โครงการ
การเรียนการสอนเพื่อเสริมประสบการณ์

ชื่อโครงการ การประดิษฐ์เครื่องพล็อตเตอร์ที่ควบคุมด้วยไมโครคอนโทรเลอร์อาดูโน่

ชื่อนิสิต อนันตศักดิ์ น้ำเย็น เลขประจำตัว 6033445823

ภาควิชา ฟิสิกส์

ปีการศึกษา 2563

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

หัวข้อโครงการ	การประดิษฐ์เครื่องพล็อตเตอร์ที่ควบคุมด้วยไมโครคอนโทรเลอร์อาดูโน่
ผู้จัดทำโครงการ	นายอนันตศักดิ์ น้ำเย็น
อาจารย์ที่ปรึกษา	ผู้ช่วยศาสตราจารย์ อำนาจ สาธานนท์
ภาควิชา	ฟิสิกส์
ปีการศึกษา	2563

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต ภาควิชาฟิสิกส์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ภาคการศึกษาปลาย ปีการศึกษา 2563

คณะกรรมการได้ตรวจรับรองรายงานฉบับนี้แล้ว

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.มนต์เทียน เทียนประทีป)

.....กรรมการสอบ

(รองศาสตราจารย์ ดร.รัฐชาติ มงคลนาวิน)

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ อำนาจ สาธานนท์)

หัวข้อโครงการ การประดิษฐ์เครื่องพล็อตเตอร์ที่ควบคุมด้วยไมโครคอนโทรเลอร์อาดูโน่

ผู้จัดทำโครงการ นายอนันตศักดิ์ น้ำเย็น

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ อำนาจ สาธานนท์

ภาควิชา ฟิสิกส์

ปีการศึกษา 2563

บทคัดย่อ

โครงการนี้นำเสนอการสร้างเครื่องพล็อตเตอร์ที่มีพื้นฐานการเคลื่อนที่เป็นแบบสองแกน (XY-Based) จากการประยุกต์ใช้ไมโครคอนโทรลเลอร์อาดูโน่เชื่อมต่อกับซีเอ็นซีชิลด์และสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ เพื่อควบคุมสเต็ปปิ้งมอเตอร์สองตัว โดยใช้ข้อมูลรูปภาพในรูปแบบของไฟล์จีโค้ด โครงการนี้ได้ออกแบบให้เครื่องพล็อตเตอร์เคลื่อนที่ด้วยระบบล้อและเพลา แบบโฮโลโนมิคไดรฟ์ ทำให้สามารถเคลื่อนที่ได้ทุกทิศทาง อุปกรณ์จับปากกาขับเคลื่อนด้วยเซอร์โวมอเตอร์ที่รับสัญญาณพัลส์วิดมอดูเลชั่น (Pulse Width Modulation, PWM) ในการควบคุมการยกปากกา จากการคำนวณค่าความละเอียดในแต่ละแกน พบว่าทั้งแกนเอกซ์และวายมีค่าเท่ากับ 0.87 และ 0.57 มิลลิเมตรตามลำดับ จากการทดสอบพบว่าแนวคิดการสร้างเครื่องพล็อตเตอร์ในลักษณะนี้มีความเป็นไปได้ แต่ยังมีจุดบกพร่องหลายจุด จำเป็นต้องแก้ไขเพื่อใช้งานจริงต่อไป

คำสำคัญ: การออกแบบพล็อตเตอร์, สเต็ปปิ้งมอเตอร์, เซอร์โวมอเตอร์, การเคลื่อนที่แบบโฮโลโนมิค, พัลส์วิดมอดูเลชั่น

Title Movable CNC plotter based on Arduino.

Name Anantasak Namyen

Adviser Asst.Prof. Umnart Sathanon

Major Physics

Academic year 2563

Abstract

This project presents the creation of a two-axis motion plotter using Arduino with CNC Shield and stepping motor driver to control two stepping motors of the x-axis and the y-axis. The plotter was operated by G-code image file and moved wheel and axle system with holonomic drive. The pen holder was attached to a PWM servo motor for pressing and lifting. The resolution on x and y axis are 0.87 and 0.57 millimeters respectively. By testing of the actual drawing, the plotter design concept of a simple movable plotter is useful and viable but there are still many improvements for actual working.

Keyword: Plotter design, Stepping motor, Servo motor, Holonomic drive, Pulse Width Modulation(PWM)

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ผู้ช่วยศาสตราจารย์ อำนาจ สาธานนท์ อาจารย์ที่ปรึกษาโครงการที่ได้ให้คำเสนอแนะ แนวทางการทำงาน แนวคิด ตลอดจนคอยช่วยแนะนำแก้ไขข้อบกพร่องต่าง ๆ มาโดยตลอด จนทำให้โครงการนี้เสร็จสมบูรณ์ ผู้จัดทำโครงการขอขอบพระคุณอย่างสูงมา ณ ที่นี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.มนต์เทียน เทียนประทีป ที่กรุณาให้เกียรติเป็นกรรมการสอบและแนะนำแนวทางการแก้ไขโครงการนี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร.รัฐชาติ มงคลนาวิน ที่กรุณาให้เกียรติเป็นกรรมการสอบและแนะนำแนวทางการทำงานในโครงการ

นาย อนันตศักดิ์ น้ำเย็น

นิสิตผู้รับผิดขอบโครงการ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....i

บทคัดย่อภาษาอังกฤษ.....ii

กิตติกรรมประกาศ.....iii

สารบัญ.....iv

สารบัญตาราง.....vii

สารบัญภาพ.....viii

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ.....1

1.2 วัตถุประสงค์ของโครงการ.....1

1.3 ขอบเขตงานวิจัย.....1

1.4 ระเบียบวิธีวิจัย.....1

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2 ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่ใช้

2.1 อุปกรณ์ที่สำคัญ.....4

2.1.1 ไมโครคอนโทรลเลอร์ อาดูโน่ (Microcontroller Arduino).....4

2.1.2 สเต็ปปิ้งมอเตอร์ NEMA17 HS4401.....5

2.1.3 สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ A4988 5

2.1.4 มู่เล่ย์และสายพานจีทีสอง.....6

2.1.5 อาดูโน่ ซีเอ็นซีชิลด์ (Arduino CNC shield).....7

2.1.6 ไมโครเซอร์โวมอเตอร์ 6 โวลต์ 180 องศา (Micro servo motor).....8

2.1.7 ล้อออมนิ ไดเรกชันนอล (Omni directional wheel).....8

2.2 ทฤษฎีที่เกี่ยวข้อง.....9

2.2.1 ส่วนประกอบพื้นฐานของโปรแกรมซีเอ็นซีหรือจีโค้ด (G code).....10

2.2.2 การหาค่าความละเอียดของเครื่องเอ็นซี.....10

2.2.3 วงจรเอ็ช-บริดจ์ (H-bridge).....10

2.2.4 ชนิดและหลักการการกระตุ้นสเต็ปปิ้งมอเตอร์.....11

2.2.5 การเคลื่อนที่แบบ โฮโลโนมิค ไดรฟ์ (Holonomic drive).....14

บทที่ 3 การออกแบบและขั้นตอนการทำงน

3.1 ศึกษาและทำความเข้าใจทฤษฎีที่เกี่ยวข้อง.....15

3.2 ศึกษาการทำงานของอุปกรณ์.....15

3.3 ออกแบบอุปกรณ์.....15

3.4 ขั้นตอนการสร้างเครื่องพล็อตเตอร์ในรูปแบบที่สามารถเคลื่อนที่ได้.....16

3.4.1 ต่อวงจรของอุปกรณ์.....16

3.4.2 แนวคิดและการสร้างระบบการเคลื่อนที่ให้กับเครื่องพล็อตเตอร์.....17

3.4.3 สร้างระบบปากกาจากเซอร์โวมอเตอร์.....19

3.4.4 ประกอบแต่ละส่วนเข้าด้วยกันทำให้สิ่งประดิษฐ์เสร็จสมบูรณ์.....20

3.5 ปรับปรุงและแก้ไขโปรแกรมเพื่อให้อุปกรณ์ต่าง ๆ ที่ออกแบบสามารถทำงานได้ตามต้องการ.....21

3.5.1 ส่วนของการขับเคลื่อนมอเตอร์.....21

3.5.2 ส่วนในการสร้างและปรับปรุง G-code.....23

3.6 การทดสอบความสามารถในการเขียนเส้นแบบต่าง ๆ 25

3.6.1 ทดสอบฟังชั่นการทำงานของเครื่องพล็อตเตอร์.....25

3.6.2 ทดสอบการลากเส้นตรง และหาความผิดพลาด.....25

3.6.3 ทดสอบความสามารถในการวาดรูปจริง.....25

บทที่ 4 ผลการทดลอง

4.1 การทดสอบฟังชั่นการทำงาน.....26

4.2 การทดปรับค่าสเต็ปต่อมิลลิเมตร ของแต่ละแกนให้สัมพันธ์กัน.....27

4.3 การทดสอบความคลาดเคลื่อนจากแรงเสียดทานและผลกระทบจากสิ่งต่าง ๆ ของสิ่งประดิษฐ์.....28

4.4 การทดสอบความคลาดเคลื่อนในการเขียนเส้นที่ไม่ใช่แนวทแยงที่เกิดจากขนาดของล้อที่ต่างกัน.....34

4.5 การทดสอบการทำงานจริงและขอบเขตงาน.....35

บทที่ 5 สรุปและอภิปรายผลการทดลอง

5.1 วิเคราะห์ผลการทดลอง.....37

5.2 สรุปและอภิปรายผลการทดลอง.....40

5.3 ปัญหาที่พบ.....42

5.4 ข้อเสนอแนะ.....42

บรรณานุกรม.....**43**

ภาคผนวก.....**44**

สารบัญตาราง

ตารางที่ 1.1 ตารางการปฏิบัติงาน....2

ตารางที่ 2.1 การจ่ายกระแสไฟเพื่อให้สเต็ปปิ้งมอเตอร์หมุนแบบ ฟูล-สเต็ป....12

ตารางที่ 2.2 การจ่ายกระแสไฟเพื่อให้สเต็ปปิ้งมอเตอร์หมุนแบบ ฟูล-สเต็ป 2 เฟส....12

ตารางที่ 2.3 การจ่ายกระแสไฟเพื่อให้สเต็ปปิ้งมอเตอร์หมุนแบบ ฮาล์ฟ-สเต็ป....13

ตารางที่ 4.1 แสดงคำสั่ง ความหมาย และจุดประสงค์ของการทดสอบ....26

ตารางที่ 4.2 แสดงข้อมูลที่สำคัญในการปรับเครื่องพล็อตเตอร์....28

ตารางที่ 4.3 แสดงการทดสอบขีดเส้นในแนวแกนเอกซ์ความคลาดเคลื่อนสัมพัทธ์และความเบี่ยงเบนมาตรฐาน....29

ตารางที่ 4.4 แสดงการทดสอบขีดเส้นในแนวแกนวายความคลาดเคลื่อนสัมพัทธ์และความเบี่ยงเบนมาตรฐาน....32

ตารางที่ 5.1 ตารางแสดงเมตริกซ์ของความไม่แน่นอนของแกนเอกซ์....38

ตารางที่ 5.2 ตารางแสดงเมตริกซ์ของความไม่แน่นอนของแกนวาย....39

สารบัญรูปภาพ

รูปที่ 2.1 พินเอาท์ไดรอแกรมของไมโครคอนโทรลเลอร์อาดูโน่ รุ่น UNO....4

รูปที่ 2.2 สเต็ปปิ้งมอเตอร์17HS4401 12V....5

รูปที่ 2.3 สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ A4988....6

รูปที่ 2.4 (ก) มู่เล่ย์ 20 ฟัน (ข) สายพาน GT2....6

รูปที่ 2.5อาดูโน่ ซีเอ็นซีชิลด์ (Arduino CNC shield)....7

รูปที่ 2.6 อาดูโน่ซีเอ็นซีชิลด์ พินเอาท์ (Arduino CNC Shield PINOUT)....7

รูปที่ 2.7 ไมโครเซอร์โวมอเตอร์ SG90 180 Degree....8

รูปที่ 2.8 ล้อออมนิ ไดเรกชันนอล (Omni directional wheel)....8

รูปที่ 2.9 แสดงส่วนประกอบพื้นฐานของจีโค้ด....9

รูปที่ 2.10 ลักษณะการขับสเต็ปปิ้งมอเตอร์ด้วยวงจร H-bridge สองวงจร 10

รูปที่ 2.11 ขดลวดของสเต็ปปิ้งมอเตอร์ชนิด ไบโพลาร์ และ ยูนิโพลาร์....11

รูปที่ 2.12 การเคลื่อนที่แบบโฮโลโนมิกส์ ไดรฟ์ (Holonomic drive)....14

รูปที่ 3.1 แผนภาพโดยรวมของการสร้างสิ่งประดิษฐ์ 15

รูปที่ 3.2 การต่อวงจรของบอร์ดซีเอ็นซีชิลด์และมอเตอร์แบบต่าง ๆ....16

รูปที่ 3.3 ทำการบัดกรี L7806 และติดตั้ง stepper motor driver A4988....16

รูปที่ 3.4 การวางมอเตอร์ลักษณะตั้งฉากกัน 17

รูปที่ 3.5 การวางระบบเพลาลักษณะตั้งฉากกัน....17

รูปที่ 3.6 การติดตั้งมู่เล่ย์เพื่อใช้ในการขับเคลื่อนเครื่องพล็อตเตอร์....18

รูปที่ 3.7 ติดตั้งล้อตามหลักกาโฮโลโนมิค ไดรฟ์....18

รูปที่ 3.8 การสลับสปริงและตัดปลายปากกาเพื่อให้สปริงต้านแรงกดปากกา....19

รูปที่ 3.9 การเจาะปากกาเพื่อนำเซอร์โวมอเตอร์มาควบคุมการเปิดปิด....19

รูปที่ 3.10 เครื่องพล็อตเตอร์บนกระดาษพร้อมวัดผล....20

รูปที่ 3.11 ผังการทำงานของโปรแกรมสั่งการมอเตอร์ 21

รูปที่ 3.12 ตัวอย่างคำสั่งโค้ดในคำสั่งต่าง ๆ.....22

รูปที่ 3.13 การกำหนดพินของเซอร์โวมอเตอร์.....22

รูปที่ 3.14 ซอฟต์แวร์ Universal G code sender.....22

รูปที่ 3.15 แผนผังการทำงานของการสร้าง จีโค้ด.....23

รูปที่ 3.16 ใช้ซอฟต์แวร์ Inkscape ในการสร้างจีโค้ดรูปสี่เหลี่ยม.....24

รูปที่ 3.17 ใช้โปรแกรม Notepad++ ในการดัดแปลงและแก้ไขจีโค้ดที่ได้จากขั้นตอนแรก.....24

รูปที่ 4.1 หน้าต่างการปรับค่าต่าง ๆ ของส่วนควบคุมมอเตอร์.....27

รูปที่ 4.2 กราฟแสดงความยาวที่วาดได้(จุดสีฟ้า) ความยาวที่กำหนดในโปรแกรม(จุดสีแดง) และความเบี่ยงเบนมาตรฐาน (เส้นสีเขียว) ของแกนเอกซ์.....30

รูปที่ 4.3 กราฟแสดงการเปรียบเทียบระหว่างค่าที่ให้ในโปรแกรมและผลต่างระหว่างค่าจริงกับค่าที่กำหนดให้ของแกนเอกซ์.....30

รูปที่ 4.4 กราฟแสดงแนวโน้มของความเบี่ยงเบนมาตรฐานของการเขียนเส้นในแนวแกนเอกซ์..... 31

รูปที่ 4.5 กราฟแสดงความยาวที่วาดได้(จุดสีฟ้า)ความยาวที่กำหนดในโปรแกรม(จุดสีแดง)และความเบี่ยงเบนมาตรฐาน(เส้นสีเขียว) ของแกนวาย.....33

รูปที่ 4.6 กราฟแสดงการเปรียบเทียบระหว่างค่าที่ให้ในโปรแกรมและผลต่างระหว่างค่าจริงกับค่าที่กำหนดให้ของแกนวาย.....33

รูปที่ 4.7 กราฟแสดงแนวโน้มของความเบี่ยงเบนมาตรฐานของการเขียนเส้นในแนวแกนวาย.....34

รูปที่ 4.8 ตัวอย่างเส้นในการลากแนวทแยง.....34

รูปที่ 4.9 ทดสอบด้วยการวาดรูปสี่เหลี่ยม (ก) รูปในโปรแกรม(ข).....35

รูปที่ 4.10 ทดสอบด้วยการวาดรูปตัวอักษร T (ก) รูปในโปรแกรม (ข).....35

รูปที่ 4.11 ทดสอบด้วยการวาดรูปสามเหลี่ยม (ก) รูปในโปรแกรม (ข) 36

รูปที่ 4.12 ทดสอบด้วยการวาดรูปตัวอักษร w (ก) รูปในโปรแกรม (ข).....36

รูปที่ 4.13 ทดสอบด้วยการวาดรูปวงรี (ก) รูปในโปรแกรม (ข).....36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

เมื่อสำรวจตลาดในปัจจุบันจะพบว่าเครื่องพล็อตเตอร์ต่าง ๆ เป็นเครื่องแต่ตั้งอยู่กับที่ ทำให้มีข้อจำกัดในเรื่องขนาดของกระดาษและปัญหาเรื่องการเคลื่อนย้ายเครื่อง ซึ่งเป็นอุปสรรคในการใช้งานในบางโอกาส

ผู้ทำโครงการนี้จึงต้องการที่จะประดิษฐ์เครื่องพล็อตเตอร์อย่างง่ายที่มีราคาถูกเพื่อแก้ปัญหาดังกล่าว โดยมีแนวความคิดในการสร้างพล็อตเตอร์สองแกน (xy-plotter) ที่สามารถเคลื่อนที่อย่างอิสระ โดยไม่มีข้อจำกัดในเรื่องขนาดของกระดาษ โดยมีตัวแปรในการวัดความสามารถของสิ่งประดิษฐ์ และนำไปเป็นต้นแบบเพื่อพัฒนาต่อยอดต่อไป

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อประยุกต์ใช้ไมโครคอนโทรเลอร์อาดูโน่และอื่น ๆ ในการสร้างเครื่องพล็อตเตอร์อย่างง่าย เพื่อเป็นต้นแบบในการนำไปพัฒนาต่อไป

1.2.2 เพื่อศึกษาเกี่ยวกับโปรแกรม รูปแบบ และการใช้งานในการส่งและรับข้อมูลจากคอมพิวเตอร์

1.2.3 เพื่อเรียนรู้หลักการทำงาน ระบบอิเล็กทรอนิกส์เชิงกล (mechatronics) เพื่อใช้เป็นความรู้ในการทำงานต่อไป

1.3 ขอบเขตงานวิจัย

โครงการนี้เป็นการสร้างต้นแบบของ เอกซ์วายพล็อตเตอร์ ที่มีในการเคลื่อนที่บนกระดาษ ควบคุมโดยอาดูโน่ แล้วทดสอบและวัดผลการทำงานโดยการลากเส้นในแบบต่าง ๆ เพื่อหากำหนดขีดจำกัดในการใช้งานจริง และบอกข้อบกพร่องของสิ่งประดิษฐ์เพื่อสามารถนำไปต่อยอดและพัฒนาต่อไปได้

1.4 ระเบียบวิธีวิจัย

1.4.1 แผนการศึกษา

1. ศึกษาความรู้เกี่ยวกับอิเล็กทรอนิกส์เชิงกล

1.1 ศึกษาและทดสอบการใช้อาดูโน่ ในเรื่องการรับและส่งสัญญาณควบคุมมอเตอร์ โดยเน้นไปที่การส่งสัญญาณแบบ พัลส์วิดมอดูเลชั่น

1.2 ศึกษาหลักการทำงานของสเต็ปปิ้งมอเตอร์แต่ละชนิด คุณสมบัติ เพื่อที่จะได้นำความรู้มาใช้เลือกรูปแบบที่เหมาะสมกับงานที่ทำ

1.3 ศึกษาวงจรขับสเต็ปปิ้งมอเตอร์ด้วยอาดูโน่ เพื่อทำให้สามารถเชื่อมต่อได้อย่างถูกต้องและป้องกันความเสียหายที่เกิดอาจจะเกิดขึ้น

2. ทำการศึกษาเกี่ยวกับการเขียนโปรแกรมในการส่งและรับข้อมูลจีโค้ด

2.1 ศึกษาการเขียนโปรแกรมด้วย อาดูโน่ ไอดีอี (Arduino IDE)

2.2 ศึกษาเกี่ยวกับการใช้งานจีโค้ด การส่ง การใช้งานคำสั่งต่าง ๆ

3. คัดเลือกเลือกอุปกรณ์ที่จะใช้

3.1 เลือกรุ่นของไมโครคอนโทรลเลอร์อาดูโน่ ผู้วิจัยเลือกใช้รุ่น อาดูโน่ อูโน่ (Arduino UNO) เนื่องจากมีราคาที่ไม่แพง และมีคุณสมบัติพื้นฐานเพียงพอที่จะนำมาใช้ในการสร้างพล็อตเตอร์

3.2 เลือกสเต็ปปิ้งมอเตอร์ให้มีขนาดและกำลังเหมาะสมกับเครื่องพล็อตเตอร์ที่จะสร้าง รวมทั้งเลือกสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์และอุปกรณ์เสริมต่าง ๆ ให้สามารถใช้ทำงานด้วยกันได้โดยไม่เกิดปัญหา

3.3 กำหนดขนาดของอุปกรณ์เชิงกลได้แก่ สายพานและอุปกรณ์ประกอบอื่น ๆ ให้เหมาะสมกับขนาดของอาดูโน่และมอเตอร์ โดยอุปกรณ์ประกอบนี้จะเป็นตัวกำหนดความละเอียดของพล็อตเตอร์

4. **เริ่มทำการออกแบบ สร้าง และดัดแปลงแก้ไขเครื่องพล็อตเตอร์**

4.1เริ่มทำการออกแบบรูปร่างของพล็อตเตอร์ ลักษณะการเดินสายพานและออกแบบหลักการเคลื่อนที่ของล้อพล็อตเตอร์

