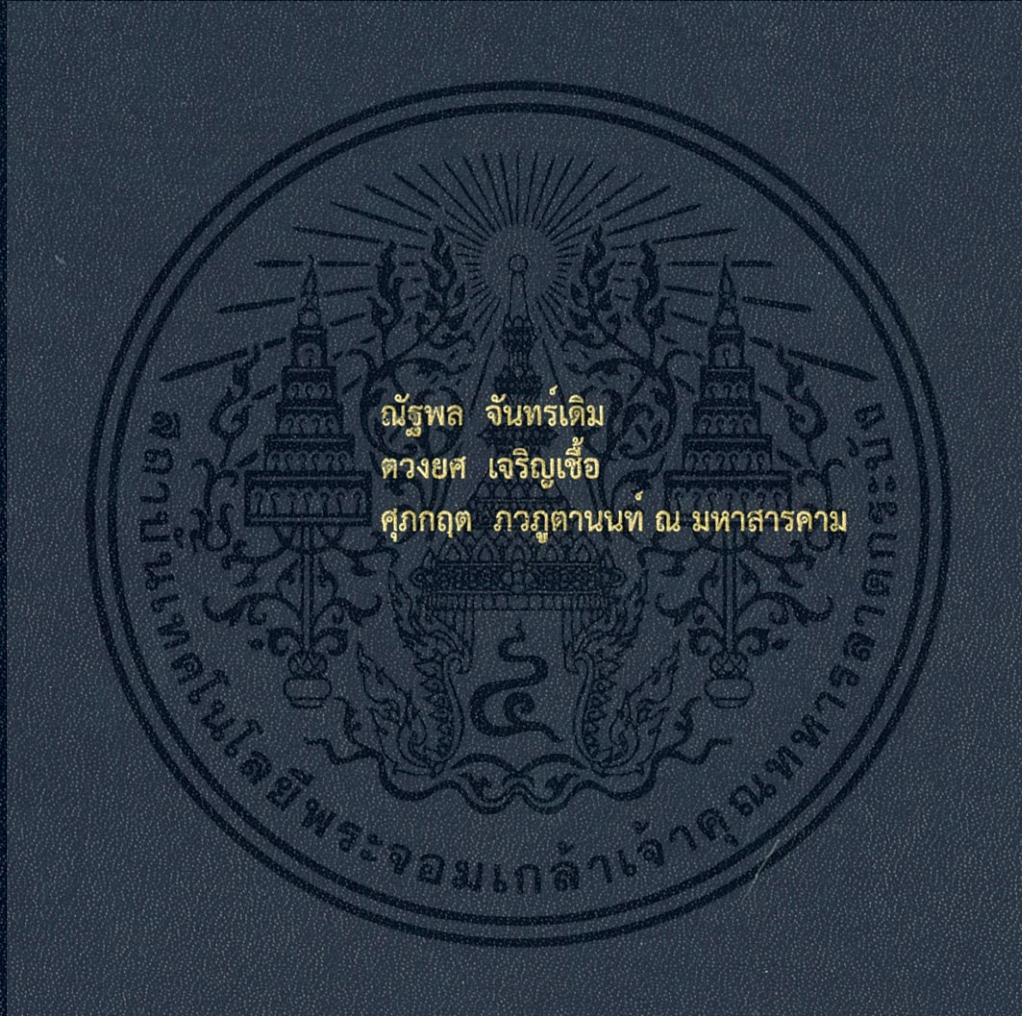


การควบคุมมอเตอร์ไฟฟ้ากระแสตรงสี่ควอดแรนต์
4-QUADRANT DC MOTOR DRIVE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

การควบคุมมอเตอร์ไฟฟ้ากระแสตรงสี่ควอดแรนต์
4-QUADRANT DC MOTOR DRIVE



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4-QUADRANT DC MOTOR DRIVE



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2559

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมมอเตอร์ไฟฟ้ากระแสตรงสี่ควอดแรนต์
4-QUADRANT DC MOTOR DRIVE

ผู้จัดทำ	นายณัฐพล	จันทร์เต็ม	56010410
	นายตวงยศ	เจริญเชื้อ	56010462
	นายศุภกฤต	ภาภูตานนท์ ณ มหาสารคาม	56011231



.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตหิรัญรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมมอเตอร์กระแสตรงสี่ควอดแรนต์

โดย

นายณัฐพล จันทร์เดิม 56010410

นายตวงยศ เจริญเชื้อ 56010462

นายศุภกฤต ภาภูตานนท์ ณ มหาสารคาม 56011231

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตศิริภรณ์

ปีการศึกษา 2559

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงด้วยวงจรขับเคลื่อนแบบ H-Bridge ประกอบด้วยมอสเฟต 4 ตัว ออกแบบและประกอบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบสี่จตุภาค (4-Quadrant) โดยใช้มอสเฟสแบบ N-channel 4 ตัวต่อกันแบบ H-Bridge จตุภาคที่หนึ่งคือ การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงให้หมุนไปด้านหนึ่งๆ จตุภาคที่สองคือ การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงให้หมุนไปด้านตรงกันข้าม จตุภาคที่สามและจตุภาคที่สี่คือ การหยุดมอเตอร์ไฟฟ้ากระแสตรงในแต่ละทิศทางการหมุนโดยใช้ Microcontroller Arduino UNO เป็นตัวควบคุมความเร็วรอบมอเตอร์ไฟฟ้ากระแสตรงให้คงที่โดยจะรับสัญญาณความเร็วรอบจากเอ็นโคดเดอร์ (Encoder) กลับเข้ามาเพื่อทำการปรับความเร็วรอบให้คงที่ เพื่อนำมอเตอร์ไฟฟ้ากระแสตรงไปใช้เป็นต้นกำลังให้กับสายพานลำเลียง เมื่อสายพานมีภาระน้ำหนักเพิ่มขึ้น ระบบจะทำการชดเชยแรงดันไฟฟ้าส่งให้กับมอเตอร์ไฟฟ้ากระแสตรง เพื่อเพิ่มความเร็วรอบขึ้นเข้าสู่จุดที่ได้ทำการตั้งไว้ โดยรวมสายพานสามารถเคลื่อนที่ได้สองทิศทาง และสามารถรักษาระดับความเร็วการวิ่งไว้ได้ ซึ่งหากนำระบบควบคุมนี้ไปประยุกต์ใช้กับระบบส่งของด้วยสายพาน จะทำให้ระบบมีความเร็วคงที่และแม่นยำ ทำให้สามารถสร้างสายพานลำเลียงที่มีความเที่ยงตรงสูง ทำให้ระบบการผลิตได้ผลผลิตทันตามเวลาที่กำหนด และมีความน่าเชื่อถือซึ่งเพิ่มข้อได้เปรียบเชิงธุรกิจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4-QUADRANT DC MOTOR DRIVE

By

Mr. Natthapon Chanderm 56010410

Mr. Tuangyos Char oenshue 56010462

Mr. Supakit Pavaputanon 56011231

Advisor

Asst.Prof.Dr.Kongsak Anuntahirunrat

Academic Year 2016

ABSTRACT

This thesis presents a DC motor driven by a motor-driven cycle in H-bridge and used as a switch to control the MOSFET circuit. H-Bridge circuits H-bridge consists of 4 switches control power flow direction into direct current motor in this thesis option using a MOSFET switch by will drive the MOSFET must be composed with the supply signal using microcontroller Microcontroller ArduinoUno pulse and pulse signal in order to switch on the control MOSFET is divided into 4 control by the two upper sides of the circuit must make a separate Ground point from a supply of the motor using the integrated circuit to separate points of two switches on by 4-Quadrant dc motor drive is consists of clockwise rotation, counterclockwise rotation, and stop clockwise, stop counterclockwise rotation, stop by put the position of the rotary encoder position and position value reading, is in the form of a direct current motor speed shown by the land owner. Selective emulation of the carriage belt when the belt with a burden it will cause the speed of the motor pulley, then OK, thus bringing the rotary encoder to find the speed of the conveyor belt steps have obligations and returns back to the microcontroller the microcontroller then processed out of order, a pulse, close-open with much more frequency value is DC motor makes it rotate around the same speed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริมาณนิพนธ์ฉบับนี้ดำเนินการสำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.คงศักดิ์ อนันตศิริรัตน ซึ่งท่านได้ให้ข้อคิดเห็นและคำแนะนำต่างๆ อันเป็นประโยชน์อย่างยิ่งในการดำเนินงาน อีกทั้งยังช่วยแก้ปัญหาต่างๆ ที่เกิดขึ้นด้วย

ขอขอบพระคุณคณะอาจารย์และเจ้าหน้าที่ภาควิชาวิศวกรรมการวัดและควบคุม รวมถึงเพื่อนพี่น้องสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความช่วยเหลือและให้คำแนะนำที่ดีมาโดยตลอด

สุดท้ายนี้ คณะผู้จัดทำขอกราบขอบพระคุณผู้ปกครองและครอบครัว ซึ่งให้การสนับสนุนในด้านต่างๆ และคอยเป็นกำลังใจมาโดยตลอดจนสำเร็จการศึกษา คณะผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างยิ่งสำหรับทุกความช่วยเหลือ และขอขอบคุณทุกท่านที่มีส่วนเกี่ยวข้องมา ณ ที่นี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญ(ต่อ)	V
สารบัญรูป	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1 ความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขอบเขตการศึกษา	1
1.4 รายละเอียดของปริญญานิพนธ์	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 แผนการดำเนินโครงการ	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	3
2.1 มอเตอร์ไฟฟ้ากระแสตรง	3
2.1.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง	3
2.1.2 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง	4
2.2 วงจรที่ใช้ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง	6
2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)	9
2.3.1 คุณสมบัติและการต่อใช้งาน Microcontroller Arduino Uno R3	9
บทที่ 3 การออกแบบวงจร การสร้างแบบจำลอง และการเขียนโปรแกรม	11
3.1 วงจรควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงพื้นฐาน	11
3.2 วงจรขับมอเตอร์ไฟฟ้ากระแสตรง (Drive Motor Circuit)	12
3.2.1 วงจรขับมอเตอร์ไฟฟ้ากระแสตรงโดยใช้ Microcontroller Arduino	12
3.2.2 วงจรขับมอเตอร์ไฟฟ้ากระแสตรงที่มี Encoder ควบคุมค่าความเร็ว	13
3.2.3 วงจรเอนโค้ดเดอร์ (Encoder)	14
3.3 หลักการสร้างแบบจำลองของคอนเวเยอร์	15

สารบัญ (ต่อ)

	หน้า
3.3.1 วัสดุหลักที่ใช้ในการสร้างแบบจำลองคอนเวเยอร์	16
3.3.2 ขั้นตอนการประกอบและติดตั้ง	17
3.4 การเขียนโปรแกรมควบคุม	18
3.4.1 โปรแกรมอ่านค่าจาก Encoder	18
3.4.2 โปรแกรมควบคุมสัญญาณ RPM	20
บทที่ 4 การทดลองและผลการทดลอง	23
4.1 การทดลองการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง Mode Forward	23
4.2 การทดลองการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง Mode Reverse	25
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	26
5.1 สรุปผลการทดลอง	26
5.2 ปัญหาที่พบและแนวทางแก้ไขปัญหา	26
5.2.1 ปัญหาที่พบ	26
5.2.2 แนวทางแก้ไข	26
5.3 ข้อเสนอแนะและแนวทางการพัฒนา	27
เอกสารอ้างอิง	28

