การออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่

DESIGN AND BUILD OF THE MOBILE ROBOT

นางสาวศุภากร ตันสกุล
MS. SUPAKORN TANSAKUL
นายเอกรัฐ ผิวเพ็ง
MR.EKARAT PEWPENG

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่

DESIGN AND BUILD OF THE MOBILE ROBOT

นางสาวศุภากร ตันสกุล

MS. SUPAKORN TANSAKUL

นายเอกรัฐ ผิวเพ็ง

MR. EKARAT PEWPENG

สขหมู่.....
เลขทะเบียน 144432
วัน.เดือน.ปี 24 พ.ย. 2559

.b 12816838
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

DESIGN AND BUILD OF THE MOBILE ROBOT

MS. SUPAKORN TANSAKUL

MR. EKARAT PEWPENG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2015

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่

DESIGN AND BUILD OF THE MOBILE ROBOT

นักศึกษา

นางสาวศุภากร ตันสกุล	รหัสประจำตัว 55011255
นายเอกรัฐ ผิวเพ็ง	รหัสประจำตัว 55011489

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(ดร.พลชัย โชติปรายนกุล)

หัวข้อปริญญานิพนธ์	การออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่
นักศึกษา	นางสาวศุภากร ตันสกุล
	นายเอกรัฐ ผิวเพ็ง
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2558
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ดร.พลชัย โชติปรายนกุล

บทคัดย่อ

โครงงานการออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่ มีวัตถุประสงค์เพื่อออกแบบและสร้างหุ่นยนต์เคลื่อนที่ในลักษณะการทำงานของล้อตีนตะขาบ ที่สามารถเคลื่อนที่ได้บนหลายสภาพพื้นผิว และแบกรับน้ำหนักสิ่งของได้ถึง 17 กิโลกรัม ใช้ในการจัดการด้านการลำเลียงวัสดุอุปกรณ์หรือชิ้นส่วนต่างๆ ในอุตสาหกรรมแทนการใช้แรงงานมนุษย์ ซึ่งในการออกแบบจะแบ่งเป็น 2 ส่วนหลักๆ คือ การออกแบบทางด้านฮาร์ดแวร์ และการออกแบบทางด้านซอฟต์แวร์ ในการออกแบบส่วนของฮาร์ดแวร์จะใช้โปรแกรม Solidwork ออกแบบชิ้นส่วนทั้งหมด และใช้แผ่นเหล็กมาทำเป็นตัวโครงหุ่นยนต์ประกอบกับมอเตอร์ เพลา และโซ่ เพื่อใช้ในการขับเคลื่อน ซึ่งชิ้นส่วนอิเล็กทรอนิกส์ที่ใช้ ได้แก่ มอเตอร์ขับเคลื่อน ไมโครคอนโทลเลอร์ แบตเตอรี่ และอื่นๆ จะถูกติดตั้งไว้ภายในตัวฐานหุ่นยนต์ทั้งหมด ในส่วนของซอฟต์แวร์ จะสร้างชุดคำสั่งสำหรับควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ ด้วยโปรแกรม Visual Basic และ Matlab ในการเขียนชุดคำสั่ง เพื่อโหลดเข้าไมโครคอนโทลเลอร์ ที่ใช้เป็นส่วนควบคุมมอเตอร์ที่บริเวณล้อของหุ่นยนต์ ซึ่งสามารถควบคุมการทำงานได้ทั้งระบบแมนนวลผ่านอินเตอร์เฟสและเคลื่อนที่ตามวัตถุได้ด้วยตัวเอง สำหรับการทดสอบการใช้งานของฐานหุ่นยนต์เคลื่อนที่จะทดสอบการแบกรับน้ำหนักที่ต้องการ และการเคลื่อนที่ไปได้ตามวัตถุที่ถูกวางไว้บนเส้นทางที่ต้องการนั้นๆ ซึ่งผลการทดสอบแสดงให้เห็นถึงลักษณะการเคลื่อนที่ที่สามารถเคลื่อนที่ไปตามเส้นทางและพื้นที่ที่ต้องการได้ อย่างไรก็ตามความแม่นยำในการเคลื่อนที่ และความสามารถในการแบกรับสิ่งของยังสามารถปรับปรุงให้ดีขึ้นได้ต่อไปได้อีก

Thesis Title	Design and Build of The Mobile Robot
Student	Ms. Supakorn Tansakul
	Mr. Ekarat Pewpeng
Degree	Bachelor of Engineering in Industrial Engineering
	King Mongkut's Institute of Technology Ladkrabang
Academic Year	2015
Thesis Advisor	Dr. Pholchai Chotiprayanakul

ABSTRACT

This project presents design and built of a mobile robot. The objective of this project is to build a robot with two caterpillar wheels controlled to follow an object autonomously. This mobile robot has all terrain capability and carries parts and equipment up to 17 kg of weight for transportation in industrial use in order to replace human worker. The project contains both hardware and software parts. All robot's hardware was designed with Solidwork program and made by steel sheet cut and assembled with motors and caterpillar chain wheels. All electronics parts include motor drivers, a microcontroller board, battery and else were installed inside the body of the robot. On the software part, the controller is programmed with visual basic and Matlab for interface to a computer that has a manual control program and an object following program. Finally, the mobile robot is tested to carry a required load and move tracking objects that put along a desired path.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี กลุ่มผู้จัดทำขอกราบขอบพระคุณทุกคนที่มีส่วนเกี่ยวข้องส่งผลให้ปริญญานิพนธ์นี้เสร็จสมบูรณ์ โดยเฉพาะอย่างยิ่ง ดร.พลชัย โชติปรายนกุล อาจารย์ที่ปรึกษาปริญญานิพนธ์ สำหรับการให้คำปรึกษา ความรู้ แนวคิด และคำแนะนำที่ดี ตลอดจนความช่วยเหลือและความเอาใจใส่ในทุกๆ ด้าน

ขอขอบคุณอาจารย์คณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมอุตสาหการทุกท่านที่ให้ความช่วยเหลือ และคำแนะนำที่ดี

ขอขอบคุณบุคคลากร และเจ้าหน้าที่ห้องปฏิบัติการ ภาควิชาวิศวกรรมอุตสาหการทุกท่านที่ได้จัดเตรียมสิ่งอำนวยความสะดวกต่างๆ รวมทั้งการให้ความช่วยเหลือและคำแนะนำที่ดีในการใช้เครื่องจักร เพื่อให้การทำปริญญานิพนธ์นี้สำเร็จลุล่วง

ขอขอบคุณคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับทุนสนับสนุน และการจัดกิจกรรมเพื่อนำเสนอผลงานในการจัดทำปริญญานิพนธ์ฉบับนี้

ขอขอบคุณผู้เกี่ยวข้องคนอื่นๆ ที่มิได้กล่าวถึงไว้ ณ ที่นี้ ที่ได้ให้คำปรึกษาและเป็นกำลังใจที่ดีมาโดยตลอด จนปริญญานิพนธ์เสร็จสมบูรณ์

นางสาวศุภากร ตันสกุล

นายเอกรัฐ ผิวเพ็ง

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ซ

สารบัญรูปภาพ ..... ฌ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของโครงงาน 2

1.4 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 การแบ่งประเภทของหุ่นยนต์ตามลักษณะการใช้งาน 3

2.2 การควบคุมและรูปแบบการเคลื่อนที่ของหุ่นยนต์ชนิดเคลื่อนที่ได้ (Mobile Robot) 3

2.2.1 การควบคุมการทำงานของหุ่นยนต์แบบเคลื่อนที่ได้ 4

2.2.2 การเคลื่อนที่ของหุ่นยนต์ 4

2.2.3 รูปแบบการเคลื่อนที่บนพื้นด้วยล้อและสายพาน 6

2.3 ส่วนประกอบของหุ่นยนต์ 9

2.3.1 อุปกรณ์ทางด้านเครื่องกล (Mechanical) 9

2.3.1.1 การส่งกำลังโดยเฟือง (Gear) 10

2.3.1.2 การส่งกำลังด้วยโซ่ส่งกำลัง (Chain Driver) 13

2.3.1.3 เพลา (Shaft) 14

2.3.1.4 ตลับลูกปืนและปลอกสวม (Bearing and Bush) 15

หน้า

2.3.2 อุปกรณ์ขับ (Actuator) 15
2.3.2.1 มอเตอร์สเต็ปปิ้ง (Stepping Motor) 15
2.3.2.2 มอเตอร์ไฟฟ้ากระแสตรง (DC Motor) 17
2.3.3 อุปกรณ์ไฟฟ้า (Electronics) 19
2.3.3.1 เซนเซอร์แสง 19
2.3.3.2 เอ็นโคดเดอร์ 19
2.3.3.3 แบตเตอรี่ 19
2.3.3.4 อุปกรณ์แสดงผล 20
2.3.4 อุปกรณ์ควบคุม (Controller) 20
2.4 ความรู้พื้นฐานกลศาสตร์ในการคำนวณแรงบิด กำลัง และประสิทธิภาพ ของมอเตอร์ไฟฟ้ากระแสตรง 21
2.5 หลักการสร้างสัญญาณ PWM 22
2.6 การประมวลผลภาพ (Image Processing) 23
2.6.1 อุปกรณ์ไฟฟ้า (Electronics) 23
2.6.2 อุปกรณ์ไฟฟ้า (Electronics) 23
2.7 ความรู้เบื้องต้นเกี่ยวกับ Arduino 24
2.7.1 จุดเด่นของ Arduino 24
2.7.2 โครงสรางการเขียนโปรแกรม ภาษาซีของ Arduino 24

บทที่ 3 วิธีการดำเนินงาน

3.1 การออกแบบ 26
3.1.1 ขั้นตอนการออกแบบฮาร์ดแวร์ 26
3.1.2 การออกแบบรายละเอียดของฐานหุ่นยนต์เคลื่อนที่ 26
3.1.2.1 การออกแบบโครงสร้างของฐานหุ่นยนต์เคลื่อนที่ 27
3.1.2.2 การออกแบบส่วนอิเล็กทรอนิกส์และส่วนควบคุมการทำงาน 28
3.1.2.3 การออกแบบวงจรควบคุมการทำงาน 30
3.1.3 การออกแบบซอฟต์แวร์ 31

สารบัญ (ต่อ)

หน้า

3.2 ขั้นตอนการดำเนินการสร้างและประกอบส่วนของระบบส่งกำลัง 31
3.2.1 เฟืองขับโซ่ 31
3.2.2 ดุมล้อ 32
3.2.3 ตุ๊กตาลูกปืน 33
3.2.4 เพลา 33
3.2.5 เฟืองตัวหนอนและตัวหนอน 33
3.2.6 ตัวโครง 34
3.2.7 ล้อตีนตะขาบ 33
3.3 ขั้นตอนการสร้างซอฟต์แวร์ 35
3.3.1 อุปกรณ์ขับ (Actuator) 35
3.3.2 อุปกรณ์ขับ (Actuator) 36
3.4 วิธีทดสอบและวัดผล 37

บทที่ 4 ผลการดำเนินงาน

4.1 การทดสอบแรงในการขับเคลื่อนมอเตอร์ของฐานหุ่นยนต์เคลื่อนที่ 38
4.2 การทดสอบการรับน้ำหนัก 39
4.3 การทดสอบเพื่อหาประสิทธิภาพในการเคลื่อนที่บนพื้นที่ไม่ปกติ 39
4.4 การทดสอบการเคลื่อนที่ตามวัตถุของฐานหุ่นยนต์เคลื่อนที่ 40

บทที่ 5 สรุปผลการดำเนินงาน

5.1 ผลการทดสอบ 42
5.1.1 ผลการทดสอบฮาร์ดแวร์ 42
5.1.2 การออกแบบรายละเอียดของฐานหุ่นยนต์เคลื่อนที่ 43
5.2 ข้อเสนอแนะ 43

สารบัญ (ต่อ)

หน้า

เอกสารอ้างอิง.....44
ภาคผนวก ก โปรแกรมสั่งงานการเคลื่อนที่ของหุ่นยนต์.....ผก1
ภาคผนวก ข แบบรายละเอียดของหุ่นยนต์.....ผข1

สารบัญตาราง

หน้า

ตารางที่ 4.1 ผลการทดสอบการรับน้ำหนักของฐานหุ่นยนต์เคลื่อนที่ 39
ตารางที่ 4.2 ผลการทดสอบการเคลื่อนที่ของฐานหุ่นยนต์เคลื่อนที่..... 40

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 มุมมองหุ่นยนต์จากด้านบน 4
รูปที่ 2.2 รูปแบบการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ 5
รูปที่ 2.3 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Differential Drive 6
รูปที่ 2.4 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Skid Steer 6
รูปที่ 2.5 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Ackermann Steerin 6
รูปที่ 2.6 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Articulate Drive 7
รูปที่ 2.7 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Synchro Drive 7
รูปที่ 2.8 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Swirl Drive 7
รูปที่ 2.9 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Singularity Drive 8
รูปที่ 2.10 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Omni Drive 8
รูปที่ 2.11 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Mecanum Drive 9
รูปที่ 2.12 ลักษณะและส่วนประกอบของเฟืองตรง (Spur Gear) 10
รูปที่ 2.13 ลักษณะของเฟืองตัวหนอน 12
รูปที่ 2.14 โครงสร้างและส่วนประกอบของโซ่ส่งกำลัง 14
รูปที่ 2.15 ลักษณะของเพลา 15
รูปที่ 2.16 โครงสร้างภายในมอเตอร์สเต็ปปิ้ง 16
รูปที่ 2.17 โครงสร้างภายในมอเตอร์ไฟฟ้ากระแสตรง 17
รูปที่ 2.18 H-BRIDGE DRIVER DC MOTOR 18
รูปที่ 2.19 การทำงานของวงจร H-Bridge สำหรับควบคุมการหมุนของ DC Motor 18
รูปที่ 2.20 ตัวอย่างการควบคุมระยะสัญญาณสูง-ต่ำที่ต่างกัน 22
รูปที่ 2.21 ตัวอย่างฟังก์ชั่น setup() และ loop() 25
รูปที่ 3.1 แบบร่างของหุ่นยนต์เคลื่อนที่ในส่วนของระบบขับเคลื่อนและส่งกำลัง
ออกแบบโดยโปรแกรม Solidwork 27
รูปที่ 3.2 มอเตอร์รุ่น MY1016Z2 DC 24V 250W 28
รูปที่ 3.3 มอเตอร์รุ่น ZGX38RH DC 12V 29
รูปที่ 3.4 แบตเตอรี่ LION ® HGL7.3-12 12V7.3Ah/20HR (MP7.2A-12V) 29

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.5 Single Board Computer 29
รูปที่ 3.6 บอร์ด Arduino ใช้เป็นไมโครคอนโทลเลอร์ในการควบคุมมอเตอร์ 29
รูปที่ 3.7 H-BRIDGE DRIVER DC MOTOR 30
รูปที่ 3.8 การต่อมอเตอร์กระแสตรงกับเอ็นโค้ดเดอร์ในการบังคับทิศทางการเคลื่อนที่ของหุ่นยนต์ 30
รูปที่ 3.9 การตัดเฟืองขับโซ่ด้วยเครื่องพลาสม่า 32
รูปที 3.10 ลักษณะของดุมล้อหน้า 32
รูปที 3.11 ลักษณะของดุมล้อหลัง 33
รูปที่ 3.12 ลักษณะของตุ๊กตาลูกปืน 33
รูปที่ 3.13 ลักษณะของชุดเฟืองตัวหนอน 33
รูปที่ 3.14 ตัวโครงที่ได้จากการตัดพลาสมาและพับขึ้นรูป 34
รูปที่ 3.15 ลักษณะของล้อตีนตะขาบ 34
รูปที่ 3.16 แสดงหน้ายูสเซอร์ อินเตอร์เฟสที่ใช้ในการบังคับหุ่นยนต์
จากโปรแกรม Visual Studio 2012 35
รูปที่ 3.17 แผนผังการทำงานของโปรแกรมสำหรับให้หุ่นยนต์เคลื่อนที่ตามวัตถุ 36
รูปที่ 4.1 แสดงการทดสอบวัดค่าแรงในการขับเคลื่อนมอเตอร์ 38
รูปที่ 4.2 การทดสอบการเดินตามเส้นทาง 40
รูปที่ 4.3 การตรวจจับสีของวัตถุด้วยกล้อง 41
รูปที่ 5.1 การออกแบบเพื่อแก้ปัญหาโซ่หย่อน 43

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

หุ่นยนต์ คือ เครื่องจักรกลที่มีเครื่องกลไกลขับทำให้เคลื่อนไหวได้คล้ายสิ่งมีชีวิต โดยหุ่นยนต์มักถูกสร้างขึ้นมาเพื่อใช้ในงานที่เสี่ยงอันตราย หรืองานทีต้องการความแม่นยำและความละเอียดสูง หุ่นยนต์แต่ละประเภทจะมีหน้าที่การทำงานในด้านต่างๆ ตามลักษณะการใช้งานแบ่งออกได้เป็น 2 ประเภท คือ หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) และหุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) ในปัจจุบันมีโรงงานอุตสาหกรรม ที่มีการนำระบบควบคุมที่เกี่ยวข้องกับคอมพิวเตอร์มาใช้กันอย่างแพร่หลาย แต่อุตสาหกรรมของประเทศไทยถูกจัดไว้อยู่ในกลุ่มของผู้ใช้และนำเข้าเทคโนโลยี ซึ่งรวมทั้งเทคโนโลยีด้านระบบหุ่นยนต์อุตสาหกรรม และด้วยการเติบโตของอุตสาหกรรมอย่างต่อเนื่อง แน่นอนว่าความต้องการด้านแรงงานที่สูงขึ้นทำให้เกิดปัญหาต้นทุน และการขาดแคลนแรงงานตามมา จึงมีการนำหุ่นยนต์เข้ามาช่วยเหลือในงานอุตสาหกรรม ในการจัดการงานด้านการลำเลียงวัสดุอุปกรณ์ หรือชิ้นส่วนการประกอบต่างๆ จากจุดหนึ่งไปยังอีกจุดหนึ่ง แทนการใช้แรงงานคน และลดปัญหาพื้นที่การทำงานแออัดจากคนพลุกพล่าน นอกจากนี้ยังสามารถนำมาประยุกต์ใช้กับหุ่นยนต์ชนิดอื่นๆ ที่สำคัญในงานอุตสาหกรรมการผลิตได้อีก เช่น หุ่นยนต์ชนิดแขนกล ซึ่งข้อด้อยของแขนกลก็คือไม่สามารถเคลื่อนที่เคลื่อนย้ายไปมา จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งได้ ทำให้มีข้อจำกัดเกี่ยวกับขอบเขตพื้นที่ในการทำงาน

จากเหตุผลดังกล่าวข้างต้นผู้จัดทำจึงมีแนวคิดว่าการทำหุ่นยนต์เคลื่อนที่จะสามารถนำมาใช้แทนแรงงานคนได้อย่างมีประสิทธิภาพ เป็นการนำเทคโนโลยีมาประยุกต์ใช้ให้เกิดประโยชน์ช่วยเพิ่มมูลค่าให้แก่การทำงานและช่วยลดต้นทุนจากแรงงานคนได้อีกระดับหนึ่ง

1.2 วัตถุประสงค์

1. เพื่อศึกษาและพัฒนาการทำงานของหุ่นยนต์เดินตามเส้นทางอัตโนมัติ
2. เพื่อออกแบบฐานหุ่นยนต์เคลื่อนที่ให้สามารถเคลื่อนที่บนพื้นผิวขรุขระหรือพื้นที่ผิวต่างระดับ
3. เพื่อออกแบบฐานหุ่นยนต์เคลื่อนที่ที่สามารถลากจูง หรือ ขนส่ง วัสดุอุปกรณ์ ชิ้นส่วน ต่างๆได้
4. เพื่อนำไปใช้เป็นตัวฐานเคลื่อนที่ให้กับแขนกล

1.3 ขอบเขตของโครงงาน

1. ฐานหุ่นยนต์เคลื่อนที่จะต้องขับเคลื่อนได้ ด้วยแบตเตอรี่ขนาด 12 โวลล์ จำนวน 3 ก้อน และมอเตอร์ 3 ตัว ขนาด 250 วัตต์ และ 36 วัตต์

2. ฐานหุ่นยนต์เคลื่อนที่สามารถเคลื่อนที่ได้ในทางราบและขึ้นเนินได้ โดยล้อเป็นลักษณะตีนตะขาบ ด้านหน้าสามารถบังคับให้ยกขึ้นได้ 80 องศา

3. ฐานหุ่นยนต์เคลื่อนที่สามารถลากจูงหรือเคลื่อนย้ายสิ่งของที่มีน้ำหนักไม่เกิน 17 กิโลกรัมได้

4. ฐานหุ่นยนต์เคลื่อนที่ใช้การควบคุมการทำงานด้วยมือและเคลื่อนที่ตามวัตถุได้ด้วยตัวเอง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ฐานหุ่นยนต์เคลื่อนที่สามารถช่วยเพิ่มฟังก์ชันการทำงานให้กับหุ่นยนต์ชนิดแขนกลได้
2. สามารถนำไปใช้ในงานอุตสาหกรรมบางอย่างแทนแรงงานคนได้
3. สามารถนำฐานหุ่นยนต์เคลื่อนที่ไปประยุกต์ใช้ในการขนส่งหรือเคลื่อนย้ายสิ่งของได้
4. สามารถนำไปต่อยอดเป็นต้นแบบของหุ่นยนต์อัตโนมัติอื่นๆ ได้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 การแบ่งประเภทของหุ่นยนต์ตามลักษณะการใช้งาน

หุ่นยนต์ถูกแบ่งออกเป็น 2 ประเภทตามลักษณะการใช้งาน คือ 1. หุ่นยนต์ชนิดที่ติดตั้งอยู่กับที่ (Fixed Robot) เป็นหุ่นยนต์ที่ไม่สามารถเคลื่อนที่ไปไหนได้ด้วยตัวเอง มีลักษณะเป็นแขนกล สามารถขยับและเคลื่อนไหวได้เฉพาะแต่ละข้อต่อ ภายในตัวเองเท่านั้น มักนำไปใช้ในโรงงานอุตสาหกรรม เช่น โรงงานประกอบรถยนต์ แขนหุ่นยนต์ (Robot Arm, Manipulator) คือ หุ่นยนต์ที่สามารถหยิบ จับ เคลื่อนย้าย มีปฏิสัมพันธ์กับสิ่งของได้ในพื้นที่การทำงาน (Workespace) ของมัน แต่ไม่สามารถทำงานนอกพื้นที่การทำงานได้ 2. หุ่นยนต์ชนิดที่เคลื่อนที่ได้ (Mobile Robot) หุ่นยนต์ประเภทนี้จะแตกต่างจากหุ่นยนต์ที่ติดตั้งอยู่กับที่เพราะสามารถเคลื่อนที่ไปไหนมาไหนได้ด้วยตัวเอง โดยการใช้ล้อหรือการใช้ขา แขนกลเคลื่อนที่ได้ (Mobile Manipulator) เป็นการผสมกันของหุ่นยนต์แบบแขนกล กับหุ่นยนต์ชนิดเคลื่อนที่ได้ โดยการนำแขนกลมาติดตั้งบนหุ่นยนต์เคลื่อนที่ จะได้เคลื่อนที่ไปหยิบจับ สิ่งต่างๆ นอกพื้นที่การทำงานได้

2.2 การควบคุมและรูปแบบการเคลื่อนที่ของหุ่นยนต์ชนิดเคลื่อนที่ได้ (Mobile Robot)

โครงสร้างของหุ่นยนต์แบบเคลื่อนที่ได้จะประกอบไปด้วยกลไกที่ใช้สำหรับขับเคลื่อนตัวหุ่นยนต์ไปยังบริเวณสถานที่ทำงานที่ต้องการ เช่นการขับเคลื่อนด้วยล้อหมุน การขับเคลื่อนด้วยล้อสายพาน ในปัจจุบันได้มีการประยุกต์นำเอาหุ่นยนต์แบบเคลื่อนที่ได้มาใช้ในกระบวนการผลิต เช่น การใช้ยานขนส่งลำเลียงแบบอัตโนมัติ (Automated Guided Vehicles, AGV) เพื่อทำการขนส่งลำเลียงชิ้นส่วนไปยังสถานีการผลิตต่างๆ หรือนำไปทำการจัดเก็บในคลังจัดเก็บสินค้าต่อไป จุดเด่นของหุ่นยนต์ชนิดเคลื่อนที่ได้ คือ 1. หุ่นยนต์สามารถเคลื่อนที่ไปยังบริเวณพื้นที่ปฏิบัติงานที่ต้องการได้ 2. สามารถประยุกต์ไปใช้ในงานซึ่งเป็นการควบคุมการทำงานของหุ่นยนต์จากระยะทางไกล เช่น ทำหน้าที่เป็นหุ่นยนต์กู้ภัย หรือทำหน้าที่ในการเก็บกู้วัตถุระเบิดได้

2.2.1 การควบคุมการทำงานของหุ่นยนต์แบบเคลื่อนที่ได้

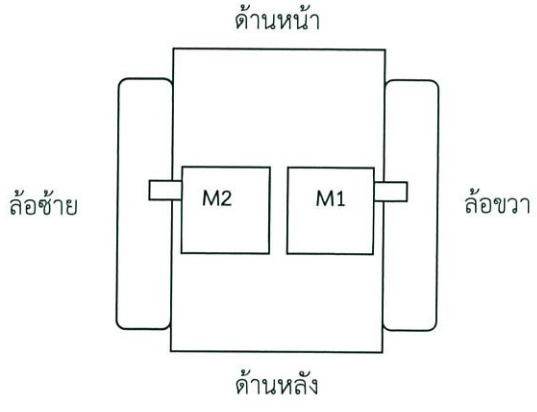
1. การควบคุมด้วยมนุษย์จากระยะทางไกล (Remote Control) วิธีการนี้มนุษย์จะควบคุมการทำงานของหุ่นยนต์ ผ่านทางสายสัญญาณผ่านทางคลื่นความถี่วิทยุหรือสัญญาณจากดาวเทียม