4.2 เริ่มทำการสร้างตามที่ออกแบบไว้ ทำการต่อวงจร และอุปกรณ์ต่าง ๆ เข้าด้วยกัน โดยให้ อาดูโน่เป็นตัวกลางคอยสั่งการสเต็ปปิ้งมอเตอร์ และคอยรับข้อมูลรูปภาพหรือจีโค้ดจากคอมพิวเตอร์ด้วย

4.3 ทดสอบการใช้งานของเครื่องด้วยการลากเส้น เพื่อวัดผลจากการหาค่าความละเอียดและความผิดพลาดของเครื่องพล็อตเตอร์

4.4 ดัดแปลงแก้ไขส่วนที่ผิดพลาด

5 สรุปผลการทำงาน

5.1 สรุปผลและความสามารถของเครื่องพล็อตเตอร์

5.2 เขียนรายงานการทำโปรเจกต์

5.3 เตรียมตัวนำเสนอผลงาน

1.4.1 ระยะเวลาที่ศึกษา

ตารางที่ 1.1 ตารางการปฏิบัติงาน

	รายการ	Sep /20	Oct /20	Nov /20	Dec /20	Jan /21	Feb /21	Mar /21	Apr /21
๑	ศึกษาเกี่ยวกับพื้นฐานของอิเล็กทรอนิกส์								
๒	คัดเลือกเลือกวัสดุอุปกรณ์ที่จะใช้								
๓	ออกแบบสิ่งประดิษฐ์								
๔	สร้างเครื่องพล็อตเตอร์ ทดสอบ ดัดแปลงและแก้ไข								
๕	ทดสอบและประเมินประสิทธิภาพของเครื่องพล็อตเตอร์								
๖	สรุปผล								
๗	จัดทำรายงาน								
๘	นำเสนอผลงาน								

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ในด้านความรู้และประสบการณ์ต่อตัวนิสิตเอง

1. ได้รับความรู้ความเข้าใจฟิสิกส์เกี่ยวกับอิเล็กทรอนิกส์ วงจร ในสเต็ปปิ้งมอเตอร์ และความรู้ในการนำไปประยุกต์ใช้จริง

2. ได้รับความรู้เกี่ยวกับการใช้งานและการเขียนโปรแกรม การควบคุม การรับและส่งข้อมูล

1.5.2 ความรู้ ความเข้าใจ ที่นำปสู่การแก้ไขปัญหาของสังคมหรือสภาพแวดล้อม

1. ได้ต้นแบบในการสร้างต้นแบบของเครื่องพล็อตเตอร์ที่มีอิสระในการเคลื่อนที่แบบที่ยังไม่มีในตลาด

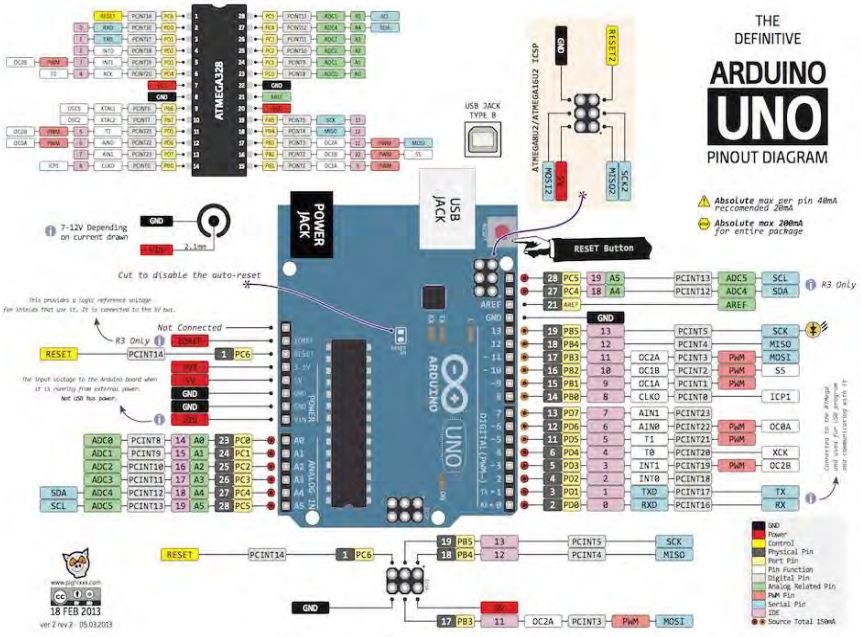
บทที่ 2

ทฤษฎีที่เกี่ยวข้องและอุปกรณ์ที่สำคัญ

2.1 อุปกรณ์ที่สำคัญ

2.1.1 ไมโครคอนโทรลเลอร์ อาดูโน่ (Microcontroller Arduino)

อาดูโน่ (Arduino) เป็นไมโครคอนโทรลเลอร์บอร์ดแบบสำเร็จที่ถูกพัฒนาในหลายด้าน เช่น การเชื่อมต่อของบอร์ดเข้ากับคอมพิวเตอร์และการอัพโหลดโปรแกรมนั้นทำได้ง่ายขึ้นอย่างมากด้วยพอร์ตการเชื่อมต่อด้วย ไมโครยูเอสบี (micro-USB) และมีการพัฒนาการรับและส่งออกสัญญาณ (Input-output) ทั้งแบบดิจิตอลและอนาล็อค สามารถใช้กับเซ็นเซอร์ได้หลากหลาย ทั้งยังมีการส่งสัญญาณแบบพิเศษ เช่นการส่งแบบ พัลส์วิตมอดูเลชั่น (PWM) คือการส่งสัญญาณที่ความถี่คงที่แบบสวิตช์ ทำให้การควบคุมสัญญาณมีโดยปรับขอบเขตการทำงาน (duty cycle) ได้ตั้งแต่ 0 ถึง 100 เปอร์เซ็นต์ จึงทำให้อาดูโน่สามารถประยุกต์ใช้กับงานได้หลากหลายตามแต่อุปกรณ์ที่นำมาต่อ เช่นการนำไปวัดค่าความชื้น วัดอุณหภูมิ หรือการนำเอาอาดูโน่ไปควบคุมมอเตอร์ เป็นต้น อาดูโน่จะสั่งการด้วยภาษาซีพลัสพลัส ที่นักพัฒนาส่วนใหญ่รู้จักเป็นอย่างดีและมีการพัฒนาเป็นแบบโอเพ่นซอร์ส คือมีการเปิดเผยหลักการทำงาน ตัวอย่างโปรแกรมให้บุคคลภายนอกสามารถนำไปใช้งาน ศึกษา แก้ไขได้อย่างเสรี ทำให้เกิดกลุ่มผู้ร่วมกันเป็นชุมชนผู้พัฒนาต่อยอดที่แข็งแรง แพร่หลาย และที่สำคัญอาดูโน่มีราคาที่ไม่แพงด้วย ขึ้นอยู่กับรูปแบบงานที่ใช้ มีราคาตั้งแต่หลักร้อยบาทถึงหลักพันต้นๆ ดังนั้นผู้ทำโครงการจึงเลือกที่จะใช้ในการสร้างเครื่องพล็อตเตอร์จากข้อดีที่กล่าวข้างต้น



รูปที่ 2.1 พินเอาท์ไดรอแกรมของไมโครคอนโทรลเลอร์อาดูโน่ รุ่น UNO

2.1.2 สเต็ปปิ้งมอเตอร์ NEMA17 HS4401

สเต็ปปิ้งมอเตอร์เป็นมอเตอร์ไฟฟ้าชนิดนึงที่ขับเคลื่อนด้วยสัญญาณพัลส์ที่มากระตุ้นขดลวดภายในมอเตอร์หรือเฟส โดย 1 พัลส์จะทำให้มอเตอร์ขยับไป 1 สเต็ป (step) ซึ่งการขยับในแต่ละสเต็ปจะขับเคลื่อนได้ 1, 1.5, 1.8 องศาขึ้นกับความสามารถของมอเตอร์ตัวนั้น ๆ ซึ่งมุมที่มอเตอร์ขยับไปนี้เรียกว่ามุมสเต็ป (step angle)

เมื่อลองเปรียบเทียบสเต็ปปิ้งมอเตอร์กับมอเตอร์รูปแบบอื่น พบว่าสเต็ปปิ้งมอเตอร์จะให้แรงบิดที่น้อยกว่ามอเตอร์กระแสตรงเมื่อมีขนาดความต่างศักย์เท่ากันแต่ว่าจะให้ความละเอียดในการหมุนที่สูงกว่า หรือเมื่อเปรียบเทียบกับเซอร์โวมอเตอร์สเต็ปปิ้งมอเตอร์จะมีราคาที่ถูกและใช้งานได้ง่ายกว่าแม้ว่าแรงบิดและความเสถียรจะน้อยกว่าเป็นต้น คุณสมบัตินี้เป็นจุดเด่นนี้ทำให้ผู้ทำโครงการได้เลือกใช้มอเตอร์ชนิดนี้ในการสร้างเครื่องพล็อตเตอร์

ผู้ทำโครงการได้เลือกใช้ Nema17HS4401 จัดอยู่ในหมวดไบโพลาร์สเต็ปปิ้งมอเตอร์ 2 อัน ซึ่งมีคุณสมบัติต่าง ๆ ที่สำคัญดังนี้

- ความต่างศักย์ : 12 โวลต์
- มุมสเต็ป : 1.8 องศา
- แรงบิด : 40 นิวตันเซนติเมตร
- จำนวนเฟส : 4
- น้ำหนัก : 280 กรัม

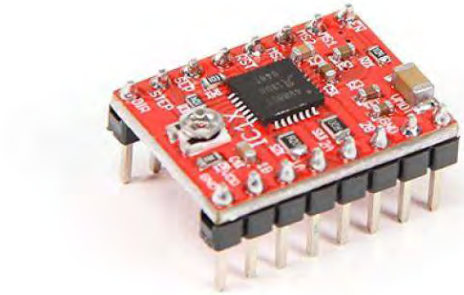
รูปที่ 2.2 สเต็ปปิ้งมอเตอร์17HS4401 12 V.

2.1.3 สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ A4988 (A4988 stepping motor driver)

สเต็ปปิ้งมอเตอร์จะไม่สามารถนำไปเชื่อมต่อกับอาดูโน่ตรง ๆ ได้จำเป็นต้องต่อวรจรเพื่อช่วยขับเช่น วงจรส่งกระแส วงจรควบคุมสัญญาณเอ็ช-บริดจ์ซึ่งช่วยควบคุมทิศทางการหมุนและป้องก้นกระแสตกค้างที่เกิดจากการหมุนของมอเตอร์ซึ่งอาจสร้างความเสียหายต่ออาดูโน่ได้ ทั้งหมดนี้ทำให้ยากต่อการเชื่อมต่อ จึงเกิดเป็นอุปกรณ์ขนาดเล็กที่เรียกว่า สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์

สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์หรือตัวช่วยขับมอเตอร์ คืออุปกรณ์ไอซีขนาดเล็กที่ภายในจะบรรจุวงจรต่าง ๆ ที่จำเป็นต่อการขับสเต็ปปิ้งมอเตอร์ ทำให้สะดวกในการใช้งาน โครงการนี้ผู้ทำโครงการเลือกใช้ A4988 สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์จำนวน 2 ตัวในการขับสเต็ปปิ้งมอเตอร์ เนื่องจากเป็นโมดูลที่ได้รับความนิยมและเพียงพอต่อการใช้งานทำให้นำไปประยุกต์ใช้ได้หลากหลายทั้งงานหุ่นยนต์หรืองานทางด้านเครื่องพิมพ์เป็นต้น มีคุณสมบัติที่สำคัญดังนี้

- ความต่างศักย์ที่รับได้ : 8 ถึง 35 โวลต์
- ความสามารถในการปรับไมโครสเต็ป : full, 1/2, 1/4, 1/8, 1/16
- กำลังขับ : 24 วัตต์



รูปที่ 2.3 สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ A4988

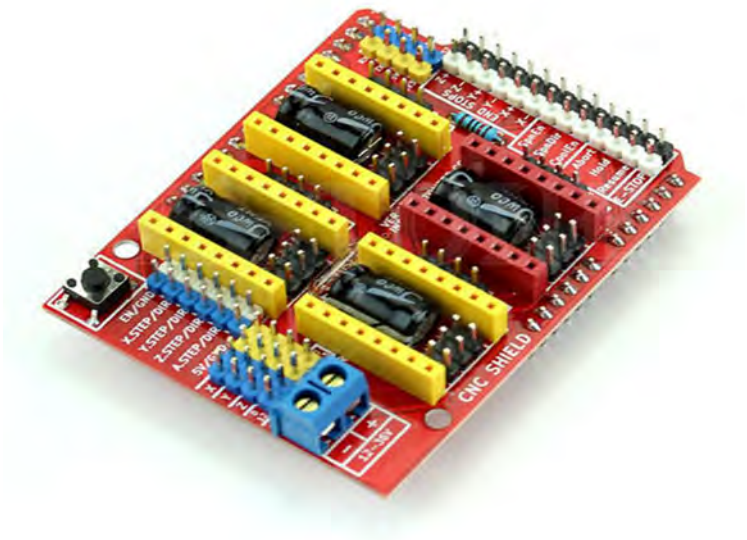
2.1.4 มู่เล่ย์และสายพานจีทีสอง

เป็นส่วนประกอบเล็ก ๆ ที่มีความสำคัญอย่างมากในเครื่องซีเอ็นซี เครื่องพิมพ์สามมิติและพล็อตเตอร์ ช่วยในการกำหนดความละเอียด

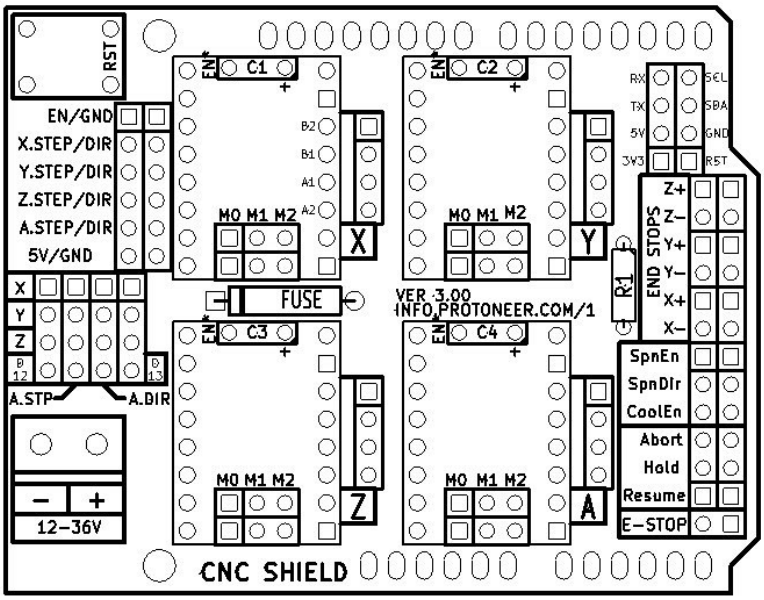
รูปที่ 2.4 (ก) มู่เล่ย์ 20 ฟัน (ข) สายพาน GT2

2.1.5 อาดูโน่ ซีเอ็นซีชิลด์ (Arduino CNC shield)

เป็นอุปกรณ์ต่อขยายที่เพิ่มความสามารถให้กับไมโครคอนโทรเลอร์อาดูโน่ เพิ่มความสะดวกในการใช้งานคู่กับสเต็ปปิ้งมอเตอร์และสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ มีหน้าที่ในการจัดการความเป็นระเบียบและกำหนดพินต่าง ๆ ในการเขียนโปรแกรม มีตัวเก็บประจุเพื่อช่วยในการหมุนมอเตอร์ อุปกรณ์นี้มีความแตกต่างกับสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์เพราะมันไม่ได้ช่วยขับมอเตอร์แต่อย่างใด แต่ใช้เพื่อเพิ่มความง่ายต่อการใช้งานเท่านั้น



รูปที่ 2.5 อาดูโน่ ซีเอ็นซีชิลด์ (Arduino CNC shield)



รูปที่ 2.6 อาดูโน่ซีเอ็นซีชิลด์ พินเอาท์ (Arduino CNC Shield PINOUT)

2.1.6 ไมโครเซอร์โวมอเตอร์ 6 โวลต์ 180 องศา (Micro servo motor)

เป็นอุปกรณ์ที่สำคัญในส่วนของการสร้างมือจับปากกา (Pen Holder) ประกอบไปด้วย 4 ส่วนหลัก คือ มอเตอร์กระแสตรง ชุดเฟืองทดรอบ (Gear system) โพเทนเชียลมิเตอร์ (Potential Meter) และวงจรควบคุม (Control Electronics) หลักการทำงานเริ่มจากการที่วงจรควบคุมได้รับข้อมูลในลักษณะของ พัลส์วิดมอดูเลชั่น เมื่อได้รับมาแล้ววงจรควบคุมจะทำให้มอเตอร์หมุนในทิศทางตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา เมื่อมอเตอร์เริ่มหมุน ตัวโพเทนเชียลมิเตอร์ที่ติดอยู่กับชุดเฟืองมอเตอร์จะตรวจสอบตำแหน่งที่มอเตอร์หมุนไป โดยหากตรวจพบว่าตำแหน่งที่มอเตอร์หมุนเริ่มใกล้กับองศาที่ผู้ใช้กำหนด วงจรส่วนควบคุมจะเริ่มสั่งให้มอเตอร์หมุนช้าลงเพื่อให้หมุนเข้าใกล้องศาที่กำหนดได้มากที่สุด การทำงานแบบนี้จำเป็นต้องเป็นเซอร์โวมอเตอร์แบบ 180 องศาเท่านั้นเพราะไมโครเซอร์โวแบบ 360 องศาสามารถควบคุมได้แค่ทิศทางและความเร็วในการหมุน

รูปที่ 2.7 ไมโครเซอร์โวมอเตอร์ SG90 180 Degree

2.1.7 ล้อออมนิ ไดเรกชันนอล (Omni directional wheel)

เนื่องจากในโครงการนี้เครื่องเพล็อตเตอร์จะมีลักษณะพิเศษในการเคลื่อนที่ไปทั้งแกน เอกซ์และแกนวาย ดังนั้นล้อที่ใช้จำเป็นที่จะต้องมีลักษณะพิเศษเพื่อที่จะลดผลของแรงเสียดทานที่จะเกิดขึ้นหากใช้ล้อรูปแบบปกติ ล้อออมนิ ไดเรกชั่นนอล มีความพิเศษคือ เป็นล้อที่มีลูกกลิ้งติดอยู่รอบ ๆ ล้อและทำให้สามารถเคลื่อนที่ในแนวตั้งฉากกับการเคลื่อนที่เดิมได้ และยังสามารถทำการเคลื่อนที่แบบ โฮโลโนมิกส์ไดรฟ์ซึ่งคือการเคลื่อนที่ที่สามารถไปได้ในทุกทิศทาง

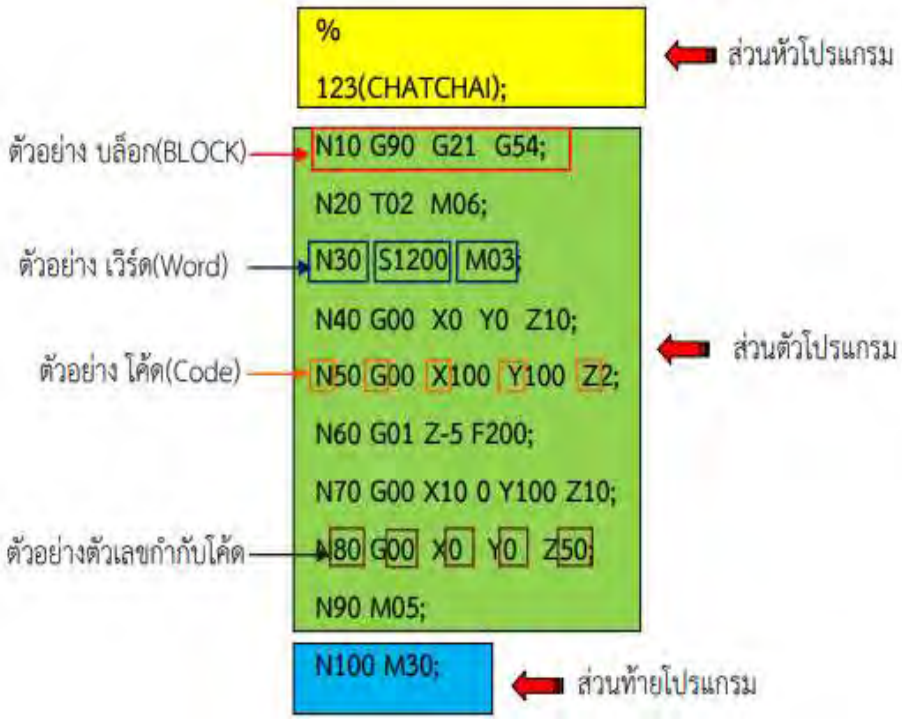
รูปที่ 2.8 ล้อออมนิ ไดเรกชันนอล (Omni directional wheel)

2.2 ทฤษฎีที่เกี่ยวข้อง

2.2.1 ส่วนประกอบพื้นฐานของโปรแกรมซีเอ็นซีหรือจีโค้ด (G code)

1) ส่วนประกอบพื้นฐานของโปรแกรมเอ็นซีจะแยกออกเป็น 3 ส่วนหลัก ๆ ซึ่งประกอบไปด้วย

- ส่วนหัวของโปรแกรม ส่วนนี้จะเป็นส่วนกำหนดรายระเอียดของโปรแกรม การกำหนดค่าเริ่มต้นเช่นค่าความไวของการส่งสัญญาณโปรแกรม จุดเริ่มต้นของงานเป็นต้น
- ส่วนตัวโปรแกรม ในส่วนนี้จะประกอบไปด้วย จีโค้ดและโค้ดอื่น ๆ มีรูปแบบเป็นบล็อก (block) หรือบรรทัด 1 บล็อคจะประกอบไปด้วย คำสั่ง (word) ซึ่งมีลักษณะเป็นตัวหนังสือและตัวเลขนั่นเอง
- ส่วนท้ายของโปรแกรม ส่วนนี้จะเป็นส่วนคำสั่งจบการทำงาน



รูปที่ 2.9 แสดงส่วนประกอบพื้นฐานของจีโค้ด

2) คำสั่งพื้นฐานที่สำคัญของโปรแกรมซีเอ็นซี

และยังมีตัวอย่างที่สำคัญอื่น ๆ เช่น โค้ดกำหนดตำแหน่งระยะทางและความยาว ได้แก่ X, Y, Z อัตราการป้อนได้แกโค้ด F และเวลาได้แกโค้ด P เป็นต้น

