ 3.5 จะได้

$$x_1 = \text{Latitude}_2 - \text{Latitude}_1 \\ 10.715766 - 10.715668 = 0.000106$$

$$y_1 = \text{Longitude}_2 - \text{Longitude}_1 \\ 99.379159 - 99.379141 = 0.000018$$

หาค่า  $C_1$  จากสมการที่ 3.6

$$C_1 = \sqrt{(x_1)^2 + (y_1)^2} = \sqrt{(0.000106)^2 + (0.000018)^2} = 0.000108$$

หาระยะทาง (D) หน่วยเป็นเมตร จากสมการที่ 3.7

$$D = \left(\frac{C_1}{P}\right) \times 1000 = \left(\frac{0.000108}{9.925925 \times 10^{-3}}\right) \times 1000 = 11.88 \text{ เมตร}$$

คำนวณจำนวนครั้งที่จจะแล้ว จากจุดที่ 1 ไปยังจุดที่ 2 ดังสมการที่ 3.8 จะได้

$$\text{channel} = \frac{D}{w} = \frac{11.88}{0.5} = 23 \text{ ช่อง}$$

และหาพิกัด ละติจูด และ ลองจิจูด ทั้ง 23 จุดดังสมการที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่า  $C_1$  มาหารกับความกว้างของใบตัดหญ้า เพื่อหาค่า  $C_{pn}$

$$C_{pn} = \frac{C_1}{w} = \frac{0.000108}{23} \times n$$

โดย  $n$  จะมีค่า 1,2,3,... ไปเรื่อย ๆ แต่จะไม่เกินช่องคือ 23

จะได้ดังนี้

$C_{p1} = 0.000005$	$C_{p9} = 0.000042$	$C_{p17} = 0.000079$
$C_{p2} = 0.000009$	$C_{p10} = 0.000047$	$C_{p18} = 0.000084$
$C_{p3} = 0.000014$	$C_{p11} = 0.000051$	$C_{p19} = 0.000089$
$C_{p4} = 0.000019$	$C_{p12} = 0.000056$	$C_{p20} = 0.000093$
$C_{p5} = 0.000023$	$C_{p13} = 0.000061$	$C_{p21} = 0.000098$
$C_{p6} = 0.000028$	$C_{p14} = 0.000065$	$C_{p22} = 0.00010$
$C_{p7} = 0.000033$	$C_{p15} = 0.000070$	$C_{p23} = 0.00010$
$C_{p8} = 0.000037$	$C_{p16} = 0.000075$	

หาความชันจากสมการที่ 3.9 และ 3.10

$$m = \frac{\text{Longitude2-Longitude1}}{\text{Latitude2-Latitude1}} = \frac{0.000018}{0.000106} = 0.1125$$

$$\theta = \tan^{-1} m = \tan^{-1} 0.1125 = 6.4187^\circ$$

หา  $\text{Latitude}_{pn}$  และ  $\text{Longitude}_{pn}$  จากสมการที่ 3.12 และ 3.13

$\text{Latitude}_1 = 0.000005$	$\text{Latitude}_{13} = 0.000060$
$\text{Latitude}_2 = 0.000009$	$\text{Latitude}_{14} = 0.000065$
$\text{Latitude}_3 = 0.000014$	$\text{Latitude}_{15} = 0.000069$
$\text{Latitude}_4 = 0.000018$	$\text{Latitude}_{16} = 0.000074$
$\text{Latitude}_5 = 0.000023$	$\text{Latitude}_{17} = 0.000078$
$\text{Latitude}_6 = 0.000028$	$\text{Latitude}_{18} = 0.000083$
$\text{Latitude}_7 = 0.000032$	$\text{Latitude}_{19} = 0.000088$
$\text{Latitude}_8 = 0.000037$	$\text{Latitude}_{20} = 0.000092$
$\text{Latitude}_9 = 0.000041$	$\text{Latitude}_{21} = 0.000097$
$\text{Latitude}_{10} = 0.000046$	$\text{Latitude}_{22} = 0.000101$
$\text{Latitude}_{11} = 0.000051$	$\text{Latitude}_{23} = 0.000106$
$\text{Latitude}_{12} = 0.000055$	

$$\text{Latitude}_{pn} = (\cos \theta)(C_{pn}) = (\cos 6.4187)(C_{pn})$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ดังนี้

$$\text{Longitude}_{pn} = (\sin \theta)(C_{pn}) = (\sin 6.4187)(C_{pn})$$

จะได้ดังนี้

Longitude <sub>1</sub> = 0.000000	Longitude <sub>13</sub> = 0.000009
Longitude <sub>2</sub> = 0.000001	Longitude <sub>14</sub> = 0.000010
Longitude <sub>3</sub> = 0.000002	Longitude <sub>15</sub> = 0.000011
Longitude <sub>4</sub> = 0.000002	Longitude <sub>16</sub> = 0.000012
Longitude <sub>5</sub> = 0.000003	Longitude <sub>17</sub> = 0.000013
Longitude <sub>6</sub> = 0.000004	Longitude <sub>18</sub> = 0.000013
Longitude <sub>7</sub> = 0.000005	Longitude <sub>19</sub> = 0.000014
Longitude <sub>8</sub> = 0.000005	Longitude <sub>20</sub> = 0.000015
Longitude <sub>9</sub> = 0.000006	Longitude <sub>21</sub> = 0.000016
Longitude <sub>10</sub> = 0.000007	Longitude <sub>22</sub> = 0.000017
Longitude <sub>11</sub> = 0.000008	Longitude <sub>23</sub> = 0.00001
Longitude <sub>12</sub> = 0.000009	

หา Latitude<sub>n</sub> และ Longitude<sub>n</sub>

$$\text{Latitude}_n = \text{Latitude}_{pn} + \text{Latitude}_1$$

$$\text{Longitude}_n = \text{Longitude}_{pn} + \text{Longitude}_1$$

โดยการนำ Latitude<sub>pn</sub> มาบวกกับ Latitude<sub>1</sub> และ Longitude<sub>pn</sub> มาบวกกับ Longitude<sub>1</sub> ทุกค่าจะได้  
ออกมาจำนวน 23 ค่าตามค่าของช่องจะได้พิกัดจุดทุกจุดที่รถตัดหญ้าต้องวิ่งไป ดังนี้

Latitude <sub>1</sub> = 10.715660	Longitude <sub>1</sub> = 99.379141
Latitude <sub>2</sub> = 10.715665	Longitude <sub>2</sub> = 99.379142
Latitude <sub>3</sub> = 10.715669	Longitude <sub>3</sub> = 99.379143
Latitude <sub>4</sub> = 10.715674	Longitude <sub>4</sub> = 99.379143
Latitude <sub>5</sub> = 10.715678	Longitude <sub>5</sub> = 99.379144
Latitude <sub>6</sub> = 10.715683	Longitude <sub>6</sub> = 99.379145
Latitude <sub>7</sub> = 10.715688	Longitude <sub>7</sub> = 99.379146
Latitude <sub>8</sub> = 10.715692	Longitude <sub>8</sub> = 99.379146

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Latitude <sub>9</sub> = 10.715697	Longitude <sub>9</sub> = 99.379147
Latitude <sub>10</sub> = 10.715701	Longitude <sub>10</sub> = 99.379148
Latitude <sub>11</sub> = 10.715706	Longitude <sub>11</sub> = 99.379149
Latitude <sub>12</sub> = 10.715711	Longitude <sub>12</sub> = 99.379150
Latitude <sub>13</sub> = 10.715715	Longitude <sub>13</sub> = 99.379150
Latitude <sub>14</sub> = 10.715720	Longitude <sub>14</sub> = 99.379151
Latitude <sub>15</sub> = 10.715725	Longitude <sub>15</sub> = 99.379152
Latitude <sub>16</sub> = 10.715729	Longitude <sub>16</sub> = 99.379153
Latitude <sub>17</sub> = 10.715734	Longitude <sub>17</sub> = 99.379154
Latitude <sub>18</sub> = 10.715738	Longitude <sub>18</sub> = 99.379154
Latitude <sub>19</sub> = 10.715743	Longitude <sub>19</sub> = 99.379155
Latitude <sub>20</sub> = 10.715748	Longitude <sub>20</sub> = 99.379156
Latitude <sub>21</sub> = 10.715752	Longitude <sub>21</sub> = 99.379157
Latitude <sub>22</sub> = 10.715757	Longitude <sub>22</sub> = 99.379157
Latitude <sub>23</sub> = 10.715761	Longitude <sub>23</sub> = 99.379158

โดยจุดที่ 3 และ จุดที่ 4 ก็จะทำเช่นกัน เพื่อหา Latitude<sub>n</sub> และ Longitude<sub>n</sub>

### 3.8 การลดความเร็วรอบของมอเตอร์ขับเคลื่อน

ใช้การทดเฟืองในการลดความเร็วของมอเตอร์ โดยมอเตอร์ กำลัง 120 W มีความเร็วรอบ 1500 รอบต่อนาที จากมอเตอร์ 6 ฟัน ไปยังล้อ 90 ฟัน โดยความเร็วสามารถหาได้จากสมการที่ 3.15

$$\text{rpm} = \frac{N_1 \times \text{rpm}_{\text{ก่อน}}}{N_2} \quad (3.15)$$

$$\text{rpm} = \frac{6 \times 1500}{90} \quad (3.16)$$

$$\text{rpm} = 100 \text{ (รอบต่อนาที)}$$

### 3.9 การคำนวณเปอร์เซ็นต์แบตเตอรี่จากแรงดันแบตเตอรี่

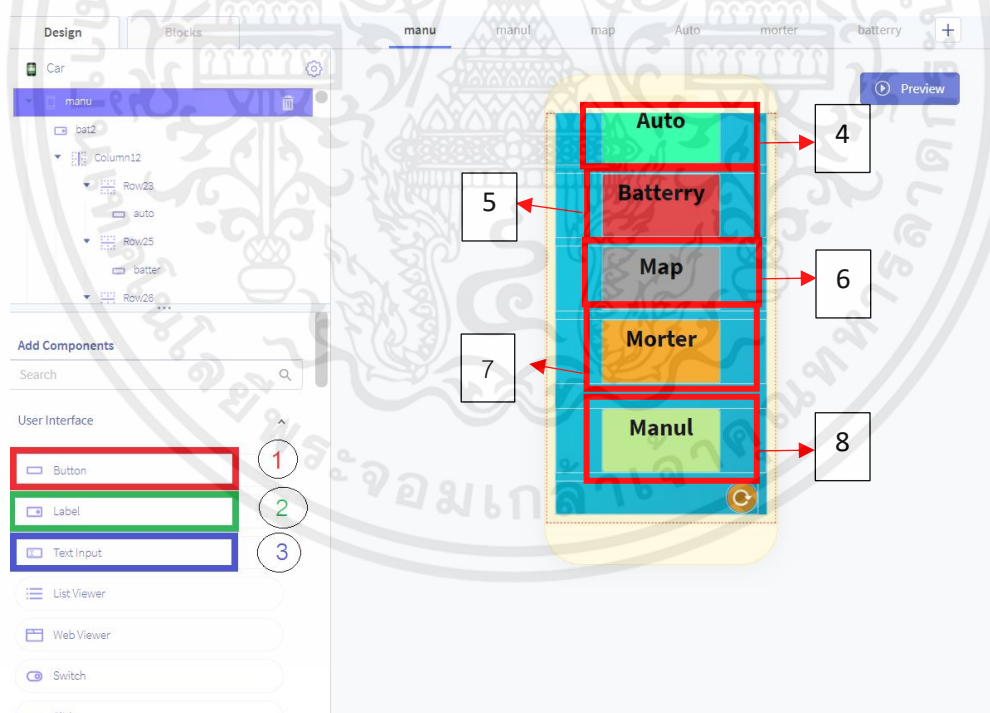
ใช้โวลต์มิเตอร์ 0 - 25 v เพื่อวัดแรงดันของแบตเตอรี่ โดยการใช้การเทียบบัญญัติตรงยาค์เมื่อแรงดันน้อยกว่าหรือเท่ากับ 11 โวลต์ แอปพลิเคชันจะแสดงค่า 0 เปอร์เซ็นต์ และเมื่อเท่ากับ 13 โวลต์ จะแสดงค่า 100 เปอร์เซ็นต์