2. การควบคุมการทำงานด้วยระบบนำทาง เช่น การใช้แถบนำแสง หรือราง

3. การควบคุมการทำงานแบบอัตโนมัติ (Automatic Control) หรือการควบคุมการทำงานด้วยตนเอง (Autonomous Control) วิธีการนี้หุ่นยนต์จะเกิดการทำงานตามเงื่อนไข หรือข้อกำหนดที่ถูกเขียนเอาไว้ในโปรแกรม

2.2.2 การเคลื่อนที่ของหุ่นยนต์

การเคลื่อนที่ของหุ่นยนต์จะมีการออกแบบตามวัตถุประสงค์การใช้และสภาพการทำงานของหุ่นยนต์เป็นสำคัญ หุ่นยนต์แต่ละชนิดส่วนใหญ่แล้วจะใช้มอเตอร์เป็นตัวกำหนดทิศทางในการเคลื่อนที่



รูปที่ 2.1 มุมมองหุ่นยนต์จากด้านบน

จากรูปที่ 2.1 เป็นการมองหุ่นยนต์จากทางด้านบน โดยมีมอเตอร์ M1 อยู่ทางด้านขวามือ และมอเตอร์ M2 อยู่ทางด้านซ้ายมือ

1. การขับเคลื่อนหุ่นยนต์เคลื่อนที่เดินหน้าตรง ทำได้โดยสั่งให้มอเตอร์ M1 และ M2 หมุนไปข้างหน้า โดยมอเตอร์ M1 หมุนตามเข็มนาฬิกา และมอเตอร์ M2 หมุนทวนเข็มนาฬิกา

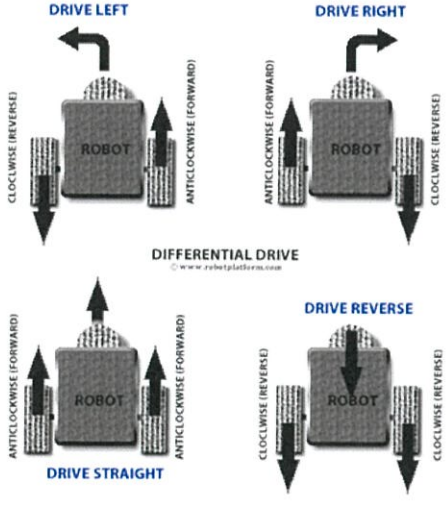
2. การขับเคลื่อนหุ่นยนต์ถอยหลังทำได้โดยสั่งให้มอเตอร์ M1 และ M2 หมุนไปด้านหลัง โดยมอเตอร์ M1 หมุนทวนเข็มนาฬิกา และมอเตอร์ M2 หมุนตามเข็มนาฬิกา

3. การขับเคลื่อนหุ่นยนต์เลี้ยวซ้าย ทำได้โดยการสั่งให้มอเตอร์ M1 และ M2 หมุนตามเข็มนาฬิกา ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่กึ่งกลางลำตัวของหุ่นยนต์ หรือจะใช้มอเตอร์ M1 แค่เพียงตัวเดียวหมุนไปตามเข็มนาฬิกา ส่วนมอเตอร์ M2 หยุดกับที่ ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่ล้อซ้ายของหุ่นยนต์ ทำให้หุ่นยนต์เคลื่อนที่เลี้ยวซ้ายได้

4. การขับเคลื่อนหุ่นยนต์เลี้ยวขวา ทำได้โดยการสั่งให้มอเตอร์ M1 และ M2 หมุนทวนเข็มนาฬิกา ซึ่งจะทำให้จุดศูนย์กลางการหมุน อยู่ที่กึ่งกลางลำตัวของหุ่นยนต์ หรือจะใช้มอเตอร์ M2 แค่เพียงตัวเดียวหมุนทวนเข็มนาฬิกา ส่วนมอเตอร์ M2 หยุดกับที่ ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่ล้อขวาของหุ่นยนต์ ทำให้หุ่นยนต์เคลื่อนที่ไปทางขวาได้

5. การขับเคลื่อนหุ่นยนต์เคลื่อนที่ถอยหลังเลี้ยวซ้าย ทำได้โดยการสั่งให้มอเตอร์ M1 และ M2 หมุนทวนเข็มนาฬิกา ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่กึ่งกลางลำตัวของหุ่นยนต์ หรือจะใช้มอเตอร์ M1 แค่เพียงตัวเดียวหมุนทวนเข็มนาฬิกา ส่วนมอเตอร์ M2 หยุดกับที่ ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่ล้อซ้ายของหุ่นยนต์ ทำให้หุ่นยนต์เคลื่อนที่ถอยหลังไปทางซ้ายได้

6. การขับเคลื่อนหุ่นยนต์เคลื่อนที่ถอยหลังเลี้ยวขวา ทำได้โดยการสั่งให้มอเตอร์ M1 และ M2 หมุนตามเข็มนาฬิกา ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่กึ่งกลางลำตัวของหุ่นยนต์ หรือจะใช้มอเตอร์ M2 แค่เพียงตัวเดียวหมุนตามเข็มนาฬิกา ส่วนมอเตอร์ M2 หยุดกับที่ ซึ่งจะทำให้จุดศูนย์กลางการหมุนอยู่ที่ล้อขวาของหุ่นยนต์ ทำให้หุ่นยนต์เคลื่อนที่ถอยหลังไปทางขวาได้ลักษณะการเคลื่อนที่ดังแสดงในรูปที่ 2.2



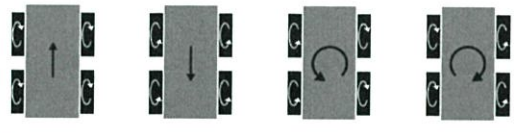
รูปที่ 2.2 รูปแบบการเคลื่อนที่ของหุ่นยนต์ในทิศทางต่างๆ
[http://www.robotplatform.com/knowledge/Classification_of_Robots/wheel_control_theory.html]

2.2.3 รูปแบบการเคลื่อนที่บนพื้นด้วยล้อและสายพาน

1. รูปแบบ Differential Drive เป็นรูปแบบการเคลื่อนที่ด้วยล้อที่ง่ายที่สุด คือมีล้อ 2 ล้อ ล้อซ้ายและล้อขวา ดังรูปที่ 2.3 ถ้าต้องการเคลื่อนที่ไปข้างหน้าตรงๆ ก็หมุนล้อซ้ายและขวาไปทางเดียวกันด้วยความเร็วเท่ากัน ถอยหลังตรงๆ ก็หมุนล้อซ้ายและขวาถอยหลังทั้งคู่ ถ้าอยากเลี้ยวซ้ายก็หมุนล้อขวาไปข้างหน้าให้เร็วกว่าล้อซ้าย อยากเลี้ยวขวาก็หมุนล้อซ้ายไปข้างหน้าให้เร็วกว่าล้อขวา หรือจะหมุนตัวอยู่กับที่ก็ได้ ก็ให้หมุนล้อข้างหนึ่งไปข้างหน้า และล้ออีกข้างหมุนกลับหลังด้วยความเร็วเท่ากัน

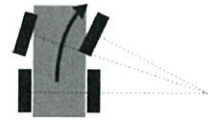
รูปที่ 2.3 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Differential Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

2. รูปแบบ Skid Steer คล้ายกันกับ Differential Drive ใช้หลักการเดียวกันเรื่องการหมุนล้อเดินหน้า ถอยหลังเพื่อเคลื่อนที่และเลี้ยว ต่างกันตรงที่ Skid Steer จะใช้เรียกกับหุ่นยนต์ที่ใช้มากกว่า 2 ล้อ ดังรูปที่ 2.4 และตีนตะขาบ เนื่องจากในหุ่นยนต์ที่มีล้อมากๆ หรือสายพานตีนตะขาบ เวลาเลี้ยวหรือหมุนตัว จะเกิดลื่นไถล (Slip) บนล้อเทียบกับพื้น แน่นอนว่าการลื่นไถลนี้ทำให้ Skid Steer ใช้พลังงาน ในการเลี้ยวเยอะกว่า Differential Drive แต่ด้วยล้อที่มากกว่า มีพื้นที่สัมผัสพื้นมากกว่าก็ทำให้มีแรงขับเคลื่อนมากกว่า



รูปที่ 2.4 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Skid Steer
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

3. รูปแบบ Ackermann Steering หรือการเคลื่อนที่แบบรถยนต์ คือ มีล้อขับเคลื่อนและล้อเลี้ยว (ล้อเลี้ยวอาจจะเป็นล้อเดียวกับล้อขับเคลื่อนก็ได้ ในกรณีนี้จะต้องมีล้อหมุนอิสระคอยประคองด้วยเหมือนรถขับเคลื่อนล้อหน้า ดังรูปที่ 2.5 เวลาเลี้ยวจะทำการหมุนล้อเลี้ยวให้ตั้งฉากกับเส้นที่ลากไปสู่จุดศูนย์กลางการเลี้ยว เพื่อทำให้ล้อทุกล้อเคลื่อนที่ไปตามเส้นโค้งของการเคลื่อนที่โดยไม่มีการลื่นไถล



รูปที่ 2.5 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Ackermann Steering
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

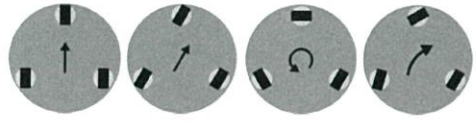
4. Articulate Drive การเคลื่อนที่คล้ายกับ Ackermann Steering แต่จะหมุนล้อเลี้ยวไปด้วยกันทั้งชุด ดังรูปที่ 2.6 ตัวอย่างเช่น หุ่นยนต์ตีนตะขาบหลายท่อน

รูปที่ 2.6 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Articulate Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

5. รูปแบบ Synchro Drive เป็นการขับเคลื่อนที่ล้อทุกล้อหมุนด้วยความเร็วเท่ากัน และมีกลไกลงบังคับเลี้ยวล้อทุกล้อไปในทิศต่างๆ พร้อมกัน ดังรูปที่ 2.7 ทำให้สามารถเคลื่อนที่ไปยังทิศใดๆ ก็ได้ (ต้องรอหมุนล้อก่อนสักครู่หนึ่ง) แต่ Synchro Drive จะไม่สามรถหมุนเลี้ยวรอบตัวเองได้

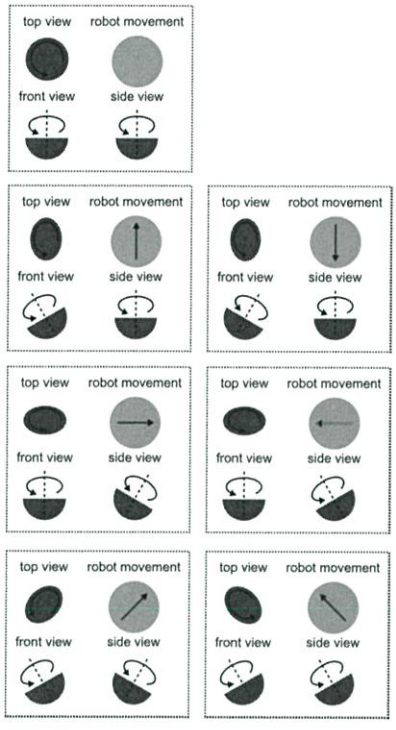
รูปที่ 2.7 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Synchro Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

6. รูปแบบ Swirl Drive การเคลื่อนที่คล้ายกับ Synchro Drive ต่างกันตรงที่ล้อทุกล้อของ Swirl Drive สามารถหมุนเลี้ยวได้เป็นอิสระต่อกัน ดังรูปที่ 2.8 ทำให้เคลื่อนที่ไปทิศใด ๆ ก็ได้ และหมุนรอบตัวเองได้



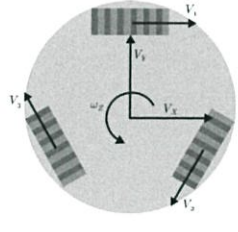
รูปที่ 2.8 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Swirl Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

7. รูปแบบ Singularity Drive ไม่ค่อยเป็นที่นิยม เป็นการเคลื่อนที่ ที่เข้าใจยากซักหน่อย คือจะใช้ล้อทรงส่วนของทรงกลม หมุนให้การหมุนตั้งฉากกับพื้น ดังรูปที่ 2.9 จุดที่ล้อสัมผัสกับพื้นจะมีความเร็วเป็นศูนย์ ถ้าต้องการเคลื่อนที่ไปทิศทางใด ก็เอียงล้อ ทำให้จุดที่ล้อสัมผัสกับพื้นมีความเร็วไม่เป็นศูนย์ การเอียงล้อทรงกลมไปยังทิศต่างๆ ทำให้เกิดแรงผลักบนพื้นในทิศต่างๆ ทำให้ Singularity Drive สามารถเคลื่อนที่ไปยังทิศใดๆ ก็ได้ (แต่ต้องรอเอียงล้อก่อนสักเล็กน้อย)



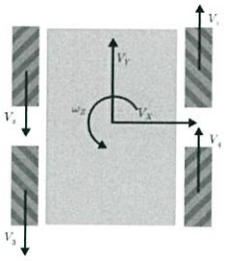
รูปที่ 2.9 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Singularity Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

8. รูปแบบ Omni Drive ล้อของ Omni Drive จะพิเศษคือ มี Roller อยู่รอบๆ ดังรูปที่ 2.10 ในทิศทางการหมุนที่ตั้งฉากกับการหมุนของล้อ Omni Drive ต้องใช้ล้ออย่างน้อย 3 ล้อ วางทำมุมกัน เนื่องจากในแต่ละล้อมี Roller อยู่ จึงเกิดการเคลื่อนที่ได้ 2 แกนพร้อมๆ กันในแต่ละล้อ คือ ตามแนวการหมุนของล้อ และตามแนวการหมุนของ Roller เมื่อต้องการเคลื่อนที่ไปยังทิศใดๆ ก็สามารถแตกความเร็วการเคลื่อนที่เข้าแนวการเคลื่อนที่ของล้อและของ Roller ได้เสมอ ทำให้ Omni Drive สามารถเคลื่อนที่ไปยังทิศใด ๆ ก็ได้ เป็น Holonomic การเคลื่อนที่ของ Omni drive จะต้องมีสมการที่เรียกว่า Jacobian/Inverse-jacobian Matrix เพื่อแปลงความเร็วของหุ่นยนต์ ไปสู่ความเร็วของล้อแต่ละล้อ



รูปที่ 2.10 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Omni Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

9. รูปแบบ Mecanum Drive การเคลื่อนที่คล้ายกับ Omni Drive ต่างกันตรงที่ Roller บนล้อไม่วางตั้งฉากกับล้อ แต่จะวางทำมุมกัน ส่วนมากจะทำมุม 45 องศา และล้อสามารถวางตัวขนานกันได้ตามปกติ ไม่ต้องวางทำมุมกัน ดังรูปที่ 2.11 การเคลื่อนที่ก็ใช้หลักการเดียวกับ Omni Drive จึงต้องมี Jacobian/Inverse-jacobian Matrix ในการคำนวณความเร็วล้อเช่นกัน แต่สมการจะง่ายกว่า Mecanum Drive เป็น Holonomic



รูปที่ 2.11 รูปแบบการเคลื่อนที่ด้วยล้อแบบ Mecanum Drive
[https://www.thairobotics.com/2013/10/11/byob-wheel-and-track/]

2.3 ส่วนประกอบของหุ่นยนต์

ในหุ่นยนต์จะประกอบด้วยอุปกรณ์และชิ้นส่วนต่างๆมากมาย ซึ่งอุปกรณ์แต่ละชนิดนั้นจะมีหน้าที่แตกต่างกันไป ตามลักษณะและวัตถุประสงค์ของการใช้งาน การเลือกใช้จึงต้องอาศัยความรู้ความเข้าใจรวมถึงความเหมาะสม เพื่อให้หุ่นยนต์สามารถทำงานได้อย่างมีประสิทธิภาพ รวดเร็ว คงทน และประหยัดพลังงานหุ่นยนต์จะแบ่งส่วนประกอบใหญ่ๆ ออกเป็น 4 ส่วน ดังนี้

1. อุปกรณ์ทางด้านเครื่องกล (Mechanical)
2. อุปกรณ์ขับ (Actuator)
3. อุปกรณ์ไฟฟ้า (Electronic)
4. อุปกรณ์ควบคุม (Controller)

2.3.1 อุปกรณ์ทางด้านเครื่องกล (Mechanical)

อุปกรณ์ทางด้านเครื่องกล คือชิ้นส่วนกลไกลต่างๆ ของหุ่นยนต์ ส่วนมากจะอยู่ในรูปของการส่งถ่ายกำลังแบบกลไก (Mechanism) คือ จะมีการส่งถ่ายกำลังจากจุดหนึ่งไปยังอีกจุดหนึ่ง เพื่อให้เกิดการเคลื่อนที่ เช่น เฟืองส่งกำลัง สายพาน โซ่ เพลา เป็นต้น

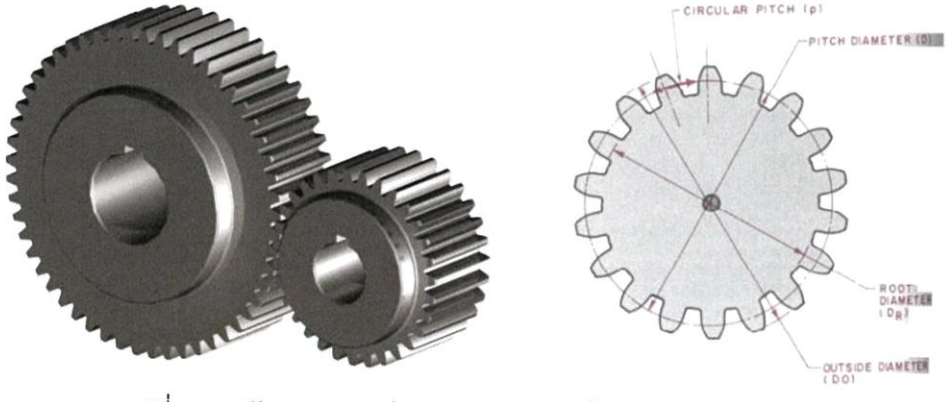
2.3.1.1 การส่งกำลังโดยเฟือง (Gear)

หน้าที่ของเฟืองแต่ละชนิดจะมีหน้าที่หลักที่เหมือนกันคือการส่งกำลังจากเพลาหนึ่งไปสู่อีกเพลาหนึ่ง โดยทิศทางการเคลื่อนที่หรือทิศทางการหมุนนั้นจะเปลี่ยนแปลงหรือไม่ก็จะต้องขึ้นอยู่กับชนิดของคู่เฟืองที่จะเลือกนำมาขบกัน

1. เฟืองตรง (Spur Gears)

เฟืองตรงเป็นเฟืองที่มีใช้งานกันมากที่สุดในบรรดาเฟืองชนิดต่างๆ จะมีลักษณะเฉพาะ คือ ฟันของเฟืองจะเป็นแนวขนานไปกับรูเพลา โดยเฟืองตรงเรียกอีกอย่างหนึ่งว่าเฟืองขนานกับเพลา (Parallel–Shaft Gear) ดังรูปที่ 2.12 โดยทั่วไปแล้วเฟืองตรงที่ใช้ส่งกำลังแต่ละคู่นั้นจะมีขนาดของฟันเฟืองหรือโมดูล (Module) เท่าๆ กัน หมุนด้วยความเร็วเชิงเส้นที่เท่ากันแต่การได้เปรียบเชิงกลที่เกิดขึ้นจะเกิดจากจำนวนฟันที่ต่างกัน (Ratio) ของเฟืองแต่ละตัว เฟืองตรงส่วนมากจะนำมาใช้ในระบบส่งกำลัง (Transmission Component) นอกจากเฟืองตรงสามารถส่งกำลังได้เฉพาะแกนเพลาที่ขนานกันเท่านั้น เฟืองตรงนั้นเหมาะสำหรับใช้งานที่ความเร็วรอบไม่สูงมากนักเพราะสามารถที่จะให้แรงฉุดและแรงบิดได้ดี

ข้อดีของเฟืองตรงคือ ขณะใช้งานจะไม่เกิดแรงกระทำในแนวแกนของเพลา ประสิทธิภาพการทำงานสูง และสามารถเพิ่มหน้ากว้างของเฟืองตรงได้อีกเพื่อเพิ่มพื้นที่ผิวสัมผัสของเฟืองได้มากขึ้น รวมทั้งลดการสึกหรอของเฟืองด้วย



รูปที่ 2.12 ลักษณะและส่วนประกอบของเฟืองตรง (Spur Gear)
[http://www.hercus.com.au/spur-gears, http://www.moro.co.th/อัตราทดเฟืองและตัวอย่า/]

การคำนวณเฟืองตรง

$$\text{ขนาดเส้นผ่านศูนย์กลางพิต (Pitch Diameter)} = \frac{\text{จำนวนฟัน} \times \text{ระยะห่างระหว่างฟัน}}{\text{Pitch}}$$

$$\text{โมดุล} = \frac{\text{ระยะห่างระหว่างฟัน}}{\text{Pitch}}$$

การหาอัตราทดของคู่เฟือง

อัตราทด (m) คืออัตราส่วนระหว่างความเร็วเชิงมุมของเฟืองขับต่อความเร็วเชิงมุมของเฟืองตาม จากความรู้ด้านกลศาสตร์จะได้ว่า

$$m_\omega = \frac{\omega_1}{\omega_2} = \frac{n_1}{n_2} = \frac{N_2}{N_1}$$

โดยที่ m_ω คือ อัตราทด

ω คือ ความเร็วเชิงมุมมีหน่วยเป็น rad/s

n คือ ความเร็วรอบมีหน่วยเป็น rpm

D คือ เส้นผ่านศูนย์กลางพิทช์มีหน่วยเป็นมิลลิเมตร

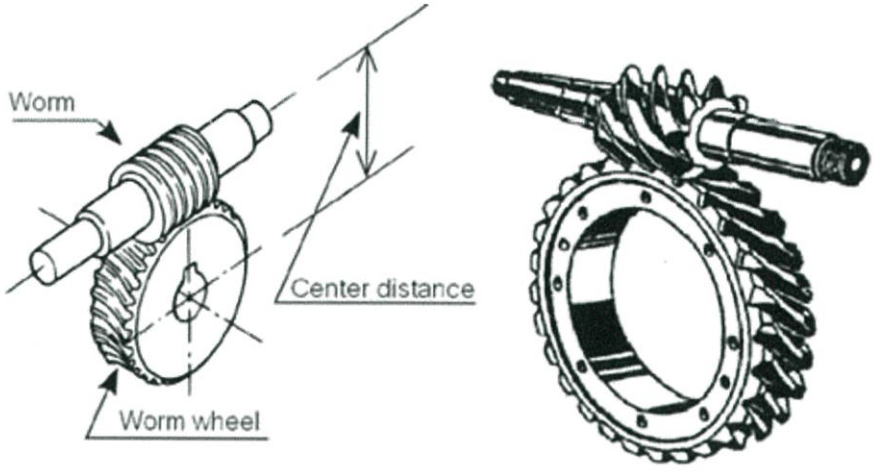
N คือ จำนวนฟัน

เลข 1 แทนเฟืองขับ

เลข 2 แทนเฟืองตาม

2. เฟืองตัวหนอน (Worm Gear)

เฟืองตัวหนอนเป็นชุดเฟืองที่ประกอบด้วยเกลียวตัวหนอน (Worm) ซึ่งมีลักษณะของเกลียวที่วางอยู่บนก้านเกลียวตัวหนอน (Shank) เหมือนลักษณะของสกรูและเฟือง (Worm Wheel) ซึ่งมีลักษณะเป็นล้อเฟืองคล้ายๆ กับเฟืองเฉียง (Helical Gear) แต่จะต่างกัน โดยเฉพาะอย่างยิ่งตรงสันฟันเฟืองจะมีลักษณะเว้าเพื่อให้รับกับความโค้งของเกลียวตัวหนอน ดังรูปที่ 2.13 แนวเพลาขับ (Worm Shaft) และเพลาตาม (Worm Wheel Shaft) ของเฟืองตัวหนอนจะทำมุมกันที่มุมฉาก 90 องศา การทำงานของเฟืองตัวหนอนจะเงียบและมีแรงสั่นสะเทือนเกิดขึ้นน้อยเนื่องจากการส่งถ่ายกำลังจากเฟืองขับไปยังเฟืองตามนั้นการส่งถ่ายกำลังจะเป็นไปในลักษณะของการลื่นไถล (Sliding) เฟืองตัวหนอนใช้ส่งกำลังระหว่างเพลาที่ไม่ขนานกันและไม่ตัดกัน ซึ่งมีอัตราทดสูง



รูปที่ 2.13 ลักษณะของเฟืองตัวหนอน

[http://thailandindustry.com/guru/view.php?id=13479&section=9]

อัตราทดของเฟืองตัวหนอน

อัตราทดของเฟืองตัวหนอนสามารถทำได้มาก เนื่องจากลักษณะเฉพาะทางรูปแบบของเฟือง โดยอัตราทดสามารถคำนวณได้จากระยะห่างระหว่างศูนย์กลางของก้านเกลียวตัวหนอน (Shank) ถึงศูนย์กลางของเฟือง (Worm Wheel) หรือที่เรียกว่าระยะห่างระหว่างศูนย์กลาง (Center Distance) โดยถ้า Center Distance ยิ่งมากแสดงว่าอัตราทดของเฟืองจะยิ่งมาก ซึ่งในบางชุดเฟืองอาจทดมากกว่า 1 ชุด โดยอาจเป็นสองหรือสามชุด ในการส่งถ่ายกำลังของเฟืองตัวหนอนนั้นความเค้นที่เกิดขึ้นบนผิวฟันเฟืองจะมากกว่าเฟืองแบบเฟืองตรงหรือแบบเฟืองเฉียง อัตราทดของเฟืองตัวหนอน (Worm Gear Ratio) สามารถคำนวณได้จากสมการ

$$m_{GW} = \frac{\omega_W}{\omega_G} = \frac{N_G}{N_W}$$

โดยที่ m_{GW} คือ อัตราทดของเฟืองตัวหนอน

ω_W คือ ความเร็วเชิงมุมของสกรูหนอน (เรเดียน/นาที)

