

รถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้
WHEELCHAIR PROGRAMMABLE PATH CONTROL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WHEELCHAIR PROGRAMMABLE PATH CONTROL



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2565
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

.....

หัวข้อปริญญาานิพนธ์ รถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้
WHEELCHAIR PROGRAMMABLE PATH CONTROL

นักศึกษาผู้จัดทำ นายกิตติภาพ ธนจิตสิริ รหัสนักศึกษา 63015016
 นายธราธิป สีนราช รหัสนักศึกษา 63015075
 นายภควัต ภูขาว รหัสนักศึกษา 63015129

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2565

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ. ทรงชัย วีระทวีมาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ รถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้

WHEELCHAIR PROGRAMMABLE PATH CONTROL

นักศึกษาผู้จัดทำ	นายกิตติภาพ	ธนจิตสิริ	รหัสนักศึกษา	63015016
	นายธราธิป	สินราช	รหัสนักศึกษา	63015075
	นายภควัต	ภูขาว	รหัสนักศึกษา	63015129
อาจารย์ที่ปรึกษา	รศ. ทรงชัย วีระทวีมาศ			
ปีการศึกษา	2565			

บทคัดย่อ

ปริญญานิพนธ์นี้จัดทำขึ้นโดยมีวัตถุประสงค์เพื่อสร้างรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้ เพื่อศึกษากระบวนการการนำอุปกรณ์เซนเซอร์การตรวจจับและเครื่องมือวัดมาใช้ในการสร้างรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้ โดยมีจุดมุ่งหมายในการทำงานคือรถวีลแชร์สามารถเคลื่อนที่ไปยังเส้นทางที่กำหนดได้ หลักการทำงานคือจะมีการโปรแกรมเส้นทางก่อน โดยการบังคับให้รถวีลแชร์เคลื่อนที่ไปตามเส้นทางที่เราต้องการด้วย Joy-Stick ในขณะที่รถวีลแชร์เกิดการเคลื่อนที่ระบบการทำงานจะทำหน้าที่บันทึกทิศทางการทำงานของ Joy-Stick ที่เคลื่อนที่ตามเส้นทางที่กำหนด โดยจะเก็บข้อมูลไว้ในหน่วยความจำของ Micro SD Card หลังจากโปรแกรมเส้นทางเรียบร้อยแล้วเมื่อสั่งให้รถวีลแชร์เคลื่อนที่ไปตามเส้นทางเดิมที่โปรแกรมไว้ รถวีลแชร์ก็จะสามารถเคลื่อนที่ไปตามเส้นทางที่โปรแกรมไว้ได้โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	WHEELCHAIR PROGRAMMABLE PATH CONTROL
Authors	Mr. Kittipob Thanajitsiri Mr. Tharathip Sinrach Mr. Pakawat Pookhao
Thesis Advisor	Assoc. Prof. Songchai Weerathaweemas
Year	2022

ABSTRACT

This thesis is prepared with the objective of creating a wheelchair programmable path control. To study the process of incorporating sensing sensors and instrumentation in building a wheelchair programmable path control. The purpose of the work is that the wheelchair can move to the specified route. The working principle is to program the route first, and force the wheelchair to move along the route we want by using Joy-Stick. When the wheelchair moves, the system records the direction of the Joy-Stick moving along the specified route. The data will be stored in the memory of the Micro SD Card. After the route has been successfully programmed when instructing the wheelchair to move along the same programmed route the wheelchair can automatically follow the programmed route.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับความปรึกษาและความอนุเคราะห์จาก อาจารย์ที่ปรึกษา รองศาสตราจารย์ ทรงชัย วีระทวีมาศ อีกทั้งยังเอื้อเพื่ออุปกรณ์และเครื่องมือต่าง ๆ ในการทำปริญญาานิพนธ์นี้ ผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์สาขาวิศวกรรมการวัดและควบคุมทุกท่าน ที่ให้คำแนะนำ และได้ถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้าอันเป็นประโยชน์ต่อการทำปริญญาานิพนธ์ฉบับนี้

และที่ลืมเสียไม่ได้คือ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ซึ่งเป็นที่รักและเคารพยิ่ง ที่ได้ให้การสนับสนุนและเป็นแรงบันดาลใจในการทำปริญญาานิพนธ์ฉบับนี้ คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์ฉบับนี้ผู้วิจัยขอมอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญรูป.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	3
2.1 พัฒนาการของรถวีลแชร์.....	3
2.1.1 รถเข็นธรรมดา.....	3
2.1.2 รถเข็นแบบไฟฟ้า.....	4
2.1.3 สรุปรถเข็น.....	4
2.2 ประเภทของมอเตอร์สำหรับวีลแชร์ไฟฟ้า.....	4
2.2.1 มอเตอร์กระแสตรง (DC Motor).....	4
2.2.1.1 มอเตอร์กระแสตรงแบบไร้แปรงถ่าน.....	5
2.3 ประเภทแบตเตอรี่สำหรับวีลแชร์ไฟฟ้า.....	7
2.3.1 แบตเตอรี่ลิเทียมฟอสเฟต (LiFePO4 Battery).....	7
2.3.2 เทคโนโลยีระบบบริหารจัดการแบตเตอรี่.....	7
2.4 ชุดควบคุม.....	10
2.4.1 ก้านควบคุม (Joystick).....	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.4.1.1 หลักการทำงานของก้านควบคุม.....	10
2.4.2 บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano.....	11
2.4.2.1 ภาควัดไฟ.....	12
2.4.2.2 หน่วยความจำ.....	12
2.4.2.3 พอร์ตอินพุต - เอาต์พุต.....	13
2.4.2.4 การสื่อสาร.....	13
2.4.3 บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560.....	13
2.4.3.1 ขาหรือ Pin ทั่วไป.....	14
2.4.3.2 หน่วยความจำ.....	14
2.4.3.3 พอร์ตอินพุต - เอาต์พุต.....	14
2.4.3.4 ฟังก์ชันการทำงานอื่น ๆ.....	15
2.4.4 ไจโรสโคป.....	15
2.4.4.1 หลักการทำงานของไจโรสโคป.....	16
2.5 โปรแกรม Arduino.....	16
2.6 Ultrasonic Module HC-SR04.....	17
2.7 LCD Display.....	18
2.8 Micro servo motor.....	19
2.9 Encoder Omron E6B2-CWZ6C.....	20
2.10 GY-273 3-axis Compass Module.....	21
2.11 PCA9685 16-Channel 12-bit PWM Servo shield I2C interface.....	22
2.12 MicroSD Card Adapter.....	22
2.13 การสื่อสารผ่าน UART ของบอร์ดไมโครคอนโทรลเลอร์.....	22
2.14 บอร์ดขับเคลื่อนมอเตอร์ BTS7960B H-Bridge Motor Driver 6-27V 43A.....	24
บทที่ 3 วิธีดำเนินการ.....	25
3.1 ขั้นตอนการทำงาน.....	25
3.2 ตรวจสอบการทำงานของรถวิลแชร์.....	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.3 ศึกษาวงจรและโปรแกรมการทำงานของรถวีลแชร์ไฟฟ้า.....	28
3.4 ศึกษาระบบการเคลื่อนที่แบบอัตโนมัติ.....	28
3.5 ศึกษา และจัดหาอุปกรณ์ที่ใช้รับ – ส่ง และบันทึกข้อมูล อุปกรณ์ตรวจจับสิ่งกีดขวาง.....	28
3.6 ออกแบบวงจร และออกแบบโปรแกรมการทำงานให้รถวีลแชร์.....	30
สามารถเคลื่อนที่ไปอย่างอัตโนมัติ	
บทที่ 4 ผลการทดลอง.....	37
4.1 การทดลองการเก็บข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์.....	37
4.2 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์.....	38
4.3 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาทางแยก.....	39
4.4 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายทางแยก.....	41
4.5 การทดลองการเคลื่อนที่แบบอัตโนมัติในเส้นทางที่มีการเลี้ยว มากกว่า 1 ครั้ง.....	42
4.6 การทดลองการส่งค่าของ Joy-Stick ระหว่างบอร์ดรับข้อมูลและบอร์ดขับเคลื่อน.....	44
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....	46
5.1 สรุปผลการทดลอง.....	46
5.2 ปัญหาและอุปสรรคที่พบระหว่างดำเนินการ.....	47
5.3 ข้อเสนอแนะเพิ่มเติม.....	47
บรรณานุกรม.....	48
ภาคผนวก.....	49
ภาคผนวก ก.....	50
ภาคผนวก ข.....	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
4.1 การเก็บข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์.....	38
4.2 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์.....	39
4.3 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยก.....	40
4.5 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยก.....	42
4.6 การทดลองการเคลื่อนที่แบบอัตโนมัติในเส้นทางที่มีการเลี้ยวมากกว่า 1 ครั้ง.....	43



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1	รถเข็นธรรมดา..... 3
2.2	รถเข็นแบบไฟฟ้า..... 4
2.3	โครงสร้างภายในของมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน..... 5
2.4	กำลังสูญเสียทั้งหมดของมอเตอร์กระแสตรงแบบไร้แปรงถ่าน..... 6
2.5	แบตเตอรี่ลิเทียมฟอสเฟต..... 7
2.6	ขั้นตอนการอัดประจุของแบตเตอรี่ชนิดลิเทียมไอออน..... 9
2.7	ก้านควบคุม (Joy-Stick)..... 10
2.8	เซ็นเซอร์ตำแหน่งอีกรูปแบบหนึ่งที่ใช้กับดิจิทัลจอยสติ๊ก..... 11
2.9	บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano..... 12
2.10	บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560..... 14
2.11	ใจโรสโคป..... 15
2.12	หลักการทำงานของใจโรสโคป..... 16
2.13	โปรแกรม Arduino..... 17
2.14	Ultrasonic Module HC-SR04..... 18
2.15	LCD Display..... 19
2.16	Micro servo motor..... 20
2.17	Encoder Omron E6B2-CWZ6C..... 21
2.18	GY-273 3-axis Compass Module..... 21
2.19	PCA9685 16-Channel 12-bit PWM Servo shield I2C interface..... 22
2.20	MicroSD Card Adapter..... 22
2.21	การเชื่อมต่อการทำงานของการสื่อสารผ่าน UART..... 23
2.22	ลักษณะการเชื่อมต่อการรับ - ส่งข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์..... 23
2.23	บอร์ดขับมอเตอร์ BTS7960B..... 24
3.1	แผนผังขั้นตอนการดำเนินการ..... 25
3.2	แผนผังขั้นตอนการดำเนินการ(ต่อ)..... 26
3.3	แผนผังขั้นตอนการดำเนินการ(ต่อ)..... 27
3.4	ตรวจสอบรถวิลแชร์ไฟฟ้า..... 27
3.5	บัตรกริเพิ่ม Socket ลงในวงจร..... 28
3.6	โปรแกรมการทำงานของรถวิลแชร์ไฟฟ้า..... 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.7	ไดอะแกรมการเชื่อมต่อของวงจรการทำงานแบบคร่าว ๆ.....29
3.8	รูปลักษณะการเชื่อมต่อของวงจรการทำงาน.....30
3.9	ตัวอย่างโปรแกรมการทำงานของอุปกรณ์แต่ละชนิด.....30
3.10	การทดสอบการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวาง.....31
3.11	การทดสอบการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวาง (ต่อ).....31
3.12	การทดสอบการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวางร่วมกับอุปกรณ์จดจำตำแหน่ง.....32
3.13	วงจรการทำงานที่ต่อ Encoder และ Adapter Micro SD Card เรียบร้อยแล้ว.....32
3.14	ค่าตัวอย่างข้อมูลของการทำงานของอุปกรณ์หรือเซ็นเซอร์ที่ต่ออยู่กับวงจร.....32
3.15	การออกแบบการติดตั้งอุปกรณ์หรือเซ็นเซอร์.....33
3.16	การออกแบบกล่องบรรจุวงจรการทำงานของรถวีลแชร์.....33
3.17	ประกอบอุปกรณ์หรือเซ็นเซอร์ วงจรการทำงานเข้ากับตัวรถตามที่ออกแบบไว้.....34
3.18	เดินสายไฟของอุปกรณ์หรือเซ็นเซอร์.....34
3.19	แผนผังโปรแกรมการทำงานของรถวีลแชร์เคลื่อนที่แบบอัตโนมัติ.....34
3.20	แผนผังโปรแกรมการทำงานของรถวีลแชร์เคลื่อนที่แบบอัตโนมัติ (ต่อ).....35
3.21	การเขียนโปรแกรมการทำงานของรถวีลแชร์เคลื่อนที่แบบอัตโนมัติ.....36
3.22	ทดลองโปรแกรมการทำงานของรถวีลแชร์เคลื่อนที่แบบอัตโนมัติ.....36
4.1	ระยะเส้นทางการเคลื่อนที่ของรถวีลแชร์ในแนวทางตรง.....38
4.2	ระยะเส้นทางการเคลื่อนที่ของรถวีลแชร์ในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยก.....40
4.3	ระยะเส้นทางการเคลื่อนที่ของรถวีลแชร์ในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยก.....41
4.4	ระยะเส้นทางการเคลื่อนที่ของรถวีลแชร์ในเส้นทางที่มีการเลี้ยวมากกว่า 1 ครั้ง.....43
4.5	บล็อกไดอะแกรมการทำงานในการส่งค่า Joy-Stick ระหว่างบอร์ดรับข้อมูล.....45 และบอร์ดขับเคลื่อน
ก-1	ตัวอย่างพื้นที่เส้นทางการเดินรถวีลแชร์.....51
ก-2	ตัวอย่างพื้นที่เส้นทางการเดินรถวีลแชร์ (ต่อ).....51
ก-3	ตั้งรถวีลแชร์บนเส้นทางการเคลื่อนที่.....52
ก-4	การเปิดเครื่องเพื่อเริ่มการทำงานของรถวีลแชร์.....52
ก-5	หน้าจอแสดงผล และปุ่มกดเพื่อเลือกการทำงานของรถวีลแชร์.....53
ก-6	กดปุ่มเพื่อเลือกโหมดการทำงานของรถวีลแชร์.....53
ก-7	หน้าจอแสดงผลเมื่อกดปุ่มที่ 1 หรือหน้าจอแสดงผลการทำงาน Manual Mode.....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก-8 การเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick เพื่อเก็บเส้นทางการเคลื่อนที่ที่เลือกไว้.....	54
ก-9 การเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick เพื่อเก็บเส้นทางการเคลื่อนที่ที่เลือกไว้ (ต่อ).....	54
ก-10 การเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick เพื่อเก็บเส้นทางการเคลื่อนที่ที่เลือกไว้ (ต่อ).....	54
ก-11 กดปุ่มเพื่อเลือกโหมดการทำงานของรถวีลแชร์อีกครั้ง.....	55
ก-12 หน้าจอแสดงผลเมื่อกดปุ่มที่ 2 หรือหน้าจอแสดงผลการทำงาน Auto Mode.....	55
ก-13 การเคลื่อนที่รถวีลแชร์อัตโนมัติ ตามเส้นทางการเคลื่อนที่ที่บันทึกข้อมูลเอาไว้.....	55
ก-14 การเคลื่อนที่รถวีลแชร์อัตโนมัติ ตามเส้นทางการเคลื่อนที่ที่บันทึกข้อมูลเอาไว้ (ต่อ).....	56



บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญานิพนธ์

ในประเทศไทยนั้นมีผู้ป่วยและผู้สูงอายุจำนวนมากเพิ่มขึ้นซึ่งก็จะรวมไปถึงผู้ป่วยหรือผู้สูงอายุที่ช่วยเหลือตัวเองไม่ได้ทำให้ในการเคลื่อนที่ไปไหนมาไหนนั้นจะเกิดความลำบากมากเพราะเนื่องด้วยอายุกับปัญหาทางสุขภาพ จึงได้มีนวัตกรรมต่างๆ ที่ได้เข้ามาช่วยบรรเทาในส่วนนี้ อย่างเช่น รถเข็น หรือ Wheelchair จะเป็นอุปกรณ์ที่ใช้ในการอำนวยความสะดวกในเรื่องของการเคลื่อนย้ายผู้ป่วยหรือคนสูงอายุ ไปในที่ต่างๆ เพื่อให้สามารถที่จะรับบริการต่างๆ ได้ อย่างเช่น บริการด้านสุขภาพ

ซึ่งในปัจจุบันรถเข็น หรือ Wheelchair นั้นก็ถือว่าเป็นอุปกรณ์สำคัญอย่างหนึ่งภายในโรงพยาบาลที่ใช้ในการเคลื่อนย้ายผู้ป่วยภายในโรงพยาบาล เพื่อให้สามารถเข้ารับบริการในการตรวจวินิจฉัยตลอดจนการรักษาพยาบาลตามหน่วยงานต่างๆ ของโรงพยาบาล โดยการเคลื่อนย้ายผู้ป่วยนั้นจะต้องมีความรวดเร็ว ทันเวลา ถูกต้อง และปลอดภัย ซึ่งหากเกิดข้อผิดพลาดขึ้น อย่างเช่น ส่งลำช้า ส่งผิดหน่วยงาน อาจส่งผลเสียต่อระบบการรักษาพยาบาลของผู้รับบริการรายถัดไปได้ และในการใช้งานรถเข็น หรือ Wheelchair นั้นจำเป็นที่จะต้องใช้เจ้าหน้าที่ที่อยู่ภายในโรงพยาบาลเพื่อเข็นผู้ป่วยไปตามหน่วยงานต่างๆ ทำให้ในบางครั้งเกิดเหตุที่บุคลากรมีจำนวนไม่พอ จึงทำให้เกิดความล่าช้าในการเคลื่อนย้ายได้

จากปัญหาที่เกิดขึ้นข้างต้น คณะผู้จัดทำจึงได้ริเริ่มทำโครงการวีลแชร์ที่สามารถโปรแกรมเส้นทางได้ เพื่อแก้ปัญหาการเคลื่อนย้ายผู้ป่วยไปยังหน่วยงานต่างๆ ของโรงพยาบาล และลดการใช้บุคลากรในการเข็นรถ โดยจะเป็นการประยุกต์การทำงานของรถเข็นไฟฟ้าที่มีการเคลื่อนที่แบบกึ่งอัตโนมัติให้สามารถมีการเคลื่อนที่เป็นแบบอัตโนมัติโดยที่ไม่ต้องใช้คนเข็น สามารถที่จะทำการตั้งค่าเส้นทางการเดินทางไปตามจุดต่างๆ ได้อย่างอัตโนมัติ

1.2 วัตถุประสงค์ของปัญญานิพนธ์

1. เพื่อสร้างรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้
2. เพื่อศึกษากระบวนการการนำอุปกรณ์เช่น เซอร์การตรวจจับและเครื่องมือวัด มาใช้ในการสร้างรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้
3. เพื่อศึกษาโปรแกรมการทำงานร่วมกับอุปกรณ์ไมโครคอนโทรลเลอร์ที่ใช้ควบคุมวีลแชร์ที่สามารถโปรแกรมเส้นทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญาโท

1. สร้างรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้
2. สร้างรถวีลแชร์ที่มีการเคลื่อนที่แบบอัตโนมัติได้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของตัวรถวีลแชร์ไฟฟ้า
2. ศึกษาการทำงานของอุปกรณ์ในการตรวจจับสิ่งกีดขวาง
3. ศึกษาข้อมูลการเขียนโปรแกรม
4. ศึกษาเกี่ยวกับการเขียนโค้ดโปรแกรมการทำงานอัตโนมัติ
5. ศึกษาการออกแบบการติดตั้งอุปกรณ์
6. ศึกษาวิธีการรับและส่งข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์
7. ศึกษาวิธีการออกแบบเส้นทางการเคลื่อนที่ของรถวีลแชร์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถประยุกต์การทำงานของรถวีลแชร์ไฟฟ้าที่มีการเคลื่อนที่โดยใช้ Joy-Stick ให้สามารถเคลื่อนที่แบบอัตโนมัติตามเส้นทางที่ต้องการได้
2. สามารถพัฒนาโปรแกรมการทำงานของรถวีลแชร์ให้มีการเคลื่อนที่เป็นแบบอัตโนมัติได้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 พัฒนาการของรถวีลแชร์

ปัจจุบันรถวีลแชร์ของผู้พิการและผู้สูงอายุได้มีการพัฒนาไปอย่างมากและมีคุณภาพสามารถอำนวยความสะดวกได้หลากหลาย โดยรถวีลแชร์ของผู้พิการและผู้สูงอายุนั้นที่มีราคาที่สูง ค่อนข้างสูง เช่น Ultra-lightweight wheelchair ซึ่งมีลักษณะโครงสร้างเป็นโลหะเบา อะลูมิเนียม หรือไททานเนียม มีระบบการบังคับ การเอนพนัก ระบบความปลอดภัยรวมถึงระบบล้อที่เคลื่อนไหวได้คล่องแคล่ว ผู้พิการและผู้สูงอายุจึงสามารถเรียนรู้และบังคับได้อย่างรวดเร็ว เหมาะสำหรับผู้พิการและผู้สูงอายุที่จำเป็นต้องใช้ในระยะเวลา สำหรับรถวีลแชร์ผู้พิการและผู้สูงอายุที่มีราคาต่ำลงมา ถึงแม้จะไม่ได้ผลิตด้วยโลหะเบา แต่มีความทนทาน ใช้งานง่าย มีระบบอำนวยความสะดวกพอสมควร จึงเหมาะสำหรับคนใช้ทั่วไป แต่แตกต่างกันตรงที่ผลิตทั้งในประเทศและผลิตต่างประเทศ การใช้วัสดุที่ไม่มี ความทนทาน มักโค้งงอ หักหรือบิดง่าย ที่สำคัญระบบการบังคับและระบบล้อไม่มีความคล่องตัว ซึ่งคนใช้จะมีปัญหากับการใช้งาน

2.1.1 วีลแชร์แบบธรรมดา

วีลแชร์หรือรถเข็นธรรมดานั้นช่วยให้ผู้สูงอายุหรือผู้พิการสามารถเคลื่อนที่ไปยังทิศทางที่ต้องการโดยการให้ผู้อื่นเข็น แต่ถ้าไม่มีคนเข็นให้ผู้สูงอายุหรือผู้พิการจะต้องออกแรงในการหมุนล้อเองเพื่อให้รถเข็นเคลื่อนที่ไปในทิศทางที่ต้องการซึ่งเป็นการยากและทำให้ต้องออกแรงอย่างมากในการเคลื่อนที่ แต่รถเข็นชนิดนี้ไม่เหมาะสำหรับผู้พิการที่ช่วยเหลือตนเองไม่ได้ เพราะลำบากในการเคลื่อนที่ซึ่งรถเข็นมีขนาดใหญ่และมีน้ำหนักมาก แต่มีราคาที่สูง ทนทาน ใช้งานง่ายหาซื้อได้ทั่วไป



รูปที่ 2.1 รถเข็นธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 วีลแชร์แบบไฟฟ้า

วีลแชร์หรือรถเข็นไฟฟ้านั้น ช่วยให้ผู้สูงอายุหรือผู้พิการสามารถเคลื่อนที่ไปยังทิศทางที่ต้องการได้โดยไม่ต้องใช้คนเข็น เพราะมีการติดตั้งระบบไฟฟ้าในการเคลื่อนที่ผู้ใช้สามารถยังผ่านจาก Joystick โดยไม่ต้องออกแรงเหมือนกับรถเข็นธรรมดา แต่มีราคาสูงมาก



รูปที่ 2.2 รถเข็นแบบไฟฟ้า

2.1.3 สรุปรถวีลแชร์

รถเข็นแบบธรรมดาผู้ใช้ต้องออกแรงอย่างมากเพื่อให้รถเข็นเคลื่อนที่และไม่เหมาะสมกับผู้พิการมากนัก แต่มีราคาถูก ส่วนรถเข็นไฟฟ้าผู้ใช้ควบคุมการเคลื่อนที่และทิศทางผ่าน Joy-Stick แต่มีราคาค่อนข้างสูงดังนั้นงานวิจัยนี้จึงเป็นการนำเอารถเข็นแบบธรรมดาซึ่งมีราคาถูกนำมาติดตั้งระบบไฟฟ้าเพื่อให้ผู้สูงอายุและผู้พิการสามารถเคลื่อนที่ไปอย่างสะดวก