### 3.10 การออกแบบแอปพลิเคชันแอนดรอยด์ (Application android) โดยใช้ Thunkable ในการเขียน

โดย Thunkable เป็นเว็บช่วยในการเขียนโดยสามารถเลือกปุ่มแล้วจัดตำแหน่งแต่ละจุดต่างๆ โดยมีรูปจำลองในเว็บและสามารถทดลองใช้ในโทรศัพท์ได้เช่นกัน

#### 3.10.1 การออกแบบแอปพลิเคชัน

การออกแบบแอปพลิเคชัน(Application) ด้วย Thunkable บนมือถือโดยมีฟังก์ชันการทำงานของแอปพลิเคชันทั้งหมด 5 ฟังก์ชัน แสดงดังรูปที่ 3.17 คือ ปุ่มอัตโนมัติ (Auto) ปุ่มแบตเตอรี่ (Battery) ปุ่มแผนที่ตำแหน่งระบุตำแหน่ง (Map) ปุ่มมอเตอร์ตัดหญ้า (Motor) ปุ่มบังคับ (Manual) โดยแต่ละดังต่อไปนี้ ปุ่มอัตโนมัติจะให้ผู้ใช้กรอกพิกัดจุด 4 จุด คือค่า ละติจูดและ ลองจิจูด ลงไปในแอปพลิเคชันแสดงดังรูปที่ 3.18 เพื่อให้เครื่องตัดหญ้าทำงานได้ ปุ่มแบตเตอรี่จะแสดงปริมาณของแบตเตอรี่ให้ผู้ใช้ได้ทราบออกมาในรูปแบบเปอร์เซ็นต์ แสดงดังรูปที่ 3.19 ปุ่มแผนที่ระบุตำแหน่งจะแสดงตำแหน่งของเครื่องตัดหญ้าว่าอยู่ตำแหน่งใดบนแผนที่และ เซ็ต (Set) ค่าพิกัด 4 จุด แสดงดังรูปที่ 3.20 ปุ่มมอเตอร์ตัดหญ้า จะให้ผู้ใช้งานกดเปิด/ปิด (ON/OFF) เพื่อให้ใบมีดตัดของเครื่องตัดหญ้าทำงานแสดงดังรูปที่ 3.21 และปุ่มบังคับ จะเป็นการบังคับเองโดยให้ผู้ใช้งาน แสดงดังรูปที่ 3.22



รูปที่ 3.17 โครงสร้างของแอปพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายรูปที่ 3.17

หมายเลข 1 คือคำสั่งปุ่ม

หมายเลข 2 คือคำสั่งโชว์ข้อความ

หมายเลข 3 คือคำสั่งรับค่าข้อมูลเมื่อมีการป้อนข้อความเข้ามา

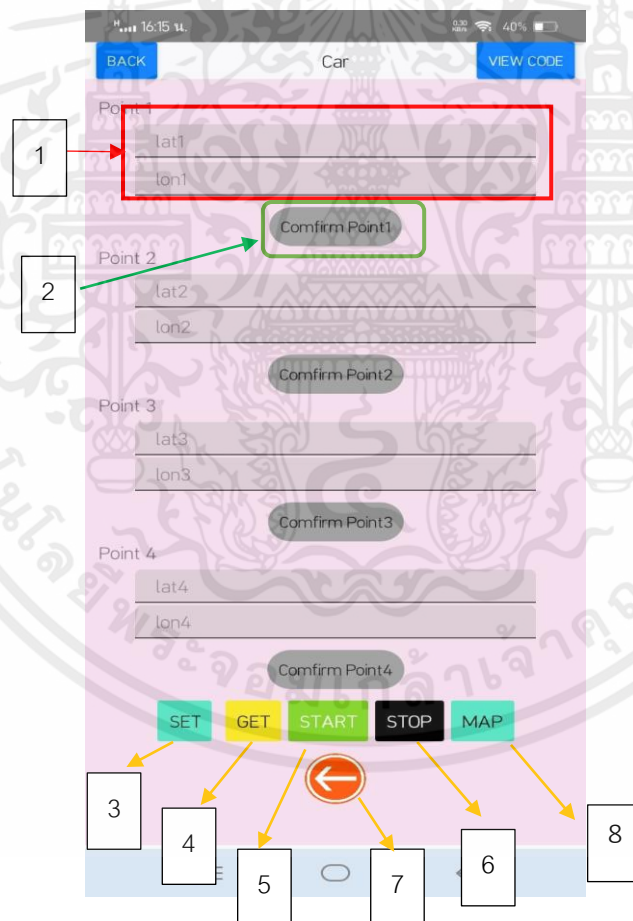
หมายเลข 4 คือปุ่มไปยังหน้าอัตโนมัติ

หมายเลข 5 คือปุ่มไปยังหน้าเบตเตอร์รี่

หมายเลข 6 คือปุ่มไปยังหน้าแผนที่ของตัวเครื่อง

หมายเลข 7 คือปุ่มไปยังหน้าควบคุมมอเตอร์ตัดหญ้า

หมายเลข 8 คือปุ่มไปยังหน้าบังคับ

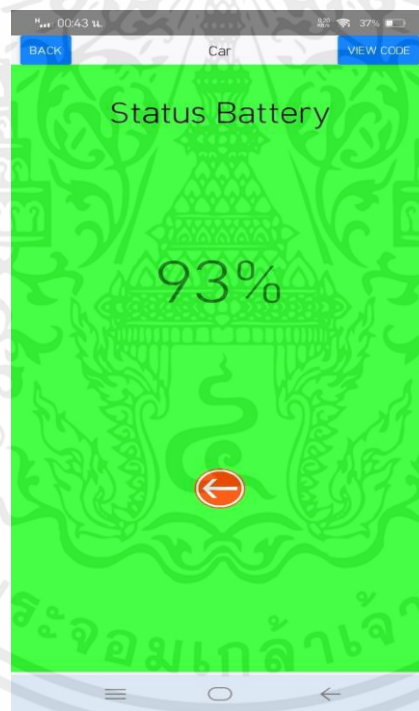


รูปที่ 3.18 การกรอกพิกัด 4 จุดเพื่อเริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

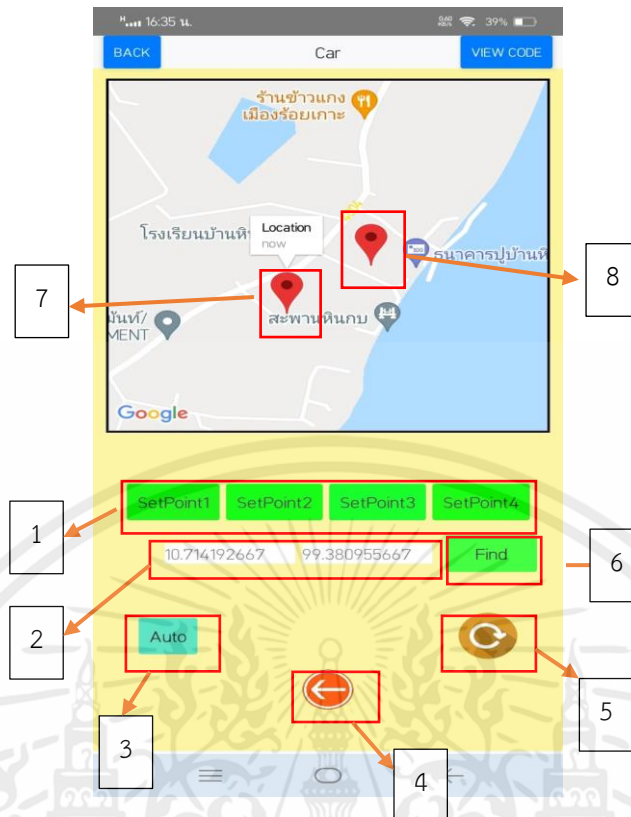
- อธิบาย รูปที่ 3.18
- หมายเลข 1 อินพุท (input) รับค่าจากการพิมพ์
  - หมายเลข 2 ปุ่มส่งค่าจากอินพุทไปยังไฟร์เบส (Firebase)
  - หมายเลข 3 เช็ตค่า ละติจูด และ ลองจิจูด ตอนนี
  - หมายเลข 4 ส่งค่าละติจูด และ ลองจิจูด จากไฟร์เบสทั้ง 4จุดไปยังEsp32
  - หมายเลข 5 ส่งค่าจากไฟร์เบสไปยัง Esp32 เพื่อเริ่มการทำงาน
  - หมายเลข 6 ส่งค่าจากไฟร์เบสไปยัง Esp32 เพื่อหยุดการทำงาน
  - หมายเลข 7 ปุ่มกลับไปยังเมนู
  - หมายเลข 8 ปุ่มไปยังหน้าระบุตำแหน่งเพื่อเลือกเป็นรูปแบบการเช็ตค่าพิกัด ดังรูปที่ 3.20

3.20



รูปที่ 3.19 หน้าแสดงสถานะเปอร์เซ็นต์แบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แสดงพิกัดปัจจุบันของเครื่องตัดหญ้า

อธิบาย รูปที่ 3.20

หมายเลข 1 ปุ่มเซตค่าพิกัด 4 จุด

หมายเลข 2 โข้วค่า ละติจูด และ ลองจิจูด ของเครื่องตัดหญ้า

หมายเลข 3 ปุ่มไปยังหน้าอัตโนมัติเพื่อเลือกเป็นรูปแบบการรอกค่าพิกัด  
ดังรูปที่ 3.18