ω_G คือ ความเร็วเชิงมุมของเฟืองเกียร์ (เรเดียน/นาที)

N_G คือ จำนวนฟันของเกียร์

N_W **คือ จำนวนฟันของสกรู**

2.3.1.2 การส่งกำลังด้วยโซ่ส่งกำลัง (Chain Driver)

โซ่ส่งกำลังจะส่งกำลังโดยอาศัยเฟืองโซ่ร่วมกันกับโซ่เป็นการส่งกำลังจากต้นกำลังไปยังชุดตามรับกำลังเพื่อส่งกำลังไปขับเคลื่อนชิ้นส่วนต่างๆ โซ่สามารถส่งกำลังให้ได้โมเมนต์บิด (หมุน) สูงมากโดยที่ให้เป็นชุดส่งกำลังมีขนาดเล็กได้ เป็นลักษณะการส่งกำลังด้วยรูปร่าง และที่รองเพลาจะรับภาระน้อยมาก ไม่มีการให้ลื่นไถลในขณะส่งกำลัง ระบบส่งกำลังด้วยโซ่จะแตกต่างจากระบบส่งกำลังด้วยสายพาน โดยสายพานจะอาศัยความฝืดเป็นหลักการทำงานให้เกิดการหมุน ส่วนโซ่จะเป็นการส่งกำลังด้วยฟันเฟืองจานโซ่ร่วมกับโซ่ การเลือกใช้ระบบส่งกำลังโดยโซ่นั้นจะเลือกใช้ก็ต่อเมื่อการทำงานไม่เหมาะสมกับการที่จะต้องใช้สายพานในการส่งกำลัง เช่น การส่งกำลังของรถจักรยาน เนื่องจากระยะห่างของเพลานั้นน้อย ซึ่งไม่เหมาะกับการใช้สายพานขับแต่สามารถใช้โซ่ขับได้ โซ่ที่นิยมใช้มีอยู่ 2 ชนิด ได้แก่ โซ่แบบลูกกลิ้ง (Roller Chains) เป็นโซ่ที่นิยมใช้กันมาก และโซ่แบบบูช (Bush Chains) ออกแบบเพื่อทำให้มีน้ำหนักเบา ใช้กับแรงเหวี่ยงได้ดี เหมาะกับเครื่องยนต์ที่ใช้ความเร็วสูง ในขณะเดียวกันก็มีข้อเสีย คือ อัตราการเสียดสีสูงขณะการหมุนทำงานและอิทธิพลของฝุ่นผงจะมีผลต่อการทำงาน ในการใช้งานจึงต้องระมัดระวังเป็นอย่างยิ่ง และในปัจจุบันไม่นิยมใช้งานกัน หันมาใช้แบบลูกกลิ้ง (Roller Chains) แทน แต่ก็ยังมีเหลือให้เห็นใช้อยู่บ้าง

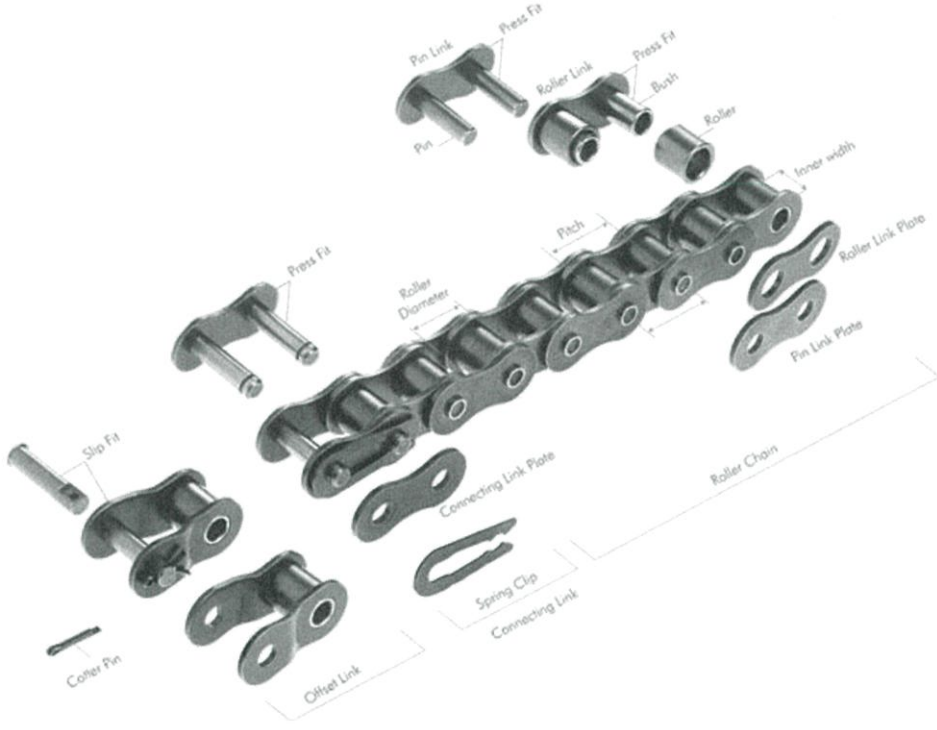
โซ่จะประกอบด้วยชุดของลูกกลิ้งทรงกระบอกสั้นที่ยึดเชื่อมโยงต่อเข้าด้วยกัน ดังรูปที่ 2.14 โซ่จะถูกขับเคลื่อนด้วยล้อฟันที่เรียกว่า สเตอร์ (Sprocket) มีขนาดร่องฟันเท่ากับซี่ฟันบนสเตอร์ และจะล็อคเข้าไปในซี่ฟันบนของสเตอร์ โซ่จะมีอัตราทดเหมือนกับชุดเฟืองขับ และอาจมีเสียงดังขณะทำงาน แต่สามารถใช้งานส่งกำลังสำหรับแรงบิดสูงๆ ได้ดี และมีประสิทธิภาพในการส่งกำลังงานสูง

ข้อดีของการส่งกำลังด้วยโซ่

1. ไม่ต้องใช้แรงดึงในช่วงเริ่มต้นของการหมุนส่งกำลังเหมือนกับสายพาน ทำให้แบริ่งที่ใช้จะมีอายุการใช้งานได้นานขึ้น
2. อัตราทดที่ได้แน่นอน ไม่มีการลื่นไถลเหมือนสายพาน
3. การติดตั้งประกอบเข้ากับฟันของจานเฟืองโซ่ทำได้ง่าย
4. สามารถใช้งานได้ในบริเวณที่มีความร้อนสูงและฝุ่นละอองมาก

ข้อเสียของการส่งกำลังด้วยโซ่

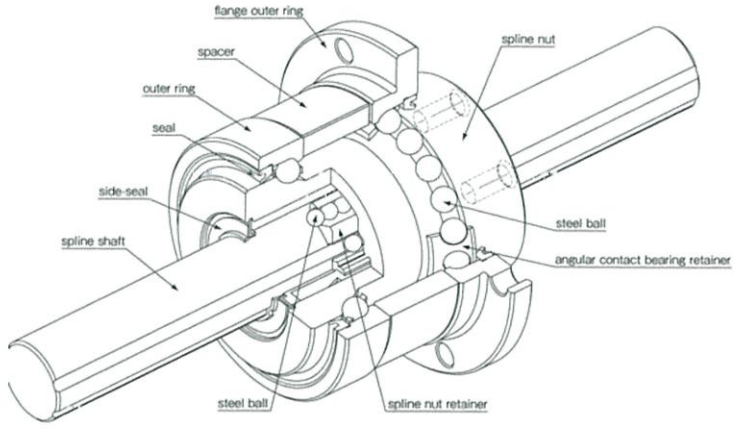
1. ขณะการหมุนส่งกำลังจะมีเสียงดัง
2. กรณีโซ่ขาดขณะทำงาน ถ้าหมุนด้วยความเร็วสูงจะเกิดอันตรายได้โดยง่าย
3. ไม่สามารถส่งกำลังบิดเหมือนสายพานได้
4. ไม่มีความยืดหยุ่นเหมือนสายพานขณะหมุนส่งกำลัง
5. เพลาของจานเฟืองโซ่ที่ใช้จะต้องขนานกันเสมอ



รูปที่ 2.14 โครงสร้างและส่วนประกอบของโซ่ส่งกำลัง
[http://tsubaki.eu/chain/introduction-roller-chain/]

2.3.1.3 <u>เพลา (Shaft)</u>

เพลาเป็นชิ้นส่วนที่มีลักษณะเป็นก้านทรงกระบอกที่หมุนได้ ดังรูปที่ 2.15 ใช้ในการส่งถ่ายกำลังจากอุปกรณ์ขับเร้า เช่น มอเตอร์ ไปยังส่วนที่เคลื่อนไหวของหุ่นยนต์ เพลาเป็นชิ้นส่วนที่สำคัญมากในหุ่นยนต์ที่เคลื่อนไหวได้ทุกชนิด นอกจากเพลาแล้ว ยังมี แกน (Axle) ซึ่งจะเป็นชิ้นส่วนลักษณะเดียวกันกับเพลาแต่ไม่สามารถหมุนได้ ส่วนใหญ่ทำหน้าที่รองรับชิ้นส่วนที่หมุน เช่น ล้อ เป็นต้น



รูปที่ 2.15 ลักษณะของเพลา

[http://www.directindustry.com/prod/nb/product-7383-792011.html]

2.3.1.4 ตลับลูกปืนและปลอกสวม (Bearing and Bush)

ตลับลูกปืนและปลอกสวม ต่างก็เป็นอุปกรณ์ที่ใช้รองรับจุดหมุน หรือจุดต่างๆที่เคลื่อนที่ของหุ่นยนต์ ทำหน้าที่ลดแรงเสียดทานที่เกิดจากการเคลื่อนไหวของอุปกรณ์

2.3.2 อุปกรณ์ขับ (Actuator)

อุปกรณ์ขับ คือ อุปกรณ์ที่สามารถเปลี่ยนแปลงพลังงานไฟฟ้าที่ป้อนเข้าให้กลายเป็นการกระจัดการเคลื่อนที่ หรือแรง ได้แก่ มอเตอร์ไฟฟ้า สเตปเปอร์มอเตอร์ เซอร์โวมอเตอร ระบบนิวแมติกส์ และระบบไฮโดรลิกส์

2.3.2.1 มอเตอร์สเต็ปปิ้ง (Stepping Motor)

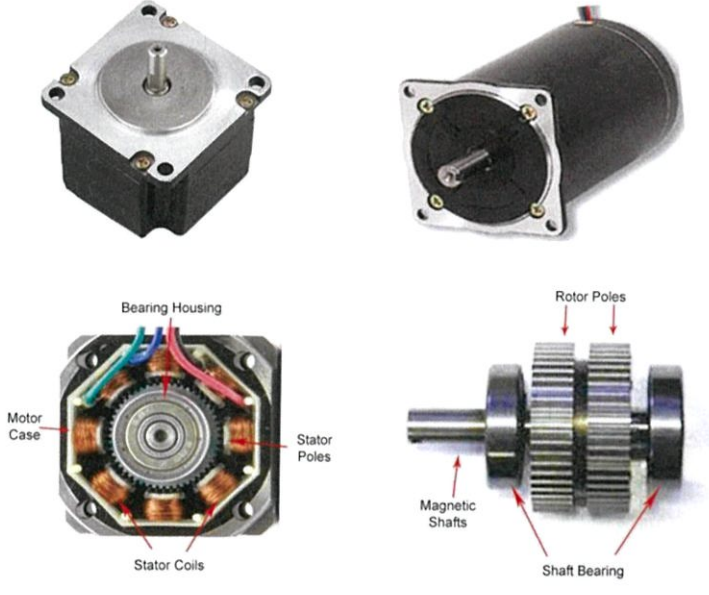
สเตปปิ้งมอเตอร์เป็นอุปกรณ์เอาต์พุตอย่างหนึ่ง ซึ่งสามารถนำไอซีไมโครคอนโทรลเลอร์ มาทำการควบคุมได้สะดวก และเป็นมอเตอร์ที่เหมาะสมสำหรับใช้ในงานควบคุมการหมุน ที่ต้องการตำแหน่งและทิศทางที่แน่นอน โดยทั่วไปแล้วมอเตอร์ไฟฟ้ากระแสตรง (DC Motor) หรือมอเตอร์ไฟฟ้ากระแสสลับ (AC Motor) จะมีการหมุนที่ต่อเนื่อง ไม่สามารถควบคุมการหมุนที่เป็นองศาหรือระยะทางได้ แต่สเต็ปปิ้งมอเตอร์สามารถควบคุมทิศทางการหมุนตามความต้องการได้ ลักษณะการทำงานของสเต็ปปิ้งมอเตอร์ จะต้องป้อนสัญญาณพัลส์ (Pulse) ให้กับขด สเตเตอร์ (Stator) ทำให้เกิดแรงผลักที่โรเตอร์ (Rotor) จึงเกิดการหมุน โดยปกติแล้วการหมุนของมอเตอร์จะหมุนเป็นจังหวะ (Step) ตามสัญญาณพัลส์ที่ป้อนเข้ามามอเตอร์แบบสเต็ปปิ้งมีลักษณะและโครงสร้าง ดังรูปที่ 2.16 การให้ Stepping Motor หมุนได้นั้น จะต้องมีการป้อนสัญญาณพัลส์ ที่ขดลวดแต่ละเฟสของ

สเต็ปปิ้งมอเตอร์ ให้ไปทิศทางเดียวกันหรือเรียงลำดับ (Sequence) และถ้าต้องการให้หมุนกลับทิศทางก็จะต้องป้อนสัญญาณพัลส์ ในทิศทางตรงกันข้ามการควบคุมการหมุนสามารถ แบ่งออกเป็น 3 ลักษณะ

1. แบบฟูลสเต็ปหนึ่งเฟส (Full-Step 1 Phase) การควบคุมลักษณะนี้จะเป็นการกระตุ้นขดลวดทีละขดเรียงตามลำดับ 1, 2, 3, 4 การกระตุ้นจะมีขดลวดขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น เช่น ขดที่ 1, 2, 3, 4 หรือถ้าต้องการให้หมุนสวนทิศทางกัน ก็จะกระตุ้นขดลวดที่ 4, 3, 2, 1 เป็นต้น การกระตุ้นแบบนี้จะทำงานง่ายที่สุด และกินกระแสไฟฟ้าน้อยที่สุด

2. แบบฟูลสเต็ปสองเฟส (Full-Step 2 Phase) การควบคุมลักษณะนี้จะเป็นการกระตุ้นขดลวดทีละ 2 ขด ที่อยู่ใกล้กันให้ทำงานพร้อมกัน และเรียงลำดับกันไป เช่น ขดที่ 12, 23, 34, 41 หรือถ้าต้องการให้หมุนสวนทิศทางกันก็จะกระตุ้นขดลวดที่ 14, 43, 32, 21 เป็นต้น ข้อดีของการกระตุ้นแบบนี้ คือ Stepping Motor จะมีแรงบิด (Torque) มากกว่าแบบ 1 เฟส ส่วนข้อเสีย คือ จะต้องใช้กำลังไฟฟ้าเพิ่มขึ้นเป็นสองเท้าของแบบ 1 เฟส

3. แบบฮาล์ฟสเต็ป (Half step) การควบคุมลักษณะนี้เป็นการกระตุ้นแบบ 1 เฟส กับ 2 เฟส มารวมกัน โดยการกระตุ้นจะเรียงลำดับ เช่น ขดที่ 1, 12, 2, 23, 3, 34, 4, 41 หรือถ้าต้องการให้หมุนสวนทิศทางกันก็จะกระตุ้น ขดลวดที่ 1, 14, 4, 43, 3, 32, 2, 21 เป็นต้น ข้อดีของการกระตุ้นแบบนี้ คือ Stepping Motor จะมีแรงบิดเพิ่มมากขึ้น ละเอียดขึ้น และการควบคุมตำแหน่งถูกต้องมากยิ่งขึ้น ส่วนข้อเสีย คือ จะต้องใช้กำลังไฟฟ้าเพิ่มเป็น 2 เท่า เช่นเดียวกันกับการกระตุ้นแบบ 2 เฟส



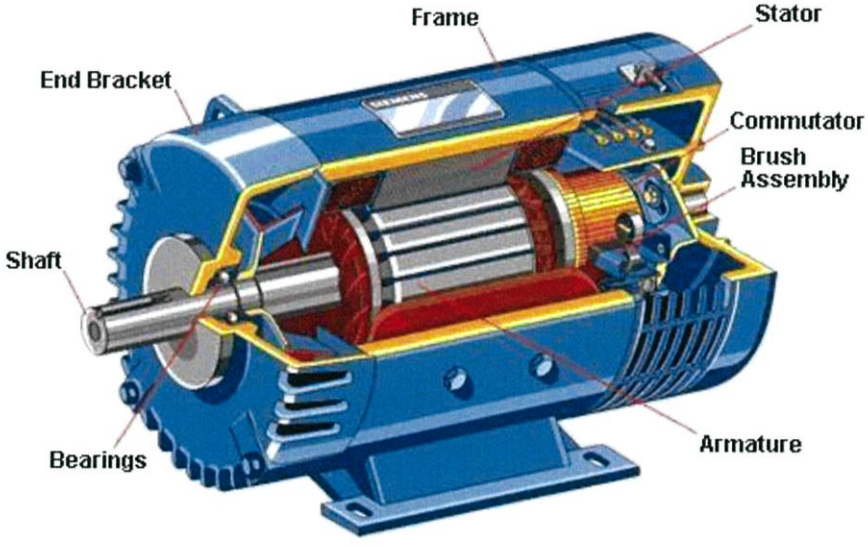
รูปที่ 2.16 โครงสร้างภายในมอเตอร์สเต็ปปิ้ง

[http://www.engineersgarage.com/articles/stepper-motors]

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.3.2.2 มอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

มอเตอร์ไฟฟ้ากระแสตรง (DC: Direct Current Motor) จะทำหน้าที่เปลี่ยนพลังงานจากพลังงานไฟฟ้าเป็นพลังงานกล โดยประกอบด้วย 2 ส่วน คือ ส่วนที่อยู่กับที่ (Stator) และส่วนที่เคลื่อนที่ (Rotor) ดังรูปที่ 2.17 แสดงโครงสร้างภายในมอเตอร์ไฟฟ้ากระแสตรง ลักษณะการหมุนของ DC Motor จะมีความแตกต่างกับ Stepping Motor คือ DC Motor จะหมุนอย่างต่อเนื่อง ถ้าหมุนกลับทิศทางก็จะหมุนอย่างต่อเนื่องเช่นกัน ซึ่งการเคลื่อนที่ขณะหมุนจะสม่ำเสมอแต่ไม่สามารถควบคุมตำแหน่งได้เหมือนกับ Stepping Motor ที่สามารถควบคุมตำแหน่งการหมุนได้



รูปที่ 2.17 โครงสร้างภายในมอเตอร์ไฟฟ้ากระแสตรง
[https://www.pinterest.com/pin/414401603183480028]

การขับและกลับทิศทางของมอเตอร์กระแสตรง

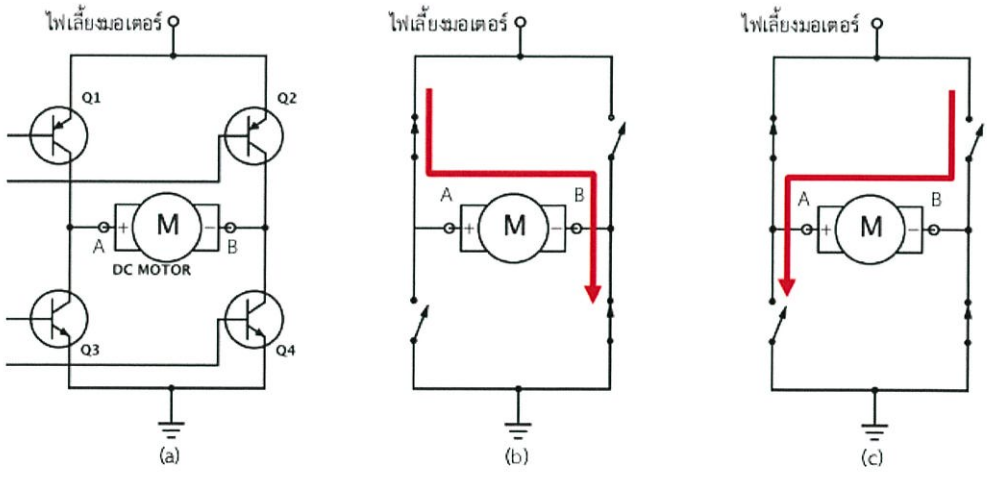
ชุดขับกระแสดีซีมอเตอร์ส่วนใหญ่จะเป็นแบบ H-Bridge ดังรูปที่ 2.18 ซึ่งวงจรประกอบด้วยทรานซิสเตอร์หรือมอตเฟส โดยทำหน้าที่เป็นสวิตช์เปิด-ปิด จำนวน 4 ชุด (Q1-Q4) โดยต่อกับ DC Motor ดังรูปที่ 2.19 (a) ซึ่งสามารถควบคุมทิศทางการไหลของกระแสได้ เมื่อส่งสัญญาณควบคุมให้ทรานซิสเตอร์ Q1 และ Q4 ทำงาน และปิดการทำงานของทราซิสเตอร์ Q2 และ Q3 กระแสจะไหลจากจุด A ไปจุด B ดังรูปที่ 2.19 (b) จึงทำให้มอเตอร์เริ่มหมุนเมื่อส่งสัญญาณควบคุมให้ทรานซิสเตอร์ Q2 และ Q3 ทำงาน และปิดการทำงานของทราซิสเตอร์ Q1 และ Q4 กระแสจะไหลจากจุด B ไปจุด A ดังรูปที่ 2.19 (c) เป็นผลให้มอเตอร์หมุนกลับทิศ

144432

การควบคุมความเร็วของมอเตอร์กระแสตรง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั้วไปเช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชั่นทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

รูปที่ 2.18 H-BRIDGE DRIVER DC MOTOR



รูปที่ 2.19 การทำงานของวงจร H-Bridge สำหรับควบคุมการหมุนของ DC Motor [http://aimagin.com/blog/motor]

2.3.3 อุปกรณ์ไฟฟ้า (Electronics)

อุปกรณ์ไฟฟ้า คือ อุปกรณ์ที่ใช้สัญญาณทางระบบไฟฟ้า ได้แก่ เซ็นเซอร์ เอ็นโค้ดเดอร์ อุปกรณ์แสดงผล และชุดขับมอเตอร์

2.3.3.1 <u>เซนเซอร์แสง</u>

เซนเซอร์ชนิดใช้แสง (Optical Sensor หรือ Photo Sensor) โดยทั่วไปใช้ในงานการตรวจจับการเคลื่อนไหว การตรวจจับวัตถุ และการตรวจสอบขนาดรูปร่างของวัตถุ เซนเซอร์ชนิดนี้ทำงานโดยอาศัยหลักการส่งและรับแสง มีส่วนประกอบสำคัญ 2 ส่วนคือ ตัวส่งแสง (Emitter) และตัวรับแสง (Receiver) ลักษณะการตรวจจับเกิดจากการที่ลำแสงจากตัวส่งแสง ส่งไปสะท้อนกับวัตถุหรือถูกขวางกั้นด้วยวัตถุ ส่งผลให้ตัวรับแสงรู้สภาวะที่เกิดขึ้นและเปลี่ยนแปลงสภาวะของสัญญาณทางด้านเอาต์พุตเพื่อนำไปใช้งานต่อไป

2.3.3.2 <u>เอ็นโคดเดอร์</u>

เอ็นโคดเดอร์ เป็นอุปกรณ์ตรวจรู้รูปแบบหนึ่ง ซึ่งมีความสำคัญมากใช้ในการวัดมุมเพลาของมอเตอร์ เอ็นโคดเดอร์ประกอบด้วย จานหมุน และอุปกรณ์ตรวจจับ โดยจานหมุนจะมีช่องเล็กๆ เมื่อเพลาของมอเตอร์หมุนจะทำให้จานหมุนไปตัดลำแสงของอุปกรณ์ตรวจจับ ทำให้ชุดรับแสงมีการรับสัญญาณเป็นช่วงๆ จึงทำให้สัญญาณที่ได้มีลักษณะเป็นพัลส์ ซึ่งสัญญาณพัลส์ที่ได้จะแปรผันตรงกับการหมุนของเพลามอเตอร์ ซึ่งมีอยู่ 2 ชนิด คือ 1. เอ็นโคดเดอร์ อินคริเมนต์ (Incremental Encoder) โดยทั่วไปเรียกว่า เอ็นโคดเดอร์แบบโรตารี (Rotary Encoder) เป็นเอ็นโคดเดอร์แสดงความเร็ว สัญญาณที่ได้จะเป็นสัญญาณแบบดิจิตอล (Digital) ง่ายต่อการแปรผล 2. เอ็นโคดเดอร์แบบสัมบูรณ์ (Absolute Encoder) หรือโดยทั่วไปเรียกว่าโพเทนชิโอมิเตอร์ (Potentiometer) โดยทั่วไปแล้วการทำงานคล้ายกับเอ็นโคดเดอร์แบบโรตารี แต่สัญญาณที่ได้จะเป็นเลขฐานสอง (Binary) การใช้งานจะยากกว่าเอ็นโคดเดอร์แบบโรตารี่ แต่เอ็นโคดเดอร์ชนิดนี้จะให้ความเที่ยงตรงและสามารถบอกได้ทุกตำแหน่งของการเคลื่อนที่

2.3.3.3 <u>แบตเตอรี่</u>

แบตเตอรี่โดยทั่วไปจะมีแหล่งจ่ายไฟอย่างน้อย 2 ชุด คือ ภาคจ่ายไฟให้กับภาคคอนโทรล และภาคจ่ายไฟให้กับภาคแมคคานิค