2.2 ประเภทของมอเตอร์สำหรับวีลแชร์ไฟฟ้า

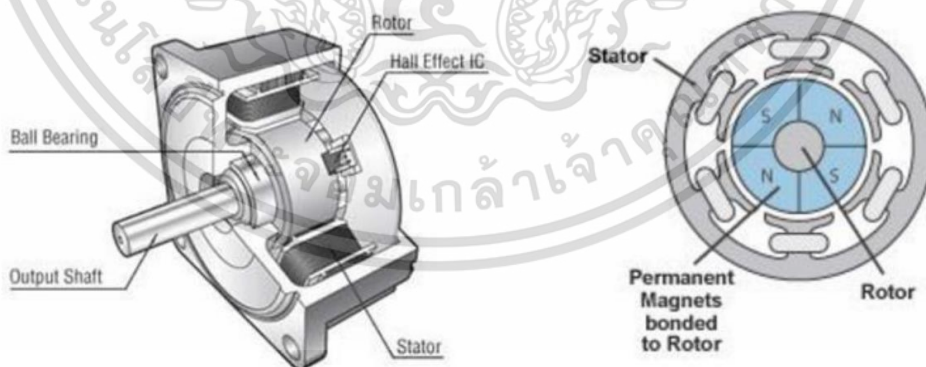
หากพิจารณาประเภทของมอเตอร์สำหรับวีลแชร์ไฟฟ้าพบว่า มีการใช้งานมอเตอร์ในรูปแบบ คือ มอเตอร์กระแสตรง (DC Motor) ซึ่งจะกล่าวถึงรายละเอียดและประเภทย่อยของมอเตอร์ที่ดังนี้

2.2.1 มอเตอร์กระแสตรง (DC Motor)

มอเตอร์กระแสตรงเป็นรูปแบบของมอเตอร์ที่มีความซับซ้อนในการจ่ายพลังงานไฟฟ้าน้อย เนื่องจากมอเตอร์กระแสตรงสามารถรับไฟฟ้ากระแสตรงจากแหล่งจ่าย เช่น แบตเตอรี่ และสามารถทำงานได้โดยไม่ต้องการระบบไฟฟ้าเพื่อเพิ่มกำลัง โดยมอเตอร์กระแสตรงที่ได้รับความนิยมนำมาใช้อย่างแพร่หลายในยานยนต์ไฟฟ้า

2.2.1.1 มอเตอร์กระแสตรงแบบไร้แปรงถ่าน (Brushless DC Motor)

ด้วยราคาของอุปกรณ์ไฟฟ้าที่ลดต่ำลงส่งผลให้ราคาของมอเตอร์กระแสตรงแบบไร้แปรงถ่านสูงกว่ามอเตอร์กระแสตรงแบบมีแปรงถ่านเล็กน้อย จึงทำให้มอเตอร์กระแสตรงแบบไร้แปรงถ่านได้รับความนิยมนำมาใช้เพื่อขับเคลื่อนยานยนต์ไฟฟ้าเพิ่มมากขึ้น มอเตอร์กระแสตรงแบบไร้แปรงถ่านมีโครงสร้างสลับกับมอเตอร์กระแสตรงแบบมีแปรงถ่าน โดยมีแกนหมุน (Rotor) เป็นแม่เหล็กถาวร และมีขดลวดเหนี่ยวนำอยู่ที่สเตเตอร์ โดยขดลวดเหนี่ยวนำมีจำนวนอย่างน้อยสามชุด มอเตอร์ชนิดนี้จึงสามารถทำงานได้โดยการจ่ายไฟฟ้าไปยังขดลวดแต่ละชุดเป็นเฟสย่อยๆ สลับกันไปเรื่อยๆ เพื่อให้เกิดสนามแม่เหล็กดึงและผลักแกนหมุนอย่างต่อเนื่อง โดยมีการตรวจจับตำแหน่งเพื่อเริ่มทำงานโดยใช้อุปกรณ์ตรวจจับสนามแม่เหล็ก (Hall Sensor) ดังแสดงในรูปที่ 2.3 มอเตอร์ชนิดนี้สามารถปรับแรงบิดได้โดยปรับการจ่ายกระแสไฟฟ้า และปรับความเร็วรอบได้โดยการปรับความถี่ในการสลับกระแสไฟฟ้าของขดลวด ซึ่งสามารถเรียกการทำงานของมอเตอร์ที่มีความเร็วในการหมุนตรงกับความเร็วของการหมุนของสนามแม่เหล็กนี้ว่าเป็นการทำงานแบบ Synchronous นั่นเอง ถึงแม้ว่ามอเตอร์กระแสตรงแบบไร้แปรงถ่านจะมีข้อดีคือมีประสิทธิภาพที่สูงและไม่ต้องมีแปรงถ่านซึ่งจะสึกหรอได้ง่ายเมื่อใช้งานเป็นเวลานาน แต่อย่างไรก็ตามจะมีข้อเสียคือต้องมีชุดควบคุมอิเล็กทรอนิกส์ที่ซับซ้อน ซึ่งสามารถทำงานในสภาวะการใช้งานต่างๆ ของยานยนต์ได้ เช่น สามารถกันน้ำ ทนต่อการสั่นสะเทือน และความร้อน เป็นต้น นอกจากนี้สายไฟที่ต่อเข้ามอเตอร์และชุดควบคุมอิเล็กทรอนิกส์ยังมีหลายสาย ต่างกับมอเตอร์กระแสตรงแบบมีแปรงถ่านที่เพียงจ่ายไฟที่สายไฟทั้งสองด้านก็ทำงานได้ ดังนั้นการติดตั้งสายไฟจึงต้องระมัดระวังเพื่อป้องกันไม่ให้เกิดการลัดวงจรเกิดขึ้น

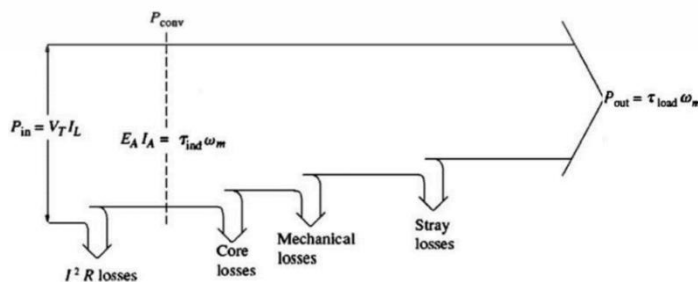


รูปที่ 2.3 โครงสร้างภายในของมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้งานมอเตอร์กระแสตรงแบบไร้แปรงถ่าน พบว่ามีลักษณะการสร้างแรงบิดและกำลังคล้ายกับมอเตอร์กระแสตรงแบบมีแปรงถ่าน แต่จะแตกต่างกันตรงแกนหมุนที่เป็นแม่เหล็กถาวร จึงทำให้เกิดปัญหาแรงดันไฟฟ้าย้อนกลับเมื่อใช้งานมอเตอร์ในรอบความเร็วที่สูงขึ้น เมื่อพิจารณาตัวโรเตอร์ประกอบด้วย 4 ส่วนหลักๆด้วยกัน คือ

- แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลาจะวางอยู่บนแบร์ริงเพื่อบังคับให้หมุนอยู่ในแนวหนึ่งไม่มีการสั่นสะเทือนได้
- แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งจะสร้างแรงบิด (Torque)
- คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่ๆแต่ละซี่ฉนวนเรียกว่าไมก้า (mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ส่วนหัวซี่ในคอมมิวเตเตอร์จะมีร่องสำหรับใส่ปลายสายของขดลวดอาร์มาเจอร์ในตัวคอมมิวเตเตอร์เพื่อให้มีความแน่นกับแกนเพลลาซึ่งเป็นรูปกลมทรงกระบอก โดยการสัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยังขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กเพื่อให้เกิดการหักล้างขึ้นและเสริมกลับกับเส้นแรงแม่เหล็กอีกส่วนซึ่งเกิดจากขดลวดของขั้วแม่เหล็กที่เรียกกันว่า ปฏิกริยามอเตอร์
- ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดที่พันอยู่ในร่องของมอเตอร์เรียกว่า สล็อต (Slot) ของแกนอาร์มาเจอร์ซึ่งขนาดของลวดจะมีความใหญ่เล็กหรือเล็กนั้น จะสัมพันธ์กับจำนวนรอบที่มากหรือน้อยโดยขึ้นอยู่กับการออกแบบของตัวโรเตอร์ ให้เหมาะสมกับงานต่างๆ ในการใช้งานมอเตอร์กระแสตรงแบบไร้แปรงถ่าน ซึ่งเราจะพบว่ามีลักษณะการสร้างแรงบิด และกำลังคล้ายกับมอเตอร์กระแสตรงแบบมีแปรงถ่าน และจากการที่แกนหมุนเป็นแม่เหล็กถาวร จึงทำให้เกิดปัญหาแรงดันไฟฟ้าย้อนกลับเมื่อใช้งานมอเตอร์ที่รอบความเร็วสูง
- หากพิจารณากำลังสูญเสียและประสิทธิภาพของมอเตอร์กระแสตรงแบบไร้แปรงถ่าน ที่แสดงดังรูปที่ 2.4 พบว่ามีการสูญเสียกำลังจากการหมุนของมอเตอร์และแรงเสียดทานภายในมอเตอร์เป็นหลัก



รูปที่ 2.4 กำลังที่สูญเสียทั้งหมด (Total Losses) ของมอเตอร์กระแสตรงแบบไร้แปรงถ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ประเภทแบตเตอรี่สำหรับวีลแชร์ไฟฟ้า

แบตเตอรี่ถือเป็นอุปกรณ์ทางไฟฟ้าที่สำคัญมากขึ้นหนึ่งของวีลแชร์ไฟฟ้า เนื่องจากเป็นอุปกรณ์ที่ใช้ปฏิกิริยาทางเคมีไฟฟ้าเพื่อกักเก็บพลังงานเพื่อใช้ในการขับเคลื่อน โดยส่วนประกอบหลักของแบตเตอรี่ประกอบด้วย 3 ส่วนหลักๆคือ ขั้วบวก (แคโทด) ขั้วลบ (แอโนด) และสารละลายอิเล็กโทรไลต์ ซึ่งเมื่อแบตเตอรี่ถูกใช้งาน ขั้วลบหรือขั้วแอโนดเป็นขั้วที่เกิดปฏิกิริยาเคมีส่งผลให้เกิดการเคลื่อนที่ของอิเล็กตรอน (ปฏิกิริยาออกซิเดชัน) และขั้วบวก หรือขั้วแคโทดเป็นขั้วที่เกิดปฏิกิริยาเคมีที่รับอิเล็กตรอน (ปฏิกิริยารีดักชัน) โดยอิเล็กตรอนที่ได้จากปฏิกิริยาออกซิเดชันจะวิ่งผ่านลวดโลหะก่อให้เกิดการไหลของกระแสไฟฟ้าขึ้น โดยที่อิเล็กโทรไลต์เป็นสารละลายที่ไม่นำอิเล็กตรอน แต่มีหน้าที่ส่งผ่านไอออนที่จำเป็นต่อการเกิดปฏิกิริยาดังกล่าวได้อย่างต่อเนื่องจนกว่าแบตเตอรี่จะหมด

2.3.1 แบตเตอรี่ลิเทียมฟอสเฟต (LiFePO4 Battery)

แบตเตอรี่ Life PO4 จะสามารถตอบโจทย์เรื่องความปลอดภัยได้ดีมาก เนื่องจากแบตเตอรี่ประเภทนี้จะไม่ระเบิดเมื่อไม่เกิดการกระแสที่รุนแรงหรืออยู่ในที่อุณหภูมิสูงเกินไป อันเป็นผลจากคุณสมบัติทางอุณหพลศาสตร์ และความเสถียรของตัววัสดุซึ่งทำให้มีความร้อนเกิดขึ้นต่ำกว่าลิเทียมชนิดอื่นมาก ด้วยคุณสมบัติด้านความปลอดภัยในข้อนี้ช่วงแรก แบตเตอรี่ชนิดนี้จึงถูกนำไปใช้งานในยานยนต์ไฟฟ้า และเริ่มมีความนิยมเพิ่มมากขึ้นในงานอื่นตามมา แบตเตอรี่กลุ่มนี้ จะเน้นไปในงานที่ต้องการความปลอดภัยสูง และสามารถจ่ายไฟได้จากการชาร์จไว เมื่อต้องการความทนทานสามารถทำได้ทุกขนาด ทุกแรงดัน ตามความต้องการของลูกค้า ตั้งแต่ขนาดเล็กๆ เช่น ใช้ในอุปกรณ์พกพาขนาดเล็ก หรืออุปกรณ์ไร้สายต่าง (สว่าน เครื่องมือวัด เครื่องมือแพทย์ โดรน สกู๊ตเตอร์ไฟฟ้า Robot ไปถึงระดับแบตเตอรี่สำหรับรถยนต์ไฟฟ้า ระบบสำรองไฟโซล่าเซลล์ ระบบจ่ายไฟ สำรองไฟขนาดใหญ่สำหรับเครื่องจักร อาคาร/โรงงาน ระบบสื่อสาร



รูปที่ 2.5 แบตเตอรี่ลิเทียมฟอสเฟต

2.3.2 เทคโนโลยีระบบบริหารจัดการแบตเตอรี่

ในการใช้งานยานยนต์ไฟฟ้าอย่างมีประสิทธิภาพและมีความปลอดภัย จำเป็นต้องอาศัยการจัดการพลังงานอย่างเป็นระบบ โดยอุปกรณ์ตัวหนึ่งที่มีบทบาทสำคัญมากในการบริหารจัดการเอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพลังงานเรียกว่า "ระบบบริหารจัดการแบตเตอรี่ หรือ Battery Management System (BMS)" ซึ่งอุปกรณ์ดังกล่าวมีหน้าที่ในการจัดการ โดยให้มีการใช้พลังงานที่กักเก็บภายในแบตเตอรี่อย่างมีประสิทธิภาพสูงสุดและลดความเสี่ยงที่แบตเตอรี่จะเกิดความเสียหายจากการใช้งานให้มากที่สุดผ่านกระบวนการตรวจสอบและควบคุมกระบวนการอัดและคายประจุของแบตเตอรี่ ทั้งนี้กระบวนการตรวจสอบและควบคุมการใช้งานแบตเตอรี่มีรายละเอียดดังต่อไปนี้

- ควบคุมการอัดประจุไฟฟ้าของแบตเตอรี่ โดยป้องกันไม่ให้เกิดการอัดประจุมากเกินไป (Overcharging) เพื่อยืดอายุการใช้งานของแบตเตอรี่
- ตรวจสอบการจ่ายประจุไฟฟ้าของแบตเตอรี่เพื่อป้องกันความเสียหายที่อาจเกิดกับแบตเตอรี่ โดยตัดการจ่ายกระแสไฟฟ้าในกรณีที่ประจุไฟฟ้าในแบตเตอรี่หมด
- ตรวจสอบค่า SOC ของแบตเตอรี่เพื่อควบคุมการอัดและจ่ายประจุไฟฟ้า
- ให้กำลังไฟฟ้าโดยใช้แรงดันต่ำสุด ผ่านอุปกรณ์แปลงผันพลังงานไฟฟ้ากระแสตรง (DC/DC conversion) เพื่อยืดอายุการใช้งาน

ระบบบริหารจัดการแบตเตอรี่สำหรับยานยนต์ไฟฟ้า จะมีความซับซ้อนมากกว่าระบบเดียวกันของอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ทั้งนี้เนื่องมาจากแบตเตอรี่ของยานยนต์ไฟฟ้ามีจำนวนเซลล์แบตเตอรี่มากกว่าอุปกรณ์ไฟฟ้าอื่นๆมาก อีกทั้งแบตเตอรี่ของยานยนต์ไฟฟ้าจะถูกใช้งานที่สถานะกระแสและแรงดันที่สูงอยู่ตลอดเวลาเพื่อให้กำลังไฟฟ้าเพียงพอต่อการขับเคลื่อน ดังนั้นระบบบริหารจัดการแบตเตอรี่จะต้องมีหน้าที่หลัก 3 ส่วน ได้แก่

1. Cell monitoring คือการแสดงผลค่าสถานะต่าง ๆ ของแบตเตอรี่ เช่น แรงดัน อุณหภูมิ และ State of Charge (SOC) เป็นต้น
2. Cell protection คือการตัดการทำงานของแบตเตอรี่ เมื่ออยู่ในสถานะอันตราย เช่นแรงดันเกิน (Over-voltage) กระแสเกิน (Over-current อุณหภูมิสูงเกิน (Over-temperature) และแรงดันต่ำ (Undervoltage) เป็นต้น
3. Cell balancing คือการปรับให้เซลล์แต่ละเซลล์มีพลังงานเท่าๆ กันระหว่าง การอัดและคายประจุไฟฟ้า

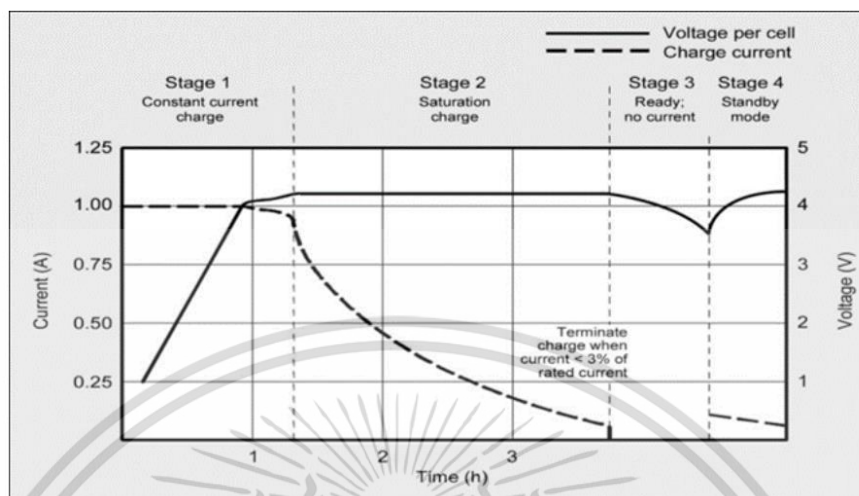
ในการใช้งานแบตเตอรี่ประเภทลิเทียมไอออน ซึ่งเป็นประเภทแบตเตอรี่ ที่นิยมใช้ในยานยนต์ไฟฟ้าจะต้องมีการอัดประจุไฟฟ้าตามวัฏจักร ดังแสดงในรูปที่ 2.6 ซึ่งแบ่งการอัดประจุไฟฟ้าออกเป็น 4 ระยะ ได้แก่

ระยะที่ 1 Constant current charge คือ การอัดประจุไฟฟ้าด้วยการแสดงให้แรงดันจะเข้าสู่ค่าที่กำหนด

ระยะที่ 2 Saturation charge เมื่อแรงดันเข้าสู่ค่าที่กำหนดแรงดันจะคงที่จนกระทั่งอัดประจุได้เต็ม ในขณะเดียวกันนั้น กระแสจะลดลงไปเรื่อย ๆ

ระยะที่ 3 Ready; no current เมื่ออัดประจุจนเต็มแล้ว กระบวนการอัดประจุหยุด

ระยะที่ 4 Standby mode เมื่ออัดประจุเรียบร้อยแล้ว และแบตเตอรี่ยังไม่ได้ถูกใช้งาน แรงดันจะลดลง ระยะนี้จึงมีไว้เพื่ออัดประจุกลับไปให้เต็มอีกครั้ง



รูปที่ 2.6 ขั้นตอนการอัดประจุของแบตเตอรี่ชนิดลิเทียมไอออน

ระบบการบริหารจัดการแบตเตอรี่จะทำหน้าที่ควบคุมการอัดประจุตามขั้นตอนดังกล่าวข้างต้นและจะแสดงผลสถานะของแบตเตอรี่ (Cell monitoring) โดยอ่านค่าแรงดันของแต่ละเซลล์และอ่านค่าอุณหภูมิของแบตเตอรี่ก่อนที่จะส่งผ่านข้อมูลดังกล่าวไปยังบอร์ดไมโครคอนโทรลเลอร์เพื่อแสดงผลให้ผู้ใช้งานต่อไป สำหรับขั้นตอนการปรับสมดุลของแต่ละเซลล์ จะเริ่มจากบอร์ดไมโครคอนโทรลเลอร์ทำการตรวจสอบค่าแรงดันของแบตเตอรี่แต่ละเซลล์และเมื่อต้องการที่จะปรับความสมดุลของแต่ละเซลล์ BMS จะตรวจสอบว่าเซลล์ใดที่มีค่าแรงดันมากกว่าเซลล์อื่น และเซลล์นั้นจะถูกปรับลดแรงดันลงให้มีค่าแรงดันเท่ากับเซลล์อื่นๆ และสำหรับหน้าที่การป้องกันความเสียหายของเซลล์แบตเตอรี่ (Cell protecting) ระบบบริหารจัดการแบตเตอรี่จะอ่านค่าแรงดันของแบตเตอรี่ ซึ่งเมื่อค่าแรงดันของแบตเตอรี่ที่อ่านได้มีค่าต่ำเกินกว่ากำหนด หรืออุณหภูมิที่วัดได้สูงเกินกว่ากำหนด โดยจะมีสัญญาณเตือนดังขึ้น และหากในขณะที่อัดประจุไฟฟ้า ถ้าแบตเตอรี่เกิดการอัดประจุมากเกินไป (Overcharge) ระบบบริหารจัดการแบตเตอรี่จะส่งสัญญาณไปที่สวิตช์รีเลย์เพื่อตัดวงจรการอัดประจุให้หยุดทำงาน โดยระบบบริหารจัดการแบตเตอรี่มีความสำคัญต่อการใช้งานยานยนต์ไฟฟ้าอย่างมีประสิทธิภาพและปลอดภัย เนื่องจากระบบดังกล่าวจะทำหน้าที่ตรวจสอบสถานะของเซลล์แบตเตอรี่การปรับความสมดุลของเซลล์แต่ละเซลล์ และตัดการทำงานของแบตเตอรี่หากตรวจพบการทำงานที่ผิดปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ชุดควบคุม

เป็นอุปกรณ์นำข้อมูลเข้ารูปแบบหนึ่งของคอมพิวเตอร์ มีลักษณะเป็นคันโยกบนฐานใช้ควบคุมการเคลื่อนที่ของตัวชี้ หรือ pointer บนจอภาพ

2.4.1 ก้านควบคุม (Joy-Stick)

โปรแกรมบางประเภท การใช้แป้นพิมพ์หรือเมาส์อาจไม่เหมาะสม เช่น โปรแกรมประเภทเกม ด้วยเหตุนี้เอง จึงได้มีการผลิตก้านควบคุม หรือ joystick ขึ้นมา ซึ่งสามารถใช้งานได้ง่ายขึ้นและทำให้การเล่นเกมน่าสนใจมากขึ้น ถ้าคุณกำลังเล่นเกมขับรถบินจำลอง หรือคุณต้องการให้เครื่องบินไปทางซ้ายคุณก็โยกคันบังคับไปทางซ้ายได้ตลอดเวลา เกมคอนโทรลเลอร์มีการพัฒนาอย่างต่อเนื่อง โดยเริ่มแรก เป็นเพียงกล่องเล็กๆ มีปุ่มบังคับเพียงไม่กี่ปุ่ม จนในปัจจุบันมีการพัฒนาเกมคอนโทรลเลอร์ที่เฉพาะเจาะจงกับเกมบางประเภทมากขึ้น เช่น พวงมาลัยสำหรับเกมขับรถ เป็นต้น ก้านควบคุมบางอันอาจมีปุ่มเพิ่มเติมเพื่อใช้สั่งงานอื่น ๆ



รูปที่ 2.7 ก้านควบคุม (Joy-Stick)