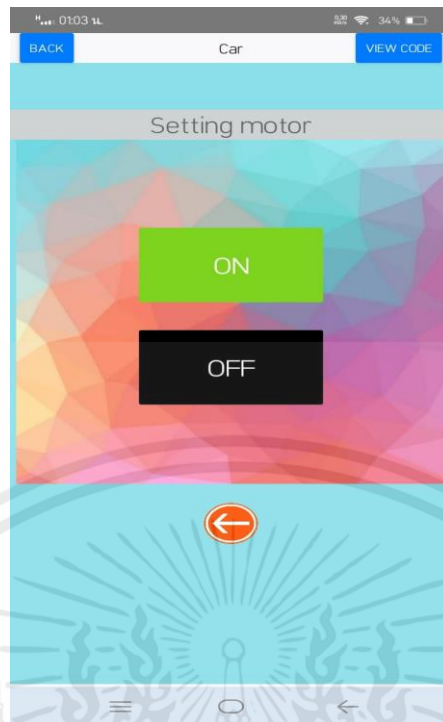
หมายเลข 4 ปุ่มกลับไปยัง หน้า Manu

หมายเลข 5 ปุ่มรีเฟรช เมื่อเซตค่าผิดพลาด

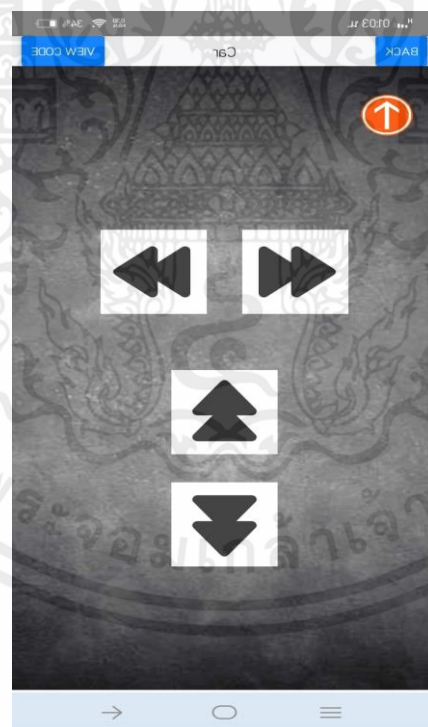
หมายเลข 6 ปุ่มหาพิกัดเครื่องตัดหญ้า

หมายเลข 7 พิกัดปัจจุบันของโทรศัพท์

หมายเลข 8 พิกัดปัจจุบันของเครื่องตัดหญ้า



รูปที่ 3.21 ปุ่มสวิตช์ควบคุมการเปิด-ปิดมอเตอร์ตัดหญ้า

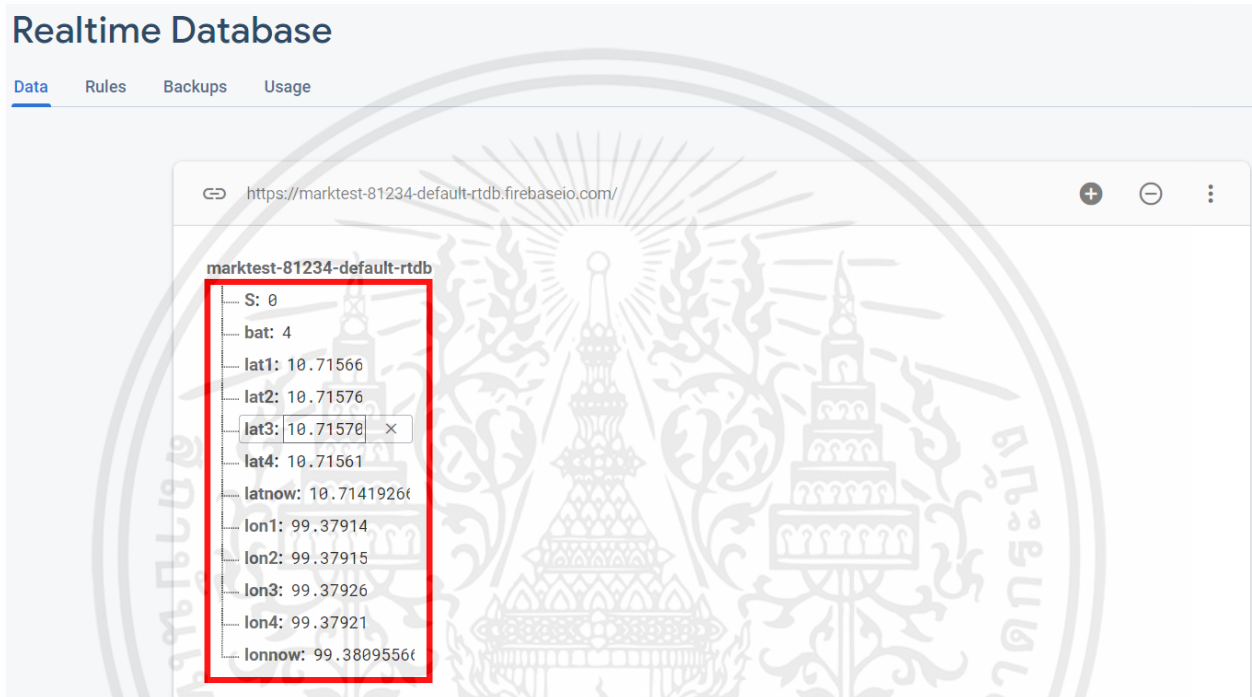


รูปที่ 3.22 หน้าควบคุมแบบบังคับ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.11 ตัวกลางในการรับส่งค่าไฟร์เบส

ทำงานเป็นตัวกลางที่ใช้สื่อสารระหว่างแอปพลิเคชันแอนดรอยด์ กับ Esp32 โดยสามารถส่งแบบเรียลไทม์หรือแบบเก็บข้อมูล โดยจะส่งค่าจากแอปพลิเคชันแอนดรอยด์ ไปเก็บไว้ใน หัวข้อ Bucket ต่างๆ จากนั้น Esp32 จะดึงค่าไปยัง ไฟร์เบส โดยใช้คำสั่ง SetDouble หรือ SetInt ตามด้วย (“หัวข้อ Bucket”) ที่จะดึงข้อมูลจากไฟร์เบส และ Esp32 จะส่งค่ากลับมาด้วยคำสั่ง GetDouble หรือ GetInt ตามด้วย (“ค่าที่ต้องการส่ง”) กลับมายังไฟร์เบส แสดงดังรูปที่ 3.23



รูปที่ 3.23 ไฟร์เบส เก็บข้อมูล หัวข้อ Bucket และข้อมูลต่างๆที่ได้รับค่ามา

## บทที่ 4

### วิธีการและการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบการทำงานในการตัดหญ้าของเครื่องตัดหญ้าสนามโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน เช่น ความเร็วในการเคลื่อนที่, ความเร็วในการตัดหญ้า ปริมาณของแบตเตอรี่ที่ใช้ในการตัดหญ้าแต่ละครั้ง และความคลาดเคลื่อนของจีพีเอส (GPS)

#### 4.1 ทดสอบความเร็วของเครื่องตัดหญ้าในการเคลื่อนที่ในระยะทาง 50 เมตร

การทดสอบความเร็วที่ระยะทาง 50 เมตร แสดงดังรูปที่ 4.1 โดยจับเวลาที่ระยะเริ่มต้นจนถึงระยะสิ้นสุด จากนั้นจึงนำมาคำนวณหาความเร็วจากสมการที่ 4.1

$$V = \frac{s}{t} \quad (4.1)$$

ตารางที่ 4.1 ทดสอบความเร็วในการเคลื่อนที่ (ในการชาร์จแบตเตอรี่ครั้งเดียว)

ครั้งที่	ระยะทาง (เมตร)	เวลา (วินาที)
1	50	124
2	50	111
3	50	119
4	50	130
5	50	137
6	50	113
7	50	117
8	50	124
9	50	110
10	50	131

เวลาโดยเฉลี่ยที่ได้จากการทดสอบความเร็วจากตารางที่ 4.1 เท่ากับ 121.6 วินาที จากนั้นจึงหาความเร็วเฉลี่ยในการเคลื่อนที่จากสมการการเคลื่อนที่ได้จากสมการที่ 4.1

$$V = \frac{s}{t}$$

$$V = \frac{50}{121.6}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับ  $V = 0.411$  เมตรต่อวินาที เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ทดสอบความเร็วในการเคลื่อนที่

#### 4.2 ทดสอบความเร็วในการเคลื่อนที่ 20 ตารางเมตร

จากการทดสอบความเร็วในการเคลื่อนที่พื้นที่ 20 ตารางเมตร แสดงดังรูปที่ 4.2 โดยการเคลื่อนที่ไป-กลับ แสดงดังรูปที่ 4.2

ตารางที่ 4.2 ทดสอบความเร็วในการวิ่งพื้นที่ 20 ตารางเมตร โดยไม่เปิดมอเตอร์ตัดหญ้า (ในการชาร์จแบตเตอรี่ครั้งเดียว)

ครั้งที่	พื้นที่ (ตารางเมตร)	เวลา (วินาที)
1	20	102
2	20	104
3	20	98
4	20	106
5	20	94

จากการทดลองจะพบว่าความเร็วในการเคลื่อนที่เฉลี่ยในพื้นที่ 20 ตารางเมตร จากตารางที่ 4.2 เท่ากับ 100.8 วินาที จากนั้นจากนั้นจึงนำมาหาความเร็วเฉลี่ยได้จากสมการที่ 4.1

$$V = \frac{s}{t}$$

$$V = \frac{20}{100.8}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$V = 0.198$  ตารางเมตรต่อวินาที



รูปที่ 4.2 ทดสอบความเร็วในการเคลื่อนที่พื้นที่ 20 ตาราง

#### 4.3 ทดสอบการตัดหญ้า 20 ตารางเมตร

จากการทดสอบการตัดหญ้าในการเคลื่อนที่พื้นที่ 20 ตารางเมตร แสดงดังรูปที่ 4.3 โดยการเคลื่อนที่ไป-กลับ แสดงดังรูปที่ 4.3

ตารางที่ 4.3 ทดสอบการตัดหญ้าพื้นที่ 20 ตารางเมตร (ในการชาร์จแบตเตอรี่ครั้งเดียว)

ครั้งที่	พื้นที่ (ตารางเมตร)	เวลา (วินาที)
1	20	106
2	20	109
3	20	123
4	20	145
5	20	170

จากผลการทดลองจะพบว่าใช้เวลาเฉลี่ยที่ตัดหญ้าพื้นที่ 20 ตารางเมตร เท่ากับ 130.6 วินาที จากนั้นจึงหาความเร็วเฉลี่ยได้จากสมการที่ 4.1

$$V = \frac{S}{t}$$

$$V = \frac{20}{130.6}$$

$$V = 0.153 \text{ ตารางเมตรต่อวินาที}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ทดสอบการตัดหญ้า 20 ตารางเมตร

#### 4.4 ทดสอบการตัดหญ้า 100 ตารางเมตร

จากการทดสอบการตัดหญ้าในการเคลื่อนที่พื้นที่ 20 ตารางเมตร แสดงดังรูปที่ 4.3 และรูปที่ 4.4

#### ตารางที่ 4.4 ทดสอบปริมาณของแบตเตอรี่ที่ใช้ในการตัดหญ้า

ครั้งที่	พื้นที่ (ตารางเมตร)	เปอร์เซ็นต์แบตเตอรี่ที่ใช้งาน	เวลา(นาที)
1	100	55	13.37
2	100	49	13.26
3	100	53	13.43
4	100	46	14.10
5	100	58	14.19

จากผลการทดลองจะพบว่าได้พบว่าได้เปอร์เซ็นต์แบตเตอรี่ที่ใช้ไปเฉลี่ย 52.5% และใช้เวลาเฉลี่ยที่ในการตัดหญ้า 13.51 นาที

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ทดสอบการตัดหญ้าที่พื้นที่ 100 ตารางเมตร



รูปที่ 4.5 ทดสอบการตัดหญ้าที่พื้นที่ 100 ตารางเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 10 เมตร

การทดสอบความคลาดเคลื่อนในระยะ 10 เมตร แสดงได้ดังรูปที่ 4.5 โดยที่วัดจากระยะเริ่มต้นจนถึงระยะสิ้นสุด จากนั้นจึงนำมาคำนวณได้จากสมการที่ 4.2

$$\text{ความคลาดเคลื่อนร้อยละ} = \left| \frac{\text{ค่าจริง} - \text{ค่าทดลอง}}{\text{ค่าจริง}} \right| \times 100 \quad (4.2)$$

#### ตารางที่ 4.5 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 10 เมตร

ครั้งที่	ระยะจริง (m)	ระยะทางที่วัดได้ (m)	เปอร์เซ็นต์ความคลาดเคลื่อน
1	10	9.68	3.2%
2	10	10.45	4.5%
3	10	10.22	2.2%
4	10	10.37	3.7%
5	10	9.76	2.4%

จากตารางที่ 4.5 สามารถแสดงตัวอย่างการคำนวณดังสมการที่ 4.2

$$\begin{aligned} \text{ความคลาดเคลื่อนร้อยละ} &= \left| \frac{10 - 9.68}{10} \right| \times 100 \\ &= 3.2\% \end{aligned}$$