สารบัญรูป

รูปที่	หน้า
2.1 วงจรภายในของมอเตอร์ไฟฟ้ากระแสตรง	3
2.2 ลักษณะของสัญญาณ PWM ที่ %Duty Cycle ที่แตกต่างกัน	5
2.3 ระบบควบคุมตำแหน่งที่ประกอบด้วยการควบคุมแบบป้อนกลับ	5
2.4 การสร้าง Pulse ของ Rotary Encoder	6
2.5 วงจร H-Bridge ที่ควบคุมโดยหน้าสัมผัสสี่เหลี่ยม	6
2.6 วงจรการควบคุมแบบลิเนียร์	7
2.7 วงจรการควบคุมความเร็ว DC Motor แบบสวิตชิง หรือ PWM	7
2.8 การเคลื่อนที่ของมอเตอร์ไฟฟ้ากระแสตรงในวงจร H-Bridge ที่ควบคุมด้วยมอสเฟต	8
2.9 โครงสร้างภายนอกของ Microcontroller Arduino Uno R3	9
2.10 ตำแหน่งของขาบน ATMEGA328	10
3.1 การต่อวงจรควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงโดยใช้ตัวต้านทานปรับค่าได้อย่างง่าย	11
3.2 การต่อวงจร H-Bridge	12
3.3 การต่อวงจร H-Bridge กับ Encoder	13
3.4 การต่อวงจรเอนโค้ดเดอร์ (Encoder)	14
3.5 แบบจำลองคอนเวเยอร์	15
3.6 การ Coupling ระหว่างมอเตอร์ไฟฟ้ากระแสตรงและเฟลา	15
3.7 การ Coupling ระหว่างมอเตอร์ไฟฟ้ากระแสตรงและ Encoder	16
3.8 แบบจำลองคอนเวเยอร์	17
3.9 Flow Chart แสดงโปรแกรมอ่านค่าจาก Encoder	18
3.10 ฟังก์ชัน Void Setup ส่วน Encoder	19
3.11 ฟังก์ชัน Void Loop และ ฟังก์ชันอินเตอร์รัพท์ (Interrupt) ส่วน Encoder	19
3.12 Flow Chart โปรแกรมควบคุมสัญญาณ RPM	20
3.13 ฟังก์ชัน Void Setup ส่วนควบคุมสัญญาณ RPM	21
3.14 ฟังก์ชัน Void Loop และ ฟังก์ชันประกอบควบคุมสัญญาณ RPM	21
3.15 ฟังก์ชัน Void Loop และ ฟังก์ชันประกอบควบคุมสัญญาณ RPM(ต่อ)	22
4.1 รูปแสดงการทำงานแบบ Forward Motoring	23
4.2 รูปแสดงความสัมพันธ์ระหว่าง %Duty Cycle กับความเร็วรอบมอเตอร์	24
4.3 รูปแสดงการทำงานแบบ Reverse Motoring	25

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินโครงการระหว่างเดือนสิงหาคม 2559 ถึงเดือนเมษายน 2560	2
4.1 แสดงความสัมพันธ์ระหว่าง %Duty Cycle กับความเร็วรอบมอเตอร์	23



บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ระบบอุตสาหกรรมในปัจจุบันนั้นการใช้งานของมอเตอร์ไฟฟ้ากระแสตรงนั้น มีการใช้งานกันอย่างแพร่หลายในอุตสาหกรรม โดยจะนำไปใช้ขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงที่ทำหน้าที่เป็นที่ยึดผ้าหรือม้วนกระดาษ เป็นต้น จึงเห็นถึงความสำคัญของการควบคุม อัตราเร็วและการออกแบบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงได้ด้วยตนเองในแบบที่สามารถเข้าใจได้โดยง่าย และสามารถนำไปทดลองปรับใช้ได้กับงานที่ผู้ศึกษาด้วยตนเองต้องการ

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. เพื่อศึกษาการออกแบบโปรแกรมที่ใช้ในการควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง ซึ่งเขียนโดย Microcontroller Arduino
2. เพื่อศึกษาวงจรการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง และการออกแบบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง

1.3 ขอบเขตการศึกษา

1. สามารถใช้โปรแกรม Microcontroller Arduino ในการควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรงให้เคลื่อนที่ครบ 4 แบบได้ซึ่งก็คือ เคลื่อนที่หมุนไปด้านหน้า (Forward Motoring), เคลื่อนที่หมุนกลับด้าน (Reverse Motoring), เคลื่อนที่หยุดขณะทิศทางไปด้านหน้า (Forward Break) และเคลื่อนที่หยุดขณะทิศทางกลับด้าน (Reverse Break)
2. สามารถออกแบบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรง โดยใช้โปรแกรม Altium Designer

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึง วัตถุประสงค์ ขั้นตอนการศึกษา ขอบเขตของการศึกษา และการจัดทำโครงการ พร้อมทั้งรายละเอียดของปริญญานิพนธ์ในแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้องของหลักการ ทฤษฎีระบบควบคุม หลักการทำงานของดีซีมอเตอร์ และวงจรอิเล็กทรอนิกส์ที่ใช้ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง

บทที่ 3 หลักการออกแบบวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรง และหลักการออกแบบโปรแกรมที่ใช้ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การทดลอง แสดงผลการทดสอบมอเตอร์ไฟฟ้ากระแสตรง

บทที่ 5 บทวิจารณ์และสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจการออกแบบวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรง และการออกแบบลายวงจร โดยใช้โปรแกรม Altium Designer
2. เข้าใจการจำลองการทำงานของวงจรโดยใช้โปรแกรม Proteus
3. ได้ใช้ความคิดในการทำโครงงาน เพื่อเป็นการฝึกหัดมีการทำงานที่เป็นระบบ
4. เข้าใจการเขียนโปรแกรม แก้อุปกรณ์ การควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรงมากยิ่งขึ้น

1.6 แผนการดำเนินโครงการ

ตารางที่ 1.1 แผนการดำเนินโครงการระหว่างเดือนสิงหาคม 2559 ถึงเดือนเมษายน 2560

ขั้นตอนการดำเนินโครงการ	2559					2560				
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	
1.ศึกษาข้อมูลและทฤษฎีที่เกี่ยวข้อง										
2.ศึกษาวงจรขับมอเตอร์ไฟฟ้ากระแสตรง										
3.ออกแบบวงจรและทดสอบการทำงาน										
4.ซื้ออุปกรณ์สำหรับทำวงจรขับมอเตอร์ไฟฟ้ากระแสตรง										
5.ศึกษาการเขียนโปรแกรม										
6.ออกแบบสายพาน										
7.เตรียมซื้ออุปกรณ์สำหรับทำโครงสร้างรากฐานของสายพาน										
8.ทำวงจรขับมอเตอร์ไฟฟ้ากระแสตรง										
9.เริ่มสร้างและพัฒนาสายพานต้นแบบ										
10.เขียนโปรแกรมและทดสอบการทำงานของสายพาน										
11.ตรวจสอบและแก้ไขปัญหา										
12.สรุปผลการดำเนินงาน										

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 มอเตอร์ไฟฟ้ากระแสตรง

2.1.1 หลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง

หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อป้อนแรงดันและกระแสไฟฟ้าเข้าไปในมอเตอร์ กระแสไฟฟ้าส่วนหนึ่งจะผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์เมเจอร์เพื่อสร้างแรงบิดจากการที่กระแสไฟฟ้าเคลื่อนที่ตัดกับสนามแม่เหล็ก และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field Coil) เพื่อสร้างสนามแม่เหล็ก 2 สนามในขณะเดียวกัน คุณสมบัติของเส้นแรงแม่เหล็กจะหักล้างกันและจะเสริมแรงกันในขณะเดียวกัน ทำให้เกิดแรงบิดในขดลวดอาร์เมเจอร์ทำให้ขดลวดอาร์เมเจอร์นี้หมุนได้ ขณะที่ตัวขดลวดอาร์เมเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่า การที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิริยาต่อกัน ทำให้ขดลวดอาร์เมเจอร์หรือโรเตอร์ หมุนไปนั้น เป็นไปตามกฎมือซ้ายของเฟลมมิง (Fleming left hand rule) เขียนได้ดังรูปที่ 2.1



รูปที่ 2.1 วงจรภายในของมอเตอร์ไฟฟ้ากระแสตรง

พิจารณาให้มีค่าความต่างศักย์ตกคร่อมขดลวดอาร์เมเจอร์เป็น V_A และค่าแรงดันตกคร่อมความต้านทานภายในที่ค่าความต้านทานมีค่าเป็น V_R ดังนั้นจะสามารถกล่าวได้ว่า

แรงดัน V_A ถูกเรียกว่า แรงเคลื่อนเหนี่ยวนำป้อนกลับ (Back EMF) ซึ่งเกิดขึ้นในขดลวดเคลื่อนที่ขณะที่หมุนแรงดันที่เกิดขึ้นเป็นไปตามกฎของแรงจากร การเคลื่อนที่ของตัวนำในสนามแม่เหล็ก สัมพันธ์กับแรงเคลื่อนเหนี่ยวนำแม่เหล็ก และความเร็วในการเคลื่อนที่ของตัวนำแรงดันที่เกิดขึ้นจะมีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้กับมอเตอร์ไฟฟ้ากระแสตรง และแปรผันตรงกับความเร็วในการหมุน ผลบวกของแรงดันที่โรเตอร์ (V_A) และแรงดันตกคร่อมขดลวด (V_R) ต้องเท่ากับแรงดันที่ป้อนให้กับมอเตอร์ไฟฟ้ากระแสตรงจากสมการที่ (2.1)

$$V = V_A + V_R \quad (V) \quad (2.1)$$

เมื่อพิจารณาตั้งแต่มอเตอร์ไฟฟ้ากระแสตรงหยุดนิ่ง ความเร็วมีค่าเป็นศูนย์ ดังนั้น $V_A = 0$, $V_R = V$ กระแสที่ไหลในมอเตอร์ไฟฟ้ากระแสตรงหาได้จากสมการที่ (2.2)

$$I = V_R / R \quad (A) \quad (2.2)$$

เมื่อมอเตอร์ไฟฟ้ากระแสตรงเริ่มหมุนจะมีความเร็ว และ V_A เพิ่มขึ้นเป็นเส้นตรงตามความเร็ว ดังนั้น V_R ซึ่งมีค่าเท่ากับความแตกต่างระหว่าง V_A และ V จะเริ่มลดลงและกระแส I ก็เริ่มลดลงเช่นกัน ขณะที่มอเตอร์ไฟฟ้ากระแสตรงยังมีความเร็วอยู่ ความเร็วจะเพิ่มขึ้น แรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดของมอเตอร์ไฟฟ้ากระแสตรงรับภาระโหลดได้สมดุลพอดี ขณะที่มอเตอร์ไฟฟ้ากระแสตรงไม่มีโหลด และหมุนอย่างอิสระจะมีเพียงค่าความฝืดของแบร์ริง และแรงต้านอากาศทำให้ค่า V_A เกือบเท่ากับค่า V