2.4.1.1 หลักการทำงานของก้านควบคุม

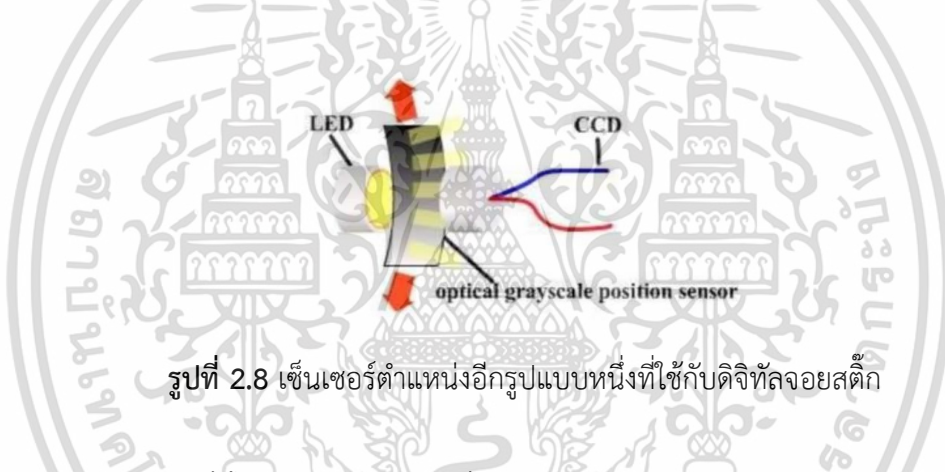
ก้านควบคุมหรือจอยสติ๊ก ออกแบบมาเพื่อบอกให้คอมพิวเตอร์รับคำสั่งจัดการตำแหน่งของวัตถุในเวลาหนึ่งๆ ได้อย่างไร วิธีการก็คือ จอยสติ๊กจะส่งตำแหน่งพิกัดในแนวราบหรือในแนวตั้งหรือตำแหน่งพิกัดใน แกน X-Y ไปยังคอมพิวเตอร์เพื่อประมวลผลต่อไปยังโปรแกรมที่กำลังใช้อยู่ด้านล่างของคันบังคับเป็นให้เป็นอิสระ แกนคอนโทรลเลอร์อื่นๆ เช่น พวงมาลัย หรือ เกมแพดอาจจะมีลักษณะแตกต่างจากจอยสติ๊ก แต่หลักการกำเนิดสัญญาณจะคล้ายๆ กันกับจอยสติ๊ก สวิตช์ หรือปุ่มต่างๆ ที่ใช้สำหรับบังคับ เมื่อมีการกดปุ่ม ก็จะมีการส่งสัญญาณไปยังการ์ดอะแดปเตอร์เพื่อสร้างข้อมูล เช่น ถ้ากดปุ่ม ก็จะเปลี่ยนค่าข้อมูลให้เป็น 1 ถ้าไม่ได้กด ก็จะมีค่าเป็น 0 เป็นต้น การสร้างข้อมูลเหล่านี้ขึ้นอยู่กับแต่ละเกม

ตัวเซนเซอร์ตรวจจับตำแหน่งจะเชื่อมติดกับแต่ละแกนของจอยสติ๊กเพื่อตอบสนองกับพิกัด X-Y และส่งสัญญาณไปที่การ์ดอะแดปเตอร์ของเกม ซึ่งมีซอฟต์แวร์ที่จะนำข้อมูลสัญญาณที่ได้มาเปลี่ยนให้เป็นตำแหน่งของเกมคอนโทรลเลอร์ตัวเซนเซอร์ในจอยสติ๊ก ส่วนใหญ่จะทำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวเก็บประจุ และ potentiometer หรือ POT ซึ่งประกอบด้วยตัวต้านทานแบบปรับค่าได้ซึ่งควบคุมด้วยการเคลื่อนที่ทั้งสองทิศทางของจอยสติ๊ก กระแสไฟฟ้าจะไหลผ่านจาก POT ไปยังตัวเก็บประจุ เมื่อประจุเพิ่มขึ้นจนเกิดศักย์ไฟฟ้าถึง 5 โวลต์ ตัวเก็บประจุก็จะคายประจุออก

เมื่อจอยสติ๊กถูกดันไปในทิศทางใดทิศทางหนึ่ง ความต้านทานจะเพิ่มขึ้น ทำให้ตัวเก็บประจุใช้เวลาเพิ่มขึ้นในการชาร์จประจุเข้า และคายประจุออก เมื่อถูกดันไปในทิศทางอื่น ค่าความต้านทานจะลดลงกระแสไฟฟ้าจะไหลไปที่ตัวเก็บประจุได้มากขึ้น ทำให้การชาร์จและคายประจุทำได้รวดเร็วขึ้น อะแดปเตอร์จะจับเวลาในหน่วยมิลลิวินาทีสำหรับการชาร์จและคายประจุ จากนั้นจะคำนวณตำแหน่งของจอยสติ๊กทั้งสองแกน

Optical grayscale position sensor จะใช้ LED (Light-Emitting Diode) และ CCD (Charge Coupled Device) ที่เปลี่ยนแสงจาก LED เป็นกระแสไฟฟ้า ระหว่าง LED กับ CCD เป็นฟิล์มที่มีการไล่สีจากสว่างไปมืด จากปลายด้านหนึ่ง เมื่อจอยสติ๊กมีการเคลื่อนไหวฟิล์ม จะเคลื่อนไหวด้วยปริมาณของแสงที่ผ่านฟิล์มโดยจะเปลี่ยนแปลงไป ซึ่งสามารถตรวจจับได้ด้วย CCD



รูปที่ 2.8 เซ็นเซอร์ตำแหน่งอีกรูปแบบหนึ่งที่ใช้กับดิจิทัลจอยสติ๊ก

2.4.2 บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano

Arduino Nano เป็นบอร์ดขนาดเล็กที่เข้ากันได้กับแผ่นที่ซัดซึ่งเข้ากันได้กับ AT Mega328 มีประโยชน์เทียบเท่ากับ Arduino Nano แต่เมื่อมาถึงแพ็คเกจ DIP โมดูลจะทำงานร่วมกับการเชื่อมต่อ Mini-B USB บอร์ดโคลน Arduino นี้สามารถทำงานร่วมกับ Arduino IDE ได้ดีในกรณีต่างๆ ซอฟต์แวร์ Arduino (IDE) ใช้สำหรับโปรแกรม Arduino Nano ซอฟต์แวร์ Arduino เป็นสภาพแวดล้อมการพัฒนาในรูปแบบบูรณาการที่ใช้ร่วมกันกับบอร์ด Arduino ทั้งหมดนี้จะทำงานทั้งแบบออนไลน์และออฟไลน์ ข้อกำหนดรายละเอียดของบอร์ด Arduino Nano มีดังต่อไปนี้

1. เป็นบอร์ดไมโครคอนโทรลเลอร์มีขนาดเล็ก ยืดหยุ่นและใช้งานง่าย
2. ขึ้นอยู่กับไมโครคอนโทรลเลอร์ Atmel AT mega 328p หรือ MCU ในเวอร์ชัน 3.x และใน AT mega 168 ในเวอร์ชันก่อนหน้า ไม่ว่าจะในกรณีใดก็ตาม จะทำงานที่ความถี่ 16 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หน่วยความจำประกอบด้วยแฟลช 16 KB หรือ 32 KB ขึ้นอยู่กับเวอร์ชัน (2KB ที่ใช้สำหรับ bootloader โดยมีหน่วยความจำ SRAM 1 หรือ 2 KB และ EEPROM 512 ไบต์หรือ 1 KB ขึ้นอยู่กับ MCU)
4. มีแรงดันไฟฟ้า 5v แต่แรงดันไฟฟ้าขาเข้าอาจแตกต่างกันไปตั้งแต่ 7 ถึง 12v
5. มีพินดิจิทัล 14 พินอนาล็อก 8 พิน 2 พินรีเซ็ตและ 6 พินเพาเวอร์ (VCC และ GND) สำหรับพินอนาล็อกและดิจิทัลพวกเขาได้รับการกำหนดฟังก์ชันพิเศษหลายอย่าง เช่น pinMode () และ digitalWrite () และ analogRead () สำหรับแอนะล็อก ในกรณีของแอนะล็อกพวกเขาอนุญาตให้มีความละเอียด 10 บิต ตั้งแต่ 0 ถึง 5v บินดิจิทัล 22 สามารถใช้เป็นเอาต์พุตได้ PWM.
6. ไม่รวมซ็อกเก็ตไฟฟ้ากระแสตรง
7. ใช้ mini USB มาตรฐานสำหรับการเชื่อมต่อกับคอมพิวเตอร์เพื่อตั้งโปรแกรมหรือเปิดเครื่อง
8. การใช้พลังงานคือ 19 mA
9. PCB ขนาด 18x45 มม. น้ำหนักเพียง 7 กรัม



รูปที่ 2.9 บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano

2.4.2.1 ภาคจ่ายไฟ

บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano สามารถใช้พลังงานไฟฟ้าจากพอร์ต USB ของคอมพิวเตอร์ หรือแหล่งพลังงานไฟฟ้าจากภายนอกโดยบอร์ดสามารถเลือกแหล่งจ่ายไฟฟ้าได้โดยอัตโนมัติ ในส่วนของแหล่งพลังงานไฟฟ้าจากภายนอก สามารถใช้ได้ทั้งแบบไฟฟ้ากระแสสลับและไฟฟ้ากระแสตรงจากอะแดปเตอร์ หรือจากแบตเตอรี่โดยมีขั้วไฟฟ้าของอะแดปเตอร์สามารถเชื่อมต่อด้วยการเสียบปลั๊กขนาด 2.1 มม. เข้ากับแจ็คพาวเวอร์ของบอร์ด

2.4.2.2 หน่วยความจำ

ไมโครคอนโทรลเลอร์ AT mega 328 มีหน่วยความจำแบบแฟลชสำหรับการจัดเก็บโปรแกรม ขนาด 16 หรือ 32 กิโลไบต์ (มีหน่วยความจำใช้สำหรับการบูต ขนาด 0.5 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิโลไบต์) มีหน่วยความจำชั่วคราวแบบสแตติกแรม (SRAM) ขนาด 2 กิโลไบต์ และมีหน่วยความจำถาวรแบบอีพรีอิม(EEPROM) ขนาด 0.5 กิโลไบต์

2.4.2.3 พอร์ตอินพุต - เอาต์พุต

บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano มีพอร์ตดิจิทัลทั้งหมด 14 ขา สามารถเป็นได้ทั้งพอร์ตอินพุตหรือเอาต์พุต (แต่ละขาทำงานที่แรงดัน 5 โวลต์ สามารถจ่ายหรือรับกระแสไฟฟ้าได้สูงสุด 40 มิลลิแอมป์ และมียังมีหน้าที่พิเศษ ดังนี้ ตัวต้านทานต่อแบบพูลอัพอยู่ภายใน มีค่าความต้านทาน 20 - 50 กิโลโห์ม นอกจากนี้แล้ว บางพอร์ต

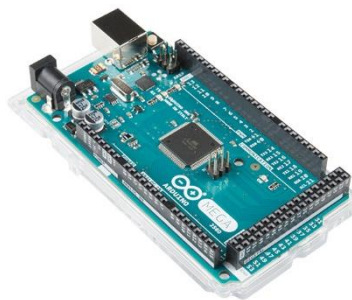
2.4.2.4 การสื่อสาร

บอร์ดไมโครคอนโทรลเลอร์ Arduino Nano มีพอร์ตสื่อสารเพื่อเชื่อมต่อกับคอมพิวเตอร์ หรือบอร์ด Arduino อื่นๆ หรือไมโครคอนโทรลเลอร์เบอร์ อื่น ๆ หลายรูปแบบ ตามความสามารถของไมโครคอนโทรลเลอร์ AT mega 328 ที่มีพอร์ตสื่อสารอนุกรมแบบ UART ที่พอร์ตดิจิทัล ขา 0 (R และพอร์ตดิจิทัล ขา 1 (TX) ช่องทางการสื่อสารแบบอนุกรมยังเชื่อมโยงผ่านพอร์ต USB และยังสามารถเป็นพอร์ต COM เสมือนซอฟต์แวร์บนเครื่องคอมพิวเตอร์ ด้วยซอฟต์แวร์ 8U2 คอมพิวเตอร์ สามารถเชื่อมต่อได้โดยใช้ไดร์เวอร์ USB มาตรฐาน และไม่ต้องใช้ไดร์เวอร์ภายนอกแต่อย่างใด ซอฟต์แวร์ Arduino แสดงผลการสื่อสารผ่านพอร์ตแบบอนุกรมทางพอร์ต USB ระหว่างบอร์ด Arduino กับคอมพิวเตอร์ผ่านขา RX และ TX ซึ่งทำให้ไฟ LED กะพริบเมื่อข้อมูลถูกส่งผ่านทางพอร์ต USB ประจุให้หยุดทำงาน โดยสรุป ระบบบริหารจัดการแบตเตอรี่มีความสำคัญต่อการใช้งานยานยนต์ไฟฟ้าอย่างมีประสิทธิภาพ และปลอดภัย เนื่องจากระบบดังกล่าวจะทำหน้าที่ตรวจสอบสถานะของเซลล์แบตเตอรี่ ปรับความสมดุลของเซลล์แต่ละเซลล์ และตัดการทำงานของแบตเตอรี่หากตรวจพบการทำงานที่ผิดปกติ

2.4.3 บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560

บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีขนาดใหญ่ที่สุดในกลุ่มของบอร์ดไมโครคอนโทรลเลอร์ Arduino ซึ่งเป็นบอร์ดที่มีการพัฒนามาจากบอร์ดไมโครคอนโทรลเลอร์ Arduino Uno R3 ที่ใช้ชิป Atmega2560 จึงทำให้มีขาที่เป็น Digital Input/Output ถึง 54 ขา ซึ่งเป็นขาที่ใช้เป็น Output แบบ PWM 14 ขา และมีขา Analog Inputs จำนวน 16 ขา มีขาการสื่อสารแบบ UARTs (Hardware Serial Port) จำนวน 4 ขา มีย่านการทำงานอยู่ที่ 16 MHz สามารถเชื่อมต่อการทำงานกับคอมพิวเตอร์ด้วยสายเคเบิลแบบ USB Type B to USB Type A เพื่อทำการอัปโหลดโปรแกรมการทำงานที่ต้องการลงไป หรือใช้สามารถใช้ Adapter AC-to-DC เพื่อให้บอร์ดสามารถทำงานได้ และยังมีปุ่ม Reset ที่สามารถเชื่อมต่อกับ Shields ที่ออกแบบเพื่อใช้งานกับพวก Arduino Duemilanove หรือ Diecimila

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 บอร์ดไมโครคอนโทรลเลอร์ Arduino Mega 2560

2.4.3.1 ขาหรือ Pin ทั่วไป

1. ขา Vin ทำหน้าที่เป็น Input Voltage หรือเป็นขาที่ทำหน้าที่รับแหล่งจ่ายจากภายนอกเพื่อจ่ายให้บอร์ด Arduino Mega 2560 ทำงาน โดยที่ย่านของแรงดันที่รองรับการทำงานจะอยู่ที่ 7-12 V
2. ขา 5V จะทำหน้าที่เป็นขา Output Voltage หรือเป็นขาที่ทำหน้าที่ในการปล่อยแรงดัน เพื่อที่ใช้สามารถไปเป็นแหล่งจ่ายให้กับอุปกรณ์หรือเซนเซอร์ที่ต่อทำงานร่วมกับบอร์ดได้
3. ขา 3.3V จะเป็นขา Volt Supply โดยที่แรงดัน 3.3V นี้จะถูกสร้างขึ้นมาจาก Regulator ที่อยู่บนบอร์ด โดยจะทำการแปลงไฟจากแหล่งจ่ายที่ได้มาให้เหลือเพียง 3.3V และให้กระแสได้สูงสุดถึง 50 mA
4. ขา GND จะทำหน้าที่เป็นขา Ground ของบอร์ด
5. ขา IOREF เป็นขาที่ทำหน้าที่เป็น Voltage Reference ให้กับไมโครคอนโทรลเลอร์ เพื่อเลือกค่าแรงดันการทำงานให้กับ Shield ที่มาเชื่อมต่อการทำงานกับบอร์ด

2.4.3.2 หน่วยความจำ

ใช้ชิป ATmega2560 ซึ่งเป็นชิปตัวเดียวที่อยู่บนบอร์ด Arduino Uno R3 โดยมีหน่วยความจำอยู่ที่ 256KB (8KB ใช้สำหรับการ Bootloader) และยังมีหน่วยความจำรวม 8KB สำหรับ SRAM และ 4KB สำหรับ EEPROM

2.4.3.3 พอร์ตอินพุต - เอาต์พุต

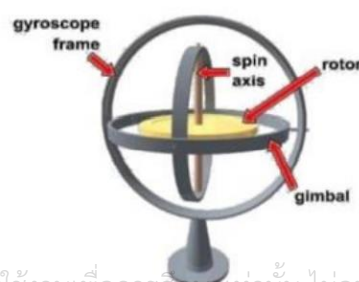
ขาของบอร์ด Arduino Mega 2560 ที่เป็น Digital Pins ทั้ง 54 Pins นั้นสามารถทำงานให้เป็นได้ทั้ง Input และ Output โดยที่แต่ละขาจะทำงานที่แรงดัน 5V และให้กระแสสูงสุดถึง 40 mA

2.4.3.4 พิงก์ชันการทำงานอื่น ๆ

1. ขาที่ทำหน้าที่เป็น External Interrupts ได้แก่ ขา 2 (Interrupt 0), ขา 3 (Interrupt 1), ขา 18 (Interrupt 5), ขา 19 (Interrupt 4), ขา 20 (Interrupt 3), ขา 18 (Interrupt 2) โดยขาการทำงานเหล่านี้สามารถทำการกำหนดการทำให้เป็นแบบ Interrupt จากการเขียนโปรแกรมการทำงาน
2. ขาที่สามารถให้ Output PWM ขนาด 8-bits ได้แก่ ขา 2 ถึง ขา 13 และขา 44 ถึง 46
3. ขาที่รองรับการสื่อสารแบบ SPI (Serial Peripheral Interface) หรือการสื่อสารข้อมูลแบบอนุกรมแบบซิงโครนัส ได้แก่ ขา 50 (MISO), ขา 51 (MOSI), ขา 52 (SCK) และขา 53 (SS) ซึ่งเป็นขาที่รองรับการสื่อสารแต่จะไม่เกี่ยวข้องกันกับ ICSP Header
4. ขาที่เป็น LED ได้แก่ ขา 13 ซึ่งเป็นขาที่เชื่อมต่อกับ Build-In LED โดยสามารถสั่งการทำงานจากโปรแกรมถ้าหากสั่ง HIGH LED ก็จะมีติด แต่ถ้าสั่งให้ LOW LED ก็จะมีดับ
5. ขาที่รองรับการเชื่อมต่อการทำงานแบบ I2C ได้แก่ ขา 20 (SDA) และขา 21 (SCL)
6. บอร์ด Arduino Mega 2560 โดยมี Pins ที่เป็น Analog Inputs จำนวน 16 Pins โดยแต่ละ Pins ให้ความละเอียดอยู่ที่ 10-bits
7. บอร์ด Arduino Mega 2560 มี AREF เป็นแรงดันอ้างอิงสำหรับ Analog Input
8. ปุ่ม Reset จะใช้ในการ Reset การทำงานของไมโครคอนโทรลเลอร์ โดยทั่วไปจะใช้การเพิ่มปุ่ม Reset ไว้บน Shield เพื่อป้องกันปุ่มที่อยู่บนบอร์ด

2.4.4 ไจโรสโคป

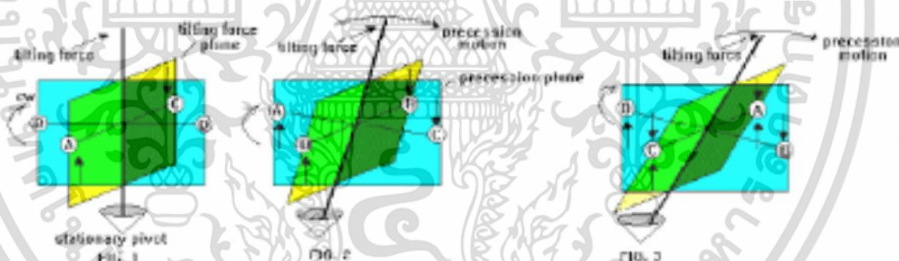
ไจโรสโคป (Gyroscope) เป็นอุปกรณ์ชนิดหนึ่งที่ใช้แรงเฉื่อยของล้อเพื่อหมุน และรักษาระดับทิศทางของแกน ประกอบด้วยการหมุนของล้อ ความเร็วในการหมุนถูกบันทึกไว้ในกรอบอีก ทำให้เอียงในทิศทางต่าง ๆ ได้โดยอิสระ นั่นคือหมุนในแกนใด ๆ ก็ได้ โมเมนตัมเชิงมุมของล้อดังกล่าวทำให้มันคงรักษาตำแหน่งของมันไว้แม้กรอบล้อจะเอียง จากคุณสมบัติดังกล่าวทำให้สามารถนำหลักการนี้ไปประยุกต์ใช้เพื่อประโยชน์ต่าง ๆ มากมาย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลง **รูปที่ 2.11** ไจโรสโคปถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.1 หลักการทำงานของไจโรสโคป

จากรูปที่ 2.12 เราจะแทนที่ขอบด้วย A , B , C , D ทั้งแถบ ในการดูว่า Gyroscope ทำงานอย่างไร จุดล่างสุดเป็นแกนคงที่แต่สามารถหมุนได้รอบทิศทาง เมื่อแรงกระทำ (fitting force) ที่ส่วนบนของแกน Gyroscope จุด A จะเคลื่อนที่ ขึ้นตามแนวตั้ง จุด C เลื่อนลงตามแนวนอนพร้อมกัน A และ B หมุนไป 90 องศา เช่นเดียวกับที่เกิดกับ C และ D โดยที่ A นั้นยังคงเลื่อนขึ้นในตำแหน่ง 90 องศา และ C เลื่อนลง ผลของการเลื่อนของ A และ C ทำให้แกนของ Gyro หมุนตามการกระทำของ precession plane เรียกการเกิดขึ้นของลักษณะนี้ว่า precession แกนของ Gyro จะหมุนไปทางมุมขวาเนื่องจากการหมุน ถ้า Gyro ถูกทำให้หมุนทวนเข็มนาฬิกา มันก็จะไปทางมุมซ้ายหมายความว่าแรงกระทำตอนต้นเป็นการตั้ง เมื่อ Gyro หมุนไปอีก 90 องศา ตาม Fig.3 จุด C จะมาแทนจุด A ในจุดที่แรงกระทำไปแล้วครั้งแรก การเคลื่อนที่ลงของจุด C จะถูกต้านโดย fitting force ทำให้แกนของ Gyro ไม่เปลี่ยนแปลง ยังมีแรง tilting กระทำมากขึ้นแกนของ Gyro ก็จะมีแรงติดกลับมาเมื่อขอบของ precession plane อยู่ที่ 180 องศา จากข้างต้นทำให้ทราบว่า การหมุนของแกน Gyro เนื่องจากแรงจุด A และ C เคลื่อนที่ขึ้นลง ฉะนั้นเมื่อหมุน Gyro ในทิศทางตรงกันข้ามกับข้างต้นจะเกิดแรงเคลื่อนที่ขึ้นลงมากขึ้น บางครั้งการเกิดขึ้นของ precession เป็นสิ่งที่ไม่ต้องการ จึงมักมีการสร้าง Gyro แบบมีแกนที่เรียกว่า Gimbaled Gyroscope ดังรูปที่ 2.11 ซึ่งเป็น Gyro พื้นฐานถูกติดตั้งไว้ในระนาบที่ตั้งฉากกับแนวแรง



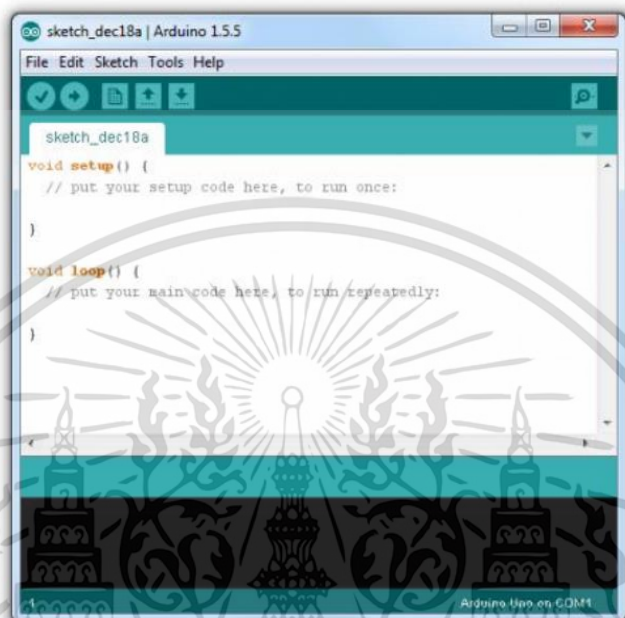
รูปที่ 2.12 หลักการทำงานของไจโรสโคป

2.5 โปรแกรม Arduino

สื่อเรียนรู้นี้ เราเลือกใช้โปรแกรม Sketch ของ Arduino (Arduino IDE) ในการเขียนโปรแกรมควบคุมการทำงานของบอร์ด Arduino Board เพราะเป็นโปรแกรมที่ใช้งานได้ง่าย เขียนด้วยภาษา C และความเป็น Open Source ทำให้ใช้งานได้โดยไม่มีค่าใช้จ่าย ได้รับความนิยมสูง จึงทำให้มีแหล่งข้อมูลให้ศึกษาค้นคว้าเพิ่มเติมในเว็บบอร์ดหรือเว็บไซต์ในอินเทอร์เน็ตอีกมากมาย และในส่วนของบอร์ด Arduino Board เองนั้น เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีขาพอร์ตอินพุตและเอาต์พุตที่มากพอในการนำไปใช้งานจริงสามารถต่อกับเซนเซอร์ได้ทั้งแบบดิจิตอลและแอนาล็อกและยังต่อเพื่อขับอุปกรณ์เอาต์พุตให้ทำงานโดยที่เราจะต้องเขียนโปรแกรมเพื่อสั่งงานให้บอร์ด

