การศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติ

A Study Of Automatic Control System For Intelligent Wheelchair

ญาณภัทร ดียิ่ง

ณชา เอื้อบกงไชย

ธิม ทองมา

สรยุทธ ศรีเมือง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A Study Of Automatic Control System For Intelligent Wheelchair

YANAPAT DEEYING

NACHA IABKONGCHAI

TIM THONGMA

SORAYUT SRIMAUNG

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 INSTITUE OF TECHNOLOGY LARDKRABANG

ACADEMIC YEAR 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2565

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ การศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติ

A Study Of Automatic Control System For Intelligent Wheelchair

นักศึกษาผู้จัดทำ	นายญาณภัทร	ดียิ่ง	**รหัสนักศึกษา**	63015043
	นางสาวณชา	เอี๊ยบกงไชย	**รหัสนักศึกษา**	63015045
	นายธิม	ทองมา	**รหัสนักศึกษา**	63015084
	นายสรยุทธ	ศรีเมือง	**รหัสนักศึกษา**	63015179

ปริญญา วิศวกรรมศาสตร์บัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม(ต่อเนื่อง)

ปีการศึกษา 2565

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
ผู้ช่วยศาสตราจารย์ ดร.นภศูล วงษ์วานิช	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติ
A STUDY OF AUTOMATIC CONTROL SYSTEM FOR INTELLIGENT WHEELCHAIR

นักศึกษาผู้จัดทำ	นายญาณภัทร	ดียิ่ง	**รหัสนักศึกษา**	63015043
	นางสาวณชา	เอี๊ยบกงไชย	**รหัสนักศึกษา**	63015045
	นายธิม	ทองมา	**รหัสนักศึกษา**	63015084
	นายสรยุทธ	ศรีเมือง	**รหัสนักศึกษา**	63015179
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตร์ตราจารย์ ดร.นภศูล วงษ์วานิช			

ปีการศึกษา 2565

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษา เพื่อพัฒนาการศึกษาค้นคว้าและพัฒนารถเข็นผู้พิการ และผู้สูงอายุจากแนวคิดยานยนต์ขับเคลื่อนด้วยมอเตอร์โดยใช้พลังงานไฟฟ้า ซึ่งใช้สำหรับผู้พิการทางร่างกายและผู้สูงอายุผู้ที่ลำบากในการเดินหรือการเคลื่อนที่ เช่น ผู้สูงอายุ ผู้ที่อยู่ระหว่างการรักษาตัว ให้มีประสิทธิภาพทางร่างกายเพิ่มความสะดวกสบายมากยิ่งขึ้นและในด้านความปลอดภัยของผู้ใช้งาน ในการศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติประกอบไปด้วยระบบต่าง ๆ ได้แก่ ระบบการควบคุม การขับเคลื่อน และการเก็บค่าของรอบของมอเตอร์ในการควบคุม มอเตอร์กระแสตรง ให้มีความคงที่สามารถใช้งานจริงให้ได้ประสิทธิภาพการใช้งานของแต่ละอุปกรณ์ให้มากที่สุด ซึ่งสามารถควบคุมผ่านอัตโนมัติได้ สามารถทำให้รถเข็นผู้พิการสามารถเดินในรูปแบบต่าง ๆ ที่กำหนดและนำการออกแบบไปติดตั้งอุปกรณ์เสริมอื่น ๆ ให้มีประสิทธิภาพ โดยจัดทำเพื่อศึกษาการทำงานของมอเตอร์โดยใช้หลักการไฟฟ้าและออกแบบโปรแกรมโดยใช้อุปกรณ์คอนโทรลเลอร์ในการควบคุม เพื่อนำให้ได้ความรู้ไปต่อยอดมากขึ้นและยังมีประโยชน์ต่อผู้พิการในอนาคตได้ โดยโครงนี้จะทำการศึกษาเพิ่มเติมเกี่ยวกับการควบคุมแต่จะคำนึงถึงความปลอดภัยของผู้ใช้งานเป็นหลักสำคัญที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	A Study Of Automatic Control System For Intelligent Wheelchair	
Austhors	Mr.Yanapat	Deeying
	Miss.Nacha	Iabkongchai
	Mr.Tim	Thongma
	Mr.Sorayut	Srimuang
Thesis Advisor	Asst.Prof.Dr.Napasool Wongvanich	
Year	2022	

ABSTRACT

This thesis aims to develop and study electric motor-driven wheelchairs for disabled and elderly individuals. These wheelchairs are intended for individuals with physical disabilities and elderly people who have difficulty walking or moving, such as those undergoing medical treatment. To improve physical effectiveness, increase comfort, and prioritize user safety, the study of automatic intelligent wheelchair control includes various systems, such as a control system for a DC motor and a control device programmed using the Arduino IDE to drive the motor. The stability, practicality, and efficiency of the devices are important considerations in the design of the automatic control system, which allows the wheelchair to move in a specific pattern. The control system is designed and installed using the Arduino IDE, allowing for the modification of the speed of both motors on either side and the installation of an encoder rotary counting device to store values in a memory module card. The project also involves studying the operation of the motor using electrical principles and designing a program using a controller to control it. Future research will focus on controls while prioritizing user safety.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเนื่องด้วยการช่วยเหลือจากบุคคลหลายท่าน โดยเฉพาะอย่างยิ่งผู้มีพระคุณท่านแรก คณะผู้จัดทำต้องขอขอบคุณ ผศ.ดร.นภศูล วงษ์วานิช อาจารย์ที่ปรึกษา ที่ได้ให้ความรู้ ให้คำแนะนำในการปรับปรุงแก้ไขปัญหาต่างๆ รวมถึงสนับสนุนงบประมาณ สำหรับอุปกรณ์ในการจัดทำโครงงาน และได้กรุณาตรวจทานความเรียบร้อยของปริญญานิพนธ์ ด้วยความเอาใจใส่ทุกขั้นตอนจนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกๆ ท่าน ที่ได้อบรมสั่งสอน ให้ความรู้ในด้านต่างๆ คณะกรรมการสอบปริญญานิพนธ์ที่ได้ให้คำแนะนำแก้ไขข้อบกพร่อง ตลอดการจัดทำที่ผ่านมา

ขอขอบพระคุณบิดา มารดา และครอบครัวของผู้จัดทำ รวมถึงขอขอบคุณเพื่อนๆ ของผู้จัดทำทุกท่าน ที่ได้คอยให้คำปรึกษาแลกเปลี่ยนความรู้ ช่วยเหลือสนับสนุนในทุกๆ ด้านจนโครงงานสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณผู้เขียนหนังสือเอกสารอ้างอิงและเว็บไซต์ต่างๆ ที่คณะผู้จัดทำได้นำมาใช้อ้างอิงประกอบการศึกษา จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงโดยสมบูรณ์ที่สุด

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ VI

สารบัญ (ต่อ) VII

สารบัญตาราง VIII

สารบัญรูป VIIII

บทที่ 1 บทนำ 1

1.1 ความสำคัญของปริญญานิพนธ์ 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1

1.3 ขอบเขตของปริญญานิพนธ์ 1

1.4 ขั้นตอนการศึกษา 2

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 3

2.1 มอเตอร์ไฟฟ้ากระแสตรง (Direct current motor) 3

2.1.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง (Direct current motor) 3

2.1.2 องค์ประกอบหลักของมอเตอร์ไฟฟ้ากระแสตรง 4

2.1.3 การขับเคลื่อด้วยมอเตอร์ไฟฟ้า 4

2.2 รถเข็นผู้พิการ 5

2.2.1 อุปกรณ์มาตรฐานรถเข็นผู้พิการ 5

2.3 Arduino Nano 7

2.4 Arduino Mega 2560 8

2.5 เอ็นโค้ดเดอร์ (Encoder Omron E6B2-CWZ6C) 9

2.6 แบตเตอรี่ (Battery) 11

2.6.1 ชนิดของแบตเตอรี่ แบตเตอรี่แบ่งออกเป็น 2 ประเภท 11

2.6.1.1 ชนิดแห้ง (Dry Cell) 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6.1.2 ชนิดน้ำ (Wet Cell) 12
2.6.2 การทำงานของแบตเตอรี่ 13
2.6.3 ความสามารถในการจัดเก็บพลังงาน 13
2.6.4 ข้อควรระวังเกี่ยวกับแบตเตอรี่ 14
2.7 GY-521 Module MPU6050 14
2.8 Pulse Width Modulation (PWM) 15
2.8.1 ตัวอย่างสัญญาณ (PWM) 14
2.9 ระบบควบคุม (Control System) 16
2.10 สายไฟฟ้าแรงดันต่ำ (Low Voltage Power Cable) 17
2.11 สวิตช์ไฟฟ้า (Switch) 18
2.12 DC Motor Drive H-Bridge SE-HB40-1 19

บทที่ 3 วิธีดำเนินการ 21
3.1 การออกแบบการทำงานของระบบ 21
3.1.1 การทำงานของระบบควบคุมการขับเคลื่อน 21
3.2 วิธีดำเนินการ 26
3.2.1 ศึกษาหลักการทำงาน 26
3.2.2 ออกแบบระบบการทำงานของวีลแชร์ 26
3.2.3 จัดซื้ออุปกรณ์ 26
3.2.4 ขั้นตอนการติดตั้ง 27
3.2.5 การเขียนโปรแกรมในรูปแบบต่างๆ 29
3.2.5.1 ขั้นตอนการทดลอง Encoder 29
3.2.5.2 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบสี่เหลี่ยม 30
3.2.5.3 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบวงกลม 30
3.2.5.4 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบเลขแปด 31
3.2.6 ขั้นตอนการทดลอง GY-521 Module MPU6050 31
3.2.6.1 ขั้นตอนการทดลอง GY-521 Module MPU6050 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.7 ขั้นตอนการทดลอง Module SD Card 32

บทที่ 4 ผลการทดลอง 34

4.1 กล่าวนำ 34

4.2 ผลการทดลอง 34

4.2.1 การทำงานของระบบนับจำนวนรอบ Encoder 34

4.2.2 การทำงานของ GY-521 35

4.2.3 การขับเคลื่อนตามคำสั่งให้เดินอัตโนมัติ 35

บทที่ 5 สรุปผลการดำเนินงานและข้อเสนอแนะ 43

5.1 สรุปผลการดำเนินงาน 43

5.2 ปัญหาที่พบและแนวทางการแก้ไข 43

5.2.1 ปัญหาที่พบขณะทำการทดลอง 43

5.2.2 แนวทางการแก้ไข 43

5.3 ข้อเสนอแนะ 44

บรรณานุกรม 45

ภาคผนวก ก 46

ภาคผนวก ข 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่..... หน้า

4.1 ตารางผลการทดลองการทำงานของระบบนับจำนวนรอบ Encoder 34

4.2 ตารางผลการทดลองการทำงานของ GY-521 35

4.3 ตารางผลการทดลองการขับเคลื่อนรถวีลแชร์เป็นสี่เหลี่ยมเมื่อทำการออกคำสั่ง.....35

4.4 ตารางผลการทดลองการขับเคลื่อนรถวีลแชร์เป็นวงกลมเมื่อทำการออกคำสั่ง 38

4.5 ตารางผลการทดลองการขับเคลื่อนรถวีลแชร์เป็นเลขแปดเมื่อทำการออกคำสั่ง.....40

4.6 ตารางผลการทดลองการเกิดค่าความผิดพลาดในการทำงานขับเคลื่อนรถวีลแชร์เป็นเลขแปดเมื่อทำการออกคำสั่งการวนหลายรอบ 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ ..... หน้า

2.1 มอเตอร์ไฟฟ้ากระแสตรง 3

2.2 Board Arduino Nano 7

2.3 Arduino Mega 2560 8

2.4 ส่วนประกอบของเอ็นโค้ดเดอร์ 10

2.5 Encoder Omron E6B2-CWZ6 10

2.6 แบตเตอรี่ (Battery) 12

2.7 แสดงการต่อแบบขนาน 12

2.8 รูปแสดงการต่อแบบอนุกรม 12

2.9 แสดงการต่อแบบอนุกรมกับแบบขนาน 12

2.10 Module MPU 6050 14

2.11 แสดงสัญญาณเปิด(HIGH) และปิด(LOW) 15

2.12 ตัวอย่างสัญญาณ (PWM) 15

2.13 สายไฟฟ้าแรงดันต่ำ (Low Voltage Power Cable) 17

2.14 วงจรสวิตช์ไฟฟ้า (Switch) 18

2.15. สวิตช์ไฟฟ้า (Switch) 19

2.16 DC Motor Drive H-Bridge SE-HB40-1 19

3.1 ขั้นตอนการออกแบบการทำงานของระบบ 21

3.2 ผังการแสดงการทำงานของระบบขับเคลื่อน 22

3.3 วงจรควบคุมด้วยจอยสติ๊ก 23

3.4 Flow Chart แสดงการทำงานระบบควบคุมการขับเคลื่อน 24

3.5 Flow Chart การทำงาน 25

3.6 จัดซื้อ Arduino Mega 2560 26

3.7 จัดซื้อ Encoder E6B2-CWZ6C 26

3.8 จัดซื้อ GY-521 หรือ Gyro Module 6050 27

3.9 เชื่อมเหล็กติดกับตัวโครงรถวีลแชร์ 27

3.10 เจาะรูต๊าปเกลียวที่ตัวเอ็นโค้ดเดอร์ 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ ..... หน้า
3.11 ติดตั้งเอ็นโค้ดเดอร์ที่ตัวรถวีลแชร์ 28
3.12 ทดลองต่อสายจากตัวเอ็นเข้าไปยัง Arduino Mega 2560 29
3.13 โปรแกรมการทำงานของ Encoder 29
3.14 โปรแกรมควบคุมแบบอัตโนมัติในรูปแบบสี่เหลี่ยม 30
3.15 เขียนโปรแกรมควบคุมแบบอัตโนมัติรูปแบบวงกลม 30
3.16 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบเลขแปด 31
3.17 ทดลอง GY-521 Module MPU6050 31
3.18 โปรแกรมควบคุมรับ-ส่งค่าจาก GY-521 Module MPU6050 เพื่อไปควบคุมตัวรถวีลแชร์ 32
3.19 Module SD Card 32
3.20 ผลการทดลองจากการเขียนโปรแกรมเพื่อเก็บค่ามาวิเคราะห์ 33
3.21 การทดลองจากการเขียนโปรแกรมให้รถเคลื่อนที่อัตโนมัติ 33
4.1 รูปแบบสี่เหลี่ยม2X2เมตร จุด A 36
4.2 รูปแบบสี่เหลี่ยม2X2เมตร จุด B 36
4.3 รูปแบบสี่เหลี่ยม2X2เมตร จุด C 37
4.4 รูปแบบสี่เหลี่ยม2X2เมตร จุด D 37
4.5 รูปแบบวงกลมส่วนโค้ง1 38
4.6 รูปแบบวงกลมส่วนโค้ง2 39
4.7 ระยะความกว้างของรูปแบบวงกลม 39
4.8 รูปแบบเลขแปด 1 40
4.9 รูปแบบเลขแปด 2 40
4.10 รูปแบบเลขแปด 3 41
4.11 ระยะความกว้างของรูปแบบเลขแปด 41
ก.1 ต่อสายจาก Arduino เข้า drive ขับมอเตอร์ทั้งสองฝั่ง 47
ก.2 ชุดเก็บข้อมูล Encoder และ GY-521 เข้า SD card 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ปัจจุบันประเทศไทยได้มีการพัฒนาการศึกษาเกี่ยวกับการดูแลผู้พิการหรือผู้สูงอายุนี้มีโดยคณะผู้จัดทำได้ตั้งวัตถุประสงค์เพื่อพัฒนาและศึกษารถเข็นวีลแชร์ที่ขับเคลื่อนด้วยมอเตอร์ไฟฟ้าสำหรับผู้พิการและผู้สูงอายุ รถเข็นเหล่านี้ออกแบบมาสำหรับบุคคลที่มีความพิการทางร่างกายและผู้สูงอายุที่มีปัญหาในการเดินหรือเคลื่อนไหว เช่น ผู้ที่เข้ารับการรักษาทางการแพทย์ เพื่อปรับปรุงประสิทธิภาพทางกายภาพ เพิ่มความสะดวกสบาย และจัดลำดับความสำคัญของความปลอดภัยของผู้ใช้ การศึกษาระบบควบคุมรถเข็นอัจฉริยะอัตโนมัติประกอบด้วยระบบต่างๆ เช่น ระบบควบคุมสำหรับมอเตอร์ไฟฟ้ากระแสตรงและอุปกรณ์ควบคุมที่ตั้งโปรแกรมโดยใช้ Arduino IDE เพื่อขับเคลื่อนมอเตอร์เสถียรภาพ การใช้งานจริง และประสิทธิภาพของอุปกรณ์เหล่านี้เป็นข้อพิจารณาที่สำคัญในการออกแบบระบบควบคุมอัตโนมัติ ซึ่งช่วยให้รถเข็นสามารถเคลื่อนที่ในรูปแบบเฉพาะได้ ระบบควบคุมได้รับการออกแบบและติดตั้งโดยใช้ Arduino IDE ช่วยให้สามารถปรับเปลี่ยนความเร็วของมอเตอร์ทั้งสองด้านใดด้านหนึ่งได้ และติดตั้งอุปกรณ์นับแบบโรตารี่ของเอ็นโค้ดเดอร์เพื่อจัดเก็บค่าในการ์ดโมดูลหน่วยความจำ โครงงานนี้ยังเกี่ยวข้องกับการศึกษาการทำงานของมอเตอร์โดยใช้หลักการทางไฟฟ้าและการออกแบบโปรแกรมโดยใช้คอนโทรลเลอร์เพื่อควบคุม การวิจัยในอนาคตจะมุ่งเน้นไปที่การควบคุมในขณะที่ให้ความสำคัญกับความปลอดภัยของผู้ใช้

จากปัญหาข้างต้น ทางคณะผู้จัดทำจึงได้จัดทำโครงการ การศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติ ขึ้นมาเพื่อทำการศึกษาการควบคุมวีลแชร์ให้สามารถใช้งานได้อัตโนมัติโดยคำนึงถึงหลักความปลอดภัย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาโปรแกรมการทำงานการขับเคลื่อนด้วยมอเตอร์ไฟฟ้าและหลักการของวีลแชร์
2. เพื่อศึกษาในวิธีการออกแบบตัวควบคุมของระบบรถวีลแชร์
3. เพื่อศึกษาโปรแกรมการทำงานให้สามารถเดินได้อัตโนมัติ

1.3 ขอบเขตของปริญญานิพนธ์

1. สร้างรถวีลแชร์ที่สามารถขับเคลื่อนโดยใช้มอเตอร์ไฟฟ้าแทนการใช้แรงงานจากคน
2. สร้างรถวีลแชร์ที่สามารถเดินเป็นอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

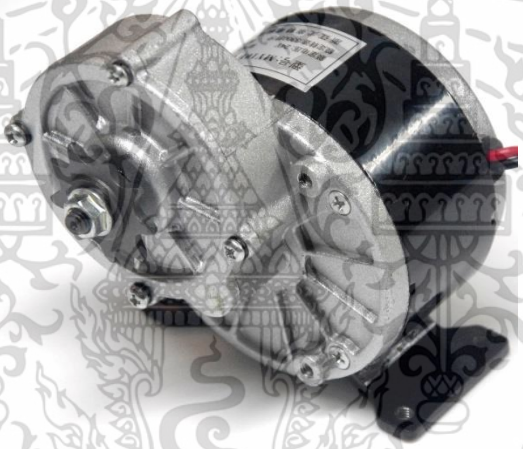
1. ศึกษาการทำงานของตัวอุปกรณ์ต่างๆ
2. ศึกษาหลักการของรถวีลแชร์
3. ศึกษาการออกแบบการติดตั้งตัวอุปกรณ์
4. ศึกษาเกี่ยวกับการวัดจำนวนรอบของ Encoder เพื่อไปทำการควบคุมมอเตอร์
5. ศึกษาการออกแบบตัวควบคุมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 มอเตอร์ไฟฟ้ากระแสตรง (Direct current motor)

มอเตอร์ไฟฟ้ากระแสตรง (Direct current motor) หรือเรียกว่า ดี.ซี มอเตอร์ (D.C. MOTOR) เป็นเครื่องกลไฟฟ้ากระแสตรงที่เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล มีทั้งชนิดกระตุ้นฟิลด์จากภายนอก (Separated excited motor) และชนิดกระตุ้นฟิลด์ด้วยตัวเอง (Self-excited motor) มอเตอร์ไฟฟ้ากระแสตรงเป็นต้นกำลังขับที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรม เพราะมีคุณสมบัติที่เด่นในด้านการปรับความเร็วรอบตั้งแต่ความเร็วรอบต่ำสุดไปจนถึงความเร็วรอบสูงสุดนิยมที่จะนำมาใช้สำหรับยานยนต์ไฟฟ้า