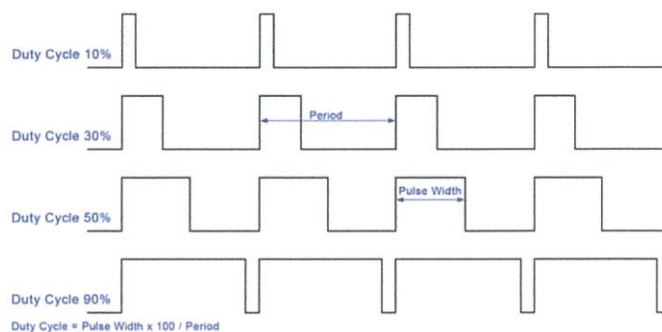
2.1.2 การควบคุมมอเตอร์ไฟฟ้ากระแสตรง

การควบคุมมอเตอร์ไฟฟ้ากระแสตรงประกอบไปด้วยการควบคุมความเร็ว และการควบคุมตำแหน่งหรือมุมของมอเตอร์ไฟฟ้ากระแสตรงซึ่งมีรายละเอียดดังนี้

2.1.2.1 การควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง

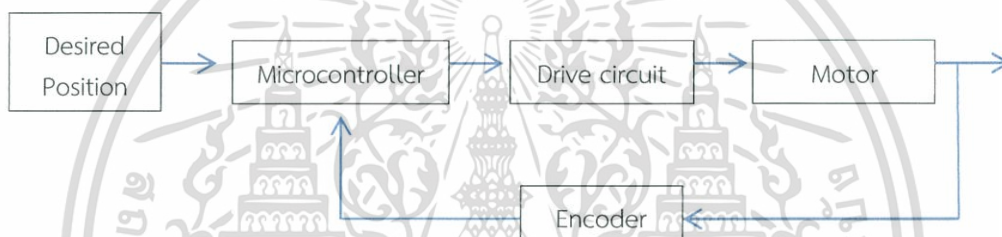
PWM (Pulse Width Modulation) เป็นการควบคุมความเร็วโดยการปรับความกว้างของพัลส์ โดยการนำเอาสัญญาณสองสัญญาณมาเปรียบเทียบกัน ประกอบด้วย สัญญาณสามเหลี่ยม และสัญญาณที่ต้องการปรับความกว้างของพัลส์ ซึ่งสร้างได้ด้วยการใช้วงจร Comparator หรือวงจรเปรียบเทียบ สามารถสร้างได้ด้วยการใช้ Op-Amp ทั้งนี้ไมโครคอนโทรลเลอร์ส่วนมากจะสามารถสร้างสัญญาณ Output PWM ได้เช่นเดียวกัน

ข้อดีของสัญญาณ PWM คือ มีประสิทธิภาพเพราะ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF โดยในโครงงานนี้จะมีการใช้มอสเฟตเป็นตัวตัด-ต่อวงจรของ H-bridge เพื่อสามารถที่จะควบคุมจังหวะในการจ่ายกระแสด้วยการ ON-OFF ในสัดส่วน (Duty Cycle) ที่ต่างกันด้วยความถี่ที่เหมาะสม หากไม่เช่นนั้นที่ความถี่ต่ำมอเตอร์ไฟฟ้ากระแสตรงจะมีการกระตุก แต่ที่ความถี่ที่สูงมาก ก็จะมีการสูญเสียพลังงานในวงจรที่มากขึ้นเช่นกันดังรูปที่ 2.2



รูปที่ 2.2 ลักษณะของสัญญาณ PWM ที่ %Duty Cycle ที่แตกต่างกัน

2. การควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงด้วย Encoder



รูปที่ 2.3 ระบบควบคุมตำแหน่งที่ประกอบด้วย การควบคุมแบบป้อนกลับ

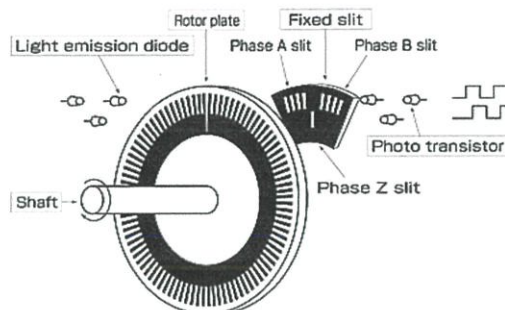
จากรูปที่ 2.3 คือ บล็อกไดอะแกรมของการควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง เมื่อมีการกำหนดค่าของความเร็วให้กับ Microcontroller Arduino

Microcontroller Arduino จะทำการสร้างสัญญาณควบคุมให้กับวงจรขับมอเตอร์ไฟฟ้ากระแสตรง เมื่อมอเตอร์ไฟฟ้ากระแสตรงมีการหมุน Encoder จะทำการตรวจจับแปลงค่าจาก Encoder ที่มอเตอร์ไฟฟ้ากระแสตรงได้หมุน พร้อมทั้งส่งสัญญาณป้อนกลับให้กับไมโครคอนโทรลเลอร์ไปคำนวณค่าความเร็ว

เนื่องจากระบบนั้นมีการควบคุมความเร็ว จึงได้มีการนำ Rotary Encoder เพื่อที่จะนำมาเปรียบเทียบความแตกต่างระหว่าง Input สำหรับการควบคุมต่อไป

Incremental Encoder หรือโดยทั่วไปเรียกว่า Rotary Encoder จะสร้างสัญญาณพัลส์ (Pulse) ที่แปรผันตรงกับการหมุนของเพลามอเตอร์ไฟฟ้ากระแสตรง หรือจะหมุนด้วยความเร็วเท่ากับเพลลาของมอเตอร์ไฟฟ้ากระแสตรงนั่นเอง โดย Rotary Encoder จะประกอบด้วยจานหมุน (Rotary Disk) และอุปกรณ์ตรวจจับ (Sensor) โดยจานหมุนจะมีช่องเล็ก (Slit) เมื่อเพลลาของมอเตอร์ไฟฟ้ากระแสตรงหมุนจะทำให้จานหมุนไปตัดลำแสงของ Sensor ทำให้ชุดรับแสงมีการรับสัญญาณเป็นช่วง ส่งผลให้สัญญาณเอาต์พุต (Output) มีลักษณะ Pulse แสดงดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การสร้าง Pulse ของ Rotary Encoder

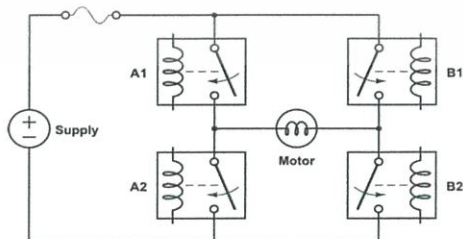
2.2 วงจรที่ใช้ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง

จากหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง เมื่อมีกระแสตรงมาตัดผ่านสนามแม่เหล็ก จะเหนี่ยวนำให้เกิดแรงบิดขึ้นในทิศทางที่เหมาะสม และทำให้เกิดการหมุนของแกนโรเตอร์ ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรงนั้นมีอยู่ 3 ลักษณะคือ

1. หมุน-หยุดหมุน (ON – OFF Control)
2. ทิศทางการหมุน (Direction Control)
3. ความเร็วการหมุน (Speed Control)

โดยทั่วไปวงจรที่ใช้ควบคุมจะเป็นวงจร H-Bridge ที่สามารถควบคุมการหมุน-หยุดหมุน พร้อมทั้งการกลับทิศทางการหมุนได้ จากการทำให้ Pulse Width Modulation (PWM) และควบคุมโดยการสั่งงานจากคอนโทรลเลอร์

อุปกรณ์ที่นิยมใช้ควบคุมส่วนมาก ได้แก่ หน้าสัมผัสสรีเลย์ นำมาต่อเป็นวงจรในลักษณะ H-Bridge ที่สามารถควบคุมการหมุน-หยุดหมุน พร้อมทั้งการกลับทิศทางการหมุนได้ แต่การใช้หน้าสัมผัสสรีเลย์จะไม่สามารถควบคุมความเร็วของการหมุนได้ และต้องระวังในการหมุนกลับทิศทางแบบทันทีทันใด เพราะอาจเกิดความเสียหายกับตัวมอเตอร์ไฟฟ้ากระแสตรงได้ แต่ข้อดีของการใช้หน้าสัมผัสสรีเลย์คือ ทำให้กระแสที่ไหลผ่านมอเตอร์ไฟฟ้ากระแสตรงไม่ตกและสามารถใช้งานได้เต็มกำลัง



รูปที่ 2.5 วงจร H-Bridge ที่ควบคุมโดยหน้าสัมผัสสรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรง(DC Motor) มีการควบคุม 2 แบบ คือ

1. การควบคุมแบบลิเนียร์ (Linear DC Voltage Control)

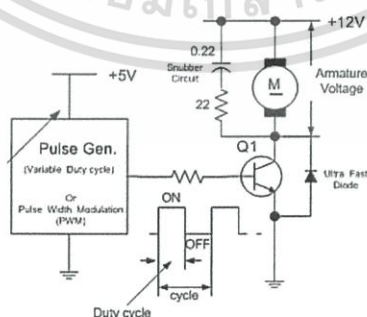
จากรูปที่ 2.6 พบว่าความเร็วของมอเตอร์ไฟฟ้ากระแสตรงจะแปรผันไปตามแรงเคลื่อนไฟฟ้าที่ตกคร่อมขดลวดอาเมเจอร์ (Armature Voltage) ดังนั้นหากสามารถปรับค่าแรงเคลื่อน V_{CE} ได้ ก็ทำให้ Armature Voltage เปลี่ยนแปลงตามไปด้วย แต่เนื่องจาก V_{CE} จะแปรผันตามแรงเคลื่อน V_{BE} โดยการปรับค่าตัวต้านทานที่ปรับค่าได้ ดังนั้นจะสามารถควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงได้ โดยการปรับค่าตัวต้านทาน SPEED ได้ แต่การควบคุมวิธีนี้ไม่นิยมใช้กันโดยเฉพาะมอเตอร์ไฟฟ้ากระแสตรงที่มีกำลังมากๆ เนื่องจากจะเกิดพลังงานสูญเสียภายในตัวทรานซิสเตอร์ในรูปของความร้อนสูงมาก ในขณะที่ทรานซิสเตอร์ทำงานอยู่ในช่วง Active ดังรูปที่ 2.6



รูปที่ 2.6 วงจรการควบคุมแบบลิเนียร์

2. การควบคุมแบบสวิตชิง (Switching Control)

คือ การป้อนไฟ V_{BE} หรือ Control Voltage ให้ทรานซิสเตอร์ทำงานในลักษณะหยุดการทำงาน(Cutoff) และจุดอิ่มตัว(Saturate) โดยจะหลีกเลี่ยงการทำงานในช่วง Active ซึ่งจะทำให้มีการสูญเสียพลังงาน