จากผลการทดลองจากตารางที่ 4.5 จะพบว่าได้เปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ยทั้งหมดเท่ากับ 3.2%



#### รูปที่ 4.6 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 10 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.6 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 20 เมตร

การทดสอบความคลาดเคลื่อนในระยะ 20 เมตร แสดงได้ดังรูปที่ 4.6 โดยที่วัดจากระยะเริ่มต้นจนถึงระยะสิ้นสุด จากนั้นจึงนำมาคำนวณได้จากสมการที่ 4.2

ตารางที่ 4.6 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 20 เมตร

ครั้งที่	ระยะจริง (m)	ระยะทางที่วัดได้ (m)	เปอร์เซ็นต์ความคลาดเคลื่อน
1	20	19.75	1.25%
2	20	20.71	3.55%
3	20	20.54	2.7%
4	20	19.47	2.65%
5	20	20.55	2.75%

จากตารางที่ 4.6 สามารถแสดงตัวอย่างการคำนวณดังสมการที่ 4.2

$$\begin{aligned} \text{ความคลาดเคลื่อนร้อยละ} &= \left| \frac{20 - 20.71}{20} \right| \times 100 \\ &= 3.55\% \end{aligned}$$

จากผลการทดลองจากตารางที่ 4.6 จะพบว่าได้เปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ยทั้งหมดเท่ากับ 2.58%



รูปที่ 4.7 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 20 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.7 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 30 เมตร

การทดสอบความคลาดเคลื่อนในระยะ 30 เมตร แสดงได้ดังรูปที่ 4.7 โดยที่วัดจากระยะเริ่มต้นจนถึงระยะสิ้นสุด จากนั้นจึงนำมาคำนวณได้จากสมการที่ 4.2

ตารางที่ 4.7 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 30 เมตร

ครั้งที่	ระยะจริง (m)	ระยะทางที่วัดได้ (m)	เปอร์เซ็นต์ความคลาดเคลื่อน
1	30	30.6	2%
2	30	29.8	0.67%
3	30	30.20	0.66%
4	30	29.23	2.5%
5	30	29.74	0.86%

จากตารางที่ 4.6 สามารถแสดงตัวอย่างการคำนวณดังสมการที่ 4.2

$$\begin{aligned} \text{ความคลาดเคลื่อนร้อยละ} &= \left| \frac{30 - 29.74}{30} \right| \times 100 \\ &= 0.86\% \end{aligned}$$

จากผลการทดลองจากตารางที่ 4.7 จากผลการทดลองจะพบว่าได้เปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ย 1.34%



รูปที่ 4.8 ทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 30 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# วิเคราะห์และสรุปผลการทดลอง

### 5.1 สรุปผลการทดลอง

จากการทดลองเมื่อขับเคลื่อนเครื่องตัดหญ้าจะมีความเร็วเฉลี่ย 0.411 เมตรต่อวินาที ในส่วนของการตัดหญ้าพื้นที่ 20ตารางเมตร พบว่ามีความเร็วเฉลี่ย 0.153 ตารางเมตรต่อวินาที เมื่อตัดหญ้าครบ 100 ตารางเมตร เปอร์เซ็นต์แบตเตอรี่ที่ใช้ไปเฉลี่ย 52.5 % และใช้เวลาในการทำงานโดยเฉลี่ย 13.51 นาที โดยเมื่อเปิดมอเตอร์ใบตัดทั้ง 3 ตัวพร้อมกันจะทำให้แรงดันของแบตเตอรี่ลดลงและความเร็วในการตัดก็จะลดลงเช่นกัน การทดสอบความคลาดเคลื่อนของจีพีเอสในระยะทาง 10 เมตร 20 เมตร และ 30 เมตร พบว่าได้เปอร์เซ็นต์ความคลาดเคลื่อนเฉลี่ย 3.2% 2.58% และ 1.34% ตามลำดับ จากการทดสอบความคลาดเคลื่อนของจีพีเอสในระยะดังกล่าวพบว่าเปอร์เซ็นต์ความคลาดเคลื่อนในระยะไกลจะน้อยกว่าในระยะทางที่ใกล้ และการนำทางของโมดูลเข็มทิศจะถูกรบกวนโดยสนามแม่เหล็กที่ถูกปล่อยออกมาจากมอเตอร์ทั้ง 4 ตัวของเครื่องตัดหญ้า

### 5.2 ปัญหาและอุปสรรค

การทดสอบเครื่องตัดหญ้าสนามโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน บางครั้งการเชื่อมต่อสัญญาณจีพีเอสอาจไม่ดีเท่าที่ควรเนื่องจากสภาพอากาศมีผลต่อการเชื่อมต่อสัญญาณ เช่น ท้องฟ้ามีเมฆหนาเป็นจำนวนมาก ทำให้สัญญาณอาจเกิดการเชื่อมต่อได้ล่าช้า และในส่วนของบอร์ดควบคุม ESP 32 อาจเชื่อมต่อได้ช้าเพราะเกิดจากการความเร็วของอินเตอร์เน็ตล่าช้า ในการทำงานเมื่อมอเตอร์ขับเคลื่อนและมอเตอร์ตัดหญ้าทำงานจะปล่อยสนามแม่เหล็กออกมาทำให้ไปรบกวนโมดูลเข็มทิศ GY-271M ที่ใช้สนามแม่เหล็กในการหาทิศเหนือและทำให้ค่าของเข็มทิศคลาดเคลื่อนความเร็วของรถตัดจะช้าลงเมื่อแรงดันแบตเตอรี่ลดลง

### 5.3 ข้อเสนอแนะ

1. การใช้งานตัดหญ้าแบบอัตโนมัติควรจะสามารถเลือกพิกัดจุดที่ต้องการในกูเกิลแมพ (Google Map) ได้
2. ควรใช้ลวดที่มีขนาดเส้นผ่านศูนย์กลางที่มากกว่า 20 เซนติเมตร
3. โครงสร้างของเครื่องตัดหญ้าควรมีน้ำหนักที่เบา

## เอกสารอ้างอิง

- [1] ณัฐวุฒิ สังข์แก้ว, พีรพัฒน์ พลเสน และ สุชาธิณี แซ่ตั้ง. (2560). “เครื่องตัดหญ้าสนามควบคุมโดยระบบไมโครคอนโทรลเลอร์”. วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล และสาขาวิศวกรรมพลังงาน คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพรเขตรอุดมศักดิ์.
- [2] หนังสือประกอบการอบรมเชิงปฏิบัติการหัวข้อการประยุกต์ใช้ IOT ขั้นสูง. หลักสูตรเทคโนโลยีสื่อสารคมนาคม วิทยาลัยเทคโนโลยีสารสนเทศและการสื่อสาร มหาวิทยาลัยรังสิต
- [3] ชัยวัฒน์ ลิ้มพรจิตรวิสัย, คู่มือนักอิเล็กทรอนิกส์ ฉบับ POCKET BOOK. กรุงเทพฯ: วี.พริ้นท์
- [4] โมดูลแปลงไฟ เรกูลเตอร์ DC-to-DC Step (ระบบออนไลน์)  
แหล่งที่มา : <https://www.mosfex.com/product/83/> เข้า ครั้งสุดท้าย 9 เมษายน 2564
- [5] ปทุมพร วงศ์ใหญ่. (2560). “แหล่งจ่ายพลังงานไฟฟ้าพกพา”. งานวิจัย, สาขาวิศวกรรมไฟฟ้า, มหาวิทยาลัยเทคโนโลยีสุรนารี
- [6] ไชยชาญ หินเกิด, มอเตอร์และการควบคุม, พิมพ์ครั้งที่ 2, กรุงเทพฯ: ส.ส.ท
- [7] นิกธ จิตวิฆาม, วัชรา บัวเพ็ญ, ประชา คำภักดี ,วรการ วงศ์สายเชื้อ และสุริยา โชคสวัสดิ์การออกแบบและสร้างระบบควบคุมมอเตอร์เพื่อขับเคลื่อนแท่งอิเล็กทรอนิกส์ของเตาหลอมเหลวแบบอาร์คโดยใช้ไมโครคอนโทรลเลอร์. วิทยานิพนธ์ ปริญญาวิศวกรรมศาสตรบัณฑิต ภาควิชาวิศวกรรมไฟฟ้าและอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยอุบลราชธานี.
- [8] จีราวุธ วารินทร์, การพัฒนาแอปพลิเคชันด้วย Firebase, พิมพ์ครั้งที่ 1, กรุงเทพฯ: ชิมพลีฟาย
- [9] อติศักดิ์ มหารวรรณ บทเรียนออนไลน์ ทฤษฎีบทของปีทาโกรัส  
แหล่งที่มา : <https://sites.google.com/a/web1.dara.ac.th/> ครั้งสุดท้าย 9 เมษายน 2564
- [10] จีพีเอส (GPS) (ระบบออนไลน์)  
แหล่งที่มา : <https://www.prosoftgps.com/> เข้าครั้งสุดท้าย 22 เมษายน 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [11] ระบบจีพีเอส (ระบบออนไลน์)  
แหล่งที่มา : <https://www.garmin.com/th-TH/aboutgps/> เข้าครั้งสุดท้าย 30 เมษายน 2564
- [12] ระบบทางภูมิศาสตร์ Geographic coordinate system (ระบบออนไลน์)  
แหล่งที่มา : <https://www.global5thailand.com> เข้าครั้งสุดท้าย 30 เมษายน 2564
- [13] เศวริน กันหา, “ระบบติดตามยานพาหนะโดยใช้จีพีเอสและทำนายตำแหน่งด้วยคาลมานอัลกอริทึม,”  
วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต, สาขาวิศวกรรมโทรคมนาคม, มหาวิทยาลัยเทคโนโลยีสุรนารี,  
0032
- [14] ชัยพร เขมะภาคะพันธ์. (2555). “ระบบติดตามตรวจสอบตำแหน่งและเส้นทางของรถยนต์ด้วยสัญญาณดาวเทียม”. งานวิจัย มหาวิทยาลัยธุรกิจบัณฑิต.
- [15] ธงชัย แก้วกิริยา. (2557). “การควบคุมและติดตามยานพาหนะด้วยระบบ GPS โดยใช้โทรศัพท์มือถือผ่านเครือข่าย GPRS/3G”. วารสารร่วมพฤษ มหวิทยาลัยเกริก.
- [16] แอปพลิเคชัน Thinkable (ระบบออนไลน์)  
แหล่งที่มา : <https://www.dailytech.in.th/thinkable/> / 1 พฤษภาคม 2564



ภาคผนวก ก

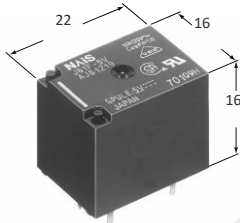
คู่มือการใช้งานอุปกรณ์ (Datasheet)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# NAIS

## ULTRA-MINIATURE PC BOARD TYPE POWER RELAY

# JS RELAYS



mm inch

### FEATURES

- Ultra-miniature size with universal terminal footprint
- High contact capacity: 10 A
- Class B coil insulation type available
- TV-5 type available
  - 1 Form A type → TV-5
  - 1 Form C type → TV-5 (N.O. side only)
- VDE, TÜV also approved
- Sealed construction for automatic cleaning

### SPECIFICATIONS

#### Contact

Arrangement	1 Form A, 1 Form C	
Initial contact resistance, max. (By voltage drop 6 V DC 1 A)	100 mΩ	
Contact material	Silver alloy	
Rating (resistive load)	Nominal switching capacity	10 A 250 V AC 10 A 125 V AC 6 A 277 V AC
	Max. switching power	2,500 VA
	Max. switching voltage	250 V AC, 100 V DC
	Max. switching current	10 A (AC), 5 A (DC)
	Expected life (min.ope.)	Mechanical (at 180 cpm)
Electrical at 10 A 125 V AC, 6 A 277 V AC resistive (at 20 cpm)		10 <sup>5</sup>
10 A 250 V AC resistive (at 20 cpm)		5 × 10 <sup>4</sup> (No contact only)

#### Coil

Nominal operating power	360 mW
-------------------------	--------

#### Remarks

- \* Specifications will vary with foreign standards certification ratings
- \*<sub>1</sub> Detection current: 10mA
- \*<sub>2</sub> Excluding contact bounce time
- \*<sub>3</sub> Half-wave pulse of sine wave: 11ms; detection time: 10μs
- \*<sub>4</sub> Half-wave pulse of sine wave: 6ms
- \*<sub>5</sub> Detection time: 10μs
- \*<sub>6</sub> Refer to 5. Conditions for operation, transport and storage mentioned in AMBIENT ENVIRONMENT (Page 24).
- \*<sub>7</sub> When using relays in a high ambient temperature, consider the pick-up voltage rise due to the high temperature (a rise of approx. 0.4% V for each 1°C 33.8°F with 20°C 68°F as a reference) and use a coil impressed voltage that is within the maximum allowable voltage range.