Board สามารถควบคุมอุปกรณ์ต่าง ๆ เช่น ควบคุมการเปิดปิดหลอดไฟ ปลั๊กไฟฟ้า หรือเครื่องรดน้ำ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่เรียนด้านการศึกษา ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้นไม้ เป็นต้น นอกจากนี้เพื่อให้เกิดการประยุกต์ใช้งานแบบ IoT เราสามารถควบคุมบอร์ดผ่านทางอินเทอร์เน็ตได้โดยต่อบอร์ดเสริม Arduino Ethernet Shield เข้ากับบอร์ดหลักเนื่องจากบอร์ด Arduino Board มีราคาไม่สูงมาก จึงเป็นการช่วยประหยัดต้นทุนวัสดุอุปกรณ์ หากผู้ศึกษาต้องการนำบอร์ดไปพัฒนาต่อให้เป็นอุปกรณ์หรือแอปพลิเคชัน IoT



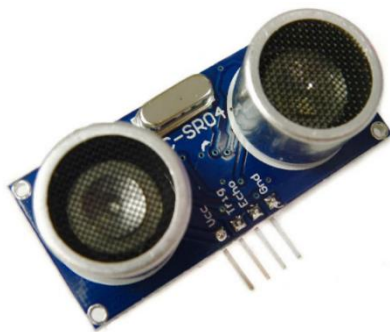
รูปที่ 2.13 โปรแกรม Arduino

2.6 Ultrasonic Module HC-SR04

โมดูลอัลตราโซนิกเป็นอุปกรณ์ที่ใช้สำหรับวัดระยะทางโดยไม่มีการสัมผัสกับตำแหน่งที่ต้องการวัด วัดได้ตั้งแต่ 2 cm ถึง 400 cm โดยส่งสัญญาณการส่งของอัลตราโซนิกนั้น มีความถี่ 40 kHz เพื่อนำไปวัดวัตถุที่ต้องการและสามารถรับสัญญาณที่สะท้อนกลับมาได้ พร้อมทั้งจับเวลาเพื่อนำมาใช้ในการคำนวณระยะทาง

หลักการทำงานของโมดูล HC-SR04 เป็นอุปกรณ์อิเล็กทรอนิกส์ สำหรับวัดระยะทางด้วยคลื่นอัลตราโซนิก (ใช้คลื่นเสียงความถี่ ประมาณ 40kHz) มีสองส่วนหลักคือ ตัวส่งคลื่นที่ทำหน้าที่สร้างคลื่นเสียงออกไปในการวัดระยะแต่ละครั้ง ("Ping") แล้วเมื่อไปกระทบวัตถุหรือสิ่งกีดขวาง คลื่นเสียงถูกสะท้อนกลับมายังตัวรับแล้วประมวลผลด้วยวงจรอิเล็กทรอนิกส์ภายในโมดูล ถ้าจับเวลาในการเดินทางของคลื่นเสียงในทิศทางไปและกลับ และถ้าทราบความเร็วเสียงในอากาศ ก็จะสามารถคำนวณระยะห่างจากวัตถุกีดขวางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 Ultrasonic Module HC-SR04

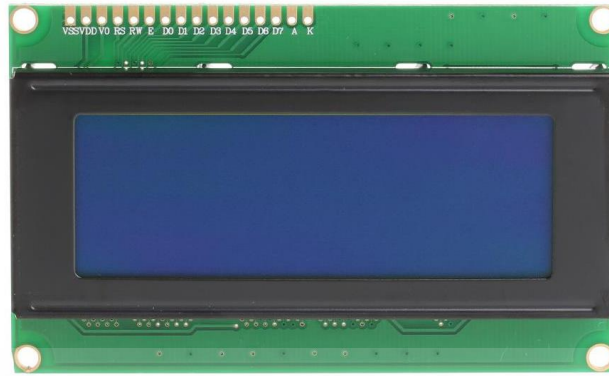
2.7 LCD Display

จอ Liquid Crystal Display (LCD) เป็นจอแสดงผลที่มีรูปแบบที่นิยมนำมาใช้งานกันอย่างแพร่หลาย กับระบบสมองกลฝัง ซึ่งตัวอย่างของการแพร่หลาย บนจอ LCD นั้น มีแบบที่แสดงผลเป็นตัวอักษรเรียกว่า หรือเรียกว่า Character LCD ซึ่งมีการกำหนดตัวอักษรหรือตัวอักษรที่ใช้ในการแสดงผลไว้ และแบบที่สามารถแสดงผลเป็นรูปภาพโดยเรียกการใช้งานนี้ว่า Graphic LCD นอกจากนี้ชนิดที่เป็นจอ ที่มีการผลิตขึ้นมาใช้สำหรับเฉพาะงาน ทำให้มีรูปแบบและรูปร่างเฉพาะเจาะจงในการแสดงผล เช่น นาฬิกาดิจิตอล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น

โครงสร้างของ LCD ทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่ โดยเว้นช่องว่างตรงกลางไว้ 6-10 ไมโครเมตร ผิวด้านในของแผ่นแก้วจะเคลือบด้วยตัวนำไฟฟ้าแบบใสเพื่อใช้แสดงตัวอักษร ตรงกลางระหว่างตัวนำไฟฟ้าแบบใสกับผลึกเหลวจะมีชั้นของสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางที่แสงส่องมากระทบเรียกว่า Alignment Layer และผลึกเหลวโดยทั่วไปแล้วจะเป็นในลักษณะ Magnetic โดย LCD ซึ่งสามารถแสดงผลให้เราสามารถมองเห็นได้ทั้งหมด 3 แบบด้วยกันคือ

- แบบแรกคือการใช้การสะท้อนแสง (Reflective Mode) LCD แบบนี้ใช้สารที่เป็นประเภทของโลหะที่เคลือบอยู่บนแผ่นหลังของ LCD
- แบบที่สองคือการใช้การส่องผ่าน (Transitive Mode) LCD แบบนี้จะวางหลอดไฟไว้ด้านหลังของจอ เพื่อเป็นการแสดงค่าที่ชัดเจน
- แบบที่สามคือแบบที่ส่องผ่านและสามารถสะท้อนกลับ (Translative Mode) LCD แบบนี้เป็นการนำเอาข้อดีของจอที่แสดงผล LCD ทั้ง 2 แบบมารวมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 LCD Display

2.8 Micro servo motor

เซอร์โวมอเตอร์ (Servo Motor) เป็นการรวมมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) เข้ากับ วงจรควบคุม โดยความแตกต่างที่สำคัญของเซอร์โวมอเตอร์กับมอเตอร์แบบอื่น ๆ คือเซอร์โวมอเตอร์ จะรู้ตำแหน่งที่ตัวเองอยู่ และสั่งเปลี่ยนตำแหน่งโดยการเปลี่ยนองศาได้ นิยมใช้งานในเครื่องบินบังคับ เรือบังคับ โดยใช้กำหนดทิศทางของหางเสือเป็นองศา

การเลือกใช้งานเซอร์โวมอเตอร์ เซอร์โวมอเตอร์(ส่วนใหญ่)ใช้แรงดันไฟฟ้า 5V และมีองศาการหมุนที่ 0 ถึงประมาณ 200 องศา (ยกเว้นมีการตัดแปลงให้หมุน 360 องศา) นอกจากนี้ที่เซอร์โว มอเตอร์ทุกรุ่นมีเหมือนกัน ยังมีอีก 4 ข้อที่เซอร์โวมอเตอร์แต่ละรุ่นมีไม่เหมือนกัน

1. แรงบิด (Torque) - เป็นเลขบอกกำลังของเซอร์โวมอเตอร์ ยิ่งตัวเลขนี้มีค่ามาก แสดงว่า เซอร์โวมอเตอร์มีแรงมาก
2. ความเร็วในการหมุน (Speed) - เป็นตัวเลขที่บอกว่าเซอร์โวมอเตอร์สามารถเปลี่ยน ตำแหน่งได้เร็วแค่ไหน ยิ่งตัวเลขมีค่ามาก แสดงว่าเซอร์โวมอเตอร์มีความเร็วในการ เปลี่ยนตำแหน่งที่เร็วมาก
3. วัสดุที่ใช้ทำแกนหมุน - วัสดุที่ใช้ทำแกนหมุนของเซอร์โวมอเตอร์มี 2 ชนิด คือ พลาสติก และเหล็ก สำหรับแกนหมุนพลาสติกเมื่อใช้งานเซอร์โวมอเตอร์อย่างหนักเป็นเวลานาน จะทำให้เฟืองของเซอร์โวมอเตอร์ครูด ดังนั้นหากนำเซอร์โวมอเตอร์ไปใช้งานหนักเป็น เวลานาน จึงควรเลือกแกนเหล็ก เพราะแกนเหล็กมีโอกาสที่เฟืองครูดได้น้อยกว่า
4. ขนาดของแกนหมุน - เซอร์โวมอเตอร์แต่ละรุ่นจะมีขนาดของแกนหมุนที่แตกต่างกัน ตามแรงบิดของเซอร์โวมอเตอร์แต่ละรุ่น

หลักการการทำงานของเซอร์โวมอเตอร์เริ่มที่วงจรควบคุม เมื่อวงจรควบคุมได้รับข้อมูลองศา ที่ต้องการมาแล้ว วงจรควบคุมจะคำนวณว่ามอเตอร์จะต้องหมุนในทิศทางตามเข็มนาฬิกา หรือ ทวนเข็มนาฬิกา เพื่อให้ไปสู่องศาที่ต้องการได้ เมื่อมอเตอร์เริ่มหมุน ตัววอลุ่มที่ติดอยู่กับชุดเฟือง มอเตอร์จะตรวจสอบตำแหน่งที่มอเตอร์หมุนไป โดยหากวอลุ่มตรวจพบว่าตำแหน่งที่มอเตอร์หมุนเริ่ม

ใกล้กับองศาที่ผู้ใช้กำหนด วงจรส่วนควบคุมจะเริ่มสั่งให้มอเตอร์หมุนช้าลงเพื่อให้หมุนเข้าใกล้องศาที่กำหนดได้มากที่สุด เมื่อมอเตอร์หมุนได้ตำแหน่งองศาที่ต้องการแล้ว วงจรส่วนควบคุมจะตรวจสอบตำแหน่งของมอเตอร์เป็นระยะ ๆ โดยอ่านค่าจากวอลุ่ม หากตรวจพบว่าตำแหน่งผิดเพี้ยนไปจากค่าที่ตั้งไว้ (อันอาจเกิดจากผู้ใช้เอามือไปหมุนเล่น หรือภาระส่งผลให้ตำแหน่งเคลื่อน) วงจรควบคุมก็จะสั่งให้มอเตอร์หมุนกลับมาให้ได้ตำแหน่งเป็นระยะ ๆ

ชนิดของเซอร์โวมอเตอร์ แบ่งออกเป็น 2 ชนิด ดังนี้

1. เซอร์โวมอเตอร์ 180 องศาเป็นเซอร์โวมอเตอร์ที่นิยมใช้งานทั่วไป มีหลายรุ่น หลายขนาด และหลายราคา สามารถควบคุมให้หมุนได้ตามองศาที่ต้องการ โดยหมุนได้ 0 ถึง 180 องศา (ในบางรุ่นหมุนได้สุดที่ประมาณ 200 องศา)
2. เซอร์โวมอเตอร์ 360 องศาเป็นเซอร์โวมอเตอร์ที่ส่วนใหญ่ตัดแปลงมาจากแบบ 180 องศา โดยตัดแปลงวงจรควบคุม และตัดแกนหรือนำเอาวอลุ่มออก เพื่อให้เซอร์โวมอเตอร์สามารถหมุนได้ครบรอบ เซอร์โวมอเตอร์ชนิดนี้ไม่สามารถควบคุมองศาได้ ควบคุมได้แค่ความเร็ว และทิศทางการหมุนเท่านั้น นิยมนำมาใช้เป็นมอเตอร์สำหรับรถบังคับ รถวิ่งตามเส้น



รูปที่ 2.16 Micro servo motor

2.9 Encoder Omron E6B2-CWZ6C

เอ็นโค้ดเดอร์แบบแกนหมุน คือ เซ็นเซอร์ชนิดหนึ่งที่ทำหน้าที่ในการเข้ารหัส จากระยะทางจากการหมุนรอบตัวเอง และแปลงออกมาเป็นรหัสในรูปแบบของสัญญาณไฟฟ้า

โดยเราสามารถนำเอารหัสเหล่านี้มาแปลงกลับ เพื่อหาค่าต่างๆที่เราต้องการได้ ไม่ว่าจะเป็นระยะทางการหมุน องศาการเคลื่อนที่ หรือ ความเร็วรอบก็ได้ แล้วนำมาแสดงผลให้เราได้ทราบค่าผ่านหน้าจอแสดงผล เช่น ถ้าต้องการวัดระยะทาง เราจะต้องต่อเข้ากับตัวนับจำนวน เพื่อแสดงผลเป็นระยะทาง หรือ ถ้าต้องการวัดความเร็วรอบ เราจะต้องต่อเข้ากับตัววัดพัลส์ โดยการประยุกต์ใช้เอ็นโค้ดเดอร์นั้น สามารถใช้ทำงานได้อย่างหลากหลาย เช่น กระบวนการประกอบชิ้นส่วนอิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุตสาหกรรมเคมีคอนดักเตอร์ เครื่องมือวัดต่างๆ เช่น ในการวัดความยาว หรือ อุตสาหกรรมเครื่องมือแพทย์ เป็นต้น

ส่วนการแสดงผลเป็นความเร็วรอบของ RPM, RPS โดยอาศัยสัญญาณที่ผ่านการเข้ารหัสแล้วออกมาเป็นสัญญาณทางไฟฟ้านั้น สามารถแบ่งรูปแบบของการเข้ารหัสได้อีกหลากหลายรูปแบบ เช่น สัญญาณดิจิตอล ศูนย์กับหนึ่ง ธรรมดา หรือ เป็นแบบ Binary Code, BCD Code, Gray Code

ซึ่งในปัจจุบันก็มีสัญญาณที่เป็นการสื่อสารแบบอนุกรมอื่นๆ อีกมากมายที่ถูกนำมาใช้กับเอ็นโค้ดเดอร์ เช่น SSI, CAN, PROFIBUS, ETHERCAT โดยการสื่อสารแบบต่างๆเหล่านี้ ถูกออกแบบมาเพื่อเพิ่มระยะในการสื่อสารให้ไกลมากขึ้น อีกทั้งยังเป็นการลดปัญหาเรื่องการส่งสัญญาณ ที่ผิดพลาด และสามารถตอบสนองต่อยุคการผลิตแบบ Industry 4.0 หรือ Smart Factory ได้เป็นอย่างดี

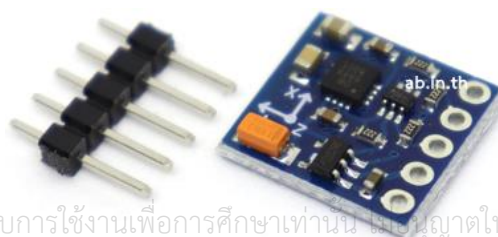
เราสามารถแบ่งประเภทของเอ็นโค้ดเดอร์ได้หลายแบบ โดยขึ้นอยู่กับว่าจะยึดเกณฑ์ใดเพื่อใช้ในการแบ่ง เช่น เราอาจแบ่งได้จากโครงสร้างการใช้งาน หรือแบ่งตามโครงสร้างของเอ็นโค้ดเดอร์ก็ได้ และนอกจากนี้เรายังสามารถใช้ประเภทเอาต์พุตของเอ็นโค้ดเดอร์มาใช้ในการแบ่งได้อีกเช่นกัน



รูปที่ 2.17 Encoder Omron E6B2-CWZ6C

2.10 GY-273 3-axis Compass Module

GY-273 คือ 3-axis Compass Module ใช้ชิป HMC5883L โมดูลวัดสนามแม่เหล็ก 3 แกน สามารถประยุกต์ใช้งานเป็นเข็มทิศได้ เชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่าน I2C ต่อใช้งานง่าย ใช้สายไฟน้อย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้นำเนื้อหาในเอกสารนี้ไปเผยแพร่ซ้ำโดยไม่ได้รับอนุญาตจากทางผู้จัดทำเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 GY-273 3-axis Compass Module

2.11 PCA9685 16-Channel 12-bit PWM Servo shield I2C interface

Module PCA9685 ขับ Servo 16 ช่อง ด้วย Arduino ความละเอียด 12 bit แบบ PWM Servo การติดต่อสื่อสารแบบ I2C Interface ใช้ IC เบอร์ PCA9685 ในการควบคุม สามารถขับ Servo ได้สูงสุดถึง 16 ตัว ด้วยการเชื่อมต่อ ด้วย 2 สาย I2C และยังสามารถใช้ Module ตัวนี้ได้หลายตัวโดยการอ้างอิงด้วยวิธีการ I2C อ้างอิง Address รองรับ Address ในช่วงระหว่าง 0x60-0x80 โดยใช้สายไฟเพียง 1 เส้น



รูปที่ 2.19 PCA9685 16-Channel 12-bit PWM Servo shield I2C interface

2.12 MicroSD Card Adapter

ตัวแปลง Micro SD เป็น SD Card แคนำ Micro SD card มาเสียบก็สามารถแปลงเป็น SD ไปใช้กับอุปกรณ์อื่น ๆ ได้ เช่น กล้องดิจิทัล กล้องวิดีโอ โดรน กล้องวงจรปิด

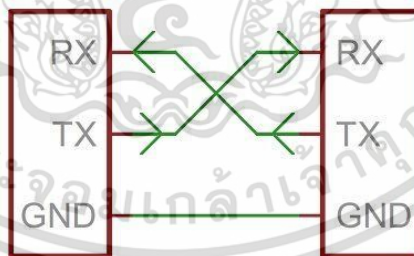


รูปที่ 2.20 MicroSD Card Adapter

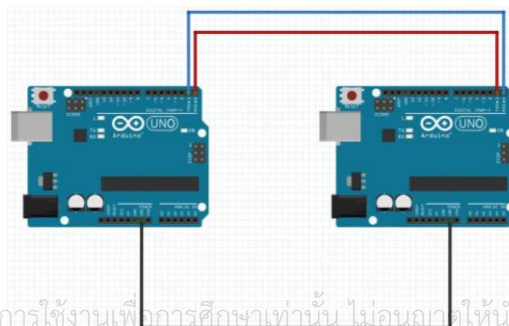
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 การสื่อสารผ่าน UART ของบอร์ดไมโครคอนโทรลเลอร์

การสื่อสารผ่าน UART หรือที่มีชื่อเต็มว่า Universal Asynchronous Receiver and Transmitter ซึ่งเป็นการสื่อสารข้อมูลผ่านพอร์ตอนุกรมแบบอะซิงโครนัสระหว่างกับอุปกรณ์ต่าง ๆ โดยจะเป็นการสื่อสารระหว่างอุปกรณ์เพียงสองชิ้นเท่านั้น ได้แก่ตัวส่ง (Transmitter) และตัวรับ (Receiver) และยังเป็นการสื่อสารข้อมูลแบบฟูลดูเพล็กซ์ (Full Duplex) หรือก็คือสามารถรับและส่งข้อมูลได้ทั้งสองทิศทางในเวลาเดียวกัน และจะเป็นการสื่อสารที่นิยมใช้เป็นการสื่อสารข้อมูลกันระหว่างไมโครคอนโทรลเลอร์ด้วยกัน โดยการทำงานแบบอะซิงโครนัสนั้นจะไม่มีสัญญาณนาฬิกาส่งออกมาจากตัวส่งหรือตัวรับ เพื่อเป็นการกำหนดจังหวะในการรับส่งข้อมูล แต่จะใช้การกำหนดความเร็วในการส่งข้อมูลแทน โดยความเร็วในการส่งข้อมูลตรงนี้จะเรียกว่า Baudrate ซึ่งมีตั้งแต่ 300 600 1200 2400 4800 9600 14400 19200 28800 38400 57600 และ 115200 ในการสื่อสารข้อมูลแบบนี้จะใช้สายสัญญาณเพียง 2 เส้น ก็คือ Tx และ Rx ในใช้เพื่อรับ - ส่งข้อมูล โดยที่เส้น Tx จะเป็นเส้นที่ใช้ในการส่งข้อมูลออกไป และเส้น Rx จะเป็นเส้นที่ใช้ในการรับข้อมูล และในการที่จะสื่อสารข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์นั้นจะต้องทำการไขว้สายการเชื่อมต่อของ Tx และ Rx หรือก็คือการที่นำสายข้อมูล Tx ของบอร์ดไมโครคอนโทรลเลอร์ตัวหนึ่งไปต่อกับช่องของ Rx ของไมโครคอนโทรลเลอร์อีกบอร์ดหนึ่ง และนำสายข้อมูล Rx ของบอร์ดไมโครคอนโทรลเลอร์ตัวที่หนึ่ง ไปเชื่อมต่อกับช่อง Tx ของไมโครคอนโทรลเลอร์อีกบอร์ดหนึ่ง และเชื่อมต่อกาวด์ของบอร์ดไมโครคอนโทรลเลอร์ทั้ง 2 บอร์ดเข้าด้วยกัน เมื่อเชื่อมต่อเสร็จแล้วก็ต้องทำการกำหนดความเร็วในการส่งข้อมูลของบอร์ดไมโครคอนโทรลเลอร์ทั้ง 2 บอร์ดให้มีความเร็วที่เท่ากัน จึงจะทำให้สามารถรับ - ส่งข้อมูลระหว่างทั้ง 2 บอร์ดได้



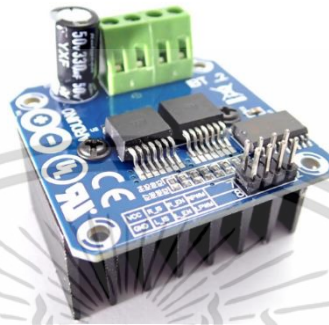
รูปที่ 2.21 การเชื่อมต่อการทำงานของการสื่อสารผ่าน UART



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ **รูปที่ 2.22** ลักษณะการเชื่อมต่อการรับ - ส่งข้อมูลระหว่างบอร์ดไมโครคอนโทรลเลอร์

2.14 บอร์ดขับมอเตอร์ BTS7960B H-Bridge Motor Driver 6-27V 43A

บอร์ดขับมอเตอร์ DC แบบ H-Bridge ตัวบอร์ดใช้ IC เบอร์ BTS7960B 2 ตัว พร้อม Heatsink ตัวไอซีรับกระแสสูงสุดได้ 43A (Peak) เหมาะกับใช้ควบคุมมอเตอร์ 12 - 24V ขนาดไม่เกิน 200W รองรับความถี่ PWM สูงสุด 25kHz เหมาะสำหรับประยุกต์ ควบคุมความเร็วมอเตอร์ด้วย แรงดัน 0 - 5V สามารถต่อร่วมใช้งานกับไมโครคอนโทรลเลอร์ได้



รูปที่ 2.23 บอร์ดขับมอเตอร์ BTS7960B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