รูปที่ 2.7 วงจรการควบคุมความเร็ว DC Motor แบบสวิตชิง หรือ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.7 ประกอบไปด้วยตัวกำเนิดสัญญาณพัลส์ ส่งไปควบคุมการทำงานของทรานซิสเตอร์ ในลักษณะ ON และ OFF ทำให้มีกระแสไหลผ่านมอเตอร์ไฟฟ้ากระแสตรง หมุนเป็นช่วงๆที่เป็นพัลส์บวก เรียกว่า “Duty Cycle” โดยที่ตัวผลิตพัลส์ (Pulse Generator) จะต้องรักษาช่วง Cycle หรือความถี่ให้คงที่อยู่ที่ค่าใดค่าหนึ่ง โดยปกติจะเลือกค่าคงที่อยู่ที่ระหว่าง 400Hz – 2KHz สำหรับงานควบคุมมอเตอร์ไฟฟ้ากระแสตรง ดังนั้นในทุกๆ Cycle หากมีการปรับเปลี่ยนค่าพัลส์บวกหรือค่า Duty Cycle ก็จะมีผลทำให้ค่าเฉลี่ยแรงเคลื่อน Output ของทรานซิสเตอร์ที่ไปขับมอเตอร์ไฟฟ้ากระแสตรงเปลี่ยนแปลงตามไปด้วย ทำให้ความเร็วของมอเตอร์ไฟฟ้ากระแสตรงเปลี่ยนแปลงตามสมการที่ (2.3)

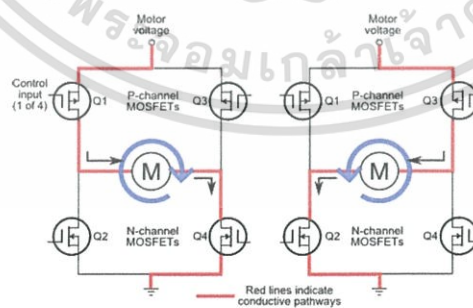
$$\%Duty\ cycle = \frac{Pulse\ Width}{Cycle} \times 100 \quad (2.3)$$

Duty Cycle จะคิดเป็นเปอร์เซ็นต์ของ Cycle และค่าเฉลี่ยของแรงเคลื่อน Output ขับโหลดได้จากสมการที่ (2.4)

$$V_{output} = \%Duty\ cycle \times V_{Supply} \quad (2.4)$$

ตามสมการที่ (2.4) พบว่าแรงเคลื่อน Output ที่จะออกไปขับมอเตอร์ไฟฟ้ากระแสตรงให้หมุนที่ความเร็วเปลี่ยนไปนั้น จะแปรผันโดยตรงกับค่า %Duty Cycle หรือความกว้างของสัญญาณควบคุม PWM นั่นหมายความว่าสามารถเขียนโปรแกรมผลิตสัญญาณ PWM ที่เปลี่ยนค่า Pulse Width ให้ออกมาควบคุมวงจรขับมอเตอร์ไฟฟ้ากระแสตรงได้ ก็จะสามารถควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงได้

ในกรณีต้องการควบคุมทั้งความเร็วและทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรง จำเป็นต้องใช้วงจร H-Bridge เพื่อให้สามารถเป็นสวิตซ์ตัด-ต่อกระแสไฟควบคุมมอเตอร์ไฟฟ้ากระแสตรงแทนหน้าสัมผัสรีเลย์ที่เร็วพอที่จะใช้สัญญาณ PWM ควบคุมความเร็วไปพร้อมกับการควบคุมการกลับทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรง



รูปที่ 2.8 การเคลื่อนที่ของมอเตอร์ไฟฟ้ากระแสตรงในวงจร H-Bridge ที่ควบคุมด้วยมอสเฟต

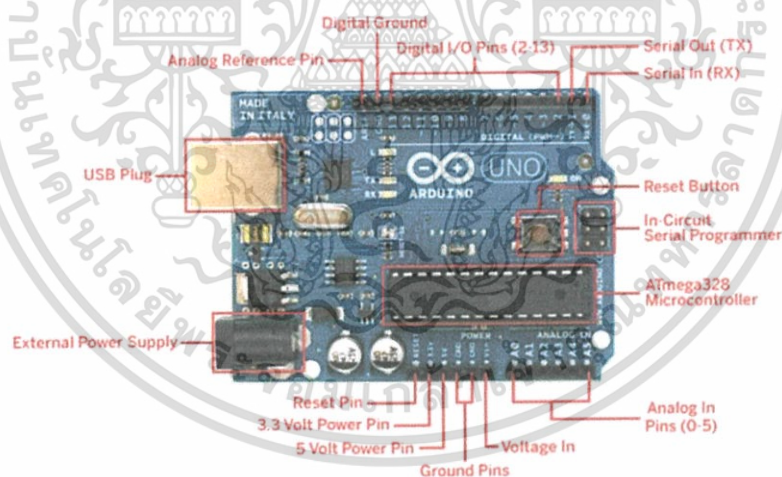
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปวงจรที่ 2.8 ประกอบด้วยมอสเฟต จำนวน 4 ตัวทำหน้าที่เป็นสวิตช์เปิด-ปิด (Q1-Q4) เมื่อส่งสัญญาณควบคุมให้มอสเฟต Q1 และ Q4 ทำงาน และปิดการทำงานของมอสเฟต Q2 และ Q3 จะทำให้กระแสไหลจากทิศทาง Q1 ไปยัง Q4 ส่งผลให้มอเตอร์ไฟฟ้ากระแสตรงเริ่มหมุนไปทางขวา และเมื่อส่งสัญญาณควบคุมให้มอสเฟต Q3 และ Q2 และปิดการทำงานของมอสเฟต Q1 และ Q4 ทำให้กระแสไหลจากจุด Q3 ไป Q2 ส่งผลให้มอเตอร์ไฟฟ้ากระแสตรงหมุนกลับทิศไปทางซ้าย

2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)

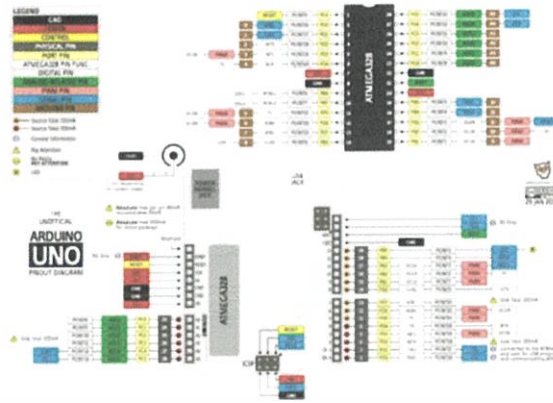
2.3.1 คุณสมบัติและการต่อใช้งาน Microcontroller Arduino Uno R3

Microcontroller Arduino Uno R3 เป็น Microcontroller Board ที่ใช้ ATmega328 เป็น MCU หลักที่รวบรวมอุปกรณ์สนับสนุนการทำงานของ CPU ไว้มากมาย อาทิเช่น Analog to Digital, SPI, UART, Timer, Counter และ PWM ซึ่งอุปกรณ์สนับสนุนการทำงานเหล่านี้ทำให้ MCU สามารถทำงานได้กว้างขวางบอร์ดไมโครคอนโทรลเลอร์จะมีขา Digital 14 ขา อินพุต/เอาต์พุต (สามารถทำเป็น PWM ได้ถึง 6 ขา) และมีขา Analog อินพุตได้อีก 6 ขา, Run ที่ความถี่ 16 MHz มี USB Connector และ Power Jack DC ดังรูปที่ 2.9 และรูปที่ 2.10



รูปที่ 2.9 โครงสร้างภายนอกของ Microcontroller Arduino Uno R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ตำแหน่งของขาบน ATMEGA328

2.3.1.1 ฟังก์ชัน ADC

ปกติใน CPU ของ AVR - ATMEGA32 นั้นจะมีฟังก์ชัน ADC อยู่ภายในตัวไอซี สำหรับฟังก์ชัน ADC นี้สามารถรับสัญญาณอนาล็อกได้สูงสุด 8 Channel โดยรับสัญญาณเข้ามาทางพอร์ต A ซึ่งสามารถเลือกใช้ฟังก์ชันนี้ทำการแปลงสัญญาณอนาล็อกทีละ Channel อย่างต่อเนื่อง หรือจะให้การแปลงสัญญาณเฉพาะ Channel ที่ต้องการได้เช่นกัน โดยสัญญาณดิจิทัลที่แปลงได้จะมีความละเอียด 10 บิต โดยการรับสัญญาณแต่ละขาของพอร์ต A

2.3.1.2 ฟังก์ชัน PWM

เนื่องจากว่า PWM เป็นฟังก์ชันการทำงานหนึ่งในโหมด PWM ของ Timer/Counter ที่อยู่ใน AVR - ATMEGA32 ดังนั้นจึงเกี่ยวข้องกับการทำงานของ Timer/Counter ของ AVR - ATMEGA32

2.3.1.3 Timer /Counter

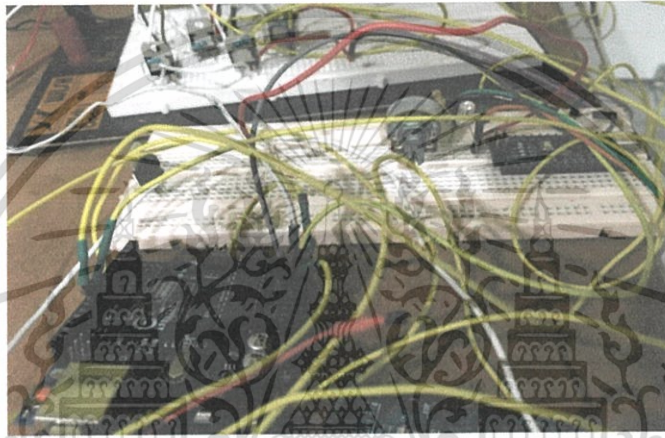
ภายใน AVR - ATMEGA32 จัดให้มี Timer/Counter 3 ชุด โดยจัดเป็น Timer/Counter ขนาด 8 บิต 2 ชุด และ Timer/Counter ขนาด 16 บิต 1 ชุด ดังนี้คือ Timer/Counter2 , Timer/Counter0 และ Timer/Counter1 ตามลำดับ ซึ่ง Timer/Counter2 สามารถรับสัญญาณ Clock จากภายนอก ซึ่งเป็น Option ที่จะนำ Timer/Counter2 มาทำเป็น RTC โดยใช้ XTAL ที่มีค่าความถี่เท่ากับ 32.768KHz มาเป็นฐานเวลา และ Timer/Counter0 และ Timer/Counter1 ใช้วงจร Prescaling ขนาด 10 บิตร่วมกัน ส่วน Timer/Counter2 ใช้วงจร Prescaling แยกออกต่างหาก

บทที่ 3

การออกแบบวงจร การสร้างแบบจำลอง และการเขียนโปรแกรม

ในการจัดทำโครงการนี้นอกจากจะอาศัยทฤษฎีข้างต้นดังที่ได้กล่าวมาแล้ว ยังอาศัยวงจร H-bridge ก็จะส่งผลสามารถทำงานได้ตรงตามเป้าหมาย ซึ่งมีรายละเอียดวงจรที่เกี่ยวข้องกับการควบคุมมอเตอร์ไฟฟ้ากระแสตรงของแบบจำลองสายพานดังนี้

3.1 วงจรควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงพื้นฐาน



รูปที่ 3.1 การต่อวงจรอย่างง่ายควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรงโดยใช้ตัวต้านทานปรับค่าได้