#### Characteristics

Max. operating speed	20 cpm	
Initial insulation resistance	Min. 100 MΩ (at 500 V DC)	
Initial breakdown voltage* <sub>1</sub>	Between open contacts	750 Vrms for 1 min.
	Between contacts and coil	1,500 Vrms for 1 min.
Operate time* <sub>2</sub> (at nominal voltage)	Approx. 10 ms	
Release time(without diode)* <sub>2</sub> (at nominal voltage)	Approx. 10 ms	
Temperature rise (at nominal voltage)	Max. 35°C	
Shock resistance	Functional* <sub>3</sub>	Min. 98 m/s <sup>2</sup> {10 G}
	Destructive* <sub>4</sub>	Min. 980 m/s <sup>2</sup> {100 G}
Vibration resistance	Functional* <sub>5</sub>	Approx. 98 m/s <sup>2</sup> {10 G}, 10 to 55 Hz at double amplitude of 1.6 mm
	Destructive	Approx. 117.6 m/s <sup>2</sup> {12 G}, 10 to 55 Hz at double amplitude of 2 mm
Conditions for operation, transport and storage* <sub>6</sub> (Not freezing and condensing at low temperature)	Ambient temp.* <sub>7</sub>	-40°C to +85°C -40°F to +185°F
	Humidity	5 to 85% R.H.
Unit weight	Approx. 12 g .423 oz	

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TYPICAL APPLICATIONS

### 1. Home appliances

Air conditioner, heater, etc.

### 2. Automotive

Power-window, car antenna, door-lock, etc.

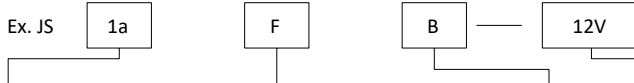
3. Office machines PPC,  
facsimile, etc.

4. Vending machines



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ORDERING INFORMATION



Contact arrangement	Protective construction	Coil insulation class	Coil voltage (DC)
1: 1 Form C 1a: 1 Form A	Nil: Sealed type	Nil: Class E insulation B: Class B insulation	5, 6, 9, 12, 18, 24,

UL/CSA, VDE, TÜV approved type is standard.

Notes: 1. Standard packing: Carton: 100 pcs. Case: 500 pcs.

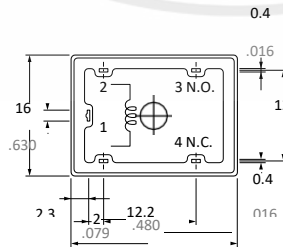
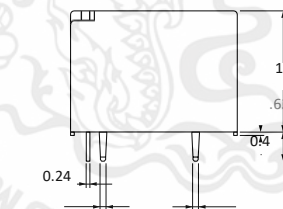
2. When ordering TV rated (TV-5) types, add suffix -TV.

### COIL DATA

Part No.				Nominal voltage, V DC	Pick-up voltage, V DC (max.) (at 20°C/68°F)	Drop-out voltage, V DC (min.) (at 20°C/68°F)	Coil resistance, Ω (±10%) (at 20°C/68°F)	Nominal operating current, mA (±10%) (at 20°C/68°F)	Nominal operating power, mW (at 20°C/68°F)	Max. allowable voltage (at 85°C/185°F)
Sealed type		Flux-resistant type								
1 Form A	1 Form C	1 Form A	1 Form C							
JS1a-5V	JS1-5V	JS1aF-5V	JS1F-5V	5	3.5	0.5	69.4	72	360	130%V of nominal voltage
JS1a-6V	JS1-6V	JS1aF-6V	JS1F-6V	6	4.2	0.6	100	60		
JS1a-9V	JS1-9V	JS1aF-9V	JS1F-9V	9	6.3	0.9	225	40		
JS1a-12V	JS1-12V	JS1aF-12V	JS1F-12V	12	8.4	1.2	400	30		
JS1a-18V	JS1-18V	JS1aF-18V	JS1F-18V	18	12.6	1.8	900	20		
JS1a-24V	JS1-24V	JS1aF-24V	JS1F-24V	24	16.8	2.4	1,600	15		
JS1a-48V	JS1-48V	JS1aF-48V	JS1F-48V	48	33.6	4.8	6,400	7.5		

### DIMENSIONS

mm inch

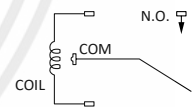


.866

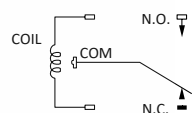
Note: Terminal No. 4 is only for 1 Form C type

### Schematic (Bottom view)

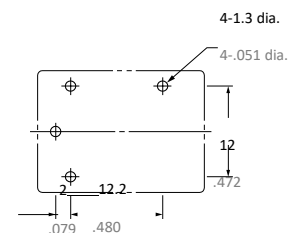
1a



1c

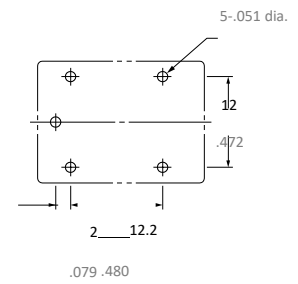


### PC board pattern (Copper-side view) 1a



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า  
 General tolerance: ±0.3 ±0.12  
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตกแก่บุคคลอื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5-1.3 dia.

Tolerance:  $\pm 0.1 \pm .004$ 

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## BD135/BD137/BD139

TRANSISTOR (NPN)

TO-126

1. EMITTER

2. COLLECTOR



1 2 3

## FEATURES

Power dissipation

 $P_{CM}$ : 1.25 W ( $T_{amb}=25^{\circ}C$ )

Collector current

 $I_{CM}$ : 1.5 AOperating and storage junction temperature range  $T_j, T_{stg}$ : $-55^{\circ}C$  to  $+150^{\circ}C$ ELECTRICAL CHARACTERISTICS ( $T_{amb}=25^{\circ}C$  unless otherwise specified)

Parameter	Symbol	Test conditions	MIN	TYP	MAX	UNIT
Collector-base breakdown voltage	$V_{(BR)CBO}$	$I_C=100\mu A, I_E=0$	BD135	45		V
			BD137	60		
			BD139	80		
Collector-emitter breakdown voltage	$V_{(BR)CEO}$	$I_C=30mA, I_B=0$	BD135	45		V
			BD137	60		
			BD139	80		
Emitter-base breakdown voltage	$V_{(BR)EBO}$	$I_E=100\mu A, I_C=0$	5			V
Collector cut-off current	$I_{CBO}$	$V_{CB}=30V, I_E=0$			0.1	$\mu A$
Emitter cut-off current	$I_{EBO}$	$V_{EB}=5V, I_C=0$			10	$\mu A$
DC current gain	$h_{FE(1)}$	$V_{CE}=2V, I_C=5mA$	25			
	$h_{FE(2)}$	$V_{CE}=2V, I_C=150mA$	BD135	40	250	
			BD137/BD139	40	160	
$h_{FE(3)}$	$V_{CE}=2V, I_C=500mA$	25				
Collector-emitter saturation voltage	$V_{CE(sat)}$	$I_C=500mA, I_B=50mA$			0.5	V

Base-emitter voltage	$V_{BE}$	$V_{CE}=2V, I_C=500mA$			1	V
----------------------	----------	------------------------	--	--	---	---

**CLASSIFICATION OF  $h_{FE(2)}$**

Rank	6	10	16
Range	40-100	63-160	100-250

**WEJ ELECTRONIC CO.**

[Http:// www.wej.cn](http://www.wej.cn)

E-mail: [wej@yongerjia.com](mailto:wej@yongerjia.com)



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ESP32 Series

## Datasheet

### Overview

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

The ESP32 series of chips includes ESP32-D0WD-V3, ESP32-D0WDQ6-V3, ESP32-D0WD, ESP32-D0WDQ6, ESP32-D2WD, ESP32-S0WD, and ESP32-U4WDH, among which, ESP32-D0WD-V3, ESP32-D0WDQ6-V3, and ESP32-U4WDH are based on ECO V3 wafer.

For details on part numbers and ordering information, please refer to Section 7. For details on ECO V3 instructions, please refer to [ESP32 ECO V3 User Guide](#).

### Featured Solutions

#### Ultra-Low-Power Solution

ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically and only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

#### Note:

For more information, refer to Section 3.7 *RTC and Low-Power Management*.

#### Complete Integration Solution

ESP32 is a highly-integrated solution for Wi-Fi-and-Bluetooth IoT applications, with around 20 external components. ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area.

ESP32 uses CMOS for single-chip fully-integrated radio and baseband, while also integrating advanced calibration circuitries that allow the solution to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi testing equipment.

### Wi-Fi Key Features

-802.11 b/g/n

-WMM

เอกสาร-802.11 n (2.4 GHz), up to 150 Mbps -TX/RX A-MPDU, RX A-MSDU นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Immediate Block ACK

- Defragmentation
- Automatic Beacon monitoring (hardware TSF)
- 4 × virtual Wi-Fi interfaces
- Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes  
Note that when ESP32 is in Station mode, performing a scan, the SoftAP channel will be changed.
- Antenna diversity

### Note:

For more information, please refer to Section 3.5 [Wi-Fi](#).

## BT Key Features

- Compliant with Bluetooth v4.2 BR/EDR and BLE specifications
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced Power Control
- +12 dBm transmitting power
- NZIF receiver with -94 dBm BLE sensitivity
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- Bluetooth 4.2 BR/EDR BLE dual mode controller
- Synchronous Connection-Oriented/Extended (SCO/eSCO)
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet
- Multi-connections in Classic BT and BLE
- Simultaneous advertising and scanning

## MCU and Advanced Features

### CPU and Memory

- Xtensa® single-/dual-core 32-bit LX6 microprocessor(s), up to 600 MIPS (200 MIPS for ESP32-S0WD/ESP32-U4WDH, 400 MIPS for ESP32-D2WD)
- 448 KB ROM
- 520 KB SRAM
- 16 KB SRAM in RTC

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Electrical Characteristics

### Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in the table below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the [recommended operating conditions](#).

Table 11: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO	Voltage applied to power supply pins per power domain	-0.3	3.6	V
$I_{output}^*$	Cumulative IO output current	-	1200	mA
$T_{store}$	Storage temperature	-40	150	°C

\* The chip worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3\_RTC, VDD3P3\_CPU, VDD\_SDIO) output high logic level to ground.