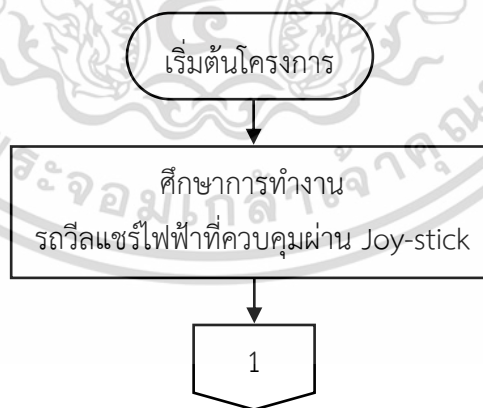
บทที่ 3

วิธีดำเนินการ

ปริญญานิพนธ์นี้ได้มีการศึกษาเรื่องของการรับ - ส่ง และบันทึกข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์ การใช้อุปกรณ์เพื่อตรวจจับสิ่งกีดขวาง มาประยุกต์ใช้ กับการที่ทำให้รถวีลแชร์ไฟฟ้าสามารถเคลื่อนที่แบบอัตโนมัติตามเส้นทางที่กำหนดได้ ซึ่งทางคณะผู้จัดทำได้เริ่มจากการศึกษาหลักการและทฤษฎีที่เกี่ยวข้องตามบทที่ 2 จากนั้นได้นำความรู้ที่ศึกษามาปรับปรุงและพัฒนาการทำงานของรถวีลแชร์ไฟฟ้าเพื่อให้สามารถเคลื่อนแบบอัตโนมัติได้ โดยได้ดำเนินการดังนี้

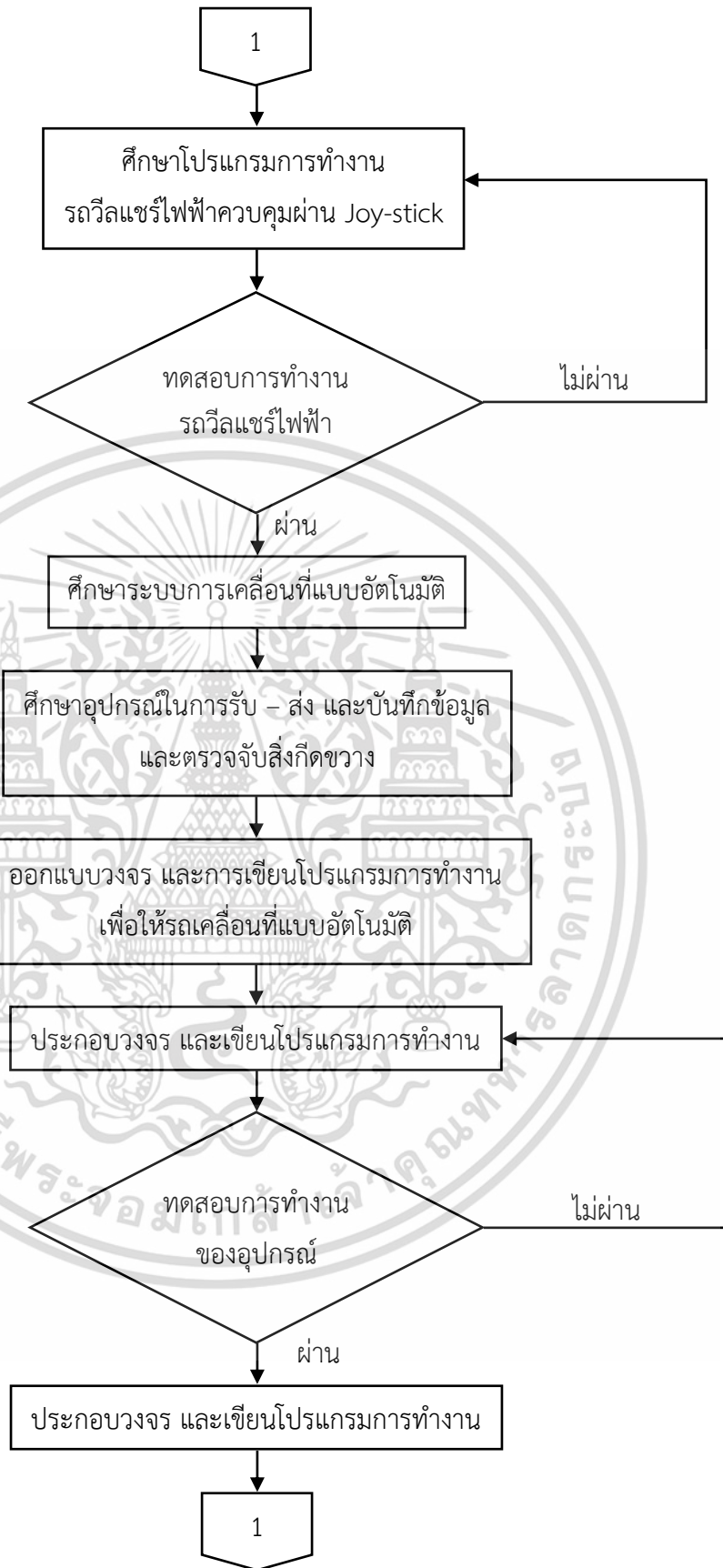
- 3.1 แผนขั้นตอนการดำเนินการ
- 3.2 ตรวจสอบรถวีลแชร์ไฟฟ้า
- 3.3 ศึกษาวงจรและโปรแกรมการทำงานของรถวีลแชร์ไฟฟ้า
- 3.4 ศึกษาระบบการเคลื่อนที่แบบอัตโนมัติ
- 3.5 ศึกษา และจัดหาอุปกรณ์ที่ใช้ในการรับ - ส่ง และบันทึกข้อมูล อุปกรณ์ในการตรวจจับสิ่งกีดขวาง
- 3.6 ออกแบบวงจร และออกแบบโปรแกรมการทำงานให้รถวีลแชร์สามารถเคลื่อนไปอย่างอัตโนมัติได้

3.1 ขั้นตอนการทำงาน



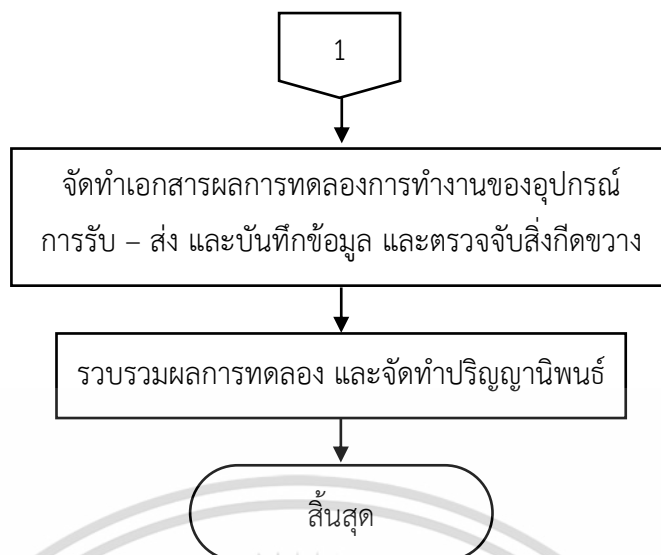
รูปที่ 3.1 แผนผังขั้นตอนการดำเนินการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนผังขั้นตอนการดำเนินการ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังขั้นตอนการดำเนินการ (ต่อ)

3.2 ตรวจสอบรถวีลแชร์ไฟฟ้า

ตรวจสอบรถวีลแชร์ไฟฟ้าที่ได้รับจากรุ่นที่วัดความขรุขระ สึกหรือตรงส่วนไหนหรือไม่ มีการเคลื่อนที่ของตัวรถตามทิศทางของการโยก Joy-Stick หรือไม่ มีลักษณะการทำงานของมอเตอร์ทำงานเท่ากันหรือไม่ และตรวจสอบแบตเตอรี่ว่าสามารถทำงานได้เต็มประสิทธิภาพได้หรือไม่

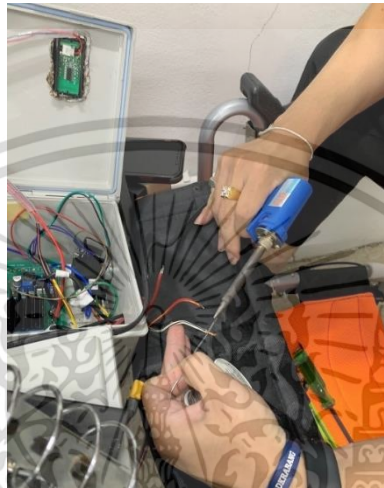


รูปที่ 3.4 ตรวจสอบการทำงานของรถวีลแชร์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ศึกษาวงจรและโปรแกรมการทำงานของรถวีลแชร์ไฟฟ้า

3.3.1 ทำการศึกษาการต่อวงจรการทำงานของรถวีลแชร์ไฟฟ้า ว่ามีลักษณะการต่อวงจรแต่ละส่วนเป็นเช่นไร สามารถแก้ไขเพิ่มเติมได้มากน้อยแค่ไหน และก็ทำการบัดกรี Socket ลงไปในวงจรตรงส่วนของบอร์ดไมโครคอนโทรลเลอร์ เพื่อให้ง่ายต่อการแก้ไขและอัปเดตโปรแกรมการทำงานของรถวีลแชร์ไฟฟ้า



รูปที่ 3.5 บัดกรีเพิ่ม Socket ลงในวงจร

3.3.2 ทำการศึกษาโปรแกรมการทำงานของรถวีลแชร์ไฟฟ้าที่ได้รับมาจากรุ่นพี่ ว่ามีการเขียนโปรแกรมการทำงานอย่างไร มีการกำหนดเงื่อนไขการทำงานเป็นเช่นไร เพื่อนำโปรแกรมการทำงานไปปรับและแก้ไขการทำงานให้สามารถมีการเคลื่อนที่แบบอัตโนมัติได้

```

Arduino IDE - Step 1: Arduino 1.8.19
File Edit Shell Debug Tools Window Help
00:00:00
Serial Plotter
Serial Project
1 #include <Wire.h>
2 // #include <Servo.h>
3 // #include <Stepper.h>
4 static int state = 0;
5 static int iState = 0;
6 static char Buffer[10];
7 static int X;
8 unsigned long timeSec = 1000; // 1 second
9 unsigned long last_time = 0;
10 unsigned char F_motor_1 , F_motor_2 , M_motor_1 , M_motor_2;
11 int gyro_X , gyro_Y , gyro_Max;
12 int serial_motor_1 , serial_motor_2 , serial_motor_1 , serial_motor_2;
13 unsigned int Vb_speed, Hsp_Speed, Cout_Speed;
14 int Speed;
15 int mode;
16 bool state = false;
17 bool autoMode = false;
18 #define PM1_M_motor_1 9
19 #define PM1_M_motor_2 11
20 #define PM2_M_motor_1 10
21 #define PM2_M_motor_2 8
22 #define Enable_motor_1 5
23 #define Enable_motor_2 6
24 #define Pot_light 8
25 #define Hsp_light 7
26 #define Left_light 4
27 #define Right_light 12

```

รูปที่ 3.6 โปรแกรมการทำงานของรถวีลแชร์ไฟฟ้า

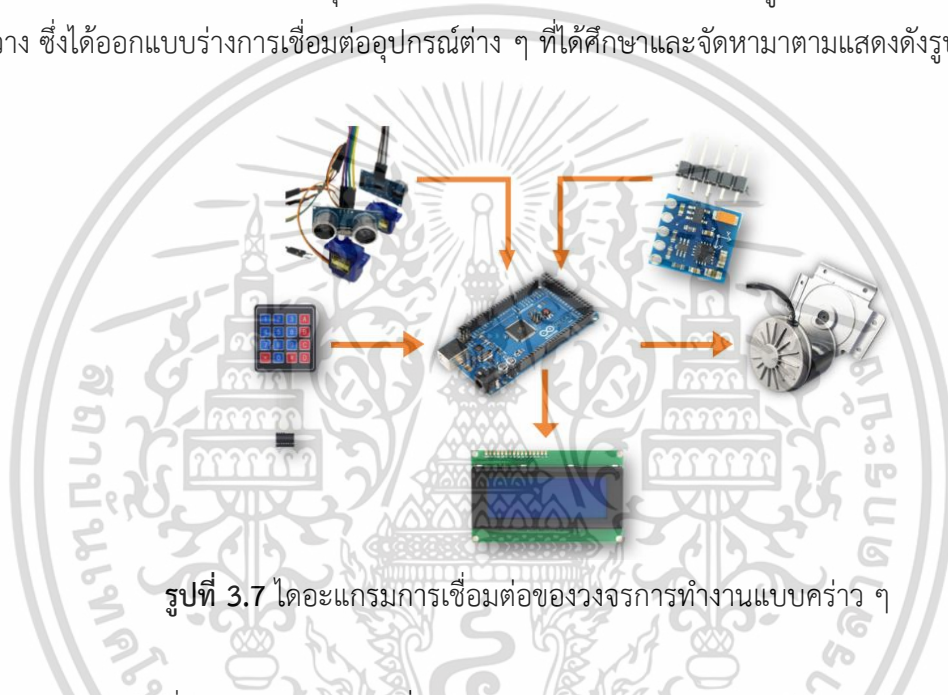
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ศึกษากระบวนการเคลื่อนที่แบบอัตโนมัติ

ทำการศึกษาทฤษฎีการเคลื่อนที่แบบอัตโนมัติของหุ่นยนต์ ตามเว็บไซต์ต่าง ๆ เพื่อนำมาเป็นแนวทางในการออกแบบโปรแกรมการทำงานการเคลื่อนที่ของรถวิลแชร์แบบอัตโนมัติ

3.5 ศึกษา และจัดหาอุปกรณ์ที่ใช้ในการรับ – ส่ง และบันทึกข้อมูล อุปกรณ์ในการตรวจจับสิ่งกีดขวาง

ได้ทำการศึกษา และจัดหาอุปกรณ์ที่ใช้การรับ – ส่ง และบันทึกข้อมูล และในการตรวจจับสิ่งกีดขวาง ซึ่งได้ออกแบบร่างการเชื่อมต่ออุปกรณ์ต่าง ๆ ที่ได้ศึกษาและจัดหามาตามแสดงดังรูป



โดยอุปกรณ์ที่อยู่ในไดอะแกรมรูปที่ 3.7 มีหลักการทำงานของอุปกรณ์แต่ละอย่างดังต่อไปนี้

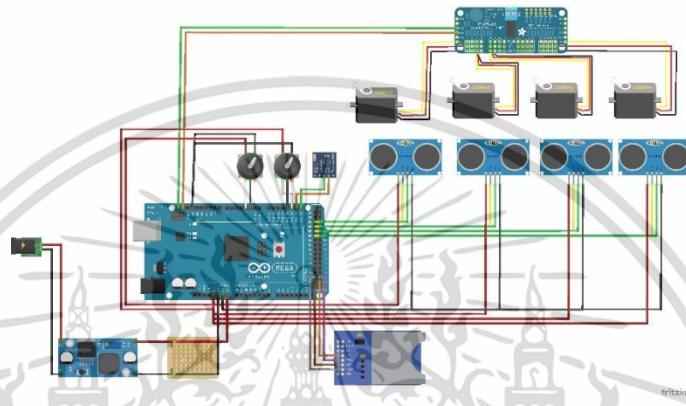
- 3.5.1 Arduino Mega 2560 ใช้เป็นอุปกรณ์ในการควบคุมการทำงานหลัก
- 3.5.2 Ultrasonic sensor ใช้เป็นอุปกรณ์ในการตรวจจับสิ่งกีดขวาง
- 3.5.3 Encoder ใช้เป็นอุปกรณ์ในการวัดความเร็วรอบการทำงานของมอเตอร์
- 3.5.5 Matrix keypad ใช้เป็นอุปกรณ์ในการเลือกเส้นทาง
- 3.5.6 LCD Display ใช้เป็นอุปกรณ์แสดงผล
- 3.5.7 DC Motor ใช้เป็นอุปกรณ์ในการขับเคลื่อนรถวิลแชร์
- 3.5.8 Micro servo motor ใช้เป็นอุปกรณ์ในการหมุน Ultrasonic sensor ให้เป็น Radar
- 3.5.9 Electronic compass module ใช้เป็นอุปกรณ์บอกทิศทางให้กับรถวิลแชร์
- 3.5.10 Adapter Micro SD Card เป็นโมดูลของ Micro SD Card ที่ใช้เชื่อมการทำงานกับไมโครคอนโทรลเลอร์ เพื่อรับ – ส่ง และบันทึกข้อมูลที่ต้องการลงใน Micro SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.11 PCA9685 16-Channel 12-bit PWM Servo shield I2C interface ทำหน้าที่เป็นตัวควบคุมการทำงานของ Micro Servo motor ที่อยู่ในวงจร

3.6 ออกแบบวงจร และออกแบบโปรแกรมการทำงานให้รถวิลแชร์สามารถเคลื่อนไปอย่างอัตโนมัติได้

3.6.1 ทำการออกแบบวงจรที่ใช้ในการบันทึกข้อมูลเส้นทางการเคลื่อนที่ของรถวิลแชร์ตามอุปกรณ์ที่ได้ศึกษาและจัดหา มา จากหัวข้อที่แล้ว



รูปที่ 3.8 รูปลักษณะการเชื่อมต่อของวงจรการทำงาน

3.6.2 เมื่อออกแบบวงจรการทำงานเสร็จเรียบร้อยแล้ว ในขั้นตอนนี้จะมีเป็นการทดสอบการทำงานของอุปกรณ์แต่ละชนิดว่าเมื่อเขียนโปรแกรมการทำงานแล้วสามารถทำงานได้ตามเงื่อนไขของโปรแกรมที่กำหนดเอาไว้หรือไม่



3.6.3 ทำการทดสอบการทำงานของอุปกรณ์ในการตรวจจับสิ่งกีดขวาง และอุปกรณ์ในการจดจำตำแหน่ง โดยอุปกรณ์ที่ใช้จะเป็นในการตรวจจับสิ่งกีดขวางคือ Ultrasonic Sensor ที่การติดอยู่บน Micro Servo Motor ซึ่งจะมีทำงานคล้ายกับการทำงานของ Radar โดยจะคอยทำเอกสสารนี้เป็นเอกสสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสแกนเพื่อตรวจจับวัดระยะห่างของวัตถุที่อยู่ทางด้านหน้า และด้านข้างรถวิลแชร์ และอุปกรณ์ที่ใช้ในการจดจำตำแหน่งของรถวิลแชร์ จะใช้เป็น Electronic compass Module หรือเซนเซอร์เข็มทิศ ในการเขียนโปรแกรมการทำงานจะทำการเปรียบเทียบค่าระยะห่าง กับค่าทิศทางที่ได้จากอุปกรณ์ในการจดจำตำแหน่ง โดยในการทดสอบการทำงานของอุปกรณ์ในการตรวจจับสิ่งกีดขวางและอุปกรณ์จดจำตำแหน่ง เมื่อเปรียบเทียบค่าระยะห่างแล้วอยู่ในเงื่อนไขที่กำหนดเอาไว้ในโปรแกรมการทำงาน ก็จะทำให้ไฟ LED แต่ละสีนั้นติดตามระยะที่ได้กำหนดเอาไว้ในเงื่อนไข และมอเตอร์จะมีการทำงานที่ช้าลง และถ้าหากค่าระยะอย่างและค่าทิศทางนั้นอยู่ในจุดที่เป็นอันตราย หรือก็คือการที่วัตถุใกล้กับอุปกรณ์ตรวจจับสิ่งกีดขวางมาก และทิศทางที่อ่านได้นั้นมีลักษณะค่าที่เอียงไปทางแกนใดแกนหนึ่งหรือพลิกกลับ ก็จะทำให้มอเตอร์หยุดการทำงานโดยทันที

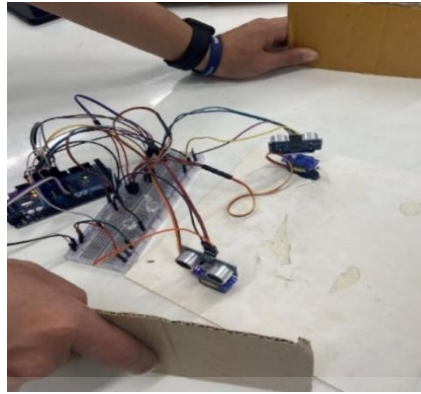


รูปที่ 3.10 การทดสอบการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวาง



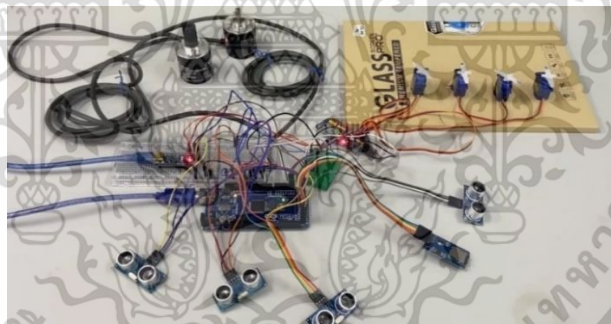
รูปที่ 3.11 การทดสอบการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวาง (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 การทดสอบการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวางร่วมกับอุปกรณ์จดจำตำแหน่ง

3.6.4 เมื่อทำการทดลองการทำงานของอุปกรณ์ตรวจจับสิ่งกีดขวาง และจดจำตำแหน่งแล้ว ก็ได้ทำการประกอบวงจรรวมทั้งหมดตามที่ได้ออกแบบในหัวข้อ 3.6.1 โดยได้มีการต่อเพิ่ม Encoder เพื่อทำการวัดความเร็วรอบการทำงานของมอเตอร์ และต่อโมดูล Adapter Micro SD Card เพื่อใช้ในการเก็บบันทึกข้อมูลค่าต่าง ๆ ที่ได้จากอุปกรณ์หรือเซนเซอร์ต่าง ๆ และทำการทดลองการทำงาน โดยเขียนโปรแกรมการทำให้สามารถวัดความเร็วรอบได้ และมีการบันทึกข้อมูลค่าต่าง ๆ ที่ได้จากอุปกรณ์หรือเซนเซอร์



รูปที่ 3.13 วงจรการทำงานที่ต่อ Encoder และ Adapter Micro SD Card เรียบร้อยแล้ว

```

DATA - Notepad
File Edit Format View Help

The Data Logger: sonar 1: 0SONIC 2 : 0SONIC 30SONIC 40Speed 1 : 0Speed 2 : 0Joy X : 0Joy Y : 0
The Data Logger: Sonic 1: 0 CM Sonic 2 : 0 CM Sonic 3 0 CM Sonic 4 0 CM Speed 1 : 0 RPM Speed 2 : 0 RPM Joy X : 0 Joy
The Data Logger: Sonic 1: 0 CM Sonic 2 : 0 CM Sonic 3 0 CM Sonic 4 0 CM Speed 1 : 0 RPM Speed 2 : 0 RPM Joy X : 0 Joy
The Data Logger: Sonic 1: 4 CM Sonic 2 : 72 CM Sonic 3 148 CM Sonic 4 81 CM Speed 1 : 0 RPM Speed 2 : 0 RPM Joy X : 0
The Data Logger: Sonic 1: 7 CM Sonic 2 : 72 CM Sonic 3 141 CM Sonic 4 79 CM Speed 1 : 635 RPM Speed 2 : 613 RPM Joy X
The Data Logger: Sonic 1: 6 CM Sonic 2 : 72 CM Sonic 3 144 CM Sonic 4 84 CM Speed 1 : 195 RPM Speed 2 : 801 RPM Joy X
The Data Logger: Sonic 1: 6 CM Sonic 2 : 72 CM Sonic 3 43 CM Sonic 4 67 CM Speed 1 : 290 RPM Speed 2 : 612 RPM Joy X
The Data Logger: Sonic 1: 6 CM Sonic 2 : 71 CM Sonic 3 134 CM Sonic 4 86 CM Speed 1 : 787 RPM Speed 2 : 372 RPM Joy X
The Data Logger: Sonic 1: 5 CM Sonic 2 : 73 CM Sonic 3 133 CM Sonic 4 84 CM Speed 1 : 979 RPM Speed 2 : 517 RPM Joy X
The Data Logger: Sonic 1: 5 CM Sonic 2 : 73 CM Sonic 3 133 CM Sonic 4 85 CM Speed 1 : 363 RPM Speed 2 : 200 RPM Joy X
The Data Logger: Sonic 1: 74 CM Sonic 2 : 67 CM Sonic 3 68 CM Sonic 4 56 CM Speed 1 : 0 RPM Speed 2 : 0 RPM Joy X : -
The Data Logger: Sonic 1: 120 CM Sonic 2 : 73 CM Sonic 3 147 CM Sonic 4 81 CM Speed 1 : 10 RPM Speed 2 : 102 RPM Joy
The Data Logger: Sonic 1: 64 CM Sonic 2 : 74 CM Sonic 3 73 CM Sonic 4 45 CM Speed 1 : 22 RPM Speed 2 : 260 RPM Joy X
The Data Logger: Sonic 1: 67 CM Sonic 2 : 72 CM Sonic 3 72 CM Sonic 4 47 CM Speed 1 : 20 RPM Speed 2 : 236 RPM Joy X
The Data Logger: Sonic 1: 60 CM Sonic 2 : 71 CM Sonic 3 78 CM Sonic 4 43 CM Speed 1 : 15 RPM Speed 2 : 199 RPM Joy X
The Data Logger: Sonic 1: 63 CM Sonic 2 : 73 CM Sonic 3 75 CM Sonic 4 43 CM Speed 1 : 22 RPM Speed 2 : 183 RPM Joy X
The Data Logger: Sonic 1: 55 CM Sonic 2 : 72 CM Sonic 3 82 CM Sonic 4 37 CM Speed 1 : 16 RPM Speed 2 : 203 RPM Joy X
The Data Logger: Sonic 1: 0 CM Sonic 2 : 0 CM Sonic 3 0 CM Sonic 4 0 CM Speed 1 : 0 RPM Speed 2 : 0 RPM Joy X : 0 Joy
The Data Logger: Sonic 1: 60 CM Sonic 2 : 70 CM Sonic 3 80 CM Sonic 4 43 CM Speed 1 : 0 RPM Speed 2 : 0 RPM Joy X : 0
The Data Logger: Sonic 1: 115 CM Sonic 2 : 72 CM Sonic 3 149 CM Sonic 4 75 CM Speed 1 : 485 RPM Speed 2 : 522 RPM Joy
The Data Logger: Sonic 1: 95 CM Sonic 2 : 71 CM Sonic 3 54 CM Sonic 4 56 CM Speed 1 : 365 RPM Speed 2 : 280 RPM Joy X
The Data Logger: Sonic 1: 104 CM Sonic 2 : 72 CM Sonic 3 47 CM Sonic 4 61 CM Speed 1 : 468 RPM Speed 2 : 352 RPM Joy X
The Data Logger: Sonic 1: 93 CM Sonic 2 : 70 CM Sonic 3 56 CM Sonic 4 56 CM Speed 1 : 414 RPM Speed 2 : 346 RPM Joy X

