



หุ่นยนต์วาดภาพขับเคลื่อนด้วยล้อ

Wheel drive drawing robot.

อัศรพล อักษรสมบัติ 6010110514

พบธรรม วราหกิจ 6110110306

โครงการวิศวกรรมคอมพิวเตอร์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยสงขลานครินทร์

ชื่อโครงการ	หุ่นยนต์วาดภาพขับเคลื่อนด้วยล้อ		
ผู้จัดทำ	นายอัศรพล อักษรสมบัติ	รหัสนักศึกษา	6010110514
	นายพชรธรรม วราห์กิจ	รหัสนักศึกษา	6110110514
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ปีการศึกษา	2564		

อาจารย์ที่ปรึกษาโครงการ

คณะกรรมการสอบ

.....

(ผศ.ดำรงค์ เกล้าดี)

.....

(รศ.ดร.มนตรี กาญจนเดชะ)



.....

(ผศ.ดร.วชิรินทร์ แก้วอภัยชัย)

โครงการนี้เป็นส่วนหนึ่งของรายวิชาโครงการวิศวกรรมคอมพิวเตอร์ 1 และ 2 ตามหลักสูตร
ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยสงขลานครินทร์

.....

(ผศ.ดร.นิคม สุวรรณวร)

หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์

หนังสือรับรองความเป็นเอกลักษณ์

ข้าพเจ้าผู้ลงนามท้ายนี้ ขอรับรองว่ารายงานฉบับนี้เป็นรายงานที่มีความเป็นเอกลักษณ์ โดยที่ข้าพเจ้ามิได้มีการคัดลอกมาจากที่ใด เนื้อหาในรายงานทั้งหมดถูกรวบรวมจากการพัฒนาในขั้นตอนต่าง ๆ ของการจัดทำโครงการ หากมีส่วนใดที่จำเป็นต้องนำข้อความจากผลงานของบุคคลอื่นใดที่ไม่ใช่ตัวข้าพเจ้า ข้าพเจ้าได้ทำการอ้างอิงถึงเอกสารเหล่านั้นไว้อย่างเหมาะสม และขอรับรองว่ารายงานฉบับนี้ไม่เคยเสนอต่อสถาบันใดมาก่อน

ผู้จัดทำ

.....

(นายอัศรพล อักษรสมบัติ)

.....

(นายพบธรรม วราหกิจ)



ชื่อโครงการ	หุ่นยนต์วาดภาพขับเคลื่อนด้วยล้อ		
ผู้จัดทำ	นายอัศรพล อักษรสมบัติ	รหัสนักศึกษา	6010110514
	นายพชรธรรม วราห์กิจ	รหัสนักศึกษา	6110110306
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ปีการศึกษา	2564		

บทคัดย่อ

โครงการฉบับนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาหุ่นยนต์วาดภาพ 2 มิติโดยการขับเคลื่อนด้วยล้อ โดยแบ่งออกเป็น 3 ส่วน คือ ระบบกลไก ระบบอิเล็กทรอนิกส์และซอฟต์แวร์และอัลกอริทึมในการทำงานโดยใช้ระบบพิกัดฉากในการสั่งงานวาดภาพและรองรับข้อมูลพิกัดที่แปลงมาจากข้อมูลในรูปแบบ Gcode โดยใช้บอร์ดไมโครคอนโทรลเลอร์ Arduino ในการควบคุมการทำงานของระบบพร้อมทั้งสามารถจำลองผลการวาดภาพก่อนวาดจริงผ่านเว็บไซต์แอปพลิเคชันที่พัฒนาขึ้น

คำสำคัญ ระบบขับเคลื่อนด้วยล้อ หุ่นยนต์วาดภาพสองมิติ การจำลองการวาดภาพจาก Gcode

Project	2D Drawing Robotic Arm Driven by Wheels.		
Author	Mr.Akarapon Oaksonsombat	Student ID	6010110514
	Mr.Phobthum Varahakij	Student ID	6110110306
Major Program	Computer Engineering		
Academic Year	2021		

Abstract

This project aims to design and develop a wheel drive drawing robot. It is divided into 3 parts which are mechanical system, electronic system and software and working algorithm by using scene coordinate system to order drawing and support coordinate data converted from data in Gcode format using a microcontroller board. Arduino to control the operation of the system and can simulate the drawing results before the actual drawing through the web application developed.

Keyword Wheel Drive, Wheel drive drawing Robot, Gcode Drawing Simulator

กิตติกรรมประกาศ

โครงการเรื่อง “แขนกลวาดภาพ 2 มิติโดยการขับเคลื่อนด้วยล้อ” สามารถดำเนินการบรรลุตามเป้าหมายในทุกขั้นตอน ตั้งแต่การศึกษาหาข้อมูล การดำเนินการต่าง ๆ รวมไปถึงการจัดทำรายงานฉบับนี้ให้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความดูแลเอาใจใส่และความเอื้อเฟื้อจาก ผศ.ดำรงค์ เคล้าดี อาจารย์ที่ปรึกษาโครงการ ซึ่งได้ให้คำแนะนำและข้อคิดเห็นต่าง ๆ รวมถึงแก้ไขปัญหาที่เกิดขึ้นในระหว่างการทำงาน และต้องขอขอบคุณไปถึง รศ.ดร.มนตรี กาญจนเดชะ, ดร.วชิรินทร์ แก้วอภิชัย, ผศ.ดร.ทวิศักดิ์ เรืองพีระกุล คณะกรรมการสอบที่ให้ข้อเสนอแนะเพื่อให้ผู้จัดทำนำมาปรับปรุงแก้ไข

ขอขอบคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ที่ช่วยสนับสนุนเงินเพื่อใช้ในการทำโครงการ

อัศรพล อักษรสมบัติ

พชรธรรม วราหกิจ



สารบัญ

บทคัดย่อ	ค
Abstract	ง
กิตติกรรมประกาศ	จ
สารบัญ	ฉ
รายการภาพประกอบ	ณ
รายการตาราง	ญ
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์.....	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.4 ขอบเขตของโครงการ.....	2
1.5 ขั้นตอนการดำเนินงาน.....	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 แขนหุ่นยนต์	3
2.1.1 Degree of Freedom	3
2.1.2 Working Space	3
2.1.3 Link	3
2.1.4 Base	4
2.1.5 Joint.....	4
2.1.6 Wrist.....	4
2.1.7 End Effector	4
2.2 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อ [1].....	8
2.2.1 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบคว่ำ	8

2.2.2 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบหางย	8
2.2.3 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยยานยนต์	9
2.2.4 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแนวตั้ง	9
2.3 มอเตอร์	10
2.3.1 Stepping Motor	10
2.3.2 RC Servo Motor	12
2.4 Stepping Motor Driver	13
2.5 Tracking Sensor	13
2.6 สมการที่เกี่ยวข้องกับการทำงานของมอเตอร์	14
2.6.1 Linear movement equation	14
2.6.2 Rotate movement equation	15
บทที่ 3 วิธีการดำเนินงาน	16
3.1 ระบบทางกลไก	16
3.1.1 ชุดทดสอบกลไกเบื้องต้น	17
3.2 ระบบอิเล็กทรอนิกส์	17
3.3 ซอฟต์แวร์และอัลกอริทึม	19
3.3.1 การคำนวณจำนวนสเต็ปต่อรอบในการคำนวณการหมุนของมอเตอร์	19
3.3.1 ลักษณะการหมุนของแผ่นเรียบบนล้อ	20
บทที่ 4 ผลและการวิเคราะห์ผล	21
4.1 ระบบทางกลไก	21
4.1.1 ชุดทดสอบกลไกเบื้องต้นรุ่นปรับปรุง	22
4.1.2 กลไกการทดเพื่อเพิ่มความละเอียด	22
4.1.3 กลไกการเติมน้ำหนัก	23
4.2 ระบบอิเล็กทรอนิกส์	23
4.2.1 กล่องควบคุม	23

4.2.2	กล่องจ่ายพลังงานไฟฟ้าและควบคุมระบบขับเคลื่อน (Power Module).....	24
4.2.3	ชุดสายเชื่อมต่อระหว่างอุปกรณ์.....	25
4.2.4	อุปกรณ์ทดสอบสายเชื่อมต่อ	25
4.3	ซอฟต์แวร์และอัลกอริทึม.....	26
4.3.1	ฟังก์ชันพื้นฐานในการเคลื่อนที่.....	26
4.3.2	ฟังก์ชันการควบคุมการทำงานของกล่องควบคุม	30
4.3.3	การแปลงข้อมูลภาพวาดเป็นชุดข้อมูลรูปแบบ G-Code	37
4.3.4	ซอฟต์แวร์แปลง GCode เป็นชุดข้อมูลพิกัดการวาดอย่างง่าย	38
4.3.5	ซอฟต์แวร์สำหรับการจำลองการวาดภาพ.....	38
4.4	ผลการทดสอบ	39
4.4.1	ผลการแปลงรูปภาพเป็น GCode	39
4.4.2	ผลการแปลงข้อมูลในรูปแบบ GCode เป็นชุดข้อมูลพิกัดการวาดอย่างง่าย	40
4.4.3	ผลการจำลองการวาดภาพ	41
4.4.4	ผลการวาดภาพ	41
บทที่ 5	ข้อเสนอแนะ	42
บรรณานุกรม		43

รายการภาพประกอบ

รูปที่ 2-1 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบคว่ำ	8
รูปที่ 2-2 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบหงาย	8
รูปที่ 2-3 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยยานยนต์.....	9
รูปที่ 2-4 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแนวแนวดิ่ง	9
รูปที่ 2-5 ทิศทางการหมุนของ Stepping Motor	10
รูปที่ 2-6 ทิศทางการหมุนของ RC Servo Motor	12
รูปที่ 2-7 Stepping Motor Driver	13
รูปที่ 2-8 Tracking Sensor	13
รูปที่ 4-1 ภาพรวมของระบบทางกลไกของโครงงาน	16
รูปที่ 4-2 ชุดทดสอบกลไกเบื้องต้น	17
รูปที่ 4-7 ภาพรวมระบบอิเล็กทรอนิกส์	18
รูปที่ 4-9 ภาพรวมกล่องจ่ายไฟฟ้าและควบคุมระบบขับเคลื่อน	18
รูปที่ 4-13 ลักษณะการหมุนของแผ่นเรียบบนล้อ	20
รูปที่ 4-3 ภาพถ่ายชุดทดสอบกลไกเบื้องต้น	21
รูปที่ 4-4 ภาพถ่ายชุดทดสอบกลไกเบื้องต้นรุ่นปรับปรุง	22
รูปที่ 4-5 กลไกการทดเพื่อออกแบบสำหรับการใช้งานในล้อขับเคลื่อน	22
รูปที่ 4-6 กลไกการเติมน้ำหมึก.....	23
รูปที่ 4-8 ภาพถ่ายกล่องควบคุม	23
รูปที่ 4-10 ภาพถ่ายกล่องพลังงานไฟฟ้าและควบคุมระบบขับเคลื่อน (Power Module).....	24
รูปที่ 4-11 ชุดสายเชื่อมต่อระหว่างอุปกรณ์	25
รูปที่ 4-12 ภาพถ่ายอุปกรณ์ทดสอบสายเชื่อมต่อ	25
รูปที่ 4-14 ทิศทางการเคลื่อนที่ของชิ้นงานไปในทิศ +Y	26
รูปที่ 4-15 คำสั่งควบคุมการเคลื่อนที่ของชิ้นงานไปในทิศ +Y	26
รูปที่ 4-16 ทิศทางการเคลื่อนที่ของชิ้นงานไปในทิศ -Y	27

รูปที่ 4-17 คำสั่งควบคุมการเคลื่อนที่ชิ้นงานไปในทิศ -Y	27
รูปที่ 4-18 ทิศทางการเคลื่อนที่ของชิ้นงานในทิศทางหมุนตามเข็มนาฬิกา.....	28
รูปที่ 4-19 คำสั่งควบคุมทิศทางการเคลื่อนที่ของชิ้นงานในทิศทางหมุนตามเข็มนาฬิกา	28
รูปที่ 4-20 ทิศทางการเคลื่อนที่ของชิ้นงานในทิศทางหมุนทวนเข็มนาฬิกา	29
รูปที่ 4-21 คำสั่งควบคุมทิศทางการเคลื่อนที่ของชิ้นงานในทิศทางหมุนทวนเข็มนาฬิกา.....	29
รูปที่ 4-22 คำสั่งที่ใช้ในการควบคุมการแสดงผลของจอ LCD แบบ I2C	30
รูปที่ 4-23 ผลลัพธ์ของการทำงานตามคำสั่งใน รูปที่ 4-22	30
รูปที่ 4-24 คำสั่งที่ใช้ในการรับข้อมูลจากแป้นตัวเลขแล้วแสดงผลในจอ LCD แบบ I2C	31
รูปที่ 4-25 ผลลัพธ์ของการทำงานตามคำสั่งใน รูปที่ 4-24	31
รูปที่ 4-26 คำสั่งที่ใช้ในการรับข้อมูลจากการป้อนจาก Analog Buttonแล้วแสดงผลในจอ LCD....	32
รูปที่ 4-27 ผลลัพธ์ของการทำงานตามคำสั่งใน รูปที่ 4-26	32
รูปที่ 4-28 คำสั่งที่ใช้ในการควบคุม RC Servo Motor ให้ไปยังองศาต่างๆ	33
รูปที่ 4-29 ภาพถ่ายการแสดงผลตามฟังก์ชันแสดงผลหน้าจอเมนู	34
รูปที่ 4-30 ภาพถ่ายการแสดงผลตามฟังก์ชันแสดงผลหน้าจอควบคุมการทำงานแบบกำหนดเอง	35
รูปที่ 4-31 ขั้นตอนที่ 1 ของการแปลงรูปภาพเป็น G-Code	37
รูปที่ 4-32 ขั้นตอนที่ 2 ของการแปลงรูปภาพเป็น G-Code	37
รูปที่ 4-33 ขั้นตอนที่ 3 ของการแปลงรูปภาพเป็น G-Code	38
รูปที่ 4-34 หน้าเว็บไซต์สำหรับการจำลองการวาดภาพ	38
รูปที่ 5-3 ผลการใช้ฟังก์ชัน Trace Bitmap... บนโปรแกรม InkScape.....	39
รูปที่ 5-4 GCode ที่ได้จากการแปลงจากรูป 4 3.....	39
รูปที่ 5-5 ผลการแปลงข้อมูลในรูปแบบ GCode เป็นชุดข้อมูลพิกัดการวาดอย่างง่าย.....	40
รูปที่ 5-6 ผลการจำลองการวาดภาพ	41
รูปที่ 5-7 ผลการวาดภาพ.....	41

รายการตาราง

ตารางที่ 2-1 ประเภทของแขนหุ่นยนต์.....	5
ตารางที่ 2-2 การควบคุม Stepping Motor ด้วยสัญญาณพัลส์แบบ Full-Step 1 Phase	10
ตารางที่ 2-3 การควบคุม Stepping Motor ด้วยสัญญาณพัลส์แบบ Full-Step 2 Phase	11
ตารางที่ 2-4 การควบคุม Stepping Motor ด้วยสัญญาณพัลส์แบบ Half-Step.....	11



บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ

หุ่นยนต์เป็นหุ่นยนต์ชนิดหนึ่งที่น่าสนใจนำมาใช้งานแทนแรงงานมนุษย์โดยมักถูกใช้ในงานที่ต้องทำซ้ำ ๆ และต้องการความแม่นยำสูง ซึ่งต้นทุนในการประดิษฐ์หุ่นยนต์นั้นมีราคาค่อนข้างสูงและเป็นที่ต้องการอย่างมากในอุตสาหกรรมการผลิตต่าง ๆ ในปัจจุบัน

หุ่นยนต์วาดภาพเป็นหนึ่งในรูปแบบของหุ่นยนต์ที่ถูกออกแบบมาให้มีความสามารถแบบเฉพาะเจาะจงในการวาดภาพและสามารถใช้ประกอบการศึกษากลไกการขับเคลื่อนของหุ่นยนต์ผ่านการแสดงออกในรูปแบบของลายเส้นที่เป็นผลลัพธ์จากการทำงานของหุ่นยนต์อีกด้วย

การใช้ระบบขับเคลื่อนด้วยล้อทำให้ชิ้นงานมีความสามารถในการเคลื่อนหรือหมุนไปตามทิศต่าง ๆ ได้ซึ่งทำให้สามารถลดต้นทุนในการประดิษฐ์หุ่นยนต์วาดภาพโดยยังคงความสามารถในการทำงานไว้ได้อย่างครบถ้วน