2.2.2 การหาค่าความละเอียดของเครื่อง

การคำนวณความละเอียดของเครื่องพล็อตเตอร์ที่จะสร้างนั้นนอกจากจะต้องดูมุมสเต็ปของมอเตอร์แล้ว ยังต้องดูจำนวนฟันของรอกและสายพานที่ใช้ร่วมกันด้วย เมื่อนำมาคิดเป็นสมการจะได้ความสัมพันธ์ของค่าสเต็ปต่อมิลลิเมตรดังนี้

$$\text{ค่าสเต็ปต่อมิลลิเมตร} = \frac{S_{rev} \cdot f_m}{p \cdot N_t} \qquad 2.1$$

S_{rev} คือ จำนวนสเต็ปต่อรอบการหมุนของมอเตอร์หาได้จากการเอา 360/มุมสเต็ป

f_m คือ ไมโครสเต็ปของการหมุนที่สามารถปรับได้จากสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์

p คือ ระยะพิตช์หรือระยะห่างระหว่างร่องเกลียวของสายพาน

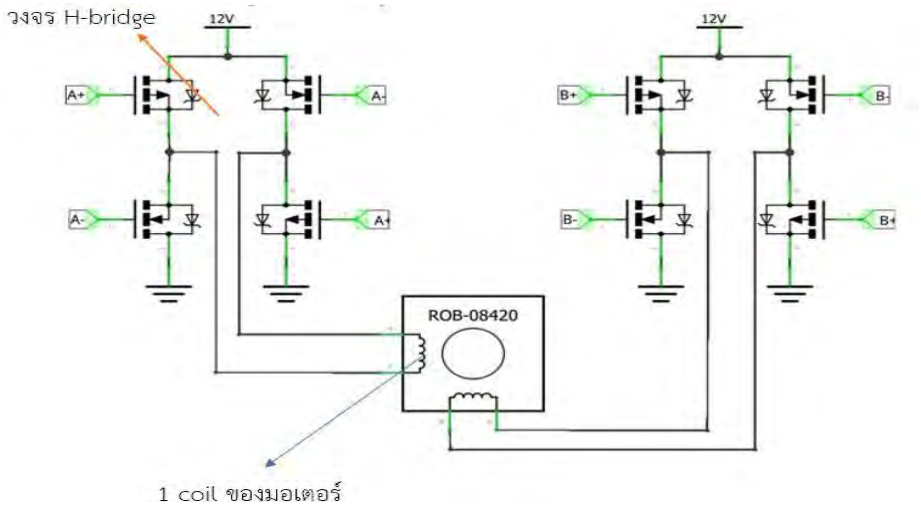
N_t คือ จำนวนฟันของรอก

ยกตัวอย่างสเต็ปมอเตอร์ที่มีมุมสเต็ปเป็น 1.8 องศา กระตุ้นแบบฮาลฟ์สเต็ป และใช้รอกที่มีฟัน 20 ซี่ สายพานระยะพิตช์เป็น 2 มม. เมื่อคำนวณจะได้ 10 สเต็ป/มม. หรือมีค่าความละเอียดเป็น 0.1 มิลลิเมตรนั่นเอง

2.2.3 วงจรเอ็ช-บริดจ์ (H-bridge)

เอ็ช-บริดจ์ เป็นวงจรอิเล็กทรอนิกส์ที่สำคัญในการขับมอเตอร์ มักจะใช้ในการขับมอเตอร์ที่ต้องการหมุนในทิศทางสลับกันไปสลับกันมาตลอดเวลาเช่นมอเตอร์กระแสตรงหรือสเต็ปปิ้งมอเตอร์ วงจรนี้จะอยู่ในสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์มีหน้าที่ในปล่อยกระแสไฟเพื่อควบคุมทิศทางการหมุนของมอเตอร์จากการใช้ทรานซิสเตอร์ พร้อมทั้งทำหน้าที่กำหนดทิศทางการไหลของกระแสจากไดโอดเพื่อป้องกันกระแสตกค้างที่เกิดจากการหมุนของมอเตอร์ซึ่งอาจทำให้อาดูโน่เกิดความเสียหายได้

วงจรเอ็ช-บริดจ์ 1 วงจรจะสามารถควบคุมการจ่ายไฟให้ขดลวดได้ 1 ขด ดังนั้นในการที่จะขับมอเตอร์ 1 ตัว จำเป็นต้องใช้ 2 วงจร



รูปที่ 2.10 ลักษณะการขับสสเต็ปปิ้งมอเตอร์ด้วยวงจร H-bridge สองวงจร

2.2.4 ชนิดและหลักการการกระตุ้นสเต็ปปิ้งมอเตอร์

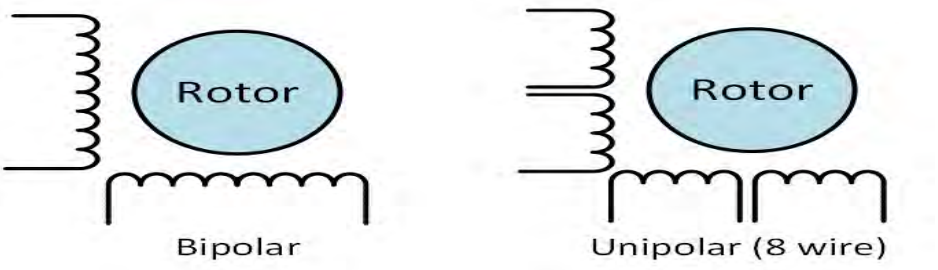
ภายในมอเตอร์ชนิดนี้จะประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ส่วนที่หมุนหรือโรเตอร์ (Rotor) และส่วนที่อยู่กับที่หรือสเตเตอร์ (Stator) วัสดุที่ใช้สร้างสองส่วนนี้จะแตกต่างกันทำให้เกิดสเต็ปปิ้งมอเตอร์ 3 รูปแบบคือ

- แบบแม่เหล็กถาวร (Permanent magnet) แบบนี้จะมีส่วนสเตเตอร์พันขดลวดไว้หลายๆขด และมีส่วนที่โรเตอร์สร้างจากแม่เหล็กถาวร เมื่อป้อนไฟฟ้ากระแสตรงให้กับขดลวดสเตเตอร์ จะทำให้เกิดแรงแม่เหล็กไฟฟ้าไปสร้างแรงผลักทำให้เกิดการหมุน และสามารถยึดโรเตอร์ให้นิ่งในขณะที่ไม่มีกระแสไฟฟ้าได้ด้วย
- แบบแปรค่ารีลักแตนซ์ (Variable reluctance) สเต็ปปิ้งมอเตอร์รูปแบบนี้ทำจากสารเฟอร์โรแมกเนติก ทำมาจากเหล็กผสมกับซิลิคอน มีลักษณะเป็นเฟืองฟันเลื่อย เมื่อป้อนไฟเข้าไปในขดลวดจะทำให้มอเตอร์เกิดแรงหมุน ซึ่งจะไปหมุนโรเตอร์ให้ไปในเส้นทางที่ค่ารีลักแตนซ์ต่ำหรือหมุนไปยังจุดที่ความต้านทานแม่เหล็กต่ำที่สุด สเต็ปปิ้งมอเตอร์แบบนี้จะให้ความเร็วรอบและความละเอียดที่สูงกว่า
- แบบผสม ปัจจุบันสเต็ปปิ้งมอเตอร์ส่วนใหญ่จะเป็นรูปแบบนี้หมด คือมีสเตเตอร์คล้ายแบบแปรค่ารีลักแตนซ์ และมีสารแม่เหล็กกำลังสูงหุ้มปลาย จากการควบคุมหมวกแม่เหล็ก ทำให้ได้แรงบิดที่สูงโดยใช้พลังงานต่ำและมีแรงฉุดยึดให้โรเตอร์นิ่งในตอนไม่จ่ายไฟซึ่งเป็นข้อดีของแบบแรก ทั้งยังได้มุมต่อการหมุนแต่ละครั้งที่น้อยและแม่นยำซึ่งเป็นข้อดีของแบบที่สอง

ในปัจจุบันการแบ่งสเต็ปปิ้งมอเตอร์นั้นได้พิจารณาจากการพันขดลวด การต่อสายออกมาใช้งาน และวงจรขับสามารถแบ่งได้ 2 ชนิดหลัก คือ

- ไบโพลาร์ มีลักษณะการพันขดลวดดังแสดงในรูป (2.11) สเต็ปปิ้งมอเตอร์แบบนี้มีขดลวด 2 ขด อาจจะเรียกว่าสเต็ปปิ้งมอเตอร์แบบ 2 เฟสก็ได้
- ยูนิโพลาร์ มีลักษณะการพันขดลวดดังแสดงในรูป (2.11) การขับจะต้องป้อนสัญญาณเข้าที่ขั้วของมอร์เตอร์คล้ายไบโพลาร์แต่จะมีจำนวนขดลวดที่มากกว่าอาจจะมี 4 ,5 หรือ 8 เลยก็ได้

ทั้งสองแบบมีข้อดีข้อเสียต่างกัน เนื่องจากขดลวดในมอเตอร์แบบยูนิโพลาร์มีจำนวนมากกว่าถึงสองเท่าในพื้นที่เท่ากัน แต่กลับสามารถใช้งานได้แค่ครึ่งนึง ดังในรูป (2.11) ทำให้ในการทำงานแรงบิดที่ได้มีค่าน้อยกว่าแบบไบโพลาร์เกือบ 50 เปอร์เซ็นต์ แม้ว่าจะได้แรงบิดที่มากกว่าแต่ไบโพลาร์สเต็ปปิ้งมอเตอร์จำเป็นต้องใช้วงจรเอ็ช-บริดจ์ เพื่อป้องกันความเสียหายดังที่ได้กล่าวไปในหัวข้อ 2.2.3



รูปที่ 2.11 ขดลวดของสเต็ปปิ้งมอเตอร์ชนิด ไบโพลาร์ และ ยูนิโพลาร์

การจะทำให้สเต็ปปิ้งมอเตอร์หมุนนั้นต้องจ่ายไฟเข้าไปกระตุ้นขดลวด ดังนั้นการจ่ายไฟให้แต่ละขดลวดในรูปแบบที่แตกต่างกันก็จะให้ผลการหมุนของมอเตอร์ที่แตกต่างกันด้วย โดยมีการจ่ายกระแสรูปแบบหลัก 3 แบบดังนี้

1) การจ่ายไฟกระตุ้นแบบ ฟูล-สเต็ป 1 เฟส (full-step 1 phase)

การจ่ายไฟแบบนี้ เป็นการจ่ายไฟไปกระตุ้นขดลวดของสเต็ปปิ้งมอเตอร์ทีละเฟส และจะเรียงลำดับกันไปตามตารางที่ (2.1) กำหนดให้ 1 คือการจ่ายกระแสไฟ และ 0 คือไม่จ่ายกระแสไฟ

ตารางที่ 2.1 การจ่ายกระแสไฟเพื่อให้สเต็ปปิ้งมอเตอร์หมุนแบบ ฟูล-สเต็ป

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

2) การจ่ายไฟกระตุ้นแบบ ฟูล-สเต็ป 2 เฟส (full-step 2 phase)

การจ่ายไฟไปกระตุ้นแบบนี้ เป็นการจ่ายไฟไปกระตุ้นขดลวดทีละสองเฟส และเรียงกันไปคล้ายแบบแรกดังแสดงในตารางที่ (2.2) การจ่ายไฟในลักษณะนี้จะใช้กำลังไฟที่มากกว่าแบบแรก แต่จะได้แรงบิดที่มากกว่า

ตารางที่ 2.2 การจ่ายกระแสไฟเพื่อให้สเต็ปปิ้งมอเตอร์หมุนแบบ ฟูล-สเต็ป 2 เฟส

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

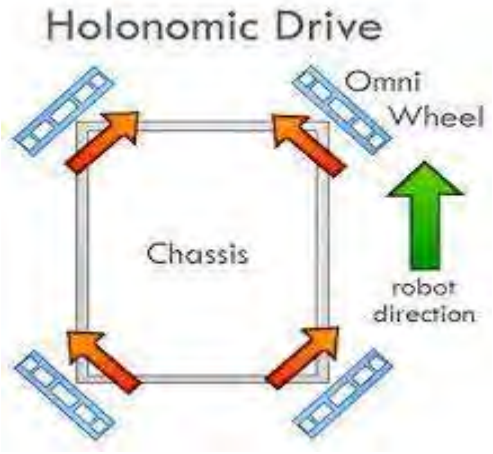
3) การจ่ายไฟกระตุ้นแบบ ฮาล์ฟสเต็ป (half-step) คือการกระตุ้นแบบฟูล-สเต็ป 1 เฟสและ 2 เฟสเรียงลำดับกันไป ตามตารางที่ (2.3) การกระตุ้นแบบนี้ข้อเด่นคือจะได้ความละเอียดที่มากกว่าสองแบบแรก แต่ว่าจะต้องใช้ไฟในการกระตุ้นมากขึ้นเพื่อที่จะให้ได้ระยะเท่าเดิม

ตารางที่ 2.3 การจ่ายกระแสไฟเพื่อให้สเต็ปปิ้งมอเตอร์หมุนแบบ ฮาล์ฟ-สเต็ป

เฟสที่ / สเต็ปที่	1	2	3	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

2.2.5 การเคลื่อนที่แบบ โฮโลโนมิค ไดรฟ์ (Holonomic drive)

การเคลื่อนที่แบบ โฮโลโมนิค ไดรฟ์ คือแนวคิดของการเคลื่อนที่ของสิ่งประดิษฐ์ที่สามารถเคลื่อนที่ได้ในทุกทิศทาง มีลักษณะคือ ใช้ล้อออมนิ ไดเรกชันนอล 4 ล้อ โดยล้อที่อยู่ติดกันจะวางในลักษณะตั้งฉากกันและล้อที่อยู่ในทิศตรงข้ามกันจะวางขนานกัน การวางล้อในลักษณะนี้ทำให้เครื่องพล็อตเตอร์สามารถเคลื่อนที่ได้อย่างอิสระ



รูปที่ 2.12 การเคลื่อนที่แบบโฮโลโนมิกส์ ไดรฟ์ (Holonomic drive)

บทที่ 3

การออกแบบการทำงานและขั้นตอนการทำงาน

ขั้นตอนการดำเนินการแบ่งออกเป็น 6 ขั้นตอนดังต่อไปนี้

3.1 ศึกษาและทำความเข้าใจทฤษฎีที่เกี่ยวข้อง

3.1.1 เรียนรู้เกี่ยวกับการสร้าง คำสั่งต่าง ๆ ของ G-code

3.1.2 เรียนรู้เรื่องความละเอียดของเครื่องมือ การคำนวณความผิดพลาด

3.2 ศึกษาการทำงานของอุปกรณ์

3.2.1 ศึกษาหลักการทำงานของสเต็ปปิ้งมอเตอร์

3.2.2 ศึกษาการทำงานของมอเตอร์ไดรฟ์เวอร์และบอร์ดขับมอเตอร์

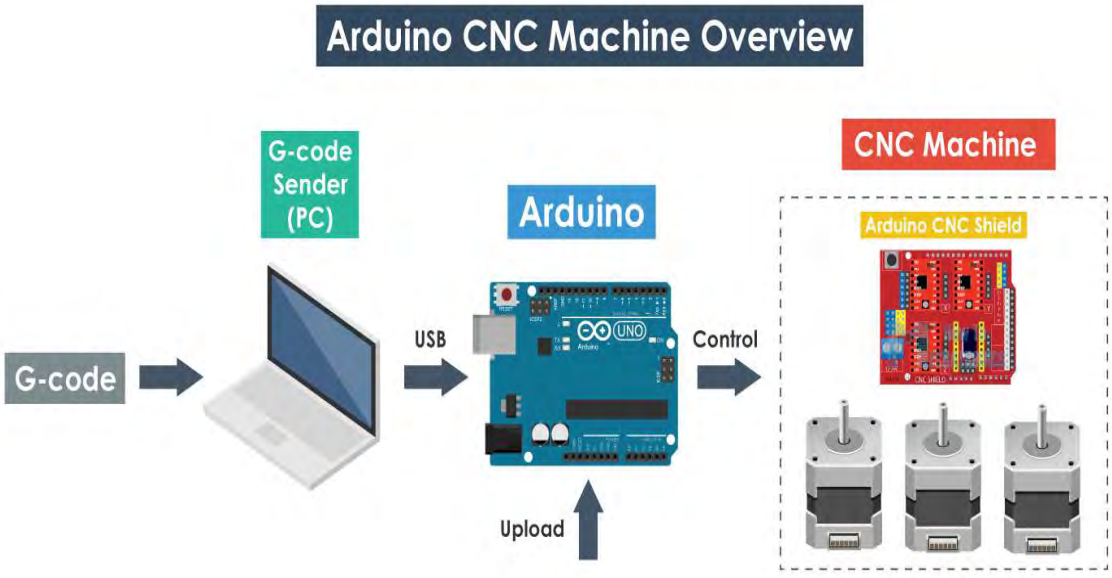
3.2.3 ศึกษาการใช้โปรแกรม Arduino IDE

3.3 ออกแบบอุปกรณ์

3.3.1 ออกแบบการต่อวงจรของสิ่งประดิษฐ์

3.3.2 ออกแบบรูปร่างและหลักการเคลื่อนที่ของสิ่งประดิษฐ์ การวางล้อ เพลา

3.3.3 ออกแบบอุปกรณ์จับปากกา

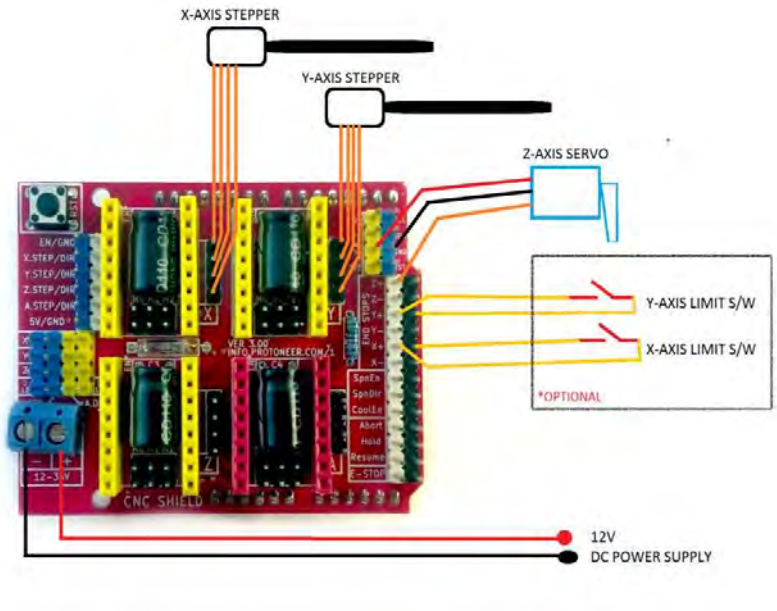


รูปที่ 3.1 แผนภาพโดยรวมของการสร้างสิ่งประดิษฐ์

3.4 การสร้างเครื่องพล็อตเตอร์ในรูปแบบที่สามารถเคลื่อนที่ได้

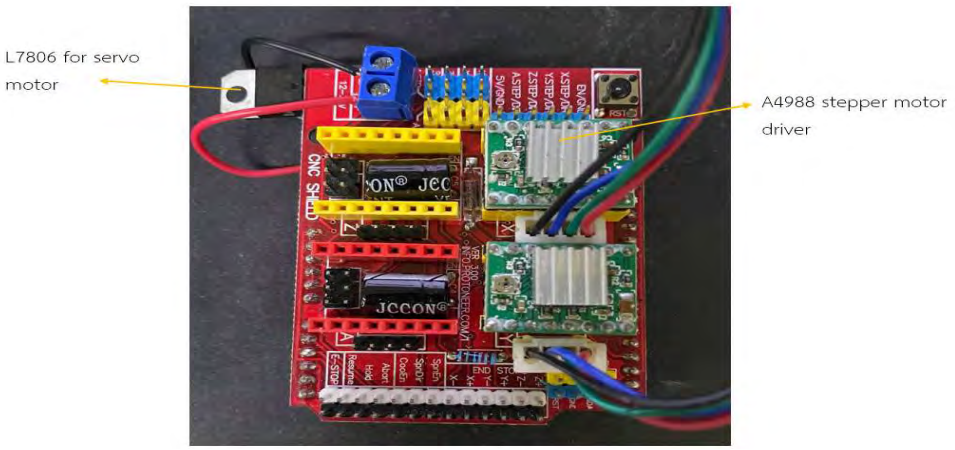
3.4.1 ต่อวงจรของอุปกรณ์

1) ทำการต่อไมโครคอนโทรลเลอร์อาดูโน่เข้ากับซีเอ็นซีชิลด์พร้อมทำการติดตั้งสเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์ตามรูปที่ (3.9)



รูปที่ 3.2 การต่อวงจรของบอร์ดซีเอ็นซีชิลด์และมอเตอร์แบบต่าง ๆ

เมื่อทำการต่อวงจรดังรูปและเพิ่ม A4988 สเต็ปปิ้งมอเตอร์ไดรฟ์เวอร์เข้าไปแล้ว ทำการเชื่อมกระแสไฟฟ้า 12 V จากไมโครคอนโทรเลอร์อาดูโน่เข้าที่ตัวรับของซีเอ็นซีชิลด์ เนื่องจากมอเตอร์ในแกน z ที่เราใช้เป็นเซอร์โวมอเตอร์ที่ต้องการความต่างศักย์ที่ 6 V ดังนั้นต้องทำการบัดกรี L7806 เข้าเพื่อแปลงไฟจาก 12 V เป็น 6 V ในการนำไปขับมอเตอร์นั่นเอง ในส่วนของ Optional มีไว้เพื่อเพิ่มสวิตช์ในการหยุดการทำงานของเครื่องฉุกเฉินในเครื่องซีเอ็นซีแบบปกติเมื่อปากกาจะหลุดจากกรอบเพื่อป้องกันความเสียหายที่อาจจะเกิดขึ้น