2.3.3.4 อุปกรณ์แสดงผล

อุปกรณ์แสดงผล คืออุปกรณ์ที่ใช้แสดงค่าสถานะต่างๆ ของหุ่นยนต์ให้มนุษย์ทราบ ซึ่งอุปกรณ์ที่ใช้แสดงผลของหุ่นยนต์มีอยู่ด้วยกันหลายรูปแบบ เช่น จอภาพ ใช้ในการบอกสถานะด้วยภาพ และลำโพง ใช้ในการบอกสถานะด้วยเสียง หรือแม้แต่กระทั่งหลอดไฟก็ใช้ในการบอกสถานะของหุ่นยนต์ได้เช่นกัน

2.3.4 อุปกรณ์ควบคุม (Controller)

อุปกรณ์ควบคุม เป็นหัวใจหลักของหุ่นยนต์ เพราะเป็นตัวประมวลผลและสั่งการ ให้หุ่นของเราสามารถคิด ตัดสินใจ ควบคุม รักษาสภาพตัวเอง และทำงานต่าง ๆ ตามที่ออกแบบไว้เองได้ โดยใช้การรับรู้จากภาคเซนเซอร์

ไมโครคอนโทลเลอร์

ในหุ่นยนต์อัตโนมัติสมัยใหม่จะใช้อุปกรณ์อิเล็กทรอนิกส์ที่เรียกว่า "ไมโครคอนโทรลเลอร์" (Microcontroller) เป็นอุปกรณ์หลักในการควบคุมและประมวลผล โดยในส่วนควบคุมนี้จะบรรจุโปรแกรมควบคุมที่ผู้ใช้งานเขียนขึ้นลงในหน่วยความจำ และสามารถเปลี่ยนแปลงแก้ไขได้ โดยส่วนควบคุมนี้จะมีส่วนต่ออุปกรณ์ภายนอก 2 ลักษณะคือ พอร์ตอินพุตและพอร์ตเอาต์พุต สำหรับพอร์ตอินพุตยังแบ่งออกเป็น 2 แบบคือ พอร์ตอินพุตดิจิตอล (Digital Input Port) และพอร์ตอินพุตอะนาลอก (Analog Input Port) โดยพอร์ตอินพุตดิจิตอลจะรับสัญญาณที่เป็นลอจิก "0" หรือ "1" แล้วส่งไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผล ส่วนพอร์ตอินพุตอะนาลอก จะรับแรงดันไฟฟ้าเข้ามาแล้วผ่านวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล (Analog to digital converter, ADC) เพื่อให้เป็นข้อมูลดิจิตอลก่อนส่งเข้าไปยัง ไมโครคอนโทรลเลอร์ โดยปกติพอร์ตอินพุตอะนาลอกจะสามารถรับแรงดันได้ไม่เกิน 5 โวลต์ ส่วนความละเอียดในการแปลง สัญญาณอยู่ระหว่าง 8-16 บิต แต่ในหุ่นยนต์ที่มีประสิทธิภาพดีควรใช้ตั้งแต่ 10 บิตขึ้นไป

โครงสร้างของไมโครคอนโทลเลอร์ แบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ

1. หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งหมด โดยนำข้อมูลจากอุปกรณ์รับข้อมูลมาทำงานประมวลผลข้อมูลตามคำสั่งของโปรแกรม และส่งผลลัพธ์ออกไปหน่วยแสดงผล

2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ 1. หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือ ข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง 2. หน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วๆ ไป

3. ส่วนติดต่อกับอุปกรณ์ภายนอกหรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณหรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียูหน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้น มีความเร็วในการประมวลผลสูงตามไปด้วย

2.4 ความรู้พื้นฐานกลศาสตร์ในการคำนวณแรงบิด กำลัง และประสิทธิภาพของมอเตอร์ไฟฟ้ากระแสตรง

แรงบิด (Torque) $T = F \times R$ (N.m)

งานที่ได้จากแรงใน 1 รอบการหมุน $W =$ แรง x ระยะทาง

$= F \times 2\pi r$ (Jule)

กำลังที่เกิดขึ้น $P_O = F \times 2\pi r \times N/60$ (Jule/s)

$= (F \times r)(2\pi N/60)$ (Jule/s)

เมื่อความเร็วเชิงมุม $\omega = 2\pi N/60$ (Rad/s)

ดังนั้น กำลังที่เกิดขึ้น $P_O = T \times \omega$ (Jule/s, W)

กำลังไฟฟ้า $P = V \times I$ (W)

ประสิทธิภาพ $\eta = (P_O / P_I) \times 100$

เมื่อ P_O = กำลังด้านเอาท์พุท

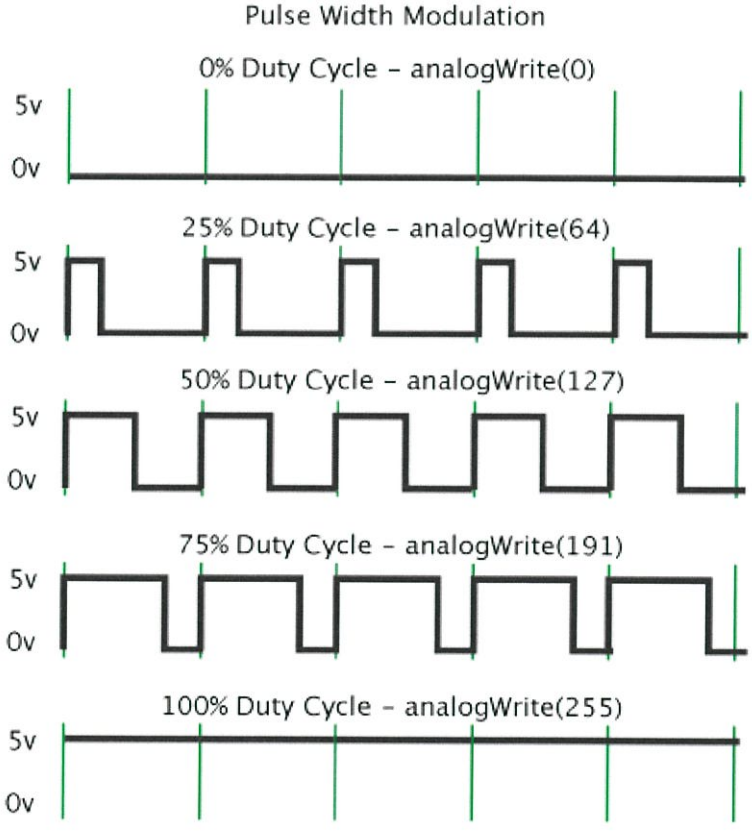
P_I = กำลังด้านอินพุท

2.5 หลักการสร้างสัญญาณ PWM

PWM ย่อมาจาก Pulse Width Modulation คือ เทคนิดการส่งสัญญาณแบบสวิตช์หรือส่งค่าดิจิตอล 0-1 โดยให้สัญญาณความถี่คงที่ การควบคุมระยะเวลาสัญญาณสูงและสัญญาณต่ำที่ต่างกันจะทำให้ค่าแรงดันเฉลี่ยของสัญญาณสวิตช์ต่างกันด้วย ดังตัวอย่างรูปที่ 2.20

- ถ้าอยากให้ V_{out} ออกมา 0 โวลต์ ก็ให้ระยะเวลาสัญญาณ LOW เต็มคาบเวลา
- ถ้าอยากให้ V_{out} ออกมา 25% เราก็ให้สัญญาณ HIGH 25% และ สัญญาณ LOW 75%
- ถ้าอยากให้ V_{out} ออกมา 50% เราให้ระยะเวลาของสัญญาณ HIGH และ LOW เท่ากัน ก็จะได้ค่าเป็น 2.5 โวลต์ เช่นเดียวกัน ถ้าเราอยากให้ V_{out} ออกมา 75% เราให้ระยะสัญญาณ HIGH เป็น 75% และ สัญญาณ LOW 25% ซึ่งสุดท้าย ถ้าเราอยากให้ V_{out} ออกมา 5 โวลต์ เราเปิดสัญญาณ HIGH เต็มคาบเวลา

สำหรับโมดุล PWM ของ Arduino มีความละเอียด 8 bit หรือ ปรับได้ 255 ระดับ ดังนั้นค่าสัญญาณ 0 โวลต์ถึง 5 โวลต์ จะถูกแสดงเป็นสัญญาณแบบดิจิตอล จะได้ 0 ถึง 255 ซึ่งเราสามารถเทียบสัดส่วนคำนวนจากเลขจริง เป็นเลขทางดิจิตอลได้



รูปที่ 2.20 ตัวอย่างการควบคุมระยะสัญญาณสูง-ต่ำที่ต่างกัน

[http://www.ayarafun.com/2015/05/dimmer-led-funbasic-io/]

2.6 การประมวลผลภาพ (Image processing)

การมองรูปภาพในแบบของคอมพิวเตอร์ คือ การที่จุดสีหลายๆ จุดมาเรียงต่อๆ กัน จนสามารถบอกได้ว่าเรียงแล้วออกมาเป็นรูปอะไร

2.6.1 การดึงลักษณะเด่นของภาพ (Feature Extraction)

การดึงลักษณะเด่นของภาพ เป็นการดึงเอาลักษณะเฉพาะของแต่ละรูปภาพออกมาเป็นเวกเตอร์เพื่อนำไปใช้ในการฝึกฝนระบบและทดสอบระบบ บางครั้งการใช้สีเพื่อทำการแยกแยะวัตถุนั้น มักไม่มีความทนทานเพียงพอต่อการเปลี่ยนแปลงที่เกิดขึ้น เช่น ความเปลี่ยนแปลงของสภาพแสง มุมมองของภาพและการบดบังจากวัตถุอื่นๆ ก็สามารถทำให้การแยกแยะวัตถุโดยใช้สีผิดพลาดได้ การดึงจุดเด่นในภาพออกมา และทำการ Matching กับจุดเด่นของภาพในฐานข้อมูลที่ได้กำหนดไว้ จุดเด่นของภาพนั้นสามารถกำหนดได้จากหลายวิธีด้วยกัน เช่น การหาขอบภาพ หามุมในภาพจากความต่างของค่าสีในพิกเซลที่พิจารณาและพิกเซลรอบข้าง เมื่อทำการแปลงภาพให้แสดงค่าจุดเด่นของภาพ เช่น edge value, corner value แล้ว ค่าจุดเด่นนั้นจะนำมาเทียบกับค่าจุดเด่นของภาพตัวอย่างที่เก็บไว้ในฐานข้อมูล เพื่อหาค่าความเหมือนซึ่งจะได้นำไปใช้ในการแยกแยะต่อไป

2.6.2 การรู้จักรูปแบบ (Pattern Recognition)

เป็นการให้คอมพิวเตอร์แยกแยะสิ่งต่างๆ ยกตัวอย่างง่ายๆ เช่น แยกแยะได้ว่าสิ่งไหนคือ วงกลม สี่เหลี่ยมจัตุรัส หรือสี่เหลี่ยมผืนผ้า ก็นับเป็นสี่เหลี่ยมเหมือนกัน จากตัวอย่างข้างต้น จะเห็นได้ว่าสิ่งต่างๆ ที่จะนำมาแยกแยะนั้น ต้องมีลักษณะเฉพาะที่บ่งบอกความเป็นตัวมันเอง และอาจจะมีลักษณะปลีกย่อยต่างกันได้ วิธีการในการทำการรู้จักรูปแบบ คือ พยายามสอนให้คอมพิวเตอร์ รู้ว่าลักษณะสำคัญนั้นคืออะไร เมื่อมีข้อมูลใหม่เข้ามา ก็จะพิจารณาจากสิ่งที่เรียนรู้มา แล้วตัดสินใจได้ ซึ่งการรู้จักรูปแบบ (Pattern Recognition) มีอยู่ด้วยกัน 3 วิธีคือ

1. การรู้จักรูปแบบเชิงสเปกตรัม (Spectral Pattern Recognition) เป็นกลุ่มของจุดภาพที่อาศัยค่าความสว่างหรือค่าการสะท้อนของวัตถุในการ

2. จำแนกการรู้จักรูปแบบเชิงพื้นที่ (Spatial Pattern Recognition) เป็นกระบวนการจำแนกจุดภาพโดยอาศัยความสัมพันธ์เชิงพื้นที่ของจุดภาพนั้นๆ กับจุดภาพใกล้เคียง เช่น ความหยาบละเอียดของภาพขนาด รูปร่างของวัตถุในภาพ

3. การรู้จักรูปแบบเชิงกาลเวลา (Temporal Pattern Recognition) เป็นกระบวนการที่ใช้เวลาในช่วงต่างๆ เป็นตัวช่วยในการจำแนก โดยใช้ข้อมูลจากภาพที่ต่างวันและเวลา

2.7 ความรู้เบื้องต้นเกี่ยวกับ Arduino

[เอกชัย มะการ, เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทลเลอร์ตระกูล AVR ด้วย Arduino]

Arduino เป็นโครงการพัฒนาระบบไมโครคอนโทลเลอร์ (MCU) ตระกูล AVR ด้วยภาษา C++ แบบ Open Source ปัจจุบัน Arduino มีราคาถูกมากและออกมาหลายรุ่นซึ่งแต่ละรุ่นนั้นก็เหมาะกับงานที่แตกต่างกัน สามารถเขียนโปรแกรมจากคอมพิวเตอร์ที่ใช้งานอยู่ ไม่ว่าจะเป็น Linux, OSx หรือ MS windows และภาษาที่ใช้ก็คือภาษา C++ ง่ายต่อการเรียนรู้ เมื่อเขียนเสร็จแล้วก็อัพโหลดผ่าน COM Port หรือ USB Port ของคอมพิวเตอร์ได้ทันที

2.7.1 จุดเด่นของ Arduino

1. ราคาไมแพง เนื่องจากมี Source Code และวงจรแจกใหฟรีสามารถตอวงจรขึ้นมาใชงานไดเอง

2. โปรแกรมที่ใชพัฒนาของ Arduino รองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX

3. มีรูปแบบคำสั่งที่งายตอการใชงาน แตสามารถนำไปใชงานจริงๆที่มีความซับซอนมากๆได และยังสามารถสรางคำสั่งและ Library ใหมๆขึ้นมาใชงานเองไดเมื่อมีความชำนาญมากขึ้นแลว

4. มีการเปดเผยวงจรและ Source Code ทั้งหมดทำใหสามารถนำไปพัฒนาตอยอดเพิ่มเติมไดตามความตองการทั้ง Hardware และ Software

2.7.2 โครงสรางการเขียนโปรแกรม ภาษาซีของ Arduino

บอร์ด Arduino ที่มีอยู่หลายแบบ จะใช้โปรแกรมตัวเดียวกันในการเขียนชุดคำสั่ง โดยโปรแกรมมีชื่อเป็นทางการว่า Arduino IDE (IDE ย่อมาจาก Integrated Developemnt Environment) ซึ่งใช้งานได้ทั้งบนระบบปฏิบัติการ Window (XP Vista 7 8) ทั้ง 32 และ 64 บิต, Mac OS X และ Linux เรียกได้ว่าใช้งานได้กับทุกระบบปฏิบัติการ และเป็นอิสระจากการทำงานของ OS ทุกชนิด ทำให้ไม่ต้องมีการ Install โปรแกรม ให้วุ่นวายแบบโปรแกรมอื่น เพียงแค่ Download ตัวโปรแกรมมา จากนั้น Unzip ไว้ใน Directory ที่ต้องการ ซึ่งโปรแกรม Arduino IDE นี้เป็นโปรแกรมที่แจกฟรี

ภาษาซีของ Arduino จะจัดแบงรูปแบบโครงสรางของการเขียนโปรแกรมออกเปนสวนยอยๆ หลายๆ สวน โดยเรียกแตละสวนวาฟงกชั่น และเมื่อนำฟงกชัน มารวมเขาดวยกัน ก็จะเรียกวาโปรแกรม โดยโครงสรางการเขียนโปรแกรมของ Arduino นั้น ทุกๆ โปรแกรมจะตองประกอบไปดวยฟงกชันจำนวนเทาใดก็ไดแตอยางนอยที่สุดตองมีฟงกชันจำนวน 2 ฟงกชั่น คือ setup() และ loop() ดังตัวอยางรูปที่ 2.21

```
#include <Servo.h>          // สั่งผนวกไฟล์ ชื่อ Servo.h เข้ามาใช้ในโปรแกรม
int Servo1 = 9;             // กำหนดให้ Servo1 แทน Pin Digital-9
Servo myservo;              // สร้าง object ชื่อ myservo เพื่อควบคุม Servo

void setup()
{
 myservo.attach(Servo1);    // กำหนดให้ใช้ขา Digital-9 สร้างสัญญาณควบคุม Servo
}

void loop()
{
 myservo.write(180);        // กำหนดค่าตำแหน่งให้กับ Servo = 180 องศา
}
```

รูปที่ 2.21 ตัวอย่างฟังก์ชัน setup() และ loop()

[http://www.ett.co.th/product2009/BOOKS/sample-book-AVR-Arduino.pdf]

โครงสรางพื้นฐานของภาษาซีที่ใชกับ Arduino นั้น จะประกอบไปดวย 3 สวนใหญ่ๆ ด้วยกัน คือ

1. Header ในสวนนี้จะมหรือไมมีก็ได ถ้ามีต้องกำหนดไวในสวนเริ่มตนของโปรแกรม ซึ่งสวนของ Header ไดแก สวนที่เปน Compiler Directive ตางๆ รวมไปถึงสวนของการประกาศตัวแปรและคาคงที่ต่างๆ ที่จะใชในโปรแกรม

2. setup() ในสวนนี้เปนฟงก์ชั่นบังคับที่ตองกำหนดใหม่ในทุกๆ โปรแกรม ถึงแม้ว่าในบางโปรแกรมจะไมตองการใช้งานก็ย้งจำเป็นต้องประกาศไวดวยเสมอ เพียงแตไมตองเขียนคำสั่งใดๆ ไวในระหว่างวงเล็บปีกกา { } ที่ใชเป็นตัวกำหนดขอบเขตของฟงก์ชั่น โดยฟังก์ชั่นนี้จะใชสำหรับบรรจุคำสั่งในสวนที่ตองการใหโปรแกรมทางานเพียงรอบเดียวตอนเริ่มต้นทำงานของโปรแกรมครั้งแรกเทานั้น ซึ่งไดแกคำสั่งเกี่ยวกับการ Setup คาการทำงานตางๆ เชน การกำหนดหน้าที่การใชงานของ PinMode และการกำหนดคา Baudrate สำหรับใชงานพอรตสื่อสารอนุกรม เปนตน

3. loop() เปนสวนฟงกชั่นบังคับที่ตองกำหนดใหมีในทุกๆ โปรแกรมเชนเดียวกันกับฟงก์ชั่น setup( ) โดยฟงกชั่น loop( ) นี้จะใช้บรรจุคำสั่งที่ตองการใหโปรแกรมทำงานเป็นวงรอบซ้ำๆ กันไปไมรูจบซึ่งถาเปรียบเทียบกับรูปแบบของ ANSI-C สวนนี้ก็คือ ฟงกชั่น main( ) นั่นเอง

บทที่ 3
วิธีการดำเนินงาน

สำหรับส่วนของการดำเนินงาน การออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่นั้น มีแผนการดำเนินงานในเรื่องของวัสดุอุปกรณ์ และเครื่องมือ ขั้นตอนการออกแบบและสร้าง รวมไปถึงการทดสอบ โดยแบ่งออกเป็นสองส่วน คือส่วนของฮาร์ดแวร์ และส่วนของซอฟต์แวร์ ดังนี้

3.1 การออกแบบ

ส่วนของการออกแบบฐานหุ่นยนต์เคลื่อนที่ มีการออกแบบการทำงานออกเป็นสองส่วน คือ การออกแบบทางด้านฮาร์ดแวร์ และการออกแบบทางด้านซอฟต์แวร์

3.1.1 ขั้นตอนการออกแบบฮาร์ดแวร์

1. กำหนดวัตถุประสงค์ในการสร้าง
   - ในโปรเจคนี้เป็นการสร้างหุ่นยนต์ที่ใช้ในการจัดการงานด้านการลำเลียงวัสดุและเป็นฐานหุ่นยนต์เคลื่อนที่ให้กับหุ่นยนต์ประเภทแขนกล
2. กำหนดขอบเขตของระบบและความต้องการของระบบ
   - บรรทุกของได้ไม่เกิน 17 กิโลกรัม และเคลื่อนที่บนทางลาดชันไม่เกิน 30 องศาได้
   - เคลื่อนที่ได้ทั้งพื้นผิวเรียบและขรุขระ
   - มีแหล่งพลังงานในตัว ใช้งานต่อเนื่องได้อย่างน้อย 30 นาที
   - ควบคุมการเคลื่อนที่ได้ด้วยระบบไร้สาย
3. ออกแบบระบบที่สามารถตอบสนองความต้องการแต่ละข้อนั้นได้โดยคร่าวๆ
   - เคลื่อนที่ได้ทั้งพื้นเรียบและขรุขระจึงเลือกใช้ล้อแบบตีนตะขาบ
   - หุ่นยนต์ต้องเคลื่อนที่จะมีสายส่งพลังงานลากไปไม่ได้ ต้องใช้แหล่งพลังงานในตัวเอง จึงเลือกใช้พลังงานจากแบตเตอรี่

4. เลือกวัสดุอุปกรณ์

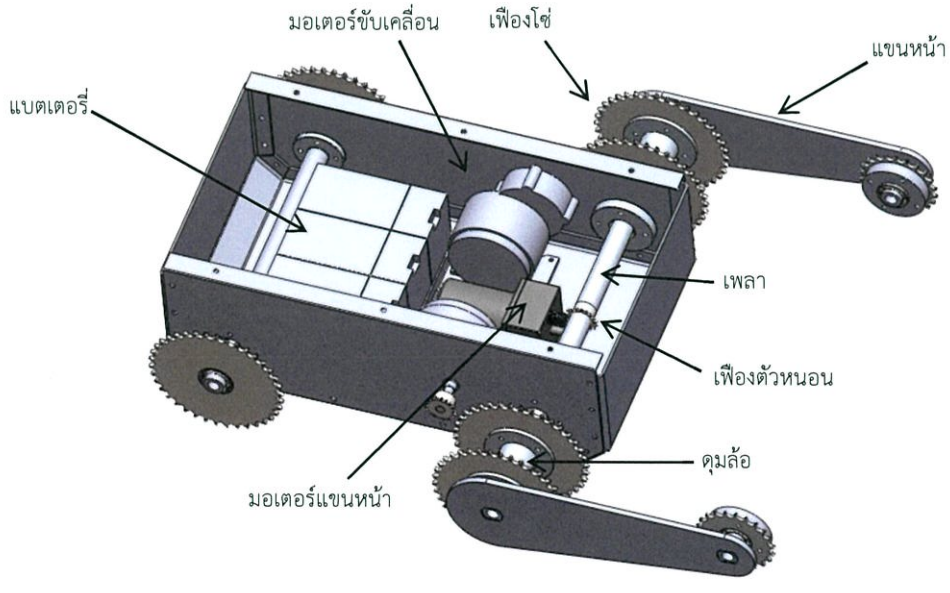
การเลือกหาวัสดุอุปกรณ์ เริ่มจากศึกษาถึงชิ้นส่วนที่มีจำหน่ายในท้องตลาด (ชิ้นส่วนมาตรฐาน) และชิ้นส่วนที่ต้องทำขึ้นมาเอง โดยพิจารณาในส่วนของที่ทำเองไม่ได้และต้องซื้อก่อน แล้วค่อยออกแบบส่วนอื่น ถ้าหาชิ้นส่วนตามแบบที่ต้องการไม่ได้ ก็ต้องลงมือดำเนินการผลิตชิ้นส่วนนั้นขึ้นมาเอง พิจารณาถึงเครื่องมืออุปกรณ์และความสามารถของผู้จัดทำที่จะดำเนินการผลิตชิ้นส่วนที่ผลิตเองได้

3.1.2 การออกแบบรายละเอียดของฐานหุ่นยนต์เคลื่อนที่

การออกแบบรายละเอียดของฐานหุ่นยนต์เคลื่อนที่ เป็นการออกแบบในส่วนของระบบขับเคลื่อน ลักษณะของตัวโครงหุ่นยนต์ และการออกแบบส่วนอิเล็กทรอนิกส์และควบคุมการทำงาน

3.1.2.1 <u>การออกแบบโครงสร้างของฐานหุ่นยนต์เคลื่อนที่</u>

การออกแบบโครงสร้างของหุ่นยนต์นี้จะใช้โปรแกรม Solidworks ในการออกแบบชิ้นส่วนทั้งหมดและในส่วนของระบบขับเคลื่อนและส่งกำลัง จะออกแบบโดยใช้มอเตอร์ทั้งหมด 3 ตัว ในการขับเคลื่อนและควบคุมทิศทางการเคลื่อนที่ โดยติดตั้งที่ล้อด้านซ้าย-ขวา และแขนหน้าในการบังคับให้ยกขึ้น-ลง โดยในส่วนนี้มีส่วนประกอบที่สำคัญ ดังรูปที่ 3.1 โดยแบบรายละเอียดชิ้นส่วนทั้งหมดสามารถดูได้ใน ภาคผนวก ข



รูปที่ 3.1 แบบร่างของหุ่นยนต์เคลื่อนที่ในส่วนของระบบขับเคลื่อนและส่งกำลัง ออกแบบโดยโปรแกรม Solidworks

3.1.2.2 การออกแบบส่วนอิเล็กทรอนิกส์และส่วนควบคุมการทำงาน

ชิ้นส่วนและอุปกรณ์อิเล็กทรอนิกส์ที่ใช้ในการประกอบหุ่นยนต์นี้ จะถูกติดตั้งไว้ในตัวโครงของหุ่นยนต์ทั้งหมด ซึ่งประกอบด้วย