### Recommended Operating Conditions

Table 12: Recommended Operating Conditions

Symbol	Parameter	Min	Typ	Max	Unit
VDDA, VDD3P3_RTC <sup>1</sup> VDD3P3, VDD_SDIO (3.3 V mode) <sup>2</sup>	Voltage applied to power supply pins per power domain	2.3	3.3	3.6	V
VDD3P3_CPU	Voltage applied to power supply pin	1.8	3.3	3.6	V
$I_{VDD}$	Current delivered by external power supply	0.5	-	-	A
$T^3$	Operating temperature	-40	-	125	°C

- When writing eFuse, VDD3P3\_RTC should be at least 3.3 V.
- VDD\_SDIO works as the power supply for the related IO, and also for an external device. Please refer to the Appendix [IO\\_MUX](#) of this datasheet for more details.
  - VDD\_SDIO can be sourced internally by the ESP32 from the VDD3P3\_RTC power domain:
    - When VDD\_SDIO operates at 3.3 V, it is driven directly by VDD3P3\_RTC through a 6 Ω resistor, therefore, there will be some voltage drop from VDD3P3\_RTC.
    - When VDD\_SDIO operates at 1.8 V, it can be generated from ESP32's internal LDO. The maximum current this LDO can offer is 40 mA, and the output voltage range is 1.65 V ~ 2.0 V.
  - VDD\_SDIO can also be driven by an external power supply.
  - Please refer to Power Scheme, section [2.3](#), for more information.
- The operating temperature of ESP32-D2WD and ESP32-U4WDH ranges from -40 °C to 105 °C, due to the flash embedded in them. The other chips in this series have no embedded flash, so their range of operating temperatures is -40 °C ~ 125 °C.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DC Characteristics (3.3 V, 25 °C)

Table 13: DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter	Min	Typ	Max	Unit	
$C_{IN}$	Pin capacitance	-	2	-	pF	
$V_{IH}$	High-level input voltage	$0.75 \times VDD^1$	-	$VDD^1 + 0.3$	V	
$V_{IL}$	Low-level input voltage	-0.3	-	$0.25 \times VDD^1$	V	
$I_{IH}$	High-level input current	-	-	50	nA	
$I_{IL}$	Low-level input current	-	-	50	nA	
$V_{OH}$	High-level output voltage	$0.8 \times VDD^1$	-	-	V	
$V_{OL}$	Low-level output voltage	-	-	$0.1 \times VDD^1$	V	
$I_{OH}$	High-level source current ( $VDD^1 = 3.3$ V, $V_{OH} \geq 2.64$ V, output drive strength set to the maximum)	VDD3P3_CPU power domain <sup>1, 2</sup>	-	40	-	mA
		VDD3P3_RTC power domain <sup>1, 2</sup>	-	40	-	mA
		VDD_SDIO power domain <sup>1, 3</sup>	-	20	-	mA
$I_{OL}$	Low-level sink current ( $VDD^1 = 3.3$ V, $V_{OL} = 0.495$ V, output drive strength set to the maximum)	-	28	-	mA	
$R_{PU}$	Resistance of internal pull-up resistor	-	45	-	k $\Omega$	
$R_{PD}$	Resistance of internal pull-down resistor	-	45	-	k $\Omega$	
$V_{IL\_nRST}$	Low-level input voltage of CHIP_PU to power off the chip	-	-	0.6	V	

### Notes:

1. Please see Table IO\_MUX for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2. For VDD3P3\_CPU and VDD3P3\_RTC power domain, per-pin current sourced in the same domain is gradually reduced from around 40 mA to around 29 mA,  $V_{OH} \geq 2.64$  V, as the number of current-source pins increases.
3. For VDD\_SDIO power domain, per-pin current sourced in the same domain is gradually reduced from around 30 mA to around 10 mA,  $V_{OH} \geq 2.64$  V, as the number of current-source pins increases.

# NEO-M8

u-blox M8 concurrent GNSS modules



## Data sheet

### Performance

Parameter	Specification					
Receiver type	72-channel u-blox M8 engine GPS L1C/A, SBAS L1C/A, QZSS L1C/A, QZSS L1 SAIF, GLONASS L1OF, BeiDou B1I, Galileo E1B/C					
Accuracy of timepulse signal	RM	30 ns				
	S	60 ns				
	99%					
Frequency of time pulse signal	0.25 Hz...10 MHz (configurable)					
Operational limits <sup>1</sup>	Dynamics	≤ 4 g				
	Altitude	50,000 m				
	Velocity	500 m/s				
Velocity accuracy <sup>2</sup>	0.05 m/s					
Heading accuracy <sup>2</sup>	0.3 degrees					
GNSS	GPS & GLONASS	GPS	GLONASS	BeiDou	Galileo	
Horizontal position accuracy <sup>3</sup>	2.5 m	2.5 m	4 m	3 m	3 m	
NEO-M8N/Q						
Max navigation update rate	NEO-M8N	5 Hz	10 Hz	10 Hz	10 Hz	10 Hz
	NEO-M8Q	10 Hz	18 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix <sup>4</sup>	Cold start	26 s	29 s	30 s	34 s	45 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts <sup>5</sup>	2 s	2 s	2 s	3 s	7 s
Sensitivity <sup>6</sup>	Tracking & Navigation	-167 dBm	-166 dBm	-166 dBm	-160 dBm	-159 dBm
	Reacquisition	-160 dBm	-160 dBm	-156 dBm	-157 dBm	-153 dBm
	Cold start	-148 dBm	-148 dBm	-145 dBm	-143 dBm	-138 dBm
	Hot start	-157 dBm	-157 dBm	-156 dBm	-155 dBm	-151 dBm
NEO-M8J/M						
Max navigation update rate	NEO-M8J	5 Hz	10 Hz	10 Hz	10 Hz	10 Hz
	NEO-M8M	10 Hz	18 Hz	18 Hz	18 Hz	18 Hz
Time-To-First-Fix <sup>5</sup>	Cold start	26 s	30 s	33 s	39 s	57 s
	Hot start	1 s	1 s	1 s	1 s	1 s
	Aided starts <sup>6</sup>	3 s	3 s	3 s	7 s	7 s

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# HMC634

*GaAs PHEMT MMIC DRIVER*

*AMPLIFIER, 5 - 20 GHz*



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Typical Applications

The HMC634 is ideal for:

- Point-to-Point Radios
- Point- to-Multi-Point Radios & VSAT
- LO Driver for Mixers
- Military & Space

## Features

Gain: 22 dB

P1dB: +23 dBm

Output IP3: +31 dBm

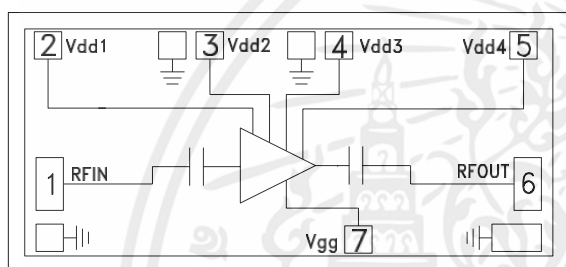
Saturated Power: 24 dBm @ 23% PAE

Supply Voltage: +5 V @ 180 mA

50 Ohm Matched Input/Output

Die Size: 2.07 x 0.97 x 0.10 mm

## Functional Diagram



## General Description

The HMC634 is a GaAs MMIC PHEMT Driver Amplifier die which operates between 5 and 20 GHz. The amplifier provides up to 22 dB of gain, +31 dBm Output IP3, and up to +23 dBm of output power at 1 dB gain compression, while requiring 180 mA from a +5V supply. The HMC634 is an ideal driver amplifier for microwave radio applications from 5 to 20 GHz, and may be biased at +5V, 130 mA to provide 2 dB lower gain with improved PAE. The HMC634 amplifier I/O's are DC blocked and internally matched to 50 Ohms facilitating easy integration into Multi-Chip-Modules (MCMs). All data is taken with die connected at input and output RF ports via one 1 mil wedge bond with minimal length of 0.31 mm (12 mils).

## Electrical Specifications, $T_A = +25^\circ\text{C}$ , Vdd1, Vdd2, Vdd3, Vdd4= 5V, Idd= 180mA<sup>[1]</sup>

Parameter	Min.	Typ.	Max.	Min.	Typ.	Max.	Units
Frequency Range	5 - 16		16 - 20				GHz
Gain	17	22		17	20		dB
Gain Variation Over Temperature		0.030	0.040		0.025	0.035	dB/°C
Input Return Loss		12			9		dB
Output Return Loss		12			11		dB
Output Power for 1 dB Compression (P1dB)	21	23		18	21		dBm
Saturated Output Power (Psat)		24			22		dBm
Output Third Order Intercept (IP3)		31			30		dBm
Noise Figure		7			7.5		dB
Supply Current (Idd1 + Idd2 + Idd3 + Idd4)		180			180		mA

<sup>[1]</sup> Adjust Vgg between -2 to 0V to achieve Idd = 180mA Typical

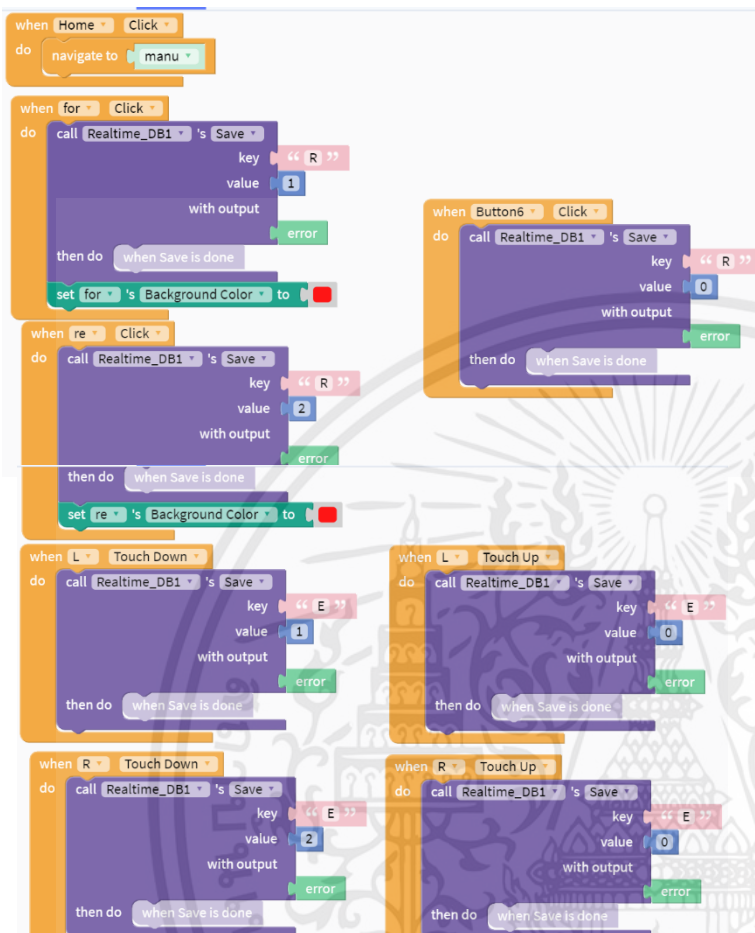


ภาคผนวก ข

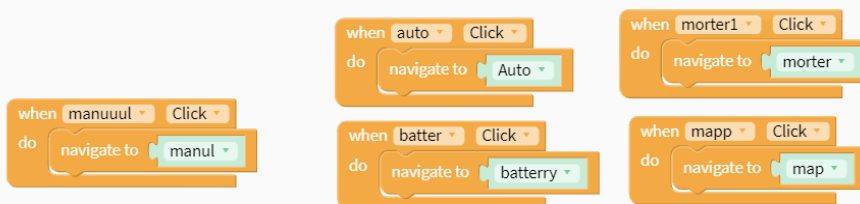
(โปรแกรมควบคุมระบบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การออกแบบแอปพลิเคชัน Thinkable แบบ Mannul