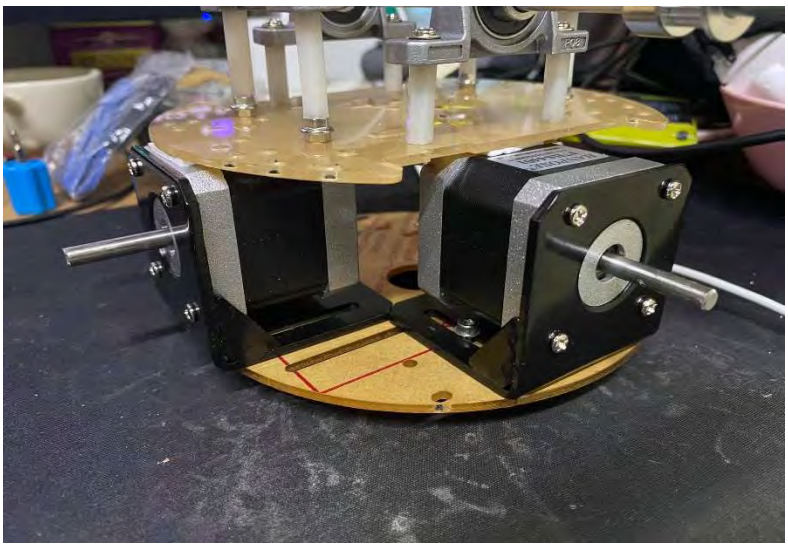


รูปที่ 3.3 ทำการบัดกรี L7806 และติดตั้ง stepper motor driver A4988

3.4.2 แนวคิดและการสร้างระบบการเคลื่อนที่ให้กับเครื่องพล็อตเตอร์

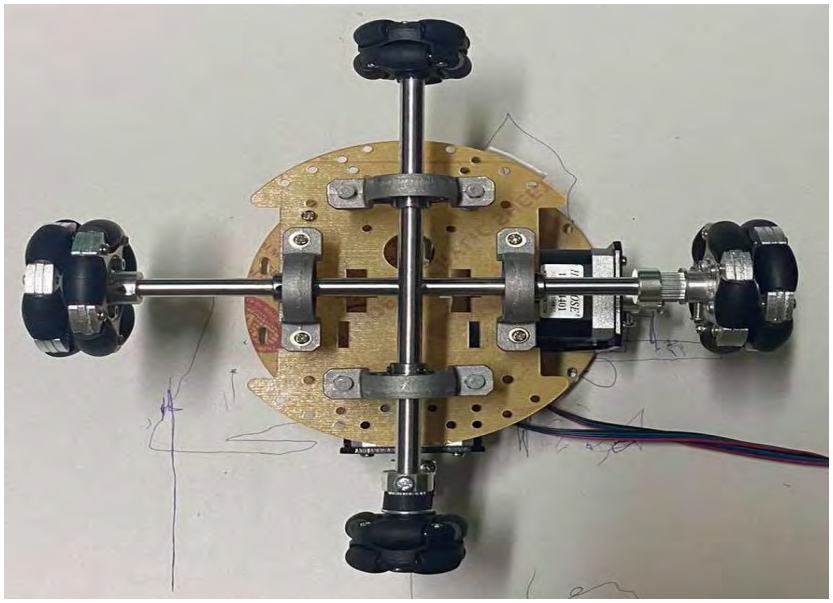
เนื่องจากเครื่องพล็อตเตอร์ส่วนใหญ่ในตลาดจะมีลักษณะเป็นเครื่องตั้งอยู่กับที่ แต่ในโครงการนี้ต้องการที่จะสร้างเครื่องพล็อตเตอร์ที่สามารถขยับได้ ดังนั้นต้องประยุกต์ใช้สเต็ปปิ้งมอเตอร์ในการขับเคลื่อนเพลาที่มีลักษณะแบ่งออกเป็น แกนเอกซ์และแกนวายทำให้เกิดความซับซ้อนในการสร้างสิ่งประดิษฐ์

1) ทำการติดตั้งสเต็ปปิ้งมอเตอร์เข้ากับฐานอคริลิคและทำการยึดให้แน่น โดยมีลักษณะพิเศษคือมอเตอร์สองตัวต้องตั้งฉากกันเนื่องจากทั้งสองตัวจะเป็นตัวควบคุมแกนเอกซ์และแกนวายนั่นเอง



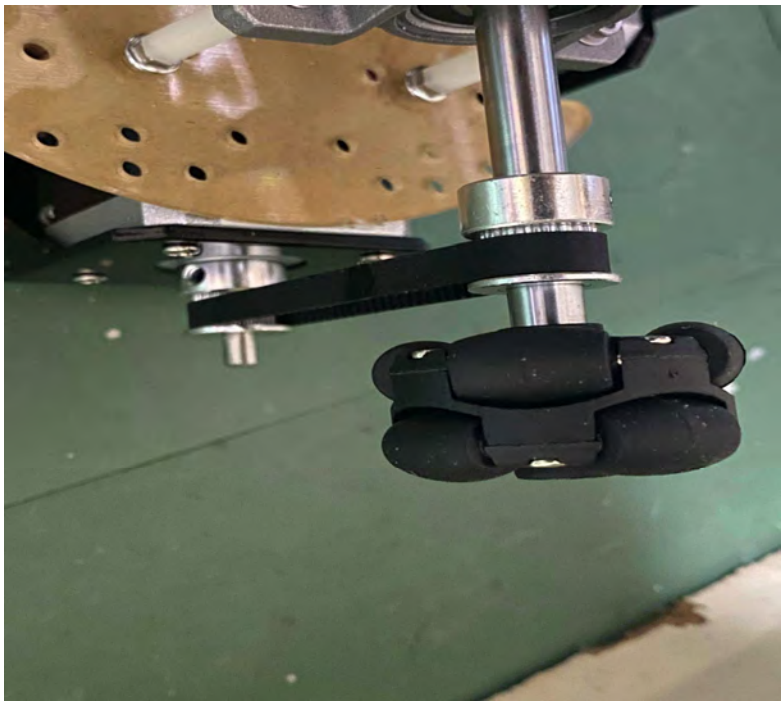
รูปที่ 3.4 การวางมอเตอร์ลักษณะตั้งฉากกัน

2) ทำการติดตั้งระบบเพลาที่มีลักษณะตั้งฉากซึ่งกันและกัน พร้อมกับเพลาแต่ละเส้นจะต้องขนานกับสเต็ปปิ้งมอเตอร์ แต่เนื่องจากเพลาสองเส้นไม่สามารถอยู่ในระนาบเดียวกันได้ ทำให้ต้องติดระบบเพลาที่ไม่ได้อยู่ในระนาบเดียวกัน



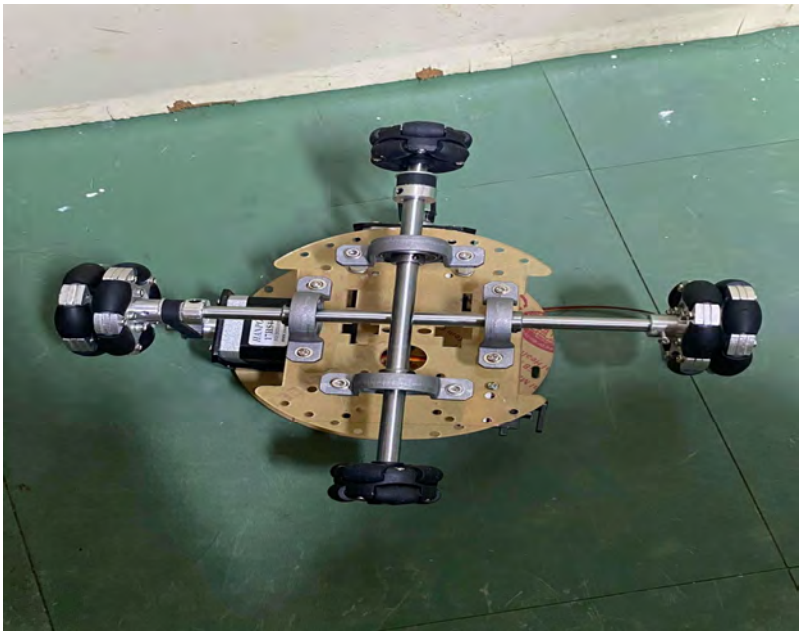
รูปที่ 3.5 การวางระบบเพลาลักษณะตั้งฉากกัน

3) ติดตั้งมู่เล่ย์และสายพานจีทีสองเพื่อให้สเต็ปปิ้งมอเตอร์ขับเคลื่อนระบบเพลา ด้วยการใช้ระบบการขนานกันของมู่เล่ย์



รูปที่ 3.6 การติดตั้งมู่เล่ย์เพื่อใช้ในการขับเคลื่อนเครื่องพล็อตเตอร์

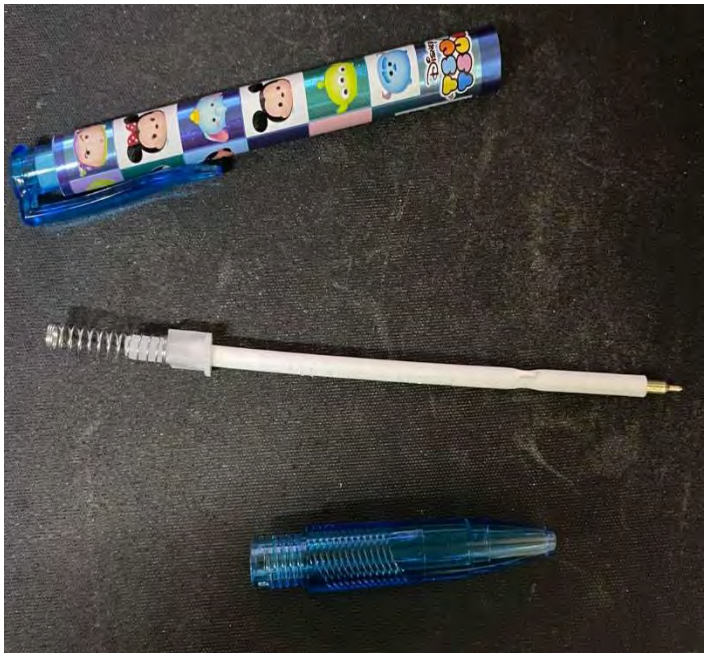
4) ติดตั้งระบบเพลาและล้อ โดยล้อที่ติดตั้งจะมีลักษณะเป็นสองล้อที่ไม่เท่ากันและทำการสอบเทียบค่าสเต็ปต่อมิลลิเมตรของแกนเอกซ์และแกนวายให้มีค่าเท่ากันมากที่สุดเท่าที่จะทำได้ ในที่นี้ล้อของแกนเอกซ์มีเส้นผ่าศูนย์กลางที่ 58 มิลลิเมตร และแกนวายมีเส้นผ่าศูนย์กลาง 38 มิลลิเมตร การติดตั้งล้อลักษณะนี้เพื่อจะนำแนวทางของการเคลื่อนที่แบบ โฮโลโนมิค ไดรฟ์ เพื่อให้เครื่องพล็อตเตอร์สามารถเคลื่อนที่ได้ทุกทิศทางนั่นเอง



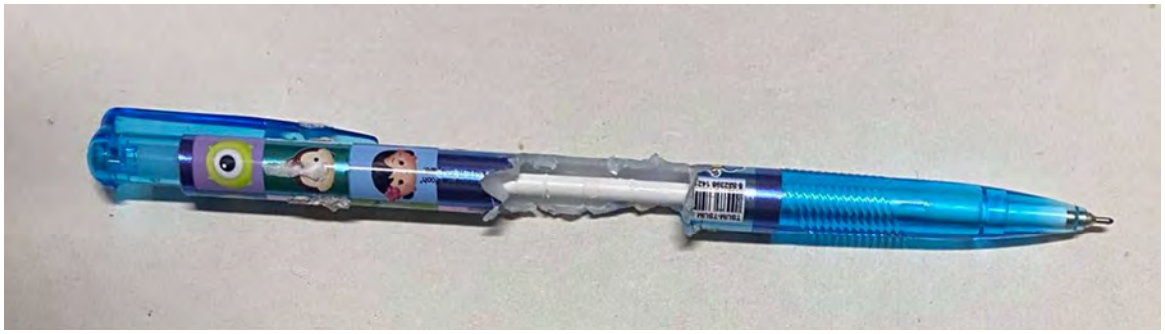
รูปที่ 3.7 ติดตั้งล้อตามหลักการฮาโลโนมิค ไดรฟ์

3.4.3 สร้างระบบปากกาจากเซอร์โวมอเตอร์

1) ระบบปากกา เป็นแบบประยุกต์ใช้อุปกรณ์ทั่วไปที่สามารถหาได้ โดยมีการควบคุมจากเซอร์โวมอเตอร์ 1 ตัว และปากกาที่หาได้ทั่วไป โดยมีการปรับแต่งเพื่อให้สปริงในปากการองรับแรงกดที่มาจากสเต็ปปิ้งมอเตอร์ เนื่องจาก ทำการเจาะปากกาเพื่อใช้เซอร์โวมอเตอร์ในการควบคุมการยกปากกาหลังจากนั้นทำการประกอบเข้ากับอุปกรณ์จับปากกา

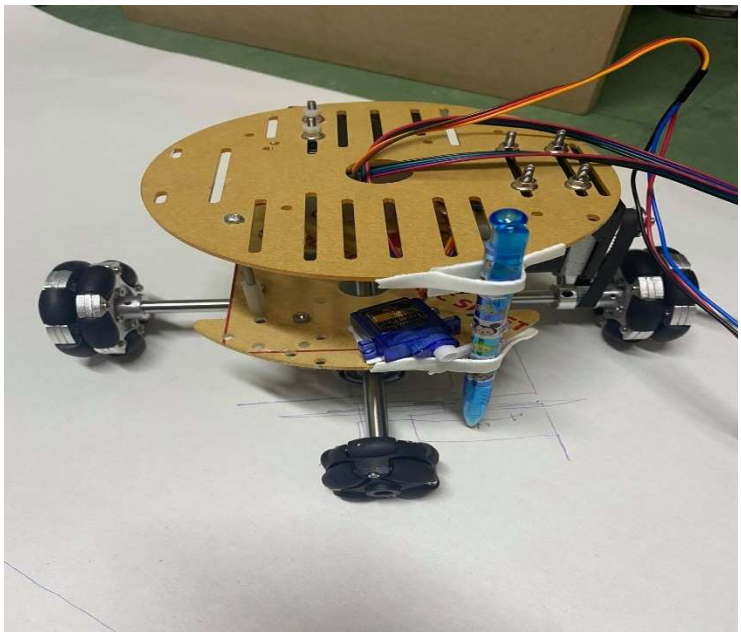


รูปที่ 3.8 การสลับสปริงและตัดปลายปากกาเพื่อให้สปริงต้านแรงกดปากกา



รูปที่ 3.9 การเจาะปากกาเพื่อนำเซอร์โวมอเตอร์มาควบคุมการเปิด ปิด

3.4.4 ประกอบแต่ละส่วนเข้าด้วยกันทำให้สิ่งประดิษฐ์เสร็จสมบูรณ์



รูปที่ 3.10 เครื่องพล็อตเตอร์บนกระดาษพร้อมวัดผล

ข้อมูลของเครื่องพล็อตเตอร์

มีน้ำหนักรวมประมาณ 800 กรัม

ขนาด กว้าง x ยาว ที่ 16 x 16 เซนติเมตร และ กว้าง x ยาว ของล้อ 22 x 22 เซนติเมตร

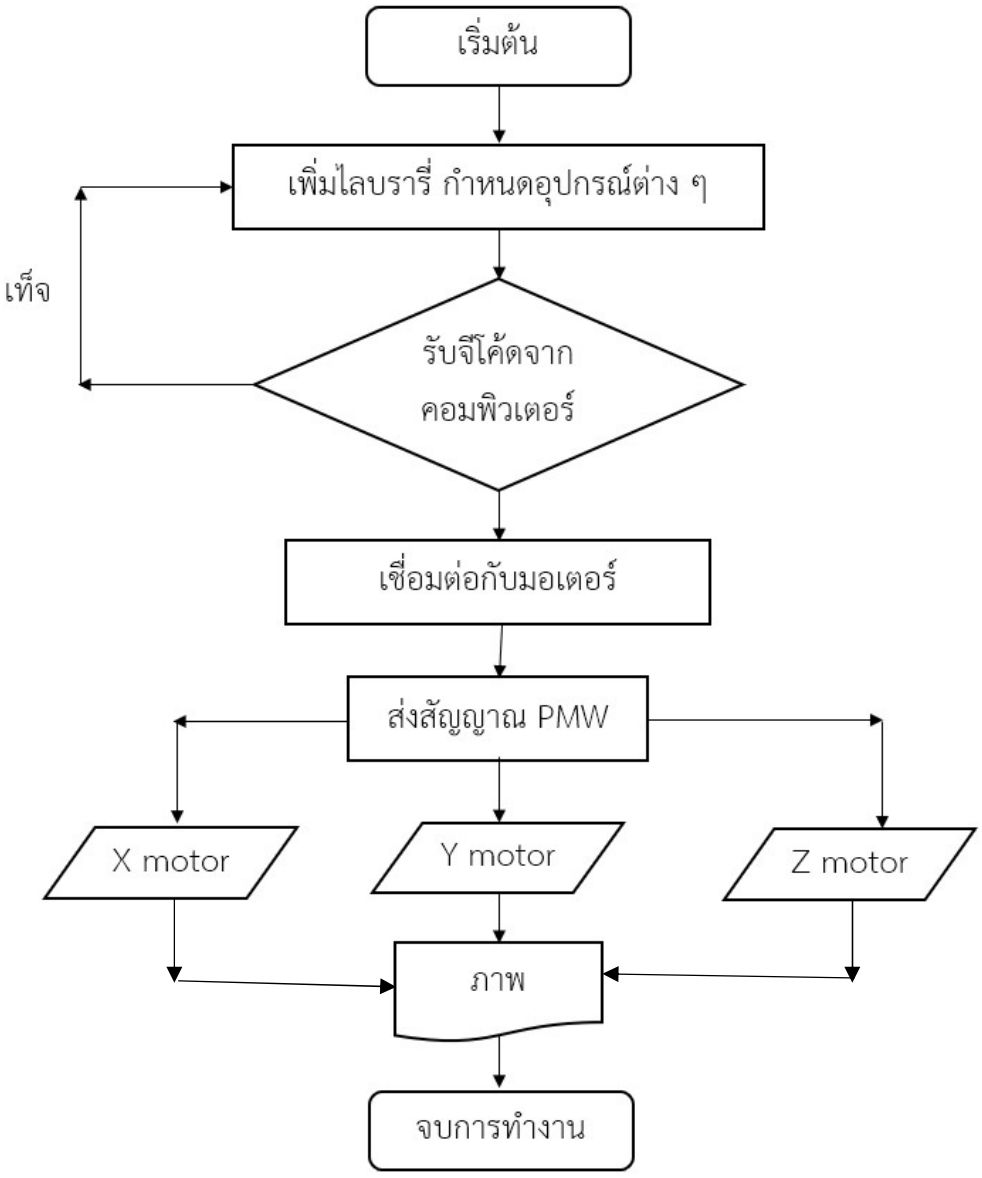
ล้อแกนเอกซ์มีขนาดเส้นผ่าศูนย์กลาง 5.8 เซนติเมตร และแกนวาย 3.8 เซนติเมตร

3.5 ปรับปรุงและแก้ไขโปรแกรมเพื่อให้อุปกรณ์ต่าง ๆ ที่ออกแบบสามารถทำงานได้ตามต้องการ

ในเรื่องเกี่ยวกับโปรแกรมจะแบ่งออกเป็น 2 ส่วนหลัก ๆ

3.5.1 ส่วนของการขับเคลื่อนมอเตอร์

ในส่วนนี้การทำงานจะเป็นไปตามโฟลวชาร์ตดังรูปต่อไปนี้



รูปที่ 3.11 ผังการทำงานของโปรแกรมสั่งการมอเตอร์

1) ทำการแก้ไขโอเพ่นซอร์สซอฟต์แวร์ชื่อ Grbl ให้พินตรงกับอุปกรณ์ที่เราต้องการใช้ และทำการเพิ่มส่วนที่ใช้ขยับมอเตอร์แกน z เนื่องจากเป็นเซอร์โวมอเตอร์ โดย Grbl เป็นโปรแกรมคุณภาพสูงในงานทางด้านซีเอ็นซี ตัวโปรแกรมได้ถูกสร้างในปี 2009 โดย Simen Svale Skogsrud โปรแกรมเป็นโอเพ่นซอร์สจึงได้รับอนุญาตให้พัฒนาและดัดแปลงแก้ไขเพื่อใช้ในงานลักษณะต่าง ๆ ซึ่งรูปแบบที่นำมาใช้จะเป็นการใช้คู่กับเซอร์โวมอเตอร์

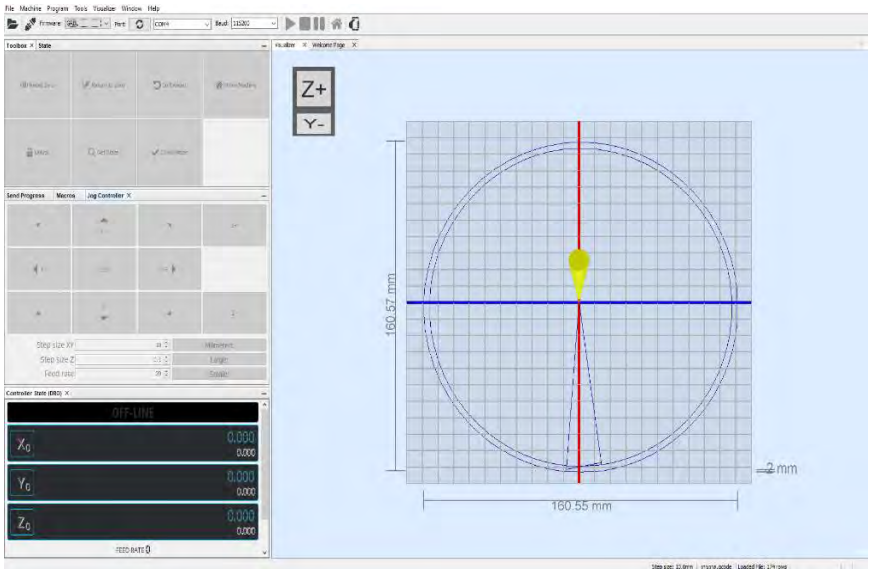
```
// Modal Group G1: Motion modes
#define MOTION_MODE_SEEK 0 // G0 (Default: Must be zero)
#define MOTION_MODE_LINEAR 1 // G1
#define MOTION_MODE_CW_ARC 2  // G2
#define MOTION_MODE_CCW_ARC 3  // G3
#define MOTION_MODE_PROBE_TOWARD 4 // G38.2 NOTE: G38.2, G38.3, G38.4, G38.5 must be sequential. See report_gcode_modes().
#define MOTION_MODE_PROBE_TOWARD_NO_ERROR 5 // G38.3
#define MOTION_MODE_PROBE_AWAY 6 // G38.4
#define MOTION_MODE_PROBE_AWAY_NO_ERROR 7 // G38.5
#define MOTION_MODE_NONE 8 // G80
```