1. มอเตอร์ไฟฟ้ากระแสตรง รุ่น MY1016Z2 DC 24V 250W ดังรูปที่ 3.2 จำนวน 2 ตัว ในการขับเคลื่อนล้อด้านซ้าย-ขวา ของหุ่นยนต์ และใช้มอเตอร์ รุ่น ZGB70R DC 12V 36W ดังรูปที่ 3.3 ในการบังคับแขนหน้าของหุ่นยนต์ขึ้น-ลง

2. แบตเตอรี่ LION ® HGL7.3-12 12V7.3Ah/20HR (MP7.2A-12V) ดังรูปที่ 3.4 จำนวน 3 ก้อน โดยต่ออนุกรม 2 ก้อนเพื่อใช้กับมอเตอร์ซ้าย-ขวา และอีก 1 ก้อนต่อตรงกับวงจรอิเล็กทรอนิกส์และมอเตอร์ยกแขนหน้า

3. Single Board Computer (SBC) ใช้สำหรับการควบคุมขนาดเล็กที่สามารถบรรจุลงไปในตัวหุ่นยนต์ได้ ดังรูปที่ 3.5

4. ไมโครคอนโทลเลอร์ (Microcontroller) ในที่นี้ใช้บอร์ด Arduino ดังรูปที่ 3.6 ในการเขียนโปรแกรมจะใช้โปรแกรม Arduino IDE ซึ่งใช้ภาษา C++ ในการเขียนชุดคำสั่งควบคุมการทำงานของหุ่นยนต์ การทำงานของไมโครคอนโทลเลอร์จะเป็นการสั่งงานมอเตอร์ที่ต่อกับล้อทั้งสองด้านซ้าย-ขาว และแขนหน้าของตัวหุ่นยนต์เพื่อให้เคลื่อนไหวไปในทิศทางที่ต้องการ

5. H-bridge Driver DC Motor ดังรูปที่ 3.7 เป็นโมดูลที่ใช้ในการควบคุมความเร็วและทิศทางของมอเตอร์

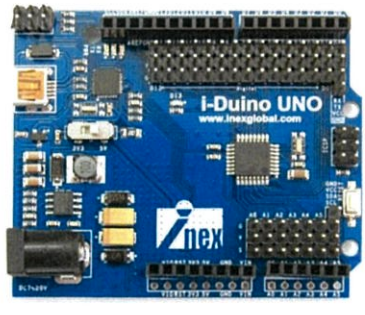
6. เอ็นโค้ดเดอร์ (encoder) ใช้ในการตรวจวัดความเร็วรอบ ทิศทางการหมุนมุมการหมุน และจำนวนรอบการหมุนมอเตอร์ เพื่อใช้ในการควบคุมระยะทางและการเคลื่อนที่ของหุ่นยนต์

รูปที่ 3.2 มอเตอร์รุ่น MY1016Z2 DC 24V 250W

รูปที่ 3.3 มอเตอร์รุ่น ZGX38RH DC 12V

รูปที่ 3.4 แบตเตอรี่ LION ® HGL7.3-12 12V7.3Ah/20HR (MP7.2A-12V)

รูปที่ 3.5 Single Board Computer

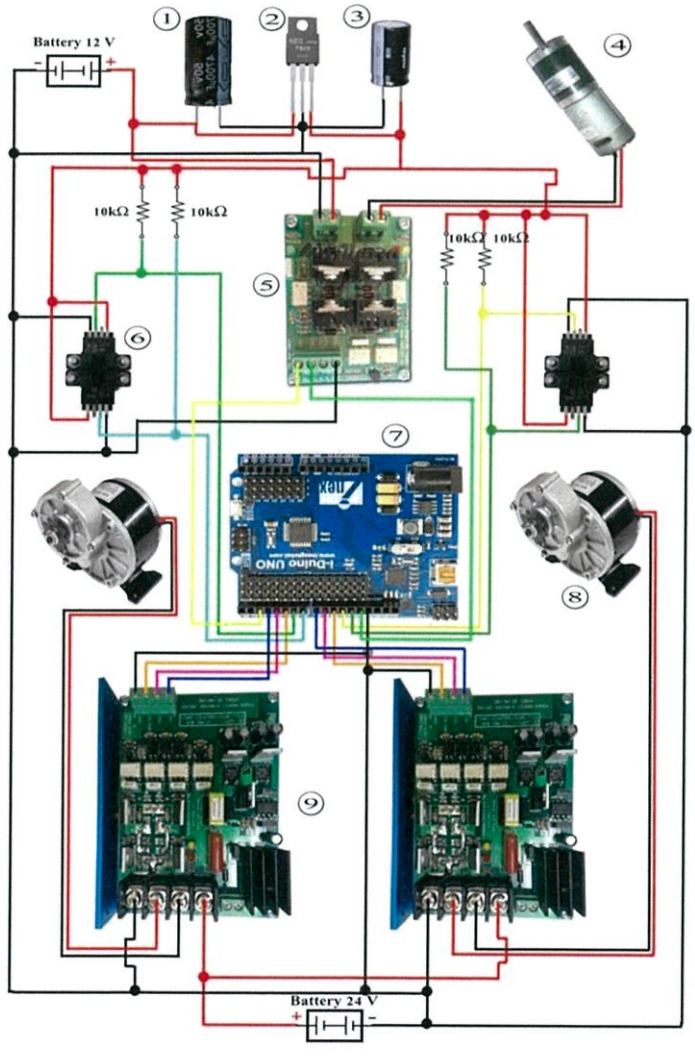


รูปที่ 3.6 บอร์ด Arduino ใช้เป็นไมโครคอนโทลเลอร์ในการควบคุมมอเตอร์

รูปที่ 3.7 H-BRIDGE DRIVER DC MOTOR

3.1.2.3 <u>การออกแบบวงจรควบคุมการทำงาน</u>

ออกแบบวงจรควบคุมการทำงานเพื่อใช้เป็นตัวกลาง ในการควบคุมอุปกรณ์ ไฟฟ้าให้สามารถทำงานได้ตามคำสั่งของโปรแกรม โดยมีการต่ออุปกรณ์ดังนี้



รูปที่ 3.8 การต่อมอเตอร์กระแสตรงกับเอ็นโค้ดเดอร์ในการบังคับทิศทางการเคลื่อนที่ของหุ่นยนต์

หมายเลข 1 คือ ตัวเก็บประจุ 50V 4700μF

หมายเลข 2 คือ Mosfet N-Channel 60V/105A, 110W Rds(on) 4.4mΩ Typ.

หมายเลข 3 คือ ตัวเก็บประจุ

หมายเลข 4 คือ มอเตอร์ยกแขนหน้า รุ่น ZGX38RH DC 12V

หมายเลข 5 คือ H-Bridge driver DC motor รุ่น SE-HB40-1

หมายเลข 6 คือ เอ็นโค้ดเดอร์เซนเซอร์

หมายเลข 7 คือ บอร์ด Arduino

หมายเลข 8 คือ มอเตอร์ขับเคลื่อน รุ่น MY1016ZZ DC 24V 250W

หมายเลข 9 คือ H-Bridge driver DC motor รุ่น SE-HB-100

3.1.3 การออกแบบซอฟต์แวร์

ในการออกแบบซอฟต์แวร์ คือการออกแบบให้หุ่นยนต์มีหลักการทำงานของระบบที่สามารถใช้งานได้ 2 รูปแบบ ได้แก่ การใช้งานแบบแมนนวลจากผู้ใช้งานโดยกดปุ่มบนหน้าอินเตอร์เฟส แล้วให้หุ่นยนต์เคลื่อนที่ได้ตามคำสั่ง และอีกรูปแบบคือตั้งค่าให้หุ่นยนต์เคลื่อนที่ตามวัตถุได้ด้วยตัวเอง โดยวางวัตถุไว้บนเส้นทางที่ต้องการให้หุ่นยนต์เคลื่อนที่ตามเส้นทางนั้น

3.2 ขั้นตอนการดำเนินการสร้างและประกอบส่วนของระบบส่งกำลัง

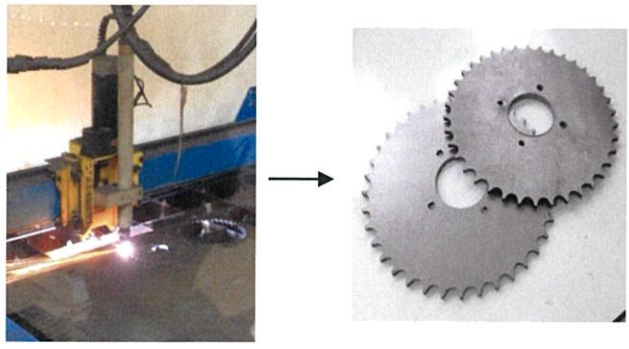
ในขั้นตอนการสร้างและประกอบส่วนของระบบส่งกำลัง ประกอบด้วยการทำเฟืองขับโซ่ ดุมล้อ ตุ๊กตาลูกปืน เพลา ชุดเฟืองตัวหนอน ตัวโครง และล้อตีนตะขาบ เพื่อนำมาประกอบกับมอเตอร์และชิ้นส่วนอิเล็กทรอนิกส์อื่นๆ ต่อไป

3.2.1 เฟืองขับโซ่

ในส่วนของเฟืองขับโซ่ที่ใช้ในการสร้างหุ่นยนต์นี้ จะใช้เฟืองด้วยกัน 3 ขนาดดังนี้

- เฟืองขับโซ่เบอร์ 40 เส้นผ่านศูนย์กลางพิต 154 มิลลิเมตร
- เฟืองขับโซ่เบอร์ 40 เส้นผ่านศูนย์กลางพิต 89 มิลลิเมตร
- เฟืองขับโซ่เบอร์ 25 เส้นผ่านศูนย์กลางพิต 97 มิลลิเมตร ใช้เพื่อรับการส่งกำลังจากมอเตอร์

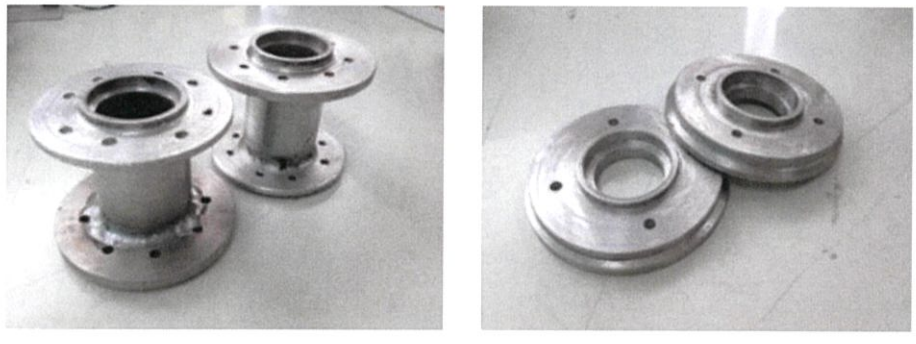
เฟืองขับโซ่เบอร์ 40 ใช้แผ่นเหล็ก ความหนา 4 มิลลิเมตร ตัดด้วยเครื่องพลาสม่า ดังรูปที่ 3.9



รูปที่ 3.9 การตัดเฟืองขับโซ่ด้วยเครื่องพลาสม่า

3.2.2 ดุมล้อ

ในส่วนของดุมล้อหน้าจะทำจากแท่งอลูมิเนียมสองขนาด คือ แท่งอลูมิเนียมตันขนาด เส้นผ่านศูนย์กลาง 89 มิลลิเมตรและแท่งอลูมิเนียมกลวงเส้นผ่านศูนย์กลาง 44 มิลลิเมตร โดยนำอลูมมิเนียมตันไปเจาะรูให้ได้ขนาดเส้นผ่านศูนย์กลาง 44 มิลลิเมตร เพื่อให้ประกอบกับแท่งอลูมิเนียมกลวงได้พอดี หลังจากนั้นกลึงรูอลูมิเนียมกลวงให้สามารถใส่ลูกปืนลงไปได้พอดี แล้วจึงนำไปตัดโดยใช้เครื่องเลื่อย และนำทั้งสองส่วนมาประกอบเข้าด้วยกันโดยการเชื่อม ดังรูปที่ 3.10 ส่วนดุมล้อหลังใช้เพียงอลูมิเนียมตันเจาะรูเท่านั้น ดังรูปที่ 3.11



รูปที่ 3.10 ลักษณะของดุมล้อหน้า

รูปที่ 3.11 ลักษณะของดุมล้อหลัง

3.2.3 ตุ๊กตาลูกปืน

ตุ๊กตาลูกปืน เป็นตัวยึดให้เพลาอยู่ในระดับที่ต้องการและรองรับการหมุนของเพลาทำจากอลูมิเนียมขนาดเส้นผ่านศูนย์กลางภายนอก 75 มิลลิเมตร เส้นผ่านศูนย์กลางภายในพอดีกับขนาดของเพลาคือ 25 มิลลิเมตร และทำรูให้สามารถใส่แบริ่งได้พอดี ใช้จำนวน 4 อัน ดังรูปที่ 3.12

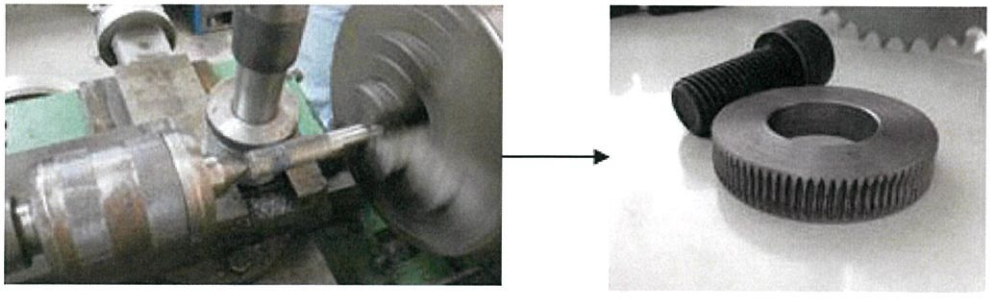
รูปที่ 3.12 ลักษณะของตุ๊กตาลูกปืน

3.2.4 เพลา

เพลาหน้า-หลัง ทำจากแท่งอลูมิเนียมกลวง ขนาดเส้นผ่านศูนย์กลางด้านนอก 25 มิลลิเมตร นำไปเข้าเครื่องกลึงเพื่อปลอกผิว ซึ่งเพลาหน้ามีความยาว 560 มิลลิเมตร ส่วนเพลาหลังมีความยาว 420 มิลลิเมตร

3.2.5 เฟืองตัวหนอนและตัวหนอน

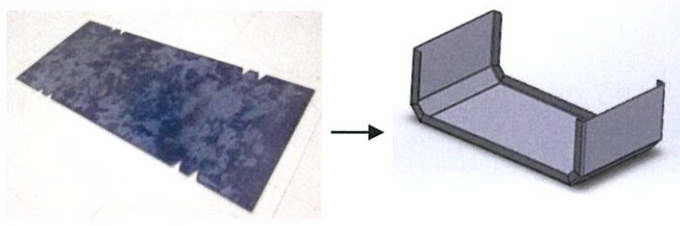
ชุดเฟืองตัวหนอน ทำจากแท่งเหล็กขนาดเส้นผ่านศูนย์กลางพิต 49.65 มิลลิเมตร ทำร่องฟันด้วยดอกต๊าปเกลียว ดังรูปที่ 3.13 สวมเข้าไปที่เพลาหน้าเพื่อใช้สำหรับเป็นตัวยกชุดแขนหน้าขึ้น-ลง



รูปที่ 3.13 ลักษณะของชุดเฟืองตัวหนอน

3.2.6 ตัวโครง

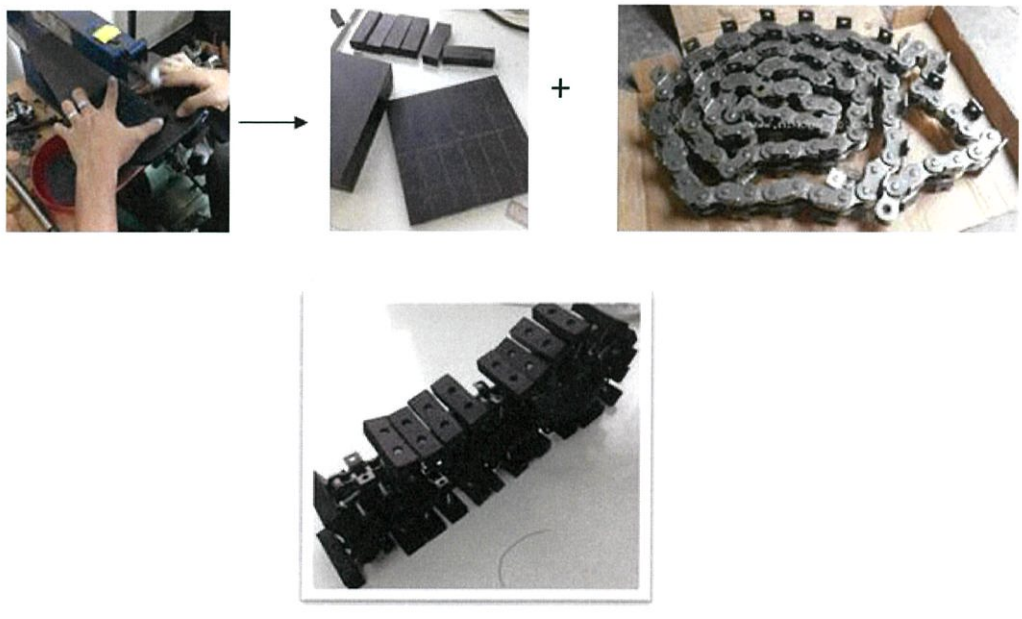
ทำจากแผ่นเหล็ก หนา 2 มิลลิเมตร ขึ้นรูปโดยใช้การตัดพลาสมา และนำมาพับตามแบบ ดังรูปที่ 3.14 พื้นที่ภายในเพียงพอสำหรับติดตั้งอุปกรณ์ต่างๆ ที่ใช้ในการเคลื่อนที่



รูปที่ 3.14 ตัวโครงที่ได้จากการตัดพลาสมาและพับขึ้นรูป

3.2.7 ล้อตีนตะขาบ

ใช้ล้อในลักษณะของตีนตะขาบเพื่อให้เคลื่อนที่ได้ดีทั้งในทางราบ ทางขรุขระ และทางลาดชัน โดยใช้โซ่ปีกเบอร์ 40 และตัดยางขนาด 60x20x10 มิลลิเมตร ติดในแต่ละปีกของโซ่ เพื่อให้ มีลักษณะเป็นตีนตะขาบโดยยางจะช่วยให้ล้อมีแรงยึดติดกับพื้นได้ดีเมื่อขึ้นทางชัน ดังรูปที่ 3.15



รูปที่ 3.15 ลักษณะของล้อตีนตะขาบ

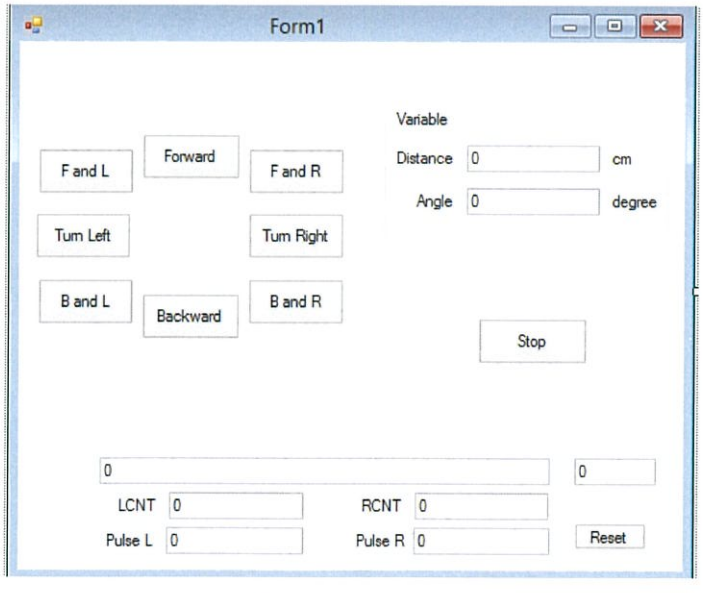
3.3 ขั้นตอนการสร้างซอฟต์แวร์

ในการสร้างซอฟต์แวร์นี้จะแบ่งเป็น 2 ส่วน ได้แก่ ส่วนของซอฟต์แวร์ที่ใช้ในการควบคุมแบบแมนนวล และซอฟต์แวร์ในการตรวจจับวัตถุที่จะให้หุ่นยนต์เคลื่อนที่ตาม

3.3.1 การสร้างชุดคำสั่งและอินเตอร์เฟสในการควบคุมแบบแมนนวล

ในส่วนนี้ผู้จัดทำใช้โปรแกรม Visual Studio ในการสร้างหน้าอินเตอร์เฟสขึ้นมา ภาษาที่ใช้เขียน คือ Visual Basic โดยมีการควบคุมการทำงานของหุ่นยนต์จากการรับคำสั่งจากปุ่มกดบนหน้าจอคอมพิวเตอร์ ที่ได้ออกแบบไว้ ดังรูปที่ 3.16 ซึ่งประกอบด้วยชุดคำสั่งควบคุมการเคลื่อนที่ ทั้งหมด 8 คำสั่ง คือ เดินหน้า ถอยหลัง เดินหน้าเลี้ยวซ้าย เดินหน้าเลี้ยวขวา หมุนตัวซ้าย หมุนตัวขวา ถอยหลังเลี้ยวซ้าย ถอยหลังเลี้ยวขวา

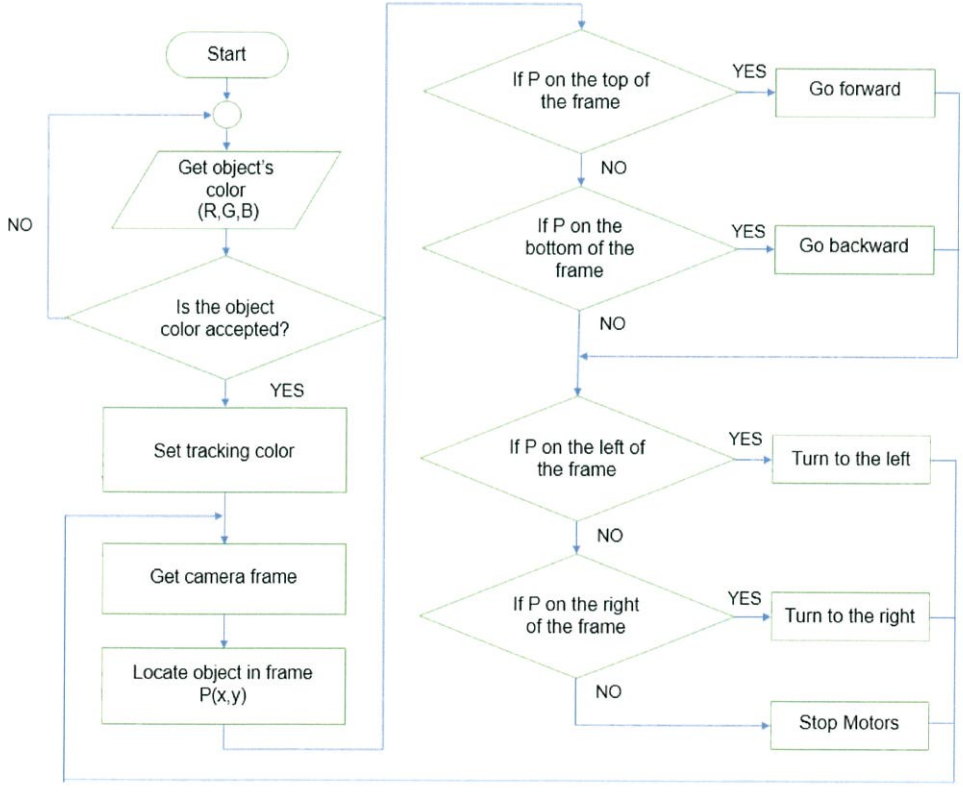
ในการเคลื่อนที่โปรแกรมจะคำนวณระยะทาง หรือองศาในการเคลื่อนที่เป็นจำนวนพัลส์ของล้อทั้งสองข้าง และสั่งให้หุ่นยนต์เคลื่อนที่ตามจำนวนพัลส์นั้น ซึ่งจะทำให้หุ่นยนต์เคลื่อนที่ได้ตามระยะทางหรือองศาในการเลี้ยวที่กำหนดไว้ ในการเคลื่อนที่ไปหน้า หรือถอยหลัง จะต้องใส่ระยะทางในการเคลื่อนที่ที่ต้องการ ส่วนในการเลี้ยว และการหมุนตัว จะต้องใส่องศาการเลี้ยว หรือการหมุน ลักษณะการใช้คำสั่งมีดังนี้ Forward และ Backward หุ่นยนต์จะเคลื่อนที่ตรงไปด้านหน้า หรือด้านหลังตามระยะทางที่ระบุ Turn Left และ Turn Right หุ่นยนต์จะหมันตัวอยู่กับที่ ตามองศาที่ระบุ โดยล้อทั้งสองข้างจะหมุนในทิศตรงกันข้าม F and L (Forward and Left), F and R (Forward and Right), B and L (Backward and Left) และ B and R (Backward and Right) หุ่นยนต์จะเคลื่อนที่เป็นเส้นโค้งตามองศาที่ระบุไว้ ในรัศมีการโค้ง 75 เซนติเมตร โดยล้อด้านที่เลี้ยวจะหมุนช้ากว่าอีกด้านหนึ่ง



รูปที่ 3.16 แสดงหน้ายูสเซอร์ อินเตอร์เฟสที่ใช้ในการบังคับหุ่นยนต์ จากโปรแกรม Visual Studio 2012