รูปที่ 2.1 มอเตอร์ไฟฟ้ากระแสตรง

2.1.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง (Direct current motor)

มอเตอร์ไฟฟ้ากระแสตรง (Direct current motor) เมื่อแรงดันไฟฟ้ากระแสตรงเข้าไปในมอเตอร์ส่วนหนึ่งจะแปรงถ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์เมเจอร์ สร้างสนามแม่เหล็กขึ้นและกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ขั้วใต้ขึ้นจนเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกันตามคุณสมบัติของเส้นแรงแม่เหล็กจะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกันและทิศทางเดียวจะเสริมแรงกันทำให้เกิดแรงบิดในตัวอาร์เมเจอร์ทำให้อาร์เมเจอร์นี้หมุนได้อาร์เมเจอร์ที่หมุนนี้เรียกว่าโรเตอร์ (Rotor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 องค์ประกอบหลักของมอเตอร์ไฟฟ้ากระแสตรง

สเตเตอร์ (Stator) เป็นส่วนที่อยู่กับที่ประกอบด้วยโครงภายนอกทำหน้าที่เป็นทางเดินเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจรและยึดส่วนประกอบอื่น ๆ ให้แข็งแรงสเตเตอร์ทำด้วยเหล็กหล่อหรือเหล็กเหนียวรูปทรงกระบอก มีลักษณะเป็นขั้วแม่เหล็กยื่นทำด้วยเหล็กแผ่นบางเคลือบด้วยฉนวนเรียงซ้อนกันผิวด้านหน้าเป็นรูปโค้งรับกับทรงกลมของอาร์เมเจอร์และที่แกนเหล็กจะพันด้วยขดลวดแดงทำหน้าที่รับกระแสไฟฟ้าจากภายนอกเพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น อาจจะมี 2 ขั้ว 4 ขั้ว หรือหลายขั้วขึ้นอยู่กับการออกแบบมอเตอร์ นอกจากนั้นยังมีแปรงถ่านและซองติดตั้งไว้สัมผัสกับคอมมิวเตเตอร์ ทำหน้าที่รับกระแสไฟฟ้าเข้าสู่มอเตอร์เพื่อเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานกล โดยมีฝาปิดหัวท้ายสำหรับรองรับแบริ่งและเพลาอีกด้วย แปรงถ่านทำด้วยคาร์บอนมีรูปร่างเป็นแท่งสี่เหลี่ยมผืนผ้าในซองแปรงมีสปิงกดอยู่ด้านบน เพื่อให้ถ่านนี้สัมผัสกับซี่คอมมิวเตเตอร์ตลอดเวลาเพื่อรับกระแสไฟฟ้าและส่งกระแสไฟฟ้าระหว่างขดลวดอาร์เมจอร์กับวงจรไฟฟ้าจากภายนอก คือ ถ้าเป็นมอเตอร์กระแสไฟฟ้าตรงจะทำหน้าที่รับกระแสจากภายนอกเข้าไปยังคอมมิวเตเตอร์ ให้ลวดอาร์เมเจอร์เกิดแรงบิดทำให้มอเตอร์หมุนได้ตัวหมุนหรืออาร์เมเจอร์มีลักษณะเป็นทุ่นทรงกระบอกทำด้วยแกนเหล็กแผ่นบางๆ อัดซ้อนกันที่ผิวด้านหน้าของทรงกระบอก

2.1.3 การขับเคลื่อนด้วยมอเตอร์ไฟฟ้า

ในการนำมอเตอร์ไปใช้งานจะต้องทำความเข้าใจเกี่ยวกับการเริ่มเดินของมอเตอร์ไฟฟ้ากระแสตรง (MotorStarting) โดยมีวัตถุประสงค์หลักเพื่อลดกระแสไฟฟ้าขณะเริ่มหมุนให้ต่ำลง นอกจากนี้เมื่อมอเตอร์หมุน จะต้องสามารถควบคุมความเร็วรอบของมอเตอร์ให้เหมาะสมกับโหลด สำหรับการควบคุมความเร็วมีความสำคัญมาก โดยเฉพาะในงานอุตสาหกรรมโดยทั่วไปจะใช้การควบคุมความเร็วด้วยวงจรอิเล็กทรอนิกส์กำลังซึ่งทำการศึกษาเอกสารและงานวิจัยที่เกี่ยวข้องดังนี้ 8 ดุสิต สูรย์ราช (2546 : 216) กล่าวว่า การเริ่มเดินหรือสตาร์ทมอเตอร์กระแสตรงจำเป็นต้องมีอุปกรณ์ควบคุมเพราะขณะสตาร์ทจะกินกระแสสูงมากและการควบคุมความเร็วมีหลายวิธีขึ้นอยู่กับชนิดของมอเตอร์ พรจิต ประทุมสุวรรณ (2548 : 31) กล่าวว่า ขณะอาร์เมเจอร์เริ่มเดินความต้านทานอาร์เมเจอร์มีค่าน้อยมากจึงทำให้กระแสขณะเริ่มออกตัวมีค่าสูงจากผลดังกล่าวทำให้เกิดการสปาร์คที่คอมมิวเตเตอร์และความร้อนที่ขดลวดจึงต้องลดแรงดันตกคร่อมที่ขั้วของมอเตอร์แล้วค่อย ๆ เพิ่มให้สูงขึ้นที่อัตราเร็วของมอเตอร์ ธนภัทร ไพคำนาน (2552 : 144) กล่าวว่า ขณะที่มอเตอร์เริ่มหมุนจะต้องการกระแสมากกว่าปกติเพื่อเอาชนะความฝืดของตัวหมุนแล้วปรับความเร็วรอบให้สูงตามพิกัดจึงต้องจำกัดค่ากระแสในอาร์เมเจอร์ให้ต่ำเพื่อป้องกันมอเตอร์ สุธน แก่นต้น (2556 : 300) กล่าวว่า การเริ่มเดินของมอเตอร์ไฟฟ้ากระสตรงจะมีกระแสสูงเพราะค่าความต้านทานอาร์เมเจอร์มีค่าต่ำทำให้กระแสไหลผ่านขดลวดอาร์เมเจอร์สูงจึงต้องต่อตัวต้านทานภายนอกเข้าไปอนุกรมกับอาร์เมเจอร์เพื่อลดกระแส Rizzoni, Giorgio (2004 :124) กล่าวว่า การเริ่มเดินหรือสตาร์ทมอเตอร์กระแสตรงจำเป็นต้องมีอุปกรณ์ที่ลดกระแสขณะสตาร์ทและการควบคุมความเร็วและแรงบิดที่เกิดขึ้นในมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งมีหลายวิธีขึ้นอยู่กับชนิดของมอเตอร์ปัจจุบันส่วนมากจะใช้อุปกรณ์อิเล็กทรอนิกส์กำลังจากที่กล่าวมาข้างต้น ในการเริ่มเดินหรือสตาร์ทมอเตอร์กระแสตรงจะมีกระแสสูงเนื่องจากความต้านทานอาร์เมเจอร์มีค่าต่ำจากผลดังกล่าวจะต้องควบคุมกระแสให้ลดต่ำลงเพื่อป้องกันขดลวดภายในมอเตอร์ซึ่งมีวิธีการต่างๆ เช่น การต่อความต้านทานอนุกรมกับอาร์เมเจอร์เพื่อลดกระแส หรือใช้อุปกรณ์อิเล็กทรอนิกส์เพื่อมาควบคุมแรงดันไฟฟ้าที่ขั้ว

2.2 รถเข็นผู้พิการ

รถเข็นของผู้พิการ (Wheelchair) คือ อุปกรณ์ที่สามรถช่วยเพิ่มความสะดวกสบายให้คนพิการมีวัตถุประสงค์ช่วยในการเคลื่อนที่ มีลักษณะคล้ายเก้าอี้แต่มีล้อส่วนใหญ่เป็นแบบสี่ล้อ โดยส่วนใหญ่จะใช้สำหรับภายในโรงพยาบาลเพื่อทำการเคลื่อนที่ในสถานที่ต่างๆเนื่องจากผู้ป่วยไม่สามารถเคลื่อนที่ได้

2.2.1 อุปกรณ์มาตรฐานรถเข็นผู้พิการ

1) ล้อหน้า (Front caster) เป็นล้อที่มีขนาดเล็ก สามารถหมุนได้รอบทิศทาง ทำหน้าที่ในการกำหนดทิศทางการขับเคลื่อนเก้าอี้รถเข็นให้ไปทางซ้ายหรือทางขวา ติดตั้งอยู่บริเวณด้านหน้าของล้อใหญ่ หรือ ที่เรียกว่าล้อหลัง ล้อหน้ามีหลายชนิดและหลายขนาด โดยจะถูกออกแบบมาให้เหมาะสมกับการงานของเก้าอี้ล้อเข็นแต่ละชนิด ตัวอย่างเช่น ล้อหน้าขนาดเล็กที่เป็นยางตันช่วยให้เก้าอี้รถเข็นผู้ป่วยมีความสามารถในการเคลื่อนตัวได้ดีบนพื้นผิวเรียบและแข็ง เช่น พื้นถนน พื้นปูน ส่วนล้อหน้าที่มีขนาดใหญ่และเป็นยางลม มีความสามารถในการเคลื่อนตัวได้ดีบนพื้นผิวที่ขรุขระด้วย เพราะมีความยืดหยุ่นในการรับแรงกระแทกได้ดี
2) ล้อหลัง (Rear wheels)เป็นล้อหลักที่ในการเคลื่อนตัวของ รถเข็นผู้ป่วย อาจมีขนาดเล็กหรือใหญ่กว่าล้อหน้าเล็กน้อยหรืออาจเป็นล้อขนาดใหญ่ซึ่งมีเส้นผ่านศูนย์กลาง 20-24 นิ้วก็ได้ ที่พบบ่อยในปัจจุบันโครงของวงล้อ มี 2 แบบ คือ ล้อแบบซี่ลวด (Spoke wheel) และ ล้อพลาสติกขึ้นรูป (PVC Wheel)

   ล้อแบบซี่ลวด (Spoke wheel) มีลักษณะคล้ายล้อจักรยาน โดยมีซี่ลวดยึดดุมล้อที่อยู่แกนกลางกับขอบล้อซี่ลวดจะยึดอยู่วงล้อโดยรอบล้อพลาสติกขึ้นรูป (PVC Wheel) หรือ ที่เรียกกันทั่วไปว่า ล้อแม็ค มีลักษณะคล้ายล้อของจักรยานยนต์มีแกนยึดกับดุมล้อ 2-6 ชิ้น ล้อแบบพลาสติกขึ้นรูปมีข้อดีคือ ไม่มีชิ้นส่วนที่จะเป็นสนิม ดูแลรักษาง่าย และมีความแข็งแรงเหมือนล้อแบบซี่ลวดปัจจุบันในเก้าอี้รถเข็นผู้ป่วยที่ต้องการให้มีสมรรถนะสูง เช่น สำหรับการแข่งขันกีฬา มีการใช้วัสดุ เช่น คาร์บอนไฟเบอร์ มาทำเป็นส่วนของล้ออีกด้วยส่วนยางหลังของล้อหลังมี 2 ชนิด เช่นเดียวกับยางของล้อหน้า คือ แบบยางตัน และ ยางลม โดยมีสมรรถนะที่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ ยางตันเหมาะกับการใช้งานบนพื้นผิวเรียบ เช่น ในสถานที่ทำงาน อาคาร ห้างร้าน บ้านเรือนในขณะที่ยางลมเหมาะกับการใช้งานบนพื้นผิวไม่เรียบ เช่น ตามท้องถนน ผิวการจราจรที่ขรุขระ ล้อแบบยางตัน มีข้อดีที่ชัดเจนคือ ใช้งานง่าย อายุการใช้งานยาวนานไม่ต้องการการดูแลหรือเติมลมก่อนการใช้งาน

3) ที่หมุนล้อ (Hand rims หรือ Push rims) เป็นวงล้อที่มีไว้สำหรับจับเพื่อการเคลื่อนเก้าอี้ล้อเข็นจะติดเชื่อมอยู่กับล้อหลังแต่ห่างจากวงล้อของล้อหลังออกมาราว ๆ 1-2 นิ้ว โดยล้อเข็นมี 2 แบบ คือ แบบที่ 1แบบวงล้อ ที่หมุนล้อแบบนี้จะเป็นวงกลม ที่มีขนาดเล็กกว่าส่วนของล้อและยาง และ ยึดติดอยู่ด้านนอกของล้อ ที่หมุนล้อมักทำด้วยโลหะเบา หรือ พลาสติก ABS หรือ PVC ก็ได้

   แบบที่ 2 แบบล้อเข็นแกนสั้น จะมีซี่สั้นๆติดระหว่างแกนกลางกับล้อเข็นโดยรอบ ส่วนมารกมักจะติดราว 6 ชิ้น ล้อแบบนี้เหมาะสำหรับผู้ที่ไม่สามารถกำมือได้ ซี่ที่ติดอยู่รอบ ๆ จะมีไว้ให้นิ้วเกี่ยวหรือใช้มือผลักซี่ให้เก้าอี้ล้อเข็นเคลื่อนที่ไปมาได้

4) ที่ล็อกล้อและคันล็อกล้อ (Wheel locks) เป็นอุปกรณ์ที่ใช้ในการล็อกล้อไม่ให้เลื่อนไหล เมื่อเก้าอี้รถเข็นผู้ป่วยอยู่กับที่ดังนั้นที่ล็อกล้อจะไม่มีไว้สำหรับเบรกหรือห้ามล้อขณะเก้าอี้ล้อเข็นกำลังเคลื่อนที่หากใช้ในขณะเก้าอี้ล้อเข็นเคลื่อนที่อาจเกิดการพลิกคว่ำได้ส่วนการติดตั้งนั้น ขึ้นอยู่กับความถนัดของผู้เก้าอี้ล้อเข็น จะติดตั้งให้อยู่สูงหรือ ต่ำก็ได้

5) พนักพิง (Back support) เป็นส่วนประกอบที่มีไว้รองรับหลังของผู้ใช้เก้าอี้ล้อเข็น ดังนั้น ควรมีความทนทานและยืดหยุ่นเพื่อให้เกิดความสบายในการพิง ไม่ควรสูงหรือต่ำไป ในเก้าอี้ล้อเข็นบางรุ่นพนักพิงสามารถปรับเอนได้ ควรมีเข็มขัดนิรภัยสำหรับผู้ใช้เก้าอี้ล้อเข็นที่มีปัญหาเรื่องการทรงตัวเพื่อความปลอดภัย

6) ที่พักแขน (Armrest) เป็นที่สำหรับพักแขนท่อนล่าง เมื่อวางแล้วจะทำให้เกิดความสบายผ่อนคลาย ส่วนมากสามารถปรับความสูงต่ำได้ตามความสะดวกสบายของผู้ใช้ แต่สำหรับในบางรุ่นของเก้าอี้รถเข็นผู้ป่วยจะไม่มีที่พักแขนในกรณีที่ผู้ใช้ต้องการความคล่องตัว

7) ที่นั่ง (Seat) ที่นั่งทำหน้าที่ในการรองรับน้ำหนัก และ ร่างกายของผู้ใช้เก้าอี้ล้อเข็นจะเป็นแผ่นราบติดตั้งอยู่ด้านบนของคานและสายสลิงในแต่ละรุ่นอาจจะทำจากวัสดุที่แตกต่างกัน เช่น ทำจากผ้า ทำจากหนังเทียม หรือ ทำจากโลหะ

8) ที่พักเท้า (Footrest)เป็นที่วางเท้าเพื่อให้เกิดความสบายในบางรุ่นสามารถพับเก็บได้ สามารถปรับระดับได้และสามารถถอดเก็บได้ โดยที่วางเท้าจะช่วยให้เท้าของผู้ใช้ไม่ลากกับพื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Arduino Nano

บอร์ด Arduino Nano ออกแบบให้มีขนาดเล็กและใช้กับงานทั่ว ๆ ไปใช้ชิปไอซีไมโครคอนโทรลเลอร์เบอร์ ATmega168 หรือเบอร์ ATmega328 จะโปรแกรมผ่านโปรโตคอล UART มีชิป USB to UART มาใช้ Mini USB เชื่อมต่อกับคอมพิวเตอร์ มีพอร์ตดิจิตอลอินพุตเอาต์พุต 14 พอร์ต และมีพอร์ตอนาล็อกอินพุต 8 พอร์ต บนบอร์ดยังมีเรกกูเลเตอร์ สามารถจ่ายไฟได้ตั้งแต่ 7-12V เพื่อให้บอร์ดทำงานได้และรายละเอียดเพิ่มเติมของบอร์ด Arduino Nano มีดังนี้

1) ชิปไอซีไมโครคอนโทรลเลอร์ ATmega168 หรือ ATmega328
2) ใช้แรงดันไฟฟ้าที่ใช้งาน 5 V
3) รองรับการจ่ายแรงดันไฟฟ้า (ที่แนะนำ) 7-12V
4) รองรับการจ่ายแรงดันไฟฟ้า (ที่จำกัด) 6-20V
5) พอร์ต Digital I/O 14 พอร์ต (มี 6 พอร์ต PWM Output)
6) พอร์ต Analog Input 6 พอร์ต
7) กระแสไฟที่จ่ายได้ในแต่ละพอร์ต 40mA
8) กระแสไฟที่จ่ายได้ในพอร์ต 3.3V 50mA
9) พื้นที่แรม 1 หรือ 2KB
10) พื้นที่หน่วยความจำถาวร (EEPROM) 512 หรือ 1KB
11) ความถี่คลิสตัน 16MHz
12) ขนาด 45x18 mm
13) น้ำหนัก 5 กรัม

รูปที่ 2.2 Board Arduino Nano

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Arduino Mega 2560

Arduino MEGA คือ บอร์ดรุ่นใหญ่ในกลุ่มบอร์ด Arduino โดยใช้ Atmega 2560 เป็นไมโครคอนโทรลเลอร์หลัก ซึ่งไมโครคอนโทรลเลอร์ตัวนี้แตกต่างจาก ATmega328 ที่ใช้อยู่กับบอร์ด Arduino UNO โดย Arduino MEGA มี Digital Pins ขา อินพุต / เอาต์พุต ดิจิตอล จำนวน 54 ขา (เป็น PWM ได้ 15 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด และขาแหล่งจ่ายไฟ 5V จำนวน 3 ขา สามารถเขียนโปรแกรมบน Arduino IDE และโปรแกรมผ่าน USB เหมาะสำหรับผู้ที่สนใจเริ่มต้นเรียนรู้พัฒนา microcontroller ที่ต้องการใช้งาน Arduino และมีหน่วยความจำและขาสัญญาณต่างๆ ให้ต่อใช้งานมากขึ้น และรายละเอียดเพิ่มเติมของ Arduino Mega2560

1) I/OPort -Digital
2) ICSP Port - Atmega328
3) MCU Atmega328
4) I/O Port - digital I/O - A0-A5
5) PowerPort+ 3. 3 V, +5 V, GND, Vin
6) Power Jack: 7-12 V
7) MCU: Atmega16U2
8) USB Port:
9) Reset Button
10) ICSP Port

รูปที่ 2.3 Arduino Mega 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

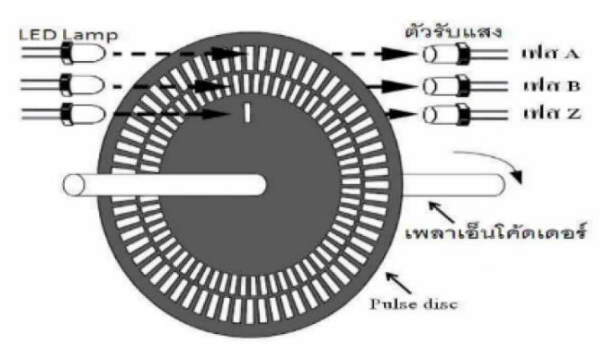
2.5 เอ็นโค้ดเดอร์ (Encoder Omron E6B2-CWZ6C)

เป็นเอ็นโค้ดเดอร์แบบแกนหมุน คือ เซนเซอร์ชนิดหนึ่งที่ทำหน้าที่ในการเข้ารหัส จากระยะทางจากการหมุนรอบตัวเอง และแปลงออกมาเป็นรหัสในรูปแบบของสัญญาณไฟฟ้า โดยสามารถนำรหัสนี้มาแปลงกลับ เพื่อหาค่าต่างๆ ที่ต้องการ ไม่ว่าจะเป็นระยะทางการหมุน องศาการเคลื่อนที่ หรือความเร็วรอบ แล้วนำมาแสดงผลให้ทราบค่าผ่านหน้าจอแสดงผล เช่น ถ้าต้องการวัดระยะทาง จะต้องต่อเข้ากับตัวนับจำนวน เพื่อแสดงผลเป็นระยะทาง หรือ ถ้าต้องการวัดความเร็วรอบจะต้องต่อเข้ากับตัววัดพัลส์ โดยการประยุกต์ใช้เอ็นโค้ดเดอร์นั้น สามารถใช้ทำงานได้หลากหลาย เช่น กระบวนการประกอบชิ้นส่วนอิเล็กทรอนิกส์ อุตสาหากรรมเซมิคอนดักเตอร์ เครื่องมือวัดต่างๆ