รูปที่ 3.12 ตัวอย่างคำสั่งโค้ดในคำสั่งต่าง ๆ

```
26  #include "grbl.h"
27
28  #define RC_SERVO_SHORT      25
29  #define RC_SERVO_LONG       39
30  //#define RC_SERVO_INVERT     1
```

รูปที่ 3.13 การกำหนดพินของเซอร์โวมอเตอร์

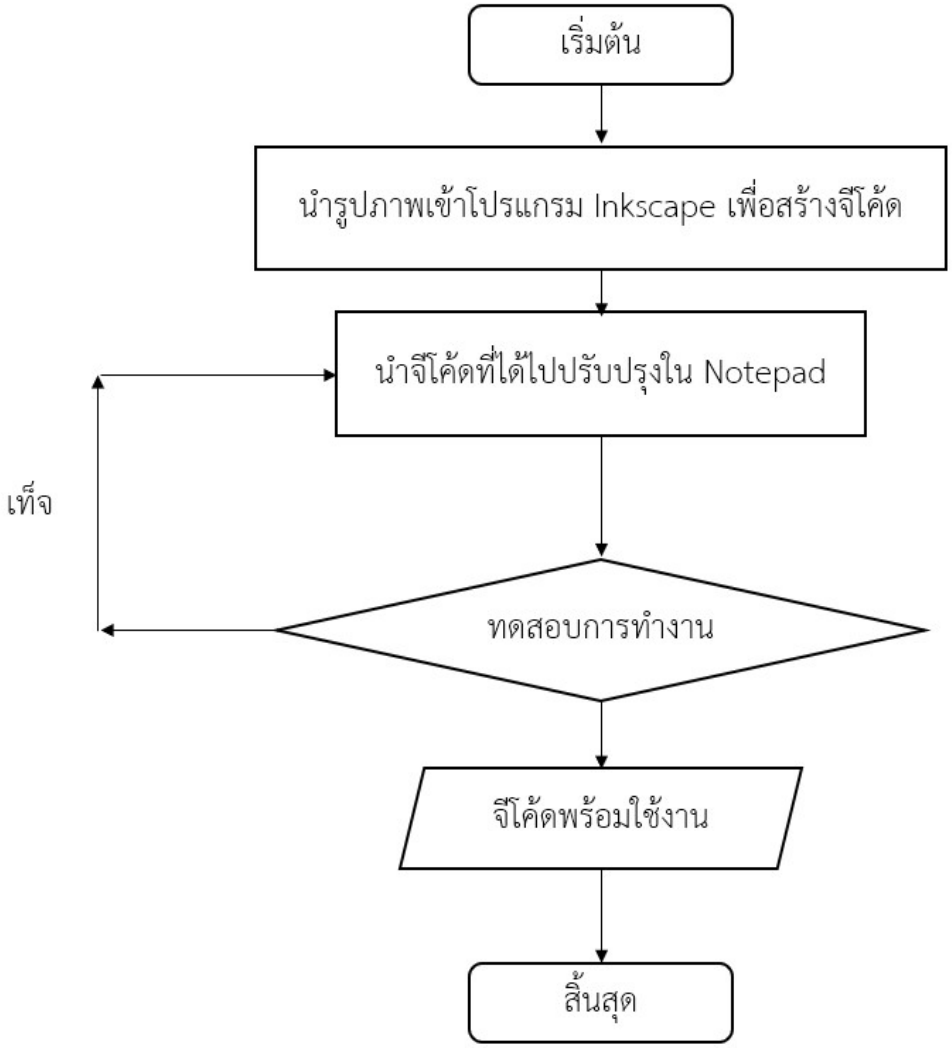
2) เมื่อเชื่อมต่อไมโครคอนโทรเลอร์อาดูโน่เข้ากับคอมพิวเตอร์แล้วการส่งจีโค้ดจะเป็นเรื่องที่ยุ่งยากเพราะตัวคำสั่งมีลักษณะเป็นบรรทัด ในงานที่ซับซ้อนอาจจะยาวถึงหลักพันบรรทัดทำให้จำเป็นต้องใช้โปรแกรมช่วยในการส่งให้กับอาดูโน่ โดยผู้ทำโครงการใช้ซอฟต์แวร์ชื่อ “Universal G code sender”เป็นโปรแกรมฟรีที่เกิดขึ้นจากการร่วมมือสร้างจากชุมชนผู้ชื่นชอบงานซีเอ็นซี ซึ่งตัวโปรแกรมจะทำหน้าที่ในการส่งคำสั่งของโปรแกรมซีเอ็นซีเข้าไปให้อาดูโน่ทีละบรรทัด และสามารถจำลองภาพร่างของจีโค้ดที่เราใส่เข้าไปได้ด้วย



รูปที่ 3.14 ซอฟต์แวร์ Universal G code sender

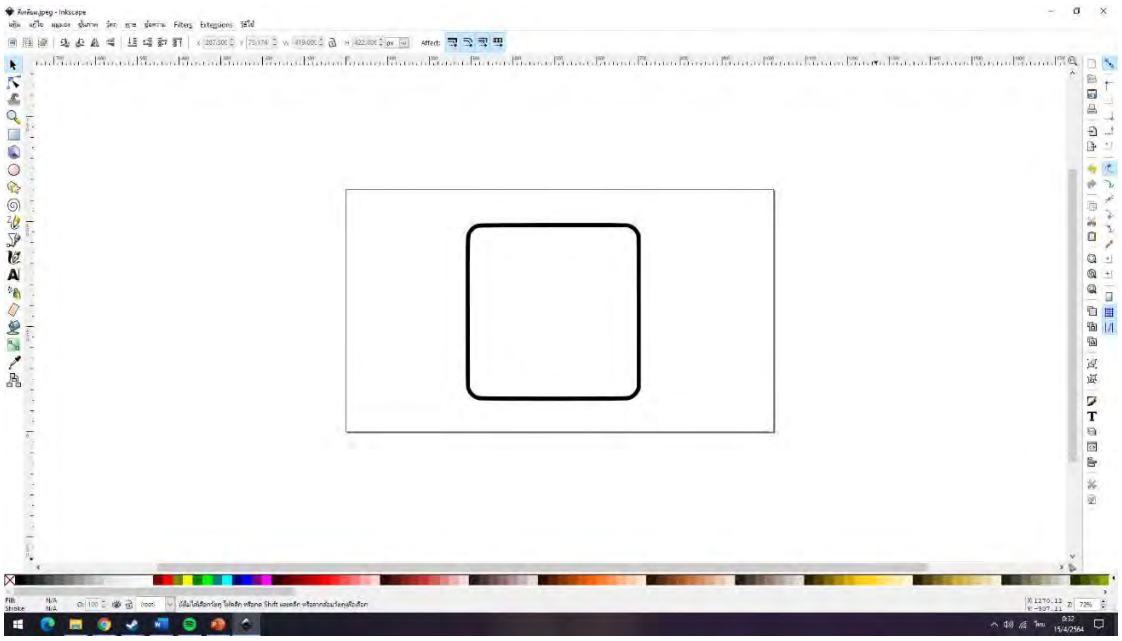
3.5.2 ส่วนในการสร้างและปรับปรุงจีโค้ด

จะแบ่งเป็นสองส่วนที่สำคัญคือ ส่วนการนำรูปภาพมาแปลงเป็นจีโค้ดและส่วนการแก้ไขและปรับปรุงตามแผนภาพ



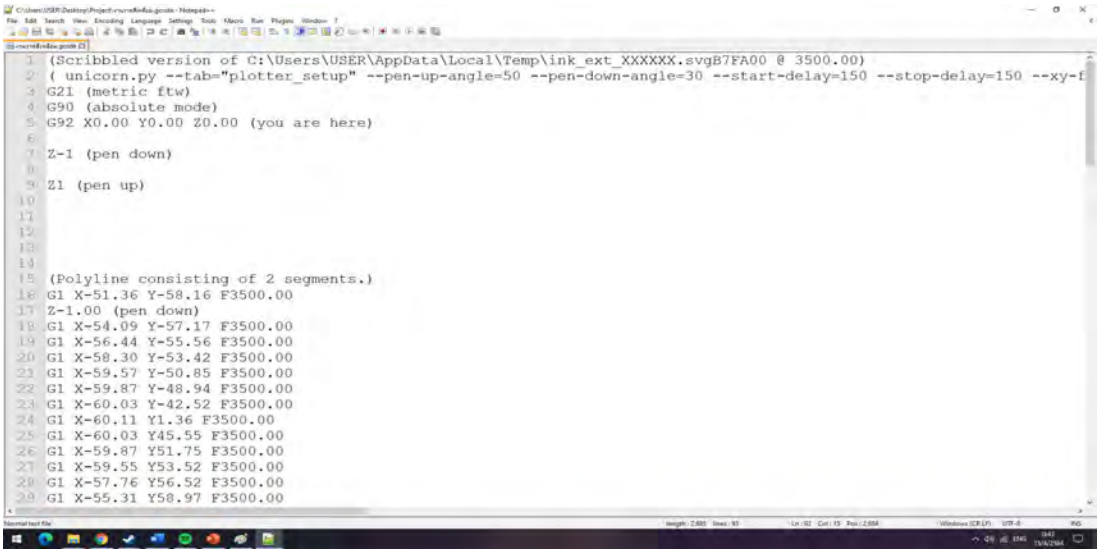
รูปที่ 3.15 แผนผังการทำงานของการสร้าง จีโค้ด

1) ในส่วนแรกจะใช้โปรแกรม Inkscape ในการเปลี่ยนไฟล์รูปภาพที่เราต้องการเป็นจีโค้ดโดยการใช้เครื่องมือที่ชื่อว่า Inkscape Unicorn Master ซึ่งเป็นโอเพ่นซอร์สที่มีประสิทธิภาพสูงและจีโค้ดที่ได้มานั้นยังสามารถแก้ไขได้ด้วยโปรแกรมอื่น ๆ



รูปที่ 3.16 ใช้ซอฟต์แวร์ Inkscape ในการสร้างจีโค้ดรูปสี่เหลี่ยม

2) ส่วนนี้จะเป็นส่วนการแก้ไขและปรับปรุงจีโค้ดที่เราได้จากขั้นตอนแรกโดยจะใช้โปรแกรม Notepad++ เนื่องจากการสร้างจีโค้ดด้วยโปรแกรมนั้นค่อนข้างมีหลายจุดที่ผิดพลาด และต้องแก้ไขให้เข้ากับโปรแกรมที่เราตั้งค่าไว้ จากรูป (3.8) จะเห็นว่าจีโค้ดที่ได้จากการสร้างมานั้นใช้คำสั่งของมอเตอร์แกน Z ด้วยคำสั่ง Z1 และ Z-1 แทนการเคลื่อนที่ของปากกา แต่ในการใช้งานจริงนั้นเราได้ประยุกต์ใช้เซอร์โวมอเตอร์ในการควบคุมปากกาจึงต้องแทน Z-1 ด้วยคำสั่ง M3S50 โดย M3 คือคำสั่งเปิดเซอร์โว และ S50 คือให้หมุนไปจุด 50 องศา นอกจากนั้นก็ใช้แก้ไขคำสั่งรอหรือคำสั่งเปิดปิดเป็นต้น



```
(Scribbled version of C:\Users\USER\AppData\Local\Temp\ink_ext_XXXXXX.svgB7FA00 @ 3500.00)
( unicorn.py --tab="plotter_setup" --pen-up-angle=50 --pen-down-angle=30 --start-delay=150 --stop-delay=150 --xy-f
G21 (metric ftw)
G90 (absolute mode)
G92 X0.00 Y0.00 Z0.00 (you are here)

Z-1 (pen down)

Z1 (pen up)




(Polyline consisting of 2 segments.)
G1 X-51.36 Y-58.16 F3500.00
Z-1.00 (pen down)
G1 X-54.09 Y-57.17 F3500.00
G1 X-56.44 Y-55.56 F3500.00
G1 X-58.30 Y-53.42 F3500.00
G1 X-59.57 Y-50.85 F3500.00
G1 X-59.87 Y-48.94 F3500.00
G1 X-60.03 Y-42.52 F3500.00
G1 X-60.11 Y1.36 F3500.00
G1 X-60.03 Y45.55 F3500.00
G1 X-59.87 Y51.75 F3500.00
G1 X-59.55 Y53.52 F3500.00
G1 X-57.76 Y56.52 F3500.00
G1 X-55.31 Y58.97 F3500.00
```

รูปที่ 3.17 ใช้โปรแกรม Notepad++ ในการดัดแปลงและแก้ไขจีโค้ดที่ได้จากขั้นตอนแรก

3.6 การทดสอบความสามารถในการเขียนเส้นแบบต่าง ๆ

3.6.1 ทดสอบฟังชั่นการทำงานของเครื่องพล็อตเตอร์

3.6.2 ทดสอบการลากเส้นตรง และหาความผิดพลาด

3.6.3 ทดสอบความสามารถในการวาดรูปจริง

บทที่ 4

ผลการทดลอง

โครงการนี้ทำการออกแบบเครื่องพล็อตเตอร์ที่มีความพิเศษคือมีความสามารถในการเคลื่อนที่ โดยมีพื้นฐานมาจากไมโครคอนโทรลเลอร์อาดูโน่ ในการสั่งการสเต็ปปิ้งมอเตอร์และเซอร์โวมอเตอร์ แต่เนื่องจากขีดจำกัดในการสร้างเครื่องพล็อตเตอร์ที่เคลื่อนที่ได้นั้นทำให้แกนมอเตอร์แกนเอกซ์และแกนวายจำเป็นต้องเกี่ยวเนื่องกันกล่าวคือไม่สามารถขยับแค่มอเตอร์แกนใดแกนหนึ่งได้จำเป็นต้องเคลื่อนที่ทั้งสองแกน ทำให้เกิดแรงเสียดท้านขึ้นทำให้การเคลื่อนที่ของเครื่องพล็อตเตอร์คลาดเคลื่อนถึงแม้จะใช้งานล้อแบบพิเศษแล้ว การทดสอบการทำงานจะเริ่มจากการทดสอบฟังก์ชั่นต่าง ๆ หลังจากนั้นจะวัดความคลาดเคลื่อนที่เกิดจากแรงเสียดทาน และตัวแปรอื่น ๆ ที่มีผลเช่นความเรียบของพื้นผิว

4.1 การทดสอบฟังชั่นการทำงาน

การทดสอบฟังก์ชั่นการทำงานของเครื่องพล็อตเตอร์โดยการป้อนคำสั่งต่าง ๆ ไปยังโปรแกรมยูนิเวอร์ซอล จีโค้ด เซนเดอร์ (Universal G code sender) โดยทดสอบส่งคำสั่ง

ตารางที่ 4.1 แสดงคำสั่ง ความหมาย และจุดประสงค์ของการทดสอบ

จีโค้ด	ความหมาย	จุดประสงค์
G1 X10 Y0 F50	เคลื่อนที่ไปที่จุด (10,0) ด้วยอัตราการป้อน 50	ทดสอบการหมุนของมอเตอร์แกนเอกซ์และทำการสอบเทียบ
G1 X0 Y10 F50	เคลื่อนที่ไปที่จุด (0,10) ด้วยอัตราการป้อน 50	ทดสอบการหมุนของมอเตอร์แกนวายและทำการสอบเทียบ
G4 P5	ทำการหยุดอยู่กับที่และรอ 5 วินาที	ทดสอบคำสั่งหยุดเพื่อรอการทำงานของปากกา
M3 S30	สั่งเซอร์โวมอเตอร์หมุนปากกาไปที่จุด 30 องศาเพื่อเริ่มเขียนงาน	ทดสอบการทำงานของเซอร์โวมอเตอร์ในการกดปากกา
M3 S50	สั่งเซอร์โวมอเตอร์หมุนปากกาไปที่จุด 50 องศาเพื่อปิดปากกา	ทดสอบการทำงานของเซอร์โวมอเตอร์ในการยกปากกา
M02	คำสั่งจบการทำงาน	ทดสอบการจบการทำงาน

สำหรับคำสั่ง F คือการกำหนดอัตราการป้อน เป็นคำสั่งที่ใช้เพิ่มหรือลดความเร็วในการในการทำงาน ในการทดสอบนี้ใช้อยู่ที่ 50 มีหน่วยเป็นมิลลิเมตรต่อนาที ซึ่งค่าที่ใช้ทดสอบนี้ค่อนข้างต่ำทำให้เครื่องทำงานช้าเพราะ ซึ่งจะทำให้เกิดความคลาดเคลื่อนจากตัวแปรต่าง ๆ น้อยลง

และในคำสั่งการเปิดปิดปากกาผู้ทำโครงการได้ติดตั้งเซอร์โวมอเตอร์ให้ทำมุม 90 องศากับพื้นและให้มอเตอร์หมุนตามเข็มนาฬิกา ทำให้การหมุนไปที่จุด 30 องศา (แกนหมุนขยับไป 60 องศา) จะทำให้หัวปากกาสัมผัสกับพื้นพอดี และให้จุด 50 องศาเป็นจุดยกปากกา

จากการทดสอบการทำงานพบว่าเครื่องพล็อตเตอร์พบว่าสามารถทำงานตามคำสั่งได้เป็นปกติ ไม่เกิดปัญหาติดขัดระหว่างดำเนินงาน สามารถปรับความเร็วช้าได้เป็นปกติ ทางด้านเซอร์โวมอเตอร์ช่วงแรกพบว่าไม่สามารถใช้เซอร์โวแบบ 360 องศา เพราะจะไม่สามารถหยุดการหมุนเพื่อควบคุมการเปิดปิดปากกาได้จึงจำเป็นต้องใช้เซอร์โวแบบ 180 องศา

4.2 การปรับค่าแต่งสเต็ปต่อมิลลิเมตรของแต่ละแกน

เนื่องจากล้อมีขนาดที่ต่างกันจึงจำเป็นต้องปรับให้ค่าสเต็ปต่อมิลลิเมตรของทั้งสองล้อมีค่าเท่ากันเพื่อที่จะทำให้การลากเส้นของทั้งแกนเอกซ์และแกนวายมีขนาดใกล้เคียงกันมากที่สุด การปรับค่าทั้งสองสามารถปรับในส่วนของตัวโปรแกรมได้แต่จะมีความยุ่งยาก จึงใช้ความสามารถของโปรแกรม ยูนิเวอร์ซอล จีโค้ด เซนเดอร์ โดยการตั้งค่าพารามิเตอร์ของโปรแกรมดังรูปที่ 4.1 จากนั้นนำมาทำการวัดเพื่อเปรียบเทียบกับค่าจริง

Setting	Value	Description
$0	10	Step pulse time
$1	25	Step idle delay
$2	0	Step pulse invert
$3	0	Step direction invert
$4	0	Invert step enable pin
$5	0	Invert limit pins
$6	0	Invert probe pin
$10	1	Status report options
$11	0.010	Junction deviation
$12	0.002	Arc tolerance
$13	0	Report in inches
$20	0	Soft limits enable
$21	0	Hard limits enable
$22	0	Homing cycle enable
$23	0	Homing direction invert
$24	25.000	Homing locate feed rate
$25	500.000	Homing search seek rate
$26	250	Homing switch debounce delay
$27	1.000	Homing switch pull-off distance
$30	255	Maximum spindle speed
$31	0	Minimum spindle speed
$32	0	Laser-mode enable
$100	11.380	X-axis travel resolution
$101	50.000	Y-axis travel resolution
$102	250.000	Z-axis travel resolution
$110	500.000	X-axis maximum rate
$111	500.000	Y-axis maximum rate
$112	500.000	Z-axis maximum rate
$120	10.000	X-axis acceleration
$121	10.000	Y-axis acceleration
$122	10.000	Z-axis acceleration
$130	200.000	X-axis maximum travel
$131	200.000	Y-axis maximum travel
$132	200.000	Z-axis maximum travel

รูปที่ 4.1 หน้าต่างการปรับค่าต่าง ๆ ของส่วนควบคุมมอเตอร์

ตารางที่ 4.2 แสดงข้อมูลที่สำคัญในการปรับเครื่องพล็อตเตอร์

ค่า	จุดประสงค์ในการปรับค่า
$100	ปรับสเต็ปต่อมิลลิเมตรของแกนเอกซ์
$101	ปรับสเต็ปต่อมมิลลิเมตรของแกนวาย
$130	ปรับระยะทางสูงสุดที่เครื่องสามารถเคลื่อนที่ไปได้ของแกนเอกซ์
$131	ปรับระยะทางสูงสุดที่เครื่องสามารถเคลื่อนที่ไปได้ของแกนวาย

เพื่อหาค่าความละเอียด (resolution) ของพล็อตเตอร์ ซึ่งหมายถึงระยะทางที่น้อยที่สุดที่พล็อตเตอร์สามารถเคลื่อนที่ได้ ซึ่งคือระยะทางที่เกิดขึ้นเมื่อมอเตอร์หมุนไป 1 สเต็ป ใช้วิธีการสอบเทียบจำนวนสเต็ปต่อมิลลิเมตรของเครื่องพล็อตเตอร์โดยการให้โปรแกรมสั่งการให้ขีดเส้น 10 มิลลิเมตร ซึ่งจากการทดลองปรับค่าพบว่าเมื่อปรับใช้ค่าสเต็ปต่อมิลลิเมตรของแกนเอกซ์และวายเป็น 11.42 และ 17.45 ตามลำดับเครื่องก็จะสามารถลากเส้น 10 เซนติเมตรได้พอดี ค่าทั้งสองจะนำไปคำนวณความละเอียดของเครื่องพล็อตเตอร์ ดังหัวข้อ 5.1 ต่อไป