ผู้จัดทำจึงยกประเด็นนี้มาเป็นหัวข้อในการทำโครงงานเพื่อออกแบบและพัฒนาหุ่นยนต์วาดภาพสองมิติที่ใช้กลไกการขับเคลื่อนด้วยล้อสำหรับการใช้การศึกษาอัลกอริทึมและกลไกการทำงานของหุ่นยนต์รูปแบบนี้ในอนาคตโดยมีหลักการในการพัฒนาคือตัวชิ้นงานจะต้องมีความเรียบง่ายและใช้ต้นทุนต่ำ

1.2 วัตถุประสงค์

1. เพื่อออกแบบหุ่นยนต์วาดภาพสองมิติที่ใช้กลไกการขับเคลื่อนด้วยล้อ
2. สร้างองค์ความรู้พื้นฐานเกี่ยวกับระบบขับเคลื่อนในสองมิติด้วยล้อ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้หุ่นยนต์ขับเคลื่อนในสองมิติด้วยล้ออย่างง่ายและมีต้นทุนในการประดิษฐ์ที่ถูกลงและง่ายต่อการพัฒนาต่อยอด
2. ได้องค์ความรู้พื้นฐานเกี่ยวกับระบบขับเคลื่อนด้วยล้อที่ควบคุมด้วยไมโครคอนโทรลเลอร์
3. ผลงานที่ถูกสร้างและพัฒนาขึ้นนี้สามารถใช้เป็นต้นแบบในการศึกษาและพัฒนาต่อยอดเพื่อเป็นประโยชน์ในอนาคตได้

1.4 ขอบเขตของโครงการ

1. หุ่นยนต์ที่พัฒนาจะวาดชิ้นงานแบบสองมิติโดยใช้การขับเคลื่อนด้วยล้อ
2. หุ่นยนต์ที่พัฒนาจะสามารถวาดภาพในพื้นที่ขนาดอย่างน้อย 4 ตารางเซนติเมตร

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษาทฤษฎีเกี่ยวกับแขนหุ่นยนต์
2. ศึกษาการทำงานและการใช้งาน Stepping Motor
3. ศึกษาบบอิเล็กทรอนิกส์ด้านการแสดงผลและการเชื่อมต่อผู้ใช้
4. ศึกษาอัลกอริทึมและการควบคุมของระบบ
5. ออกแบบโครงสร้างทางกลไก
6. ออกแบบระบบอิเล็กทรอนิกส์
7. ออกแบบอัลกอริทึมและซอฟต์แวร์ควบคุมระบบ
8. ทดสอบการใช้งานอุปกรณ์ฮาร์ดแวร์
9. พัฒนาซอฟต์แวร์ที่ใช้ในการควบคุมการทำงานแต่ละส่วน
10. นำซอฟต์แวร์ที่พัฒนามาประกอบเข้ากันเป็นระบบ
11. ทดสอบการทำงานขั้นต้นของระบบ
12. ทดสอบการทำงานจริงของทั้งระบบ
13. ทำการปรับปรุงแก้ไขข้อผิดพลาดที่ได้จากการทดสอบ
14. จัดทำรายงานฉบับสมบูรณ์

บทที่ 2 ทฤษฎีและหลักการ

เพื่อให้การดำเนินงานบรรลุตามวัตถุประสงค์ผู้จัดทำจึงได้ทำการศึกษาทฤษฎีและหลักการต่าง ๆ เพื่อเป็นองค์ความรู้ประกอบการพัฒนาระบบซึ่งแบ่งศึกษาออกเป็นสามส่วนคือ ส่วนระบบทางกลไก ส่วนระบบอิเล็กทรอนิกส์และส่วนซอฟต์แวร์และอัลกอริทึม

ส่วนระบบทางกลไกได้ทำการศึกษากลไกของแขนหุ่นยนต์ [1] ระบบขับเคลื่อนด้วยล้อและมอเตอร์ซึ่งมีรายละเอียดการศึกษาดังนี้

2.1 แขนหุ่นยนต์

แขนหุ่นยนต์ [2] เป็นชนิดของหุ่นยนต์ที่มีการเคลื่อนไหวในแบบที่เป็นไปในลักษณะเชิงกลไกที่มักถูกใช้สำหรับหยิบ จับ เคลื่อนย้ายสิ่งของได้ในพื้นที่การทำงาน หรือ Working Space ที่กำหนดไว้เท่านั้น โดยมีองค์ประกอบดังต่อไปนี้

2.1.1 Degree of Freedom

Degree of Freedom (DoF) หรือ องศาอิสระ คือ จำนวนแกนหรือทิศทางที่แขนหุ่นยนต์สามารถเคลื่อนไหวได้ ซึ่งแบ่งการเรียกตามลักษณะความสามารถในการเคลื่อนไหว คือ การเคลื่อนไหวในแนวระนาบและไม่สามารถปรับมุมการหมุนได้ เรียกว่า “2 Degree of Freedom” การเคลื่อนไหวในสามมิติแต่ไม่สามารถปรับมุมการเอียงของปลายแขนได้ เรียกว่า “3 Degree of Freedom” และการเคลื่อนไหวในสามมิติและสามารถควบคุมการหมุนของปลายแขนได้รอบทิศทาง ซึ่งเรียกว่า “6 Degree of Freedom”

สำหรับหุ่นยนต์ที่มีมากกว่า 6 Degree of Freedom คือ หุ่นยนต์ที่สามารถเคลื่อนปลายแขนไปยังตำแหน่งใด ๆ ในสามมิติและสามารถควบคุมการหมุนของปลายแขนได้มากกว่า 1 ทิศทางเพื่อให้ได้ตำแหน่งปลายแขนอยู่ในตำแหน่งเดิม เช่น กรณีของแขนมนุษย์ทั่วไปซึ่งจะมี 7 Degree of Freedom ซึ่งสามารถขยับแขนได้โดยที่มีมืออยู่ในตำแหน่งเดิม

2.1.2 Working Space

Working Space หรือ พื้นที่การทำงาน คือ พื้นที่ที่ปลายของแขนหุ่นยนต์สามารถเคลื่อนที่ไปได้

2.1.3 Link

Link หรือ ลิงค์ คือ โครงสร้างแขนแต่ละท่อนของแขนหุ่นยนต์

2.1.4 Base

Base หรือ ฐาน คือ ลิงค์ชั้นแรกสุดที่ใช้ยึดแขนหุ่นยนต์ไว้กับพื้น

2.1.5 Joint

Joint หรือ ข้อต่อ คือ ส่วนเชื่อมต่อระหว่างแต่ละลิงค์ของแขนหุ่นยนต์แบ่งเป็น 2 ประเภท คือ

1. Active Joint

Active Joint คือ ข้อต่อที่มีต้นกำลังติดตั้งอยู่ทำให้สามารถควบคุมการเคลื่อนที่ได้

2. Passive Joint

Passive Joint คือ ข้อต่อที่ไม่มีต้นกำลังติดตั้งอยู่ทำให้ไม่สามารถควบคุมการเคลื่อนที่ได้ โดยการเคลื่อนที่ของ Passive Joint จะเป็นไปตามการเคลื่อนที่ของข้อต่ออื่น ๆ ที่เชื่อมโยงกันและแบ่งออกตามรูปแบบการเคลื่อนที่ได้ 4 รูปแบบ คือ

- Revolute Joint คือ ข้อต่อที่หมุนได้ (1 DoF)
- Prismatic Joint คือ ข้อต่อที่เคลื่อนที่เข้า-ออกเป็นแนวเส้น (1 DoF)
- Screw Joint คือ ข้อต่อที่หมุนและเคลื่อนที่เข้า-ออกสัมพันธ์กัน (1 DoF)
- Spherical Joint คือ ข้อต่อที่หมุนได้รอบ (2 DoF)

2.1.6 Wrist

Wrist คือ ข้อต่อที่ไม่ทำให้เกิดการเคลื่อนที่ในสามมิติแต่ทำให้เกิดการหมุนมักไว้เป็นข้อต่อสุดท้ายที่ปลายแขน



2.1.7 End Effector


End-Effector คือ อุปกรณ์ที่ติดตั้งไว้ที่ปลายแขนเพื่อใช้ทำงาน โดยอาจเป็นมือจับ ตัวดูด สูญญากาศ ส่วน เป็นต้น



แขนหุ่นยนต์ที่ใช้ในอุตสาหกรรมแบ่งออกเป็น 5 ประเภทตามลักษณะการเคลื่อนไหวดังตารางที่

2-1

ตารางที่ 2-1 ประเภทของแขนหุ่นยนต์

ประเภทของแขนหุ่นยนต์ [3]	ข้อดี - ข้อเสีย	การประยุกต์ใช้งาน
<p>Cartesian Robot</p> 	<p>ข้อดี</p> <p>เคลื่อนที่เป็นแนวเส้นตรงทั้ง 3 มิติ สามารถทำความเข้าใจการเคลื่อนที่ได้ง่ายและมีส่วนประกอบน้อยอีกทั้งโครงสร้างยังมั่นคงตลอดการเคลื่อนที่</p> <p>ข้อเสีย</p> <p>ต้องการพื้นที่ติดตั้งมากและบริเวณที่หุ่นยนต์เข้าไปทำงานได้จะต้องมีขนาดเล็กกว่าขนาดของตัวหุ่นยนต์ อีกทั้งไม่สามารถเข้าถึงวัตถุจากทิศทางข้างใต้ได้ และยังแกนแบบเชิงเส้นป้องกันฝุ่นและของเหลวได้ยากอีกด้วย</p>	<p>เนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้นจึงเหมาะกับงานเคลื่อนย้ายของหนัก ๆ หรือเรียกว่างาน Pick-and-Place เช่น ใช้โหลดชิ้นงานเข้าเครื่องจักร (Machine loading) ใช้จัดเก็บชิ้นงาน (Stacking) นอกจากนี้ยังสามารถใช้ในงานประกอบ (Assembly) ที่ไม่ต้องการเข้าถึงในลักษณะที่มีมุมหมุน เช่น ประกอบอุปกรณ์อิเล็กทรอนิกส์ และงาน Test ต่าง ๆ</p>
<p>Cylindrical Robot</p> 	<p>ข้อดี</p> <p>ตัวหุ่นยนต์มีส่วนประกอบไม่ซับซ้อนและมีการเคลื่อนที่สามารถเข้าใจได้ง่าย สามารถเข้าถึงเครื่องจักรที่มีการเปิด-ปิด หรือเข้าไปในบริเวณที่เป็นช่องหรือโพรงได้ง่าย (Loading) เช่น การโหลดชิ้นงานเข้าเครื่อง CNC</p>	<p>โดยทั่วไปจะใช้ในการหยิบยกชิ้นงาน หรือป้อนชิ้นงานเข้าเครื่องจักร เพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่องโพรงเล็ก ๆ ได้สะดวก</p>

ประเภทของแขนหุ่นยนต์ [3]	ข้อดี - ข้อเสีย	การประยุกต์ใช้งาน
	<p>ข้อเสีย</p> <p>มีพื้นที่ทำงานจำกัดและยังแกนแบบเชิงเส้นป้องกันฝุ่นและของเหลวได้ยากอีกด้วย</p>	
<p>Spherical Robot</p> 	<p>ข้อดี</p> <p>หุ่นยนต์มีปริมาตรการทำงานมากขึ้นเนื่องจากการ หมุนของแกนที่ 2 (ไหล) และสามารถที่จะก้มลงมาจับชิ้นงานบนพื้นได้สะดวก</p> <p>ข้อเสีย</p> <p>หุ่นยนต์มีระบบพิกัด (Coordinate) และส่วนประกอบที่ซับซ้อน ทำให้การเคลื่อนที่และระบบควบคุมมีความซับซ้อนขึ้น</p>	<p>ใช้ในงานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงานเข้าออกจากเครื่องปั๊ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)</p>

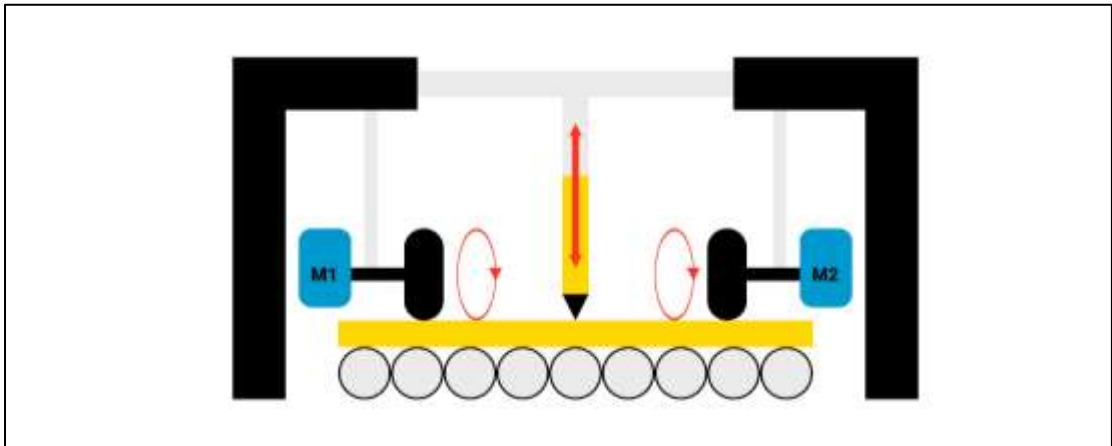
ประเภทของแขนหุ่นยนต์ [3]	ข้อดี - ข้อเสีย	การประยุกต์ใช้งาน
<p>SCARA Robot</p> 	<p>ข้อดี</p> <p>หุ่นยนต์สามารถเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วด้วยความแม่นยำสูง</p> <p>ข้อเสีย</p> <p>หุ่นยนต์มีพื้นที่ทำงานจำกัดและไม่สามารถหมุน (rotation) ในลักษณะมุมต่าง ๆ ได้อีกทั้งสามารถยกน้ำหนัก (Payload) ได้ไม่มากนัก</p>	<p>เนื่องจากการเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วจึงเหมาะกับ งานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ซึ่ง ต้องการความเร็วและการเคลื่อนที่ที่ไม่ต้องการ การหมุนมากนัก แต่จะไม่เหมาะกับงาน</p>
<p>Articulated Arm (Revolute)</p> 	<p>ข้อดี</p> <p>หุ่นยนต์สามารถเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วด้วยความแม่นยำสูง</p> <p>ข้อเสีย</p> <p>หุ่นยนต์มีพื้นที่ทำงานจำกัดและไม่สามารถหมุน (rotation) ในลักษณะมุมต่าง ๆ ได้อีกทั้งสามารถยกน้ำหนัก (Payload) ได้ไม่มากนัก</p>	<p>เนื่องจากการเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วจึงเหมาะกับ งานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ซึ่ง ต้องการความเร็วและการเคลื่อนที่ที่ไม่ต้องการ การหมุนมากนัก แต่จะไม่เหมาะกับงานประกอบชิ้นส่วนทางกล (Mechanical part) ซึ่งส่วนใหญ่ การประกอบจะอาศัยการหมุน (Rotation) ในลักษณะมุมต่าง ๆ นอกจากนี้ SCARA Robot ยังเหมาะกับงานตรวจสอบ (Inspection) งานบรรจุภัณฑ์ (Packaging)</p>

2.2 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อ [1]

ผู้จัดทำได้ทำการศึกษาาระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อใน 4 รูปแบบดังนี้