ส่วนการแสดงผลเป็นความเร็วรอบของ RPM โดยอาศัยสัญญาณที่ผ่านการเข้ารหัสแล้วออกมาเป็นสัญญาณทางไฟฟ้านั้น สามารถแบ่งรูปแบบของการเข้ารหัสได้อีกหลายรูปแบบ เช่น สัญญาณดิจิตอล ศูนย์กับหนึ่ง หรือ เป็นแบบ Binary Code, BCD Code, Gray Code และส่วนประกอบเบื้องต้นของเอ็นโค้ดเดอร์มีดังนี้

1) เพลา (Shaft) ใช้สำหรับต่อเข้ากับวัตถุที่หมุน เช่น มอเตอร์
2) แผ่นดิสซ์ (Code หรือ PulseDisc) จะเป็นแผ่นที่มีแทร็กหรือร่องเล็ก ๆ มีทั้งส่วนที่โปร่งแสงและทึบแสง เพื่อให้แสงอินฟาเรดลอดผ่านได้
3) แหล่งแสง (Light Source) เป็นไฟ LED คุณภาพสูง
4) ตัวรับแสง (Photodetector หรือ Photodiode) ใช้รับแสงจาก LED เพื่อแปลงไปเป็นรหัสข้อมูลการนำไปใช้งาน จากส่วนประกอบที่ได้กล่าวไป จะเห็นว่าตัวส่งแสงจะมี 3 ชุด เรียกว่า เฟส A, Bและ Z เอาต์พุตของ Encoder ที่ออกมาจึงมี 3 ชุด เพื่อนำไปใช้งานได้หลากหลาย โดยจะมีข้อแตกต่างคือ เฟส A และ B จะเหลื่อมกันอยู่ 90 องศา (ทำให้นำไปใช้งานเพื่อดูทิศทางการหมุนได้ ว่าหมุนซ้ายหรือขวา คือ เจอ A ก่อน B หรือ B ก่อน A) ส่วนเฟส Z เมื่อ 1 รอบจึงจะปล่อยแสงออกมา 1 ครั้ง ส่วนใหญ่ Encoder มักถูกนำไปใช้ในงานต่อกับคัปปลิ้งหรือต่อเข้ากับลูกล้อไม่ว่าจะเป็นมอเตอร์หรือสายพานลำเลียง เพื่อวัดรอบ, วัดระยะทาง, วัดทิศทางการหมุน ไปจนถึงการตรวจจับตำแหน่ง จึงเป็นที่นิยมใช้อย่างแพร่หลายในหน้างานหลากหลายรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ส่วนประกอบของเอ็นโค้ดเดอร์

รูปที่ 2.5 Encoder Omron E6B2-CWZ6

2.6 แบตเตอรี่ (Battery)

เป็นอุปกรณ์ที่สามรถทำหน้าที่จัดเก็บพลังงานไฟฟ้าเพื่อนำไปใช้ประโยชน์ได้อย่าเต็มประสิทธิภาพ โดยหลักการคือปลี่ยนรูแบบขจากพลังงานเคมี ไปเป็นไฟฟ้าได้ ที่ประกอบ ด้วยขั้วบวกและขั้วลบพร้อมกับสารละลายอิเล็กโทรไลต์ (Electrolyte solution) อุปกรณ์สำหรับจัดเก็บไฟฟ้าเท่านั้นไม่ได้ผลิตไฟฟ้าสามารถประจุไฟฟ้าเข้าไปใหม่ (Recharge) ได้หลายครั้งและประสิทธิภาพจะไม่เต็ม 100% จะอยู่ที่ประมาณ 80% เพราะมีการสูญเสียพลังงานบางส่วนไปในรูปความร้อนและปฏิกิริยาเคมีจากการประจุ/จ่ายประจุนั่นเองแบตเตอรี่จัดเป็นอุปกรณ์ที่มีราคาแพงและเสียหายได้ง่ายหากดูแลรักษาไม่ดีเพียงพอหรือใช้งานผิดวิธีรวมถึงอายุการใช้งานของแบตเตอรี่ แต่ละชนิดจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แตกต่างกันไปเนื่องด้วยวิธีการใช้, การบำรุงรักษา, การประจุและอุณหภูมิ ฯลฯ แบตเตอรี่ที่เหมาะสำหรับใช้งานกับระบบไฟฟ้าพลังงานแสงอาทิตย์มากที่สุดคือแบตเตอรี่แบบจ่าย 16 ประจุสูง (Deep discharge battery)เพราะถูกออกแบบให้สามารถจ่ายพลังงานปริมาณมากหรือน้อยได้อย่างต่อเนื่องเป็นเวลานาน ๆ โดยไม่เกิดความเสียหายเราจะสามารถใช้ไฟฟ้าที่เก็บอยู่ในแบตเตอรี่นี้ได้อย่างต่อเนื่องถึง 80% โดยแบตเตอรี่ไม่ได้รับความเสียหายซึ่งต่างจากแบตเตอรี่รถยนต์ที่ถูกออกแบบให้จ่ายพลังงานสูงใน ช่วงเวลาสั้นๆ ถ้าใช้ไฟฟ้ามากกว่า 20 - 30% ของพลังงานที่เก็บอยู่จะทำให้อายุการใช้งานสั้นลงได้ ส่วนมากแบตเตอรี่ที่ใช้ในระบบโซลาร์เซลล์จะมีลักษณะที่ฝาครอบด้านบนเปิดออกได้เพื่อให้สามารถตรวจสอบเซลล์ และเติมน้ำในเวลาที่จำเป็นได้ เรียกว่า แบตเตอรี่ แบบเซลล์เปิด (Open cell หรือ Unsealed หรือ Flooded cell battery) มีบางชนิดที่ถูกปิดแน่นและไม่ต้องการการซ่อมบำรุง เรียกว่า แบตเตอรี่แบบไม่ต้องดูแลรักษา (Maintenance free หรือ Sealed battery) ซึ่งทั้ง 2 ชนิดที่ว่ามานั้นหายากและราคาสูงมาก

2.6.1 ชนิดของแบตเตอรี่ แบตเตอรี่แบ่งออกเป็น 2 ประเภท

2.6.1.1 ชนิดแห้ง (Dry Cell)

เป็นแบตเตอรี่ที่ไม่จำเป็นต้องเติมน้ำกลั่นตลอดอายุการใช้งาน โดยจะมีแผ่นปิดซีลไว้หลายๆ ชั้นป้องกันน้ำกรดในแบตเตอรี่ระเหยออกไปข้างนอกข้อดีของแบตเตอรี่ประเภทนี้ คือ เมื่อจอดรถทิ้งไว้นานๆ แบตเตอรี่ก็ยังเก็บไฟอยู่สามารถสตาร์ทรถได้ปกติ และเนื่องจากแบตเตอรี่เป็นแบตเตอรี่ ที่ตอบโจทย์การใช้ชีวิตของผู้ใช้รถในทุกวันนี้ได้เป็นอย่างดีไม่ว่าจะเป็นในเรื่องของการเพิ่มประสิทธิภาพในการขับขี่และการเชื่อมต่อไฟฟ้าเพื่อใช้งานกับเครื่องมือสื่อสารอุปกรณ์ไฟฟ้าต่างๆ ภายในรถที่เปรียบเสมือนอุปกรณ์หลักที่รถยนต์ทุกคันจะต้องมีไว้ ไม่ว่าจะเป็นทีวีขนาดเล็กอุปกรณ์เสียบชาร์จมือถือกล้องวีดีโอสำหรับถ่ายภาพส่วนแบตเตอรี่แบบแห้งที่นิยมใช้กันอย่างมากในท้องตลาดตอนนี้ก็คือแบตเตอรี่แห้งแบบตะกั่ว-กรด ซึ่งเป็นแบตเตอรี่ที่มีราคาถูกกว่าแบตรูปแบบอื่น โดยด้านบนของแบตเตอรี่นั้นจะมีการซีลฝาปิดผนึกไว้เป็นอย่างดี และจะมีแค่ตาแมวไว้ใช้สำหรับตรวจเช็คระดับน้ำกรดและระดับไฟเท่านั้น

2.6.1.2 ชนิดน้ำ (Wet Cell)

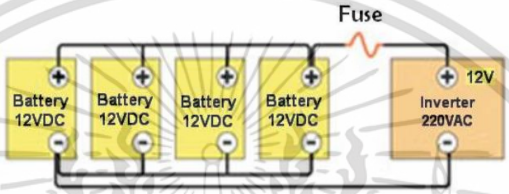
มี 2 ชนิด ดังนี้

1) แบตเตอรี่ด่าง เช่น แบตเตอรี่ในมือถือ, วิทยุสื่อสาร
2) แบตเตอรี่ตะกั่ว – กรด (Lead –Acid Battery) คือ แบตเตอรี่ที่ใช้ใน รถยนต์ทั่วไป, Traction Battery ใช้ในรถยกไฟฟ้า เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

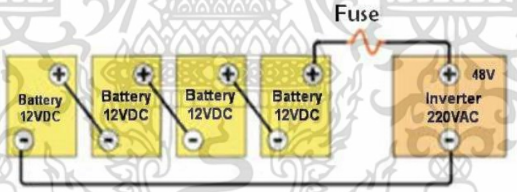
รูปที่ 2.6 แบตเตอรี่ (Battery)

ถ้าต้องการกระแสให้สูงมากขึ้นให้นำแบตเตอรี่หลายลูกมาต่อกันแบบขนานเพื่อให้ได้กระแสสูงขึ้นตามต้องการให้ใช้งานได้ยาวนานขึ้น



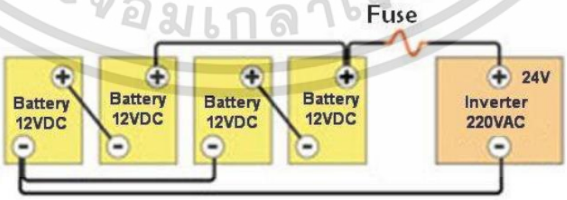
รูปที่ 2.7 แสดงการต่อแบบขนาน

ถ้าต้องการแรงดันมากขึ้นให้นำแบตเตอรี่หลายลูกมาต่อกันแบบอนุกรมเพื่อให้ได้แรงดันสูงขึ้น



รูปที่ 2.8 รูปแสดงการต่อแบบอนุกรม

ถ้าต้องการแรงดันและกระแสมากขึ้นให้นำแบตเตอรี่มาต่อกันแบบอนุกรมผสมกับขนาน



รูปที่ 2.9 แสดงการต่อแบบอนุกรมกับแบบขนาน

2.6.2 การทำงานของแบตเตอรี่

แบตเตอรี่จะมีส่วนประกอบคือเซลล์ร่วมกันโดยมี “แรงดัน” (Volt) ก็จะเกิดขึ้นที่ขั้วทั้งสองเมื่อระบบแบตเตอรี่ครบวงจร กระแสก็จะไหลทันทีเพื่อเปลี่ยนพลังงานเคมีออกมาเป็นพลังงานไฟฟ้าในกรณีนี้เรียกว่า “การคายประจุไฟ” (Discharge) ซึ่งตัวกรดในน้ำกรดผสมจะวิ่งเข้าทาปฏิกิริยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อแผ่นธาตุทั้งทางบวกและลบโดยจะค่อยๆ เปลี่ยนสภาพของแผ่นธาตุทั้งสองชนิดให้กลายเป็นตะกั่วซัลเฟต (Lead Sulfate) เมื่อแผ่นธาตุทั้งบวกและลบเปลี่ยนสภาพไปเป็นโลหะชนิดเดียวกันคือ “ตะกั่วซัลเฟรต” แบตเตอรี่ก็จะไม่มีสภาพของความแตกต่างทางแรงดันกระแสก็จะทำให้กระแสหยุดไหลหรือไฟหมด

2.6.3 ความสามารถในการจัดเก็บพลังงาน

ความจุของแบตเตอรี่ในการบรรจุพลังงานมีหน่วยเป็นแอมแปร์-ชั่วโมง(AmpereHour; Ah) พลังงานในแบตเตอรี่ 12 V 100 Ah เท่ากับ 12V x 100Ah หรือ 12V x 100A x 3600s จะได้เท่ากับ 4.32 MJ ถ้าแบตเตอรี่ 100 Ah เท่ากับว่าแบตเตอรี่จะจ่ายกระแส 1 แอมแปร์อย่าง 19 ต่อเนื่องเป็นเวลา 100 ชั่วโมง หรือ แบตเตอรี่ จ่ายกระแส 10 แอมแปร์อย่างต่อเนื่องเป็นเวลา 10 ชั่วโมง เช่นเดียวกับแบตเตอรี่จ่ายกระแส 5 แอมแปร์อย่างต่อเนื่องเป็นเวลา 20 ชั่วโมง ซึ่งทั้งหมดนี้ จ่ายกระแส เท่ากับ 100 Ah ทั้งสิ้น จะเห็นได้ว่า แบตเตอรี่ที่มีความจุเท่ากันอาจมีความเร็วในการจ่ายกระแสต่างกันได้ ดังนั้น การจะทราบความจุของแบตเตอรี่ต้องทราบถึง อัตราการจ่ายกระแสด้วยมักกำหนดเป็นจำนวนชั่วโมงของการจ่ายกระแสเต็มที่การกำหนดขนาดของแบตเตอรี่สำหรับระบบเซลล์แสงอาทิตย์นั้น ขึ้นอยู่กับความจุของ แบตเตอรี่ ในการจัดเก็บพลังงาน, อัตราการจ่ายประจุสูงสุด, อัตราการประจุสูงสุดและอุณหภูมิต่ำสุดที่จะนำแบตเตอรี่ไปใช้งาน(อุณหภูมิที่ได้ผลดีที่สุดแบตเตอรี่ตะกั่ว-กรด คือ 77 F หรือประมาณ 60-80 F)

2.6.4 ข้อควรระวังเกี่ยวกับแบตเตอรี่

1) ไม่ควรให้แบตเตอรี่จ่ายกระแสไฟเกินความสามารถ (Over Discharge) เพราะ ทำให้แบตเตอรี่มีอายุ การใช้งานสั้นลงกว่าปกติ
2) ไม่ควรประจุไฟแบตเตอรี่มากเกินไปควรประจุไฟให้ถูกต้องเหมาะสมมิเช่นนั้นแบตเตอรี่จะเสื่อมสภาพเร็วขึ้น
3) ไม่ควรให้อุณหภูมิของ Electrolyte สูงเกินกว่า 50 °C รักษาแบตเตอรี่ให้แห้ง สะอาดอยู่เสมอเพื่อป้องกันการรั่วซึมและผุกร่อน
4) ไม่ควรนำโลหะหรือเครื่องมือ เช่น ประแจหรือไขควงวางบนสะพานไฟ (Connector) เพราะอาจเกิดการ Spark สะเก็ดไฟทำให้แบตเตอรี่ชารุดเสียหาย
5) ไม่ควรสูบบุหรี่บริเวณที่มีการประจุไฟแบตเตอรี่
6) ตรวจสอบทุกครั้งเมื่อมีการเชื่อมต่อ Plug ของแบตเตอรี่เข้ากับ Plug ของ Charger หรือ Truck ต้องเป็นขนาดเดียวกัน และขั้วบวก ลบ ถูกต้อง
7) ไม่ควรถอดหรือขยับ Plug เมื่อมีการ On Charger หรือ On Key Switch ของ Truck
8) ถอด Plug ออกทุกครั้งเมื่อเลิกใช้ Truck หรือ เลิกการประจุไฟแบตเตอรี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 GY-521 Module MPU6050

GY-521 หรือที่เรียกว่า ไจโรสโคป MPU6050 (MPU6050) คือ ชื่อชิพประมวลผลบนตัว GY-521) คือโมดูลเซนเซอร์ที่ตรวจจับการเคลื่อนไหวและความเอียงของวัตถุ โดยตรวจวัดจากความเร่งเชิงเส้น (Linear Acceleration) และ ความเร็วเชิงมุม (Angular Velocity หรือใช้ Gyroscope) ถือเป็นอุปกรณ์แบบ 6 DOF (6 Degrees of Freedom) คืออุปกรณ์ที่สามารถตรวจวัดค่าได้ทั้ง 6 แกนคือ Ax ,Ay ,Az ,Gx ,Gy และ Gz

รูปที่ 2.10 Module MPU 6050

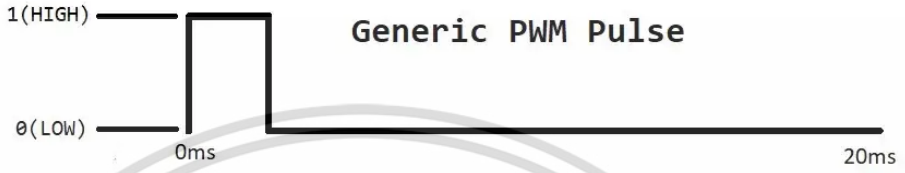
Accelerometer ประกอบด้วย ax ay และ az นิยมใช้ค่าเหล่านี้ในการบอกความเอียงเป็นองศา ณ ตำแหน่งนั้นๆ ซึ่ง Accelerometer วัดได้เฉพาะความเร่งเชิงเส้น ไม่สามารถวัดความเร็วเชิงมุมได้ แต่สามารถอ่านค่าความเอียงได้ เพราะว่าทุก ๆ ครั้งที่มีการหมุนหรือเอียง ตำแหน่งของชิพก็จะเปลี่ยน ส่งผลให้จุดต่างๆ บนตัวเซนเซอร์มีความเร่งเชิงเส้นไม่เท่ากันขึ้นอยู่กับองศาความเอียง ซึ่งความเร่งเชิงเส้นเป็นค่าที่เอาไว้เทียบกับค่า g ของโลก ดังนั้น จึงสามารถบอกได้ว่าตัวเซนเซอร์เอียงกี่องศา โดยเทียบจากค่า g

Gyroscope ประกอบด้วย gx gy และ gz เป็นค่าที่จะเปลี่ยนแปลงก็ต่อเมื่อมีการหมุนเซนเซอร์ในแกนต่างๆ และค่าจะกลับมาที่ประมาณ 90 องศาเหมือนเดิม เมื่อหยุดหมุน ซึ่งจะไม่ได้ค้างค่าตำแหน่งไว้เหมือนกับ Accelerometer เนื่องจาก Gyroscope ทำงานโดยการวัดความเร็วเชิงมุมที่หมุนรอบแกนต่างๆ ของมัน ไม่มีความเกี่ยวข้องกับค่า g มันจึงทำงานเฉพาะตอนที่มีการหมุน (ความเร็วเชิงมุมไม่เป็น 0) และไม่สามารถค้างค่าไว้ได้เมื่อหยุดหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 Pulse Width Modulation (PWM)

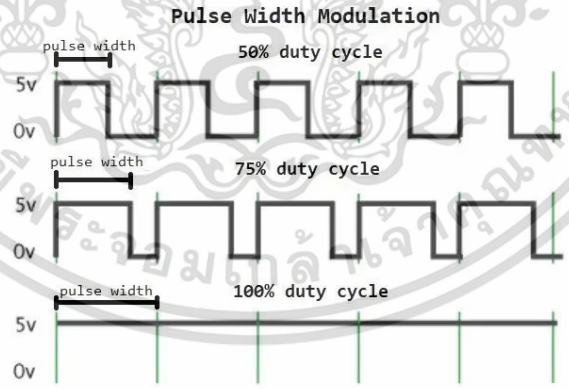
PWM (Pulse width modulation) คือ เป็นวิธีที่สามารถอ่านและเขียนข้อมูลแบบ Analog ด้วยสัญญาณ Digital ได้โดยตัวควบคุมการสร้างสัญญาณดิจิตอล (Digital control) จะสร้างสัญญาณคลื่นสี่เหลี่ยมออกมา โดยสัญญาณที่สร้างออกมาจะสลับกันระหว่าง เปิด(HIGH) กับ ปิด(LOW) รูปแบบสัญญาณเปิด-ปิดสามารถจำลองเป็นแรงดันไฟฟ้าระหว่าง เปิด (5-Volts) กับปิด (0 Volts)



รูปที่ 2.11 แสดงสัญญาณเปิด(HIGH) และปิด(LOW)