4.3 การทดสอบความคลาดเคลื่อนเชิงสถิติจากการใช้งาน

การทดสอบมีวัตถุประสงค์เพื่อเปรียบเทียบความยาวของเส้นที่พล็อตเตอร์ลากบนแผ่นกระดาษ (S) เปรียบเทียบกับค่าความยาวที่กำหนดให้โปรแกรม (S_0) โดยการเพิ่มค่าความยาว S_0 ทีละ 10 เซนติเมตรและเริ่มต้นที่ 10 เซนติเมตรไปจนถึง 80 เซนติเมตร ผลการทดสอบสำหรับแกนเอกซ์และวายเป็นดังตาราง 4.3 และ 4.4 เมื่อคิดค่าเฉลี่ยของความยาว S มาเขียนกราฟแสดงความสัมพันธ์กับ S_0 จะได้ดังรูปที่ 4.2 และ 4.5 ตามลำดับ

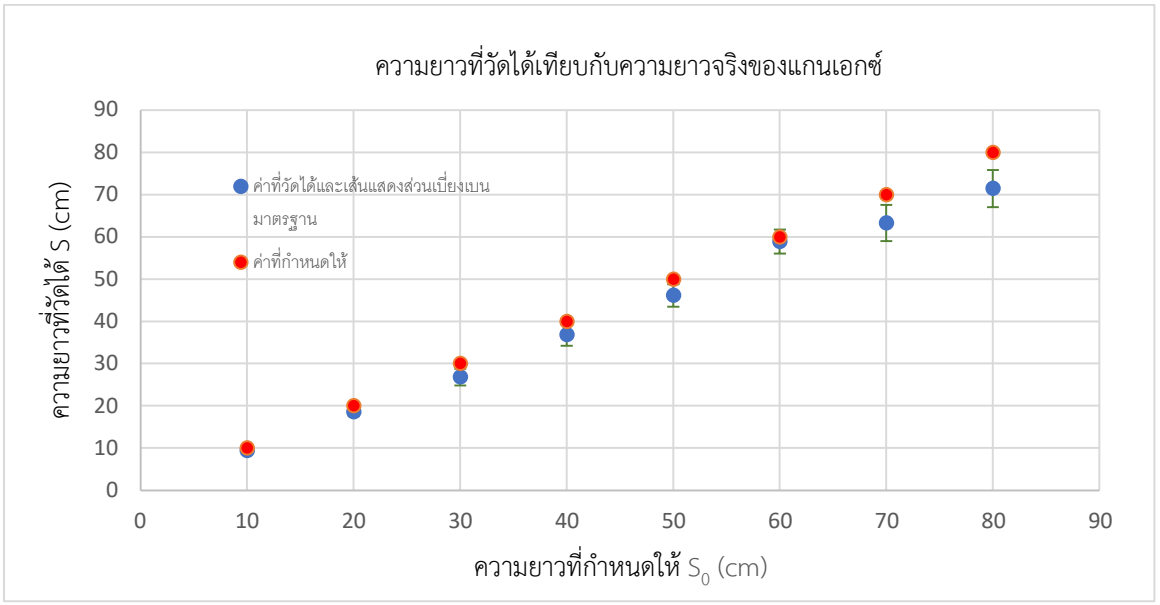
เพื่อวิเคราะห์ความแม่นยำ (accuracy) ในเชิงสถิติจึงได้นำผลต่างระหว่างระยะทางจริงและค่าที่กำหนดให้ ($\Delta S = |S-S_0|$) มาเขียนกราฟเพื่อหาความสัมพันธ์กับ S_0 ดังรูป 4.3 และ 4.6 สำหรับแกนเอกซ์และวายตามลำดับ

เพื่อวิเคราะห์ความเที่ยงตรง (precision) ในเชิงสถิติ จึงได้นำค่าเบี่ยงเบนมาตรฐานมาเขียนกราฟแสดงความสัมพันธ์กับ S_0 สำหรับแกนเอกซ์และวายดังรูปที่ 4.4 และ 4.7 ตามลำดับ

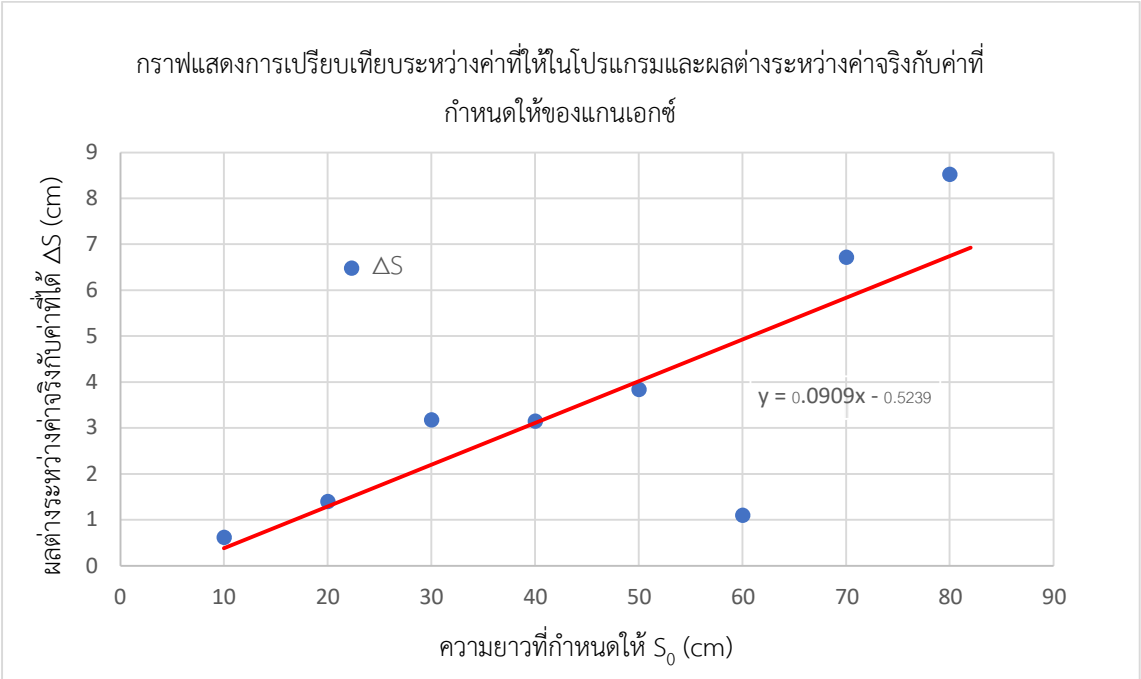
1.แกนเอกซ์

ตารางที่ 4.3 แสดงการทดสอบขีดเส้นในแนวแกนเอกซ์ความคลาดเคลื่อนสัมพัทธ์และความเบี่ยงเบนมาตรฐาน

ความยาวที่กำหนดให้ S_0 (cm.)	ความยาวที่วัดได้จริง S (cm.)						$\Delta S = \lvert S-S_0 \rvert$	ความเบี่ยงเบนมาตรฐาน (cm.)
	1	2	3	4	5	ค่าเฉลี่ย		
10	9.9	10	8.3	9.7	9	9.38	0.62	0.72
20	19.5	16	19.1	19.5	18.9	18.6	1.4	1.48
30	23.5	28.5	24.2	29.5	28.4	26.82	3.18	2.76
40	37.4.	35.5	39.9	38.7	33.3	36.85	3.15	3.01
50	49	48.2	46.2	43.1	44.3	46.16	3.84	2.50
60	59.8	61	55.2	59.5	59	58.9	1.1	2.20
70	60.3	65.5	57.8	67.2	65.6	63.28	6.72	4.02
80	70.5	73.2	70	78.2	65.5	71.48	8.52	4.66

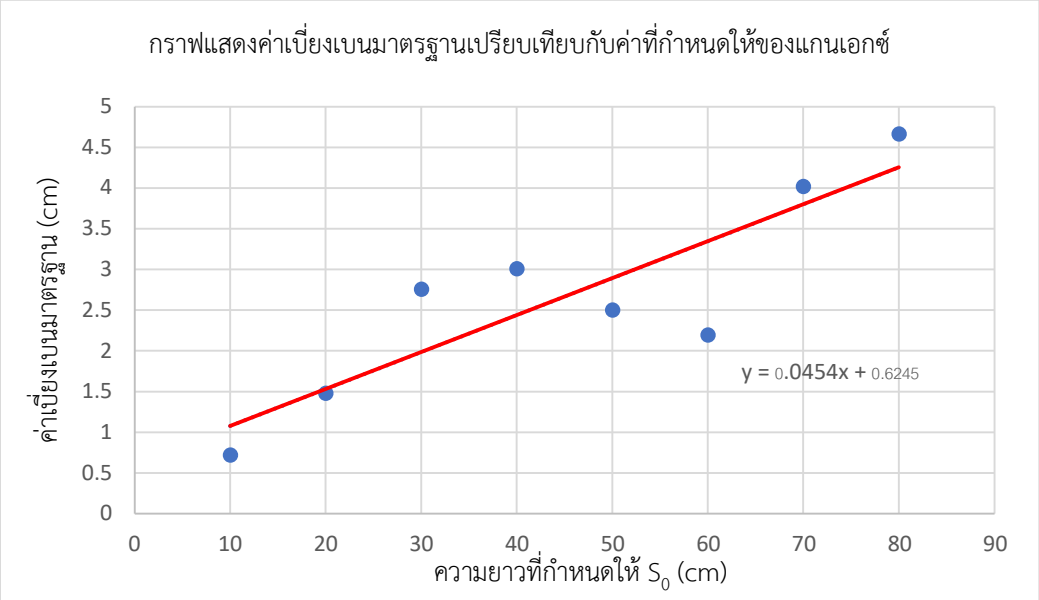


รูปที่ 4.2 กราฟแสดงความยาวเส้นที่วาดได้(จุดสีฟ้า) ความยาวเส้นที่กำหนดในโปรแกรม(จุดสีแดง) และความเบี่ยงเบนมาตรฐาน(เส้นสีเขียว) ของแกนเอกซ์



รูปที่ 4.3 กราฟแสดงการเปรียบเทียบระหว่างค่าที่ให้ในโปรแกรมและผลต่างระหว่างค่าจริงกับค่าที่กำหนดให้ของแกนเอกซ์

มีสมการของเส้นแนวโน้มคือ $y = 0.0909x - 0.524$



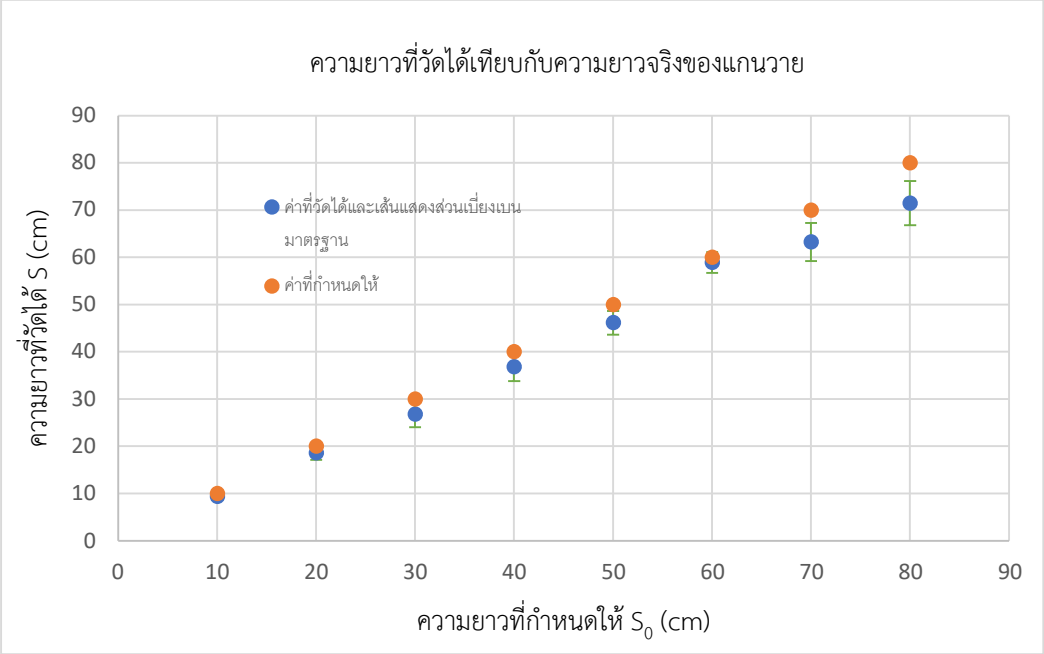
รูปที่ 4.4 กราฟแสดงแนวโน้มของความเบี่ยงเบนมาตรฐานของการเขียนเส้นในแนวแกนเอกซ์และเส้นแนวโน้ม

มีสมการของเส้นแนวโน้มคือ $y = 0.0454x + 0.624$

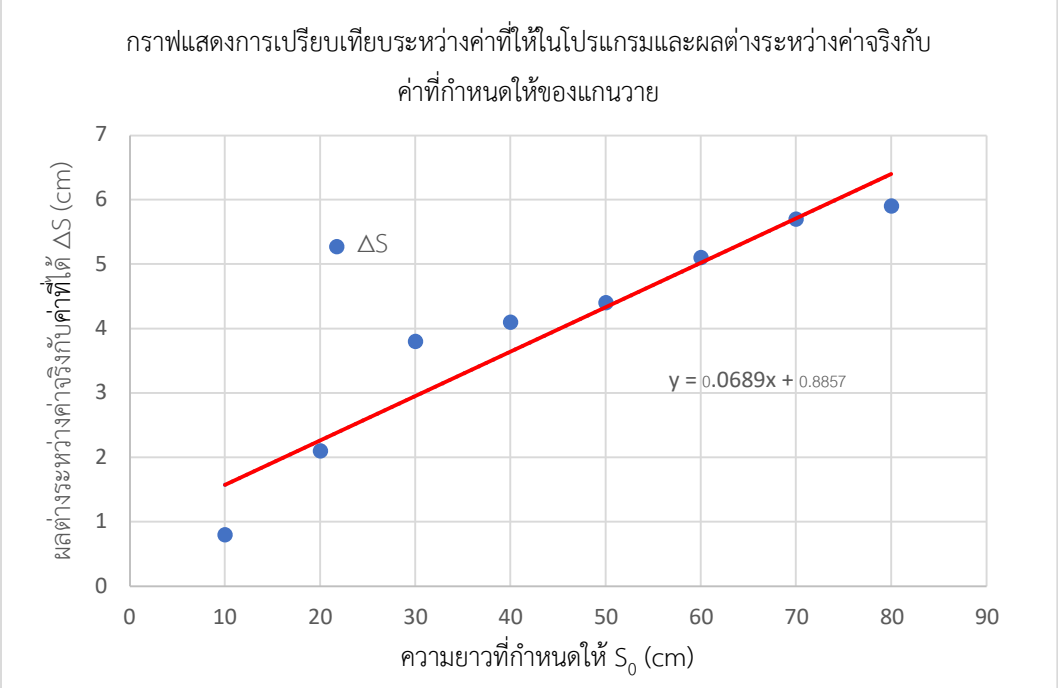
2.แกนวาย

ตารางที่ 4.4 แสดงการทดสอบขีดเส้นในแนวแกนวายความคลาดเคลื่อนสัมพัทธ์และความเบี่ยงเบนมาตรฐาน

ความยาวที่กำหนดให้ (cm.)	ความยาวที่วัดได้จริง(cm.)						$\Delta S = \lvert S-S_0 \rvert$	ความเบี่ยงเบนมาตรฐาน (cm)
	1	2	3	4	5	ค่าเฉลี่ย		
10	9.5	8.9	8.6	9.3	9.5	9.2	0.8	0.39
20	17.5	16.4	17.8	18.6	19	17.9	2.1	1.01
30	25.2	23.5	27.8	26	28.3	26.2	3.8	1.95
40	36.3	32.5	35.4	33.5	39.2	35.9	4.1	2.61
50	47.1	46.5	48.1	41.3	44.9	45.6	4.4	2.66
60	53.2	54.3	58.6	56.5	51.3	54.9	5.1	2.84
70	67.2	66.5	60.2	59.1	68.3	64.3	5.7	4.27
80	74.5	68.4	78.3	71.1	78.3	74.1	5.9	4.39

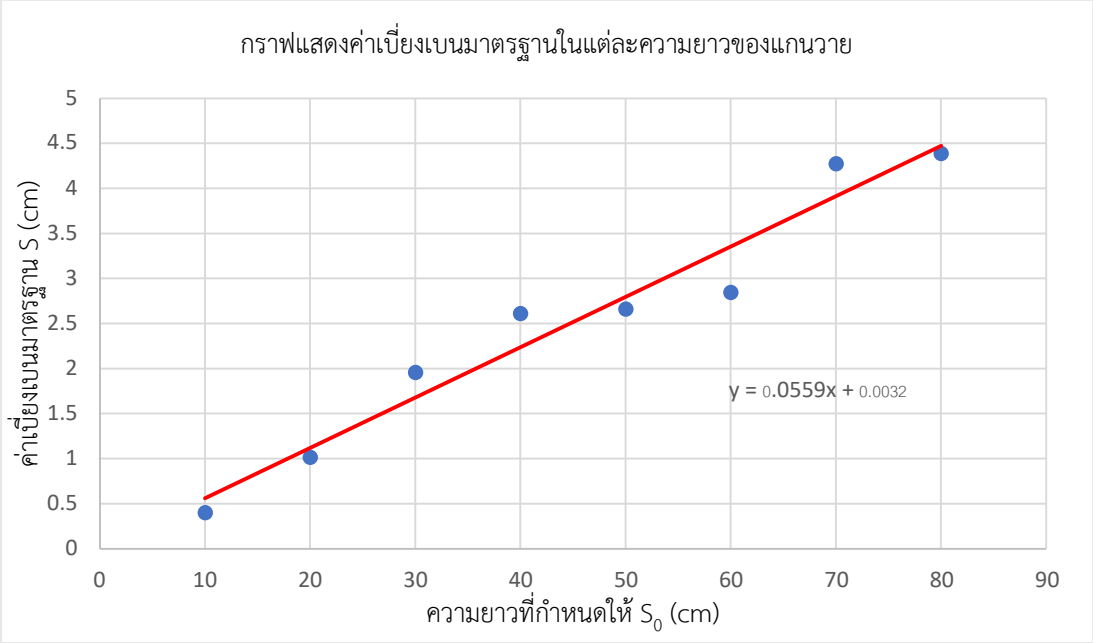


รูปที่ 4.5 กราฟแสดงความยาวที่วาดได้(จุดสีฟ้า)ความยาวที่กำหนดในโปรแกรม(จุดสีแดง)และความเบี่ยงเบนมาตรฐาน(เส้นสีเขียว) ของแกนวาย



รูปที่ 4.6 กราฟแสดงการเปรียบเทียบระหว่างค่าที่ให้ในโปรแกรมและผลต่างระหว่างค่าจริงกับค่าที่กำหนดให้ของแกนวาย

มีสมการของเส้นแนวโน้มคือ y = 0.0689x + 0.886

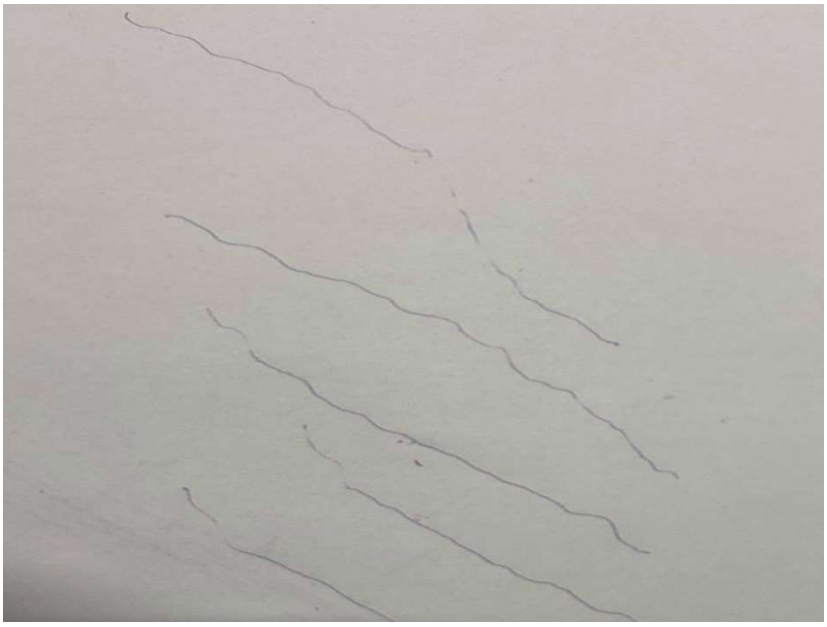


รูปที่ 4.7 กราฟแสดงแนวโน้มของความเบี่ยงเบนมาตรฐานของการเขียนเส้นในแนวแกนวาย เส้นแนวโน้ม

มีสมการของเส้นแนวโน้มคือ y = 0.0559x + 0.003

4.4 การทดสอบการเขียนเส้นที่ไม่ใช่แนวทแยงที่เกิดจากขนาดของล้อที่ต่างกัน

จากการทดสอบการเขียนรูปในแนวทแยงพบว่าเครื่องมีความสามารถในการเคลื่อนที่ในแนวทแยง แต่ว่าหากต้องการลากเส้น พบว่าเส้นมีลักษณะเป็นลูกคลื่นไม่มีความเป็นเส้นตรง เหตุที่เป็นแบบนี้เนื่องจากล้อที่นำมาใช้มีขนาดที่ต่างกันและการปรับแต่งค่าต่าง ๆ ไม่สามารถทำให้ล้อมีความสัมพันธ์กันแบบ 100 เปอร์เซ็นต์ได้