2.2.1 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบคว่ำ

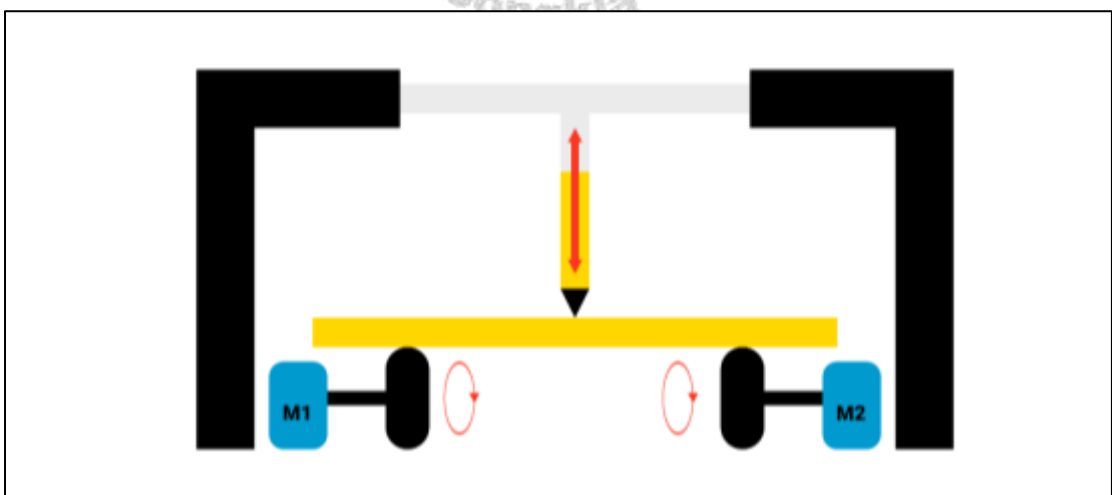
ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบคว่ำจะเป็นการนำตัวขับเคลื่อนวางทับชิ้นงานโดยจะมีกลไกการลดแรงเสียดทานอยู่ล่างชิ้นงานเพื่อให้ชิ้นงานสามารถเคลื่อนที่ได้



รูปที่ 2-1 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบคว่ำ

2.2.2 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบหงาย

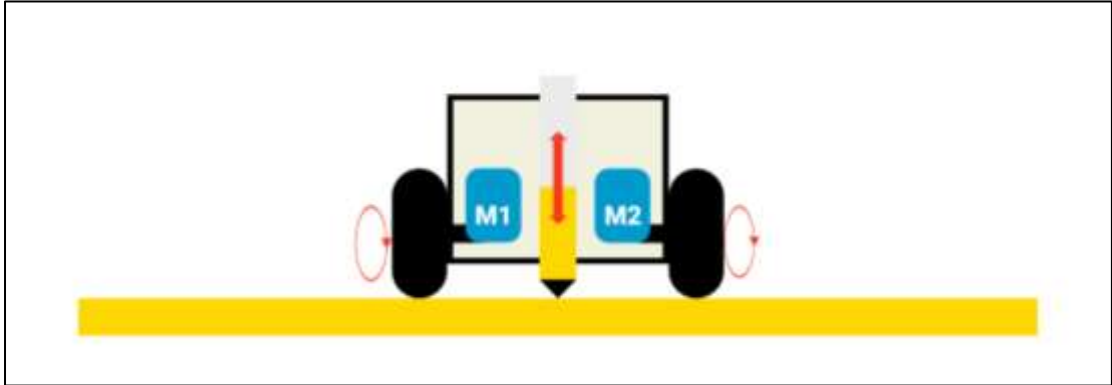
ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบหงาย เป็นการนำแผ่นเรียบวางในแนวระนาบไว้บนล้อที่ขับเคลื่อนด้วยมอเตอร์ ซึ่งอาจมีการติดตั้งล้อกลมในจุดต่าง ๆ ได้แผ่นเรียบสำหรับการช่วยพยุงแผ่นเรียบตามความเหมาะสม



รูปที่ 2-2 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแบบหงาย

2.2.3 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยยานยนต์

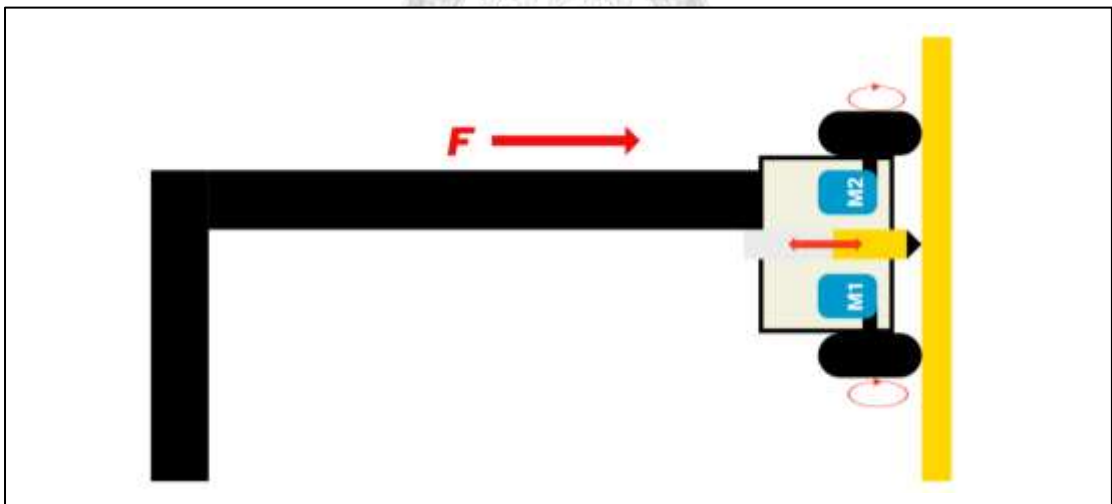
ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยยานยนต์จะอยู่ในรูปแบบคล้ายกับรถที่รวมอุปกรณ์สำหรับการทำงานต่าง ๆ ไว้ โดยชิ้นงานจะอยู่กับที่



รูปที่ 2-3 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยยานยนต์

2.2.4 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแนวตั้ง

ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแนวตั้งจะใช้หลักการเดียวกันกับระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยยานยนต์แต่ตัวรถจะถูกติดตั้งไว้ที่ปลายแขนหุ่นยนต์และถูกกดไว้ให้ติดกับผนัง โดยการทำงานของแขนหุ่นยนต์นั้นจะมีหน้าที่เพียงกดและพยุงให้รถอยู่ติดกับผนัง



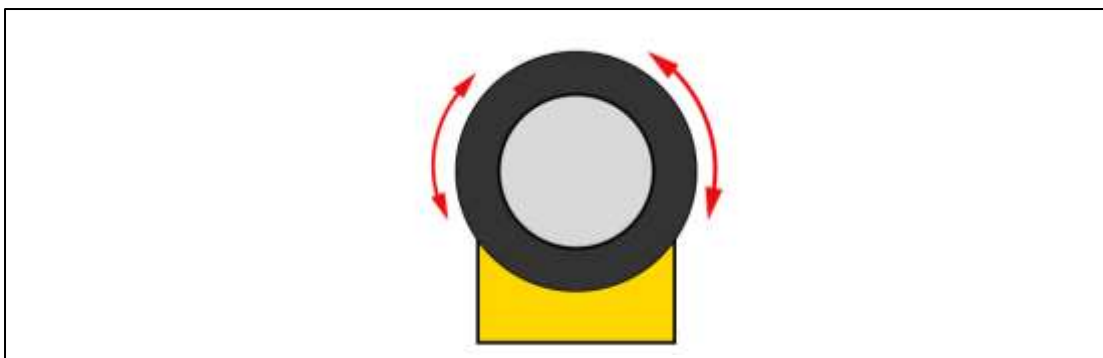
รูปที่ 2-4 ระบบวาดภาพโดยใช้กลไกการขับเคลื่อนด้วยล้อแนวตั้ง

2.3 มอเตอร์

มอเตอร์ที่ใช้ในการทำโครงงานนี้มี 2 ชนิด คือ Stepping Motor และ Servo Motor ซึ่งมีรายละเอียดการทำงานดังนี้

2.3.1 Stepping Motor

Stepping Motor [4] เป็นมอเตอร์ไฟฟ้าที่ขับเคลื่อนด้วยสัญญาณแบบพัลส์โดยจะต้องจ่ายสัญญาณพัลส์เป็นสเต็ปโดยขนาดของแต่ละสเต็ปจะขึ้นอยู่กับโครงสร้างของมอเตอร์ซึ่งสามารถหมุนได้ทั้งสองทิศทางและไม่มีการจำกัดรอบการหมุน



รูปที่ 2-5 ทิศทางการหมุนของ Stepping Motor

การควบคุม Stepping Motor แบ่งออกเป็นสามรูปแบบ คือ Full-Step 1 Phase Full-Step 2 Phase และ Half-Step โดยมีลักษณะการจ่ายสัญญาณดัง**Error! Reference source not found.** ตารางที่ 2-3 และตารางที่ 2-4 ตามลำดับ

ตารางที่ 2-2 การควบคุม Stepping Motor ด้วยสัญญาณพัลส์แบบ Full-Step 1 Phase

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

ตารางที่ 2-3 การควบคุม Stepping Motor ด้วยสัญญาณพัลส์แบบ Full-Step 2 Phase

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1

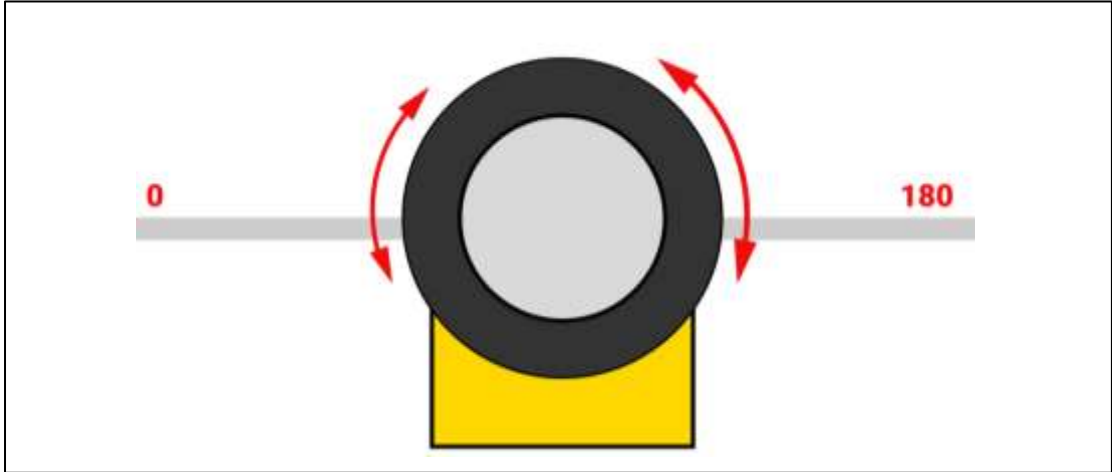
ตารางที่ 2-4 การควบคุม Stepping Motor ด้วยสัญญาณพัลส์แบบ Half-Step

สเต็ปที่	เฟสที่ 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1

ในการใช้งานจริงบอร์ดไมโครคอนโทรลเลอร์จะจ่ายไฟฟ้าได้ไม่เพียงพอในการขับเคลื่อนการทำงานของ Stepping Motor จึงมีความจำเป็นในการใช้ Stepping Motor Driver ในการช่วยขับไฟฟ้าให้เพียงพอสำหรับการทำงานของ Stepping Motor

2.3.2 RC Servo Motor

RC Servo Motor เป็นมอเตอร์ที่สามารถรับค่าองศาเพื่อให้มอเตอร์หมุนไปยังองศาที่กำหนดได้ระหว่าง 0 ถึง 180 องศาและไม่สามารถหมุนเกินขอบเขตที่จำกัดไว้ได้



รูปที่ 2-6 ทิศทางการหมุนของ RC Servo Motor

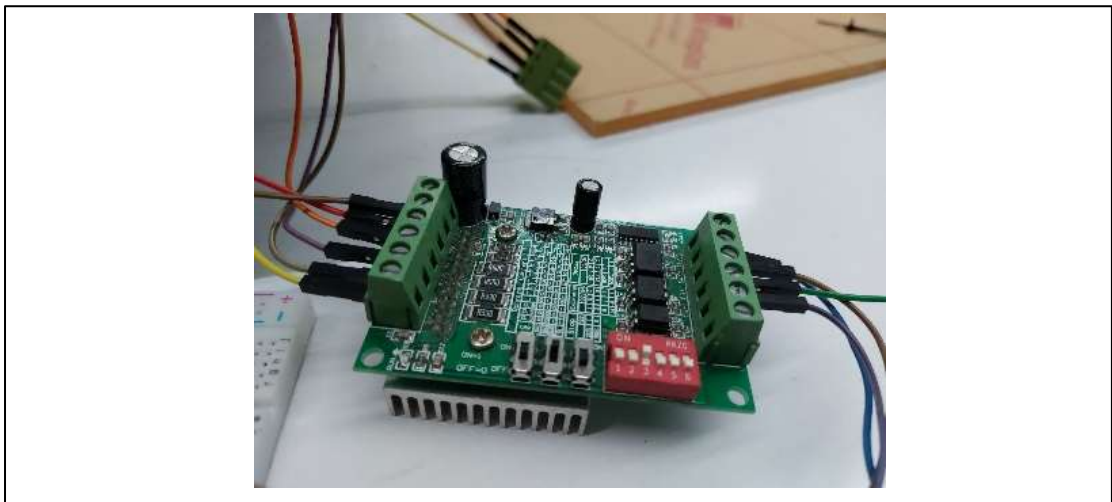
การควบคุมการทำงานของ RC Servo Motor สามารถทำได้ง่ายผ่านไลบรารี Servo.h ซึ่งมีให้ใช้งานใน Arduino IDE และจ่ายสัญญาณควบคุมโดยใช้ช่องเชื่อมต่อเดียว



ส่วนระบบอิเล็กทรอนิกส์ได้ทำการศึกษาการใช้งานอุปกรณ์สำหรับการประดิษฐ์กล่องควบคุม และอุปกรณ์การควบคุมการทำงานของหุ่นยนต์ซึ่งมีรายละเอียดการศึกษาดังนี้

2.4 Stepping Motor Driver

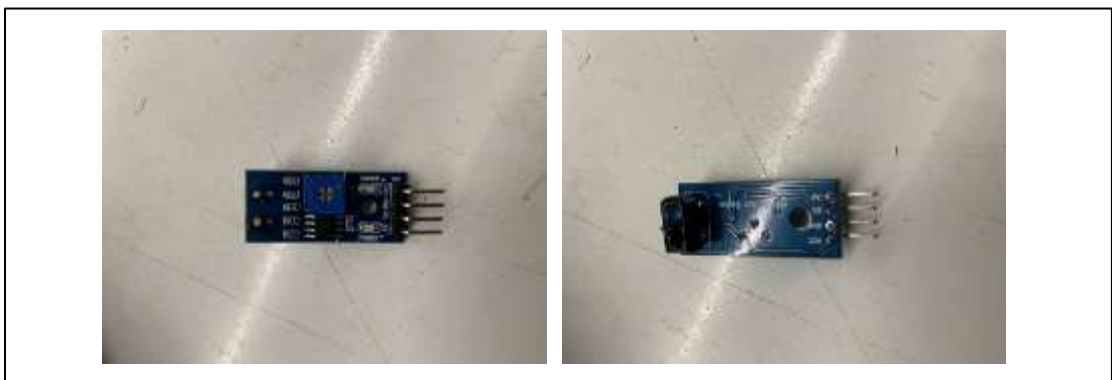
Stepping Motor Driver [5] ใช้สำหรับต่อแหล่งจ่ายไฟและรับสัญญาณควบคุมที่มาจาก ไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของมอเตอร์ ซึ่งสามารถจ่ายไฟฟ้าไปยังมอเตอร์ได้ มากกว่าไมโครคอนโทรลเลอร์



รูปที่ 2-7 Stepping Motor Driver

2.5 Tracking Sensor

โมดูลอ่านค่าสะท้อนกลับของแสง ใช้ไฟ 3.3-5V เหมาะสำหรับใช้กับบอร์ด Arduino โดยสามารถให้เอาต์พุตได้ 2 รูปแบบคือ แบบดิจิตอลสามารถปรับค่าที่ต้องการได้ เมื่อค่าที่อ่านได้ถึงระดับที่ต้องการโมดูลจะส่งออกค่าลอจิก 1 แต่ถ้าหากยังไม่ถึงระดับโมดูลจะส่งค่าลอจิก 0 และแบบอนาล็อกสามารถอ่านค่าได้เป็นตัวเลขระหว่าง 0-1023 ซึ่งอยู่ในรูปแบบของแรงดันไฟฟ้าในช่วง 0-5V



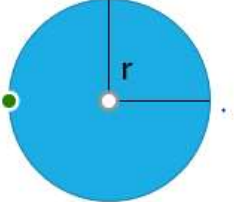
รูปที่ 2-8 Tracking Sensor

ส่วนซอฟต์แวร์และอัลกอริทึมได้ทำการศึกษาสมการสำหรับการคำนวณค่าที่เกี่ยวข้องต่าง ๆ ในการทำโครงงานซึ่งมีรายละเอียดการศึกษาดังนี้