ในขั้นตอนแรกของการศึกษาการควบคุมมอเตอร์ไฟฟ้ากระแสตรง จะเริ่มจากการทดลองควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสตรง ซึ่งสามารถต่อวงจรอย่างง่ายได้ดังรูปที่ 3.1 โดยมีตัวต้านทานชนิดปรับค่าได้ เป็นอุปกรณ์หลักที่ใช้ในการควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงและใช้ Microcontroller Arduino Uno R3 เป็นชุด Microcontroller ที่ใช้ในการเขียนโปรแกรมควบคุม

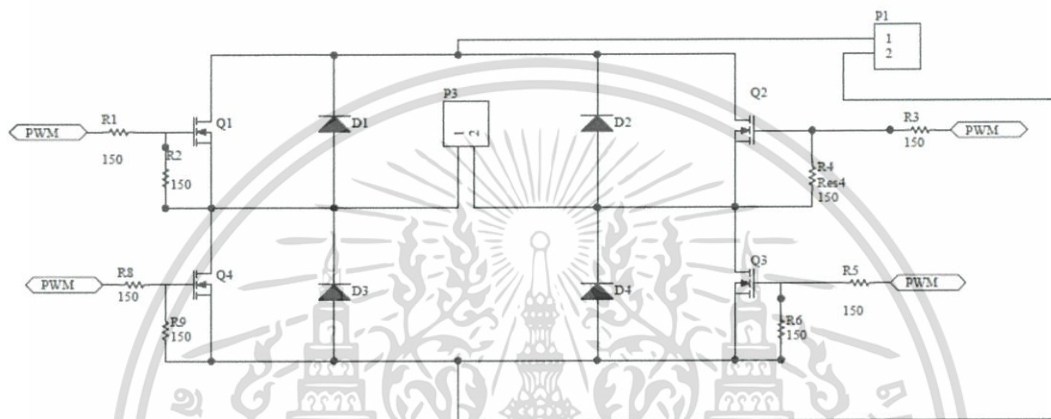
โปรแกรมควบคุมความเร็วด้วยการใช้ตัวต้านทานแบบปรับค่าได้

ลักษณะของโปรแกรมนั้น ต้องการสัญญาณรับค่าความต้านทานจากตัวต้านทานชนิดปรับค่าได้ โดยค่าที่อ่านมาจะเป็นค่าอนาล็อกจากตัวต้านทานปรับค่าได้ ซึ่งจะให้ค่าความละเอียด 10 บิต หรือค่าเท่ากับ 1024 ในระบบฐานสิบ ดังนั้นเมื่อต้องการให้มอเตอร์ไฟฟ้ากระแสตรงสามารถหมุนทั้งซ้ายและขวา จึงนำค่าความต้านทานสำหรับการกำหนดตำแหน่งตรงกลาง เมื่อหมุนไปตามเข็มนาฬิกา มอเตอร์ไฟฟ้ากระแสตรงก็จะหมุนไปในทิศทางหนึ่งด้วยความเร็วที่เพิ่มขึ้นเรื่อยๆ

3.2 วงจรขับมอเตอร์ไฟฟ้ากระแสตรง (Drive Motor Circuit)

3.2.1 วงจรขับมอเตอร์ไฟฟ้ากระแสตรงโดยใช้ Microcontroller Arduino

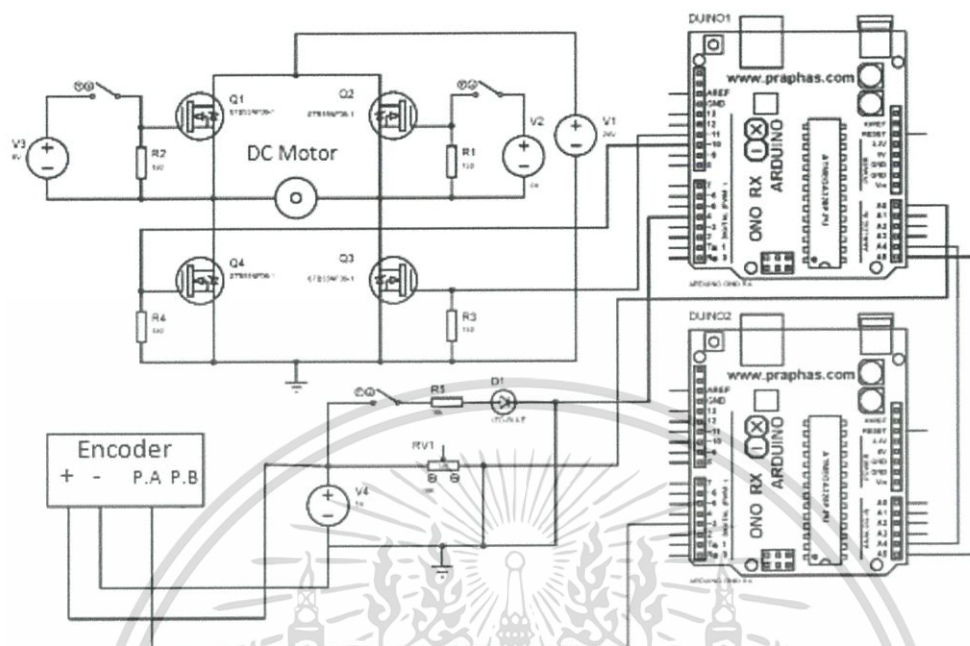
สำหรับวงจรขับมอเตอร์ไฟฟ้ากระแสตรงตามทฤษฎีที่ได้กล่าวมาข้างต้นนั้น ได้เลือกใช้มอสเฟต จำนวน 4 ตัวมาต่อกันแบบวงจร H-Bridge เพื่อใช้ในการควบคุมทิศทางของมอเตอร์ไฟฟ้ากระแสตรง ส่วน Microcontroller ที่ใช้เป็นบอร์ด Microcontroller Arduino Uno R3 ซึ่งมี ATmega 328 เป็นหน่วยไมโครคอนโทรลเลอร์หลักในการควบคุม



รูปที่ 3.2 การต่อวงจร H-Bridge

จากรูปที่ 3.2 เป็นการออกแบบวงจร H-Bridge โดยอุปกรณ์หลักสำหรับวงจรนี้คือมอสเฟตซึ่งเลือกใช้เบอร์ STP55NF06 จำนวน 4 ตัว และใช้ไฟจาก Power Supply ขนาด 24V ในการขับวงจร

3.2.2 วงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงที่มี Encoder ควบคุมค่าความเร็ว

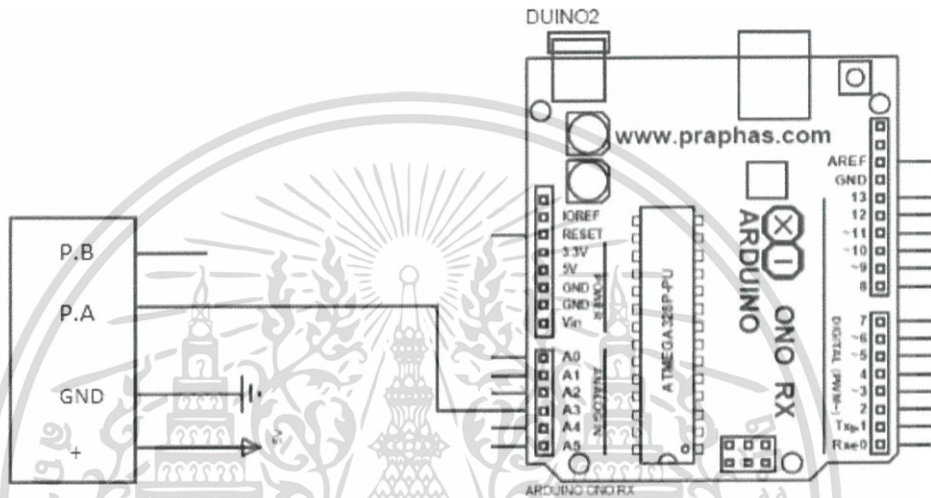


รูปที่ 3.3 การต่อวงจร H-Bridge กับ Encoder

จากรูปที่ 3.3 เป็นการออกแบบวงจร H-Bridge โดยมีการต่อ Encoder 1 ตัว ร่วมกับ Microcontroller ArduinoUno 2 ตัวโดยมีความแตกต่างจากรูปที่ 3.2 คือ ที่ Mosfet N-Channel ที่ตำแหน่ง Q3 และ Q4 จะใช้การควบคุมสัญญาณจาก Microcontroller ArduinoUno ตัวที่ 1 และ Mosfet N-Channel ที่ตำแหน่ง Q1 และ Q2 ใช้ Switch ร่วมกับ DC Source ในการ ON-OFF การทำงานของ Mosfet และใช้ Microcontroller ArduinoUno ตัวที่ 2 ร่วมกับ Encoder เพื่ออ่านค่าความเร็วของมอเตอร์ไฟฟ้ากระแสตรงและให้ Microcontroller ArduinoUno ตัวที่ 2 ส่งข้อมูลให้ Microcontroller ArduinoUno ตัวที่ 1 เพื่อปรับค่า rpm ในกรณีที่มีโหลดเกิดขึ้น

3.2.3 วงจรเอนโค้ดเดอร์ (Encoder)

ในการควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรง จะใช้ Encoder เป็นตัวรับสัญญาณการเคลื่อนที่โดยตัว Encoder จะมีตัวส่งสัญญาณ 2 สัญญาณคือ สัญญาณ A และ สัญญาณ B ต่อเข้ากับตัว Microcontroller ArduinoUno ตัวที่ 2 โดยในการอ่านค่าของความเร็วมอเตอร์ไฟฟ้ากระแสตรงนั้น จะใช้สัญญาณเพียงแค่อ่านค่าของสัญญาณ A ในการอ่านค่า Pulse ที่เกิดขึ้นเนื่องจากมอเตอร์ไฟฟ้ากระแสตรงหมุนแล้วส่งข้อมูลมาให้ไมโครคอนโทรลเลอร์ใช้ในการคำนวณค่า RPM

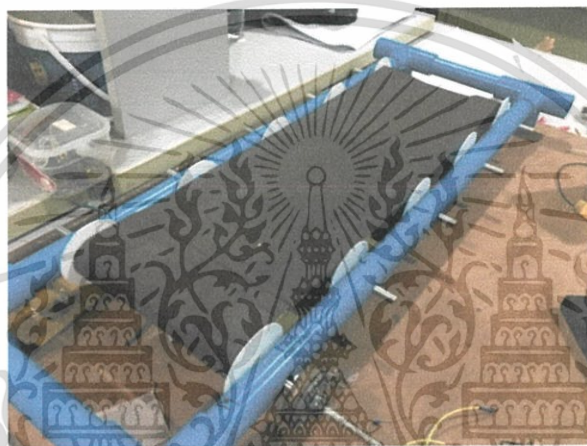


รูปที่ 3.4 การต่อวงจรเอนโค้ดเดอร์ (Encoder)

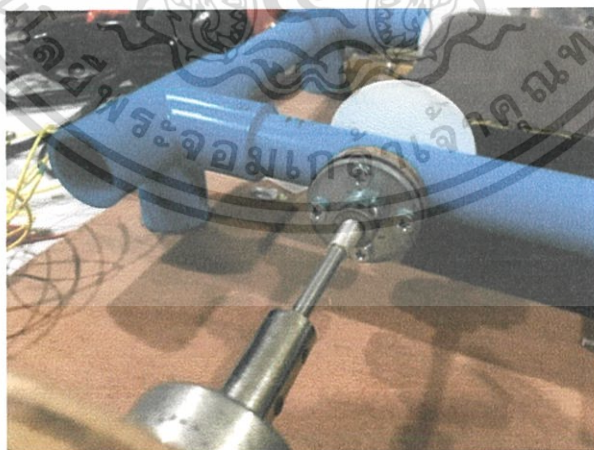
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 หลักการสร้างแบบจำลองของคอนเวเยอร์