```

รูปที่ 3.14 ค่าตัวอย่างข้อมูลของการทำงานของอุปกรณ์หรือเซนเซอร์ที่ต่ออยู่กับวงจร

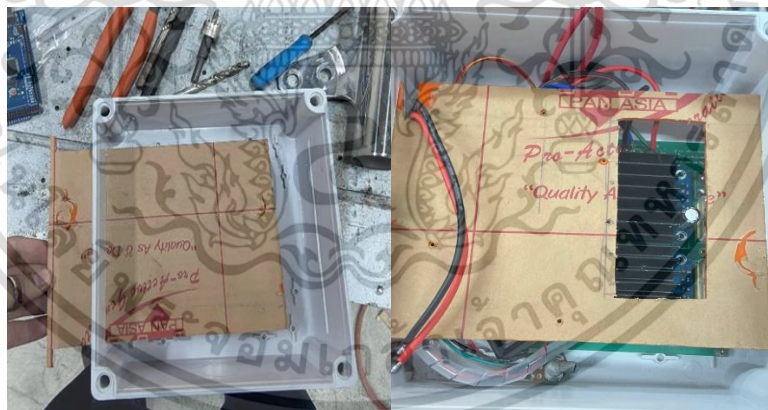
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.5 เมื่อทำการทดสอบการทำงานของอุปกรณ์ทั้งหมดเรียบร้อยแล้ว หลังจากนั้นก็ทำการออกแบบการติดตั้งอุปกรณ์หรือเซ็นเซอร์ที่ได้ทดลองไว้เข้ากับตัวรถวีลแชร์ไฟฟ้า



รูปที่ 3.15 การออกแบบการติดตั้งอุปกรณ์หรือเซ็นเซอร์

3.6.6 เมื่อได้ทำการออกแบบตำแหน่งการติดตั้งอุปกรณ์หรือเซ็นเซอร์ที่ใช้เรียบร้อยแล้วก็ทำการออกแบบกล่องสำหรับบรรจุวงจรที่ใช้ในการทำงาน ประกอบไปด้วยวงจรขับเคลื่อน และวงจรรับข้อมูล เมื่อออกแบบเสร็จก็ทำการติดตั้งอุปกรณ์หรือเซ็นเซอร์เข้าไปที่ตัวรถ บรรจุวงจรลงในกล่องที่ได้ทำการออกแบบไว้ และเดินสายไฟของอุปกรณ์หรือเซ็นเซอร์เพื่อต่อเข้ากับกล่องที่บรรจุวงจรการทำงาน



รูปที่ 3.16 การออกแบบกล่องบรรจุวงจรการทำงานของรถวีลแชร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

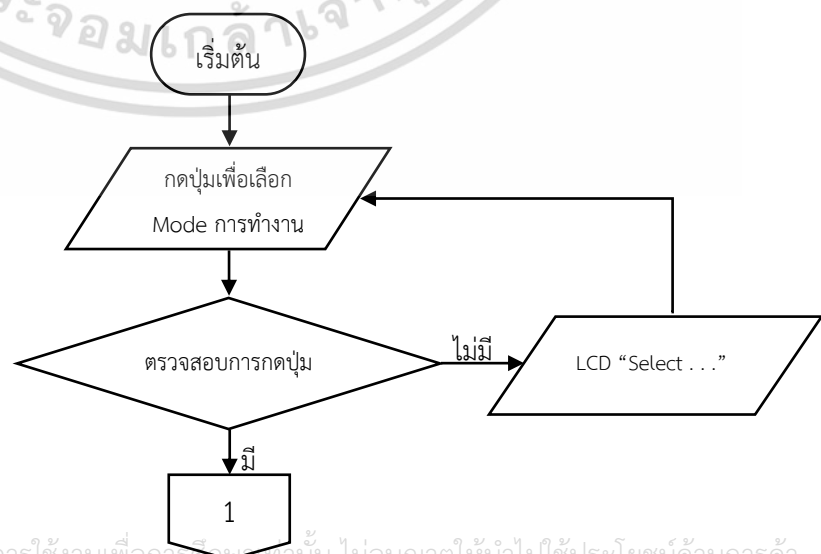


รูปที่ 3.17 ประกอบอุปกรณ์หรือเซ็นเซอร์ วงจรการทำงานเข้ากับตัวรถตามที่ออกแบบไว้

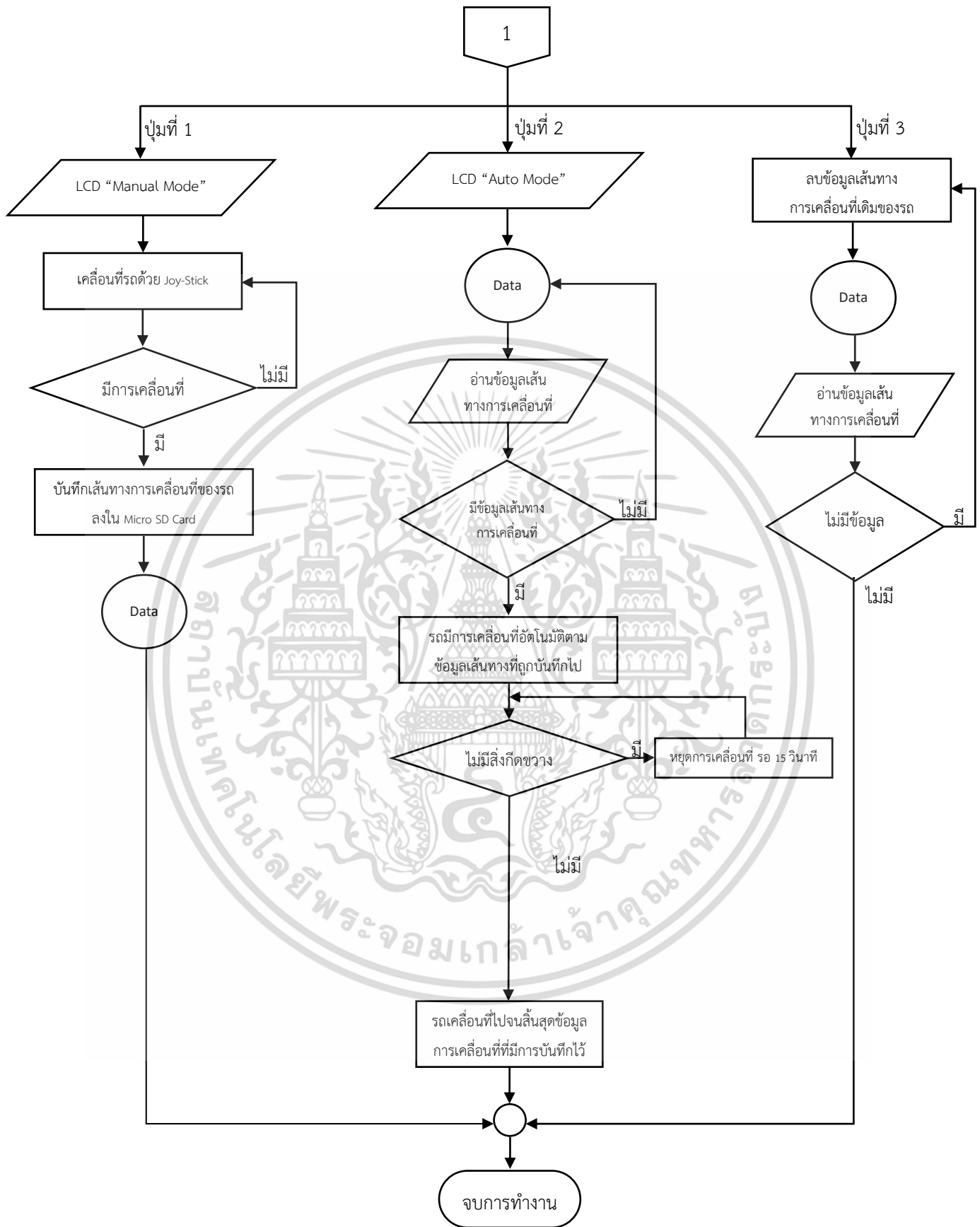


รูปที่ 3.18 เดินสายไฟของอุปกรณ์หรือเซ็นเซอร์

3.6.7 เมื่อติดตั้งอุปกรณ์หรือเซ็นเซอร์ พร้อมทั้งเดินสายไฟเรียบร้อยแล้ว ทางคณะผู้จัดทำได้ศึกษาข้อมูลเกี่ยวกับการเคลื่อนที่อัตโนมัติ จึงได้ทำการออกแบบโปรแกรมการทำงานของรถวิลแชร์ไฟฟ้าที่มีการเคลื่อนที่โดยใช้ Joy-Stick ให้มีการเคลื่อนที่เป็นแบบอัตโนมัติ ตาม Flow Chart การทำงานดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 3.19 แผ่นผังโปรแกรมการทำงานของรถวิลแชร์เคลื่อนที่แบบอัตโนมัติให้นำไปใช้



รูปที่ 3.20 แผนผังโปรแกรมการทำงานของรถวีลแชร์เคลื่อนที่แบบอัตโนมัติ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้จะเป็นผลการทดลองการทำงานของรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้ โดยจะใช้เป็นการกำหนดเส้นทางเคลื่อนที่ให้กับรถวีลแชร์ และทำการเก็บข้อมูลเส้นทางการเคลื่อนที่ด้วยการเคลื่อนที่แบบใช้ Joy-Stick เมื่อเก็บเส้นทางการเคลื่อนที่เสร็จเรียบร้อยแล้วก็จะให้มีการเคลื่อนที่แบบอัตโนมัติ และเพื่อให้เป็นตามวัตถุประสงค์ที่ได้มีการกำหนดไว้ จะสามารถแบ่งผลการทดลองออกเป็น 5 หัวข้อ ดังนี้

- 4.1 การทดลองการเก็บข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์
- 4.2 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์
- 4.3 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงแยกของรถวีลแชร์
- 4.4 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงแยกของรถวีลแชร์
- 4.5 การทดลองการเคลื่อนที่แบบอัตโนมัติของรถวีลแชร์ในเส้นทางที่มีการเลี้ยวมากกว่า 1 ครั้ง
- 4.6 การทดลองการส่งค่าจาก Joy-Stick ระหว่างบอร์ดรับข้อมูลและบอร์ดขับเคลื่อน

4.1 การทดลองการเก็บข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์

การทดลองนี้จะเป็นเขียนโปรแกรมการทำงานให้มีการบันทึกข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์ลงใน Micro SD Card เมื่อกดปุ่มเลือกการทำงานเป็นแบบ “Manual Mode” และรถเกิดการเคลื่อนที่ไปยังเส้นทางที่กำหนดแต่ละเส้นทางที่มีระยะห่างไม่เท่ากัน โดยจะใช้การควบคุมการทำงานของรถวีลแชร์ด้วย Joy-Stick เมื่อสิ้นสุดเส้นทางที่กำหนดก็จะนำ Micro SD Card มาตรวจสอบดูว่ามีข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์หรือไม่ โดยการทดลองนี้จะทำการทดลองทั้งจำนวน 5 ครั้ง เพื่อความสามารถประสิทธิภาพการทำงานของบอร์ดรับข้อมูลที่มีการบันทึกข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์ลงใน Micro SD Card

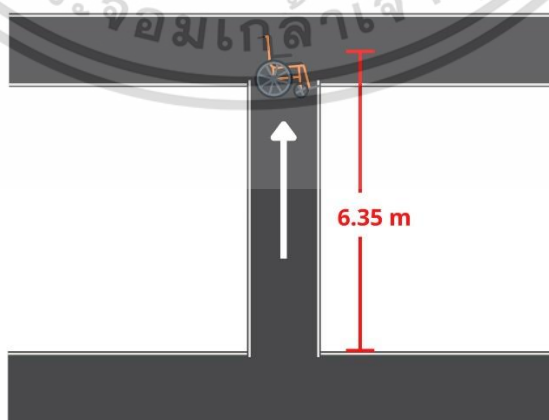
ตารางที่ 4.1 การเก็บข้อมูลเส้นทางการเคลื่อนที่ของรถวีลแชร์

ครั้งที่	ระยะห่างของเส้นทาง จากจุดเริ่มต้น	ข้อมูลเส้นทางมีการถูกบันทึก	ความถูกต้องของข้อมูลเส้นทาง
1	5 m	มีการบันทึก	ความถูกต้องน้อย
2	10 m	มีการบันทึก	ความถูกต้องสูง
3	15 m	มีการบันทึก	ความถูกต้องสูง
4	5 m	มีการบันทึก	ความถูกต้องสูง
5	25 m	มีการบันทึก	ความถูกต้องสูง

จากการทดลองการเก็บข้อมูลเส้นทางเมื่อเลือกการทำงาน และเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick ไปจากการทดลองทั้ง 5 ครั้งพบว่ามีการบันทึกเส้นทางการเคลื่อนที่ของรถวีลแชร์ตลอดทั้ง 5 ครั้ง แต่ในระยะความห่างจากจุดเริ่มที่ 5 m. นั้นมีความถูกต้องของข้อมูลเส้นทางที่น้อย มีข้อมูลตกลงออกไปบางช่วง จึงทำให้สรุปได้ว่าในระยะห่างของเส้นทางที่สั้นนั้นมีผลต่อความถูกต้องของข้อมูลเส้นทางการเคลื่อนที่

4.2 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์

การทดลองนี้จะเป็นการที่เลือกการทำงานของรถวีลแชร์เป็นแบบ “Manual Mode” ก่อนเพื่อทำการเก็บเส้นทางการเคลื่อนที่ในแนวทางตรงตามรูปที่ 4.1 โดยจะใช้การควบคุมการทำงานของรถวีลแชร์ด้วย Joy-Stick และเมื่อทำการเก็บเส้นทางการเคลื่อนที่เรียบร้อยแล้วก็นำรถวีลแชร์ไปตั้งที่จะเริ่มต้น และเปลี่ยนการทำงานจากแบบ “Manual Mode” เป็นแบบ “Auto Mode” ก็จะทำให้รถวีลแชร์มีการเคลื่อนที่แบบอัตโนมัติตามข้อมูลเส้นทางการเคลื่อนที่ที่ถูกบันทึกเอาไว้ โดยการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์จะทำการทดลองจำนวน 10 ครั้งเพื่อหาประสิทธิภาพในการเคลื่อนที่ไปยังจุดหมาย และความเที่ยงตรงในการเคลื่อนที่



รูปที่ 4.1 ระยะเส้นทางการเคลื่อนที่ของรถวีลแชร์ในแนวทางตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

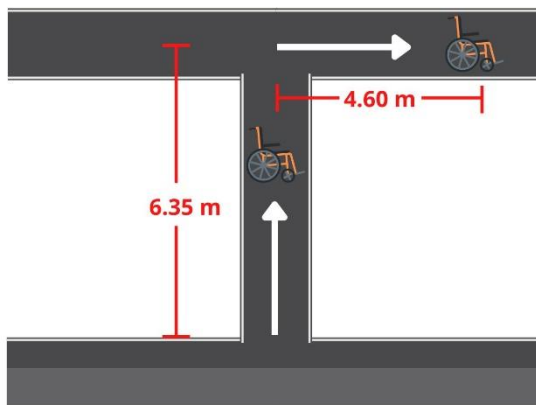
ตารางที่ 4.2 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์

ครั้งที่	จากเริ่มต้นถึงจุดหมาย เส้นทางการเคลื่อนที่	ระยะห่างจาก จุดหมายไปทางซ้าย	ระยะห่างจาก จุดหมายไปทางขวา	ความเที่ยงตรง
1	ถึงจุดหมาย	0	0	✓
2	ถึงจุดหมาย	9 cm.	0	✗
3	ถึงจุดหมาย	0	6 cm	✗
4	ถึงจุดหมาย	9 cm.	0	✗
5	ถึงจุดหมาย	9 cm.	0	✗
6	ถึงจุดหมาย	9 cm.	0	✗
7	ถึงจุดหมาย	0	5 cm.	✗
8	ถึงจุดหมาย	0	0	✓
9	ถึงจุดหมาย	0	0	✓
10	ถึงจุดหมาย	0	0	✓

จากการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวเส้นทางตรง ทั้ง 10 ครั้ง พบว่าในการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงรถวีลแชร์สามารถเคลื่อนที่จากจุดเริ่มต้นไปถึงจุดหมายได้ทั้ง 10 ครั้ง โดยที่เมื่อรถวีลแชร์ถึงจุดหมายจะมีระยะห่างจากจุดหมายที่ออกไปทางซ้ายเฉลี่ยอยู่ที่ 9 cm. และออกไปทางขวาเฉลี่ยอยู่ที่ 5.5 cm. จึงทำให้การเคลื่อนที่ในแนวเส้นทางตรงมีความเที่ยงตรงอยู่ที่ 40 %

4.3 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยกของรถวีลแชร์

การทดลองนี้จะเป็นการที่เลือกการทำงานของรถวีลแชร์เป็นแบบ “Manual Mode” ก่อนเพื่อทำการเก็บเส้นทางการเคลื่อนที่ในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยก ตามรูปที่ 4.2 โดยจะทำการควบคุมการทำงานของรถวีลแชร์ด้วย Joy-Stick และเมื่อทำการเก็บเส้นทางการเคลื่อนที่เรียบร้อยแล้วก็นำรถวีลแชร์ไปตั้งที่จะเริ่มต้น และเปลี่ยนการทำงานจากแบบ “Manual Mode” เป็นแบบ “Auto Mode” ก็จะทำให้รถวีลแชร์มีการเคลื่อนที่แบบอัตโนมัติตามข้อมูลเส้นทางการเคลื่อนที่ที่ถูกบันทึกเอาไว้ โดยการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยกของรถวีลแชร์จะทำการทดลองจำนวน 10 ครั้งเพื่อหาประสิทธิภาพในการเคลื่อนที่ไปยังจุดหมายและความเที่ยงตรงในการเคลื่อนที่



รูปที่ 4.2 ระยะเส้นทางการเคลื่อนที่ของรถวีลแชร์ในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยก

ตารางที่ 4.3 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยกของรถวีลแชร์

ครั้งที่	จากเริ่มต้นถึงจุดหมาย เส้นทางการเคลื่อนที่		ระยะห่างจาก จุดหมายไปทางขวา		ระยะห่างจาก จุดหมายไปทางซ้าย		ความ เที่ยงตรง
	ต้น - แยก	แยก - ปลาย	ต้น - แยก	แยก - ปลาย	ต้น - แยก	แยก - ปลาย	
1	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	10 cm.	X
2	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	0	0	10 cm.	X
3	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	10 cm.	0	0	X
4	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	0	0	15 cm.	X
5	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	15 cm.	X
6	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	15 cm.	X
7	ถึงจุดหมาย	ถึงจุดหมาย	0	0	5 cm.	10 cm.	X
8	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	15 cm.	X
9	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	15 cm.	X
10	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	5 cm.	X

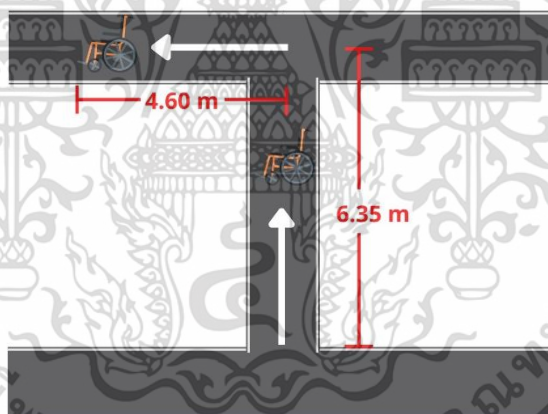
จากการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยก ทั้ง 10 ครั้ง พบว่าในการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยกสามารถเคลื่อนที่จากจุดเริ่มต้นไปถึงทางแยก พอเมื่อเลี้ยวขวาก็สามารถเลี้ยวขวาแล้วเคลื่อนที่ไปยังปลายเส้นทางได้ แต่พบว่าการเคลื่อนที่แต่ละครั้งยังมีความไม่เที่ยงตรง โดยที่เมื่อเคลื่อนที่จากจุดเริ่มต้นไปถึงทางแยกมีค่าเฉลี่ยของระยะห่างจากจุดหมายไปทางด้านขวาอยู่ที่ 5 cm. และทางซ้ายอยู่ที่ 5 cm. และเมื่อเคลื่อนที่จากแยกไปถึงปลายทางพบว่ามีค่าเฉลี่ยระยะห่างจากจุดหมายไปทางด้านขวาอยู่ที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cm. และทางด้านซ้ายอยู่ที่ 12.22 cm. จึงสามารถสรุปได้ว่าการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวายังมีการเคลื่อนที่ไม่ตรงกับจุดเป้าหมาย เนื่องการทำงานของมอเตอร์ทั้ง 2 ข้างมีการทำงานที่ไม่เท่ากัน และยังมีผลมาจากล้อคู่หน้าที่มีทิศทางการหมุนได้ถึง 360 องศา จึงทำให้เมื่อเวลารถวิลแชร์เลี้ยวแล้วเคลื่อนที่ต่อเลยไม่ตรงตามเส้นทางการเคลื่อนที่ที่กำหนดไว้

4.4 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยกของรถวิลแชร์

การทดลองนี้จะเป็นการที่เลือกการทำงานของรถวิลแชร์เป็นแบบ “Manual Mode” ก่อนเพื่อทำการเก็บเส้นทางการเคลื่อนที่ในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยก ตามรูปที่ 4.3 โดยจะทำการควบคุมการทำงานของรถวิลแชร์ด้วย Joy-Stick และเมื่อทำการเก็บเส้นทางการเคลื่อนที่เรียบร้อยแล้วก็นำรถวิลแชร์ไปตั้งที่จะเริ่มต้น และเปลี่ยนการทำงานจากแบบ “Manual Mode” เป็นแบบ “Auto Mode” ก็จะทำให้รถวิลแชร์มีการเคลื่อนที่แบบอัตโนมัติตามข้อมูลเส้นทางการเคลื่อนที่ที่ถูกบันทึกเอาไว้ โดยการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยกของรถวิลแชร์จะทำการทดลองจำนวน 10 ครั้ง เพื่อหาประสิทธิภาพในการเคลื่อนที่ไปยังจุดหมาย และความเที่ยงตรงในการเคลื่อนที่