เรียกช่วงที่เป็นปิดหรือเปิดตามการทำงาน เนื่องจากช่วงที่เป็นแรงดันไฟฟ้าจะเป็น 0 ทำให้ไม่มีการทำงาน ส่วนในช่วงที่เปิดคือช่วงที่มีแรงดันไฟฟ้า ทำให้อุปกรณ์ทำงานได้ ซึ่งช่วงของเวลาที่สัญญาณเปิด เมื่อเทียบกับช่วงเวลาที่สัญญาณออกมาทั้งหมด จะเรียกช่วงเวลานี้ว่า ความกว้างของสัญญาณ (Pulse width)

2.8.1 ตัวอย่างสัญญาณ (PWM)



รูปที่ 2.12 ตัวอย่างสัญญาณ (PWM)

ความกว้างของเส้นสีเขียวจะแสดงถึงช่วงเวลาปกติ ระยะเวลาหรือช่วงเวลาจะเรียกว่า Period เป็นค่าผกผันของความถี่อุปกรณ์ หากความถี่ของอุปกรณ์มีค่า 50Hz (ความถี่จะขึ้นอยู่กับอุปกรณ์) ตามความกว้างของเส้นสีเขียวจะเป็นช่วงเวลา 20 ms (หาได้จาก T=1/f) ถ้าอุปกรณ์ที่ใช้มีการทำงานที่ความกว้างพัลส์ 1000-2000 us การทำงาน 100% หมายความว่าเจอสัญญาณ 2000 us ถ้ามีการทำงานเพียง 50% จะเจอสัญญาณเปิดเป็นเวลา 1500 us

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 ระบบควบคุม (Control System)

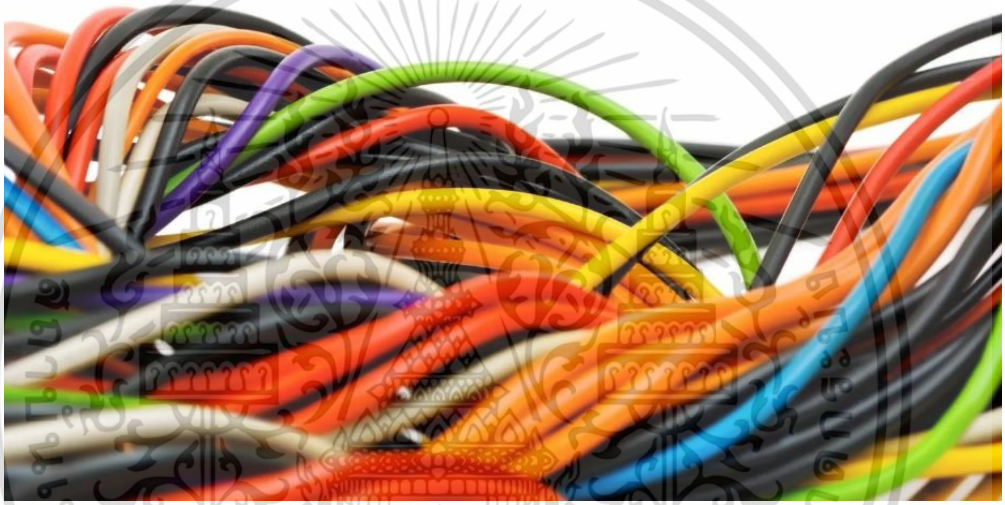
ระบบควบคุม (Control System) หมายถึง การควบคุมระบบหรือสิ่งที่ผู้ออกแบบต้องการควบคุมให้ได้ค่าผลลัพธ์ในรูปแบบของเอาต์พุตที่ต้องการซึ่งทำได้โดยการป้อนค่าอินพุตให้กับระบบโดยนิยามศัพท์พื้นฐานของวิชาระบบควบคุมมีดังนี้

1) อินพุต (Input) หมายถึง สัญญาณเข้าที่ต้องการป้อนให้กับระบบรับรู้ซึ่งอาจแสดงในรูปแบบของสัญญาณทางไฟฟ้า
2) ระบบ (System) หมายถึง สิ่งที่ต้องการหรือระบบที่ต้องการควบคุม ซึ่งจะประกอบด้วยชุดควบคุมกระบวนการ (Process) ซึ่งอาจเป็นเครื่องมืออุปกรณ์หรือเครื่องจักร
3) ระบบควบคุมวงเปิด (Open-Loop Control) หมายถึง ระบบที่มีการป้อนอินพุต (Input) ซึ่งอาจอยู่ในรูปสัญญาณทางไฟฟ้า (Electrical Signal) เข้าที่ระบบ (System) และได้สัญญาณออกหรือเอาต์พุต (Output) โดยไม่มีการนำสัญญาณป้อนกลับ (Feedback Signal) มาที่ระบบ
4) ระบบควบคุมวงปิด (Closed-Loop Control) หรือระบบป้อนกลับ (Feedback Control) หมายถึง ระบบควบคุมที่ใช้สัญญาณจากเอาต์พุตมาบ่งบอกหรือคำนวณค่าที่เหมาะสมสำหรับการควบคุม
5) เอาต์พุต (Output) หมายถึง ผลของการทำงานของระบบที่ผ่านการควบคุมซึ่งจะแสดงในรูปแบบ ผลตอบสนองทางกล (Mechanical Response) และผลตอบสนองทางไฟฟ้า (Electrical Response)
6) ระบบควบคุมแบบ ON-OFF เครื่องควบคุมจะสั่งเอาต์พุตทำงานเพียง 2 สภาวะเท่านั้น คือ ON และ OFF เป็นการควบคุมแบบง่ายๆ และมีราคาไม่แพง ดังนั้นจึงนิยมใช้กันอย่างกว้างขวางในงานควบคุมทางอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 สายไฟฟ้าแรงดันต่ำ (Low Voltage Power Cable)

เป็นสายไฟที่นิยมนทั่วไปใช้กับแรงดันจะไม่เกิน 750 V. ผลิตจากสายหุ้มฉนวน ส่วนประกอบด้วยทองแดงและออลูมิเนียม สายต่างๆจะแบ่งประเภทตาคือสายไฟจะทองแดงและสายไฟที่มี แต่สายขนาดใหญ่คือการนำแบบตัวนำตีเกลียว และสายไฟที่มีขนาดขนาดเล็กจะทำการเป็นตัวนำเดี่ยว



รูปที่ 2.13 สายไฟฟ้าแรงดันต่ำ (Low Voltage Power Cable)

การใช้งาน

- เดินลอย ต้องยึดด้วยวัสดุฉนวน (insulator)
- เดินในช่องเดินสาย ในสถานที่แห้ง
- ห้ามเดินฝังดินโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 สวิตช์ไฟฟ้า (Switch)

สวิตช์ไฟฟ้าใช้ในการเปิด-ปิดอุปกรณ์ไฟฟ้าต่างๆ โดยใช้หลักการที่สวิตช์เปิดหรือปิดหน้าสัมผัส ซึ่งคล้ายกับสะพานที่เชื่อมให้กระแสสามารถไหลได้ในวงจรไฟฟ้า หน้าสัมผัสปิด (Closed Contact) คือหน้าสัมผัสเชื่อมต่อกันทำให้กระแสไหลผ่านได้ ส่วนหน้าสัมผัสเปิด (Open Contact) คือหน้าสัมผัสแยกออกจากกันทำให้กระแสไม่สามารถไหลผ่านได้

สวิตช์มีหลายประเภท และถูกควบคุมด้วยวิธีต่างๆ กัน พวกที่ถูกควบคุมด้วยแรงจากมนุษย์ ได้แก่ Push-button Switch, Toggle Switch, Foot Switch พวกที่ถูกควบคุมด้วยแรงดันไฟฟ้าหรือความต่างศักย์ ได้แก่ Relay, Magnetic Contactor หรือพวกที่ถูกควบคุมด้วยแรงอื่นๆ เช่น Pressure Switch, Photoelectric Switch, Level Switch เป็นต้น

สวิตช์จะถูกแยกด้วยจำนวน Pole และจำนวน Throw โดยจำนวน Pole (SP-Single Pole, DP-Double Pole, 3P-Triple Pole, etc.) จะบอกจำนวนวงจรที่จะทำการเปิด-ปิด และ จำนวน Throw (ST, DT) จะบอกจำนวนของตัวเลือกของ Pole ตัวอย่างเช่น SPST- Single Pole Single Throw สวิตช์จะสามารถเลือกได้เพียงอย่างเดียวโดยจะเป็นปกติเปิด (NO-Normally Open) หรือปกติปิด (NC-Normally Closed) แต่ถ้าเป็น SPDT- Single Pole Double Throw สวิตช์จะมีหนึ่งคู่เป็นปกติเปิด (NO) และอีกหนึ่งคู่เป็นปกติปิดเสมอ (NC)

รูปที่ 2.14 วงจรสวิตช์ไฟฟ้า (Switch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 สวิตช์ไฟฟ้า (Switch)

2.12 DC Motor Drive H-Bridge SE-HB40-1

คุณสมบัติทางเทคนิค

- 1xแผงวงจรขับดีซีมอเตอร์ SE-HB40-1 พิกัด 12-24Vdc 40A พร้อมวงจรป้องกันแรงดันเกิน

เป็นบอร์ดขับดีซีมอเตอร์แบบเฮชบริดจ์ (สามารถควบคุมมอเตอร์หมุนกลับทางได้)

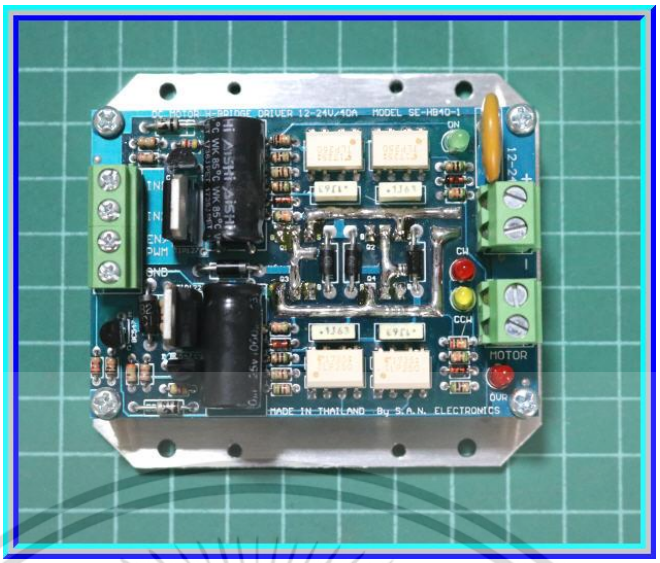
- รองรับแรงดันทางเอาต์พุต (แรงดันไฟเลี้ยงมอเตอร์) : 12-24Vdc

- รองรับแรงดันไฟอินพุต 3-5V/8mA TTL Level

- กระแสทางด้านเอาต์พุต : กระแสพีคสูงสุด 40A

- มีวงจรป้องกันแรงดันเกิน (Transient Voltage Protection)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 DC Motor Drive H-Bridge SE-HB40-1

- มีออปโต้คัปเปลอร์ (TLP250 Datasheet) สำหรับแยกกราวด์ทางด้านอินพุต ออกจากทางด้านเอาต์พุต

- ใช้เพาเวอร์มอสเฟตเบอร์จำนวน 4 ตัว

2x IRF2807 (Datasheet) ที่มีสเปคแรงดันสูงสุดถึง 75V 82A พร้อมติดตั้งแผ่นระบายความร้อนคุณภาพสูงทุกตัว

2x IRF4905 (Datasheet) ที่มีสเปคแรงดันสูงถึง 55V 74A พร้อมติดตั้งแผ่นระบายความร้อนคุณภาพสูงทุกตัว

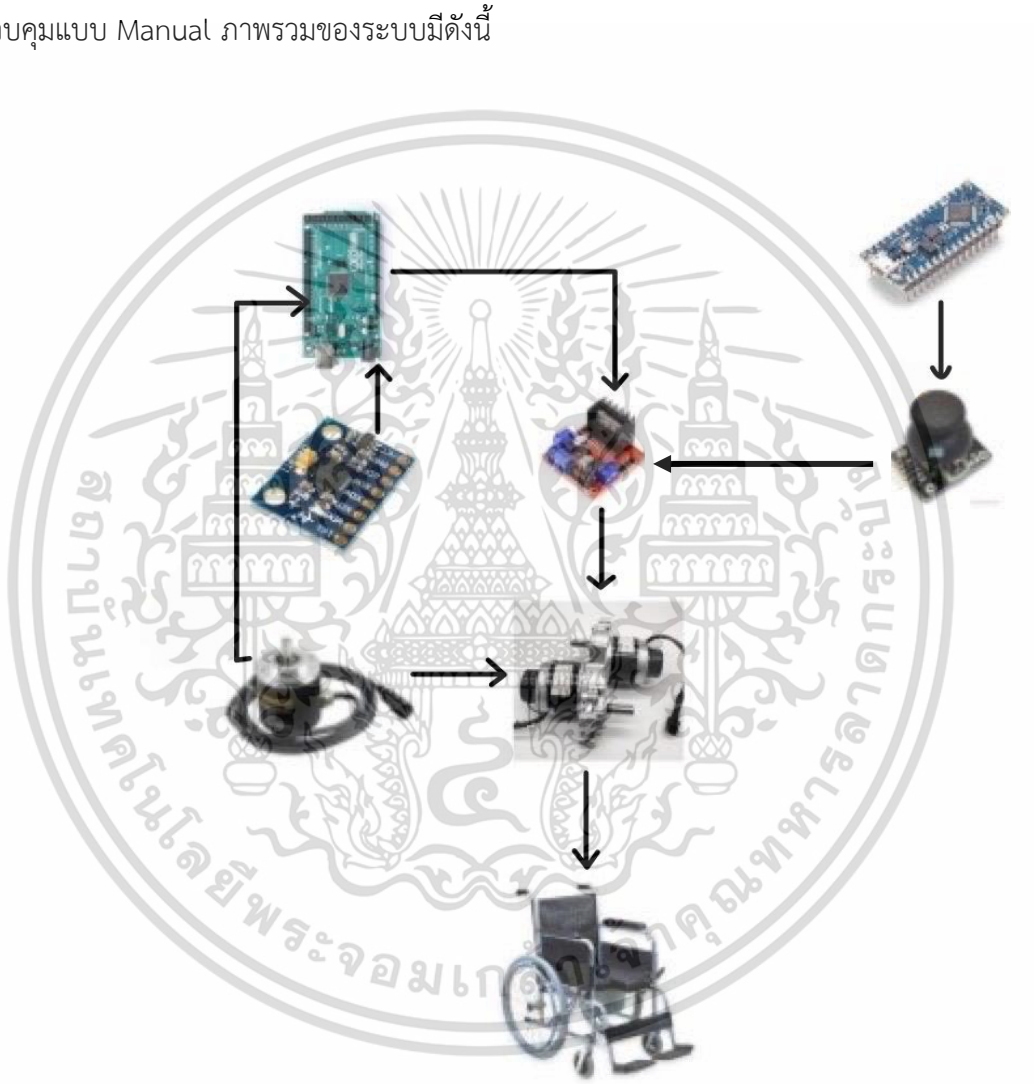
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินการ

3.1 การออกแบบการทำงานของระบบ

ในการออกแบบวีลแชร์อัจฉริยะ ด้วยเทคโนโลยีระบบยายนต์ไฟฟ้าให้เคลื่อนที่ผ่านมอเตอร์ไฟฟ้าที่ทีระบบความปลอดภัยและแบ่งส่วนการทำงานได้ 2 ส่วนคือ ระบบความคุมแบบอัตโนมัติและควบคุมแบบ Manual ภาพรวมของระบบมีดังนี้

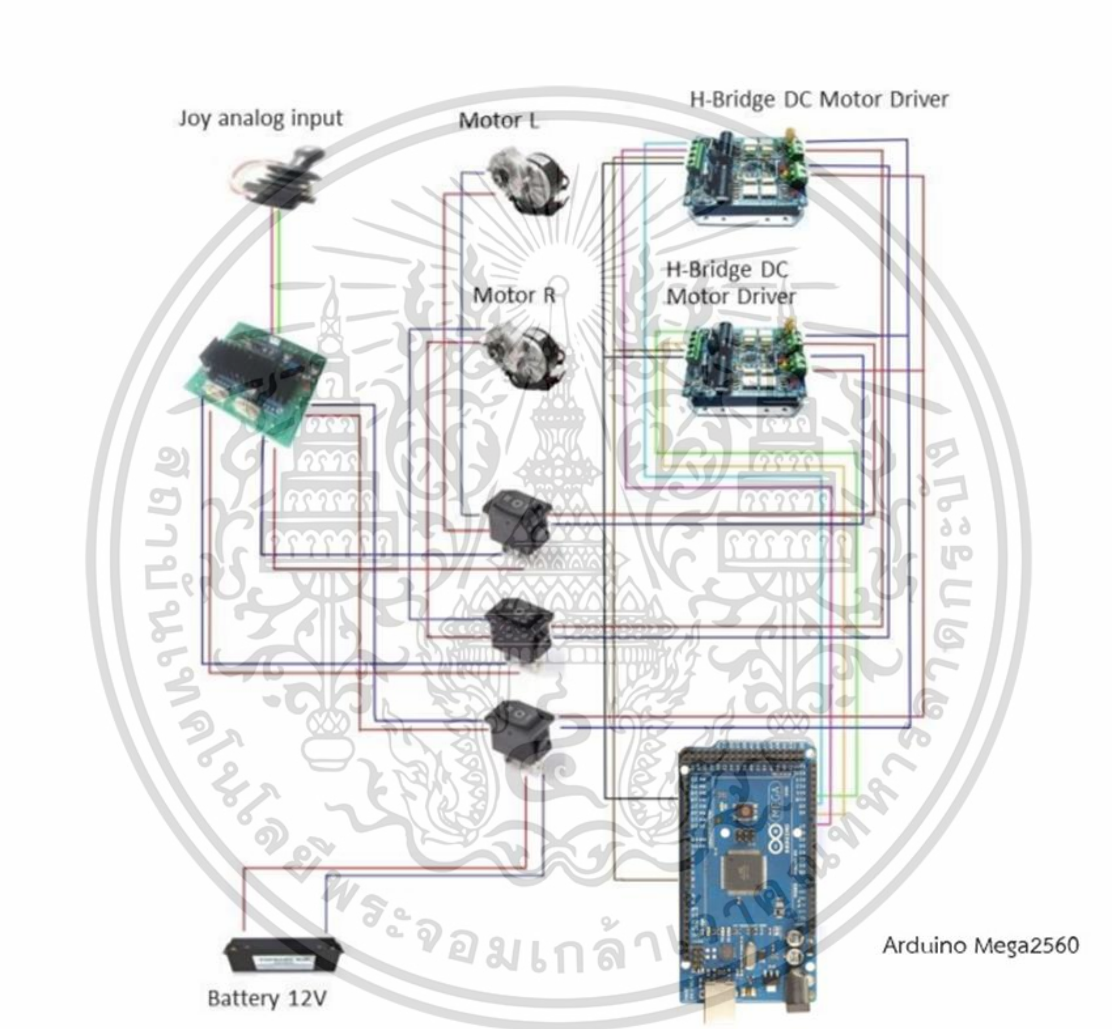


รูปที่ 3.1 ขั้นตอนการออกแบบการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

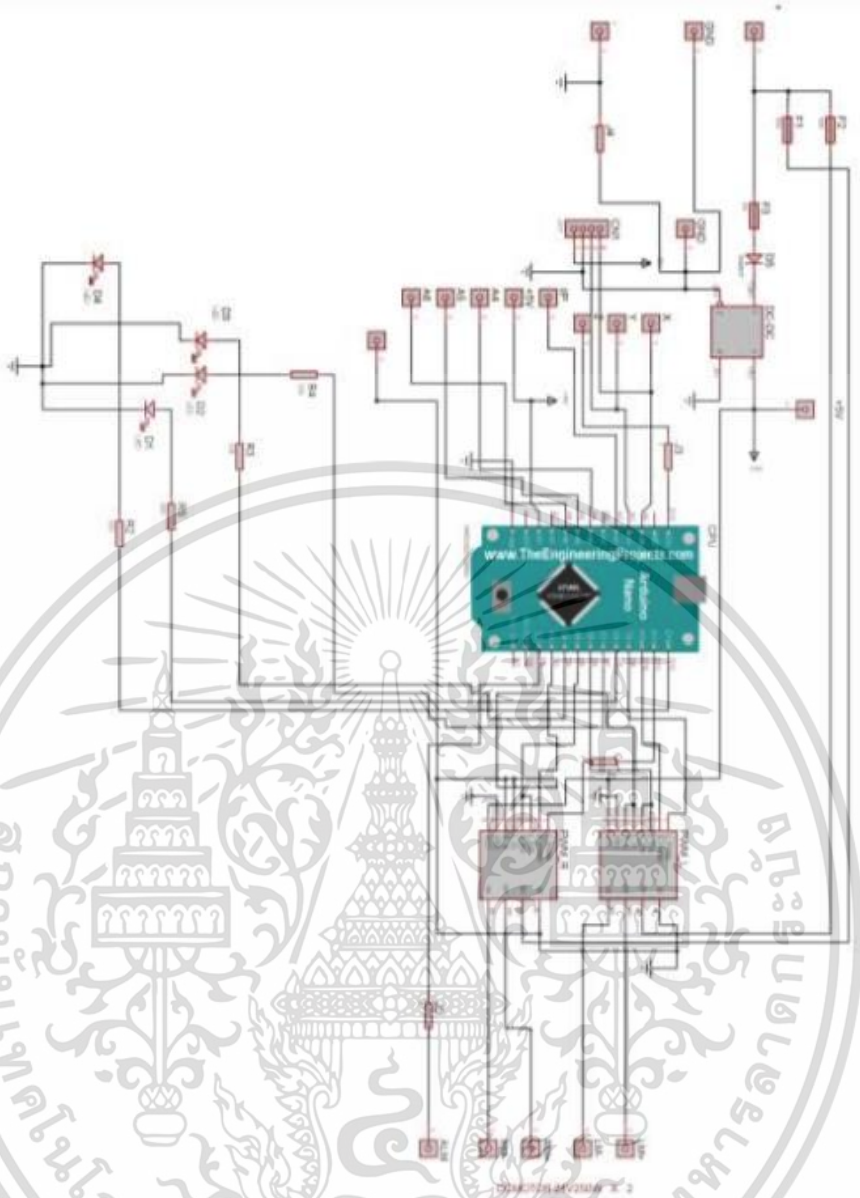
3.1.1 การทำงานของระบบควบคุมการขับเคลื่อน