3.3.2 การสร้างซอฟต์แวร์สำหรับการเคลื่อนที่ตามวัตถุ

โปรแกรมที่ใช้ในการสร้างอินเตอร์เฟสนี้คือโปรแกรม Matlab โดยในการใช้โปรแกรมหุ่นยนต์เดินตามเส้นทางนี้ จะต้องวางเส้นทางที่มีสีแตกต่างจากสีพื้นอย่างเห็นได้ชัด เพื่อป้องการคลาดเคลื่อนในการตรวจจับสี โดยโปรแกรมจะจับค่าสี R G B จากกล้อง และนำมาเปรียบเทียบกับค่าสี R G B ที่กำหนด ซึ่งค่าสีที่กำหนด จะได้จากการจับค่าสีในโปรแกรม โดยการถ่ายภาพของวัตถุที่จะใช้เป็นเส้นทางของหุ่นยนต์ และแปลงเป็นค่า R G B หลังจากที่ได้สีที่กำหนดแล้ว โปรแกรมจะเปรียบเทียบค่าสีในแต่ละพิกเซลจากกล้องกับค่าสีที่กำหนด และเปลี่ยนเป็นคำสั่งให้หุ่นยนต์เคลื่อนที่ โดยถ้าพิกเซลที่มีค่าสีตรงกับที่กำหนดอยู่ทางขวาของกล้อง หุ่นยนต์ก็จะหมุนขวา ถ้าพิกเซลที่มีค่าสีตรงกับที่กำหนดอยู่ทางซ้ายของกล้อง หุ่นยนต์ก็จะหมุนซ้าย และถ้าพิกเซลที่มีค่าสีตรงกับที่กำหนดอยู่กึ่งกลางของกล้อง หุ่นยนต์ก็จะเดินหน้าและจะหยุดเมื่อไม่มีวัตถุที่มีสีที่กำหนดอยู่ในกล้อง ซึ่งมีแผนผังการทำงานของโปรแกรมดังรูปที่ 3.18 ซึ่งรายละเอียดและโค้ดทั้งหมดของโปรแกรม สามารถดูได้ใน ภาคผนวก ก



รูปที่ 3.17 แผนผังการทำงานของโปรแกรมสำหรับให้หุ่นยนต์เคลื่อนที่ตามวัตถุ

3.4 วิธีทดสอบและวัดผล

1. ทดสอบความแข็งแรงของตัวโครงสร้างของฐานหุ่นยนต์เคลื่อนที่ คือ โครงสร้างไม่มีส่วนเสียหาย แตกร้าว หรือมีชิ้นส่วนหลุดออกมาขณะใช้งาน
2. ทดสอบการเคลื่อนที่ขึ้นทางลาดชันและเคลื่อนที่ผ่านพื้นผิวขรุขระหรือพื้นที่มีอุปสรรค
3. ทดสอบความเที่ยงตรงในการเคลื่อนที่ไปตามจุดหมายที่กำหนด
4. ทดสอบการลากจูงสิ่งของที่มีน้ำหนักไม่เกิน 17 กิโลกรัม
5. ทดสอบระยะเวลาการใช้งานของแบตเตอรี่ สามารถใช้งานได้ต่อเนื่องไม่น้อยกว่า 30 นาที

บทที่ 4

ผลการดำเนินงาน

ผลการดำเนินงานนี้ เป็นการนำฐานหุ่นยนต์เคลื่อนที่มาทดสอบและวัดผลการทำงาน หลังจากทำการออกแบบและสร้างฐานหุ่นยนต์เคลื่อนที่ พร้อมทั้งซอฟต์แวร์ในการทำงานเสร็จสิ้นแล้ว ประกอบด้วย การทดสอบความแข็งแรงของตัวโครงสร้าง การรับน้ำหนัก ประสิทธิภาพในการเคลื่อนที่บนพื้นที่ไม่ปกติ และความคลาดเคลื่อนในการเคลื่อนที่แบบอัตโนมัติของฐานหุ่นยนต์เคลื่อนที่

4.1 การทดสอบแรงในการขับเคลื่อนมอเตอร์ของฐานหุ่นยนต์เคลื่อนที่

การคำนวณหาแรงที่ฐานหุ่นยนต์เคลื่อนที่สามารถขับเคลื่อนไปได้ วิธีการทดสอบ คือ ทำการเขียนโปรแกรมสั่งการให้หุ่นยนต์เคลื่อนที่เดินหน้า และทำการวัดโดยใช้เครื่องชั่งน้ำหนักแบบใช้มือถือคล้องกับตัวโครงของฐานหุ่นยนต์ และยึดเสาหลักไว้ หลังจากนั้นลองทำการเปลี่ยนค่า PWM เพื่อทำการหาค่าแรงที่มอเตอร์สามารถทำการลากได้ ดังรูปที่ 4.1 โดยฐานหุ่นยนต์เคลื่อนที่นี้มีน้ำหนัก 36.5 กิโลกรัม ซึ่งในการเก็บข้อมูลจะเก็บจากค่าของ PWM ระดับต่างๆ ที่ได้ป้อนให้กับโปรแกรม เพื่อวัดค่าแรงที่เกิดขึ้นจริงในการเคลื่อนที่ของฐานหุ่นยนต์เคลื่อนที่ ในหน่วยกิโลกรัม

รูปที่ 4.1 แสดงการทดสอบวัดค่าแรงลากในการขับเคลื่อนมอเตอร์

จากการทดสอบพบว่าค่า PWM ที่มากขึ้นจะทำให้แรงที่อ่านค่าได้มากขึ้นตามไปด้วย และค่า PWM สูงสุดที่ทำการทดลอง คือ ค่า PWM ที่ระดับ 50 เปอร์เซ็นต์ เนื่องจากถ้าใช้ค่า PWM ที่มากกว่านี้ อาจจะทำให้ไมโครคอนโทลเลอร์ได้รับความเสียหายได้ ซึ่งที่ค่า PWM ระดับนี้สามารถวัดแรงลากสูงสุดที่ฐานหุ่นยนต์ทำได้คือ 18.4 กิโลกรัม

4.2 การทดสอบการรับน้ำหนัก

ในการทดสอบนี้ จะทำการเพิ่มน้ำหนักไปบนตัวฐานหุ่นยนต์เคลื่อนที่ ครั้งละ 3.5 กิโลกรัม ดังตารางที่ 4.1 เพื่อทดสอบว่าสามารถแบกรับน้ำหนักบนตัวฐานกับการเคลื่อนที่ในระยะ 5 เมตร และ PWM ที่ 20% บนพื้นกระเบื้องยาง

ตารางที่ 4.1 ผลการทดสอบการรับน้ำหนักของฐานหุ่นยนต์เคลื่อนที่

น้ำหนักวัตถุ (kg)	ผลการทดสอบ		
	PWM	ระยะทาง (เมตร)	เวลา (วินาที)
0	20%	5	2.1
3.5	20%	5	2.1
7	20%	5	2.1
10.5	20%	5	2.15
14	20%	5	2.2
17.5	20%	5	2.35
21	20%	5	2.35

4.3 การทดสอบเพื่อหาประสิทธิภาพในการเคลื่อนที่บนพื้นที่ไม่ปกติ

ในการทดสอบความสามารถในการเคลื่อนที่บนพื้นที่ไม่ปกติ มีการทดสอบ คือ นำฐานหุ่นยนต์เคลื่อนที่ไปทดลอง การเคลื่อนที่บนพื้นที่ที่มีสภาพพื้นที่ต่างๆ ได้แก่ พื้นกระเบื้อง พื้นเรซิ่น หินตัวหนอน พื้นขรุขระ ในระยะทาง 5 เมตร และทางลาดชัน 25 องศา ระยะทาง 1.5 เมตร โดยไม่มีการรับน้ำหนัก และ PWM ที่ 20%

ตารางที่ 4.2 ผลการทดสอบการเคลื่อนที่ของฐานหุ่นยนต์เคลื่อนที่

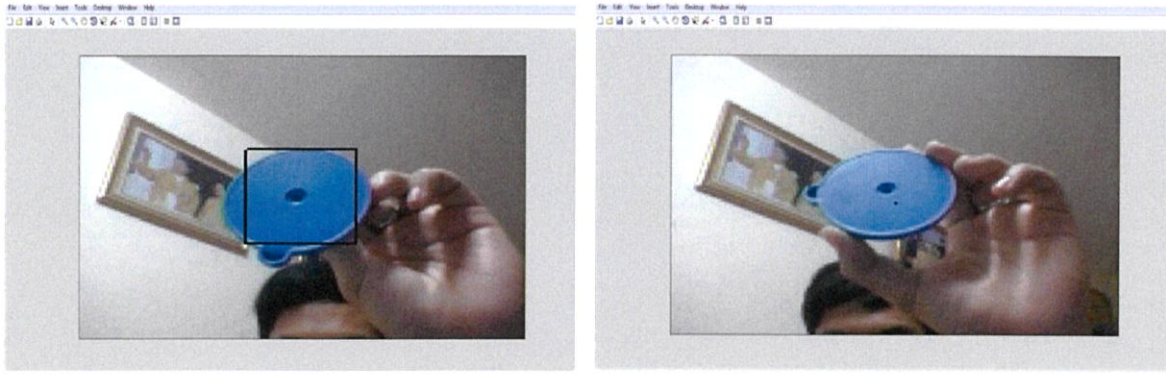
บริเวณการเคลื่อนที่	ผลการทดสอบ		
	PWM	ระยะทาง (เมตร)	เวลา (วินาที)
พื้นกระเบื้อง	20%	5	2.1
พื้นเรซิ่น	20%	5	2.1
หินตัวหนอน	20%	5	2.1
พื้นขรุขระ	20%	5	2.1
ทางชัน 25 องศา	20%	1.5	0.8

จากการทดสอบพบว่าฐานหุ่นยนต์สามารถเคลื่อนที่ในทางราบ ทางขรุขระ และทางลาดชันได้ เนื่องจากมีการออกแบบด้วยล้อแบบตีนตะขาบ

4.4 การทดสอบการเคลื่อนที่ตามวัตถุของฐานหุ่นยนต์เคลื่อนที่

การทดสอบนี้เป็นการทดสอบการเดินตามวัตถุ ดังรูปที่ 4.2 โดยทำโปรแกรมสำหรับการตรวจจับวัตถุด้วยสีแล้วใช้วัตถุที่ตรงกับสีที่ตั้งค่าไว้ในตอนแรกเป็นมาทำเป็นเส้นทางให้หุ่นยนต์เคลื่อนที่ตาม โดยใช้กล้องจากโน้ตบุ๊คแทนเซอเซอร์ ดังตัวอย่างรูปที่ 4.3(a) เป็นหน้าต่างของโปรแกรมการตรวจจับสีของวัตถุที่จะใช้ โดยโปรแกรมจะทำการอ่านค่าสีของวัตถุที่อยู่ในกรอบสีเหลี่ยม รูปที่ 4.3(b) แสดงบริเวณที่โปรแกรมตรวจจับสีของวัตถุจะเห็นเป็นจุดสีดำเล็กๆ ที่ตำแหน่งกึ่งกลางของวัตถุนั้น

รูปที่ 4.2 การทดสอบการเดินตามเส้นทาง



(a) (b)

รูปที่ 4.3 การตรวจจับสีของวัตถุด้วยกล้อง

บทที่ 5

สรุปผลการดำเนินงาน

โครงงานฐานหุ่นยนต์เคลื่อนที่ เริ่มตั้งแต่ขั้นตอนการศึกษา ออกแบบ จนถึงขั้นตอนการสร้าง ในส่วนของฮาร์ดแวร์มีการออกแบบโครงสร้างให้มีลักษณะเป็นแบบล้อตีนตะขาบ เพื่อให้สามารถเคลื่อนที่ได้ในหลายสภาพพื้นผิว โดยมีส่วนการควบคุมฮาร์ดแวร์ทั้งหมด คือ ไมโครคอนโทลเลอร์ ทำหน้าที่รับคำสั่งจากโปรแกรมการควบคุมการเคลื่อนที่ ที่มีสองระบบคือแบบแมนนวล และแบบเดินตามเส้นทาง ได้อัตโนมัติไปตามจุดหมายตามเส้นทาง ซึ่งใช้พลังงานจากแบตเตอรี่ ทั้งนี้หุ่นยนต์ต้องสามารถลากจูงหรือเคลื่อนย้ายสิ่งของที่มีน้ำหนักไม่เกิน 17 กิโลกรัมได้ จึงเลือกใช้มอเตอร์ไฟฟ้าขนาด 250 วัตต์ โดยมีวงจรขับมอเตอร์แบบ H-bridge เป็นตัวควบคุมและจ่ายไฟให้กับมอเตอร์ ในส่วนของซอฟต์แวร์ จะสร้างชุดคำสั่งสำหรับควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์ ด้วยโปรแกรม Visual Basic และ Matlab ในการเขียนชุดคำสั่งให้กับหุ่นยนต์ และจากการออกแบบ การดำเนินงานสร้าง และทดสอบฐานหุ่นยนต์เคลื่อนที่ ในส่วนนี้จะกล่าวถึง ผลการทดสอบส่วนของฮาร์ดแวร์ และซอฟต์แวร์ รวมทั้งข้อเสนอแนะเพื่อใช้เป็นแนวทางในการพัฒนาต่อไป

5.1 ผลการทดสอบ

ในการทดสอบจะแบ่งออกแบบสองส่วนคือการทดสอบด้านฮาร์ดแวร์ และการทดสอบด้านซอฟต์แวร์

5.1.1 ผลการทดสอบฮาร์ดแวร์

ผลการดำเนินงานจากการทดสอบและประเมินผล พบว่าฐานหุ่นยนต์สามารถใช้แรงในการลากสิ่งของได้ 18 กิโลกรัมที่ PWM 20 เปอร์เซ็นต์ ซึ่งกำหนดไว้เป็นค่าสูงสุดที่ใช้ในการทดสอบ และการออกแบบโดยใช้ล้อตีนตะขาบก็สามารถทำให้ฐานหุ่นยนต์สามารถเคลื่อนที่บนพื้นที่ ที่เป็นทางขรุขระและทางลาดชันได้ดี

5.1.2 ผลการทดสอบซอฟต์แวร์

ผลการทดสอบด้านซอฟต์แวร์ พบว่าในส่วนของการควบคุมการเคลื่อนที่แบบแมนนวลสามารถทำให้หุ่นยนต์เคลื่อนที่ไปในเส้นทางเป้าหมายที่ต้องการได้ แต่ยังคงมีค่าความคลาดเคลื่อนของระยะทางที่คำนวณได้จากโปรแกรมกับที่วัดได้จริงอยู่ประมาณ 2 เซนติเมตร และในส่วนของโปรแกรมที่ทำให้ฐานหุ่นยนต์สามารถเดินตามเส้นทางได้อัตโนมัติ ฐานหุ่นยนต์สามารถเดินตามเส้นทางที่มีจุดสีตรงกับสีที่กำหนดไว้ได้

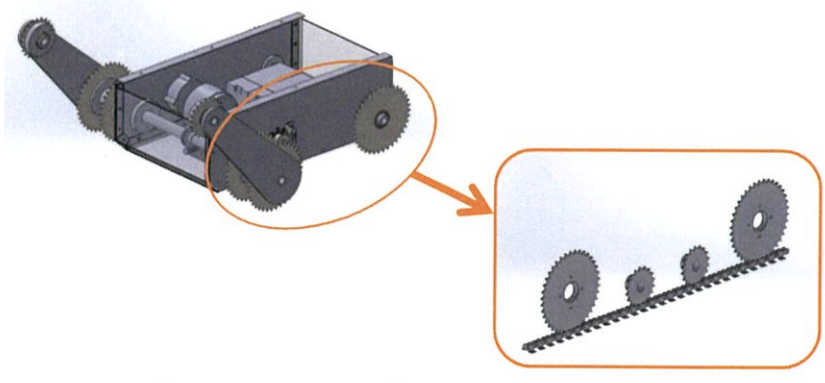
5.2 ข้อเสนอแนะ

1. ส่วนของฮาร์ดแวร์ โซ่ตีนตะขาบจะมีแรงเสียดทานกับพื้นค่อนข้างสูง ในการหมุนตัวที่องศาการหมุนมากๆ และการใช้ความเร็วสูงจะส่งผลให้โซ่ล้อตีนตะขาบหลุดออกจากเฟืองล้อได้ ดังนั้นควรติดตั้งเฟืองโซ่ขนาดเล็กกว่าเฟืองล้อไว้ระหว่างเฟืองล้อหน้า และล้อหลัง เพื่อไม่ให้โซ่หย่อนเกินไปจนหลุดจากเฟืองล้อ ดังรูปที่ 5.1

2. พัฒนาการควบคุมให้เป็นแบบไร้สาย แทนการควบคุมโดยใช้คอมพิวเตอร์โน้ตบุ๊คเพื่อความสะดวกในการใช้งานมากขึ้น

3. ติดเซนเซอร์ตรวจจับสิ่งกีดขวาง เพื่อเพิ่มประสิทธิภาพในการใช้งานอย่างปลอดภัย

4. พัฒนาโครงสร้างให้มีน้ำหนักที่เบาลง และรับน้ำหนักได้มากขึ้น



รูปที่ 5.1 การออกแบบเพื่อแก้ปัญหาโซ่หย่อน

เอกสารอ้างอิง

เดชฤทธิ์ มณีธรรม. (2553). *คัมภีร์หุ่นยนต์.* กรุงเทพฯ: เพชรเกษมพริ้นติ้ง กรุ๊ป.

ดอนสัน ปงผาบ. (2549). *ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน 2.* กรุงเทพฯ: สำนักพิมพ์ สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

เอกชัย มะการ. (2552). *เรียนรู้ เข้าใจ ใช้งาน ไมโครคอนโทลเลอร์ตระกูล AVR ด้วย Arduino.* กรุงเทพฯ: บริษัท อีทีที จำกัด.

อดิศักดิ์ ชิณะวงศ์. (8 มกราคม 2543). *Microcontroller.* เข้าถึงได้จาก adisak51: http://www.adisak51.com/page21.html

Chaiyaporn Silawatchananai. (2 กรกฎาคม 2557). *การควบคุมมอเตอร์ไฟฟ้ากระแสตรง.* เข้าถึงได้จาก aimagin blog: http://aimagin.com/blog/motor/?lang=th

Chang. (10 พฤษภาคม 2558). *PWM (Pulse Width Modulation).* เข้าถึงได้จาก *ayarafun make, creative and let's fun*: http://www.ayarafun.com/2015/05/dimmer-led-funbasic-io/

Jump Startinnovation. (26 กรกฎาคม 2556). *ไมโครคอนโทลเลอร์คืออะไร.* เข้าถึงได้จาก *Blogspot*: http://jumpstartinnovation.blogspot.com/2013/07/blog-post.html

ภาคผนวก ก

โปรแกรมสั่งงานการเคลื่อนที่ของหุ่นยนต์

โปรแกรมการควบคุมแบบแมนนวลเขียนโดย Arduino IDE

```
int incomingByte1=0;
int incomingByte2=0;
int incomingByte3=0;
int incomingByte4=0;
int PWMR = 0;
int PWML = 0;
int i;
int D1;
int D2;
int Ax1;                    //Left Sensor
int Bx1;
int Cx1;
int Cbar1;
int ACT1;
int CNT1;
int Ax2;                    //Right Sensor
int Bx2;
int Cx2;
int Cbar2;
int ACT2;
int CNT2;

void setup()
{       Serial.begin(921600);
        pinMode(8, OUTPUT);                 //Motor R
        pinMode(9, OUTPUT);
        pinMode(10, OUTPUT);
        pinMode(3, OUTPUT);                 //Motor L
        pinMode(4, OUTPUT);
```

```
        pinMode(5, OUTPUT);
        pinMode(11, INPUT);                 //Left Sensor
        pinMode(12, INPUT);
        pinMode(6, INPUT);                  //Right Sensor
        pinMode(7, INPUT);
        ACT1=0;
        ACT2=0;
        CNT1=0;
        CNT2=0;
}
void loop()
{       i++;
        if (i>20){i=0;}
        if (i<PWMR){ digitalWrite(10, HIGH);}
        if (i>PWMR){ digitalWrite(10, LOW);}
        if (i<PWML){ digitalWrite(5, HIGH);}
        if (i>PWML){ digitalWrite(5, LOW);}
        if (Serial.available() > 0)
        {       incomingByte1 = incomingByte2;
                incomingByte2 = incomingByte3;
                incomingByte3 = incomingByte4;
                incomingByte4 = Serial.read();
                D1 = 500 + CNT1;
                D2 = 500 + CNT2;
                Serial.write('@');
                if (CNT1 < -400){Serial.print('0');}
                Serial.print(D1,DEC);
                if (CNT2 < -400){Serial.print('0');}
                Serial.println(D2,DEC);
        }
```

```
if (incomingByte1 == '@')
        {       if (incomingByte2 == 'Z')
                {       CNT1 = 0;
                        CNT2 = 0;
                }
                 if (incomingByte2 == 'M')
                {       if (incomingByte3 > 93)                 //Left Forward
                        {       digitalWrite(3,HIGH);
                                        digitalWrite(4,LOW);
                                        PWML = incomingByte3-93;
                        }
                         if (incomingByte3 < 93)                //Left Backward
                        {       digitalWrite(3,LOW);
                                        digitalWrite(4,HIGH);
                                        PWML = 93-incomingByte3;
                         }
                         if (incomingByte3 == 93 )
                        {       digitalWrite(3,LOW);
                                        digitalWrite(4,LOW);
                        }
                        if (incomingByte4 > 93)                 //Right Forward
                        {       digitalWrite(8,LOW);
                                        digitalWrite(9,HIGH);
                                        PWMR = incomingByte4-93;
                        }
                         if (incomingByte4 < 93)                //Right Backward
                        {       digitalWrite(8,HIGH);
                                        digitalWrite(9,LOW);
                                        PWMR = 93-incomingByte4;
                        }
```

```
                    if (incomingByte4 == 93)
                    {       digitalWrite(8,LOW);
                            digitalWrite(9,LOW);
                    }
            }
    }

Ax1 = digitalRead(11);                  //Left Sensor
Bx1 = digitalRead(12);
Cx1 = ((Ax1 * 2)+ Bx1 );
  if (ACT1 == 0 & Cbar1 == 1 & Cx1 == 3)
  {     CNT1++;
        ACT1 = 1;
  }
  if (ACT1 == 0 & Cbar1 == 2 & Cx1 == 3)
  {     CNT1--;
        ACT1 = 1;
  }
  if (Cx1 == 0)
  {     ACT1 = 0;       }
  Cbar1 = Cx1 ;

Ax2 = digitalRead(6);                   //Right Sensor
Bx2 = digitalRead(7);
Cx2 = (( Ax2 *2)+ Bx2 );
  if (ACT2 == 0 & Cbar2 == 1 & Cx2 == 3)
  {     CNT2++;
        ACT2 = 1;
  }
```

```
        if (ACT2 == 0 & Cbar2 == 2 & Cx2 == 3)
          {     CNT2--;
                ACT2 = 1;
          }
          if (Cx2 == 0)
          {     ACT2 = 0;       }
          Cbar2 = Cx2 ;
  }
```

โปรแกรมเคลื่อนที่ตามวัตถุเขียนโดย Arduino IDE

```
int incomingByte1=0;
int incomingByte2=0;
int incomingByte3=0;
int incomingByte4=0;
int PWMR = 0;
int PWML = 0;
int i;

void setup()
{       Serial.begin(250000);
        pinMode(8, OUTPUT);                 //Motor R
        pinMode(9, OUTPUT);
        pinMode(10, OUTPUT);
        pinMode(3, OUTPUT);                 //Motor L
        pinMode(4, OUTPUT);
        pinMode(5, OUTPUT);
        pinMode(11, INPUT);                 //Left Sensor
        pinMode(12, INPUT);
        pinMode(6, INPUT);                  //Right Sensor
        pinMode(7, INPUT);
```

```
        ACT1=0;
        ACT2=0;
        CNT1=0;
        CNT2=0;
}

void loop()
{       i++;
        if (i>20){i=0;}
        if (i<PWMR){ digitalWrite(10, HIGH);}
        if (i>PWMR){ digitalWrite(10, LOW);}
        if (i<PWML){ digitalWrite(5, HIGH);}
        if (i>PWML){ digitalWrite(5, LOW);}
        if (Serial.available() > 0)
        {       incomingByte1 = incomingByte2;
                incomingByte2 = incomingByte3;
                incomingByte3 = incomingByte4;
                incomingByte4 = Serial.read();
        }
                if (incomingByte1 == '@')
                {       if (incomingByte2 == 'M')
                        {
                                if (incomingByte3 > 93)                 //Left Forward
                                {       digitalWrite(3,HIGH);
                                        digitalWrite(4,LOW);
                                        PWML = incomingByte3-93;
                                }
```

```
                    if (incomingByte3 < 93)                 //Left Backward
                    {       digitalWrite(3,LOW);
                                digitalWrite(4,HIGH);
                                PWML = 93-IncomingByte3;
                    }
                    if (incomingByte3 == 93 )
                    {       digitalWrite(3,LOW);
                                digitalWrite(4,LOW);
                    }
                    if (incomingByte4 > 93)                 //Right Forward
                    {       digitalWrite(8,LOW);
                                digitalWrite(9,HIGH);
                                PWMR = incomingByte4-93;
                    }
                    if (incomingByte4 < 93)                 //Right Backward
                    {       digitalWrite(8,HIGH);
                                digitalWrite(9,LOW);
                                 PWMR = 93-incomingByte4;
                    }
                    if (incomingByte4 == 93)
                    {       digitalWrite(8,LOW);
                                digitalWrite(9,LOW);
                    }
            }
        }
}
```