รูปที่ 4.3 ระยะเส้นทางการเคลื่อนที่ของรถวิลแชร์ในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยก

ตารางที่ 4.4 การทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยก
ของรถวิลแชร์

ครั้งที่	จากเริ่มต้นถึงจุดหมาย เส้นทางการเคลื่อนที่		ระยะห่างจาก จุดหมายไปทางขวา		ระยะห่างจาก จุดหมายไปทางซ้าย		ความ เที่ยงตรง
	ต้น - แยก	แยก - ปลาย	ต้น - แยก	แยก - ปลาย	ต้น - แยก	แยก - ปลาย	
1	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	15 cm.	0	0	X
2	ถึงจุดหมาย	ถึงจุดหมาย	0	10 cm.	5 cm.	0	X
3	ถึงจุดหมาย	ถึงจุดหมาย	0	20 cm.	0	0	X
4	ถึงจุดหมาย	ถึงจุดหมาย	0	0	0	5 cm.	X
5	ถึงจุดหมาย	ถึงจุดหมาย	0	15 cm.	0	0	X
6	ถึงจุดหมาย	ถึงจุดหมาย	0	15 cm.	5 cm.	0	X
7	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	15 cm.	0	0	X
8	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	15 cm.	0	0	X
9	ถึงจุดหมาย	ถึงจุดหมาย	10 cm.	15 cm.	0	0	X
10	ถึงจุดหมาย	ถึงจุดหมาย	5 cm.	15 cm.	0	0	X

จากการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยก ทั้ง 10 ครั้ง พบว่าในการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยกสามารถเคลื่อนที่จากจุดเริ่มต้นไปถึงทางแยก พอเมื่อเลี้ยวซ้ายก็สามารถเลี้ยวซ้ายแล้วเคลื่อนที่ไปยังปลายเส้นทางได้ แต่พบว่าในการเคลื่อนที่แต่ละครั้งยังมีความไม่เที่ยงตรง โดยที่เมื่อเคลื่อนที่จากจุดเริ่มต้นไปถึงทางแยกมีค่าเฉลี่ยของระยะห่างจากจุดหมายไปทางด้านขวาอยู่ที่ 6 cm. และทางซ้ายอยู่ที่ 5 cm. และเมื่อเคลื่อนที่จากแยกไปถึงปลายทางพบว่ามีค่าเฉลี่ยระยะห่างจากจุดหมายไปทางด้านขวาอยู่ที่ 15 cm. และทางด้านซ้ายอยู่ที่ 5 cm. จึงสามารถสรุปได้ว่าการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายมีการเคลื่อนที่ที่เหมือนกับการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวา ซึ่งก็คือมีการเคลื่อนที่ไม่ตรงกับจุดเป้าหมาย เนื่องการทำงานของมอเตอร์ทั้ง 2 ข้างมีการทำงานที่ไม่เท่ากัน และยังมีผลมาจากล้อคู่หน้าที่มีทิศทางการหมุนได้ถึง 360 องศา จึงทำให้เมื่อเวลารถวิลแชร์เลี้ยวแล้วเคลื่อนที่ต่อเลยไม่ตรงตามเส้นทางการเคลื่อนที่ที่กำหนดไว้

4.5 การทดลองการเคลื่อนที่แบบอัตโนมัติของรถวิลแชร์ในเส้นทางที่มีการเลี้ยว มากกว่า 1 ครั้ง

การทดลองนี้จะเป็นการที่เลือกการทำงานของรถวิลแชร์เป็นแบบ “Manual Mode” ก่อนเพื่อทำการเก็บเส้นทางการเคลื่อนที่ที่มีการเลี้ยวมากกว่า 1 ครั้ง ตามรูปที่ 4.4 โดยจะทำการควบคุมการทำงานของรถวิลแชร์ด้วย Joy-Stick และเมื่อทำการเก็บเส้นทางการเคลื่อนที่เรียบร้อยแล้วเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองการเคลื่อนที่แบบอัตโนมัติในเส้นทางที่มีการเลี้ยวมากกว่า 1 ครั้ง ทั้ง 10 ครั้ง พบว่าในการเคลื่อนที่แบบอัตโนมัติในเส้นทางที่มีการเลี้ยวมากกว่า 1 ครั้ง สามารถเคลื่อนที่จากจุดเริ่มต้นไปถึงทางแยก 1 พอเมื่อเลี้ยวขวาทางแยกที่ 1 ก็สามารถเลี้ยวขวาแล้วเคลื่อนที่ไปยังทางแยกที่ 2 ได้ และพอเมื่อถึงทางแยกที่ 2 ก็สามารถเลี้ยวซ้ายและเคลื่อนที่ตรงไปยังปลายเส้นทางได้ แต่พบว่าการเคลื่อนที่แต่ละครั้งไปยังแต่ละจุดยังมีความไม่เที่ยงตรง โดยที่เมื่อเคลื่อนที่จากจุดเริ่มต้นไปถึงทางแยกที่ 1 มีค่าเฉลี่ยของระยะห่างจากจุดหมายไปทางด้านขวายู่ที่ 12.5 cm. และทางซ้ายอยู่ 9.16 cm. และเมื่อเคลื่อนที่จากทางแยก 1 ไปถึงทางแยกที่ 2 มีค่าเฉลี่ยระยะห่างจากจุดหมายไปทางด้านขวายู่ที่ 11.66 cm. และทางด้านซ้ายอยู่ 10 cm. และเมื่อเคลื่อนที่จากทางแยก 2 ไปถึงปลายเส้นทาง มีค่าเฉลี่ยระยะห่างจากจุดหมายไปทางด้านขวายู่ที่ 6.66 cm. และทางด้านซ้ายอยู่ 5 cm. จึงสามารถสรุปได้ว่าที่มีการเลี้ยวมากกว่า 1 ครั้ง มีการเคลื่อนที่ที่เหมือนกับการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวา และเลี้ยวซ้ายซึ่งก็คือมีการเคลื่อนที่ไม่ตรงกับจุดเป้าหมายเนื่องการทำงานของมอเตอร์ทั้ง 2 ข้างมีการทำงานที่ไม่เท่ากัน และยังมีผลมาจากล้อคู่หน้าที่มีทิศทางการหมุนได้ถึง 360 องศา จึงทำให้เมื่อเวลารถวีลแชร์เลี้ยวแล้วเคลื่อนที่ต่อเลยไม่ตรงตามเส้นทางเคลื่อนที่ที่กำหนดไว้

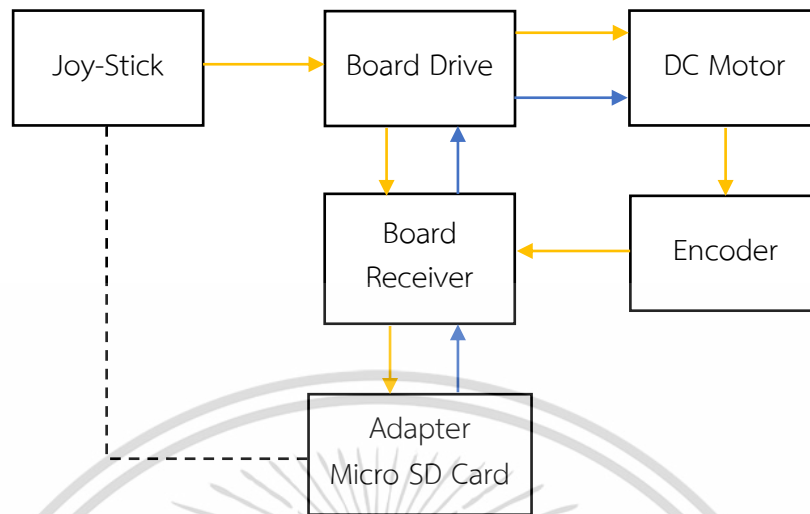
4.6 การทดลองการส่งค่าจาก Joy-Stick ระหว่างบอร์ดรับข้อมูลและบอร์ดขับเคลื่อน

ในการทดลองนี้จะเป็นการทดลองการทำงานในการส่งค่า Joy-Stick ระหว่างบอร์ดรับข้อมูลและบอร์ดขับเคลื่อนของรถวีลแชร์

1. โหมด Manual การทำงานในโหมดนี้คือการที่รถวีลแชร์จะทำการบันทึกค่าทิศทาง X,Y ของ Joy-Stick ลงไปใน Micro SD Card ซึ่งในการบันทึกข้อมูลทิศทางของ Joy-Stick จะเริ่มทำการบันทึกเมื่อรถวีลแชร์เกิดการเคลื่อนที่ไป หรือมีค่ารอบการทำงานของมอเตอร์ทั้ง 2 ข้างมากกว่า 1 RPM โดยจะทำการบันทึกข้อมูลทิศทางของ Joy-Stick ตามเส้นทางการเคลื่อนที่ที่ต้องเมื่อต้องการข้อมูลเส้นทางตามที่ต้องการแล้วต้องทำการกดอีกครั้งเพื่อหยุดการบันทึกข้อมูล

2. โหมด Auto การทำงานในโหมดนี้คือการที่บอร์ดรับข้อมูลจะทำการดึงข้อมูลทิศทาง X,Y ของ Joy-Stick ที่อยู่ใน Micro SD Card ออกมาแล้วส่งไปยังบอร์ดขับเคลื่อน บอร์ดขับเคลื่อนก็จะทำการนำข้อมูลที่ถูส่งมาเปรียบเทียบกับแต่ละเงื่อนไขว่าค่าที่ได้มาตรงกับเงื่อนไขการเคลื่อนที่แบบใด ซึ่งก็จะทำให้รถวีลแชร์เกิดการเคลื่อนที่ไปอย่างอัตโนมัติตามข้อมูลทิศทางของ Joy-Stick ที่ถูกบันทึกอยู่ใน Micro SD Card

โดยลักษณะการทำงานของารรับ - ส่งข้อมูลจะแสดงดังรูปที่ 4.5



รูปที่ 4.5 บล็อกไดอะแกรมการทำงานในการส่งค่า Joy-Stick ระหว่างบอร์ดรับข้อมูลและบอร์ดขับเคลื่อน

โดยที่ ——— คือการทำงานในโหมด Manual และทิศทางการส่งข้อมูล Joy-Stick
 ——— คือการทำงานในโหมด Auto และทิศทางการส่งข้อมูล Joy-Stick
 - - - - - คือข้อมูลทิศทางของ Joy-Stick

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ปฏิญญาฉบับนี้ได้ประยุกต์การทำงานของรถวีลแชร์ที่ใช้ Joy-Stick ในการบังคับทิศทางการเคลื่อนที่เพียงอย่างเดียว โดยไม่อาจตอบสนองต่อความต้องการทุกเพศทุกวัย หรือผู้ป่วยที่ช่วยเหลือตนเองไม่ได้ ซึ่งทางคณะผู้จัดทำได้นำรถวีลแชร์ไฟฟ้ามาทำการประยุกต์ เพื่อให้สามารถเคลื่อนที่ได้โดยอัตโนมัติตามเส้นทางที่กำหนดไว้ และเพื่อความสะดวกต่อผู้ป่วยที่ไม่สามารถช่วยเหลือตนเองได้ และจากการทดลองในบทที่ 4 ได้มีการทดสอบเพื่อหาประสิทธิภาพในการเคลื่อนที่ไปยังจุดหมาย และความเที่ยงตรงในการเคลื่อนที่ โดยใช้การที่เก็บข้อมูลเส้นทาง การเคลื่อนที่ที่ต้องการ เมื่อเก็บข้อมูลเส้นทางเสร็จเรียบร้อยแล้วก็นำรถวีลแชร์กลับไปตั้งที่จุดเริ่มต้น และเปลี่ยนการทำงานเป็นแบบอัตโนมัติเพื่อให้รถวีลแชร์เคลื่อนที่ไปตามข้อมูลเส้นทางที่ถูกเก็บเอาไว้ ซึ่งจากตารางการทดลองที่ 4.1 ถึงตารางที่ 4.2 เป็นการทดสอบการทำงานของในการเก็บข้อมูลเมื่อรถวีลแชร์เคลื่อนที่ไป ตารางการทดลองที่ 4.3 เป็นการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงของรถวีลแชร์ ตารางการทดลองที่ 4.4 เป็นการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวขวาเมื่อถึงทางแยกของรถวีลแชร์ ตารางการทดลองที่ 4.5 ถึงตารางที่ 4.6 เป็นการทดลองการเคลื่อนที่แบบอัตโนมัติในแนวทางตรงแล้วเลี้ยวซ้ายเมื่อถึงทางแยกของรถวีลแชร์ และตารางการทดลองที่ 4.7 เป็นการทดลองการเคลื่อนที่แบบอัตโนมัติในเส้นทางที่มีการเลี้ยวมากกว่า 1 ครั้ง จากการทดลองทั้งหมดได้มีการทดลองเป็นจำนวน 10 ครั้ง ซึ่งผลการทดลองพบว่า มีประสิทธิภาพการทำงานในการเก็บข้อมูลอยู่เกณฑ์ที่ดีมาก ส่วนการเคลื่อนอัตโนมัติในแนวตรง มีประสิทธิภาพอยู่ที่ 40% ซึ่งแตกต่างจากการเคลื่อนที่แบบอัตโนมัติแล้วเมื่อเจอทางแยกให้เลี้ยวซ้ายหรือเลี้ยวขวา หรือมีการเลี้ยวมากกว่า 1 ครั้ง ประสิทธิภาพการทำงานเมื่อรถวีลแชร์มีการเลี้ยว มีประสิทธิภาพที่ไม่ค่อยเที่ยงตรงเป็นเพราะว่าล้อหน้า 2 ล้อที่มีทิศทางการหมุนได้ 360 องศา จึงส่งผลเมื่อรถมีการเลี้ยวแล้วเคลื่อนที่ต่อไป เคลื่อนที่ได้ไม่ตรงจุดหมาย ยังรวมไปถึงปัจจัยของการทำงานของมอเตอร์ที่ไม่เท่ากันทั้ง 2 ข้าง ทำให้สามารถสรุปการทดลองได้ว่าการที่ประยุกต์รถวีลแชร์ไฟฟ้าให้สามารถเคลื่อนที่อัตโนมัติได้สามารถบรรลุวัตถุประสงค์ตามที่ได้มีการกำหนดไว้ เพียงแต่จะต้องมีการปรับปรุงประสิทธิภาพในการเคลื่อนที่ให้สามารถเคลื่อนที่ไปยังจุดหมายได้มีความเที่ยงตรงมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาและอุปสรรคที่พบระหว่างดำเนินการ

5.2.1 ในระหว่างการดำเนินการติดตั้งอุปกรณ์หรือเซนเซอร์นั้นมีการตัดสินใจที่ยากมากพอสมควร ต้องมีความแน่นอนที่สูงเนื่องจากโครงการของรถวีลแชร์นั้นมีโครงสร้างที่ขนาดเล็กจึงทำให้เวลาในการที่จะติดตั้งอุปกรณ์หรือเซนเซอร์เพิ่มเติมนั้นทำได้ยาก เพราะถ้าหากทำผิดพลาดอาจทำให้โครงการเกิดความชำรุดเสียหายและไม่สามารถแก้ไขได้

5.2.2 ในระหว่างการดำเนินการเขียนโปรแกรมการทำงานเพื่อรับส่งข้อมูลระหว่างบอร์ดขับเคลื่อนและบอร์ดรับข้อมูลนั้นมีข้อจำกัดในเรื่องของ Hardware ที่เยอะจึงทำให้เสียเวลาในการเขียนโปรแกรมเพื่อรับส่งข้อมูลระหว่างบอร์ด

5.2.3 ในระหว่างการดำเนินการทดลองการทำงาน เพื่อให้รถสามารถเคลื่อนที่ไปอย่างอัตโนมัติ นั้น เกิดข้อผิดพลาดในการที่ไฟย้อนกลับเข้ามาสู่ที่คอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมจึงทำให้คอมพิวเตอร์เกิดความชำรุดเสียหาย จึงจำเป็นต้องมีความระมัดระวังในเรื่องการระบบสายไฟที่เชื่อมต่ออยู่กับอุปกรณ์พอสมควร

5.2.4 เนื่องจากทางคณะผู้จัดทำได้มีการวางแผนเอาไว้ว่าจะมีการใช้ข้อมูลจากอุปกรณ์ Encoder, Ultrasonic, และ Electronic Compass Module มาช่วยเป็นข้อมูลสำหรับการเคลื่อนที่

5.3 ข้อเสนอแนะเพิ่มเติม

5.3.1 ควรที่จะรวมเอาบอร์ดรับข้อมูลและบอร์ดขับเคลื่อนเข้ามารวมกันให้อยู่ภายในบอร์ดเดียวกัน เพื่อให้เกิดความเสถียรภาพในการทำงานของตัวรถวีลแชร์

5.3.2 ควรมีการแก้ไขโครงรถให้มีขนาดที่ใหญ่ขึ้นมากกว่าของเดิม เพื่อให้สามารถที่จะติดตั้งอุปกรณ์เสริมหรือเซนเซอร์ต่างๆที่ใช้ในการควบคุมให้ได้ง่ายขึ้น

5.3.3 ควรมีการใช้อุปกรณ์หรือเซนเซอร์ที่ใช้ในการจดจำตำแหน่งให้มีความเสถียรภาพมากขึ้น

5.3.4 ควรพัฒนาให้สามารถควบคุมทางไกลหรือระบบ IOT ได้

5.3.5 ควรมีการดึงข้อมูลจากอุปกรณ์ Encoder, Ultrasonic, และ Electronic Compass Module มาใช้เพื่อทำให้มีการเคลื่อนที่ได้เที่ยงตรงมากขึ้น

บรรณานุกรม

- [1] ยุทธนา ปิติธีรภาพ. (2555). ชุดขับเคลื่อนไฟฟ้า สำหรับรถเข็น Engineering Conference Thai Biomedical ครั้งที่ 4. เตชะธัชบุรณะอัครกุล. (2555). Bright Brain มหาวิทยาลัยไฮเทค มหาวิทยาลัยราชภัฏธนบุรีปราจีนพลังสันติกุล (2549).
- [2] ณัฐพลวงศ์ สันติชัย (2547). เรียนรู้และใช้งานไมโครคอนโทรลเลอร์ PIC16F877A บริษัท อินโนเวชั่นอิเล็กทรอนิกส์ จำกัด
- [3] Ebikethaikit (2007). ชุด Kit ทำรถวีลแชร์ไฟฟ้า ควบคุมโดย Arduino Driver Board eBikeThaikit.com (2007)
- [4] Cybertice (2022). SG90 Micro Servo Motor แบบ หมุน 360 องศาต่อเนื่อง และ Photoelectric Sensor Encoder บริษัท เอส. สมาร์ทเทค ซิสเต็ม จำกัด
- [5] Robotsiam (2022). Ultrasonic Module HC-SR04 Distance Measuring Transducer Sensor HC-SR04 โรบอทสยาม [robotsiam.com]
- [6] ThaiWare (2022). การทำงานของเซนเซอร์ Gyroscope บริษัท ไทยแวร์ คอมมิวนิเคชั่น จำกัด Thaiware Communication Co.,Ltd.
- [7] Factomart (2019). เอ็นโค้ดเดอร์ Encoder คืออะไร การทำงานของ Encoder ชนิดต่าง ๆ ของ Encoder และการเลือกซื้อ Encoder บริษัท แฟ็คโตมาร์ท จำกัด
- [8] ชัยวัฒน์ ลิ้มพรจิตวิไล. (2538). คู่มือนักอิเล็กทรอนิกส์. กรุงเทพฯ : บริษัทซีเอ็ดยูเคชั่น จำกัด (มหาชน).
- [9] “UART”, 2562. [ระบบออนไลน์] แหล่งที่มา : <https://th.wikipedia.org/wiki/UART> (24 ตุลาคม 2565).
- [10] “ESP32 เบื้องต้น :: บทที่ 7 การสื่อสารแบบอนุกรม”, 2565. [ระบบออนไลน์] แหล่งที่มา : <https://www.artronshop.co.th/article/57/esp32-เบื้องต้น-บทที่-7-การสื่อสารอนุกรม>
- [11] “ARDUINO MEGA 2560”, 2565. [ระบบออนไลน์] แหล่งที่มา : <https://www.gravitechthai.com/product-detail.php?WP=qmIZAJ1CM5O0hJatrTZo7o3Q>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานรถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้

รถวีลแชร์ที่สามารถโปรแกรมเส้นทางได้จะมีวิธีการใช้งานสำหรับผู้ใช้งานดังต่อไปนี้

ก.1 ผู้ใช้งานจะต้องทำการเลือกพื้นที่เส้นทางเดินของรถวีลแชร์ก่อน ว่าต้องการให้รถวีลแชร์มีเส้นทางการเคลื่อนที่เป็นเช่นไร มีการเคลื่อนที่รุดไปในทางทิศทางไหนบ้าง โดยพื้นที่จะต้องเป็นพื้นที่เรียบ มีความขรุขระน้อย



รูปที่ ก-1 ตัวอย่างพื้นที่เส้นทางการเดินรถวีลแชร์

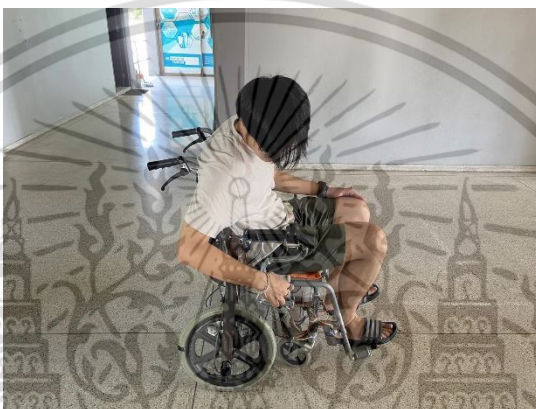


รูปที่ ก-2 ตัวอย่างพื้นที่เส้นทางการเดินรถวีลแชร์ (ต่อ)

ก.2 เมื่อได้เส้นทางการเคลื่อนที่ของรถที่ต้องการแล้ว ก็ทำการนำรถวีลแชร์มาตั้งบนที่เส้นทางที่เลือกเอาไว้ เพื่อทำการเก็บเส้นทางการเคลื่อนที่ให้กับรถวีลแชร์ เมื่อตั้งรถวีลแชร์เสร็จเรียบร้อยแล้ว ก็ทำการเปิดเครื่องเพื่อเริ่มการทำงานของรถวีลแชร์



รูป ก-3 ตั้งรถวีลแชร์บนเส้นทางการเคลื่อนที่



รูป ก-4 การเปิดเครื่องเพื่อเริ่มการทำงานของรถวีลแชร์

ก.3 เมื่อเปิดเครื่อง หน้าจอบนฝั่งซ้ายมือของผู้นั่งจะแสดงคำว่า “Select . . .” เพื่อให้เลือกโหมดการทำงานของรถวีลแชร์ โดยที่ข้าง ๆ หน้าจอมีสวิตช์ให้เลือกการทำงานอยู่ด้วยกัน 3 ปุ่ม ซึ่งทั้ง 3 ปุ่มจะมีหลักการทำงานที่แตกต่างกัน ดังนี้

ปุ่มที่ 1 จะเป็นการเลือกโหมดการทำงานเป็นแบบ Manual หรือใช้การเคลื่อนที่ด้วย Joy-Stick เพื่อใช้ในการเก็บข้อมูลเส้นทางการเคลื่อนที่ที่เลือกเอาไว้ โดยข้อมูลเส้นทางการเคลื่อนที่จะถูกเก็บไว้ใน Micro SD Card

ปุ่มที่ 2 จะเป็นการเลือกโหมดการทำงานเป็น Auto หรือเป็นการเคลื่อนที่แบบอัตโนมัติ โดยการเคลื่อนที่ของตัวรถจะเคลื่อนที่ไปตามข้อมูลที่ถูกบันทึกอยู่ใน Micro SD Card

ปุ่มที่ 3 จะเป็นการเลือกการทำงานให้ลบข้อมูลเส้นทางเก่าที่อยู่ใน Micro SD Card ทั้งหมด เพื่อให้สามารถบันทึกข้อมูลเส้นทางการเคลื่อนที่ใหม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-5 หน้าจอแสดงผล และปุ่มกดเพื่อเลือกการทำงานของรถวิลแชร์