หัวใจหลักของการขับเคลื่อน คือ ไมโครคอนโทรลเลอร์ Arduino Nano และ Arduino Mega 2560 Arduino Nano ซึ่งจะทำหน้าที่รับคำสั่งจากจอยสติ๊กที่ถูกควบคุมให้เดินหน้าเลี้ยวซ้าย หรือตัว Arduino Nano ได้รับคำสั่งจากนั้นนำไปประมวลผลเพื่อทำการสั่งให้ Motor ทำงานอีกครั้ง และในส่วนของ Arduino Mega 2560 คือการเขียนโปรแกรมป้อนเข้าไปเพื่อให้สามารถได้เดินอัตโนมัติโดยไม่ต้องควบคุมแบบ manual โดยมีผังการควบคุมดังนี้



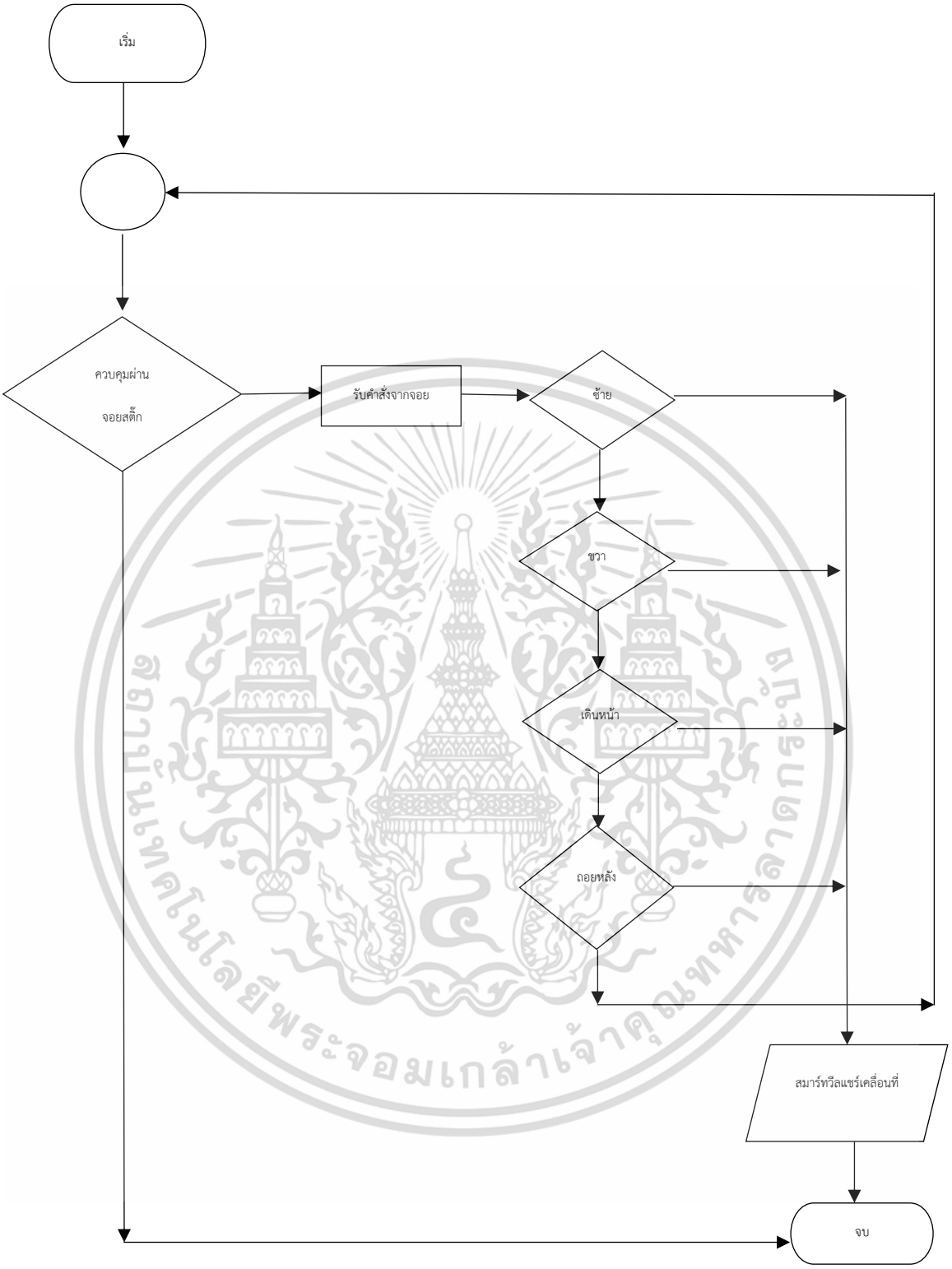
รูปที่ 3.2 ผังการแสดงการทำงานของระบบขับเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



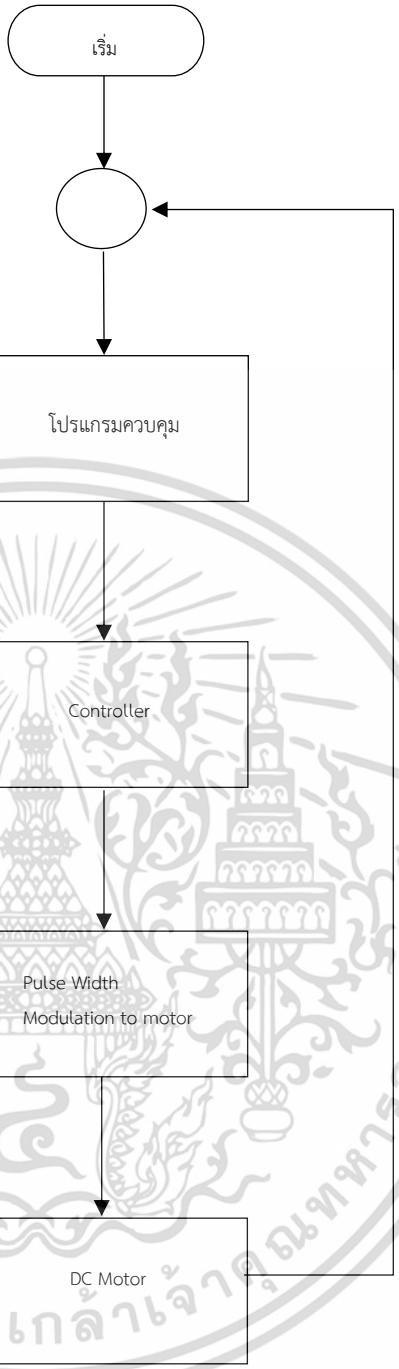
รูปที่ 3.3 วงจรควบคุมด้วยจอยสติ๊ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 Flow Chart แสดงการทำงานระบบควบคุมการขับเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Flow Chart การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วิธีดำเนินการ

3.2.1 ศึกษาหลักการทำงาน

ศึกษาเทคโนโลยีระบบยานยนต์ไฟฟ้าการขับเคลื่อนผ่านมอเตอร์ไฟฟ้า หลักการของรถวีลแชร์ ระบบอัตโนมัติ

3.2.2 ออกแบบระบบการทำงานของวีลแชร์

ออกแบบการทำงานรถวีลแชร์ อัจฉริยะโดยรวมและออกแบบระบบย่อยได้แก่ การทำงานของระบบควบคุมการขับเคลื่อนของมอเตอร์

3.2.3 จัดซื้ออุปกรณ์

จัดหาอุปกรณ์ โดยการสั่งผ่านออนไลน์ บางส่วนเนื่องจากสถานการณ์การแพร่ระบาดของเชื้อไวรัส (Covid-19) และสถานที่ใกล้เคียงไม่มีอุปกรณ์ครบตามที่ต้องการ

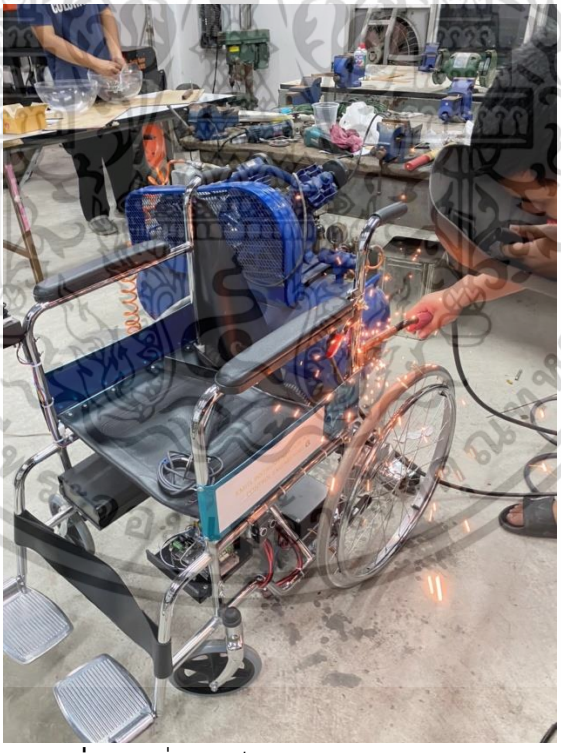
รูปที่ 3.6 จัดซื้อ Arduino Mega 2560

รูปที่ 3.7 จัดซื้อ Encoder E6B2-CWZ6C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 จัดซื้อ GY-521 หรือ Gyro Module 6050

3.2.4 ขั้นตอนการติดตั้ง



รูปที่ 3.9 เชื่อมเหล็กติดกับตัวโครงรถวีลแชร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 เจาะรูต๊าปเกลียวที่ตัวเอ็นโค้ดเดอร์

รูปที่ 3.11 ติดเอ็นโค้ดเดอร์ที่ตัวรถวีลแชร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

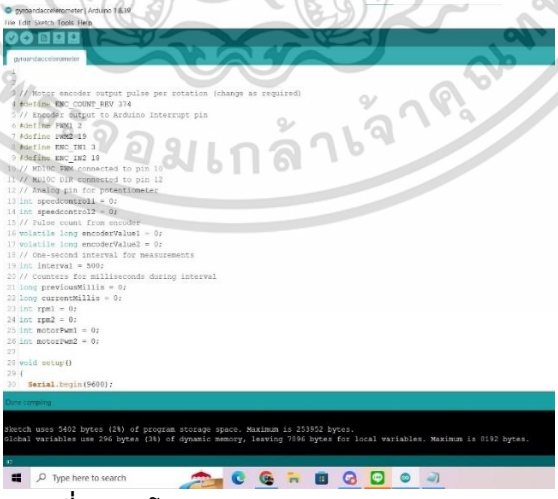
รูปที่ 3.12 ทดลองต่อสายจากตัวเอ็นเข้าไปยัง Arduino Mega 2560

3.2.5 การเขียนโปรแกรมในรูปแบบต่างๆ

เขียนโปรแกรมคำสั่ง ได้แก่ โปรแกรมทดลองการทำงานของ Encoder และทำการเชื่อมตัว Encoder เข้ากับตัวรถวีลแชร์

3.2.5.1 ขั้นตอนการทดลอง Encoder

1. ทำการทดลองเขียนโปรแกรมวัดค่าจาก Encoder ว่าทำงานได้หรือไม่
2. ทำการติดตั้งตัว Encoder เข้ากับตัวรถวีลแชร์
3. ทำการเขียนโปรแกรมควบคุม Speed Motor จาก Encoder



รูปที่ 3.13 โปรแกรมการทำงานของ Encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5.2 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบสี่เหลี่ยม

รูปที่ 3.14 โปรแกรมควบคุมแบบอัตโนมัติในรูปแบบสี่เหลี่ยม

3.2.5.3 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบวงกลม

รูปที่ 3.15 เขียนโปรแกรมควบคุมแบบอัตโนมัติรูปแบบวงกลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

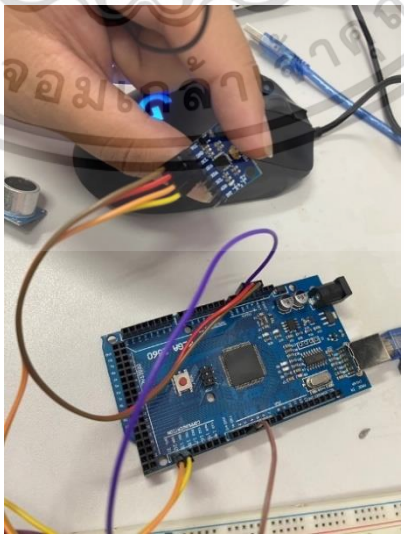
3.2.5.4 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบเลขแปด

รูปที่ 3.16 โปรแกรมควบคุมแบบอัตโนมัติรูปแบบเลขแปด

3.2.6 ขั้นตอนการทดลอง GY-521 Module MPU6050

3.2.6.1 ขั้นตอนการทดลอง GY-521 Module MPU6050

1. ทำการทดลอง GY-521 Module MPU6050 ใช้งานได้หรือไม่
2. ทดลองการส่งค่าจาก GY-521 Module MPU6050 เข้ามาที่อุปกรณ์ตัวรับสัญญาณบอร์ด Arduino Mega2560
3. เขียนโปรแกรมควบคุมรับ-ส่งค่าจาก GY-521 Module MPU6050 เพื่อไปควบคุมตัวรถวีลแชร์



รูปที่ 3.17 ทดลอง GY-521 Module MPU6050

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.18 โปรแกรมควบคุมรับ-ส่งค่าจาก GY-521 Module MPU6050 เพื่อไปควบคุมตัวรถวีลแชร์

3.2.7 ขั้นตอนการทดลอง Module SD Card

1. สั่งซื้ออุปกรณ์ Memory SD Card และเสียบเข้าไปยังตัว Module SD Card
2. ทดลองเขียนโปรแกรมเพื่อที่จะเก็บค่าต่าง ๆ และบันทึกข้อมูล
3. ผลการทดลองจากการเขียนโปรแกรมภายใน 1 วินาที เก็บได้มากสุด 46 ค่า เพื่อนำค่าที่ได้ไปวิเคราะห์ในการออกแบบเพื่อควบคุม

รูปที่ 3.19 Module SD Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File Edit Format View Help

```
The Data Logger: Counter Num 12 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.72 Y : 5.29 Z : 8.14 Gyroscope :  X : -0.06 Y : 0.02 Z : -0.02
The Data Logger: Counter Num 13 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.72 Y : 5.49 Z : 8.33 Gyroscope :  X : -0.06 Y : 0.01 Z : -0.01
The Data Logger: Counter Num 14 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.64 Y : 5.45 Z : 8.24 Gyroscope :  X : -0.06 Y : 0.02 Z : 0.00
The Data Logger: Counter Num 15 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.61 Y : 5.35 Z : 8.14 Gyroscope :  X : -0.06 Y : 0.02 Z : 0.00
The Data Logger: Counter Num 16 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.56 Y : 5.39 Z : 8.37 Gyroscope :  X : -0.06 Y : 0.03 Z : -0.00
The Data Logger: Counter Num 17 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.58 Y : 5.40 Z : 8.39 Gyroscope :  X : -0.06 Y : 0.01 Z : -0.02
The Data Logger: Counter Num 18 Speed 1 : 0 RPM  Speed 2 : 0 RPM  Accerlometer :  X : 0.14 Y : 5.71 Z : 8.22 Gyroscope :  X : -0.05 Y : -0.09 Z : -0.14
The Data Logger: Counter Num 19 Speed 1 : 1 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.60 Y : 5.12 Z : 8.13 Gyroscope :  X : -0.04 Y : -0.10 Z : -0.22
The Data Logger: Counter Num 20 Speed 1 : 1 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.62 Y : 4.09 Z : 9.66 Gyroscope :  X : -0.06 Y : -0.09 Z : -0.19
The Data Logger: Counter Num 21 Speed 1 : 1 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -1.04 Y : 4.31 Z : 9.18 Gyroscope :  X : -0.06 Y : -0.14 Z : -0.24
The Data Logger: Counter Num 22 Speed 1 : 1 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.92 Y : 5.52 Z : 7.94 Gyroscope :  X : -0.04 Y : -0.17 Z : -0.25
The Data Logger: Counter Num 23 Speed 1 : 1 RPM  Speed 2 : 0 RPM  Accerlometer :  X : -0.24 Y : 5.97 Z : 7.94 Gyroscope :  X : -0.04 Y : -0.05 Z : -0.21
The Data Logger: Counter Num 24 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -1.06 Y : 5.00 Z : 8.21 Gyroscope :  X : -0.06 Y : 0.03 Z : -0.09
The Data Logger: Counter Num 25 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.49 Y : 5.62 Z : 8.14 Gyroscope :  X : -0.08 Y : -0.03 Z : -0.06
The Data Logger: Counter Num 26 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : 0.63 Y : 6.27 Z : 7.46 Gyroscope :  X : -0.06 Y : -0.05 Z : -0.05
The Data Logger: Counter Num 27 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -1.27 Y : 5.94 Z : 8.12 Gyroscope :  X : -0.02 Y : -0.12 Z : -0.23
The Data Logger: Counter Num 28 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.80 Y : 4.97 Z : 8.44 Gyroscope :  X : -0.02 Y : -0.20 Z : -0.18
The Data Logger: Counter Num 29 Speed 1 : 2 RPM  Speed 2 : 2 RPM  Accerlometer :  X : 0.39 Y : 2.48 Z : 9.73 Gyroscope :  X : -0.09 Y : -0.09 Z : -0.24
The Data Logger: Counter Num 30 Speed 1 : 2 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.62 Y : 5.12 Z : 9.46 Gyroscope :  X : -0.07 Y : -0.11 Z : -0.25
The Data Logger: Counter Num 31 Speed 1 : 2 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.67 Y : 5.76 Z : 7.16 Gyroscope :  X : -0.05 Y : -0.18 Z : -0.21
The Data Logger: Counter Num 32 Speed 1 : 2 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.11 Y : 5.90 Z : 8.00 Gyroscope :  X : -0.03 Y : -0.09 Z : -0.27
The Data Logger: Counter Num 33 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.77 Y : 5.31 Z : 8.54 Gyroscope :  X : -0.05 Y : -0.16 Z : -0.30
The Data Logger: Counter Num 34 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.86 Y : 5.26 Z : 8.87 Gyroscope :  X : -0.04 Y : -0.19 Z : -0.33
The Data Logger: Counter Num 35 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.36 Y : 4.84 Z : 7.78 Gyroscope :  X : -0.02 Y : -0.20 Z : -0.33
The Data Logger: Counter Num 36 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.50 Y : 5.48 Z : 9.14 Gyroscope :  X : -0.03 Y : -0.19 Z : -0.33
The Data Logger: Counter Num 37 Speed 1 : 1 RPM  Speed 2 : 1 RPM  Accerlometer :  X : -0.53 Y : 5.28 Z : 8.20 Gyroscope :  X : -0.02 Y : -0.22 Z : -0.36
The Data Logger: Counter Num 38 Speed 1 : 3 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.09 Y : 5.16 Z : 7.76 Gyroscope :  X : -0.02 Y : -0.22 Z : -0.39
The Data Logger: Counter Num 39 Speed 1 : 3 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.87 Y : 5.56 Z : 9.05 Gyroscope :  X : -0.03 Y : -0.24 Z : -0.40
The Data Logger: Counter Num 40 Speed 1 : 3 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.52 Y : 4.84 Z : 7.82 Gyroscope :  X : -0.01 Y : -0.25 Z : -0.42
The Data Logger: Counter Num 41 Speed 1 : 3 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.80 Y : 4.79 Z : 7.89 Gyroscope :  X : -0.04 Y : -0.25 Z : -0.46
The Data Logger: Counter Num 42 Speed 1 : 3 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.22 Y : 5.62 Z : 8.90 Gyroscope :  X : -0.02 Y : -0.29 Z : -0.46
The Data Logger: Counter Num 43 Speed 1 : 4 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.34 Y : 4.89 Z : 7.90 Gyroscope :  X : -0.02 Y : -0.30 Z : -0.49
The Data Logger: Counter Num 44 Speed 1 : 4 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.52 Y : 5.52 Z : 8.94 Gyroscope :  X : -0.01 Y : -0.32 Z : -0.52
The Data Logger: Counter Num 45 Speed 1 : 4 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.59 Y : 4.96 Z : 8.11 Gyroscope :  X : -0.02 Y : -0.33 Z : -0.54
The Data Logger: Counter Num 46 Speed 1 : 4 RPM  Speed 2 : 2 RPM  Accerlometer :  X : -0.62 Y : 5.53 Z : 8.80 Gyroscope :  X : -0.01 Y : -0.35 Z : -0.55
```