ยูสเซอร์อินเตอร์เฟสการควบคุมแบบแมนนวล เขียนโดยโปรแกรม Visual basic

```
Public Class Form1
    Private Sub Form1_Leave(sender As Object, e As EventArgs) Handles Me.Leave
        SerialPort1.Close()
    End Sub
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        SerialPort1.Open()
    End
    Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
        Dim x As String
        Dim A1, A2, A3, A4, A5, A6, A7 As Char
        Dim lx, rx As Integer
        x = SerialPort1.ReadExisting
        TextBox7.Text = x
        A1 = Mid(x, 1, 1)
        If A1 = "@" Then
                A2 = Mid(x, 2, 1)
                A3 = Mid(x, 3, 1)
                A4 = Mid(x, 4, 1)
                A5 = Mid(x, 5, 1)
                A6 = Mid(x, 6, 1)
                A7 = Mid(x, 7, 1)
      lx = ((Val(A2) * 100) + (Val(A3) * 10) + Val(A4))
                rx = ((Val(A5) * 100) + (Val(A6) * 10) + Val(A7))
                TextBox1.Text = lx - 500
                TextBox9.Text = rx - 500
        End If
```

```
        Dim ds1, ds2, SL, SR, L, R As Integer
        Dim fs1, fs2 As Double
        L = Math.Abs(Val(TextBox6.Text))
        R = Math.Abs(Val(TextBox2.Text))
        If L <> R Then
                ds1 = Val(TextBox1.Text) - Val(TextBox6.Text)
                ds2 = Val(TextBox9.Text) - Val(TextBox2.Text)
                SL = Math.Abs(Math.Round(Val(TextBox6.Text) / 10))
                SR = Math.Abs(Math.Round(Val(TextBox2.Text) / 10))
                fs1 = ds1 / 10
                fs2 = ds2 / 10
                If fs1 > 1 Then fs1 = 1
                If fs2 > 1 Then fs2 = 1
                If fs1 < -1 Then fs1 = -1
                If fs2 < -1 Then fs2 = -1
                SerialPort1.WriteLine("@M" + Chr(93 - Math.Round(SL * fs1)) + Chr(93 -
Math.Round(SR * fs2)))
        End If
        If L = R Then
                ds1 = Val(TextBox1.Text) - Val(TextBox6.Text)
                ds2 = Val(TextBox9.Text) - Val(TextBox2.Text)
                If ds1 > 10 Then ds1 = 10
                If ds1 < -10 Then ds1 = -10
                If ds2 > 10 Then ds2 = 10
                If ds2 < -10 Then ds2 = -10
                SerialPort1.WriteLine("@M" + Chr(93 - Math.Round(ds1)) + Chr(93 -
Math.Round(ds2)))
        End If
End Sub
```

```
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button3.Click
    Dim pulseL, pulseR As Integer
    Dim P As Double
    P = 2.6
    pulseL = Math.Round(Val(TextBox4.Text) / P)
    pulseR = Math.Round(Val(TextBox4.Text) / P)
    TextBox6.Text = Val(pulseL)
    TextBox2.Text = Val(pulseR)
End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button4.Click
    SerialPort1.WriteLine("@Z")
    TextBox2.Text = 0
    TextBox6.Text = 0
End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim pulseL, pulseR As Integer
    Dim P As Double
    P = 2.6
    pulseL = Math.Round(Val(TextBox4.Text) / P)
    pulseR = Math.Round(Val(TextBox4.Text) / P)
    TextBox6.Text = Val(pulseL) * -1
    TextBox2.Text = Val(pulseR) * -1
End Sub

Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button6.Click
    Dim Width, pi, P As Double
    Dim A, pulseL, pulseR As Integer
    pi = 22 / 7
    Width = 40.2
```

```
    A = Val(TextBox5.Text)
    P = 2.6
    pulseL = Math.Round(((pi * Width) / (360 / A)) / P)
    pulseR = Math.Round(((pi * Width) / (360 / A)) / P)
    TextBox6.Text = Val(pulseL)
    TextBox2.Text = Val(pulseR) * -1
End Sub

Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button7.Click
    Dim Width, pi, P As Double
    Dim A, pulseL, pulseR As Integer
    pi = 22 / 7
    Width = 40.2
    A = Val(TextBox5.Text)
    P = 2.6
    pulseL = Math.Round(((pi * Width) / (360 / A)) / P)
    pulseR = Math.Round(((pi * Width) / (360 / A)) / P)
    TextBox6.Text = Val(pulseL) * -1
    TextBox2.Text = Val(pulseR)
End Sub

Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button8.Click
    Dim W, pi, P, DL, DR As Double
    Dim A, pulseL, pulseR, S As Integer
    pi = 22 / 7
    W = 40.2
    A = Val(TextBox5.Text)
    P = 2.6
    S = 70
    DL = (S + W) * A * pi / 180
    DR = (S) * A * pi / 180
```

```
    pulseL = Math.Round(DL / P)
    pulseR = Math.Round(DR / P)
    TextBox6.Text = Val(pulseL)
    TextBox2.Text = Val(pulseR)

End Sub
Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button5.Click
    Dim W, pi, P, DL , DR As Double
    Dim A, pulseL, pulseR, S As Integer
    pi = 22 / 7
    W = 40.2
    A = Val(TextBox5.Text)
    P = 2.6
    S = 70
    DR = (S + W) * A * pi / 180
    DL = (S) * A * pi / 180
    pulseL = Math.Round(DL / P)
    pulseR = Math.Round(DR / P)
    TextBox6.Text = Val(pulseL)
    TextBox2.Text = Val(pulseR)
End Sub

Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button9.Click
    Dim W, pi, P, DL, DR As Double
    Dim A, pulseL, pulseR, S As Integer
    pi = 22 / 7
    W = 40.2
    A = Val(TextBox5.Text)
    P = 2.6
    S = 70
```

```
        DL = (S + W) * A * pi / 180
        DR = (S) * A * pi / 180
        pulseL = Math.Round(DL / P)
        pulseR = Math.Round(DR / P)
        TextBox6.Text = Val(pulseL) * -1
        TextBox2.Text = Val(pulseR) * -1
    End Sub

    Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button10.Click
        Dim W, pi, P, DL, DR As Double
        Dim A, pulseL, pulseR, S As Integer
        pi = 22 / 7
        W = 40.2
        A = Val(TextBox5.Text)
        P = 2.6
        S = 70
        DR = (S + W) * A * pi / 180
        DL = (S) * A * pi / 180
        pulseL = Math.Round(DL / P)
        pulseR = Math.Round(DR / P)
        TextBox6.Text = Val(pulseL) * -1
        TextBox2.Text = Val(pulseR) * -1
    End Sub
End Class
```

ยูสเซอร์อินเตอร์เฟสการใช้งาน Camera tracking เขียนโดยโปรแกรม Matlab

```
clc;
clear all;
if (not(exist('s1')))
        s1= serial('com6');
        set(s1,'BaudRate',250000,'DataBits',8,'Stopbits',1,'Parity','None','RequestToSend','off');
        set(s1,'terminator', 13,'BytesAvailableFcnMode','byte');
end
if(strcmp(s1.Status,'closed'))
        fopen(s1);
end

vid = videoinput('winvideo', 2,'YUY2_160x120');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.TriggerRepeat = inf;
vid.ReturnedColorspace = 'rgb';vid.Timeout=3;
t_box=uint8(ones(120,160));
start(vid);

pos=[1 1];
ps=[1 1];
p11x = 1;
p12x = 1;
p21x = 1;
p22x = 1;
for i=1:1200
        tom = getdata(vid);
        rx=226;                         //ค่าสี R
        gx=210;                         //ค่าสี G
```

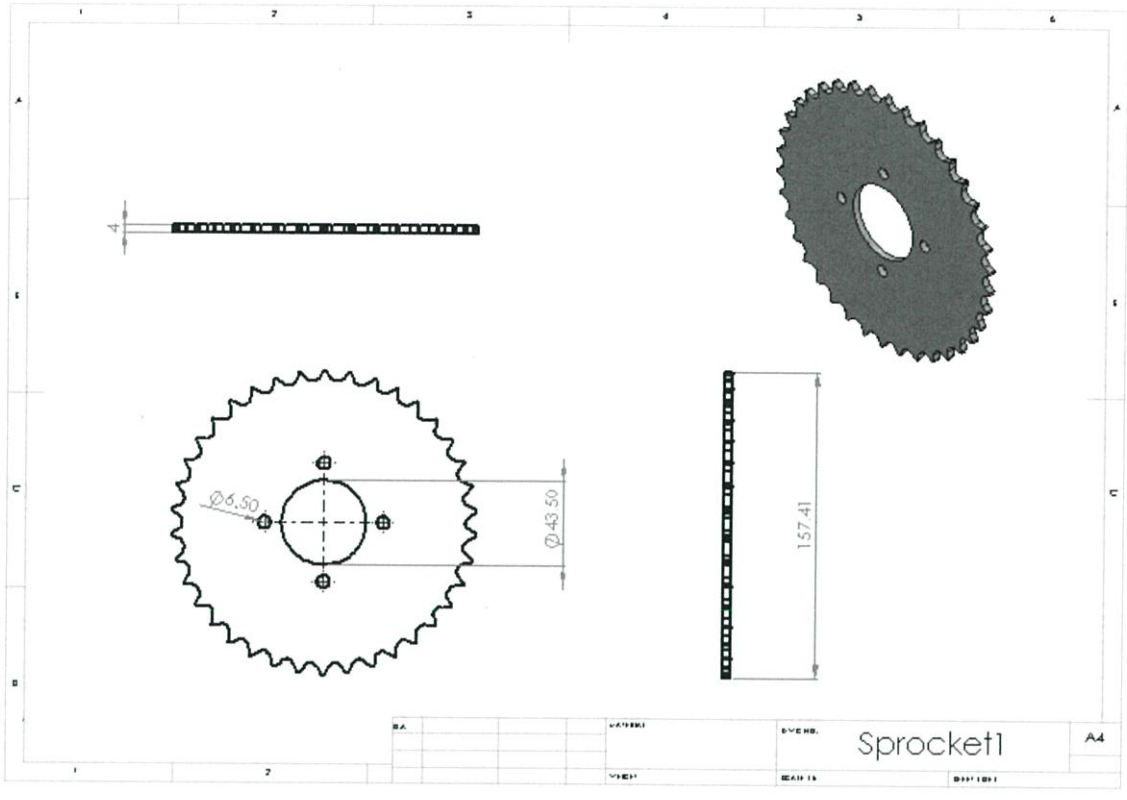
```
bx=70;                          //ค่าสี B
tx=40;                          //ค่าเผื่อช่วงเฉดสี
r=tom(:,:,1)>rx-tx & tom(:,:,1)<rx+tx ;
g=tom(:,:,2)>gx tx & tom(:,:,2)<gx+tx ;
b=tom(:,:,3)>bx-tx & tom(:,:,3)<bx+tx ;
x=r + g + b;
y=(x==3);

if (sum(sum(y))>0)
        pos(1)= floor(sum(sum(y)*(1:160)')/(sum(sum(y))));
        pos(2)= floor(sum(sum(y')*(1:120)')/(sum(sum(y))));
        p11x = pos(1);
        p12x = p11x;
        ps(1) = floor((pos(1)+p11x+p12x)/3);
        p21x = pos(2);
        p22x = p21x;
        ps(2) = floor((pos(2)+p21x+p22x)/3);
end
if (ps(1)-80 > 18)
        L = 100;
        R = 86;
end
if (ps(1)-80 < -18)
        L = 86;
        R = 100;
end

if ((ps(1)-80 >= -15 && ps(1)-80 <= 10 ) && ps(2)< 105 )
        L = 97;
        R = 97;
end
```

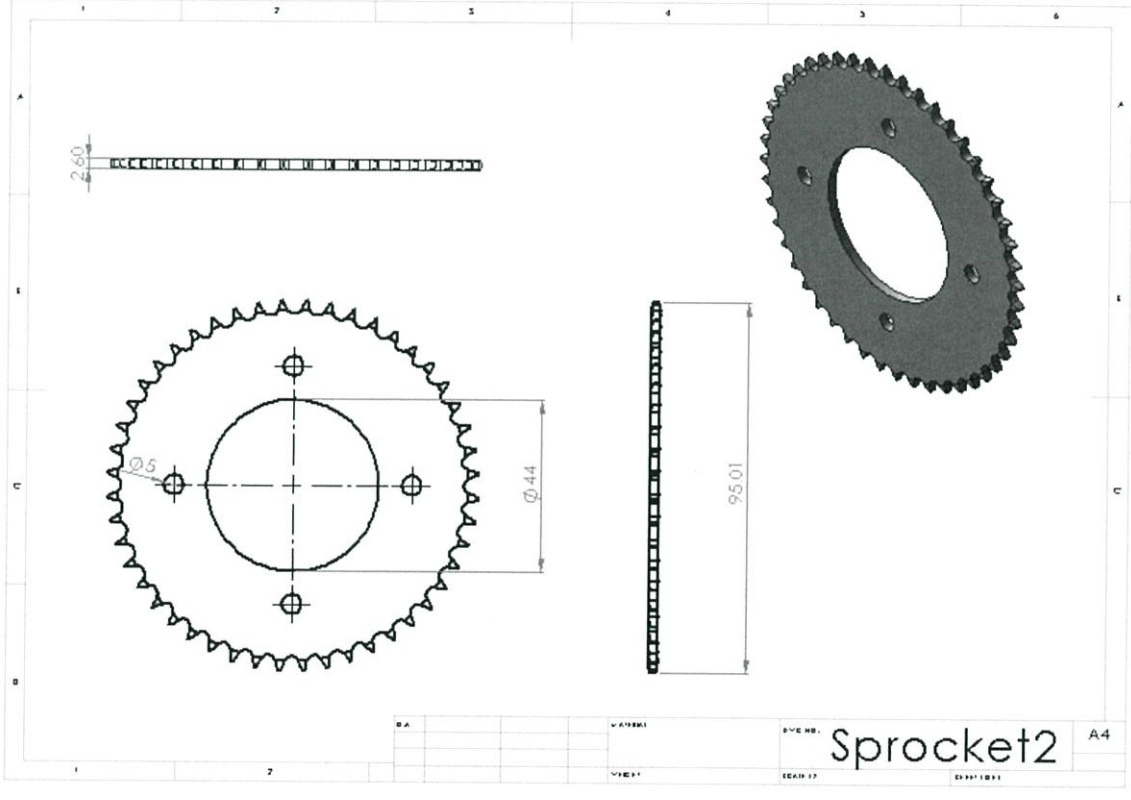
```
        if (ps(1) <= 5 || ps(1)>= 155 || ps(2) <= 5 || ps(2) >= 105)
                L = 93;
                R = 93;
        end
        fwrite(s1,['@M' L R],'uchar');
        tom(ps(2),ps(1),1)=0;
        tom(ps(2),ps(1),2)=0;
        tom(ps(2),ps(1),3)=0;
        imaqmontage(tom);
end

stop(vid);
delete(vid);
fwrite(s1,['@M' 93 93],'uchar');
fclose(s1);

//รับค่าสีจากวัตถุ
vid = videoinput('winvideo', 2,'YUY2_160x120');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.TriggerRepeat = inf;
vid.ReturnedColorspace = 'rgb';vid.Timeout=3;
t_box=uint8(ones(120,160));
for i=1:40
        t_box((120/2)-20+i,(160/2)+20)=0;
        t_box((120/2)-20+i,(160/2)-20)=0;
        t_box((120/2)+20,(160/2)-20+i)=0;
        t_box((120/2)-20,(160/2)-20+i)=0;
end
```

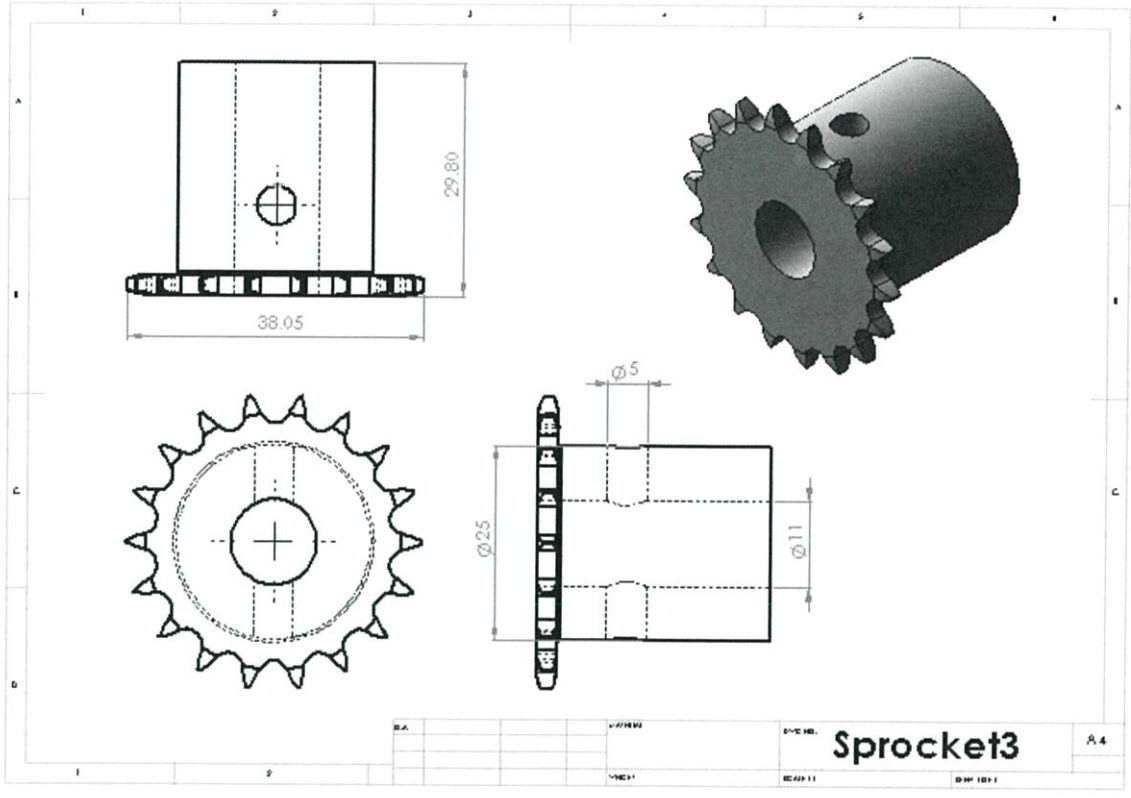
```
start(vid);
tom = getdata(vid);
tomx=tom;
for i=1:100
        tom = getdata(vid);
        tomx(:,:,1)=tom(:,:,1).*t_box;
        tomx(:,:,2)=tom(:,:,2).*t_box;
        tomx(:,:,3)=tom(:,:,3).*t_box;
        imaqmontage(tomx);
end
stop(vid);
delete(vid);
cx=uint32([0 0 0]);
cnt=0;
for i=41:79
        for j=61:99
                cx=cx+uint32([tomx(i,j,1) tomx(i,j,2) tomx(i,j,3)]);
                cnt=cnt+0.8;
        end
end
cx/cnt
red=cx(1)/cnt;
green=cx(2)/cnt;
blue=cx(3)/cnt;

red2 = double(cx(1))/(cnt*255);
green2 = double(cx(2))/(cnt*255);
blue2 = double(cx(3))/(cnt*255);
// นำค่า red2, green2, blue2 ไปใช้งานเป็นค่า RGB ต่อไป
```