2.6 สมการที่เกี่ยวข้องกับการทำงานของมอเตอร์

2.6.1 Linear movement equation

1. forward kinematics


$$d = f(\text{motorstep})$$
$$d = \frac{(2\pi R)}{m.r.} \times \text{motorstep}$$

R = รัศมีของเฟืองมอเตอร์ (mm)

m.r. = จำนวนstep ต่อรอบ (สตีปต่อรอบ)

d = ระยะทางที่เคลื่อนที่(mm)

2. inverse kinematics

$$\text{motorstep} = f(d)$$
$$\text{motorstep} = \frac{m.r.}{2\pi R} \times d$$

R = รัศมีของเฟืองมอเตอร์ (mm)

m.r. = จำนวนstep ต่อรอบ (สตีปต่อรอบ)

d = ระยะทางที่เคลื่อนที่(mm)

2.6.2 Rotate movement equation

1. Forward kinematics

$$\theta = f(\text{motorstep0}, \text{motorstep1})$$

$$\theta = \frac{180l}{R \times m.r.} \text{motorstep}$$

*เมื่อต้องการให้จุดศูนย์กลางเป็นจุดหมุน $\text{motorstep0} = -\text{motorstep1}$

θ = มุม (องศา)

R = รัศมีของล้อ (mm)

$m.r.$ = จำนวนstep ต่อรอบ (สตีปต่อรอบ)

l = ระยะห่างระหว่างล้อ (mm)

2. inverse kinematics

$$\text{motorstep} = f(\theta)$$

$$\text{motorstep} = \frac{R \times m.r.}{180l} \theta$$

*เมื่อต้องการให้จุดศูนย์กลางเป็นจุดหมุน $\text{motorstep0} = -\text{motorstep1}$

θ = มุม (องศา)

R = รัศมีของล้อ (mm)

$m.r.$ = จำนวนstep ต่อรอบ (สตีปต่อรอบ)

l = ระยะห่างระหว่างล้อ (mm)

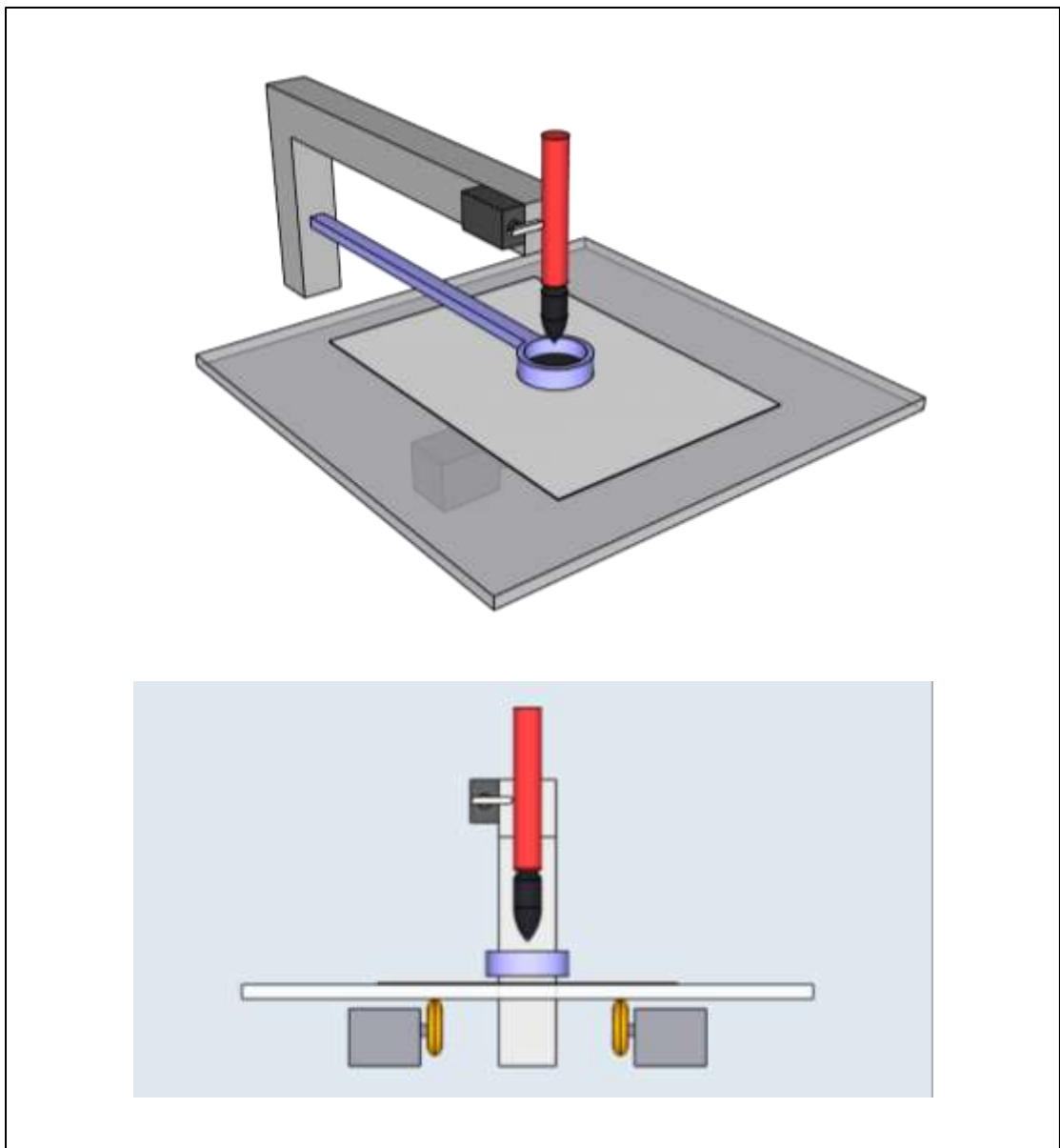
บทที่ 3 วิธีการดำเนินงาน

ในการออกแบบและพัฒนาอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ถูกแบ่งออกเป็น 3 ส่วน คือ ระบบทางกลไก ระบบอิเล็กทรอนิกส์และระบบอัลกอริทึม ซึ่งมีรายละเอียดภาพรวมในการดำเนินงานดังนี้

3.1 ระบบทางกลไก

ระบบทางกลไกประกอบด้วยแขนหุ่นยนต์ที่ขยับได้เฉพาะปลายแขนในทิศขึ้นลงซึ่งถูกควบคุมด้วย RC Servo Motor และส่วนแผ่นเรียบจะใช้เป็นวัสดุเป็นแผ่นอะคริลิกสำหรับใช้ติดตั้งงานโดยจะวางไว้บนล้อทั้งสองของอุปกรณ์ซึ่งถูกควบคุมผ่าน Stepping Motor ซึ่งมีรายละเอียดดัง

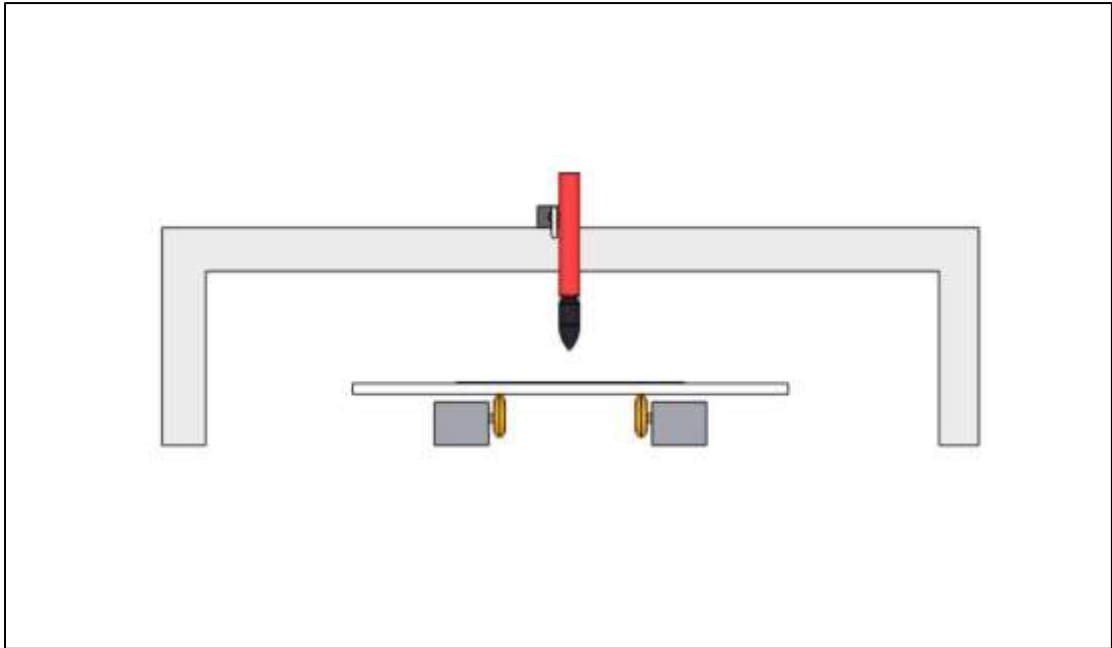
Reference source not found.



รูปที่ 3-1 ภาพรวมของระบบทางกลไกของโครงงาน

3.1.1 ชุดทดสอบกลไกเบื้องต้น

เพื่อให้การทดสอบ ออกแบบและพัฒนาเป็นไปได้อย่างรวดเร็วผู้จัดทำจึงได้จัดทำชุดทดสอบกลไกเบื้องต้นเพื่อทดสอบการทำงานเบื้องต้นของอุปกรณ์ฮาร์ดแวร์และอำนวยความสะดวกในการพัฒนาอัลกอริทึมในการควบคุมการทำงานของอุปกรณ์ในระหว่างการสั่งซื้อและประกอบชิ้นส่วนทางกลไกซึ่งอาจใช้เวลาโดยมีรายละเอียดดัง**Error! Reference source not found.**

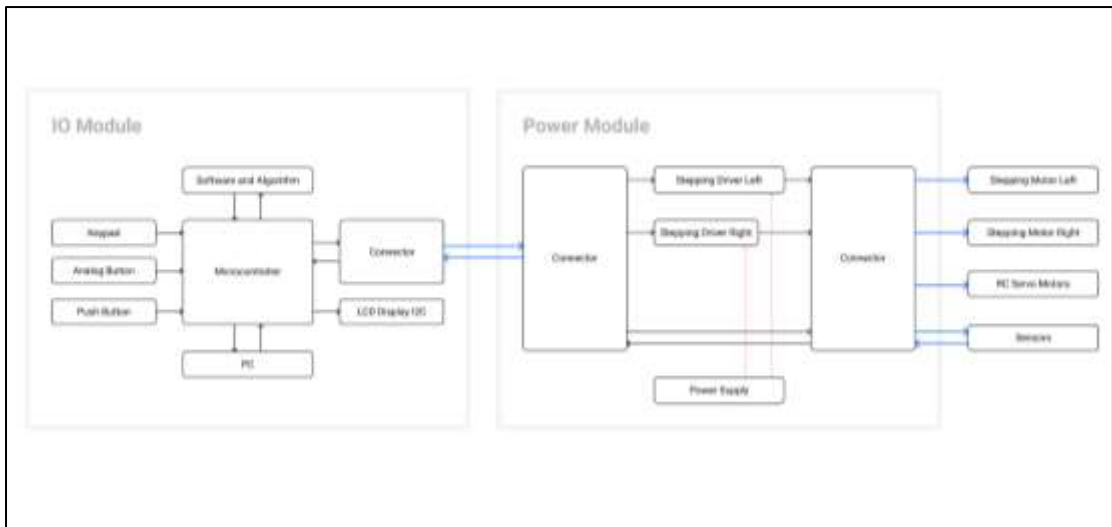


รูปที่ 3-2 ชุดทดสอบกลไกเบื้องต้น

ชุดทดสอบกลไกเบื้องต้นถูกออกแบบเพื่อใช้สำหรับการทดสอบการทำงานของ Stepping Motor และทดสอบอัลกอริทึมในการควบคุมการทำงานของ Stepping Motor ที่ใช้ในการหมุน ขยับ แผ่นอะคริลิกและทดสอบการใช้งานกล่องควบคุมการทำงานของระบบซึ่งใช้ในการรับคำสั่งจากผู้ใช้และส่งการทำงานของระบบ

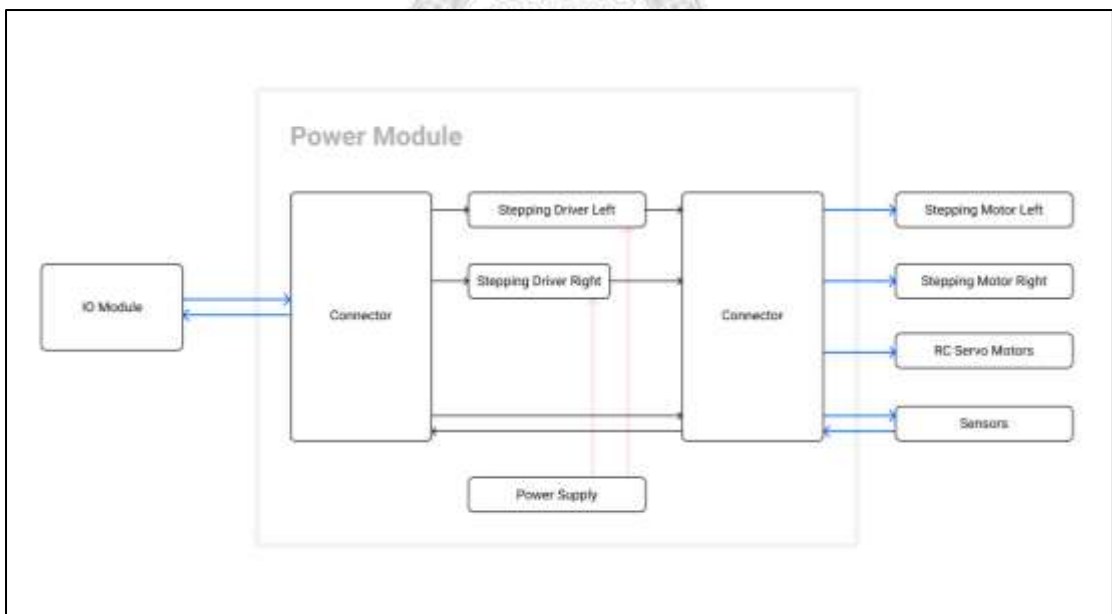
3.2 ระบบอิเล็กทรอนิกส์

เป็นส่วนฮาร์ดแวร์ที่ใช้ในการควบคุมการทำงานของระบบโดยเริ่มต้นตั้งแต่การรับข้อมูลจากผู้ใช้งานไปจนถึงการควบคุมการทำงานของมอเตอร์โดยใช้บอร์ด Arduino รุ่น ATmega 2560 เป็นตัวกลางในการควบคุมการทำงานและรับข้อมูลจากคีย์แพดและปุ่มกดแล้วจึงนำไปประมวลผลและแสดงผลในหน้าจอ LCD และส่งการทำงานของมอเตอร์ต่อไปดัง**Error! Reference source not found.**



รูปที่ 3-3 ภาพรวมระบบอิเล็กทรอนิกส์

ผู้จัดทำได้ประดิษฐ์กล่องสำหรับจ่ายพลังงานไฟฟ้าและควบคุมระบบขับเคลื่อนที่ประกอบด้วย แหล่งจ่ายพลังงานให้อุปกรณ์ทั้งหมดในระบบและอุปกรณ์สำหรับควบคุมการขับเคลื่อนของมอเตอร์ อีกทั้งยังเป็นทางผ่านในการเชื่อมต่อจากกล่องควบคุมไปยังอุปกรณ์ต่าง ๆ ดังError! Reference source not found.