### คำสั่งเชื่อมต่อหน้าเมนูแอปพลิเคชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Thunkable ระบุตำแหน่ง จุด 4 จุด

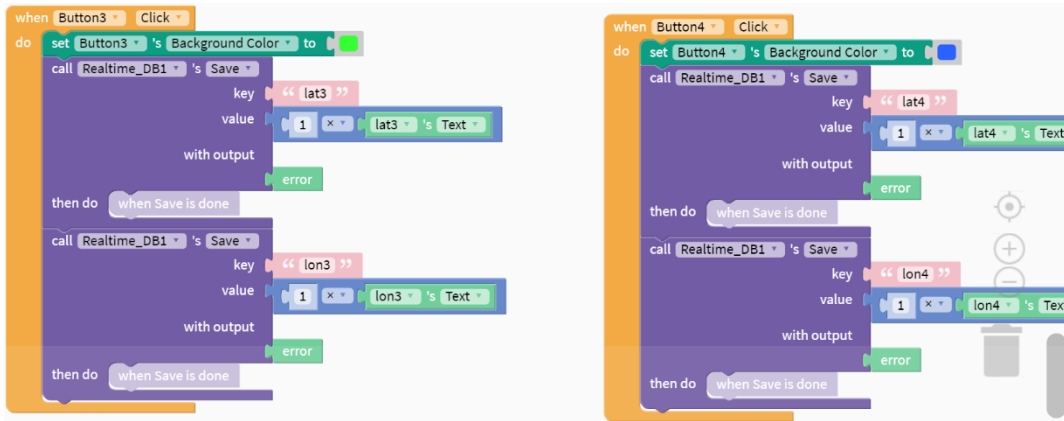
The image displays four distinct code blocks from the Thunkable visual programming environment, arranged in a 2x2 grid. Each block is designed to handle a specific user interaction related to marking points on a map.

- Top-Left Block:** Triggered by a 'Click' event on 'SetPoint1'. It calls 'Location\_Sensor1' to get the current location (latitude, longitude). It then saves these coordinates to a 'Realtime\_DB1' with keys 'lat1' and 'lon1'. After saving, it sets the background color of 'SetPoint1' to red.
- Top-Right Block:** Triggered by a 'Click' event on 'SetPoint2'. It follows a similar process to the first block but uses keys 'lat2' and 'lon2' and sets the background color of 'SetPoint2' to red.
- Bottom-Left Block:** Triggered by a 'Click' event on 'ss'. It retrieves the current latitude ('latnow') and longitude ('lonnow') from 'Realtime\_DB1'. It updates the text of 'lat' and 'lon' labels with these values. It then sets the background color of 'ss' to green and adds a marker to a 'Map1' at the current location with a title and description of 'CarCut'.
- Bottom-Right Block:** Triggered by a 'Click' event on 'Button2'. It saves the current latitude ('lat2') and longitude ('lon2') to 'Realtime\_DB1' with keys 'lat2' and 'lon2'. It then sets the background color of 'Button2' to green.

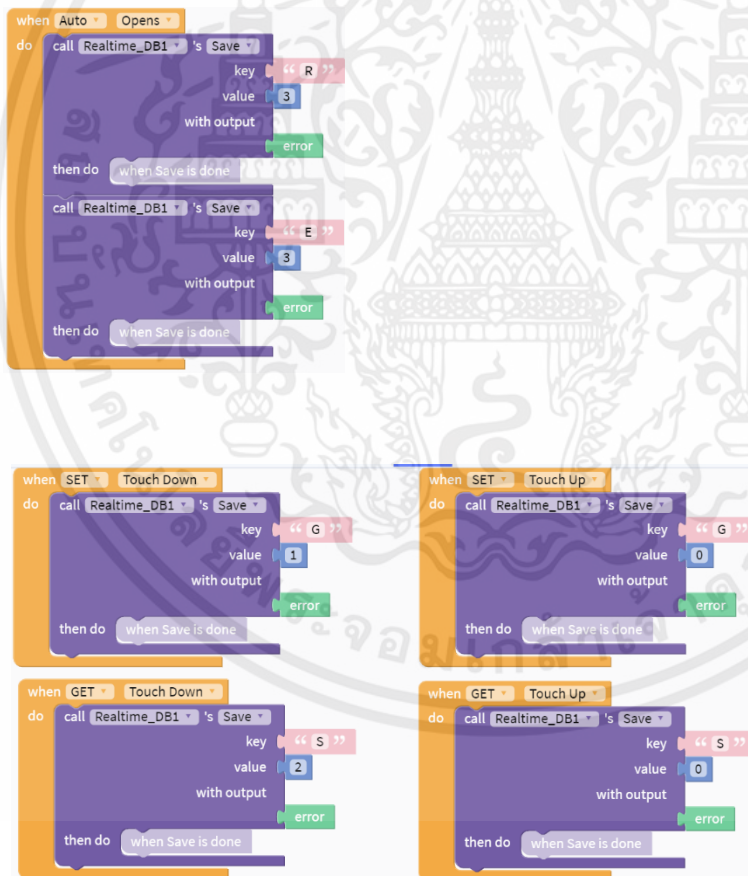
Additionally, there are two smaller code blocks in the middle of the page:

- Center-Left Block:** Triggered by a 'Click' event on 'refm', it sets the background colors of 'SetPoint1', 'SetPoint2', 'SetPoint3', and 'SetPoint4' to green.
- Center-Right Block:** Triggered by a 'Click' event on 'Auto', it navigates the application to the 'Auto' screen.

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

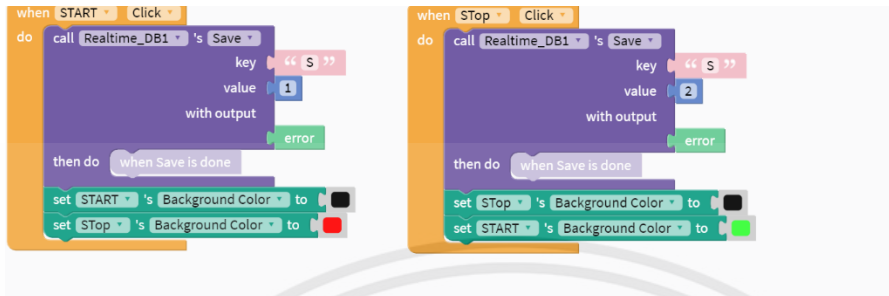


### Thunkable ควคุมมอเตอร์และเซ็ดตำแหน่ง

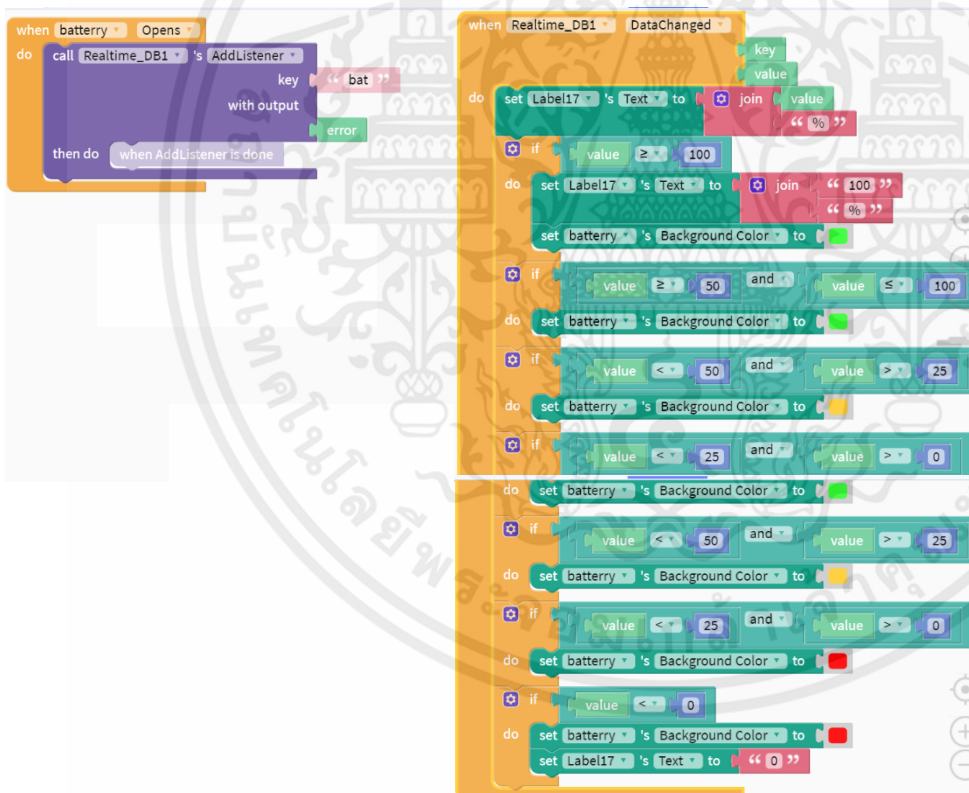


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Thunkable สิ่งเริ่มการทำงานระบบอัตโนมัติ



## Thunkable แสดงเปอร์เซ็นต์แบตเตอรี่



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมควบคุมการทำงานเครื่องตัดหญ้า

```

#include <IOXhop_FirebaseESP32.h>

#include <WiFi.h>

#define FIREBASE_HOST "marktest-81234-default-rtdb.firebaseio.com"

#define FIREBASE_AUTH "qejjKREhDpDbaZlcmZj3fsXkNqVSXbyjn5ZUngKV"

#define WIFI_SSID "AR"

#define WIFI_PASSWORD "60511057"

#include <QMC5883L.h>

#include <Wire.h>

#include <TinyGPS++.h>

#include <HardwareSerial.h>

#define RXPin (16)

#define TXPin (17)

#include <LiquidCrystal_I2C.h>

#include <ESP32Servo.h>

Servo myservo;

QMC5883L compass;

LiquidCrystal_I2C lcd(0x27, 16, 4);

static const uint32_t GPSBaud = 9600;

TinyGPSPlus gps;

HardwareSerial ss(2);

#define forward (27)

#define Reverse (26)

#define MotorCut1 (25)

#define MotorCut2 (12)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define MotorCut3 (14)

const int potPin = 34;

int potValue = 0;

int e,s,l,r,g;

float lat1,lat2,lat3,lat4,lon1,lon2,lon3,lon4;

void setup() {

  Serial.begin(115200);

  ss.begin(GPSBaud, SERIAL_8N1, RXPin, TXPin, false);

  lcd.begin();

  lcd.backlight();

  Wire.begin();

  compass.init();

  compass.setSamplingRate(50);

  Serial.begin(9600);

  Serial.println("QMC5883L Compass Demo");

  Serial.println("Turn compass in all directions to calibrate....");

  myservo.attach(32);

  pinMode (forword, OUTPUT);

  pinMode (Reverse, OUTPUT);

  pinMode(MotorCut1, OUTPUT);

  pinMode(MotorCut2, OUTPUT);

  pinMode(MotorCut3, OUTPUT);

  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);

  Serial.print("connecting");

  while (WiFi.status() != WL_CONNECTED) {