รูปที่ 3.20 ผลการทดลองจากการเขียนโปรแกรมเพื่อเก็บค่ามาวิเคราะห์

รูปที่ 3.21 การทดลองจากการเขียนโปรแกรมให้รถเคลื่อนที่อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลอง

4.1 กล่าวนำ

การทดลอง การศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติ ประกอบไปด้วย การทดลองการทำงานของระบบต่าง ได้แก่ ทิศทางการขับเคลื่อน และการทำงานของตัวนับจำนวนรอบ Encoder และการทำงานของ GY-521 หรือที่เรียกว่า ไจโรสโคป (MPU6050) เพื่อทำการเก็บค่าไว้ใน Module SD Card การทดลองควบคุมให้ตัวรถเดินอัตโนมัติตามคำสั่งที่กำหนดไว้ให้เดินเป็นเลขแปด เดินเป็นสี่เหลี่ยม และ เดินเป็นวงกลม

4.2 ผลการทดลอง

4.2.1 การทำงานของระบบนับจำนวนรอบ Encoder

ตารางที่ 4.1 ตารางผลการทดลองการทำงานของระบบนับจำนวนรอบ Encoder โดยการทำงานและทำการบันทึกผล

ครั้งที่	การเคลื่อนที่ของรถวีลแชร์	ผลการทดลอง	
		สถานะ	การแสดงผล
1	เดินหน้าตรง	ทำงาน	แสดงผลได้
2	เดินถอยหลังตรง	ทำงาน	แสดงผลได้
3	เดินซ้าย	ทำงาน	แสดงผลได้
4	เดินขวา	ทำงาน	แสดงผลได้

จากตางรางที่ 4.1 การทดลองการทำงาน ของระบบนับจำนวนรอบ Encoder โดยสั่งให้มอเตอร์ทั้ง 2 ฝั่งเดินในหลายรูปแบบคือ เดินหน้า, เดินถอยหลัง , เดินซ้าย , เดินขวา แล้วจึงทำให้ทราบค่าของ RPM (Speed) Pulses , Speed ของตัวมอเตอร์ทั้ง 2 ในการทำงานเพื่อแสดงค่าเป็น Graph Serial Plotter เพื่อทำการเก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทำงานของ GY-521

ตารางที่ 4.2 ตารางผลการทดลองการทำงานของ GY-521 เพื่อทำการเก็บค่าไว้ใน Memory Module Cards

ครั้งที่	การทำงานของรถวีลแชร์	ผลการทดลอง	
		สถานะ	การการแสดงผล
1	เดินหน้าตรง	ทำงาน	แสดงผลถูกต้อง
2	เดินถอยหลังตรง	ทำงาน	แสดงผลถูกต้อง
3	เลี้ยวซ้าย	ทำงาน	แสดงผลถูกต้อง
4	เลี้ยวขวา	ทำงาน	แสดงผลถูกต้อง

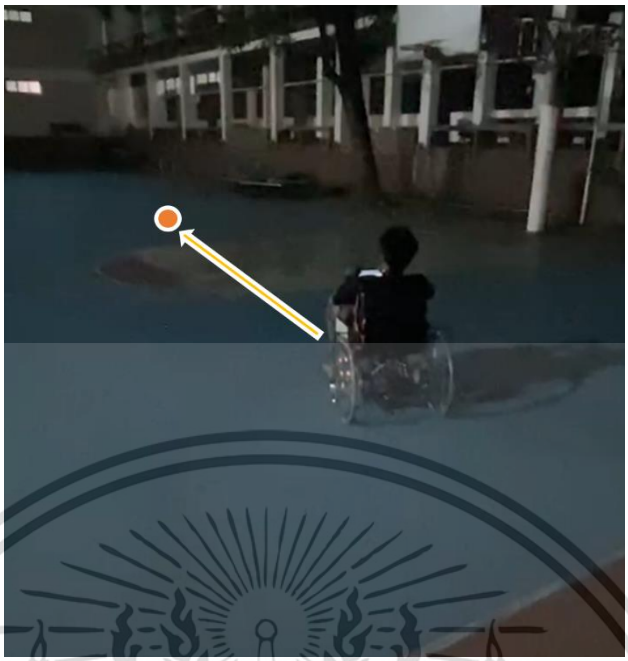
จากตางรางที่ 4.2 การทดลองการทำงานของ GY-521 เพื่อทำการเก็บค่าไว้ใน Memory Module Cards โดยสั่งให้มอเตอร์ทั้ง 2 ฝั่งเดินในหลายรูปแบบคือ เดินหน้า, เดินถอยหลัง, เลี้ยวซ้าย, เลี้ยวขวา แล้วเราจะสามารถรู้ค่าของ RPM (Speed), Accelerometer = (X,Y,Z), Gyroscope = (X,Y,Z)

4.2.3 การขับเคลื่อนตามคำสั่งให้เดินอัตโนมัติ

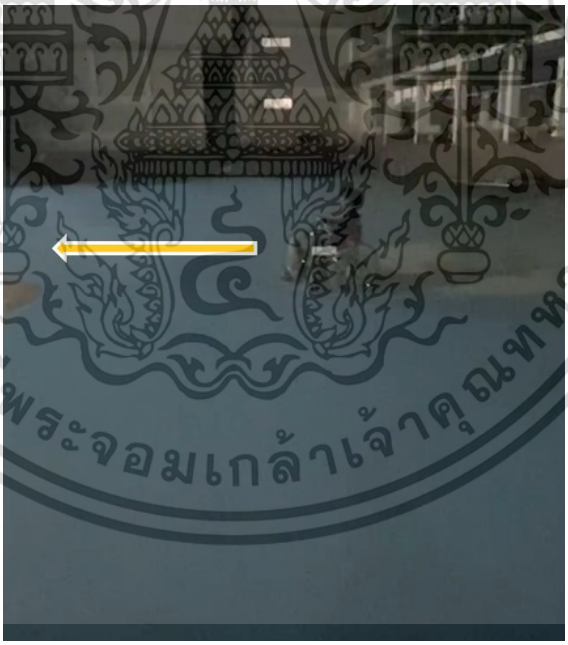
ตารางที่ 4.3 ตารางผลการทดลองการขับเคลื่อนรถวีลแชร์เป็นสี่เหลี่ยมเมื่อทำการออกคำสั่ง

การทดลองครั้งที่	การเคลื่อนที่ของรถวีลแชร์	ผลการทดลอง	
		กำหนดการเดิน	การแสดงผล
1	รูปแบบสี่เหลี่ยม2X2เมตร	รูปแบบสี่เหลี่ยม	แสดงผลถูกต้อง
2	รูปแบบสี่เหลี่ยม2X2เมตร	รูปแบบสี่เหลี่ยม	แสดงผลถูกต้อง
3	รูปแบบสี่เหลี่ยม2X2เมตร	รูปแบบสี่เหลี่ยม	แสดงผลถูกต้อง
4	รูปแบบสี่เหลี่ยม2X2เมตร	รูปแบบสี่เหลี่ยม	แสดงผลถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

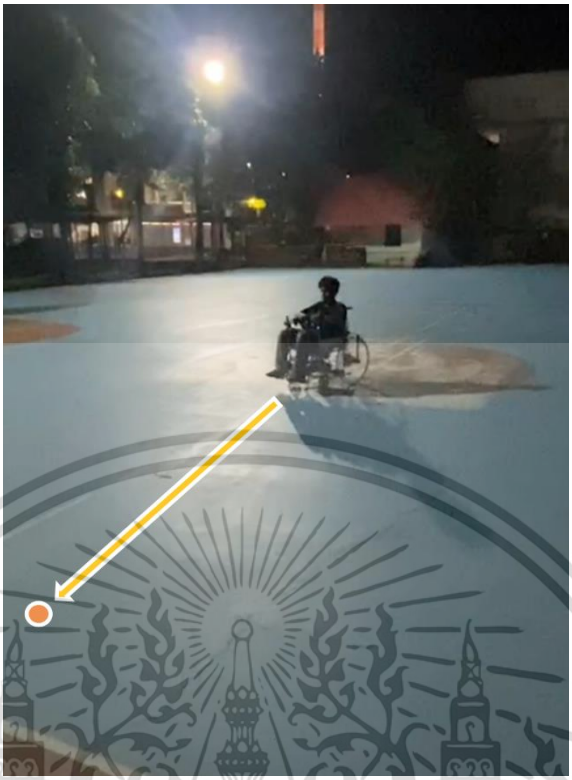


รูปที่ 4.1 รูปแบบสี่เหลี่ยม2X2เมตร จุด A

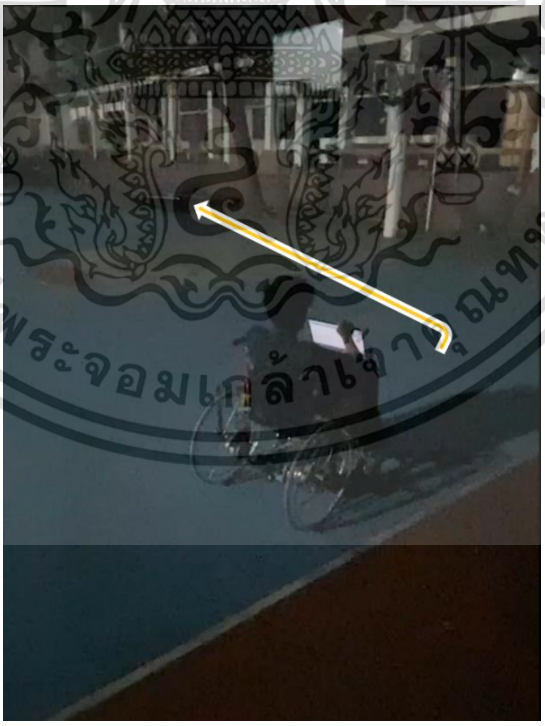


รูปที่ 4.2 รูปแบบสี่เหลี่ยม2X2เมตร **จุด B**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 รูปแบบสี่เหลี่ยม2X2เมตร **จุด C**



รูปที่ 4.4 รูปแบบสี่เหลี่ยม2X2เมตร **จุด D**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตางรางที่ 4.3 การทดลองการเคลื่อนที่เป็นสี่เหลี่ยมด้วยการป้อนคำสั่ง ผลการทดลองจะสั่งให้ตัวหมุนและทำการเดินไปจนครบหนึ่งรอบก็จะออกมาเป็นสี่เหลี่ยมทำการทดลองจำนวนสี่รอบเพื่อที่จะยืนยันว่าออกมาเป็นสี่เหลี่ยม ผลการทดลองคือตัวรถวิ่งเป็นสี่เหลี่ยมได้ตามคำสั่ง

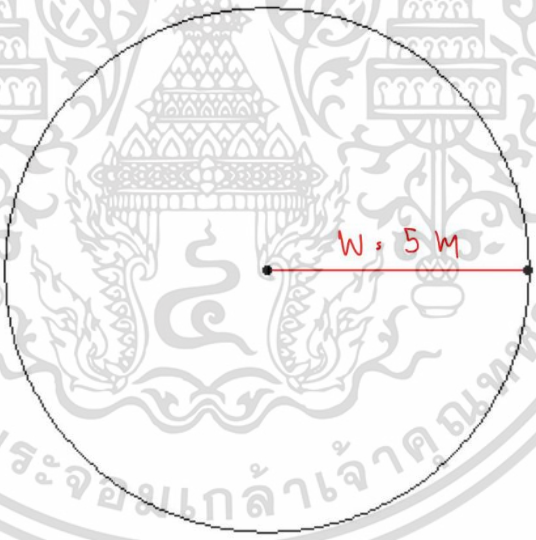
ตารางที่ 4.4 ตารางผลการทดลองการขับเคลื่อนรถวีลแชร์เป็นวงกลมเมื่อทำการออกคำสั่ง

ครั้งที่	การเคลื่อนที่เป็นวงกลม	ผลการทดลอง	
		กำหนดการเดิน	การแสดงผล
1	5 เมตร	รูปแบบวงกลม	แสดงผลถูกต้อง
2	5 เมตร	รูปแบบวงกลม	แสดงผลถูกต้อง
3	5 เมตร	รูปแบบวงกลม	แสดงผลถูกต้อง

รูปที่ 4.5 รูปแบบวงกลมส่วนโค้ง 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 รูปแบบวงกลมส่วนโค้ง 2



รูปที่ 4.7 ระยะความกว้างของรูปแบบวงกลม

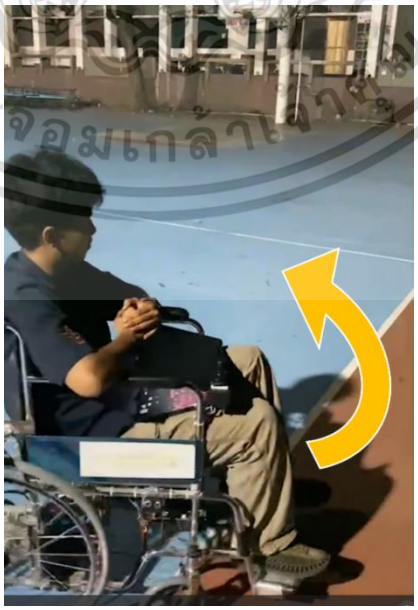
จากตางรางที่ 4.4 การทดลองการเคลื่อนที่เป็นวงกลมด้วยการออกคำสั่ง ผลการทดลอง ทำการสั่งการหมุนให้เป็นวงกลมทั้งสามรอบเพื่อยืนยันว่าออกมากลมตามที่ต้องการ ผลของการทดลองคือตัวรถวิ่งเป็นวงกลมอย่างที่ได้ออกคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 ตารางผลการทดลองการขับเคลื่อนรถวีลแชร์เป็นเลขแปดเมื่อทำการออกคำสั่ง

ครั้งที่	การเคลื่อนที่เป็นเลขแปด	ผลการทดลอง	
		ทิศทางการเคลื่อนที่	การแสดงผล
1	1.5 เมตร	รูปแบบเลขแปด	แสดงผลถูกต้อง
2	1.5 เมตร	รูปแบบเลขแปด	แสดงผลถูกต้อง
3	1.5 เมตร	รูปแบบเลขแปด	แสดงผลถูกต้อง
4	1.5 เมตร	รูปแบบเลขแปด	แสดงผลถูกต้อง

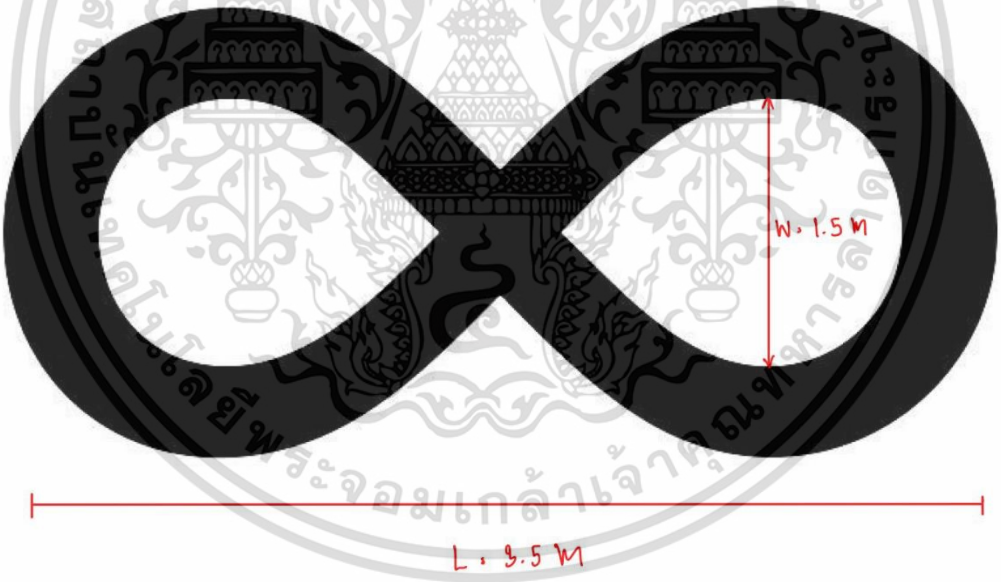
รูปที่ 4.8 รูปแบบเลขแปด 1



รูปที่ 4.9 รูปแบบเลขแปด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.10 รูปแบบเลขแปด 3



รูปที่ 4.11 ระยะความกว้างของรูปแบบเลขแปด

จากตางรางที่ 4.5 การทดลองการเคลื่อนที่เป็นเลขด้วยการออกคำสั่ง ผลการทดลอง ทำการสั่งการเดินทางของวีลแชร์เป็นเลขแปดเพื่อยืนยันว่าออกมากลมตามที่ต้องการ ผลของการทดลองคือตัวรถวิ่งเป็นวงกลมอย่างที่ได้ออกคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.6 ตารางผลการทดลองการเกิดค่าความผิดพลาดในการทำงานขับเคลื่อนรถวีลแชร์เป็นเลขแปดเมื่อทำการออกคำสั่งการวนหลายรอบ

ครั้งที่	การเคลื่อนที่เป็นเลขแปด	ผลการทดลอง	
		ทิศทางการเคลื่อนที่	ปัญหาที่พบเห็น
1	รูปแบบเลขแปด วน 1 รอบ	รูปแบบเลขแปด	แสดงผลถูกต้อง
2	รูปแบบเลขแปด วน 2 รอบ	รูปแบบเลขแปด	การเคลื่อนที่จากจุดเดิมคาดเคลื่อนเล็กน้อย 0.15 m
3	รูปแบบเลขแปด วน 3 รอบ	รูปแบบเลขแปด	เลข แปด การเคลื่อนที่จากจุดเดิมเล็กน้อย 0.2 m
4	รูปแบบเลขแปด วน 4 รอบ	รูปแบบเลขแปด	เลข แปด การเคลื่อนที่จากจุดเดิมมากพอสมควร 0.3 m

จากตางรางที่ 4.6 การทดลองการเคลื่อนที่เป็นเลขด้วยการออกคำสั่ง ผลการทดลอง ทำการสั่งการเดินทางของวีลแชร้เป็นเลขแปดเพื่อยืนยันว่าออกมากลมตามที่ต้องการ โดยผลการทดลองสามรถดูได้จากตารางที่4.6 โดยสังเกตจาการเดินหลายครั้ง ทำให้ทราบว่ารถวีลแชร์สารถเดินเป็นลข แปดได้แต่ ไม่สามารถทำให้ตรงเหมือนกันในทุกรอบได้ เนื่องจากล้อหน้าของวีลแชล์เป็น มุม 360 องศาทั้งทางด้านซ้ายและด้านขวาทำให้รถเกิดการเดินผิดพลาดไปบ้าง ต้องทำการแก้ไขล้อหน้าในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการดำเนินงานและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการทดลอง การศึกษาการควบคุมวีลแชร์อัจฉริยะอัตโนมัติ การขับเคลื่อนตามคำสั่งให้เดินอัตโนมัติมีอยู่สามรูปแบบได้แก่ การทดลองการขับเคลื่อนรถวีลแชร์ให้เป็น สี่เหลี่ยม วงกลม และ เลขแปด ขั้นตอนถัดไปเมื่อทำการป้อนโปรแกรมรูปแบบสี่เหลี่ยมเข้าไปตัวไมโครคอนโทรลเลอร์แล้วตัวไมโครคอนโทรลเลอร์จะนำไปประมวณผลและสั่งให้ไดร์ฟจ่ายไฟให้กับมอเตอร์เมื่อมอเตอร์หมุนครบหนึ่งรอบจะออกมาเป็นสี่เหลี่ยมจากนั้นทำการเดินวนรอบสองเพื่อยืนยันว่าออกมาเป็นสี่เหลี่ยมหรือไม่ ถัดไปเป็นการทดลองการขับเคลื่อนรถวีลแชร์เป็นรูปแบบวงกลมทำการป้อนโปรแกรมในรูปแบบวงกลมเข้าไปที่ตัวไมโครคอนโทรลเลอร์ จะทำการสั่งให้มอเตอร์หมุนให้เป็นวงกลมทั้งสามรอบเพื่อยืนยันว่าออกมาเป็นวงกลมตามที่ต้องการหรือไม่ อย่างสุดท้ายของการทดลองเดินอัตโนมัติเป็นการทดลองการขับเคลื่อนรถวีลแชร์เป็นรูปแบบเลขแปดเมื่อทำการป้อนโปรแกรมเลขแปดเข้าไปที่ตัวไมโครคอนโทรลเลอร์จะทำการสั่งให้มอเตอร์หมุนให้เป็นเลขแปดทั้งสองรอบเพื่อยืนยันว่าเลขแปดออกมาตามที่ต้องการหรือไม่โดยมอเตอร์ทั้ง 2 จะเดินอัตโนมัติได้ตามที่เรากำหนด ผลการทดลองคือตัวรถเคลื่อนที่เป็นรูปแบบ สี่เหลี่ยม วงกลม และ เลขแปด ได้ตามคำสั่ง โดยการจัดทำครั้งนี้ได้บรรลุตามวัตถุประสงค์ของผู้จัดทำจัดโครงการ