รูปที่ 3-4 ภาพรวมกล่องจ่ายไฟฟ้าและควบคุมระบบขับเคลื่อน

3.3 ซอฟต์แวร์และอัลกอริทึม

3.3.1 การคำนวณจำนวนสเต็ปต่อรอบในการคำนวณการหมุนของมอเตอร์

จากการที่เลือกใช้มอเตอร์แบบฮาล์ฟสเต็ป และมีการทดรอบแบบ 1 ต่อ 2 ทำให้จำนวนสเต็ปที่ต้องใช้ในการหมุนให้ครบ 1 รอบนั้นใช้ 1500 สเต็ป และการที่ล้อตัวอย่างมีขนาดเส้นผ่านศูนย์กลาง 63 มิลลิเมตร และห่างกัน 150 มิลลิเมตรจะทำให้เกิดผลดังนี้

1. การเคลื่อนที่แบบ linear

หากต้องการที่เคลื่อนที่ไป 20 มิลลิเมตรจะต้องเคลื่อนที่

$$motorsteps = \frac{distance \times 1500}{\pi d} = \frac{20 \times 1500}{\pi d} = 151 steps$$

หากหมุนไป 1000 สเต็ป จะได้ระยะทาง

$$distance = \frac{\pi d (motorsteps)}{1500} = \frac{\pi (63) (1000)}{1500} = 131 mm.$$

2. การเคลื่อนที่แบบ rotate

หากต้องการที่เคลื่อนที่ไป 20 องศาจะต้องเคลื่อนที่

$$motorsteps = \frac{\pi \times 150 \times \emptyset}{360} / (distance per step)$$

$$= \frac{\pi \times 150 \times 180}{360} / \left(\frac{\pi (63)}{1500} \right) = 198 steps$$

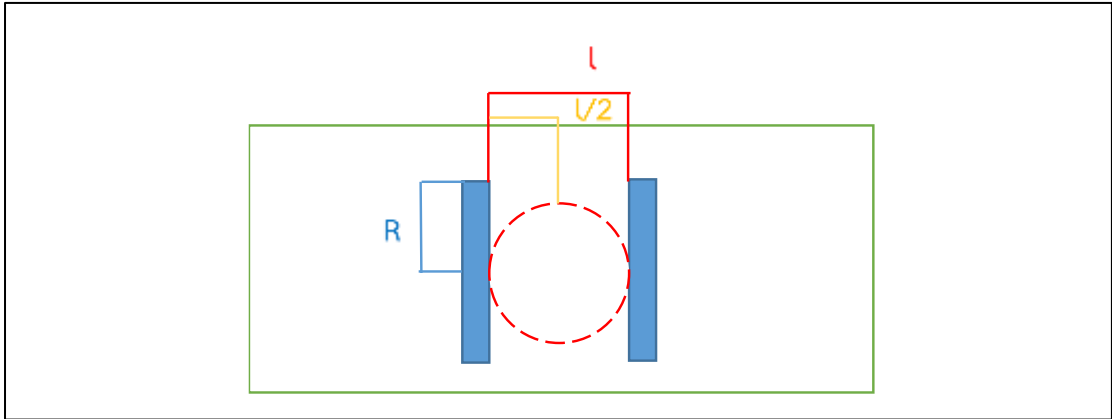
หากหมุนไป 100 สเต็ปจะได้องศา

$$\emptyset = \frac{motorsteps \times (distance per step) \times 360}{\pi \times 150}$$

$$= \frac{100 \times \left(\frac{\pi (63)}{1500} \right) \times 360}{\pi \times 150} = 10.08 \text{ องศา}$$

3.3.1 ลักษณะการหมุนของแผ่นเรียบบนล้อ

เมื่อหมุนแผ่นเรียบที่ยึดชิ้นงานไว้จะเปรียบเสมือนมีวงกลมอยู่ระหว่างล้อที่ห่างกันในระยะ L โดยจะเกิดวงกลมเสมือนที่มีรัศมี $L/2$ จะได้ว่าวงกลมด้านในมีเส้นรอบวง πL หากล้อมีรัศมี R หากหมุนล้อไป 1 รอบ แผ่นเรียบจะหมุนไปทั้งหมด $\pi L/2\pi R$ รอบ หรือ $L/2R$ รอบ



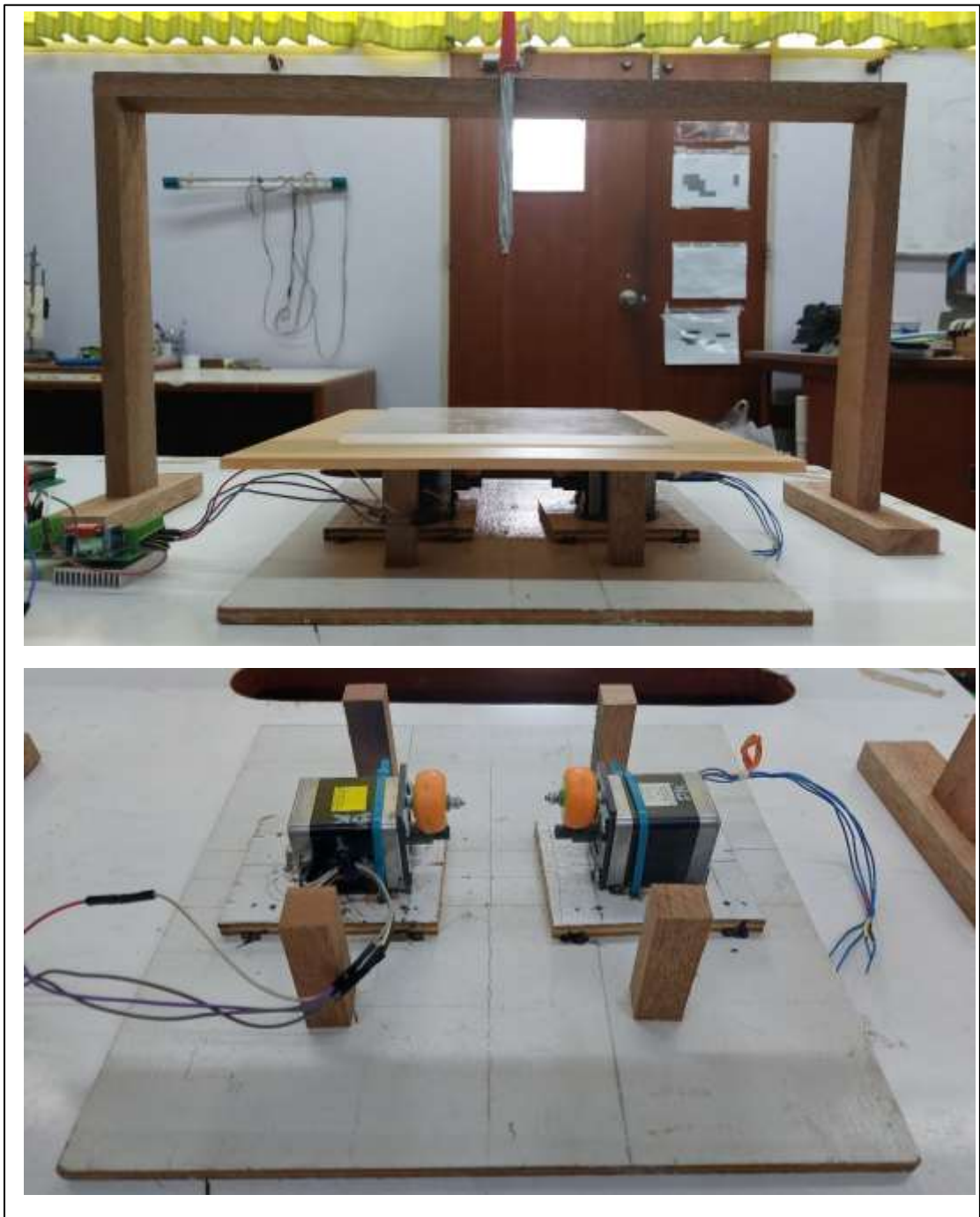
รูปที่ 3-5 ลักษณะการหมุนของแผ่นเรียบบนล้อ



บทที่ 4 ผลและการวิเคราะห์ผล

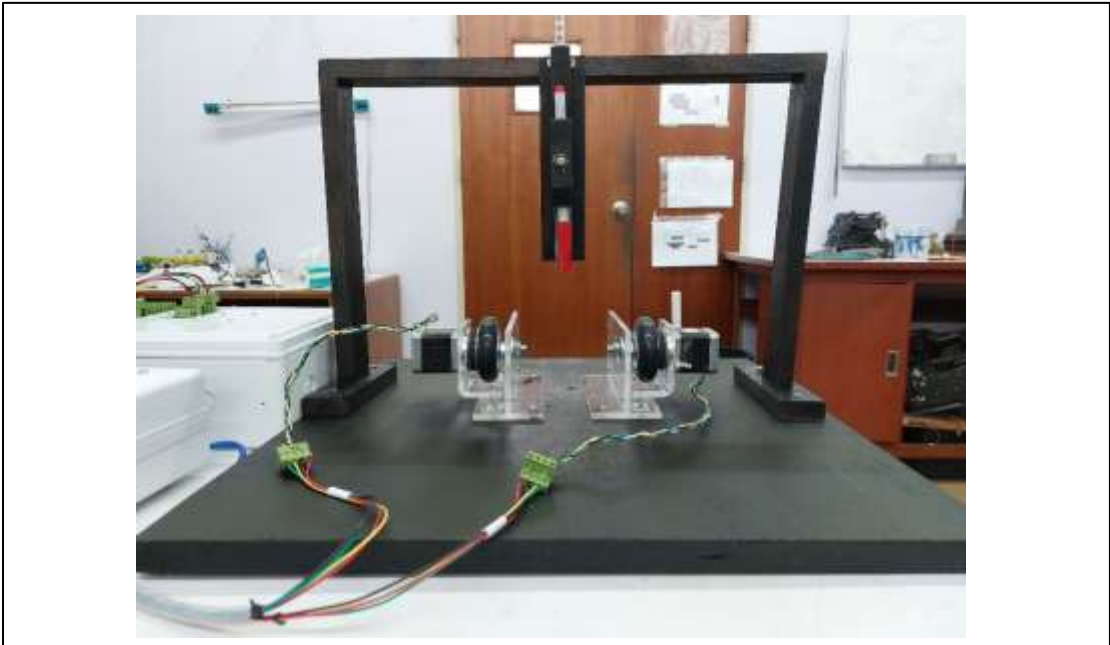
ในการออกแบบและพัฒนาอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ถูกแบ่งออกเป็น 3 ส่วน คือ ระบบทางกลไก ระบบอิเล็กทรอนิกส์และระบบอัลกอริทึม ซึ่งมีรายละเอียดผลการดำเนินงานดังนี้

4.1 ระบบทางกลไก



รูปที่ 4-1 ภาพถ่ายชุดทดสอบกลไกเบื้องต้น

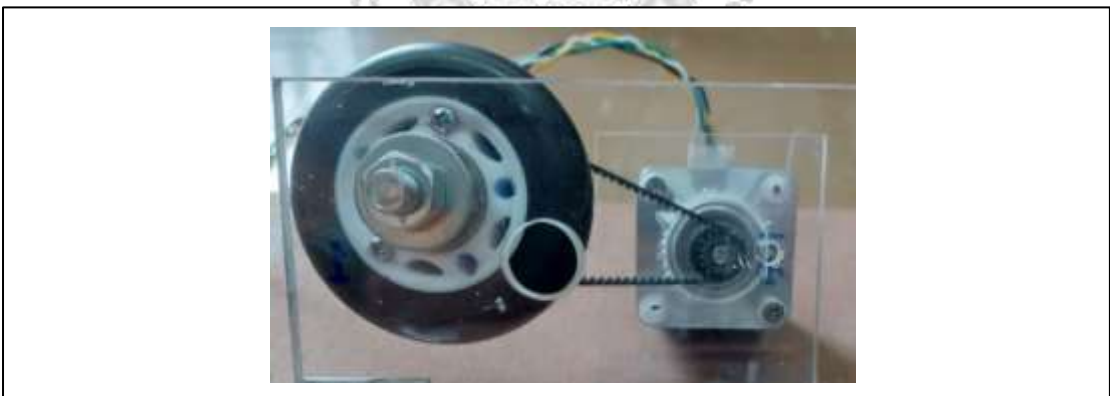
4.1.1 ชุดทดสอบกลไกเบื้องต้นปรับปรุง



รูปที่ 4-2 ภาพถ่ายชุดทดสอบกลไกเบื้องต้นปรับปรุง

4.1.2 กลไกการทดเพื่อเพิ่มความละเอียด

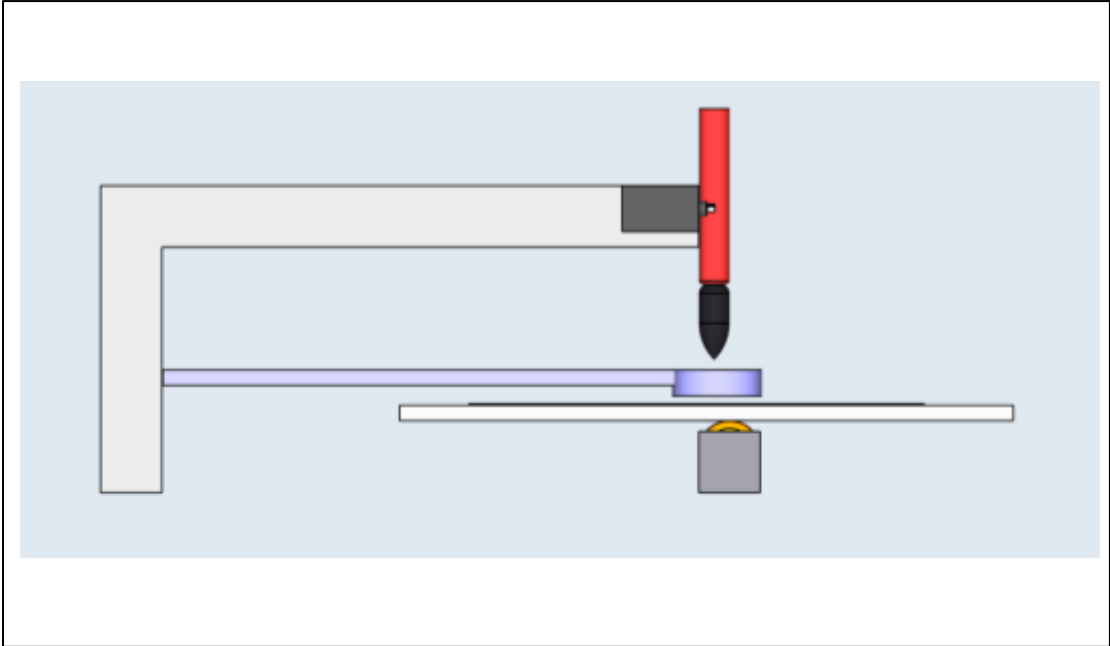
ในการออกแบบและพัฒนาเพื่อความละเอียดที่มากขึ้นได้มีการออกแบบการทดรอบ โดยการต่อพูลเลย์และสายพานและขับเคลื่อนด้วยการใช้แรงเสียดทาน พร้อมทั้งมีการเพิ่มขาที่ช่วยพยุงและติดลูกปืนเพื่อลดแรงต้านดังรูปที่ 4-3 กลไกการทดที่ออกแบบสำหรับการใช้งานในล้อขับเคลื่อน



รูปที่ 4-3 กลไกการทดที่ออกแบบสำหรับการใช้งานในล้อขับเคลื่อน

4.1.3 กลไกการเติมน้ำหมึก

สำหรับการใช้พู่กันในการวาดภาพนั้นมีความจำเป็นในการเติมน้ำหมึกแต่เนื่องจากว่าตัวกระดานนั้นมีการขยับตลอดเวลาทำให้ไม่สามารถวางที่เติมน้ำหมึกไว้ได้ผู้จัดทำจึงออกแบบระบบการเติมน้ำหมึกโดยใช้ RC Servo Motor ในการช่วยหมุนแขนถ่วงน้ำหมึกเพื่อให้สามารถจุ่มพู่กันได้โดยมีลักษณะดังรูปที่ 4-4