ในการออกแบบสร้างแบบจำลองของคอนเวเยอร์จำเป็นต้องออกแบบให้สอดคล้องกับการทำงานของ Encoder เพื่อให้ค่าที่อ่านได้จาก Encoder นั้นมีความแม่นยำที่สุด จึงดำเนินการออกแบบให้มอเตอร์ไฟฟ้ากระแสตรงประกบเข้ากับเฟลาซ์ของคอนเวเยอร์ และให้ Encoder มีแกนหมุนร่วมเฟลาซ์กับ Gear box ของมอเตอร์ไฟฟ้ากระแสตรงที่ส่วนด้านหลังของคอนเวเยอร์เพื่อให้เฟลาซ์ของ Encoder และมอเตอร์ไฟฟ้ากระแสตรงขับเคลื่อนร่วมกันผ่าน Gear box เพื่อให้ได้ค่าที่แม่นยำที่สุด และสามารถนำไปประยุกต์ใช้ในอนาคต ดังรูปที่ 3.5 , 3.6 และ 3.7

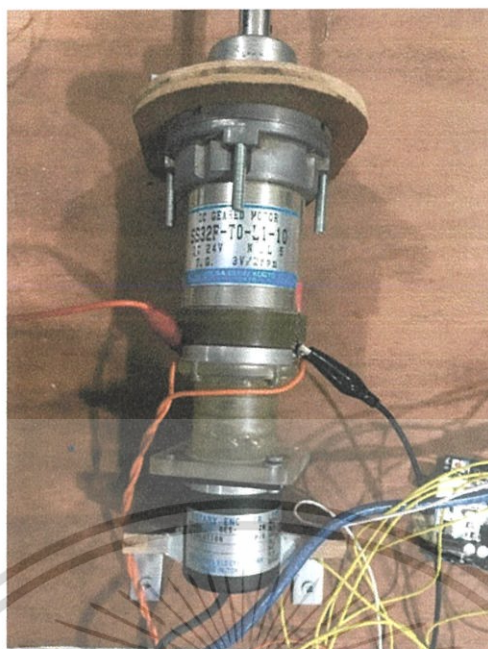


รูปที่ 3.5 แบบจำลองคอนเวเยอร์



รูปที่ 3.6 การ Coupling ระหว่างมอเตอร์ไฟฟ้ากระแสตรงและเฟลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 การ Coupling ระหว่างมอเตอร์ไฟฟ้ากระแสตรงและ Encoder

3.3.1 วัสดุหลักที่ใช้ในการสร้างแบบจำลองคอนเวเยอร์

วัสดุหลักที่ใช้ในการสร้างแบบจำลองคอนเวเยอร์

1. แผ่นไม้สำหรับทำฐาน
2. แผ่นไม้สำหรับวางมอเตอร์ไฟฟ้ากระแสตรงและEncoder
3. เหล็กฉากสำหรับยึดฐาน
4. ท่อ PVC
5. แท่งเหล็กกลม
6. Coupling Shaft
7. หนังเทียม
8. มอเตอร์ไฟฟ้ากระแสตรง 24 VDC
9. Rotary Encoder
10. น็อตและสกรู
11. เมจิกเทป (ตีนตุ๊กแก)
12. กาวร้อน, กาวอีพ็อกซี

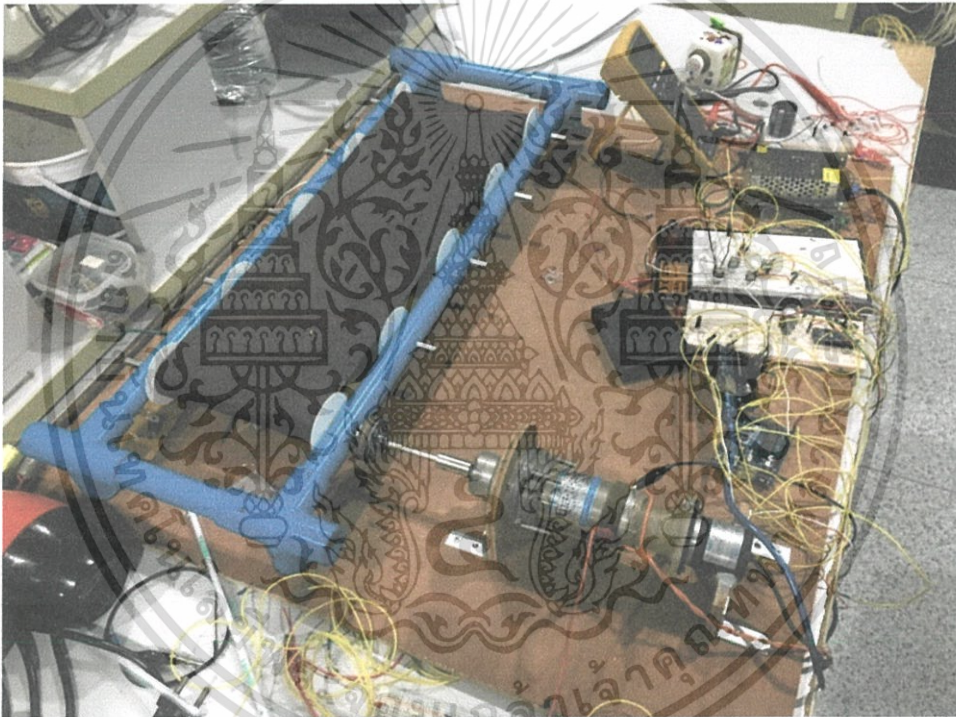
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ขั้นตอนการประกอบและติดตั้ง

จากวัสดุดังกล่าว นำมาประกอบติดตั้ง โดยเริ่มจากเจาะรูตรงกลางที่ท่อ PVC เพื่อใช้สำหรับเป็นช่อง เพื่อให้แท่งเหล็กกลมลอดผ่านและติดกาวเพื่อนำเป็นเพลาสําหรับคอนเวเยอร์ และใช้เมจิกเทปติดทั้งสองข้างของท่อ PVC โดยใช้กาวร้อนติด เพื่อให้เมจิกเทปติดกับหนังเทียมในขณะที่เพลาคอนเวเยอร์วิ่ง เสร็จแล้วจึงนำมาประกอบเข้าส่วนต่างๆ เป็นโครงคอนเวเยอร์

เมื่อทำโครงคอนเวเยอร์เสร็จแล้ว จึงติดตั้ง Coupling Shaft เพื่อให้เพลามอเตอร์ไฟฟ้า กระแสตรงและแกนคอนเวเยอร์ร่วมกันในการขับและวางบนป่าที่ติดตั้งอยู่กับฐานแผ่นไม้

ลำดับสุดท้าย จะติด Encoder ที่บริเวณ Gear Box ด้านท้ายของมอเตอร์ไฟฟ้ากระแสตรง เพื่ออ่านค่าความเร็วของคอนเวเยอร์ขณะนั้นดังรูปที่ 3.8

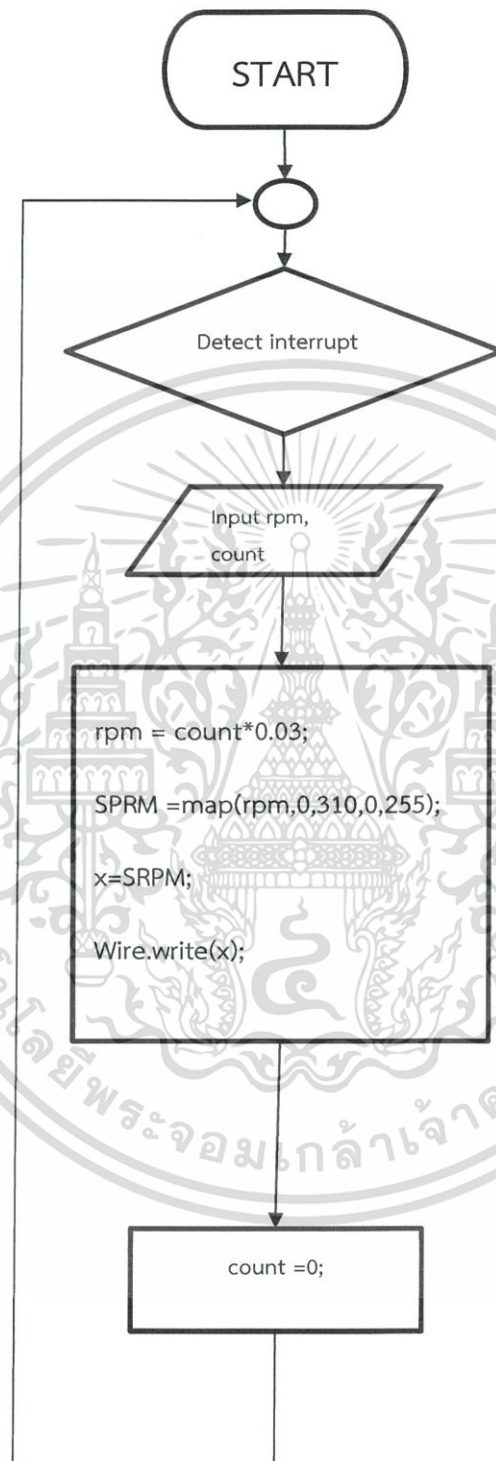


รูปที่ 3.8 แบบจำลองคอนเวเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การเขียนโปรแกรมควบคุม

3.4.1 โปรแกรมอ่านค่าจาก Encoder



รูปที่ 3.9 Flow Chart แสดงโปรแกรมอ่านค่าจาก Encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน Void Setup

```

#include <Wire.h>
volatile unsigned int count , rpm ;
unsigned long timep , shit , etime;
int SRPM;
byte x;
void setup()
{
  Wire.begin();
  Serial.begin(9600);
  pinMode(3, INPUT);
  count=0;
  timep = millis();
  attachInterrupt(1 , Pulse ,RISING);
}

```

รูปที่ 3.10 ฟังก์ชัน Void Setup ส่วน Encoder

ฟังก์ชัน Void Loop และ ฟังก์ชันอินเทอร์รัพท์ (Interrupt)

```

void loop()
{
  shit = millis();
  etime = shit - timep;
  if(etime > 1000)
  {
    detachInterrupt(1);
    rpm = count*0.03 ;
    SRPM = map(rpm, 0, 310, 0, 255);
    x = SRPM;
    Wire.beginTransmission(8); // transmit to device #8
    Wire.write(x); // sends one byte
    Wire.endTransmission(); // stop transmitting
    Serial.print("Analog = ");
    Serial.println(SRPM);
    Serial.print("RPM : ");
    Serial.println(rpm);
    Serial.print("B/S : ");
    Serial.println(count);
    Serial.println();
    count = 0;
    timep=shit;
    attachInterrupt(1 , Pulse ,RISING);
  }
}
void Pulse()
{
  count++;
}