ก.4 เมื่อเปิดเครื่องแล้วหน้าจอแสดงคำว่า “Select . . .” ขั้นตอนแรกต้องทำการกดไปที่ปุ่มที่ 1 เพื่อทำการเก็บเส้นทางเคลื่อนที่ของตัวรถ เมื่อกดเข้าไปหน้าจอจะมีการแสดงคำว่า “Manual Mode” แสดงว่ารถวิลแชร์พร้อมที่จะเก็บเส้นทางแล้ว ก็ให้ทำการเคลื่อนที่ตัวรถออกไปด้วย Joy-Stick ตามเส้นทางเคลื่อนที่ที่ได้เลือกไว้ โดยการบันทึกข้อมูลจะเริ่มขึ้นเมื่อรถมีการเคลื่อนที่ออกไป



รูปที่ ก-6 กดปุ่มเพื่อเลือกโหมดการทำงานของรถวิลแชร์



รูปที่ ก-7 หน้าจอแสดงผลเมื่อกดปุ่มที่ 1 หรือหน้าจอแสดงผลการทำงาน Manual Mode

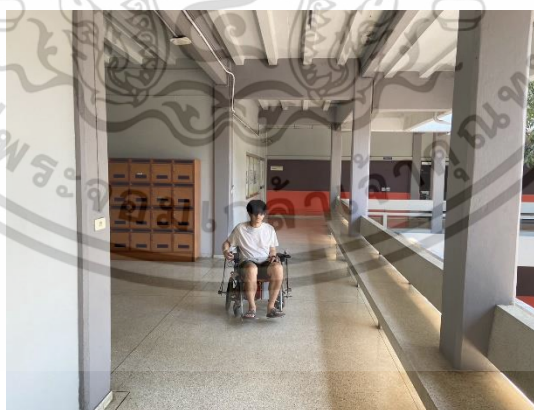
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-8 การเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick เพื่อเก็บเส้นทางการเคลื่อนที่ที่เลือกเอาไว้



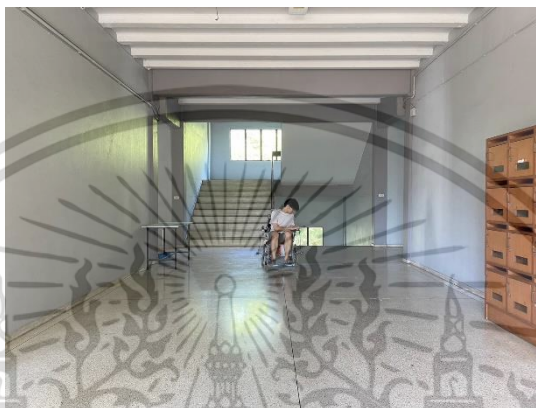
รูปที่ ก-9 การเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick เพื่อเก็บเส้นทางการเคลื่อนที่ที่เลือกเอาไว้ (ต่อ)



รูปที่ ก-10 การเคลื่อนที่รถวีลแชร์ด้วย Joy-Stick เพื่อเก็บเส้นทางการเคลื่อนที่ที่เลือกเอาไว้ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก.5 เมื่อทำการเก็บเส้นทางการเคลื่อนที่เสร็จเรียบร้อยแล้ว ก็ให้ทำการปิดเครื่องแล้วให้นำรถวีลแชร์ไปตั้งไว้ที่จุดเริ่มต้น และให้ทำการเปิดเครื่องเพื่อเริ่มการทำงานอีกครั้ง เมื่อทำการเปิดเครื่องเสร็จแล้วให้ทำการกดเลือกไปปุ่มที่ 2 เพื่อให้รถวีลแชร์เริ่มต้นการทำงานเคลื่อนที่แบบอัตโนมัติ หน้าจอแสดงผลก็จะแสดงผลคำว่า “Auto Mode” หลังจากที่แสดงคำดังกล่าวแล้วรถวีลแชร์ก็จะเริ่มเคลื่อนที่แบบอัตโนมัติตามข้อมูลเส้นทางที่ได้ถูกเก็บไว้ใน Manual Mode



รูปที่ ก-11 กดปุ่มเพื่อเลือกโหมดการทำงานของรถวีลแชร์อีกครั้ง



รูปที่ ก-12 หน้าจอแสดงผลเมื่อกดปุ่มที่ 2 หรือหน้าจอแสดงผลการทำงาน Auto Mode



รูปที่ ก-13 การเคลื่อนที่ที่รถวีลแชร์อัตโนมัติ ตามเส้นทางการเคลื่อนที่ที่บันทึกข้อมูลเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-14 การเคลื่อนที่รถวีลแชร์อัตโนมัติ ตามเส้นทางการเคลื่อนที่ที่บันทึกข้อมูลเอาไว้ (ต่อ)

ก.6 เมื่อต้องการเส้นทางการรถในเส้นทางใหม่ให้กดปุ่มที่ 3 เพื่อทำการลบข้อมูลเส้นทางการเคลื่อนที่เดิมของรถออก และเมื่อต้องการบันทึกข้อมูลเส้นทางการเคลื่อนที่ใหม่ก็ให้ทำการกดปุ่มที่ 1 เพื่อเลือกการทำงานเป็นแบบ Manual Mode และทำการเก็บข้อมูลเส้นทางการเคลื่อนที่ที่ต้องการด้วยการเคลื่อนที่โดยใช้ Joy-Stick เมื่อเก็บข้อมูลเส้นทางการเคลื่อนที่เรียบร้อยแล้วก็ให้ตามข้อ ก.5 อีกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code ของบอร์ดรับข้อมูล (Arduino Mega2560)

```

#include <Wire.h>
#include <NewPing.h>
#include <SoftwareSerial.h>
#include <Adafruit_PWMServoDriver.h>
Adafruit_PWMServoDriver pwm = Adafruit_PWMServoDriver();
#include <SD.h>
#include <SPI.h>
SoftwareSerial UnoSerial(11,10);
//encoder
#define ENC_COUNT_REV 4000
#define ENC_IN1 2
#define ENC_IN2 3
#define ENC_IN3 19
#define ENC_IN4 18
#define Switch1 4
#define Switch2 5
#define SERVOMIN 150 // This is the 'minimum' pulse length count (out of 4096)
#define SERVOMAX 600 // This is the 'maximum' pulse length count (out of 4096)
#define USMIN 600 // This is the rounded 'minimum' microsecond length based on the
minimum pulse of 150
#define USMAX 2400 // This is the rounded 'maximum' microsecond length based on the
maximum pulse of 600
#define SERVO_FREQ 50 // Analog servos run at ~50 Hz updates
#define TRIG1 26
#define ECHO1 27
#define TRIG2 28
#define ECHO2 29
#define TRIG3 30
#define ECHO3 31
#define TRIG4 32
#define ECHO4 33
#define MAX_DISTANCE 200
uint8_t servonum = 8;
uint8_t servonum1 = 10;
uint8_t servonum2 = 12;
uint8_t servonum3 = 14;
bool Manual = false;
bool Autoo = false;
//value of encoder
volatile long encoderValue1 = 0;
volatile long encoderValue2 = 0;
int rpm1 = 0;
int rpm2 = 0;
unsigned int dis1,dis2,dis3,dis4;
int Dis1,Dis2,Dis3,Dis4;
unsigned long period = 50;
unsigned long per1 = 1000;
unsigned long per2 = 50;
unsigned long last_time = 0;
unsigned long endTime = 0;
unsigned long endTime1 = 0;
unsigned long period2 = 25;
unsigned long starter = millis();
static int state = 0;

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static int index = 0;
static char Buffer[20];
static int X;
String datalog = "";
//SD Card
int chipSelect = 53;
File file;
String data;
int cnt = 0;
String line;
NewPing sonar1(TRIG1,ECHO1,MAX_DISTANCE);
NewPing sonar2(TRIG2,ECHO2,MAX_DISTANCE);
NewPing sonar3(TRIG3,ECHO3,MAX_DISTANCE);
NewPing sonar4(TRIG4,ECHO4,MAX_DISTANCE);
void setup() {
  Serial.begin(9600);
  UnoSerial.begin(9600);
  Wire.begin();
  pwm.begin();
  pinMode(ENC_IN1, INPUT_PULLUP);
  pinMode(ENC_IN2, INPUT_PULLUP);
  pinMode(ENC_IN3, INPUT_PULLUP);
  pinMode(ENC_IN4, INPUT_PULLUP);
  pinMode(Switch1, INPUT);
  pinMode(Switch2, INPUT);
  pwm.setOscillatorFrequency(27000000);
  pwm.setPWMFreq(SERVO_FREQ); // Analog servos run at ~50 Hz updates
  attachInterrupt(digitalPinToInterrupt(ENC_IN1), updateEncoder1, RISING);
  attachInterrupt(digitalPinToInterrupt(ENC_IN2), updateEncoder2, RISING);
  attachInterrupt(digitalPinToInterrupt(ENC_IN3), updateEncoder3, RISING);
  attachInterrupt(digitalPinToInterrupt(ENC_IN4), updateEncoder4, RISING);
  Serial.println("Install SD Card. . .");
  if(!SD.begin(chipSelect)){///////// ;
    Serial.println("Install SD Card Failed");
    return;
  }
  Serial.println("Install Successful");
  Serial.println("Creating File. . .");
  file = SD.open("data.txt",FILE_WRITE);
  file.close();
}
void loop() {
  if (!digitalRead(4)){
    Manual =! Manual;
    while(!digitalRead(4));
  }
  if (!digitalRead(5)){
    Autoo =! Autoo;
    while(!digitalRead(5));
  }
  if(Manual == true){
    recive_1();
    SavEDa_1();
  }
  if(Autoo == true){
    Sends_1();
  }
}

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    Encode_1();
    servo_1();
}

void recive_1(){
    if(UnoSerial.available(>0){
        char data = UnoSerial.read();
        switch(state){
            case 0: if(data == '#'){
                index = 0;
                state = 1;
            }
            break;
            case 1: if(data == ';'){
                Buffer[index] = '\0';
                X = atoi(Buffer);
                state = 0;
                // Serial.println(X);
            }else{
                Buffer[index] = data;
                index++;
            }
            break;
        }
    }
}

void Encode_1(){
    if(millis()- last_time > period){
        last_time += period;
        rpm1 = (float)(encoderValue1 * 60 / ENC_COUNT_REV);
        rpm2 = (float)(encoderValue2 * 60 / ENC_COUNT_REV);
        // Only update display when there is a reading
        if ( rpm1 > 0, rpm2 > 0){
            // Serial.print(rpm1);
            // Serial.println(rpm2);
        }
        // encoderValue1 = 0;
        // encoderValue2 = 0;
    }
}

void updateEncoder1(){
    if(digitalRead(ENC_IN2)==LOW){
        encoderValue1++;
    }else{
        encoderValue1--;
    }
}

void updateEncoder2(){
    if(digitalRead(ENC_IN1)==LOW){
        encoderValue1--;
    }else{
        encoderValue1++;
    }
}

void updateEncoder3(){

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(digitalRead(ENC_IN4)==LOW){
  encoderValue2--;
}else{
  encoderValue2++;
}
}
void updateEncoder4(){
  if(digitalRead(ENC_IN3)==LOW){
    encoderValue2++;
  }else{
    encoderValue2--;
  }
}
void servo_1(){
  if(millis()- endTime > per1){
    endTime += per1;
    for (uint16_t pulselen = SERVOMIN; pulselen < SERVOMAX; pulselen++) {
      pwm.setPWM(servonum, 0, pulselen);
      pwm.setPWM(servonum1, 0,750-pulselen);
      // pwm.setPWM(servonum2, 0, pulselen);
      //pwm.setPWM(servonum3, 0, 750-pulselen);
      dis1 = sonar1.ping();
      Dis1 = dis1/US_ROUNDTRIP_CM;
      dis2 = sonar2.ping();
      Dis2 = dis2/US_ROUNDTRIP_CM;
      dis3 = sonar3.ping();
      Dis3 = dis3/US_ROUNDTRIP_CM;
      dis4 = sonar4.ping();
      Dis4 = dis4 / US_ROUNDTRIP_CM;
      pulselen++;
      pulselen++;
      pulselen++;
      pulselen++;
      pulselen++;
    }
    for (uint16_t pulselen = SERVOMAX; pulselen > SERVOMIN; pulselen--) {
      pwm.setPWM(servonum, 0, 750-pulselen);
      pwm.setPWM(servonum1, 0,pulselen);
      // pwm.setPWM(servonum2, 0, 750-pulselen);
      // pwm.setPWM(servonum3, 0, pulselen);
      dis1 = sonar1.ping();
      Dis1 = dis1/US_ROUNDTRIP_CM;
      dis2 = sonar2.ping();
      Dis2 = dis2/US_ROUNDTRIP_CM;
      dis3 = sonar3.ping();
      Dis3 = dis3/US_ROUNDTRIP_CM;
      dis4 = sonar4.ping();
      Dis4 = dis4 / US_ROUNDTRIP_CM;
      pulselen--;
      pulselen--;
      pulselen--;
      pulselen--;
      pulselen--;
    }
  }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void setServoPulse(uint8_t n, double pulse) {
  double pulselength;
  pulselength = 1000000; // 1,000,000 us per second
  pulselength /= SERVO_FREQ; // Analog servos run at ~60 Hz updates
  Serial.print(pulselength); Serial.println(" us per period");
  pulselength /= 4096; // 12 bits of resolution
  Serial.print(pulselength); Serial.println(" us per bit");
  pulse *= 1000000; // convert input seconds to us
  pulse /= pulselength;
  Serial.println(pulse);
  pwm.setPWM(n, 0, pulse);
}

void Sends_1(){
  file = SD.open("data.txt");
  if(file){
    while(file.available())
    {
      data += (char)file.read();
      //data = file.readString();
    }
    while(cnt != -1){
      if(millis()- last_time >= period2){
        last_time += period2;
        int newlineIndex = data.indexOf("\n",cnt);
        line = data.substring(cnt,newlineIndex);
        cnt = newlineIndex +1;
        //Serial.println(line);
        String sub_S = line.substring(0,1 );
        UnoSerial.print('#');
        UnoSerial.print(sub_S);
        UnoSerial.println(';');
      }
    }
    file.close();
  }
}

void SavEDa_1(){
  unsigned long currentTime = millis();
  datalog = String(X);
  if(rpm1 > 0 && rpm2 >0 ){
    if(currentTime %25 == 0){
      if (SD.exists("data.txt")){
        file = SD.open("data.txt",FILE_WRITE);
        if(file){
          Serial.println("Data save");
          file.println(datalog);
          file.close();
        }else{
          Serial.println("Can't write");
        }
      }else{
        Serial.println("File doesn't exist");
      }
    }
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Code ของบอร์ดขับเคลื่อน (Arduino Nano)

```

#include <Wire.h>
#include <Bounce2.h>
#define NUM_BUTTONS 1
const int Switch_1_Sdata = 2;
const uint8_t BUTTON_PINS[NUM_BUTTONS] = {17}; //กำหนดพินปุ่ม ถ้าหลายปุ่มคั่นด้วย comma ,
Bounce * buttons = new Bounce[NUM_BUTTONS];
static int state = 0;
static int index = 0;
static char Buffer[20];
static int X;
unsigned long time1sec = 1000; // เวลาวินาที
unsigned long last_time = 0;
unsigned char F_motor_L , F_motor_R , R_motor_L , R_motor_R;
int Gyro_X , Gyro_Y , Gyro_Max;
int mathF_motor_L , mathF_motor_R , mathR_motor_L , mathR_motor_R;
unsigned int VR_speed,Map_Speed,Cout_Speed;
int Speed;
int sends;
bool button_state;
int Statework = false;
bool Manual = true;
bool Autoo = false;
#define PWM_F_motor_L 9
#define PWM_F_motor_R 11
#define PWM_R_motor_L 10
#define PWM_R_motor_R 3
#define Enable_motor_R 5
#define Enable_motor_L 6
#define For_light 8
#define Rev_light 7
#define Left_light 4
#define Right_light 12
#define Switch_1_Sdata 2
#define Switch_2_Auto 13
int joystick_X , joystick_Y , xPosition , yPosition , joystick_Y_positive , joystick_Y_negative;
byte joystick_X_W , joystick_Y_W , Swicht_D1;
void setup() {
  Wire.begin(1);
  Serial.begin(9600);
  pinMode(Enable_motor_R, OUTPUT);
  pinMode(Enable_motor_L, OUTPUT);
  pinMode(PWM_F_motor_L,OUTPUT);
  pinMode(PWM_F_motor_R,OUTPUT);
  pinMode(PWM_R_motor_L,OUTPUT);
  pinMode(PWM_R_motor_R,OUTPUT);
  pinMode(For_light, OUTPUT);
  pinMode(Rev_light, OUTPUT);
  pinMode(Left_light, OUTPUT);
  pinMode(Right_light, OUTPUT);
  pinMode(Switch_1_Sdata,INPUT);
  pinMode(Switch_2_Auto,INPUT);
  TCCR1B = (TCCR1B & 0b11111000) | 0x01;
  TCCR2B = (TCCR2B & 0b11111000) | 0x01;
  for (int i = 0; i < NUM_BUTTONS; i++) {
    buttons[i].attach( BUTTON_PINS[i] ,INPUT);
    buttons[i].interval(50);
  }
}

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (!digitalRead(2)){
  Manual = !Manual;
  digitalWrite(LED,Manual);
  while(!digitalRead(2)){
  }

  Switches_1();

  if(Manual == true){
    Value_1();
    if(Swicht_D1 == 0){joystick_anlog();}
    if(joystick_X > 20){forward();digitalWrite(For_light, HIGH);Cout_Speed = 1;sends = 1;}
    else {digitalWrite(For_light, LOW);}
    if(joystick_X < -20){revwerd();digitalWrite(Rev_light, HIGH);Cout_Speed = 1;}
    else {digitalWrite(Rev_light, LOW);}
    if(joystick_Y > 20 && joystick_X < 19 && joystick_X > -19){Left();digitalWrite(Left_light,
HIGH);Cout_Speed = 1;sends = 2;}
    else {digitalWrite(Left_light, LOW);}
    if(joystick_Y < -20 && joystick_X < 19 && joystick_X > -19){Right();digitalWrite(Right_light,
HIGH);Cout_Speed = 1;sends = 3;}
    else {digitalWrite(Right_light, LOW);}
    if(joystick_X < 19 && joystick_X > -19 && joystick_Y < 19 && joystick_Y > -19){
      sends = 0;
      Cout_Speed = 0;
      Speed = 0;
      if (mathF_motor_L > 0) {mathF_motor_L = mathF_motor_L-29;}
      else {mathF_motor_L = 0;}
      if (mathF_motor_R > 0) {mathF_motor_R = mathF_motor_R-29;}
      else {mathF_motor_R = 0;}
      if (mathR_motor_L > 0) {mathR_motor_L = mathR_motor_L-29;}
      else {mathR_motor_L = 0;}
      if (mathR_motor_R > 0) {mathR_motor_R = mathR_motor_R-29;}
      else {mathR_motor_R = 0;}
    }
    Gyro_X = analogRead(A3);
    Gyro_Y = analogRead(A6);
    if( millis() - last_time > time1sec) {
      if (Gyro_X < 325 || Gyro_Y < 330 || Gyro_Y > 341){Gyro_Max = 100;last_time = millis();}
      if (Gyro_X > 339){Gyro_Max = 50;last_time = millis();}
      if (Gyro_Max != 100 && Gyro_Max != 50){Gyro_Max = 0;last_time = millis();}
      VR_speed = analogRead(A2);
      if (Gyro_Max == 100){Map_Speed = 100;}
      if (Gyro_Max == 50){Map_Speed = 155;}
      else {Map_Speed = map(VR_speed,0,1023,0,254);}
      if (Cout_Speed == 1 && Map_Speed > Speed && Speed < 245){Speed = Speed+10;}
      if (Cout_Speed == 1){digitalWrite(Enable_motor_R, HIGH);digitalWrite(Enable_motor_L,
HIGH);}
      else {digitalWrite(Enable_motor_R, LOW);digitalWrite(Enable_motor_L, LOW);}
      if(joystick_Y > 0){joystick_Y_positive = joystick_Y;}
      else {joystick_Y_positive = 0;}
      if(joystick_Y < 0){joystick_Y_negative = joystick_Y;}
      else {joystick_Y_negative = 0;}
      if (mathF_motor_L > 0){F_motor_L = mathF_motor_L;}
      else {F_motor_L = 0;}
      if (mathF_motor_R > 0){F_motor_R = mathF_motor_R;}
      else {F_motor_R = 0;}
      if (mathR_motor_L > 0){R_motor_L = mathR_motor_L;}
      else {R_motor_L = 0;}
      if (mathR_motor_R > 0){R_motor_R = mathR_motor_R;}
      else {R_motor_R = 0;}
      analogWrite(PWM_F_motor_L,F_motor_L);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    analogWrite(PWM_F_motor_R,F_motor_R);
    analogWrite(PWM_R_motor_L,R_motor_L);
    analogWrite(PWM_R_motor_R,R_motor_R);
    if(joystick_X < 19 || joystick_X > -19 || joystick_Y < 19 || joystick_Y > -19){
      Value_1();
    }
  }
}
if(Statework == true){
  recive_1();
  if (X == 1){
    forwerd2();
  }
  if(X==2){
    Left2();
  }
  if(X==3){
    Right2();
  }
  if(X==0){
    Stopp();
  }
  analogWrite(PWM_F_motor_L,mathF_motor_L);
  analogWrite(PWM_F_motor_R,mathF_motor_R);
  analogWrite(PWM_R_motor_L,mathR_motor_L);
  analogWrite(PWM_R_motor_R,mathR_motor_R);
}
}
void joystick_anlog(){
  xPosition = analogRead(A0);
  yPosition = analogRead(A1);
  joystick_X = map(xPosition,0,1023,-100,100);
  joystick_Y = map(yPosition,0,1023,-200,200);
}
}
void forwerd(){
  mathF_motor_L = 0;
  mathF_motor_R = 0;
  mathR_motor_L = Speed-joystick_Y_positive;
  mathR_motor_R = Speed+joystick_Y_negative;
}
void revwerd(){
  mathF_motor_L = Speed-joystick_Y_positive;
  mathF_motor_R = Speed+joystick_Y_negative;
  mathR_motor_L = 0;
  mathR_motor_R = 0;
}
}
void Left(){
  mathF_motor_L = 0;
  mathF_motor_R = Speed;
  mathR_motor_L = 0;
  mathR_motor_R = 0;
}
}
void Right(){
  mathF_motor_L = Speed;
  mathF_motor_R = 0;
  mathR_motor_L = 0;
  mathR_motor_R = 0;
}
}
void forwerd2(){
  mathF_motor_L = 0;
  mathF_motor_R = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mathR_motor_L = 165;
mathR_motor_R = 164;
}
void Left2(){
  mathF_motor_L = 0;
  mathF_motor_R = 164;
  mathR_motor_L = 0;
  mathR_motor_R = 0;
}
void Right2(){
  mathF_motor_L = 165;
  mathF_motor_R = 0;
  mathR_motor_L = 0;
  mathR_motor_R = 0;
}
void Stopp(){
  mathF_motor_L = 0;
  mathF_motor_R = 0;
  mathR_motor_L = 0;
  mathR_motor_R = 0;
}

void Value_1(){
  static unsigned long timer = millis();
  const unsigned long period = 50;
  if(millis()- timer >= period){
    timer += period;
    Serial.print('#');
    Serial.print(sends);
    Serial.print(';');
    Serial.println();
  }
}

void recive_1(){
  if(Serial.available()>0){
    digitalWrite(Enable_motor_R, HIGH);digitalWrite(Enable_motor_L, HIGH);
    char data = Serial.read();
    switch(state){
      case 0: if(data == '#'){
        index = 0;
        state = 1;
      }
      break;
      case 1: if(data == ';'){
        Buffer[index] = '\0';
        X = atoi(Buffer);
        Serial.println(X);
        state = 0;
      }else{
        Buffer[index] = data;
        index++;
      }
      break;
    }
  }
}

void Switches_1(){
  for (int i = 0; i < NUM_BUTTONS; i++) {
    buttons[i].update();
    button_state = buttons[i].read();
    if(Autoo > button_state) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Statework = !Statework;  
digitalWrite(LED2,Statework);  
}  
if(Autoo < button_state) {  
}  
Autoo = button_state; //บันทึกค่าสถานะไว้ใช้รอบถัดไป  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้