รูปที่ 4-4 กลไกการเติมน้ำหมึก

4.2 ระบบอิเล็กทรอนิกส์

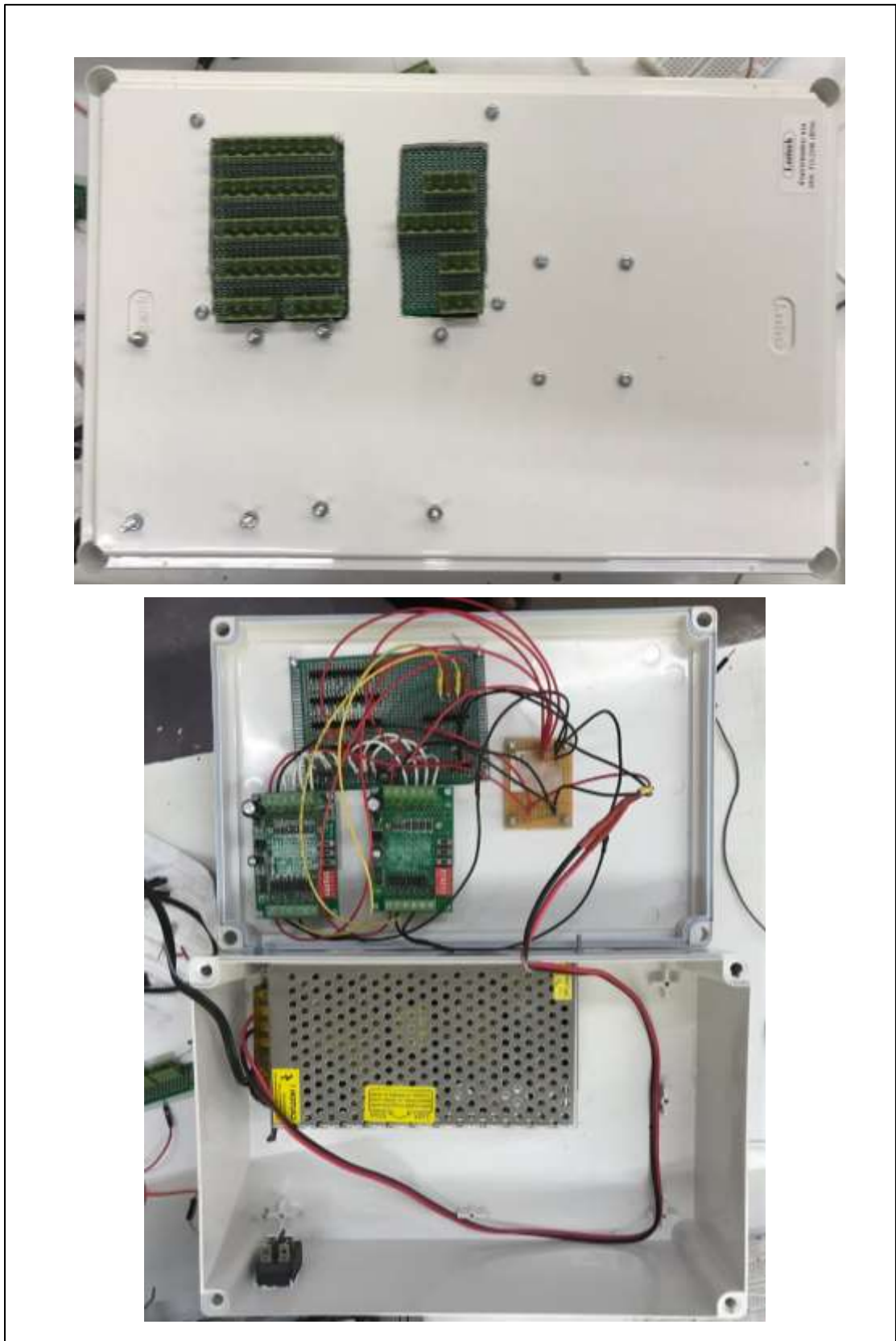
4.2.1 กล่องควบคุม

กล่องควบคุมการทำงานของระบบที่ประกอบด้วย หน้าจอแอลซีดี ปุ่มตัวเลข และปุ่มกดนำทางทิศต่าง ๆ เพื่อใช้สำหรับป้อนข้อมูลให้ซอฟต์แวร์ควบคุมการทำงานของระบบ



รูปที่ 4-5 ภาพถ่ายกล่องควบคุม

4.2.2 กล่องจ่ายพลังงานไฟฟ้าและควบคุมระบบขับเคลื่อน (Power Module)



รูปที่ 4-6 ภาพถ่ายกล่องพลังงานไฟฟ้าและควบคุมระบบขับเคลื่อน (Power Module)

4.2.3 ชุดสายเชื่อมต่อระหว่างอุปกรณ์

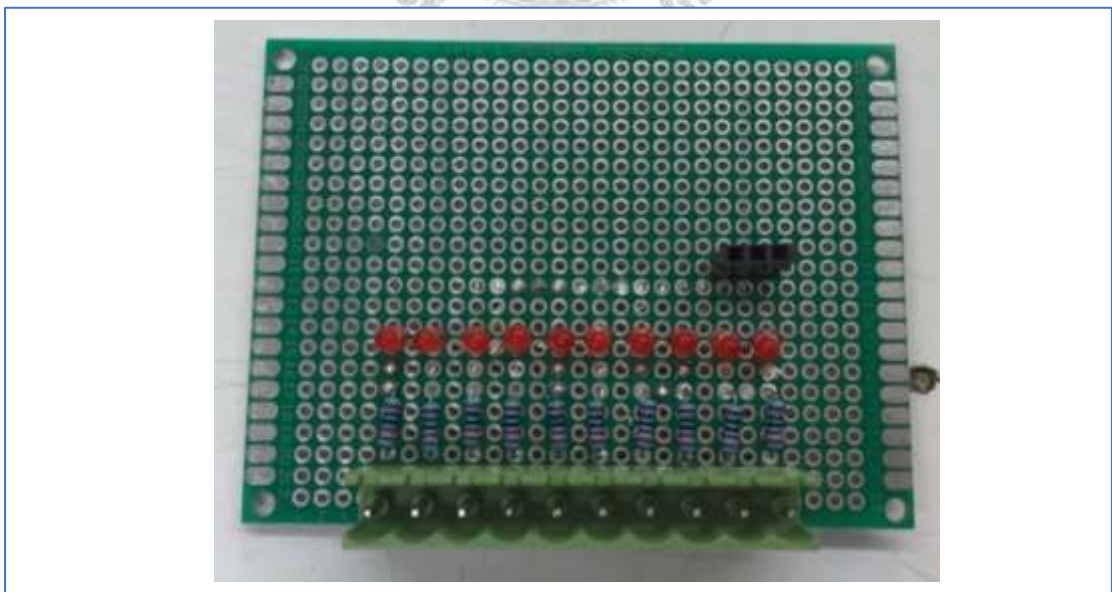
เพื่อความสะดวกในการเคลื่อนย้ายผู้จัดทำจึงทำชุดสายเชื่อมต่อที่สามารถถอดแยกได้โดยมีชนิดหัวต่อทั้งสองฝั่งสำหรับเชื่อมต่อระหว่างกล่องและใช้หัวต่อชนิดเดียวกันในการเชื่อมต่อกับอุปกรณ์แต่ละส่วนซึ่งมีลักษณะดัง



รูปที่ 4-7 ชุดสายเชื่อมต่อระหว่างอุปกรณ์

4.2.4 อุปกรณ์ทดสอบสายเชื่อมต่อ

เนื่องจากการใช้สายเชื่อมต่อนั้นมีความเสี่ยงที่สายอาจมีการชำรุดภายในได้ ผู้จัดทำจึงได้ทำอุปกรณ์สำหรับใช้ในการทดสอบการส่งสัญญาณในสายเชื่อมต่อเพื่อตรวจสอบความถูกต้องในการใช้งานโดยมีลักษณะดังรูปที่ 4 7



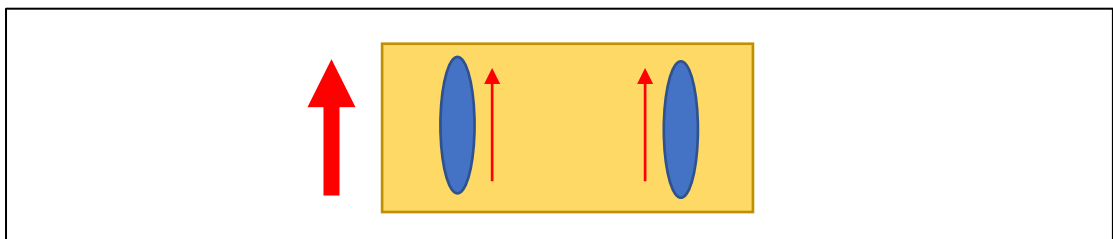
รูปที่ 4-8 ภาพถ่ายอุปกรณ์ทดสอบสายเชื่อมต่อ

4.3 ซอฟต์แวร์และอัลกอริทึม

4.3.1 ฟังก์ชันพื้นฐานในการเคลื่อนที่

ระบบที่ถูกออกแบบได้กำหนดให้ใช้แนวแกน Y เป็นแกนหลักในการเคลื่อนที่ โดยหากต้องการที่เคลื่อนที่ในแนวแกนอื่นเราจะใช้การrotateในการหมุนเพื่อเปลี่ยนแกนในการเคลื่อนที่ โดยจะได้รูปแบบพื้นฐานทั้งหมด 4 รูปแบบดังนี้

Forward มีการเคลื่อนที่ขึ้นงานไปทิศ +Y

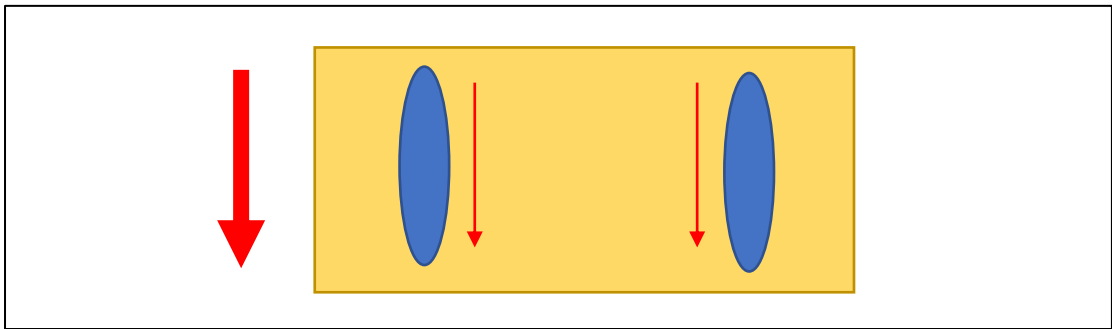


รูปที่ 4-9 ทิศทางการเคลื่อนที่ของขึ้นงานไปในทิศ +Y

```
void forward(int steps){  
    digitalWrite(d_left, HIGH);  
    digitalWrite(d_right, HIGH);  
    while(steps>0){  
        digitalWrite(c_left, HIGH);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        digitalWrite(c_left, LOW);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        steps -= 1;  
    }  
}
```

รูปที่ 4-10 คำสั่งควบคุมการเคลื่อนที่ขึ้นงานไปในทิศ +Y

Backward มีการเคลื่อนที่ชิ้นงานไปที่ -Y



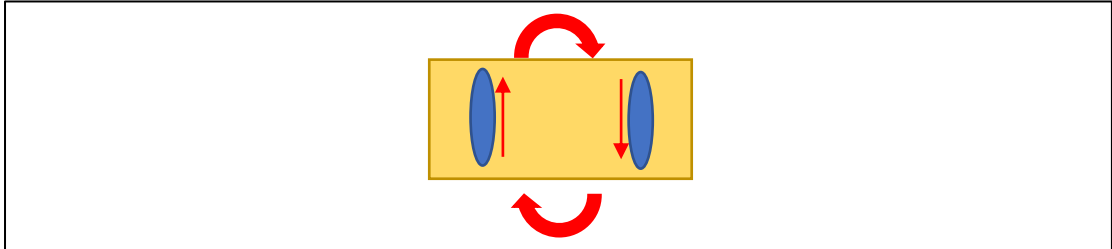
รูปที่ 4-11 ทิศทางการเคลื่อนที่ของชิ้นงานไปในทิศ -Y

```
void backward(int steps){  
    digitalWrite(d_left, LOW);  
    digitalWrite(d_right, LOW);  
    while(steps>0){  
        digitalWrite(c_left, HIGH);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        digitalWrite(c_left, LOW);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        steps -= 1;  
    }  
}
```

รูปที่ 4-12 คำสั่งควบคุมการเคลื่อนที่ชิ้นงานไปในทิศ -Y

Rotate Clockwise

เป็นการหมุนชิ้นงานตามเข็มนาฬิกา โดยล้อข้างซ้ายจะหมุนเป็นทิศ +Y และล้อข้างขวาจะหมุนไปทิศ -Y



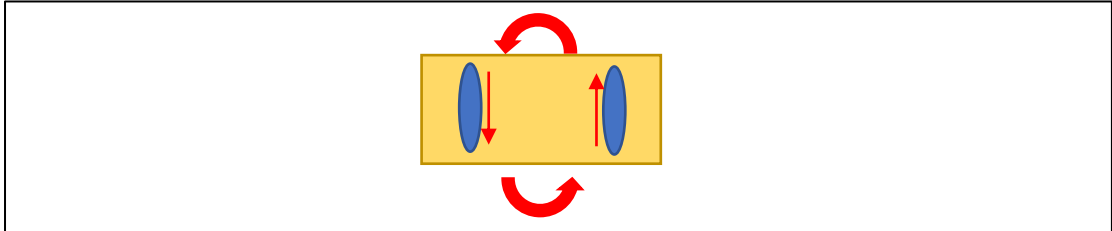
รูปที่ 4-13 ทิศทางการเคลื่อนที่ของชิ้นงานในทิศทางหมุนตามเข็มนาฬิกา

```
void clockwise(int steps){  
    digitalWrite(d_left, HIGH);  
    digitalWrite(d_right, LOW);  
    while(steps>0){  
        digitalWrite(c_left, HIGH);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        digitalWrite(c_left, LOW);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        steps -= 1;  
    }  
}
```

รูปที่ 4-14 คำสั่งควบคุมทิศทางการเคลื่อนที่ของชิ้นงานในทิศทางหมุนตามเข็มนาฬิกา

Rotate Counter-Clockwise

เป็นการหมุนชิ้นงานตามทวนเข็มนาฬิกา โดยล้อข้างซ้ายจะหมุนเป็นทิศ -Y และล้อข้างขวาจะหมุนไปทิศ +Y



รูปที่ 4-15 ทิศทางการเคลื่อนที่ของชิ้นงานในทิศทวนเข็มนาฬิกา

```
void counterclockwise(int steps){  
    digitalWrite(d_left, LOW);  
    digitalWrite(d_right, HIGH);  
    while(steps>0){  
        digitalWrite(c_left, HIGH);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        digitalWrite(c_left, LOW);  
        digitalWrite(c_right, HIGH);  
        delayMicroseconds(delay_time);  
        steps -= 1;  
    }  
}
```

รูปที่ 4-16 คำสั่งควบคุมทิศทางการเคลื่อนที่ของชิ้นงานในทิศทวนเข็มนาฬิกา

4.3.2 ฟังก์ชันการควบคุมการทำงานของกล่องควบคุม

1. การแสดงผลจอ LCD แบบ I2C

```
LcdTest
/* Libraries */
/* LCD Display */
#include <LiquidCrystal_I2C.h>
/* Others */
#include <Wire.h>

/* Variables */
LiquidCrystal_I2C lcd(0x27,20,4);

void setup() {
  /* LCD Setup */
  lcd.init(); lcd.backlight();
  lcd_text("Hello World");
}

void loop() {

}

void lcd_text(String text){
  lcd.clear();
  lcd.setCursor(0,2);
  lcd.print(text);
}
```

รูปที่ 4-17 คำสั่งที่ใช้ในการควบคุมการแสดงผลของจอ LCD แบบ I2C



รูปที่ 4-18 ผลลัพธ์ของการทำงานตามคำสั่งใน รูปที่ 4-17

2. การรับข้อมูลจากแป้นตัวเลข [6]

```

/* Libraries */
#include <Arduino.h>
#include <Keypad.h>
/* LCD Display */
#include <LiquidCrystal_I2C.h>
/* Others */
#include <Wire.h>
#include <String.h>

/* Constants */
#define KEYPAD_ROWS 4
#define KEYPAD_COLS 4

/* Keypad Framework */
const char kp4x4Keys[KEYPAD_ROWS][KEYPAD_COLS] = {'1','2','3','A','4','5','6','B'}, {'7','8','9','C'}, {'0','D','E','F'}, {'*','0','1','2'};
byte rowPins4Pin[4] = {26, 27, 28, 29};
byte colPins4Pin[4] = {22, 23, 24, 25};

/* Variables */
Keypad kp4x4 = Keypad(makeKeymap(kp4x4Keys), rowPins4Pin, colPins4Pin, KEYPAD_ROWS, KEYPAD_COLS);
LiquidCrystal_I2C lcd(0x27, 20, 4);

void setup() {
  // LCD Setup
  lcd.init(); lcd.backlight();
}

void loop() {
  // put your main code here, to run repeatedly:
  char c = readKeypad();
  lcd_text(String(c));
}

char readKeypad() {
  while(1) {
    char customKey = kp4x4.getKey();
    if (customKey) {
      return customKey;
    }
  }
}

void lcd_text(String text) {
  lcd.clear();
  lcd.setCursor(0, 2);
  lcd.print(text);
}