```

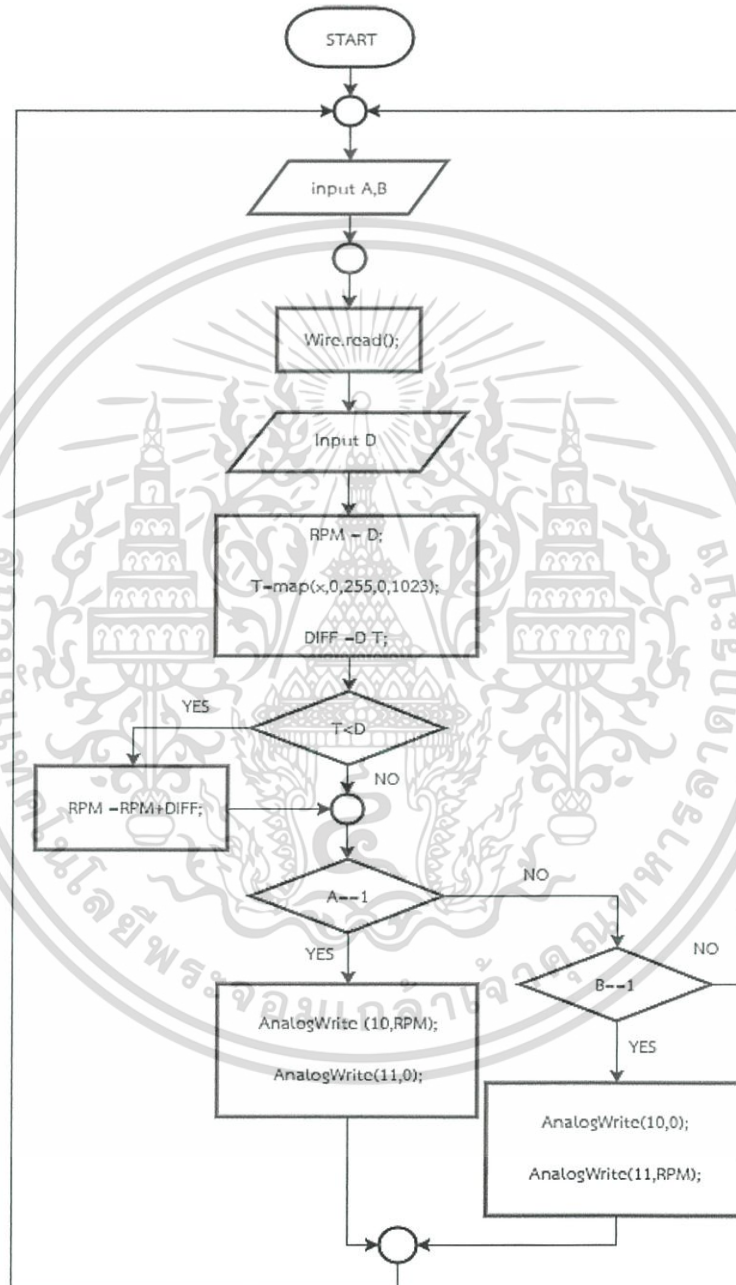
รูปที่ 3.11 ฟังก์ชัน Void Loop และ ฟังก์ชันอินเทอร์รัพท์ (Interrupt) ส่วน Encoder

จากโปรแกรมพบว่าฟังก์ชันอินเทอร์รัพท์จะทำงานเมื่อขาที่ 3 มีการทำงานแบบ Rising จะส่งให้เมื่อทำงานแบบ Rising ขึ้น 1 ครั้ง Count จะนับขึ้น 1 ครั้ง และในฟังก์ชัน Void Loop จะทำงานเมื่อระบบทำงานไปแล้ว 1 วินาที และจะเข้าลูปและจะปิดการทำงานของฟังก์ชันอินเทอร์รัพท์เมื่อนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่อินเทอร์รัพท์นับได้ใน 1 วินาทีมาคำนวณหาค่า RPM และส่งค่าให้ Microcontroller Arduino ตัวที่ 1 แล้วจะเปิดการทำงานของฟังก์ชันอินเทอร์รัพท์อีกครั้ง เพื่อนับ Pulse ในอีก 1 วินาทีเพื่อ คำนวณค่า RPM ต่อไป

3.4.2 โปรแกรมควบคุมสัญญาณ RPM



รูปที่ 3.12 Flow Chart โปรแกรมควบคุมสัญญาณ RPM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชัน Void Setup

```

#include <Wire.h>
int D,E,G,RPM,DIFF,T;
unsigned long count=0;
boolean A,B;
byte state,step;
unsigned long timep , time , etime;
unsigned long times,present = 1000;
void setup() {
  Wire.begin(8); // join i2c bus with address #8
  Wire.onReceive(receiveEvent); // register event
  Serial.begin(9600);
  pinMode(4,INPUT);
  pinMode(A0,INPUT);
  pinMode(10,OUTPUT);
  pinMode(11,OUTPUT);
  times = millis();
}

```

รูปที่ 3.13 ฟังก์ชัน Void Setup ส่วนควบคุมสัญญาณ RPM

ฟังก์ชัน Void Loop และ ฟังก์ชันประกอบ

```

void loop()
{
  void receiveEvent(int howMany)
  {
    while (Wire.available()) { // loop through all but the last
      char c = Wire.read(); // receive byte as a character
      Serial.print(c); // print the character
    }
    int x = Wire.read(); // receive byte as an integer
    //Serial.println(x); // print the integer
    D = analogRead(A0);
    RPM = D;
    T = map(x,0,255,0,1023);
    Serial.print("Analog R : ");
    Serial.println(D);
    Serial.print("Analog Encoder : ");
    Serial.println(T);
    DIFF = D - T;
    if ( T > D )
    {
      RPM = RPM + DIFF;
    }
    Serial.print("Write RPM : ");
    Serial.println(RPM);
    RPM = map(D,0,1023,0,255);
    analogWrite(10,RPM);
    E = digitalRead(4);
    G = digitalRead(5);
  }
}