5.2 ปัญหาที่พบและแนวทางการแก้ไข

5.2.1 ปัญหาที่พบขณะทำการทดลอง

1. รถวีลแชร์มีโครงสร้างเดิมที่ซับซ้อนทำให้ติดตั้ง Encoder ไม่ได้ เพราะว่าตัวล้อเคลื่อนที่อยู่ ตลอดขณะรถวีลแชร์ทำงาน
2. สายไฟกับตัวยึดเกิดความเสียหาย
3. สภาพพื้นผิวที่ทดลองไม่สม่ำเสมอ ขรุขระ

5.2.2 แนวทางการแก้ไข

1. ดัดแปลงปรับโครงสร้างรถวีลแชร์ให้สามารถประกอบติดตั้ง Encoder กับมอเตอร์ได้ทั้ง 2 ล้อโดยตรง
2. สายไฟกับตัวยึดทำการแก้ไขโดยเดินสายไฟให้ถูกต้องและปลอดภัยมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

สำหรับการพัฒนาโครงการต่อในอนาคต ควรจะพัฒนาระบบโครงสร้างรถเนื่องจากปัญหาล้อหน้ามีองศา 360 องศาจึงทำให้ขณะเลี้ยวเกิดการเลี้ยวที่ผิดพลาดเนื่องจากโครงสร้างเดิมไม่เหมาะสม ในอนาคตควรจัดหาวิธีการหรือเปลี่ยนล้อด้านหน้าทั้ง 2 ล้อให้เหมาะสมกับรถวีลแชร์อัตโนมัติ ในส่วนของระบบควบคุมอัตโนมัติให้มีประสิทธิภาพ โดยนำเอาการใช้ PID control มาประยุกต์ใช้กับรถวีลแชร์เดินอัตโนมัติที่ผู้จัดทำได้สร้างขึ้นมาจะทำให้รถวีลแชร์เดินอัตโนมัติเดินได้มีประสิทธิภาพโดยนำสมการและค่าต่างๆ ที่ได้มาประยุกต์ใช้ในอนาคตและยังมีแบตเตอรี่ที่ควรเพิ่มขนาด มากกว่าเดิมจะทำให้สามารถทดลองและทำงานได้ในระยะเวลาที่มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] The invention, “ทำความรู้จักกับบอร์ด Mega 2560”, 2564 [ระบบออนไลน์] แหล่งที่มา: https://www.ai-corporation.net/2021/11/25/arduino-mega2560/

[2] บริษัท แฟ็คโตมาร์ท จำกัด, “เอ็นโค้ดเดอร์คืออะไร”, 2562 [ระบบออนไลน์] แหล่งที่มา: https://mall.factomart.com/what-is-encoder/

[3] SolarsmileKnowledge, “ความหมายและชนิดของแบตเตอรี่”2561[ระบบออนไลน์] แหล่งที่มา: https://solarsmileknowledge.com/battery/

[4] RANGSIYABLOG “การเคลื่อนที่แนวตรง” [ระบบออนไลน์]2559 แหล่งที่มา: https://rectilinearmotion.wordpress.com/

[5] lnwShop, “การเคลื่อนที่แบบโพรเจกไทล์”2559 [ระบบออนไลน์]แหล่งที่มา: https://www.aballtechno.com/article/

[6] บริษัท อินทัช โปรดักส์ จำกัด, “Motor ทำงานอย่างไร”2564[ระบบออนไลน์] แหล่งที่มา: https://www.intouch-products.com/17214348/

[7] lnwShop, “การใช้งานโมดูลSD Card อ่าน/เขียน”2559 [ระบบออนไลน์]แหล่งที่มา: https://www.artronshop.co.th/article/13/

[8] lnwShop, “การใช้งานโมดูล GY-521”2559 [ระบบออนไลน์]แหล่งที่มา: https://www.ec-bot.com/product/69/mpu-6050-/

[9] lnwShop, “การควบคุมไดร์ฟมอเตอร์”2562 [ระบบออนไลน์]แหล่งที่มา: https://www.artronshop.co.th/article/86/

[10] CyberTice “การเขียนโปรแกรม Arduino IDE” 2554 [ระบบออนไลน์]: https://www.cybertice.com/article/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

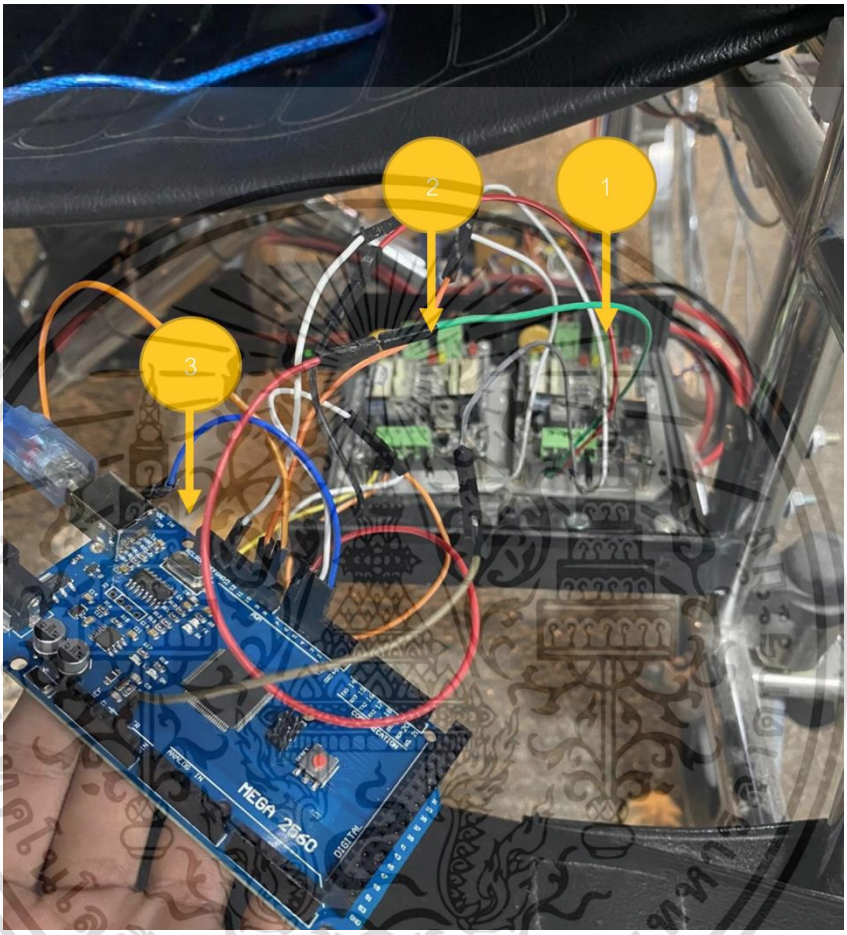
ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งาน Smart Wheelchair

ก.1 อธิบายฟังก์ชั่นของรถ

ต่อสายจาก Arduino เข้า drive ขับมอเตอร์ทั้งสองฝั่ง



รูปที่ ก.1 ต่อสายจาก Arduino เข้า drive ขับมอเตอร์ทั้งสองฝั่ง

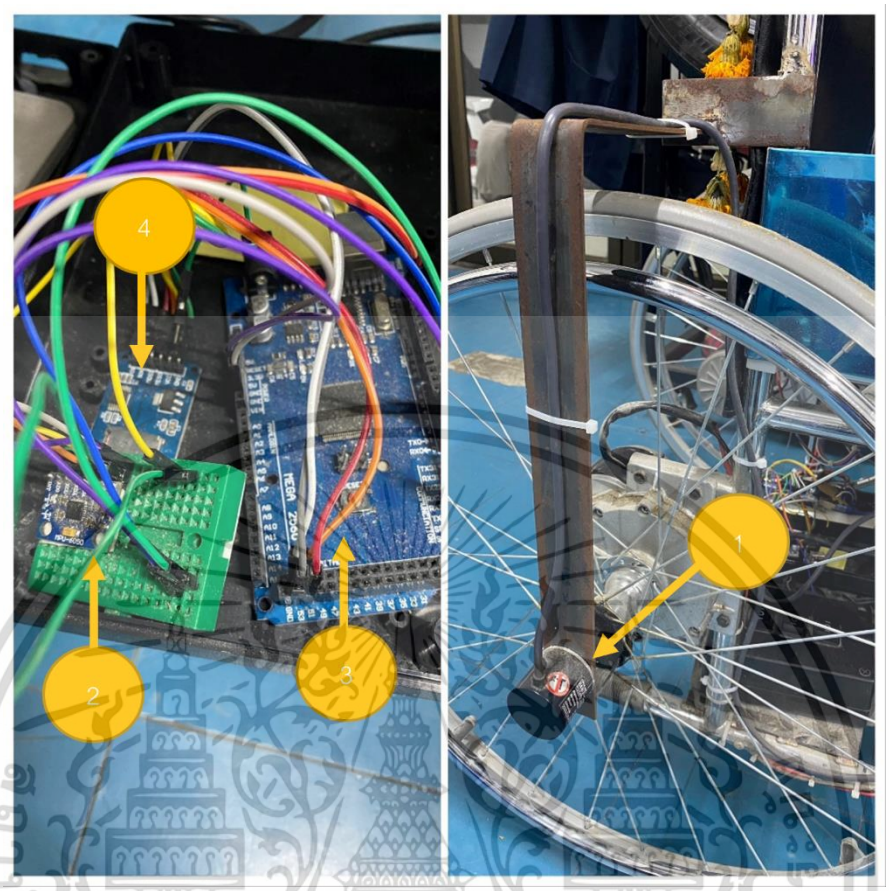
หมายเลข 1 Drive ขับมอเตอร์ kmitl 1

หมายเลข 2 Drive ขับมอเตอร์ kmitl 2

หมายเลข 3 Arduino Mega 2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดเก็บข้อมูล Encoder และ GY-521 เข้า SD card



รูปที่ ก.2 ชุดเก็บข้อมูล Encoder และ GY-521 เข้า SD card

หมายเลข 1 Rotary Encoder Omron

หมายเลข 2 Module GY-521

หมายเลข 3 Arduino Mega 2560

หมายเลข 4 Module Micro SD card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งการทำงาน

ข.1 โปรแกรมคำสั่งให้เดินตามคำสั่งต่างๆ

โค้ดคำสั่งเดินเป็นสี่เหลี่ยม

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
Adafruit_MPU6050 mpu;
#include <SD.h>
#include <SPI.h>
int chipSelect = 53;
File file;
#define ENC_COUNT_REV 374
#define PWM1 2
#define PWM2 19
#define ENC_IN1 3
#define ENC_IN2 18
int speedcontrol1 = 0;
int speedcontrol2 = 0;
volatile long encoderValue1 = 0;
volatile long encoderValue2 = 0;
int rpm1 = 0;
int rpm2 = 0;
int motorPwm1 = 0;
int motorPwm2 = 0;
  sensors_event_t a, g, temp;
//SD Card
#define enA 9
#define in1 4
#define in2 5
#define enB 10
#define in3 6
#define in4 7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
String datalog ="";
unsigned long period1 = 1000; //ระยะเวลาที่ต้องการรอ
unsigned long last_time1 = 0;
 int onesecond = 0;
void setup(){
  Serial.begin(9600);
  pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
  pinMode(bt, INPUT);
  Wire.begin();

  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens
    Serial.println("Adafruit MPU6050 test!");
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");
  mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
  Serial.print("Accelerometer range set to: ");
  switch (mpu.getAccelerometerRange()) {
  case MPU6050_RANGE_2_G:
    Serial.println("+-2G");
    break;
  case MPU6050_RANGE_4_G:
    Serial.println("+-4G");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  break;
case MPU6050_RANGE_8_G:
  Serial.println("+-8G");
  break;
case MPU6050_RANGE_16_G:
  Serial.println("+-16G");
  break;
}
mpu.setGyroRange(MPU6050_RANGE_250_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
  Serial.println("+- 250 deg/s");
  break;
case MPU6050_RANGE_500_DEG:
  Serial.println("+- 500 deg/s");
  break;
case MPU6050_RANGE_1000_DEG:
  Serial.println("+- 1000 deg/s");
  break;
case MPU6050_RANGE_2000_DEG:
  Serial.println("+- 2000 deg/s");
  break;
}

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
  Serial.println("260 Hz");
  break;
case MPU6050_BAND_184_HZ:
  Serial.println("184 Hz");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  break;
 case MPU6050_BAND_94_HZ:
  Serial.println("94 Hz");
  break;
 case MPU6050_BAND_44_HZ:
  Serial.println("44 Hz");
  break;
 case MPU6050_BAND_21_HZ:
  Serial.println("21 Hz");
  break;
 case MPU6050_BAND_10_HZ:
  Serial.println("10 Hz");
  break;
 case MPU6050_BAND_5_HZ:
  Serial.println("5 Hz");
  break;
 }
 //Serial.println("");
 //delay(100);


 pinMode(chipSelect,OUTPUT);
 pinMode(ENC_IN1, INPUT_PULLUP);
 pinMode(ENC_IN2, INPUT_PULLUP);
 attachInterrupt(digitalPinToInterrupt(ENC_IN1), updateEncoder1, RISING);
 attachInterrupt(digitalPinToInterrupt(ENC_IN2), updateEncoder2, 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. . .");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  file = SD.open("test.txt",FILE_WRITE);
  file.close();
  Serial.println("File Created!");
}
int k=0;
void loop(){

  datalog = " Counter Num " + String(k) + " Speed 1 : " + String(rpm1)+ " RPM "
+ " Speed 2 : " + String(rpm2)+ " RPM " + " Accerlometer : " + " X : "
  + String(a.acceleration.x) + " Y : " + String(a.acceleration.y) + " Z : " +
String(a.acceleration.z)+ " Gyroscope : " + " X : " + String(g.gyro.x) + " Y : " +
String(g.gyro.y)
  + " Z : " + String(g.gyro.z);
    if( millis() - last_time1 > period1) {
     last_time1 = millis();
     onesecond++;

    }
if(onesecond>5){
  onesecond=0;
  k=0;
}
    if(onesecond==3){

      if(SD.exists("test.txt")){
      Serial.println("File Exists!");
      file = SD.open("test.txt",FILE_WRITE);
      if(file){
        Serial.print(" i ");
        file.println("The Data Logger:" + datalog);
        k++;
        file.close();
        }else{Serial.println("File Cannot Be Written To!");}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }else{Serial.println("File Dosen't Exist");
   }
  }

   motermove();
   Encode_1();
   Move();
   //Sdata();
}
void Encode_1(){
  motorPwm1 = map(analogRead(speedcontrol1), 0, 1023, 0, 255);
  motorPwm2 = map(analogRead(speedcontrol2), 0, 1023, 0, 255);
  analogWrite(PWM1, motorPwm1);
  analogWrite(PWM2, motorPwm2);
  static unsigned long timer = millis();
  const unsigned long per = 100;
   if(millis()- timer >= per){
   timer += per;
    rpm1 = (float)(encoderValue1 * 60 / ENC_COUNT_REV);
  rpm2 = (float)(encoderValue2 * 60 / ENC_COUNT_REV);
  // Only update display when there is a reading
  if (motorPwm1 > 0 || rpm1 > 0,motorPwm2 > 0 || rpm2 > 0){
   Serial.print(" SPEED1: ");
   Serial.print(rpm1);
   Serial.print(" RPM");
   Serial.print('\t');
   Serial.print(" SPEED2: ");
   Serial.print(rpm2);
   Serial.println(" RPM");
  }
  encoderValue1 = 0;
  encoderValue2 = 0;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
void updateEncoder1()
{
    encoderValue1++;
}
void updateEncoder2(){
    encoderValue2++;
  }
void Move(){
//  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

 Serial.println("");
 delay(100);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Sdata(){
    datalog = " Counter Num " + String(onesecond) + " Speed 1 : " +
String(rpm1)+ " RPM " + " Speed 2 : " + String(rpm2)+ " RPM " + " Accerlometer
: " + " X : "
  + String(a.acceleration.x) + " Y : " + String(a.acceleration.y) + " Z : " +
String(a.acceleration.z)+ " Gyroscope : " + " X : " + String(g.gyro.x) + " Y : " +
String(g.gyro.y)
  + " Z : " + String(g.gyro.z);
    if( millis() - last_time1 > period1) {
     last_time1 = millis();
     onesecond++;
      if(SD.exists("test.txt")){
      Serial.println("File Exists!");
      file = SD.open("test.txt",FILE_WRITE);
      if(file){
        file.println("The Data Logger:" + datalog);
        file.close();
        }else{Serial.println("File Cannot Be Written To!");}
       }else{Serial.println("File Dosen't Exist");
      }
 }
}
void motermove(){
   buttonState = digitalRead(bt);
if (bt == HIGH){
  buttonState = digitalRead(bt);
if (bt == HIGH){
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(enA, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(enB, 255);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay(5000);
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(enA, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(enB, 255);

delay(1000);