```

รูปที่ 4-19 คำสั่งที่ใช้ในการรับข้อมูลจากแป้นตัวเลขแล้วแสดงผลในจอ LCD แบบ I2C



รูปที่ 4-20 ผลลัพธ์ของการทำงานตามคำสั่งใน รูปที่ 4-19

การรับข้อมูลจาก Analog Button

```

AnalogButtonTest §
/* LCD Display */
#include <LiquidCrystal_I2C.h>
/* Constants */
#define ANALOG_KEY_PIN A0
/* Variables */
LiquidCrystal_I2C lcd(0x27,20,4);
void setup() {
  /* Analog Key Setup */
  pinMode(ANALOG_KEY_PIN, INPUT_PULLUP);
  /* LCD Setup */
  lcd.init(); lcd.backlight();
}
void loop() {
  String str = readAnalogKeypad();
  lcd_text(str);
}
String readAnalogKeypad(){
  /* 1: UP 2: LEFT 3: RIGHT 4: DOWN 0: SELECT */
  while(1){
    int val = analogRead(ANALOG_KEY_PIN);
    if(val <= 20) return "LEFT";
    else if(val >= 100 && val <= 200) return "UP";
    else if(val >= 300 && val <= 400) return "DOWN";
    else if(val >= 500 && val <= 600) return "RIGHT";
    else if(val >= 700 && val <= 800) return "SELECT";
  }
}
void lcd_text(String text){
  lcd.clear(); lcd.setCursor(0,2); lcd.print(text);
}

```

รูปที่ 4-21 คำสั่งที่ใช้ในการรับข้อมูลจากการป้อนจาก Analog Button แล้วแสดงผลในจอ LCD



รูปที่ 4-22 ผลลัพธ์ของการทำงานตามคำสั่งใน รูปที่ 4-21

3. ฟังก์ชันการควบคุมการทำงานของ RC Servo Motor [7]

```
ServoTest
/* Libraries */

/* Servo Motor*/
#include <Servo.h>
/* Others */
#include <Wire.h>

/* Constants */
#define SERVO_PIN 8

/* Variables */
Servo sv;

void setup() {
  sv.attach(SERVO_PIN);
}

void loop() {
  rotate_servo_to_degree(0); // to rotate to 0 degree (minimum)
  rotate_servo_to_degree(90); // to rotate to 90 degree
  rotate_servo_to_degree(180); // to rotate to 180 degree (maximum)
}

void rotate_servo_to_degree(int degree){
  if(0<=degree && degree<=180){
    sv.write(degree);
  }
}
```

รูปที่ 4-23 คำสั่งที่ใช้ในการควบคุม RC Servo Motor ให้ไปยังตำแหน่งองศาต่าง ๆ

4. ฟังก์ชันแสดงผลหน้าจอเมนู

```
void menu(int page_number){  
    if(page_number==1){  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("1| MANUAL MODE");  
        lcd.setCursor(0,1);  
        lcd.print("2| EXAMPLES");  
        lcd.setCursor(0,2);  
        lcd.print("3| FROM SD CARD");  
        lcd.setCursor(0,3);  
        lcd.print("A| NEXT PAGE");  
    }else if(page_number==2){  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("4| -");  
        lcd.setCursor(0,1);  
        lcd.print("5| -");  
        lcd.setCursor(0,2);  
        lcd.print("6| -");  
        lcd.setCursor(0,3);  
        lcd.print("A| NEXT PAGE");  
    }  
}
```



รูปที่ 4-24 ภาพถ่ายการแสดงผลตามฟังก์ชันแสดงผลหน้าจอเมนู

5. ฟังก์ชันแสดงผลหน้าจอควบคุมการทำงานแบบกำหนดเอง

```
void show_manual_guide() {  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("DIR|");  
    lcd.setCursor(0,1);  
    lcd.print("KEY|");  
    lcd.setCursor(5,0); lcd.write(byte(1));  
    lcd.setCursor(7,0); lcd.write(byte(2));  
    lcd.setCursor(9,0); lcd.write(byte(3));  
    lcd.setCursor(11,0); lcd.write(byte(4));  
    lcd.setCursor(5,1); lcd.print("4");  
    lcd.setCursor(7,1); lcd.print("6");  
    lcd.setCursor(9,1); lcd.print("2");  
    lcd.setCursor(11,1); lcd.print("8");  
    lcd.setCursor(13,3); lcd.print("EXIT(C)");  
}
```



รูปที่ 4-25 ภาพถ่ายการแสดงผลตามฟังก์ชันแสดงผลหน้าจอควบคุมการทำงานแบบกำหนดเอง

4.3.3 ฟังก์ชันการตั้งศูนย์

เนื่องจากระบบมีความเป็นไปได้ที่จะเกิดค่าความผิดพลาดเล็กน้อยซึ่งยอมรับได้แต่เมื่อปล่อยไว้จะสะสมจนเกินที่รับได้จึงต้องมีการหาศูนย์เพื่อลดความผิดพลาดสะสมของระบบ แล้วค่อยกลับไปทำงานต่อโดยมีหลักการดังนี้

โปรแกรมจะคอยนับจำนวนคำสั่งหรือจำนวนรอบเมื่อถึงที่กำหนดเริ่มทำการหาศูนย์

เชื่อว่าปัจจุบันอยู่จุดใดของกระดาน

เลื่อนปรับมุมจนเส้นที่ใช้เป็นตัววัดกึ่งกลางของกระดานอยู่ในจุดที่เป็นจุดศูนย์

เลื่อนกลับไปยังจุดและมุมที่เก็บไว้ก่อนที่จะหาศูนย์แล้วทำงานต่อไป

4.3.4 ฟังก์ชันจดจำตำแหน่ง

เนื่องจากว่าเพื่อความสะดวกหลังการหาศูนย์แล้วมีความจำเป็นต้องทำงานต่อจึงต้องมีการจดจำจุดที่ทำงานล่าสุดก่อนหาศูนย์เพื่อกลับไปยังจุดล่าสุดที่อยู่โดยข้อมูลจะเก็บไว้ที่โครงสร้างข้อมูลที่เก็บข้อมูลไว้ 3 ข้อมูล คือ มุมที่หมุนไป พิกัดในแกน x พิกัดในแกน y โดยจะตั้งให้จุดกึ่งกลางเป็นพิกัด (0,0) และเก็บมุมที่มีขนาดตั้งแต่ 0 ถึง 360 องศา

4.3.5 เมนูควบคุมการทำงานของระบบ

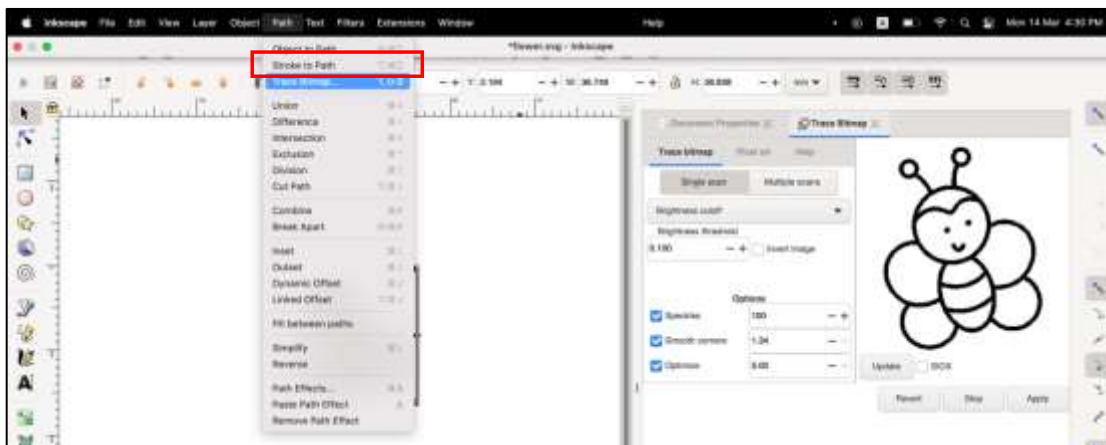
ในการทำงานของระบบจะมีการแสดงเมนูในหน้าจอแสดงผลเพื่อรอรับคำสั่งจากผู้ใช้โดยมีเมนูใช้งาน 3 โหมด คือ

- โหมดการควบคุมการทำงานแบบกำหนดเอง
- โหมดตัวอย่าง (รูปเรขาคณิตพื้นฐาน)
- โหมดวาดภาพโดยนำเข้าข้อมูลจาก SD Card

4.3.6 การแปลงข้อมูลภาพวาดเป็นชุดข้อมูลรูปแบบ G-Code

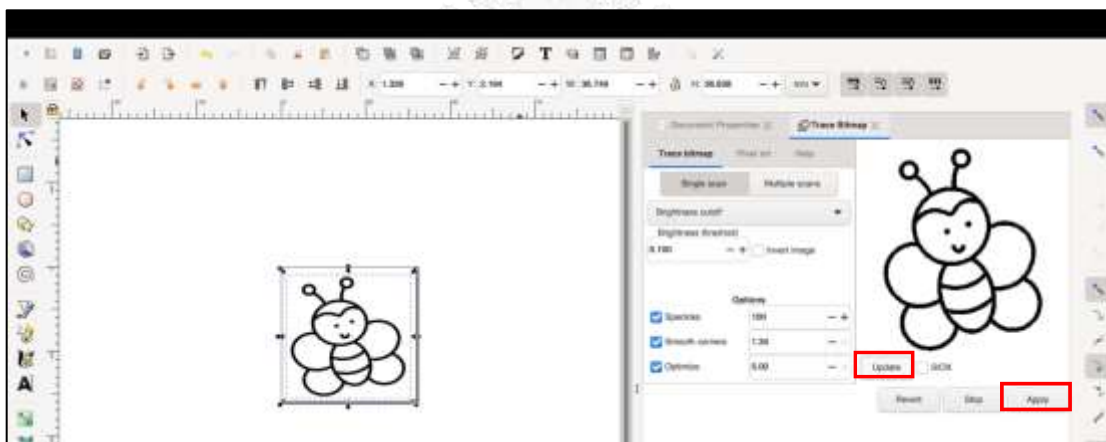
ในการแปลงข้อมูลภาพวาดที่อยู่ในรูปแบบไฟล์ภาพจะต้องใช้โปรแกรม Inkscape ในการจัดการแปลงข้อมูลภาพเป็นข้อมูลในรูปแบบ G-Code ซึ่งมีขั้นตอนดังนี้

1. หลังจากเปิดโปรแกรมและนำเข้ารูปภาพที่ต้องการแล้วเลือก Path > Trace Bitmap...



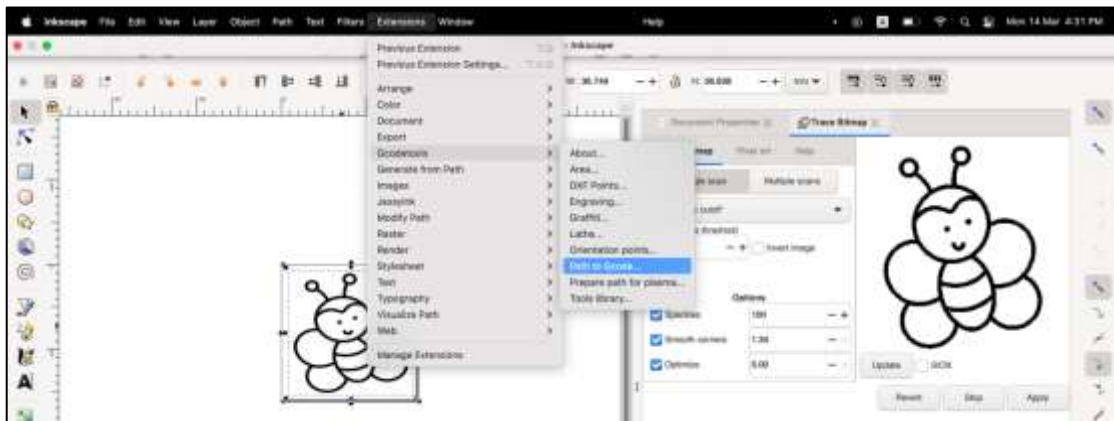
รูปที่ 4-26 ขั้นตอนที่ 1 ของการแปลงรูปภาพเป็น G-Code

2. เลือก Update และเลือก Apply



รูปที่ 4-27 ขั้นตอนที่ 2 ของการแปลงรูปภาพเป็น G-Code

3. เลือก Extensions > Gcodetools > Path to Gcode...



รูปที่ 4-28 ขั้นตอนที่ 3 ของการแปลงรูปภาพเป็น G-Code

4. จากนั้นเลือกแฟ้มที่ต้องการบันทึก และกดบันทึก

4.3.7 ซอฟต์แวร์แปลง GCode เป็นชุดข้อมูลพิกัดการวาดอย่างง่าย

เนื่องจากในการวาดภาพในโครงงานนี้ใช้ระบบพิกัดที่มีความแตกต่างกันกับระบบพิกัดที่ได้จาก GCode จึงจะต้องมีการแปลงข้อมูลซ้ำอีกครั้งเพื่อนำไปใช้งาน ซึ่งมีการกรองค่าที่ไม่จำเป็นที่ได้จากการแปลงรูปภาพเป็น GCode ออกไปด้วย

4.3.8 ซอฟต์แวร์สำหรับการจำลองการวาดภาพ

เพื่อเป็นการตรวจสอบผลการแปลงรูปภาพเป็น GCode ผู้จัดทำจึงได้พัฒนาซอฟต์แวร์สำหรับจำลองภาพวาดผลลัพธ์ที่จะได้โดยใช้ Flutter ในการพัฒนาและเผยแพร่ให้ใช้งานในรูปแบบเว็บ (ลิงค์: <https://drawing-tools-akarapon.web.app/>) ซึ่งได้รวมซอฟต์แวร์แปลง GCode เป็นชุดข้อมูลพิกัดการวาดอย่างง่ายไว้ในเว็บไซต์ดังกล่าวด้วย

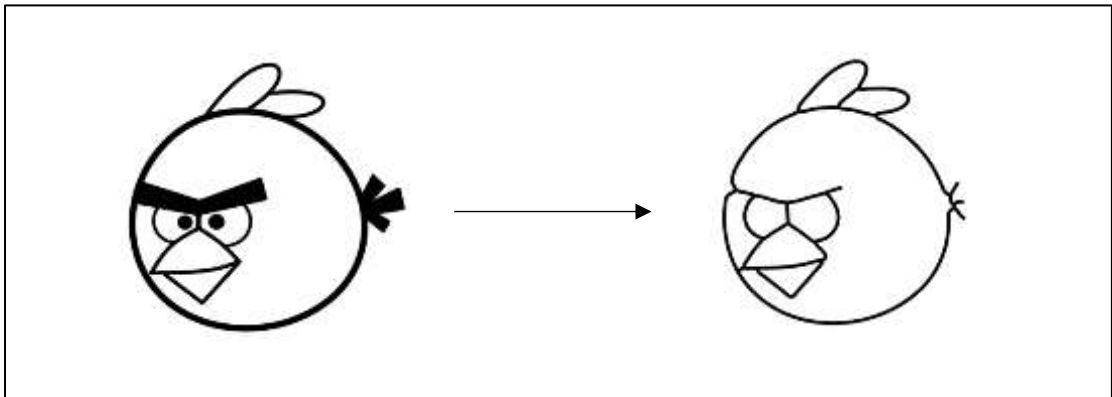


รูปที่ 4-29 หน้าเว็บไซต์สำหรับการจำลองการวาดภาพ

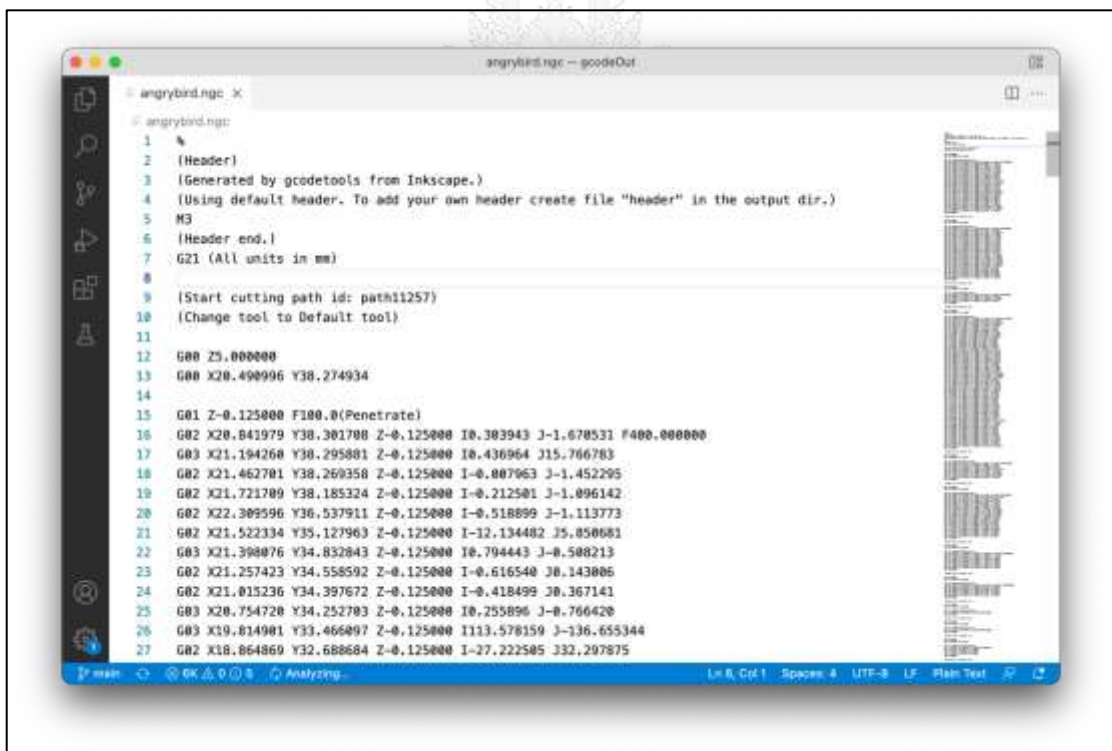
4.4 ผลการทดสอบ

ความก้าวหน้าในระยะที่ 3 นี้จะมุ่งเน้นไปที่การออกแบบและพัฒนาอัลกอริทึมและระบบอิเล็กทรอนิกส์ที่ใช้ในการควบคุมการทำงานของอุปกรณ์ฮาร์ดแวร์เป็นหลักโดยมีรายละเอียดความก้าวหน้าดังนี้