```

รูปที่ 3.14 ฟังก์ชัน Void Loop และ ฟังก์ชันประกอบควบคุมสัญญาณ RPM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( E==HIGH )
{
  analogWrite(11,RPM);
  analogWrite(10,0);
  Serial.println("L MOSFET : ON");
  Serial.println();
  if(millis()-times < present)
  {
    Serial.println("1");
    times=millis();
  }
}
if(E==LOW)
{
  analogWrite(10,RPM);
  analogWrite(11,0);
  Serial.println("R MOSFET : ON");
  Serial.println();
  if(millis()-times < present)
  {
    Serial.println("2");
    times=millis();
  }
}
if(E==LOW && G==LOW)
{
}
}
delay(1000);
}

```

รูปที่ 3.15 ฟังก์ชัน Void Loop และ ฟังก์ชันประกอบควบคุมสัญญาณ RPM(ต่อ)

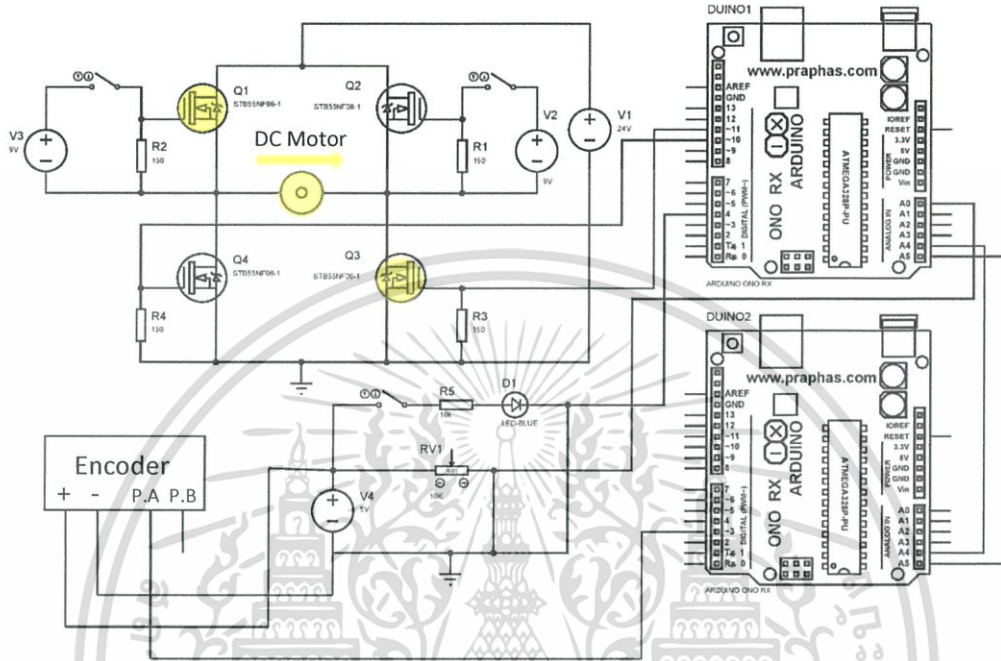
จากโปรแกรมพบว่าในฟังก์ชัน Void Setup จะมีการรับค่า RPM จากอ่าน Encoder จาก Microcontroller ArduinoUno ตัวที่ 2 มา Microcontroller ArduinoUno ตัวที่ 1 และจะไปทำงานที่ฟังก์ชัน ReceiveEvent เพื่อนำค่า RPM ที่ได้มาเปรียบเทียบกับค่า RPM ที่ตั้งไว้กับตัวต้านทานปรับค่าได้ และจะส่งค่า RPM ไปให้ Mosfet n-channel ตัวที่ Q3 และ Q4 ที่กำหนดไว้ โดยเมื่อตัวใดทำงานอยู่อีกตัวจะปิดการทำงานเพื่อขับมอเตอร์ไฟฟ้ากระแสตรงแบบ Forward Motoring และ Reverse Motoring

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองการขับมอเตอร์ไฟฟ้ากระแสตรง Mode Forward

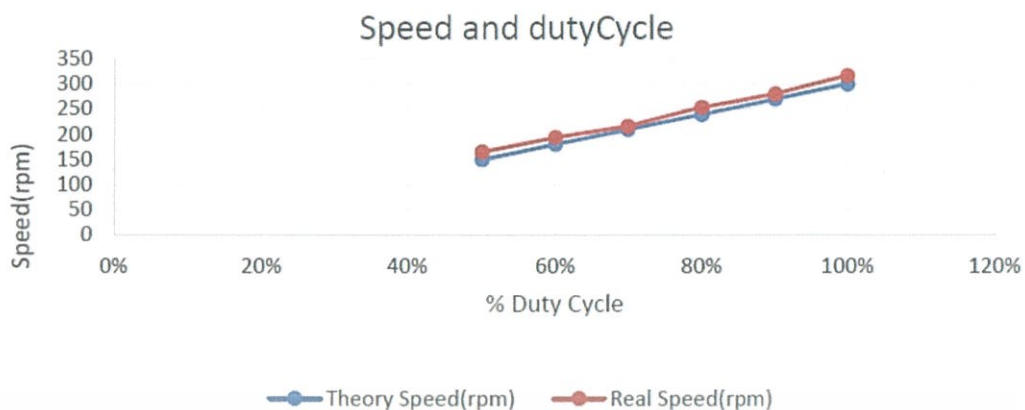


รูปที่ 4.1 รูปแสดงการทำงานแบบ Forward Motoring

ตารางที่ 4.1 แสดงความสัมพันธ์ระหว่าง %Duty Cycle กับความเร็วรอบมอเตอร์

% Duty Cycle	Theory Speed(rpm)	Real Speed(rpm)	% Error
50%	149.96	165	10%
60%	179.96	194	7.83%
70%	209.98	216	2.86%
80%	239.28	254	6.15%
90%	269.98	280	3.71%
100%	299.97	317	5.67%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

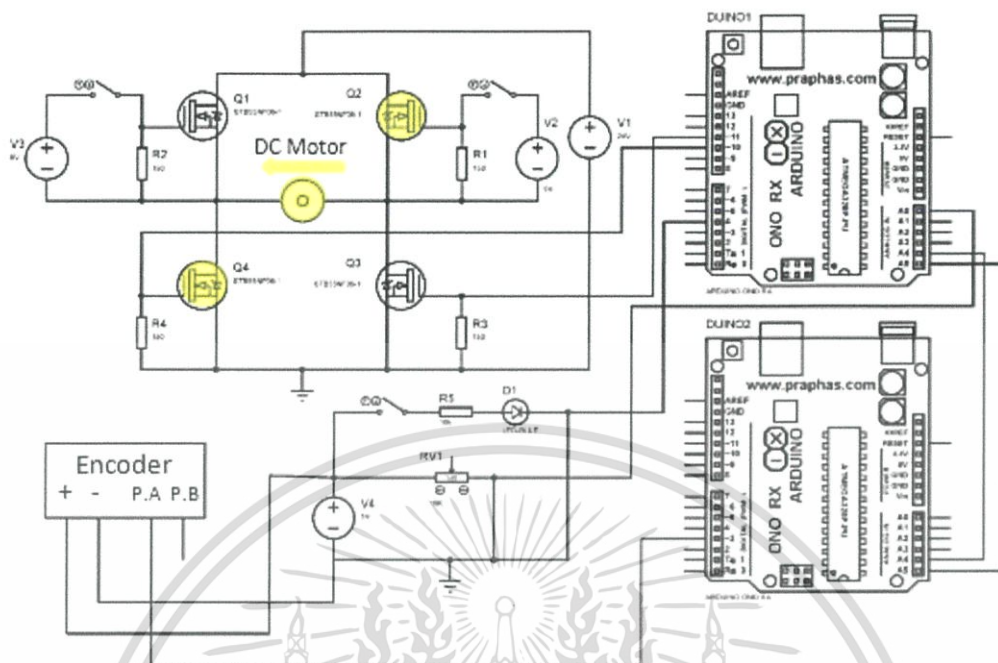


รูปที่ 4.2 รูปแสดงความสัมพันธ์ระหว่าง %Duty Cycle กับความเร็วรอบมอเตอร์

สรุปผลการทดลองที่ 4.1

ในการทดลองที่ 4.1 ได้ทำการปรับความเร็วของมอเตอร์ในโหมด Forward Motoring โดยการปรับค่า Duty cycle เป็นการควบคุมแรงดันที่ส่งไปให้มอเตอร์โดยใช้ความสัมพันธ์ $P_{in} = P_{out}$ จะทำให้สามารถคำนวณออกมาได้ว่า $V \cdot I = T \cdot W$ โดยค่าประสิทธิภาพมีค่าเป็น 75 % และค่าของแรงบิดมีค่าคงที่ ดังนั้นเมื่อแทนค่าแรงบิด T เป็นค่าคงที่ k จะทำให้ได้ความสัมพันธ์ใหม่เป็น $V \cdot I / k = W$ ดังนั้นเมื่อหาค่าคงที่ k ได้และปรับเปลี่ยนค่าของ duty cycle จะได้ว่า $V_{in} = D \cdot V_{out}$ ดังนั้นจึงทำให้สามารถปรับค่าแรงดันได้ ผ่านค่า Duty Cycle และค่าที่แสดงเป็นค่าของ Duty Cycle ที่ค่า 50 % ,60%,70%,80% และ 90% ตามลำดับโดยที่ผลการทดลองจะเป็นดังตารางที่ 4.1 และรูปภาพที่ 4.2

4.2 การทดลองการขับมอเตอร์ไฟฟ้ากระแสตรง Mode Reverse



รูปที่ 4.3 รูปแสดงการทำงานแบบ Reverse Motoring

สรุปผลการทดลองที่ 4.2

ในการทดลองที่ 4.2 ผลการทดลองนั้นมีค่าเหมือนกับการทดลองที่ 4.1 ต่างกันตรงที่ในการทดลองที่ 4.2 นี้ขับมอเตอร์ไฟฟ้ากระแสตรงในทิศทางตรงกันข้ามกับการทดลอง 4.1 และให้ผลการทดลองที่เหมือนกัน จึงสามารถสรุปผลการทดลองเป็นไปในทางเดียวกันกับการทดลองที่ 4.1

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองออกแบบและประกอบวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบสี่จุดภาค โดยใช้ มอสเฟสแบบ N-Channel 4 ตัวต่อกันแบบ H-Bridge และต่อความต้านทาน 150 โอห์มคร่อมเข้าที่ ขา Gate และ Source ของมอสเฟสแต่ละตัว จุดภาคที่หนึ่งคือการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงให้ หมุนไปด้านหนึ่งๆ จุดภาคที่สองคือ การขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงให้หมุนไปด้านตรงกันข้าม จุด ภาคที่สามและจุดภาคที่สี่คือ การหยุดมอเตอร์ไฟฟ้ากระแสตรงในแต่ละทิศทางการหมุนโดยใช้ Microcontroller Arduino UNO เป็นตัวควบคุมความเร็วรอบมอเตอร์ไฟฟ้ากระแสตรงให้คงที่ โดย จะรับสัญญาณความเร็วรอบจาก Encoder กลับเข้ามาเพื่อทำการปรับความเร็วรอบให้คงที่ เมื่อนำ มอเตอร์ไฟฟ้ากระแสตรงไปใช้เป็นตัวกำเนิดกำลังให้กับสายพานลำเลียง พบว่าสามารถเลือกความเร็วการวิ่ง ของสายพานเองได้ และเมื่อสายพานมีภาระน้ำหนักเพิ่มขึ้นระบบมีความพยายามเพิ่มแรงดันไฟฟ้าส่ง ให้กับมอเตอร์ไฟฟ้ากระแสตรง เพื่อเพิ่มความเร็วรอบขึ้นเข้าสู่จุดที่ผู้ทดลองได้ทำการตั้งไว้ โดยรวม สายพานสามารถเคลื่อนที่ได้สองทิศทางและสามารถรักษาระดับความเร็วการวิ่งไว้ได้ ซึ่งหากนำไป ประยุกต์ใช้กับระบบส่งของในสายพานจะทำให้ระบบมีความเร็วคงที่และแม่นยำเป็นผลดีต่อระบบ โดยรวม

5.2 ปัญหาที่พบและแนวทางแก้ไขปัญหา

5.2.1 ปัญหาที่พบ

ปัจจุบันระบบนี้อยู่ในช่วงของการพัฒนา ทำให้ระบบยังมีข้อจำกัดบางประการที่เกิดจากการ ทำตัวต้นแบบ ปัญหาที่พบในการทดลองนี้คือ การที่มอสเฟสแบบ N-Channel มีการทำงานผิดปกติ คือเมื่อส่งสัญญาณกระตุ้นขา Gate ในครั้งแรกจะทำให้ปิดวงจรตามปกติ แต่เมื่อยกเลิกการกระตุ้นขา Gate แล้วมอสเฟสกลับมีอาการปิดวงจรค้างไว้ และจากการทดสอบระบบสายพานพบว่าเมื่อเพิ่ม ภาระน้ำหนักลงบนสายพาน ระบบมีความพยายามชดเชยความเร็วที่ตกลงด้วยการเพิ่มแรงดันไฟฟ้า ให้กับมอเตอร์ไฟฟ้ากระแสตรง ช่วงเวลาที่ระบบมีความพยายามเพิ่มความเร็วรอบนั้นยังมีความล่าช้า แต่สามารถอยู่ในสถานะที่ทำงานได้ ในการปรับความเร็วรอบด้วยความต้านทานปรับค่าได้ไม่สามารถ ทำได้อย่างละเอียดมากนักซึ่งจำเป็นต้องใช้การแก้ไขปัญหาด้วยซอฟต์แวร์ควบคุม

5.2.2 แนวทางแก้ไข

1. ฮาร์ดแวร์ การที่มอสเฟสแบบ N-Channel มีการทำงานผิดปกติ คือ เมื่อส่ง สัญญาณกระตุ้นขา Gate ในครั้งแรกจะทำให้ปิดวงจรตามปกติ แต่เมื่อยกเลิกการกระตุ้นขา Gate แล้วมอสเฟสกลับมีอาการปิดวงจรค้างไว้ สามารถแก้ไขได้ด้วยการต่อตัวขนาด 150 โอห์มคร่อมขา

Gate และ Source ของมอสเฟสเพื่อทำการคลายประจุไฟฟ้าออกจากขา Gate ซึ่งทำให้การทำงานของมอสเฟสกลับเป็นปกติ

2. **ซอฟต์แวร์** ได้มีการแก้ไขโปรแกรมให้มีการตอบสนองที่ดีขึ้นทำให้ความเร็วรอบของสายพานเข้าสู่ความเร็วที่ตั้งไว้ด้วยเวลาที่เร็วขึ้น ส่วนในการรับคำสั่งความเร็วรอบได้ใช้การปรับค่า Duty Cycle ผ่านตัวโปรแกรมซึ่งทำให้มีความละเอียดในการปรับเลือกความเร็วของมอเตอร์ไฟฟ้า กระแสตรง

5.3 ข้อเสนอแนะและแนวทางการพัฒนา

ระบบนี้ยังสามารถพัฒนาให้มีการทำงานที่แม่นยำขึ้น โดยการใช้ตัวไมโครคอนโทรลเลอร์ที่มีการทำงานที่เร็วและเสถียรมากขึ้น ตัวโครงสร้างสายพานหากนำไปประยุกต์ใช้ในอุตสาหกรรมต้องมีการพัฒนาให้แข็งแรงมากขึ้น การทดลองนี้เป็นการทำเพื่อหาหลักการที่จะควบคุมมอเตอร์ไฟฟ้า กระแสตรงซึ่งเมื่อนำไปประยุกต์ใช้งานจริงในระบบอุตสาหกรรม จะทำให้สามารถสร้างสายพานลำเลียงที่มีความเที่ยงตรงสูง ทำให้ระบบการผลิตมีความน่าเชื่อถือและเป็นข้อได้เปรียบเชิงธุรกิจและได้ผลผลิตทันตามเวลาที่กำหนด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Muhammad Harunar Rashid, “Power Electronics – Circuit Device and Application,” Power Electronics – Circuit Device, vol.4, February 2007
- [2] “High side driver and low side driver” [Online]. Available:
<https://electronics.stackexchange.com/question/188745/high-side-driver-and-low-side-driver>
- [3] “Arduino Uno” [Online]. Available:
<https://www.arduino.cc>
- [4] “Switching Power Supply” [Online]. Available :
https://en.wikipedia.org/wiki/Switched-mode_power_supply
- [5] “Dc motor” [Online]. Available :
<https://www.electrical4u.com/dc-motor-or-direct-current-motor/>
- [6] “Rotary encoder” [Online]. Available :
https://en.wikipedia.org/wiki/Rotary_encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้