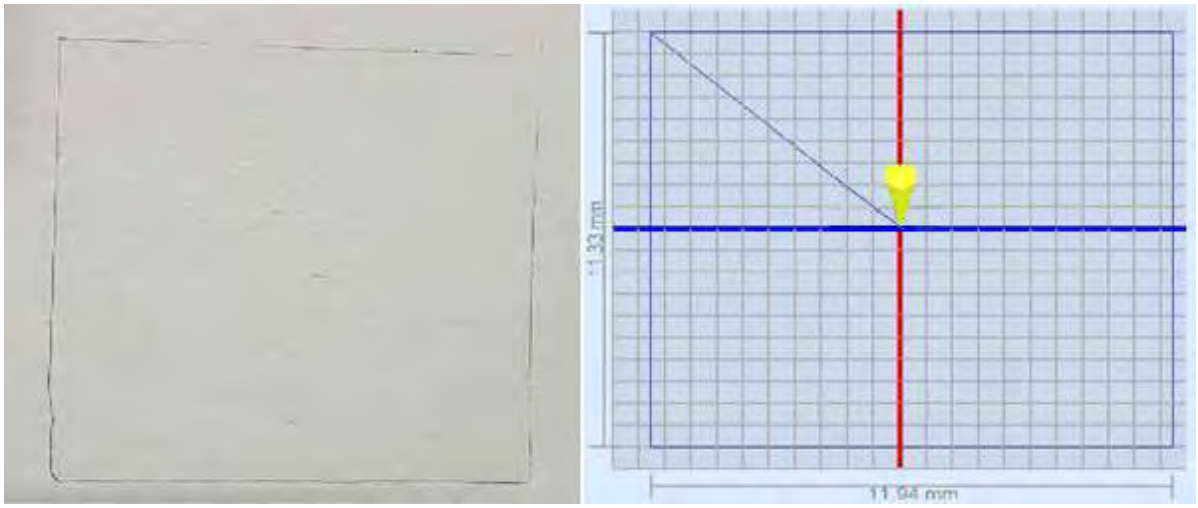


รูปที่ 4.8 ตัวอย่างเส้นในการลากแนวทแยง

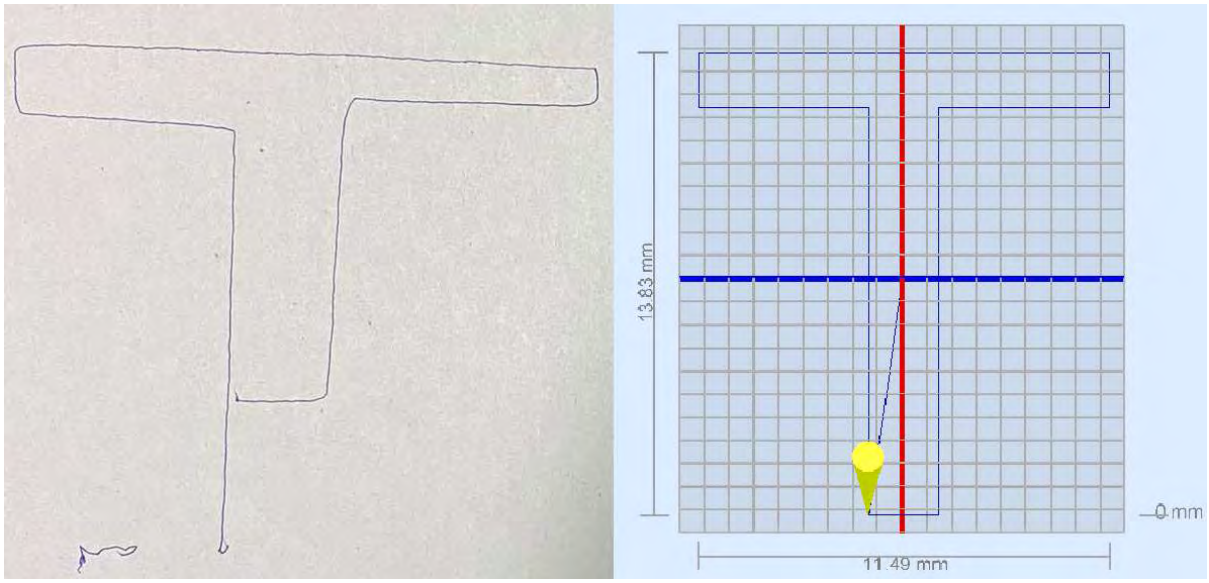
4.5 ขอบเขตความสามารถและการทดสอบการทำงานจริง

การวัดผลจากการทดลองเขียนงานจริงหลาย ๆ ลักษณะเช่นรูปที่มีองค์ประกอบแค่เส้นในแนวแกนเอกซ์และแกนวาย รูปที่มีเส้นในแนวทแยง และรูปที่มีเส้นในแนวโค้ง

4.5.1 การทดสอบด้วยการวาดรูปที่มีแต่เส้นตรงในแนวแกนเอกซ์และแกนวาย

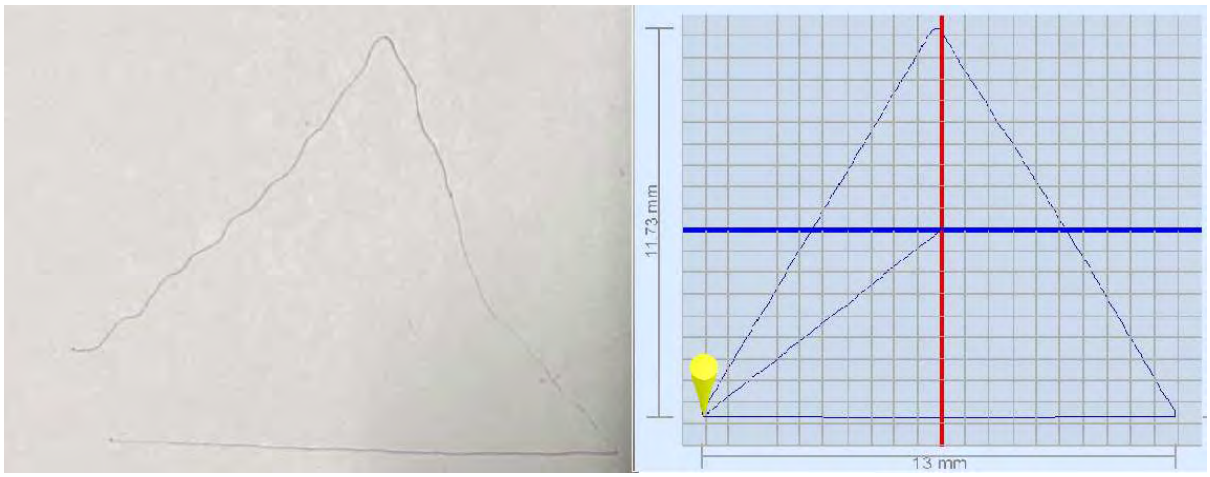


รูปที่ 4.9 (ก) ทดสอบด้วยการวาดรูปสี่เหลี่ยม (ข)รูปในโปรแกรม

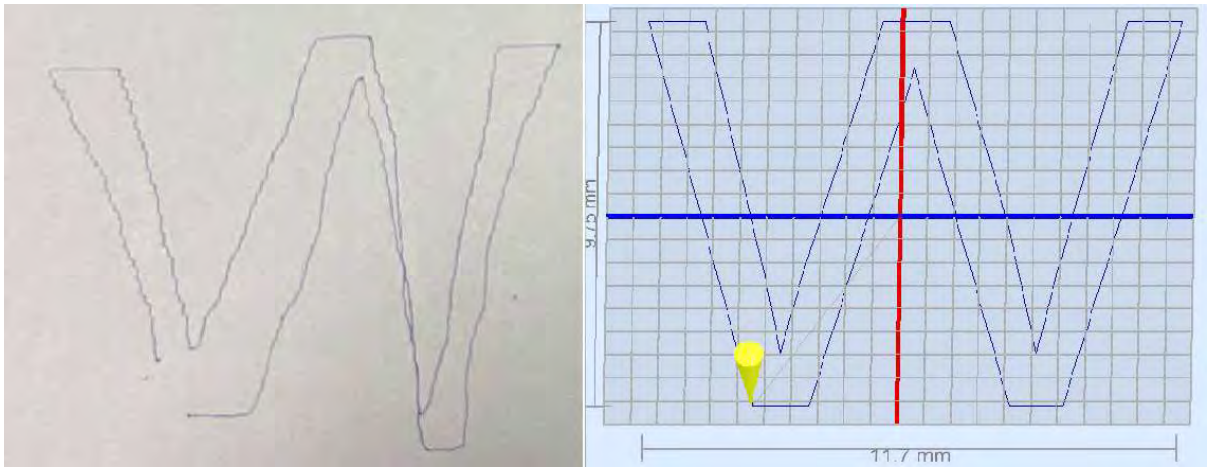


รูปที่ 4.10 (ก) ทดสอบด้วยการวาดรูปตัว T (ข)รูปในโปรแกรม

4.5.2 การทดสอบด้วยการวาดรูปที่มีเส้นแนวทแยง



รูปที่ 4.11 (ก) ทดสอบด้วยการวาดรูปสามเหลี่ยม (ข)รูปในโปรแกรม



รูปที่ 4.12 (ก) ทดสอบด้วยการวาดรูปตัวอักษร W (ข)รูปในโปรแกรม

4.5.2 การทดสอบด้วยการวาดรูปที่มีเส้นแนวโค้ง

รูปที่ 4.13 (ก) ทดสอบด้วยการวาดรูปวงรี (ข)รูปในโปรแกรม

บทที่ 5

สรุปและอภิปรายผลการทดลอง

5.1 วิเคราะห์ผลการทดลอง

การวิเคราะห์ความคลาดเคลื่อน (error) ของแกนทั้งสอง โดยแบ่งเป็น 2 แบบ ได้แก่ความคลาดเคลื่อนเชิงระบบ (systematic error) และความคลาดเคลื่อนเชิงสถิติ (statistical error) ซึ่งแต่ละแบบจะวิเคราะห์ทั้งความแม่นยำ (accuracy) และความเที่ยงตรง (precision) เพื่อที่จะหาค่าความไม่แน่นอน (uncertainty) และค่าความน่าเชื่อถือ (reliability) ของแต่ละแกน

5.1.1 หาค่าความไม่แน่นอนของแกนเอกซ์

1. หาค่าความคลาดเคลื่อนเชิงระบบที่ประเมินจากความละเอียดจากข้อมูลในหัวข้อ 4.2 สามารถคำนวณได้ดังนี้

   - เมื่อสั่งการจากโปรแกรมให้พล็อตเตอร์เคลื่อนที่ไป 10 มิลลิเมตรแต่ระยะทางจริงคือ 10.0 เซนติเมตร ดังนั้น 1 มิลลิเมตรในโปรแกรมจึงเทียบได้กับ 1.0 เซนติเมตรของระยะทางจริง
   - จากการทดลองปรับค่าในโปรแกรมได้เป็น 11.42 สเต็ปต่อมิลลิเมตรซึ่งในที่นี้คิดเป็นระยะทางจริง 1.0 เซนติเมตร

     ดังนั้นค่าความละเอียดของล้อในแกนเอกซ์จึงเป็น 1.0/11.42 เซนติเมตร เท่ากับ 0.87 มิลลิเมตร

2. หาค่าความคลาดเคลื่อนเชิงระบบโดยวิเคราะห์จากความเที่ยงตรง ในส่วนนี้ไม่สามารถวิเคราะห์ออกมาเป็นตัวเลขได้เนื่องจากความเที่ยงตรงคิดจากการวัดซ้ำหลาย ๆ ครั้ง

3. หาค่าความแม่นยำเชิงสถิติโดยวิเคราะห์จากกราฟแสดงการเปรียบเทียบระหว่างค่าที่ให้ในโปรแกรมและผลต่างระหว่างค่าจริงกับค่าที่กำหนดให้ของแกนเอกซ์ในรูปที่ 4.3

   จากสมการเส้นแนวโน้ม y = 0.0909x – 0.5239

   สามารถหาค่าความคลาดเคลื่อนสัมพัทธ์ (relative error) $\frac{\Delta S}{S_0}$ ได้จากความชัน (m) ได้คือ 0.0909

   จะได้เป็น 0.0909 x 100 = 9.09 เปอร์เซ็นต์ และมีค่าความผิดพลาดที่น้อยที่สุดจากการทดสอบซ้ำ (absolute error) คือ 5.24 มิลลิเมตร

4. หาค่าความคลาดเคลื่อนเชิงสถิติโดยวิเคราะห์จากความเที่ยงตรง ซึ่งประเมินได้จากกราฟแสดงค่าเบี่ยงเบนมาตรฐานเปรียบเทียบกับค่าที่กำหนดให้ของแกนเอกซ์ในรูปที่ 4.4

   จากกราฟได้สมการเส้นแนวโน้มคือ y = 0.0454x + 0.624

ซึ่งความชันที่ได้เป็นความแม่นยำสัมพัทธ์ในเชิงสถิติ (relative statistic precision) และเป็นตัวอธิบายว่ามีการกระจายของข้อมูลมากน้อยเท่าไร

จะได้ค่าความแม่นยำสัมพัทธ์เป็น 0.0454 x 100 = 4.54 เปอร์เซ็นต์

นำข้อมูลทั้งหมดมาใส่ในเมตริกซ์ของความไม่แน่นอน

ตารางที่ 5.1 ตารางแสดงเมตริกซ์ของความไม่แน่นอนของแกนเอกซ์

	ความเที่ยงตรง	ความแม่นยำ
ความคลาดเคลื่อนเชิงระบบ	0.87 mm.	-
ความคลาดเคลื่อนเชิงสถิติ	5.24 mm.	4.54 %
	9.09 %	

เนื่องจากในการหาค่าความไม่แน่นอนจะเลือกจากค่าที่มีความผิดพลาดมากที่สุดดังนั้นจากตารางที่ 5.1 เราสามารถวิเคราะห์ได้ว่าเครื่องพล็อตเตอร์มีความไม่แน่นอนในแกนเอกซ์ 9.09 เปอร์เซ็นต์ และมีความน่าเชื่อถือเป็น 90.91 เปอร์เซ็นต์

5.1.2 หาค่าความไม่แน่นอนของแกนวาย

1. หาค่าความคลาดเคลื่อนเชิงระบบที่ประเมินจากความละเอียดจากข้อมูลในหัวข้อ 4.2 สามารถคำนวณได้ดังนี้

- เมื่อสั่งการจากโปรแกรมให้พล็อตเตอร์เคลื่อนที่ไป 10 มิลลิเมตร แต่ระยะทางจริงคือ 10.0 เซนติเมตร ดังนั้น 1 มิลลิเมตร ในโปรแกรมจึงเทียบได้กับ 1.0 เซนติเมตรของระยะทางจริง
- จากการทดลองปรับค่าในโปรแกรมได้เป็น 17.45 สเต็ปต่อมิลลิเมตรซึ่งในที่นี้คิดเป็นระยะทางจริง 1.0 เซนติเมตร

  ดังนั้นค่าความละเอียดของล้อในแกนเอกซ์จึงเป็น 1.0/17.45 เซนติเมตร เท่ากับ 0.57 มิลลิเมตร

2. หาค่าความคลาดเคลื่อนเชิงระบบโดยวิเคราะห์จากความเที่ยงตรง เช่นเดียวกับแกนเอกซ์ ค่านี้ไม่สามารถวิเคราะห์ออกมาเป็นตัวเลขได้เนื่องจากเป็นความผิดพลาดที่เกิดขึ้นจากการทำซ้ำหลาย ๆ ครั้ง

3. หาค่าความแม่นยำเชิงสถิติโดยวิเคราะห์จากกราฟแสดงการเปรียบเทียบระหว่างค่าที่ให้ในโปรแกรมและผลต่างระหว่างค่าจริงกับค่าที่กำหนดให้ของแกนเอกซ์ในรูปที่ 4.6

จากกราฟได้สมการเส้นแนวโน้มคือ y = 0.0689x + 0.886

สามารถหาค่าความคลาดเคลื่อนสัมพัทธ์ (relative error) $\frac{\Delta S}{S_0}$ ได้จากความชัน (m) ได้คือ 0.0689

จะได้เป็น 0.0689 x 100 = 96.89 เปอร์เซ็นต์ และมีค่าความผิดพลาดที่น้อยที่สุดจากการทดสอบซ้ำ (absolute error) คือ 5.24 มิลลิเมตร

4. หาค่าความคลาดเคลื่อนเชิงสถิติโดยวิเคราะห์จากความเที่ยงตรง เช่นเดียวกับแกนเอกซ์ซึ่งหาได้จากกราฟแสดงค่าเบี่ยงเบนมาตรฐานเปรียบเทียบกับค่าที่กำหนดให้ของแกนวายในรูปที่ 4.7

จากกราฟได้สมการเส้นแนวโน้มคือ y = 0.0559x + 0.0032

จะได้ค่าความแม่นยำสัมพัทธ์เป็น 0.0559 x 100 = 5.59 เปอร์เซ็นต์

นำข้อมูลทั้งหมดมาใส่ในเมตริกซ์ของความไม่แน่นอน

ตารางที่ 5.2 ตารางแสดงเมตริกซ์ของความไม่แน่นอนของแกนวาย

	ความเที่ยงตรง	ความแม่นยำ
ความคลาดเคลื่อนเชิงระบบ	0.57 mm.	-
ความคลาดเคลื่อนเชิงสถิติ	8.86 mm.	5.59 %
	6.89 %	

จากตารางที่ 5.2 เราสามารถวิเคราะห์ได้ว่าเครื่องพล็อตเตอร์มีความไม่แน่นอนในแกนวาย 6.89 เปอร์เซ็นต์ และมีความน่าเชื่อถือเป็น 93.11 เปอร์เซ็นต์

5.2 สรุปและอภิปรายผล

การสร้างเครื่อง เอกซ์วาย พล็อตเตอร์ ที่มีความสามารถในการเคลื่อนที่ โดยใช้ไมโครคอนโทรเลอร์อาดูโน่ มีการออกแบบการจัดวางล้อและเพลาตามหลักการเคลื่อนที่แบบ โฮโลโนมิคไดรฟ์ เพื่อให้สามารถเคลื่อนที่ได้ทุกทิศทาง ล้อและเพลาถูกขับเคลื่อนด้วยสเต็ปเปอร์มอเตอร์แบบไบโพลาร์ ในส่วนของอุปกรณ์จับปากกาได้สร้างจากเซอร์โวมอเตอร์ที่รับสัญญาณแบบ พัลส์วิดมอดูเลชั่น ในการควบคุมการเปิดปิดปากกา เครื่องพล็อตเตอร์นี้สามารถทำงานโดยการรับข้อมูลรูปภาพจากคอมพิวเตอร์ในรูปแบบของจีโค้ด โดยการแปลงด้วยโปรแกรมต่าง ๆ สามารถนำไปควบคุมเครื่องพล็อตเตอณ์ได้

จากการทดสอบป้อนคำสั่งพื้นฐานของจีโค้ดให้แก่เครื่องพล็อตเตอร์ พบว่าสามารถทำงานได้ตามคำสั่ง โดยมีค่าความละเอียดซึ่งแสดงถึงความคลาดเคลื่อนเชิงระบบของเครื่องพล็อตเตอร์ คือแกนเอกซ์ที่มีล้อขนาดเส้นผ่าศูนย์กลาง 5.8 เซนติเมตร มีค่าความละเอียดที่ 0.87 มิลลิเมตรและแกนวายที่มีล้อขนาดเส้นผ่าศูนย์กลาง 3.8 เซนติเมตรมีค่าความละเอียดที่ 0.57 มิลลิเมตร

การทดลองหาค่าความคลาดเคลื่อนจากการเปรียบเทียบค่าความยาวที่ให้ในโปรแกรมกับค่าที่เครื่องวาดได้จริง พบว่าความยาวของการลากเส้นทั้งในแกนเอกซ์และแกนวาย มีแนวโน้มที่จะผิดพลาดมากขึ้นเมื่อลากเส้นยาวขึ้นเช่นเดียวกับค่าความผิดพลาดสัมพัทธ์ จากการวิเคราะห์การถดถอยเชิงเส้น จากความชันที่ได้พบว่าความคลาดเคลื่อนมีค่า 9.09 เปอร์เซ็นต์ และมีจุดตัดแกนเท่ากับ 5.24 มิลลิเมตรในแกนเอกซ์ ส่วนในแกนวายเป็น 6.89 เปอร์เซ็นต์และจุดตัดแกนเท่ากับ 8.86 มิลลิเมตร เมื่อเปรียบเทียบกับความคลาดเคลื่อนเชิงระบบที่น้อยที่สุด พบว่าทั้งสองแกนมีค่าจุดตัดแกนที่มากกว่าความละเอียด

เมื่อคำนวณค่าเบี่ยงเบนมาตรฐานซึ่งแสดงถึงความเที่ยงตรงเชิงสถิติ พบว่าทั้งสองแกนมีแนวโน้มที่ค่าความเบี่ยงเบนมาตรฐานจะมีค่าเพิ่มขึ้นตามความยาวเช่นเดียวกับค่าความผิดพลาดสัมพัทธ์ จากการวิเคราะห์การถดถอยเชิงเส้นพบว่าความคลาดเคลื่อนเชิงสถิติในแกนเอกซ์และวายเป็น 4.54 และ 5.59 เปอร์เซ็นต์ตามลำดับ

ในการเปรียบเทียบผู้ทำการทดลองได้กำหนดเกณฑ์ของความผิดพลาดสัมพัทธ์และค่าความแม่นยำสัมพัทธ์ที่ยอมรับได้ของเครื่องพล็อตเตอร์คือ 10 เปอร์เซ็นต์ ในด้านความแม่นยำเมื่อลองเปรียบเทียบกับค่าความผิดพลาดสัมพัทธ์ที่ได้พบว่าทั้งแกนเอกซ์และแกนวายมีค่าความผิดพลาดที่ 9.09 และ 6.89 เปอร์เซ็นตามลำดับ ซึ่งอยู่ในเกณฑ์ที่ยอมรับได้ในโครงการนี้ ส่วนในด้านของความเที่ยงตรงมีค่าความแม่นยำสัมพัทธ์ที่ได้ของแกนเอกซ์และวายคือ 4.54 และ 5.59 เปอร์เซ็นตามลำดับ ซึ่งอยู่ในเกณฑ์ที่ยอมรับได้เช่นกัน ถึงแม้จะอยู่ในเกณฑ์ที่ยอมรับได้ในโครงการแต่ว่าค่าทั้งสองค่อนข้างสูงเมื่อเทียบกับงานทางด้านการพิมพ์

จากผลการทดลองการขีดเส้นตรงจะพบว่ามีปัญหาในการใช้งานเพิ่มเติมนอกเหนือจากการทำงานแบบอิสระในแต่ละแกนดังนี้