```

```

Serial.print(".");

delay(500);

}

Serial.println();

Serial.print("connected: ");

Serial.println(WiFi.localIP());

Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);

Firebase.stream("", [(FirebaseStream stream)

{

if (stream.getEvent() == "put" && stream.getPath() == "/S") {

s = stream.getDataInt();

Serial.println(s); }

if (stream.getEvent() == "put" && stream.getPath() == "/G") {

g = stream.getDataInt();

Serial.println(g); }

if (stream.getEvent() == "put" && stream.getPath() == "/E") {

e = stream.getDataInt();

Serial.println(e); }

if (stream.getEvent() == "put" && stream.getPath() == "/L") {

l = stream.getDataInt();

Serial.println(l); }

if (stream.getEvent() == "put" && stream.getPath() == "/R") {

r = stream.getDataInt();

Serial.println(r); }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

double mapf(double val, double in_min, double in_max, double out_min, double out_max) {

    return (val - in_min) * (out_max - out_min) / (in_max - in_min) + out_min; }

float x,y;

float Dn=0;

float latnow,lonnow;

float Dr;

float Df;

int DD = 0;

int persen,heading,radar,point,servoR,RL,LR,Mr,Mf;

void loop() {

while (ss.available() > 0) {

    gps.encode(ss.read());

    if (gps.location.isUpdated()) {

        latnow = gps.location.lat();

        lonnow = gps.location.lng();

        double volt = mapf(analogRead(potPin), 0, 4095, 0, 17.30);

        persen = ((volt-10)/3)*100;

// Serial.println(volt);

        lcd.setCursor(12, 0);lcd.print("CmN:"); lcd.print(heading);

        lcd.setCursor(12, 3);lcd.print("r:"); lcd.print(r);

        lcd.setCursor(16, 3);lcd.print("e:"); lcd.print(e);

        lcd.setCursor(0, 3); lcd.print("SvR:");lcd.print(servoR);

        lcd.setCursor(10, 0); lcd.print("|");

        lcd.setCursor(10, 2); lcd.print("|");

```

```

lcd.setCursor(10, 3); lcd.print("|");

lcd.setCursor(0, 1); lcd.print("Dn:");lcd.print(Dn,2);

DN();

gy ();

servor ();

}}

if(s == 1){ lcd.setCursor(0, 0); lcd.print("ComS :");lcd.print(radar);

        lcd.setCursor(0, 2); lcd.print("Df:");lcd.print(Df,2);

        lcd.setCursor(11, 2); lcd.print("Dr:");lcd.print(Dr,2);

        while(Df != 500){ digitalWrite(forward, HIGH); myservo.write(90); delay (300);

            if(servoR < -2){ myservo.write(120); delay (200); myservo.write(90); delay (300); break;}

            else if(servoR > 2){ myservo.write(60); delay (200); myservo.write(90); delay (300); break;}

break;}

        while(Dr != 500){ digitalWrite(Reverse, HIGH); myservo.write(90); delay (300);

            if(servoR < -2){ myservo.write(60); delay (200); myservo.write(93); delay (300); break;}

            else if(servoR > 2){ myservo.write(120); delay (200); myservo.write(93); delay (300); break;}

break;}

            if(Dn >= Df){ DD++; step1();stop1(); x = gps.location.lat(); y = gps.location.lng();

latlon1();delay(200); Dr = 10; Df = 500; }

            else if(Dn >= Dr){ DD++; step2();stop1(); x = gps.location.lat(); y = gps.location.lng();

latlon2();delay(200); Df = 10; Dr = 500; }

        }

else if (s == 2){ Dr = 500; Df = 500; digitalWrite(forward, LOW);digitalWrite(Reverse, LOW);

                digitalWrite(MotorCut1,LOW);digitalWrite(MotorCut2,LOW);

digitalWrite(MotorCut3,LOW);}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (g == 1){ Dr = 500; Df = 50; radar = heading; x = gps.location.lat(); y = gps.location.lng();delay(500);

    lcd.setCursor(0, 0); lcd.print("ComS :");lcd.print(radar);

    lcd.setCursor(0, 2); lcd.print(x,6);

    lcd.setCursor(11, 2); lcd.print(y,6); }

else if (g == 2){ setdata (); lcd.setCursor(0, 0); lcd.print("OK.....");

    lcd.setCursor(0, 3); lcd.print("Per :"); lcd.print(percen);

    lcd.setCursor(0, 2); lcd.print("          ");

    lcd.setCursor(0, 1); lcd.print("          "); }

else if (g == 3){getdata (); lcd.setCursor(0, 0); lcd.print("Successfully sent");}

    if (l == 1)
{digitalWrite(MotorCut1,HIGH);digitalWrite(MotorCut2,HIGH);digitalWrite(MotorCut3,HIGH);}

    else {digitalWrite(MotorCut1,LOW);digitalWrite(MotorCut2,LOW); digitalWrite(MotorCut3,LOW); }

if (r == 1){ digitalWrite(forword, HIGH);}

else if (r == 2){ digitalWrite(Reverse, HIGH);}

else if (r == 0){ digitalWrite(Reverse, LOW); digitalWrite(forword, LOW); }

if (e == 2){ myservo.write(30); }

else if (e == 1){ myservo.write(150);}

else if (e == 0){ myservo.write(90); } }

void latlon2 () {

    double X2 = lat4 - lat3;

    double Y2 = lon4 - lon3;

    double m2 = Y2/X2;

    double mm2 = atan(m2);

    double M2 = mm2*57.295777;

    double c2 = sqrt((X2*X2)+(Y2*Y2));

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของโรงเรียนเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double D2 = (c2/(1/108.4))*1000;

int point2 = D2/0.5;

double cp2 = c2/point2;

double cii[point2];

double xii[point2];

double x2[point2];

double yii[point2];

double y2[point2];

for (int ii = 0; ii < point2; ++ii) {

    cii[ii] = ii*cp2;

}

for (int jj = 0; jj < point2; ++jj) {

    xii[jj] = cii[jj]*cos(mm2);

    yii[jj] = cii[jj]*sin(mm2); }

for (int kk = 0; kk < point2; ++kk) {

    x2[kk] = lat3+xii[kk];

    y2[kk] = lon3+yii[kk];

    double X2f = x2[DD] - x;

    double Y2f = y2[DD] - y;

    double mf = Y2f/X2f;

    double mmf = atan(mf);

    Mf = mmf*57.295777;

    double cf = sqrt((X2f*X2f)+(Y2f*Y2f));

    Df = (cf/(1/108.4))*1000; }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void latlon1 () {

    double X1 = lat2 - lat1;

    double Y1 = lon2 - lon1;

    double m = Y1/X1;

    double mm = atan(m);

    double M = mm*57.295777;

    double c = sqrt((X1*X1)+(Y1*Y1));

    double D = (c/(1/108.4))*1000;

    point = D/0.5;

    double cp = c/point;

    double ci[point];

    double xi[point];

    double x1[point];

    double yi[point];

    double y1[point];

    for (int i = 0; i < point; ++i) {

        ci[i] = i*cp;

    }

    for (int j = 0; j < point; ++j) {

        xi[j] = ci[j]*cos(mm);

        yi[j] = ci[j]*sin(mm);}

    for (int k = 0; k < point; ++k) {

        x1[k] = lat1+xi[k];

        y1[k] = lon1+yi[k];

        double X2r = x1[DD] - x;

```

```

double Y2r = y1[DD] - y;

double mr = Y2r/X2r;

double mmr = atan(mr);

Mr = mmr*57.295777;

double cr = sqrt((X2r*X2r)+(Y2r*Y2r));

Dr = (cr/(1/108.4))*1000; } }

```

```

void setdata (){

    Firebase.setInt("/bat", persen);

    Firebase.set("/latnow", latnow,6);

    Firebase.set("/lonnow", lonnow,6); }

```

```

void getdata (){

    lat1 = Firebase.getFloat("/lat1");

    lat2 = Firebase.getFloat("/lat2");

    lat3 = Firebase.getFloat("/lat3");

    lat4 = Firebase.getFloat("/lat4");

    lon1 = Firebase.getFloat("/lon1");

    lon2 = Firebase.getFloat("/lon2");

    lon3 = Firebase.getFloat("/lon3");

    lon4 = Firebase.getFloat("/lon4");

}

```

```

void gy () {

    heading = compass.readHeading();

    if(heading==0) {

    } else {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void step1 () {
    myservo.write(20); delay(2000); myservo.write(150);delay(2000); myservo.write(90); }

void step2 () { myservo.write(20); delay(2000); myservo.write(150);delay(2000); myservo.write(90); }

void stop1 () {digitalWrite(forword, LOW); digitalWrite(Reverse, LOW); }

void DN() {
    double Xn = latnow - x;
    double Yn = lonnow - y;
    double cn = sqrt((Xn*Xn)+(Yn*Yn));
    Dn = (cn/(1/108.4))*1000; }

void servor () { servorR = heading-radar;
Serial.println(servorR);}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน



ชื่อ-นามสกุล

นายนันทวัฒน์ เหมวงศ์

วัน เดือน ปีเกิด

16 สิงหาคม 2541

ที่อยู่

17 หมู่ 5 ต.หนองพอก อ.หนองพอก จ.ร้อยเอ็ด  
45210

ประวัติการศึกษา

จบการศึกษามัธยมปลายปีการศึกษา 2559 แผนการเรียน  
วิทยาศาสตร์-คณิตศาสตร์ โรงเรียนราชดำริ  
Tel. 098-9862128  
Email. Nanthawat.markk@gmail.com

ชื่อ-นามสกุล

นายปานท์ สุขเผือก

วัน เดือน ปีเกิด

28 เมษายน 2541

ที่อยู่

143 หมู่ 1 ต.แพรक्षा อ.เมือง จ.สมุทรปราการ  
10280

ประวัติการศึกษา

จบการศึกษามัธยมปลายปีการศึกษา 2559 แผนการเรียน  
วิทยาศาสตร์-คณิตศาสตร์ โรงเรียนสาธิตบางนา  
Tel. 092-3264848  
Email. Panat.sp28@gmail.com

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# ใบรับรองรูปเล่มปริญญาานิพนธ์

ปริญญาานิพนธ์ปีการศึกษา 2563

สาขาวิชาวิศวกรรมศาสตร์ หลักสูตรวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิทยาเขตชุมพรเขตรอุดมศักดิ์ จังหวัดชุมพร

ชื่อโครงการ เครื่องตัดหญ้าสนามอัตโนมัติโดยใช้จีพีเอสควบคุมผ่านแอปพลิเคชัน  
AUTOMATIC LAWN MOWER USING GPS CONTROLLER VIA  
APPLICATION

## ผู้จัดทำ

1. นายณัฐวัฒน์ เหมวงศ์ รหัสนักศึกษา 60511056
2. นายปาณัท สุขเผือก รหัสนักศึกษา 60511057

ด้วยข้าพเจ้านักศึกษาวิศวกรรมอิเล็กทรอนิกส์ สจล. วิทยาเขตชุมพรเขตรอุดมศักดิ์ จังหวัดชุมพร ได้จัดทำรูปเล่มปริญญาานิพนธ์ตามหลักสูตรปริญญาตรี สาขาวิศวกรรมศาสตร์ หลักสูตรวิศวกรรมอิเล็กทรอนิกส์ ซึ่งในการนี้ข้าพเจ้าได้แก้ไขเนื้อหาและจัดทำรูปเล่มตามข้อกำหนดของรูปเล่มปริญญาานิพนธ์เรียบร้อยแล้ว จึงขอให้อาจารย์ตรวจสอบ และรับรองความถูกต้องเหมาะสมของปริญญาานิพนธ์ในครั้งนี้ด้วย

## อาจารย์รับรองรูปเล่มปริญญาานิพนธ์

1. อาจารย์ ผศ.ดร.มนตรี ไชยชาญยุทธ์ ลงชื่อ
2. อาจารย์ พิมล ผลพุกษา ลงชื่อ
3. อาจารย์ที่ปรึกษา ว่าที่ร้อยตรี ศิลา ศิริมาสกุล ลงชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