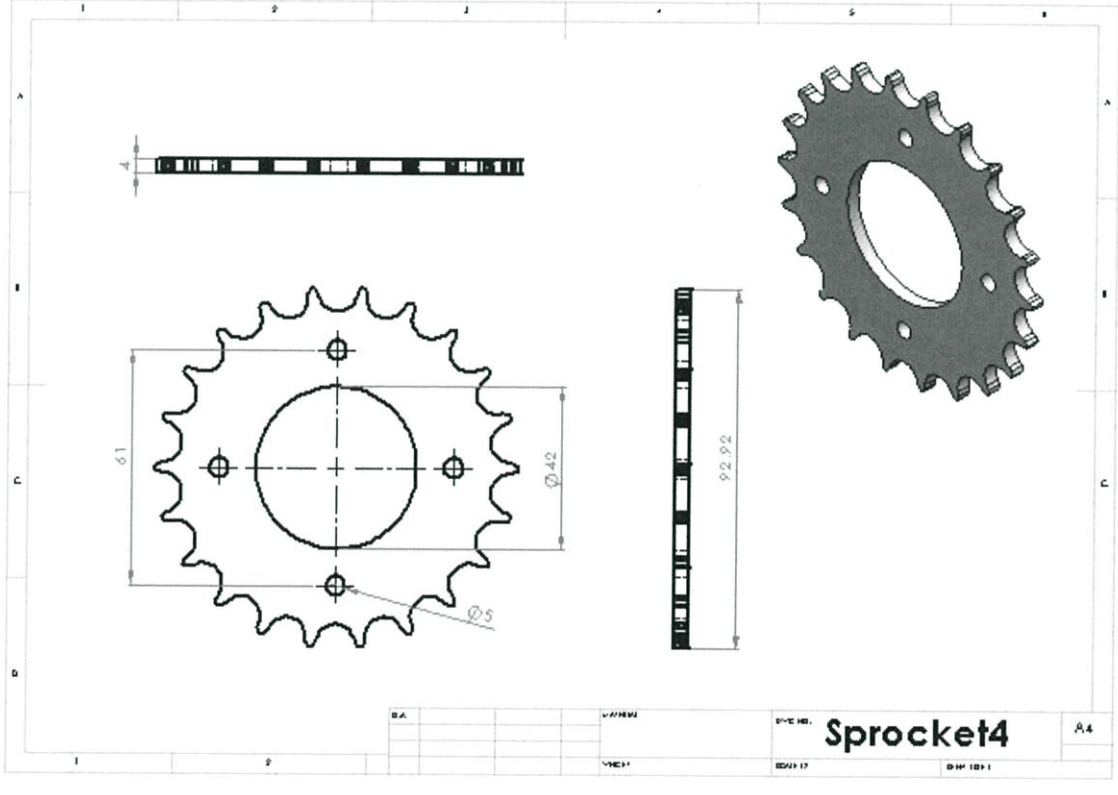
ภาคผนวก ข

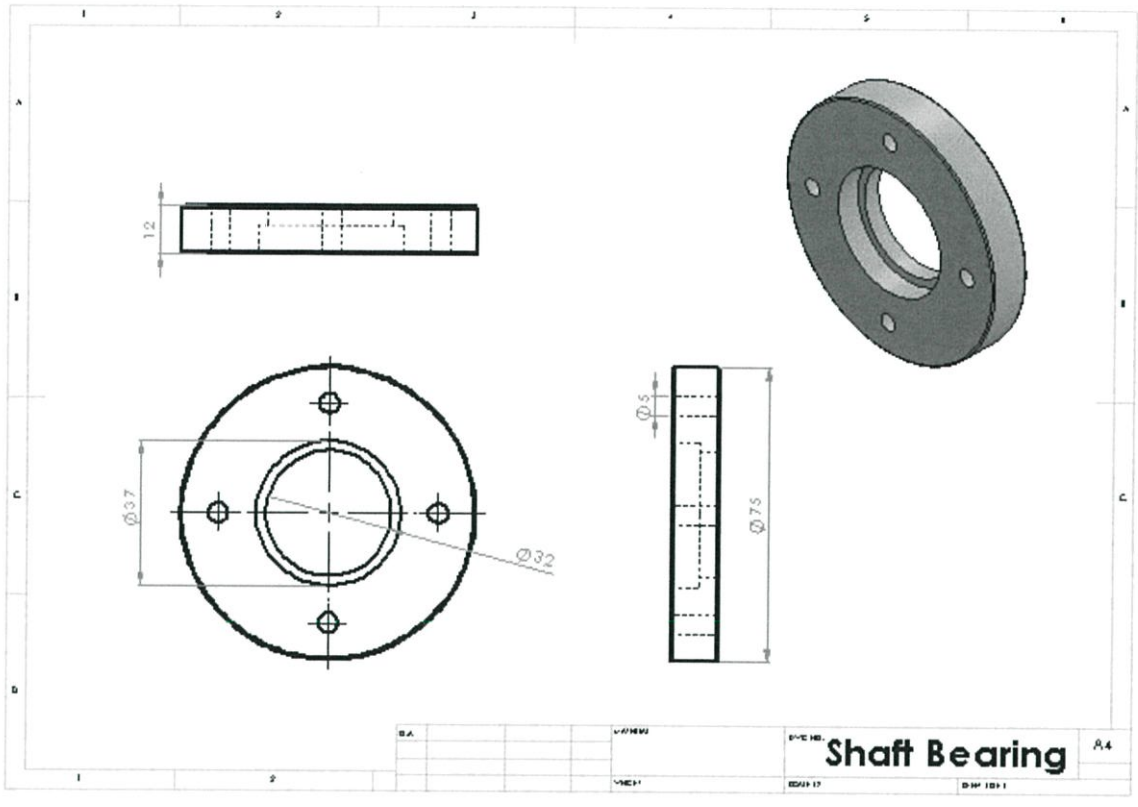
แบบรายละเอียดของหุ่นยนต์

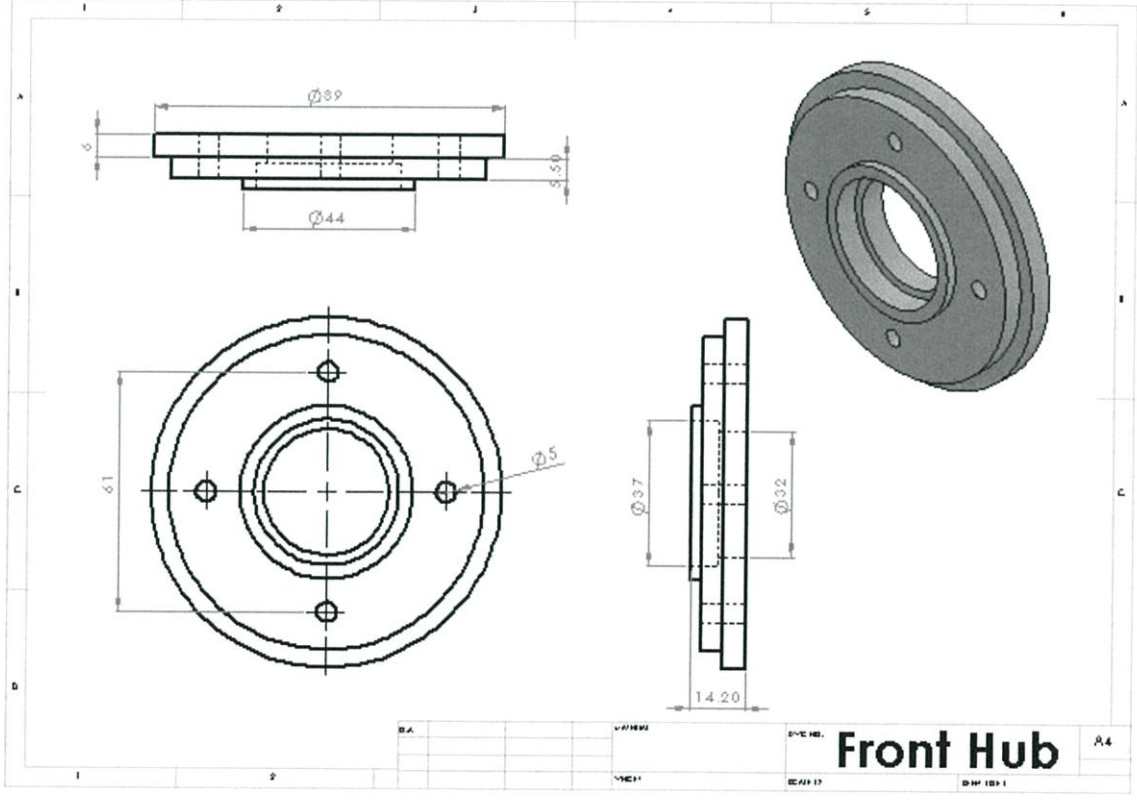

4
Ø6.50
Ø43.50
157.41
Sprocket1
A4

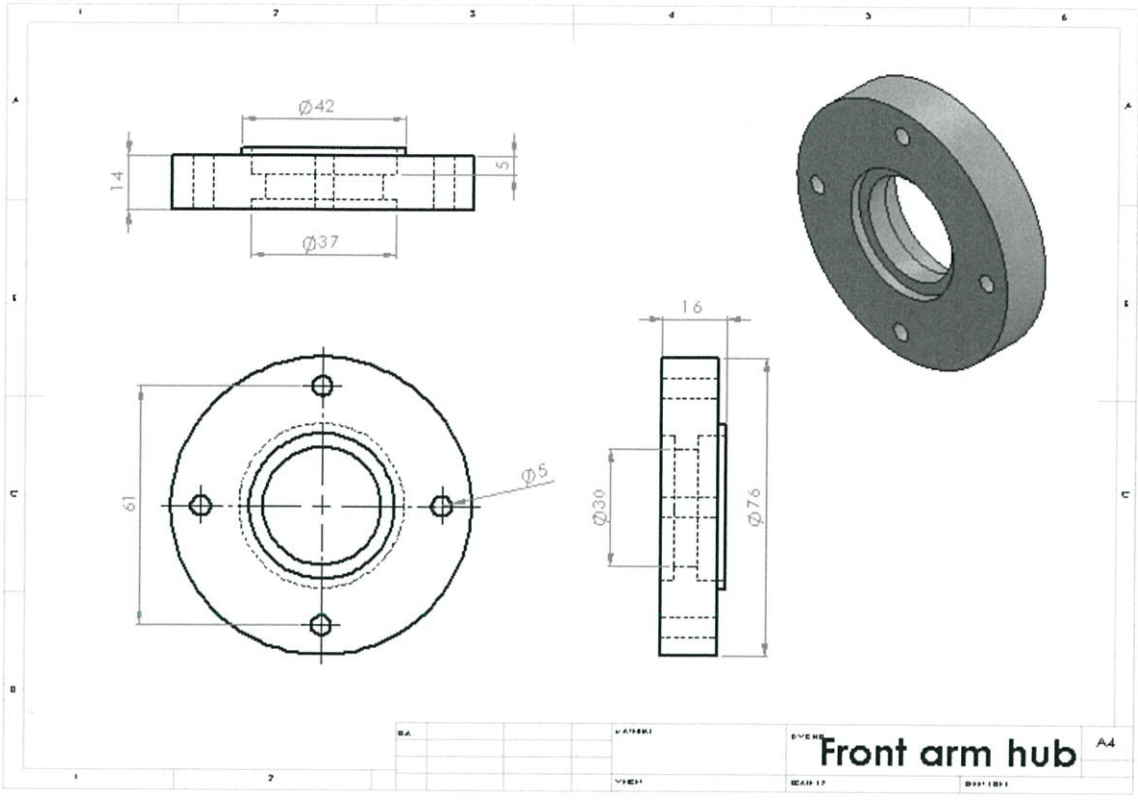

2.60
Ø5
Ø44
95.01
Sprocket2
A4

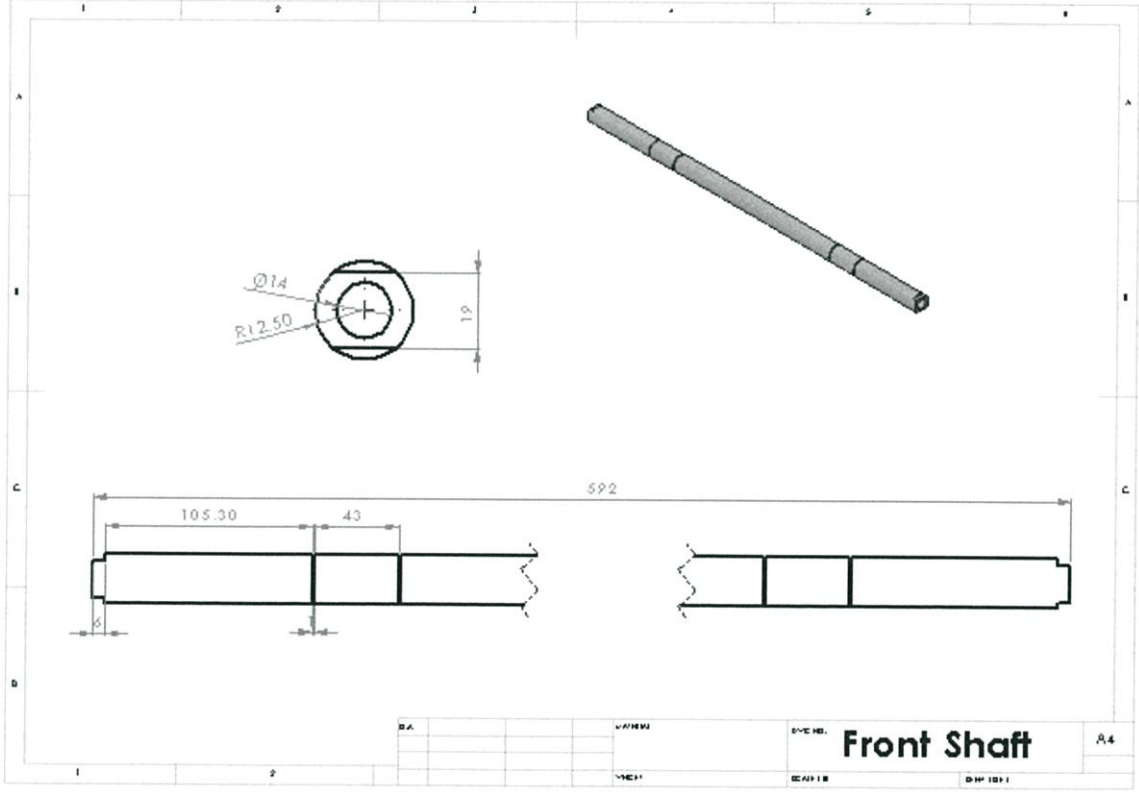

29.80
38.05
Ø5
Ø25
Ø11
Sprocket3
A4

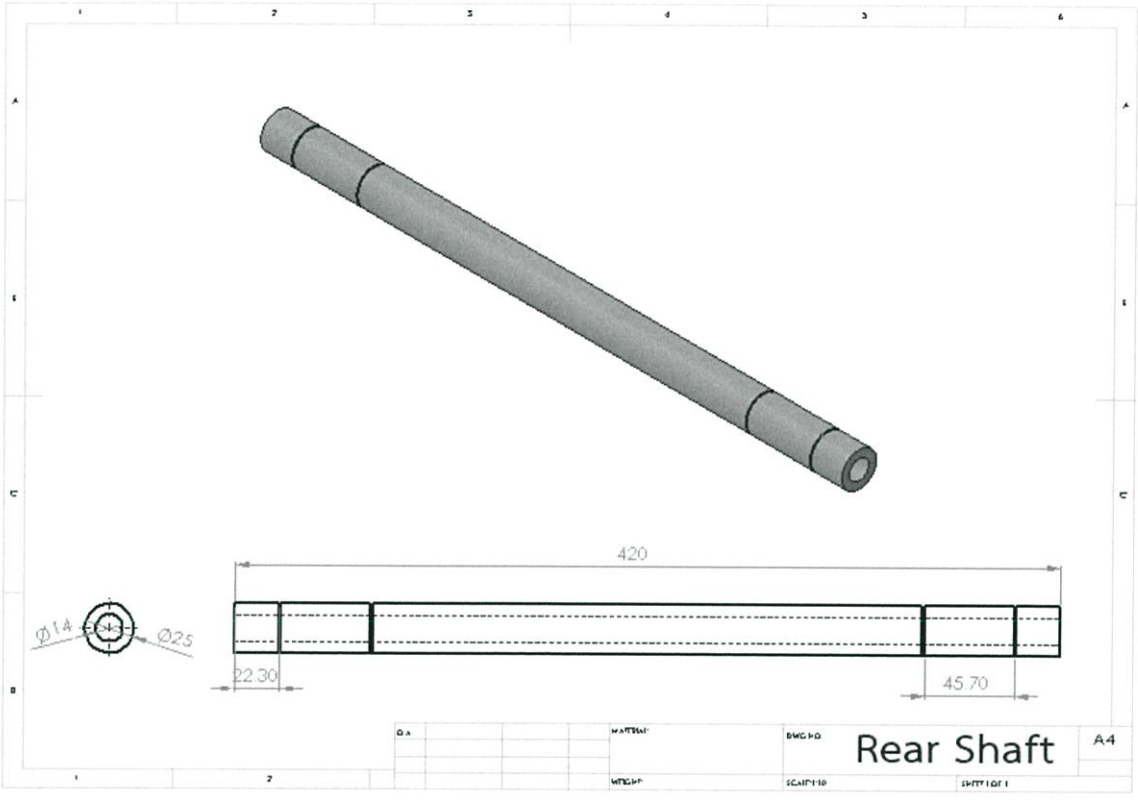

4
61
Ø42
Ø5
92.92
Sprocket4
A4

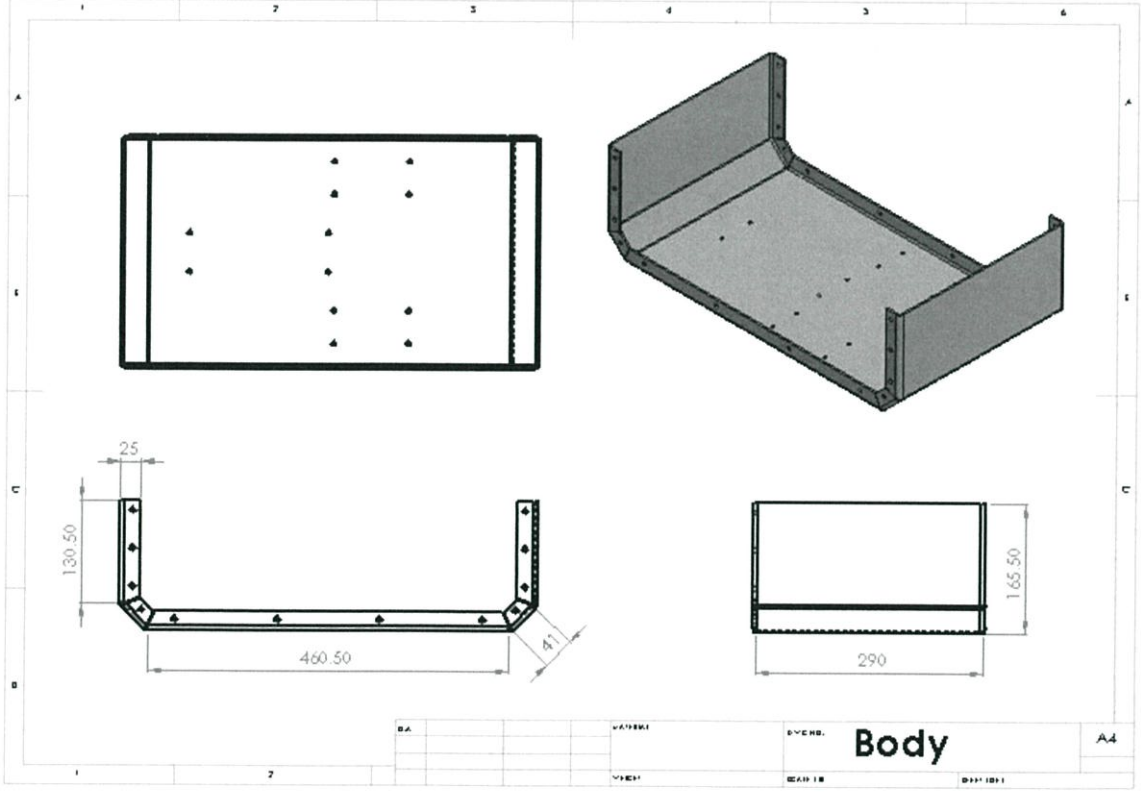

12
Ø37
Ø32
Ø5
Ø75
Shaft Bearing
A4

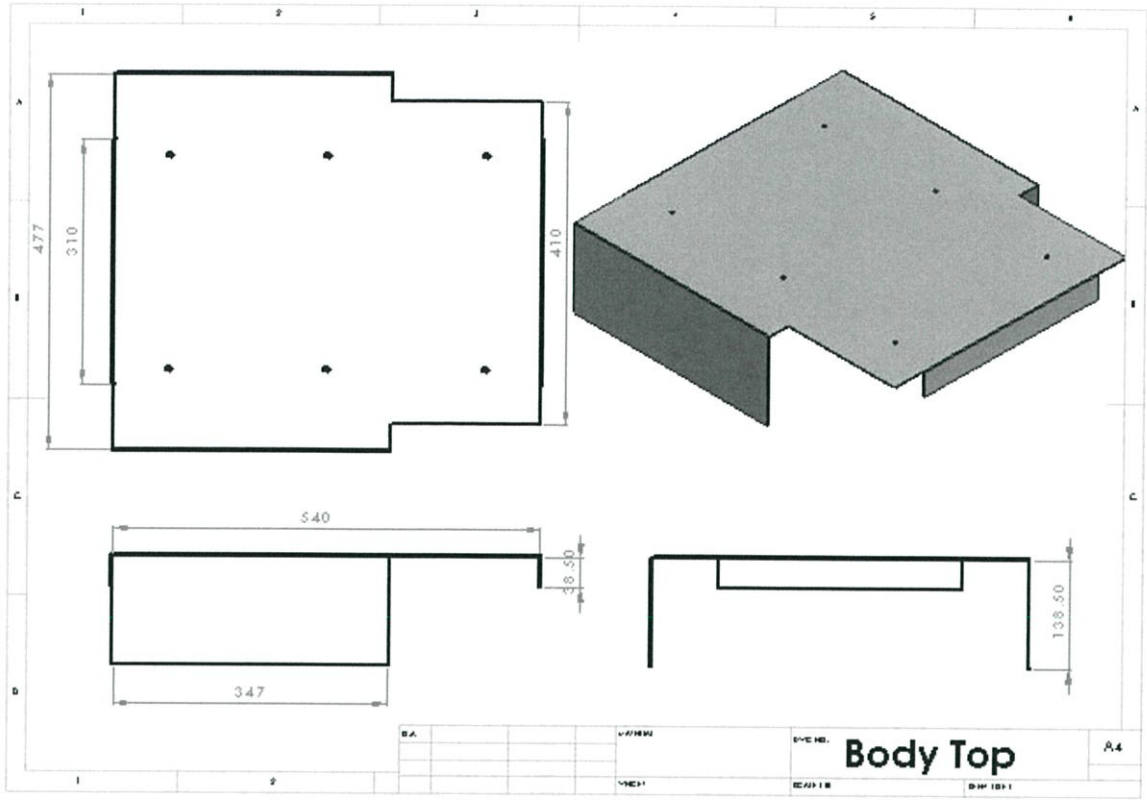

Ø89
5
3.50
Ø44
61
Ø5
Ø37
Ø32
14.20
Front Hub
A4

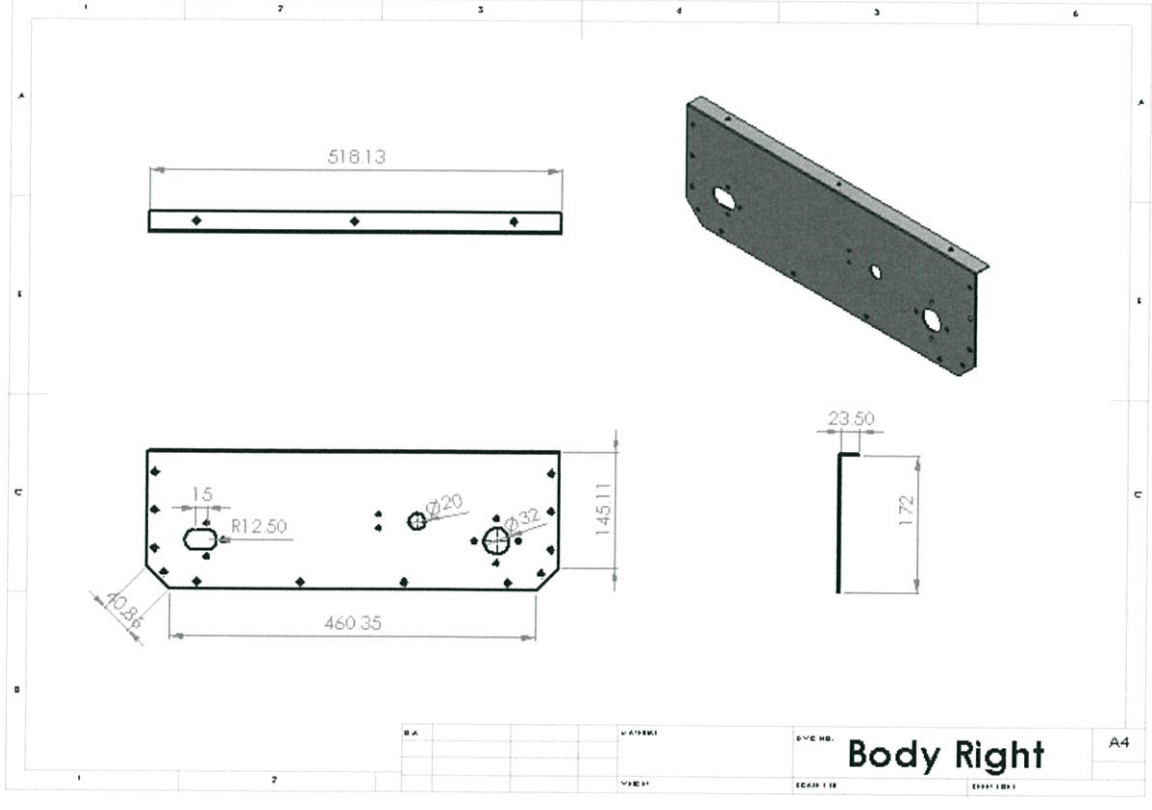

Ø42
14
5
Ø37
16
61
Ø5
Ø30
Ø76
Front arm hub
A4

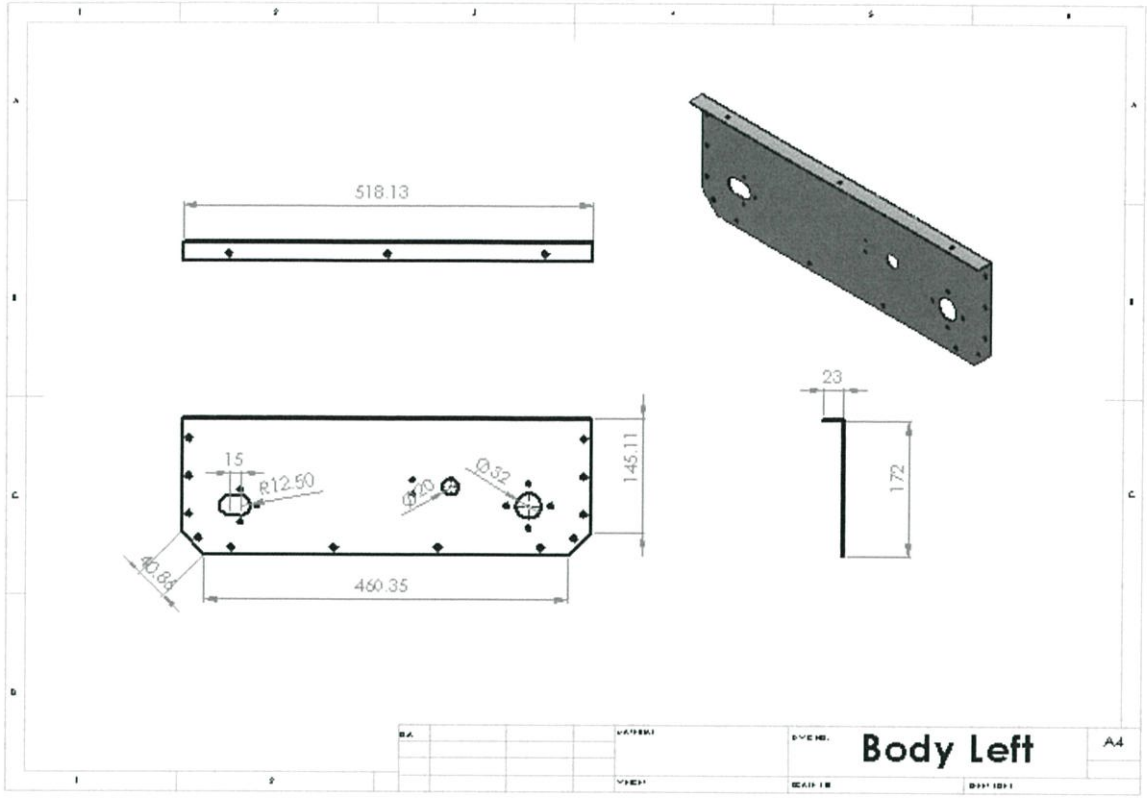

Ø14
R12.50
19
592
105.30
43
Front Shaft
A4

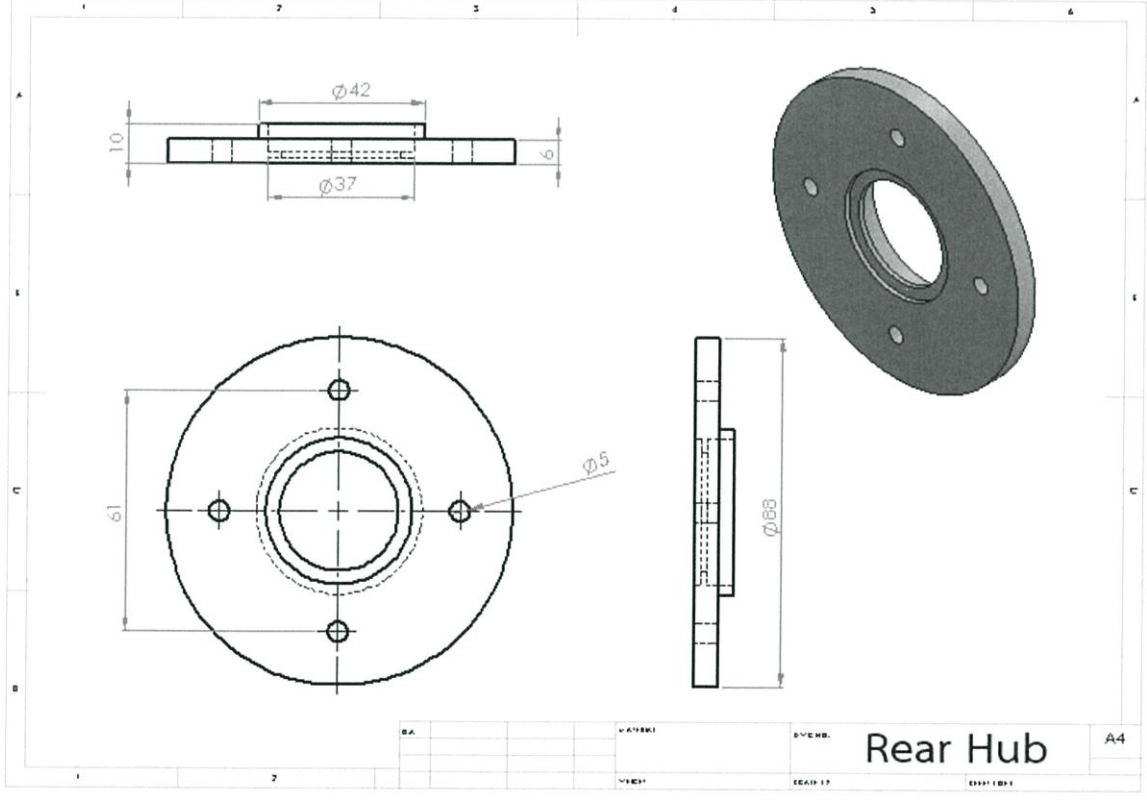

420
Ø14
Ø25
22.30
45.70
Rear Shaft
A4


25
130.50
460.50
41
165.50
290
Body
A4


477
310
410
540
38.50
347
138.50
Body Top
A4


518.13
23.50
172
1.5
R12.50
Ø20
Ø32
145.11
40.86
460.35
Body Right
A4


518.13
23
15
R12.50
Ø10
Ø32
145.11
172
40.84
460.35
Body Left
A4


Ø42
10
6
Ø37
Ø5
61
Ø68
Rear Hub
A4