delay(12000);
 }
}
```

โค้ดคำสั่งเดินเป็นวงกลม

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
Adafruit_MPU6050 mpu;

#include <SD.h>
#include <SPI.h>
int chipSelect = 53;
File file;

#define ENC_COUNT_REV 374
#define PWM1 2
#define PWM2 19
#define ENC_IN1 3
#define ENC_IN2 18
int speedcontrol1 = 0;
int speedcontrol2 = 0;
volatile long encoderValue1 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
volatile long encoderValue2 = 0;
int rpm1 = 0;
int rpm2 = 0;
int motorPwm1 = 0;
int motorPwm2 = 0;
  sensors_event_t a, g, temp;
//SD Card
#define enA 9
#define in1 4
#define in2 5
#define enB 10
#define in3 6
#define in4 7
String datalog ="";
unsigned long period1 = 1000; //ระยะเวลาที่ต้องการรอ
unsigned long last_time1 = 0;
 int onesecond = 0;
void setup(){
 Serial.begin(9600);
pinMode(enA, OUTPUT);
 pinMode(enB, OUTPUT);
 pinMode(in1, OUTPUT);
 pinMode(in2, OUTPUT);
 pinMode(in3, OUTPUT);
 pinMode(in4, OUTPUT);
 pinMode(bt, INPUT);
 Wire.begin();

 while (!Serial)
   delay(10); // will pause Zero, Leonardo, etc until serial console opens
   Serial.println("Adafruit MPU6050 test!");
 if (!mpu.begin()) {
   Serial.println("Failed to find MPU6050 chip");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  while (1) {
    delay(10);
  }
}
Serial.println("MPU6050 Found!");
mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
Serial.print("Accelerometer range set to: ");
switch (mpu.getAccelerometerRange()) {
case MPU6050_RANGE_2_G:
  Serial.println("+-2G");
  break;
case MPU6050_RANGE_4_G:
  Serial.println("+-4G");
  break;
case MPU6050_RANGE_8_G:
  Serial.println("+-8G");
  break;
case MPU6050_RANGE_16_G:
  Serial.println("+-16G");
  break;
}
mpu.setGyroRange(MPU6050_RANGE_250_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
  Serial.println("+- 250 deg/s");
  break;
case MPU6050_RANGE_500_DEG:
  Serial.println("+- 500 deg/s");
  break;
case MPU6050_RANGE_1000_DEG:
  Serial.println("+- 1000 deg/s");
  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case MPU6050_RANGE_2000_DEG:
  Serial.println("+- 2000 deg/s");
  break;
}

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
  Serial.println("260 Hz");
  break;
case MPU6050_BAND_184_HZ:
  Serial.println("184 Hz");
  break;
case MPU6050_BAND_94_HZ:
  Serial.println("94 Hz");
  break;
case MPU6050_BAND_44_HZ:
  Serial.println("44 Hz");
  break;
case MPU6050_BAND_21_HZ:
  Serial.println("21 Hz");
  break;
case MPU6050_BAND_10_HZ:
  Serial.println("10 Hz");
  break;
case MPU6050_BAND_5_HZ:
  Serial.println("5 Hz");
  break;
}
//Serial.println("");
//delay(100);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  pinMode(chipSelect,OUTPUT);
  pinMode(ENC_IN1, INPUT_PULLUP);
  pinMode(ENC_IN2, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(ENC_IN1), updateEncoder1, RISING);
  attachInterrupt(digitalPinToInterrupt(ENC_IN2), updateEncoder2, 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("test.txt",FILE_WRITE);
  file.close();
  Serial.println("File Created!");
}
int k=0;
void loop(){

  datalog = " Counter Num " + String(k) + " Speed 1 : " + String(rpm1)+ " RPM "
+ " Speed 2 : " + String(rpm2)+ " RPM " + " Accerlometer : " + " X : "
  + String(a.acceleration.x) + " Y : " + String(a.acceleration.y) + " Z : " +
String(a.acceleration.z)+ " Gyroscope : " + " X : " + String(g.gyro.x) + " Y : " +
String(g.gyro.y)
  + " Z : " + String(g.gyro.z);
    if( millis() - last_time1 > period1) {
     last_time1 = millis();
     onesecond++;


    }
if(onesecond>5){
  onesecond=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  k=0;
}
    if(onesecond==3){

      if(SD.exists("test.txt")){
      Serial.println("File Exists!");
      file = SD.open("test.txt",FILE_WRITE);
      if(file){
        Serial.print(" i ");
        file.println("The Data Logger:" + datalog);
        k++;
        file.close();
        }else{Serial.println("File Cannot Be Written To!");}
       }else{Serial.println("File Dosen't Exist");
      }
     }

     motermove();
     Encode_1();
     Move();
     //Sdata();
}
void Encode_1(){
    motorPwm1 = map(analogRead(speedcontrol1), 0, 1023, 0, 255);
    motorPwm2 = map(analogRead(speedcontrol2), 0, 1023, 0, 255);
    analogWrite(PWM1, motorPwm1);
    analogWrite(PWM2, motorPwm2);
    static unsigned long timer = millis();
    const unsigned long per = 100;
      if(millis()- timer >= per){
      timer += per;
        rpm1 = (float)(encoderValue1 * 60 / ENC_COUNT_REV);
    rpm2 = (float)(encoderValue2 * 60 / ENC_COUNT_REV);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    // Only update display when there is a reading
    if (motorPwm1 > 0 || rpm1 > 0,motorPwm2 > 0 || rpm2 > 0){
      Serial.print(" SPEED1: ");
      Serial.print(rpm1);
      Serial.print(" RPM");
      Serial.print('\t');
      Serial.print(" SPEED2: ");
      Serial.print(rpm2);
      Serial.println(" RPM");
    }
    encoderValue1 = 0;
    encoderValue2 = 0;
    }
  }
 void updateEncoder1()
 {
    encoderValue1++;
 }
 void updateEncoder2(){
    encoderValue2++;
  }
 void Move(){
 //  sensors_event_t a, g, temp;
   mpu.getEvent(&a, &g, &temp);

   /* Print out the values */
   Serial.print("Acceleration X: ");
   Serial.print(a.acceleration.x);
   Serial.print(", Y: ");
   Serial.print(a.acceleration.y);
   Serial.print(", Z: ");
   Serial.print(a.acceleration.z);
   Serial.println(" m/s^2");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 Serial.print("Rotation X: ");
 Serial.print(g.gyro.x);
 Serial.print(", Y: ");
 Serial.print(g.gyro.y);
 Serial.print(", Z: ");
 Serial.print(g.gyro.z);
 Serial.println(" rad/s");

 Serial.println("");
 delay(100);
 }

void Sdata(){
   datalog = " Counter Num " + String(onesecond) + " Speed 1 : " +
String(rpm1)+ " RPM " + " Speed 2 : " + String(rpm2)+ " RPM " + " Accerlometer
: " + " X : "
 + String(a.acceleration.x) + " Y : " + String(a.acceleration.y) + " Z : " +
String(a.acceleration.z)+ " Gyroscope : " + " X : " + String(g.gyro.x) + " Y : " +
String(g.gyro.y)
 + " Z : " + String(g.gyro.z);
   if( millis() - last_time1 > period1) {
    last_time1 = millis();
    onesecond++;
     if(SD.exists("test.txt")){
     Serial.println("File Exists!");
     file = SD.open("test.txt",FILE_WRITE);
     if(file){
       file.println("The Data Logger:" + datalog);
       file.close();
       }else{Serial.println("File Cannot Be Written To!");}
      }else{Serial.println("File Dosen't Exist");
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
}
void motermove(){
   buttonState = digitalRead(bt);
if (bt == HIGH){
  buttonState = digitalRead(bt);
if (bt == HIGH){
   digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(enA, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(enB, 25);

delay(5000);
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(enA, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(enB, 25);

delay(1000);
 }
}
```

โค้ดคำสั่งเดินเป็นเลขแปด

```
#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
Adafruit_MPU6050 mpu;
#include <SD.h>
#include <SPI.h>
int chipSelect = 53;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
File file;
#define ENC_COUNT_REV 374
#define PWM1 2
#define PWM2 19
#define ENC_IN1 3
#define ENC_IN2 18
int speedcontrol1 = 0;
int speedcontrol2 = 0;
volatile long encoderValue1 = 0;
volatile long encoderValue2 = 0;
int rpm1 = 0;
int rpm2 = 0;
int motorPwm1 = 0;
int motorPwm2 = 0;
  sensors_event_t a, g, temp;
//SD Card
#define enA 9
#define in1 4
#define in2 5
#define enB 10
#define in3 6
#define in4 7
String datalog ="";
unsigned long period1 = 1000; //ระยะเวลาที่ต้องการรอ
unsigned long last_time1 = 0;
 int onesecond = 0;
void setup(){
  Serial.begin(9600);
 pinMode(enA, OUTPUT);
  pinMode(enB, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
pinMode(in4, OUTPUT);
pinMode(bt, INPUT);
Wire.begin();

while (!Serial)
  delay(10); // will pause Zero, Leonardo, etc until serial console opens
  Serial.println("Adafruit MPU6050 test!");
if (!mpu.begin()) {
  Serial.println("Failed to find MPU6050 chip");
  while (1) {
    delay(10);
  }
}
Serial.println("MPU6050 Found!");
mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
Serial.print("Accelerometer range set to: ");
switch (mpu.getAccelerometerRange()) {
case MPU6050_RANGE_2_G:
  Serial.println("+-2G");
  break;
case MPU6050_RANGE_4_G:
  Serial.println("+-4G");
  break;
case MPU6050_RANGE_8_G:
  Serial.println("+-8G");
  break;
case MPU6050_RANGE_16_G:
  Serial.println("+-16G");
  break;
}
mpu.setGyroRange(MPU6050_RANGE_250_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case MPU6050_RANGE_250_DEG:
  Serial.println("+- 250 deg/s");
  break;
case MPU6050_RANGE_500_DEG:
  Serial.println("+- 500 deg/s");
  break;
case MPU6050_RANGE_1000_DEG:
  Serial.println("+- 1000 deg/s");
  break;
case MPU6050_RANGE_2000_DEG:
  Serial.println("+- 2000 deg/s");
  break;
}

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
  Serial.println("260 Hz");
  break;
case MPU6050_BAND_184_HZ:
  Serial.println("184 Hz");
  break;
case MPU6050_BAND_94_HZ:
  Serial.println("94 Hz");
  break;
case MPU6050_BAND_44_HZ:
  Serial.println("44 Hz");
  break;
case MPU6050_BAND_21_HZ:
  Serial.println("21 Hz");
  break;
case MPU6050_BAND_10_HZ:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Serial.println("10 Hz");
    break;
  case MPU6050_BAND_5_HZ:
    Serial.println("5 Hz");
    break;
  }
  //Serial.println("");
  //delay(100);


  pinMode(chipSelect,OUTPUT);
  pinMode(ENC_IN1, INPUT_PULLUP);
  pinMode(ENC_IN2, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(ENC_IN1), updateEncoder1, RISING);
  attachInterrupt(digitalPinToInterrupt(ENC_IN2), updateEncoder2, 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("test.txt",FILE_WRITE);
  file.close();
  Serial.println("File Created!");
}
int k=0;
void loop(){

  datalog = " Counter Num " + String(k) + " Speed 1 : " + String(rpm1)+ " RPM "
+ " Speed 2 : " + String(rpm2)+ " RPM " + " Accerlometer : " + " X : "
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  + String(a.acceleration.x) + " Y : " + String(a.acceleration.y) + " Z : " +
String(a.acceleration.z)+ " Gyroscope : " + " X : " + String(g.gyro.x) + " Y : " +
String(g.gyro.y)
  + " Z : " + String(g.gyro.z);
    if( millis() - last_time1 > period1) {
     last_time1 = millis();
     onesecond++;

    }
if(onesecond>5){
  onesecond=0;
  k=0;
}
    if(onesecond==3){

     if(SD.exists("test.txt")){
     Serial.println("File Exists!");
     file = SD.open("test.txt",FILE_WRITE);
     if(file){
       Serial.print(" i ");
       file.println("The Data Logger:" + datalog);
       k++;
       file.close();
       }else{Serial.println("File Cannot Be Written To!");}
      }else{Serial.println("File Dosen't Exist");
     }
    }

     motermove();
     Encode_1();
     Move();
     //Sdata();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void Encode_1(){
    motorPwm1 = map(analogRead(speedcontrol1), 0, 1023, 0, 255);
    motorPwm2 = map(analogRead(speedcontrol2), 0, 1023, 0, 255);
    analogWrite(PWM1, motorPwm1);
    analogWrite(PWM2, motorPwm2);
    static unsigned long timer = millis();
    const unsigned long per = 100;
      if(millis()- timer >= per){
      timer += per;
        rpm1 = (float)(encoderValue1 * 60 / ENC_COUNT_REV);
    rpm2 = (float)(encoderValue2 * 60 / ENC_COUNT_REV);
    // Only update display when there is a reading
    if (motorPwm1 > 0 || rpm1 > 0,motorPwm2 > 0 || rpm2 > 0){
      Serial.print(" SPEED1: ");
      Serial.print(rpm1);
      Serial.print(" RPM");
      Serial.print('\t');
      Serial.print(" SPEED2: ");
      Serial.print(rpm2);
      Serial.println(" RPM");
    }
    encoderValue1 = 0;
    encoderValue2 = 0;
    }
 }
void updateEncoder1()
{
    encoderValue1++;
}
void updateEncoder2(){
    encoderValue2++;
  }
void Move(){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  /* Print out the values */
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");

  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
  Serial.println(" rad/s");

 Serial.println("");
 delay(100);
  }

void Sdata(){
    datalog = " Counter Num " + String(onesecond) + " Speed 1 : " +
String(rpm1)+ " RPM " + " Speed 2 : " + String(rpm2)+ " RPM " + " Accerlometer
: " + " X : "
  + String(a.acceleration.x) + " Y : " + String(a.acceleration.y) + " Z : " +
String(a.acceleration.z)+ " Gyroscope : " + " X : " + String(g.gyro.x) + " Y : " +
String(g.gyro.y)
  + " Z : " + String(g.gyro.z);
    if( millis() - last_time1 > period1) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      last_time1 = millis();
      onesecond++;
       if(SD.exists("test.txt")){
       Serial.println("File Exists!");
       file = SD.open("test.txt",FILE_WRITE);
       if(file){
         file.println("The Data Logger:" + datalog);
         file.close();
         }else{Serial.println("File Cannot Be Written To!");}
        }else{Serial.println("File Dosen't Exist");
      }
 }
}
void motermove(){
   buttonState = digitalRead(bt);
if (bt == HIGH){
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(enA, 255);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(enB, 20);
    delay(10000);
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(enA, 20);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    digitalWrite(enB, 255);

delay(12000);
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ด Encoder

```
// Motor encoder output pulse per rotation (change as required)
#define ENC_COUNT_REV 374
// Encoder output to Arduino Interrupt pin
#define PWM1 2
#define PWM2 19
#define ENC_IN1 3
#define ENC_IN2 18
// MD10C PWM connected to pin 10
// MD10C DIR connected to pin 12
// Analog pin for potentiometer
int speedcontrol1 = 0;
int speedcontrol2 = 0;
// Pulse count from encoder
volatile long encoderValue1 = 0;
volatile long encoderValue2 = 0;
// One-second interval for measurements
int interval = 500;
// Counters for milliseconds during interval
long previousMillis = 0;
long currentMillis = 0;
int rpm1 = 0;
int rpm2 = 0;
int motorPwm1 = 0;
int motorPwm2 = 0;

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

  // Set encoder as input with internal pullup
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  pinMode(ENC_IN1, INPUT_PULLUP);
  pinMode(ENC_IN2, INPUT_PULLUP);
  // Attach interrupt
  attachInterrupt(digitalPinToInterrupt(ENC_IN1), updateEncoder1, RISING);
  attachInterrupt(digitalPinToInterrupt(ENC_IN2), updateEncoder2, RISING);
  // Setup initial values for timer
  previousMillis = millis();
}

void loop()
{
    // Control motor with potentiometer
    motorPwm1 = map(analogRead(speedcontrol1), 0, 1023, 0, 255);
    motorPwm2 = map(analogRead(speedcontrol2), 0, 1023, 0, 255);
    // Write PWM to controller
    analogWrite(PWM1, motorPwm1);
    analogWrite(PWM2, motorPwm2);
  // Update RPM value every second
  currentMillis = millis();
  if (currentMillis - previousMillis > interval) {
    previousMillis = currentMillis;
    // Calculate RPM
    rpm1 = (float)(encoderValue1 * 60 / ENC_COUNT_REV);
    rpm2 = (float)(encoderValue2 * 60 / ENC_COUNT_REV);
    // Only update display when there is a reading
    if (motorPwm1 > 0 || rpm1 > 0,motorPwm2 > 0 || rpm2 > 0) {
      Serial.print("PWM1 VALUE: ");
      Serial.print(motorPwm1);
      Serial.print('\t');
      Serial.print("PWM2 VALUE: ");
      Serial.print(motorPwm2);
      Serial.print('\t');
      Serial.print(" PULSES1: ");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      Serial.print(encoderValue1);
      Serial.print('\t');
      Serial.print(" PULSES2: ");
      Serial.print(encoderValue2);
      Serial.print('\t');
      Serial.print(" SPEED1: ");
      Serial.print(rpm1);
      Serial.print(" RPM");
      Serial.print('\t');
      Serial.print(" SPEED2: ");
      Serial.print(rpm2);
      Serial.println(" RPM");
    }
    encoderValue1 = 0;
    encoderValue2 = 0;
  }
}

void updateEncoder1()
{
  // Increment value for each pulse from encoder
  encoderValue1++;
}
void updateEncoder2(){
    encoderValue2++;
  }
```

โค้ด GY-521

```
// Basic demo for accelerometer readings from Adafruit MPU6050

#include <Adafruit_MPU6050.h>
#include <Adafruit_Sensor.h>
#include <Wire.h>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Adafruit_MPU6050 mpu;

void setup(void) {
  Serial.begin(115200);
  while (!Serial)
    delay(10); // will pause Zero, Leonardo, etc until serial console opens

  Serial.println("Adafruit MPU6050 test!");

  // Try to initialize!
  if (!mpu.begin()) {
    Serial.println("Failed to find MPU6050 chip");
    while (1) {
      delay(10);
    }
  }
  Serial.println("MPU6050 Found!");

  mpu.setAccelerometerRange(MPU6050_RANGE_2_G);
  Serial.print("Accelerometer range set to: ");
  switch (mpu.getAccelerometerRange()) {
  case MPU6050_RANGE_2_G:
    Serial.println("+-2G");
    break;
  case MPU6050_RANGE_4_G:
    Serial.println("+-4G");
    break;
  case MPU6050_RANGE_8_G:
    Serial.println("+-8G");
    break;
  case MPU6050_RANGE_16_G:
    Serial.println("+-16G");
    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
mpu.setGyroRange(MPU6050_RANGE_250_DEG);
Serial.print("Gyro range set to: ");
switch (mpu.getGyroRange()) {
case MPU6050_RANGE_250_DEG:
  Serial.println("+- 250 deg/s");
  break;
case MPU6050_RANGE_500_DEG:
  Serial.println("+- 500 deg/s");
  break;
case MPU6050_RANGE_1000_DEG:
  Serial.println("+- 1000 deg/s");
  break;
case MPU6050_RANGE_2000_DEG:
  Serial.println("+- 2000 deg/s");
  break;
}

mpu.setFilterBandwidth(MPU6050_BAND_21_HZ);
Serial.print("Filter bandwidth set to: ");
switch (mpu.getFilterBandwidth()) {
case MPU6050_BAND_260_HZ:
  Serial.println("260 Hz");
  break;
case MPU6050_BAND_184_HZ:
  Serial.println("184 Hz");
  break;
case MPU6050_BAND_94_HZ:
  Serial.println("94 Hz");
  break;
case MPU6050_BAND_44_HZ:
  Serial.println("44 Hz");
  break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  case MPU6050_BAND_21_HZ:
    Serial.println("21 Hz");
    break;
  case MPU6050_BAND_10_HZ:
    Serial.println("10 Hz");
    break;
  case MPU6050_BAND_5_HZ:
    Serial.println("5 Hz");
    break;
  }

  Serial.println("");
  delay(100);
}
void loop() {

  /* Get new sensor events with the readings */
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);
  /* Print out the values */
  Serial.print("Acceleration X: ");
  Serial.print(a.acceleration.x);
  Serial.print(", Y: ");
  Serial.print(a.acceleration.y);
  Serial.print(", Z: ");
  Serial.print(a.acceleration.z);
  Serial.println(" m/s^2");
  Serial.print("Rotation X: ");
  Serial.print(g.gyro.x);
  Serial.print(", Y: ");
  Serial.print(g.gyro.y);
  Serial.print(", Z: ");
  Serial.print(g.gyro.z);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Serial.println(" rad/s");

 // Serial.print("Temperature: ");
  //Serial.print(temp.temperature);
  //Serial.println(" degC");

  Serial.println("");
  delay(100);
}
```

โค้ด SD card

```
#define trigPin1 3
#define echoPin1 2
#define trigPin2 4
#define echoPin2 5
#define trigPin3 7
#define echoPin3 8
#define buzzer 11
#define LED 13
long duration, distance, RightSensor,BackSensor,FrontSensor,LeftSensor;
void setup()
{
Serial.begin (9600);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);
}
void loop() {
SonarSensor(trigPin1, echoPin1);
RightSensor = distance;
SonarSensor(trigPin2, echoPin2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LeftSensor = distance;
SonarSensor(trigPin3, echoPin3);
FrontSensor = distance;
Serial.print(LeftSensor);
Serial.print("ซม.");
Serial.print(FrontSensor);
Serial.print("ซม.");
Serial.println(RightSensor);
Serial.print("ซม.");
 distance = (duration*.0343)/2;
 Serial.print("ระยะวัตถุ: ");
 Serial.print(distance);
 Serial.println ("ซม.");
 Serial.print("ระยะวัตถุ: ");
 Serial.print(distance* 0.39370);
 Serial.println ("นิ้ว.");
 delay(20);
}
void SonarSensor(int trigPin,int echoPin)
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
 if(distance <= 30 && distance >= 21) // ระยะวัตถุตั้งแต่ 21-30 ซม.
 {
 digitalWrite (buzzer,HIGH);
 digitalWrite (LED,HIGH);
 delay (1000);
 digitalWrite (buzzer,LOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
digitalWrite (LED,LOW);
delay (1000);
}
else if (distance <= 21 && distance >= 11) // ระยะวัตถุตั้งแต่ 11-20 ซม.
{
digitalWrite (buzzer,HIGH);
digitalWrite (LED,HIGH);
delay (500);
digitalWrite (buzzer,LOW);
digitalWrite (LED,LOW);
delay (500);
}
else if (distance <= 11 && distance >= 6) // ระยะวัตถุตั้งแต่ 6-10 ซม.
{
digitalWrite (buzzer,HIGH);
digitalWrite (LED,HIGH);
delay (200);
digitalWrite (buzzer,LOW);
digitalWrite (LED,LOW);
delay (200);
}
else if (distance < 6 ) // ระยะวัตถุน้อยกว่า 5 ซม.
{
digitalWrite (buzzer,HIGH);
digitalWrite (LED,HIGH);
}
else if (distance > 31) // ระยะวัตถุตั้งแต่ 31 ซม. ขึ้นไป
{
digitalWrite (buzzer,LOW);
digitalWrite (LED,LOW);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้