4.4.1 ผลการแปลงรูปภาพเป็น GCode

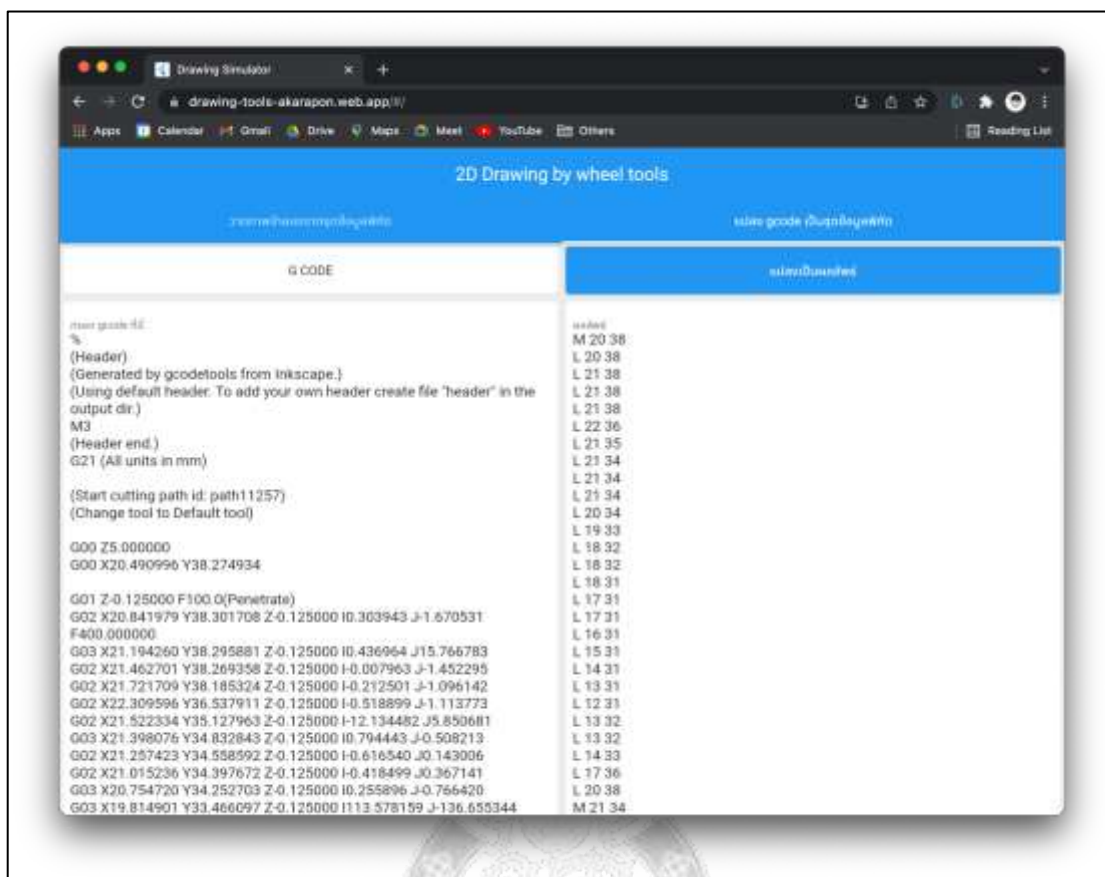


รูปที่ 4-30 ผลการใช้ฟังก์ชัน Trace Bitmap... บนโปรแกรม Inkscape



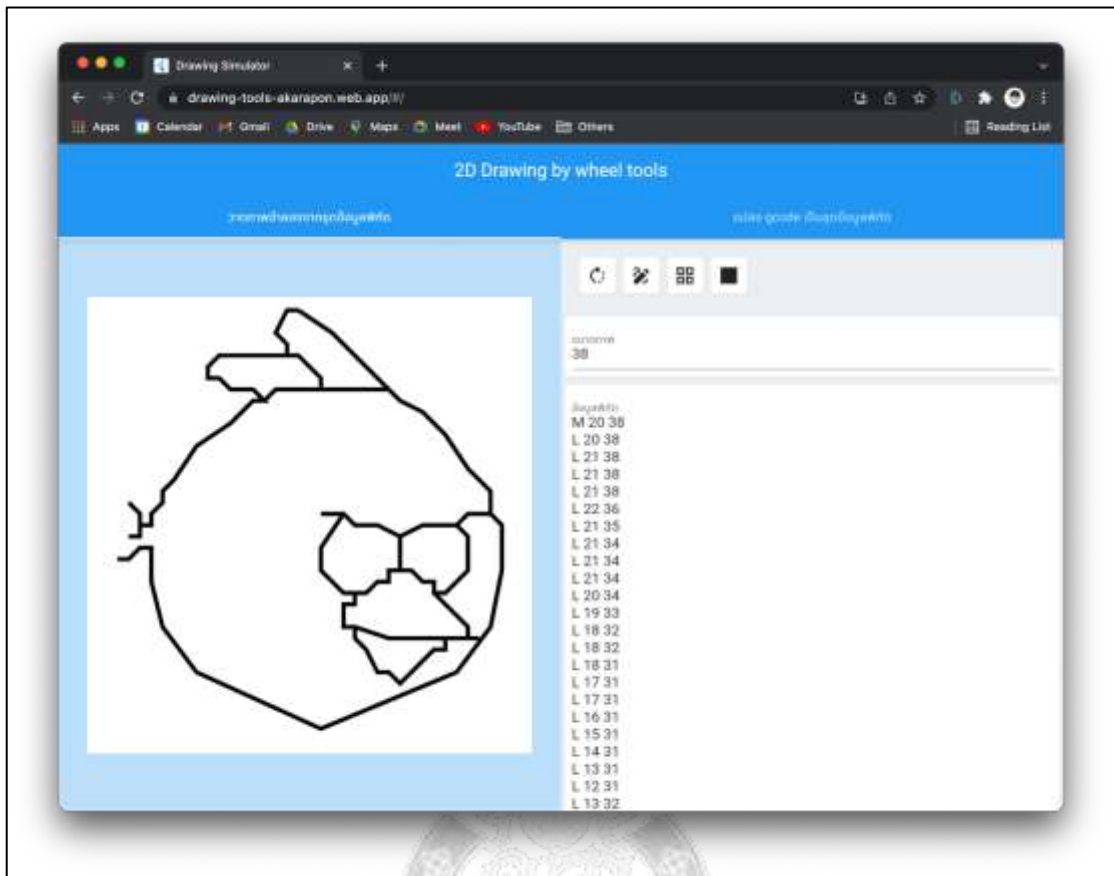
รูปที่ 4-31 GCode ที่ได้จากการแปลงจากรูป 4 3

4.4.2 ผลการแปลงข้อมูลในรูปแบบ GCode เป็นชุดข้อมูลเพื่อการวาดอย่างง่าย



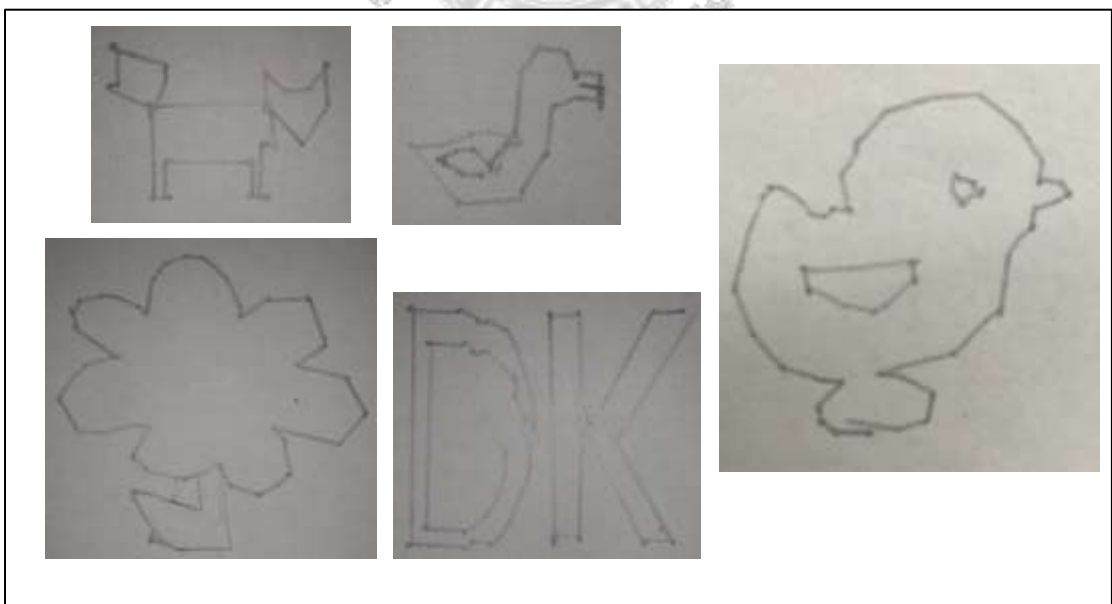
รูปที่ 4-32 ผลการแปลงข้อมูลในรูปแบบ GCode เป็นชุดข้อมูลเพื่อการวาดอย่างง่าย

4.4.3 ผลการจำลองการวาดภาพ



รูปที่ 4-33 ผลการจำลองการวาดภาพ

4.4.4 ผลการวาดภาพ



รูปที่ 4-34 ผลการวาดภาพ

บทที่ 5 สรุป

5.1 สรุป

จากการศึกษาการทำงานและออกแบบหุ่นยนต์วาดภาพ 2 มิติโดยการขับเคลื่อนด้วยล้อ สามารถพัฒนาระบบให้มีความสามารถในการเคลื่อนที่ไปยังพิกัดที่ต้องการได้ ซึ่งสามารถวาดภาพ 2 มิติ โดยระบุค่าพิกัดโดยตรงได้

ชุดกลไกวาดภาพที่พัฒนาขึ้นนี้สามารถรับคำสั่งที่ถูกควบคุมผ่านกล่องควบคุมโดยจะมี LCD-Display แสดงหน้าจอ Interface โหมดการทำงานต่าง ๆ ให้เลือกทำงานพร้อมทั้งปุ่มควบคุม โดยสามารถทำงานได้ตามโหมดต่าง ๆ ที่ระบุไว้ เช่น สามารถควบคุมให้เคลื่อนที่ไปยังจุดปลายการทำงานไปยังพิกัด (0,0) ได้โดยอัตโนมัติ , โหมด Manual ที่สามารถควบคุมส่วนการเคลื่อนที่ไปยังพิกัดต่าง ๆ บนจุดทำงาน โดยมี Joystick เป็นอุปกรณ์ควบคุม , โหมด Test sample จะควบคุมแขนหุ่นยนต์ให้ทำการวาดรูปเรขาคณิตอย่างง่ายที่ได้จากการระบุค่าพิกัดซึ่งสามารถใช้เป็นต้นแบบสำหรับการพัฒนาต่อไปในอนาคต

5.2 ปัญหาและอุปสรรค

5.2.1 งบประมาณในการจัดทำโครงการ

5.2.2 สถานการณ์การแพร่ระบาดของเชื้อไวรัสโควิด 19

ทำให้ไม่สามารถเข้าใช้พื้นที่ห้องปฏิบัติการอย่างต่อเนื่องได้ทำให้เป็นอุปสรรคในการดำเนินงาน

5.2.3 เมื่อเกิดการหมุนเป็นเวลานานจะต้องมีการตั้งศูนย์ใหม่

เนื่องจากกลไกในการแก้ไขความผิดพลาดของการทำงาน ผู้จัดทำกำหนดให้ต้องมีการตั้งศูนย์ใหม่ ซึ่งกระบวนการนี้จะต้องใช้มือในการปรับหมุนเพื่อตั้งศูนย์ของปากกา

5.2.4 การขาดความชำนาญในการใช้เครื่องมือช่าง

เช่น สว่าน หัวแร้งและอื่น ๆ ซึ่งทำให้เกิดความล่าช้าในการทำงานเนื่องจากต้องเรียนรู้วิธีการใช้งานเครื่องมือก่อนทำงาน

บรรณานุกรม

- [1] Kamolc. (2021, Jan 15). แขนหุ่นยนต์[Online].Available:
<https://www.thairobotics.com/2014/01/10/byob-robot-arm/>
- [2] MEGAN RAY NICHOLS (2021, Jan 16)Which Industrial Robot Design Is Right for You?[Online].Available:
<https://www.eeworldonline.com/which-industrial-robot-design-is-right-for-you/>
- [3] gravitechthai.com. (2021, Jan 15). Arduino Mega2560[Online].Available:
<https://www.gravitechthai.com/product-detail.php?WP=qmIzAJ1CM5O0hJatrTZo7o3Q>
- [4] thaieasyelec.com.(2021, Jan 15).บทความ "ตัวอย่างการควบคุม RC Servo Motor ด้วย Arduino"[Online].Available: <https://www.thaieasyelec.com/article-wiki/review-productarticle/Example-project-how-to-control-rc-servo-motor-with-arduino.html>
- [5] factomart.com. (2021, Jan 15).ประเภทของ Stepping Motor[online].Available:
<https://mall.factomart.com/type-of-stepping-motor/>
- [6] thaiio.com. (2021, Jan 15). ความรู้เบื้องต้น และ หลักการทำงาน Step Moter ตอนที่1[Online].Available:
<http://www.thaiio.com/Hardware/stepmoter.htm>
- [7]] factomart.com. (2021, Jan 15).บทความ” Stepping Motor Drive คืออะไร?” [online].Available:
<https://mall.factomart.com/what-is-stepping-motor-drive/>
- [8]] know2learning.blogspot.com. (2021, Jan 15). การใช้งานแป้นปุ่มกด (Keypad) แบบ 4x4 ปุ่ม [Online]. Available: <http://know2learning.blogspot.com/2017/02/keypad-4x4.html>
- [9] ioxhop.com. (2021, Jan 15). การใช้งานจอ Character LCD กับ Arduino แบบละเอียด [Online].Available: <https://www.ioxhop.com/article/>
- [10] arduinoall.com.(2021, Jan 15).Arduino สอนการใช้งาน จอยสติ๊ก PS2 joy[Online].Available: <https://www.arduinoall.com/article/307/71-arduino-%E0%B8%81-ps2-joy>