1. แรงต้านที่จุดปลายปากกา

เนื่องจากลักษณะการกดของปากกาที่ใช้ทดลองในเครื่องพล็อตเตอร์ ถูกควบคุมด้วยเซอร์โวมอเตอร์แบบ 180 องศา เมื่อทำการเปิดปากกาจะทำให้เกิดแรงกดขึ้นที่ปลายปากกากระทำกับกระดาษ แรงกดนี้จะทำให้เครื่องมีแรงเสียดทานใน

แนวที่กำลังเคลื่อนที่มากขึ้น นอกจากนั้นตรงปลายปากกาที่ถูกกดจะมีลักษณะเป็นจุดหมุน ซึ่งทำให้เกิดทอร์คจากแรงเสียดทาน ทำให้เครื่องพล็อตเตอร์เกิดการส่าย ทั้งสองสิ่งนี้ทำให้เครื่องลากเส้นไม่ตรงและมีอาการหงิกงอ

2. แรงต้านจากการเสียดทานที่เกิดจากการที่ล้อทั้งสองแกนเคลื่อนที่แบบไม่เป็นอิสระต่อกัน

เนื่องจากการที่ล้อทั้ง 4 วางบนพื้นพร้อมกันแม้ขณะไม่ได้ใช้งาน ทำให้การเคลื่อนที่ในแกนใดแกนหนึ่งนั้นย่อมมีแรงต้านที่เกิดจากล้อที่ตั้งฉากกันอยู่

3. ความไม่สม่ำเสมอและความเอียงของพื้นผิวที่เป็นฐานการทดลอง

เนื่องจากเครื่องพล็อตเตอร์ที่ได้สร้างขึ้น ล้อทั้งสี่จำเป็นที่จะต้องแนบติดสนิทกับพื้นผิวตลอดเวลาที่เคลื่อนที่ เมื่อเคลื่อนที่ไประยะหนึ่ง อาจจะพบปัญหาเช่น พื้นที่ไม่สม่ำเสมอ ซึ่งทำให้ล้อทั้งสี่ไม่แนบสนิทกับพื้น ซึ่งในการทดลองจริงมักจะพบในการขีดเส้นที่ยาว เมื่อเครื่องเคลื่อนที่ถึงจุดที่พื้นผิวไม่สม่ำเสมอจะเกิดอาการเครื่องส่าย สามารถสังเกตได้จากการลากเส้นตรงที่จะมีการขยับซ้ายขวาเล็ก ๆ หรือเมื่อเจอพื้นที่ที่มีการเอียงมาก ๆ จนล้อลอยจะทำให้ตัวเครื่องหยุดเคลื่อนที่ ทั้งหมดนี้เป็นผลให้ความยาวของเส้นที่ลากได้มีค่าความผิดพลาดสัมพัทธ์และค่าเบี่ยงเบนมาตรฐานที่สูง

4. ช่องว่างของล้อออมนิไดเรกชั่นนอลวีลทำให้แรงเสียดทานที่เกิดมีค่าเปลี่ยนแปลง

เนื่องจากล้อออมนิไดเรกชั่นนอลวีลเท่าที่ผู้จัดทำสามารถหามาได้ในช่วงการเกิดการระบาดของโควิด-19 มีราคาไม่แพงมาก ช่องว่างระหว่างล้อจึงกว้าง ทำให้เกิดความไม่ต่อเนื่องในขณะหมุน เมื่อมีการเคลื่อนที่ในแนวตั้งฉากกับล้อจึงทำให้แรงเสียดทานที่เกิดจากล้อมีไม่สม่ำเสมอ พร้อมทั้งล้อของแนวแกนเอกซ์และแกนวายยังมีขนาด น้ำหนัก และวัสดุแตกต่างกัน ทำให้เกิดความผิดพลาดของทั้งสองแกนไม่เท่ากัน

จากการทดลอง การเขียนงานจริงพบว่าเครื่องพล็อตเตอร์มีความสามารถในการเขียนเส้นตรงในแนวเอกซ์ วายได้แม้มีความคลาดเคลื่อนไปบ้างไม่เกิน 10 เปอร์เซ็นต์ แต่มีจุดบกพร่องในการลากเส้นตรงในแนวทแยงและวงกลม เนื่องจากการเคลื่อนที่ของแกนทั้งสองไม่ได้ทำการประสานจังหวะ (synchronization) กันอย่างสมบูรณ์ ค่าความผิดพลาดสะสมที่ไม่เท่ากันและมีค่าเพิ่มเมื่อลากเส้นที่ยาวขึ้น ซึ่งเป็นผลมาจากค่าความแม่นยำและความเที่ยงตรงที่ต่ำ สังเกตได้จากความคลาดเคลื่อนสัมพัทธ์ที่สูงขึ้น นอกจากนั้นการที่เส้นในแนวทแยงและวงกลมไม่ออกมาตามแบบที่ต้องการนั้นเกิดจากการที่ล้อของทั้งสองแกนของเครื่องพล็อตเตอร์ มีแรงต้านจากการเสียดทานที่ต่างกันเพราะทั้งคู่ไม่เป็นอิสระต่อกัน รวมถึงแรงต้านจากปลายปากกาที่ส่งผลให้ทั้งสองแกนก็ต่างกัน ทั้งหมดนี้เป็นผลให้ล้อของทั้งสองแกนไม่สามารถเคลื่อนที่ไปด้วยกันอย่างสมบูรณ์ได้

5.3 ปัญหาที่พบ

1. การเคลื่อนที่ของเครื่องพล็อตเตอร์นั้นเกิดความผิดพลาดค่อนข้างสูง เนื่องจากมีตัวแปรที่ไม่สามารถควบคุมได้หลายอย่างเช่นแรงเสียดทาน ลักษณะพื้นผิวที่ทำการทดลอง ความลาดเอียงเพียงเล็กน้อยก็ส่งผลต่อการทำงานของเครื่อง และเครื่องพล็อตเตอร์ที่ได้ประดิษฐ์มีน้ำหนักที่มาก จึงจำเป็นต้องใช้มอเตอร์ที่มีความสามารถในการให้แรงบิดที่สูงซึ่งไม่เหมาะกับงานที่มีความจำเป็นต้องเคลื่อนที่

2. จากการพยายามเพิ่มความสามารถในการเคลื่อนที่ของเครื่องพล็อตเตอร์ ซึ่งจำเป็นจะต้องให้เครื่องเคลื่อนที่ได้ในทุกทิศทาง ทำให้เครื่องทำงานผิดพลาดไปมากเนื่องจากลักษณะล้อที่ใช้มีขนาดใหญ่ ทำให้มีความละเอียดลดลงรวมถึงล้อที่นำมาใช้ก็มีความไม่สมบูรณ์ทั้งในด้านการหมุน การยึดเกาะ และในโครงการนี้จำเป็นที่จะต้องใช้ล้อที่ขนาดต่างกันทำให้การหมุนไม่เป็นไปตามที่ตั้งใจไว้ ซึ่งทั้งหมดส่งผลต่อความสามารถของเครื่อง

3. โครงการนี้ค่อนข้างมีความจำเป็นที่จะต้องใช้ส่วนประกอบที่พอดีกัน แต่การขาดแคลนอุปกรณ์การสร้างเนื่องจากการระบาดของโควิด-19 ทำให้ไม่สามารถจัดหาอุปกรณ์ที่ทำงานสอดประสานดังความตั้งใจได้

5.4 ข้อเสนอแนะ

จากการศึกษาของโครงการนี้ พบว่า เครื่องพล็อตเตอร์มีข้อจำกัด และมีความผิดพลาดมากจนไม่สามารถใช้งานจริงได้ จึงทำให้จำเป็นต้องมีการแก้ไขหากต้องการนำไปใช้งานจริง โดยแนวคิดของผู้ทำโครงการสามารถสรุปมาเป็นข้อ ๆ ได้ดังนี้

1. เราสามารถพยายามทำให้ล้อของทั้งสองแกนสามารถเคลื่อนที่ควบคู่กันอย่างสมบูรณ์ได้ด้วยการนำตัวแปรที่ส่งผลกระทบต่อการเคลื่อนที่ต่าง ๆ มาทำการสอบเทียบ แต่ว่าวิธีนี้ค่อนข้างที่จะทำได้ยากเนื่องจากมีตัวแปรที่ส่งผลกระทบต่อการเคลื่อนที่หลายตัว และค่าเหล่านั้นค่อนข้างที่จะมีความแปรปรวนสูง
2. เราสามารถลดความผิดพลาดที่เกิดจากล้อไม่เท่ากันได้ด้วยการเปลี่ยนเป็นการใช้มอเตอร์ 4 ตัวแทนการใช้มอเตอร์ 2 ตัว เนื่องจากการใช้มอเตอร์ 4 ตัวและให้มอเตอร์แต่ละตัวสามารถควบคุม 1 ล้อได้โดยตรง ซึ่งจะทำให้สามารถใช้ล้อที่มีขนาดเท่ากันทุกล้อได้ และยังสามารถลดความยุ่งยากจากการใช้มู่เล่ย์และสายพานจีทีสองด้วย
3. สามารถหาซื้อหรือสร้างล้อที่สามารถเคลื่อนที่ได้ดีกว่า หรือแรงเสียดทานที่เกิดจากล้อน้อยกว่าจะทำให้เครื่องพล็อตเตอร์มีความแม่นยำมากขึ้น
4. สามารถลดความยุ่งเหยิงของสายไฟของเครื่องด้วยการใช้ ไมโครคอนโทรลเลอร์ที่สามารถควบคุมแบบไร้สายได้ เช่น ESP32 เป็นต้น

บรรณานุกรม

[1] What is Arduino Why Arduino and how do I use Arduino. Retrieved 5 February 2017, from

https://www.arduino.cc/en/guide/introduction

[2] CHEAP CNC XY PLOTTER (article) | engineering | Bangladesh University of Engineering and Technology. Retrieved 9 August 2018, from

https://www.researchgate.net/publication/340132457_CHEAP_CNC_XY_PLOTTER

[3] Peter, The difference between Unipolar and Bipolar stepper motors. Retrieved 11 June 2018, from

: https://techexplorations.com/blog/arduino/blog-the-difference-between-unipolar-and-bipolar-stepper-motors/

[4] ROHM Semiconductor company, Selecting Stepper Motor Driver for Optimum Performance, Retrieved 4 January 2009, from

https://www.rohm.com/electronics-basics

[5] Nikhil Agnihotri, Stepper Motor : Basics, Types and Working (article). Retrieved 18 February 2011, from

https://www.engineersgarage.com/article_page/stepper-motor-basics-types-and-working/(2020)

[6] Pinetree Electronic LTD, Arduino CNC shield Installation Guide. Retrieved 4 July 2017, from

https://osoyoo.com/2017/04/07/arduino-uno-cnc-shield-v3-0-a4988/

[7] Using Omni Wheels to make Holonomic drive (article) | Yale engineering. Retrieved 14 June 2018, from

https://seas.yale.edu/sites/default/files/imce/other/HolonomicOmniWheelDrive.pdf

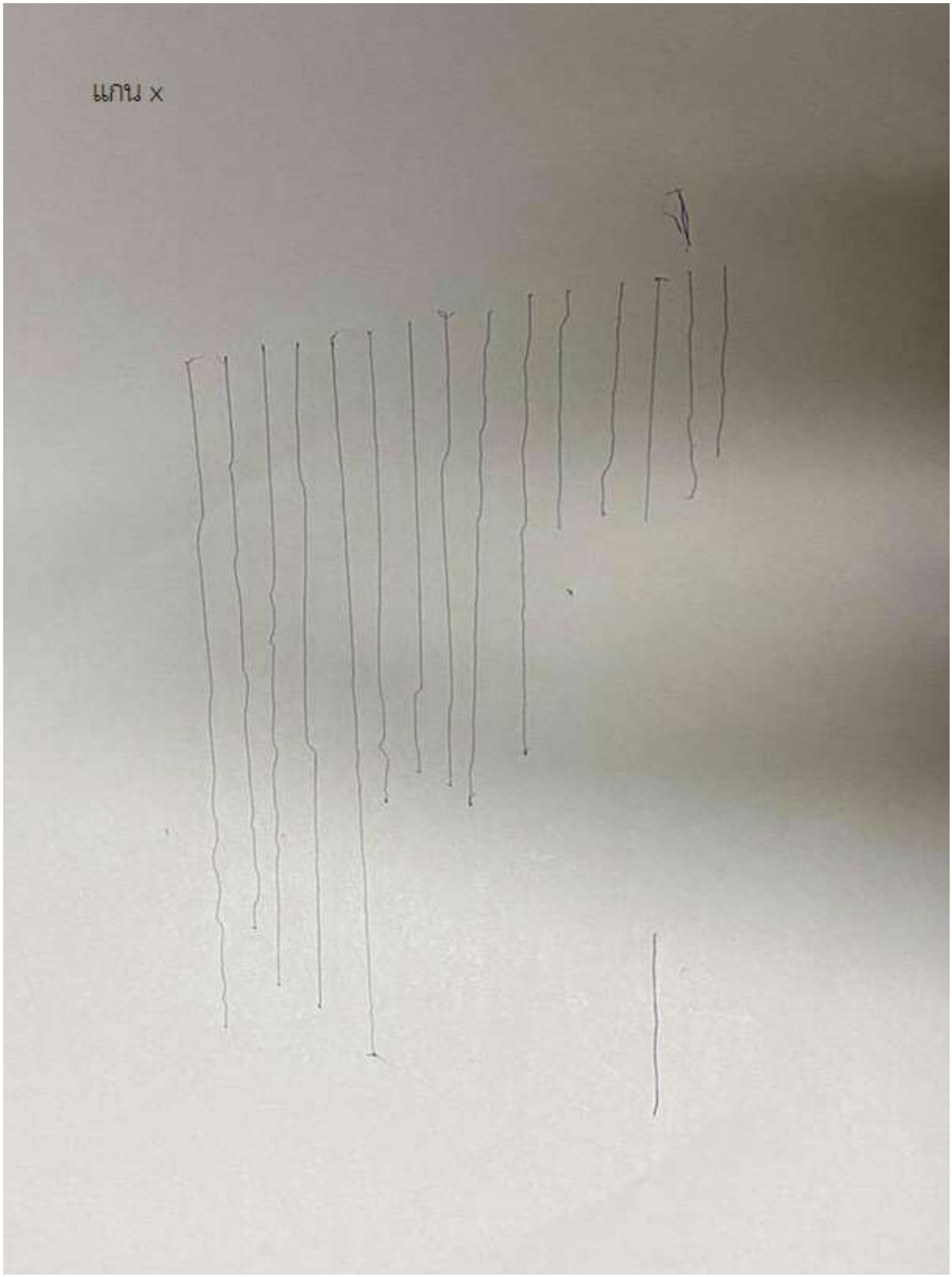
[8] Jeremy S Cook, Grbl: Inexpensive and Easy Machine Control with Arduino. Retrieved 26 December 2015, from

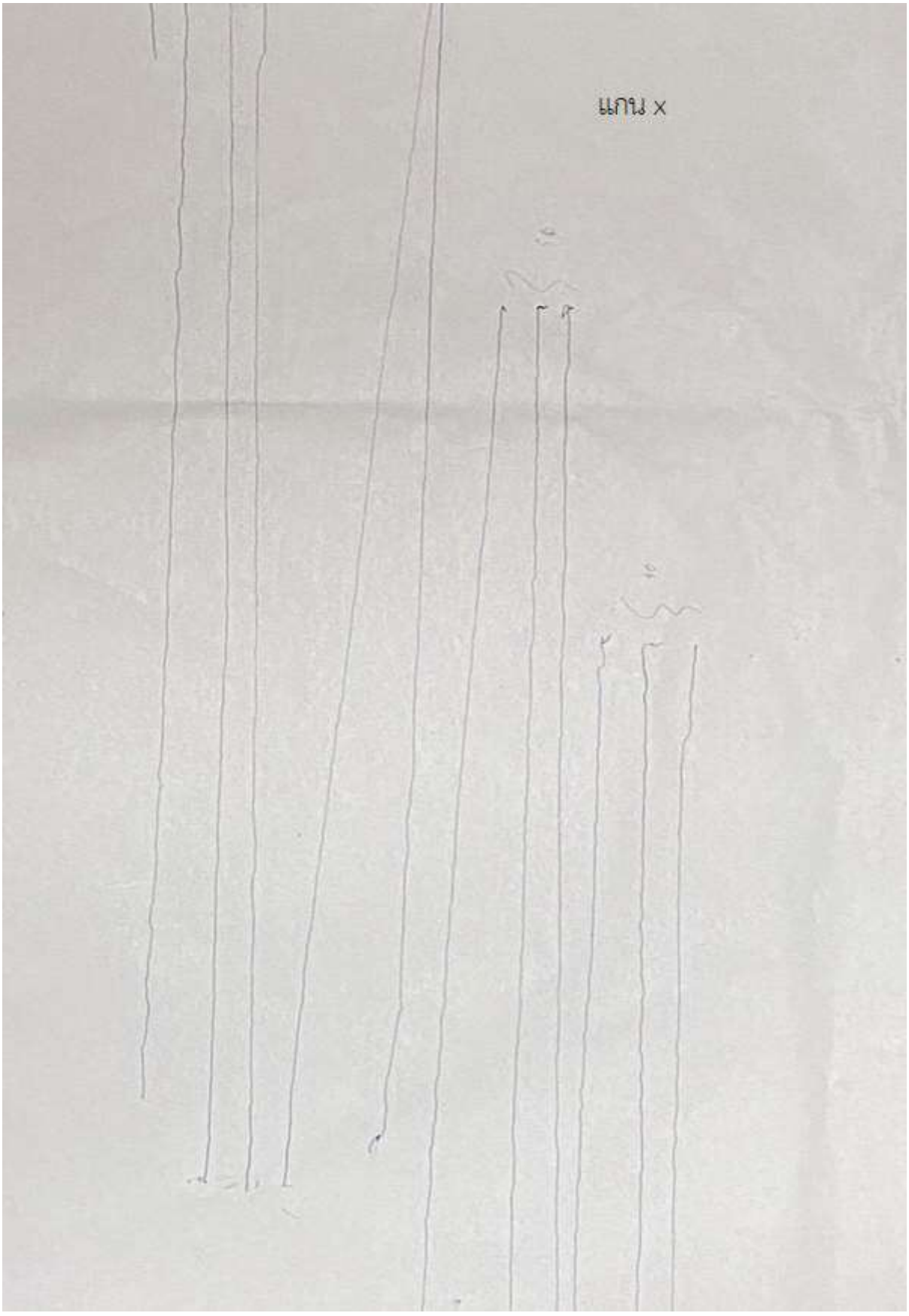
https://www.arrow.com/en/research-and-events/articles/grbl-inexpensive-and-easy-machine-control-with-arduino

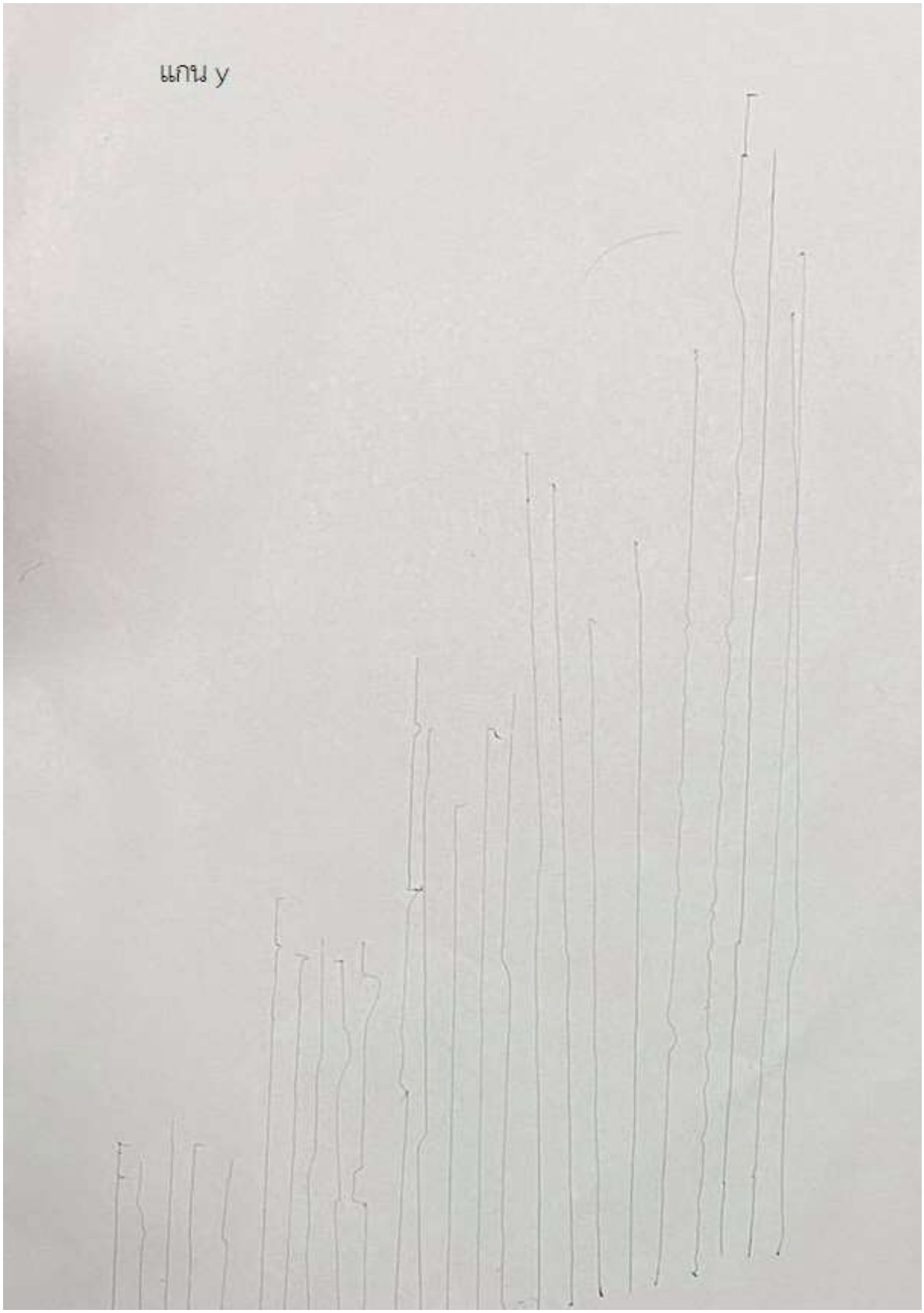
ภาคผนวก

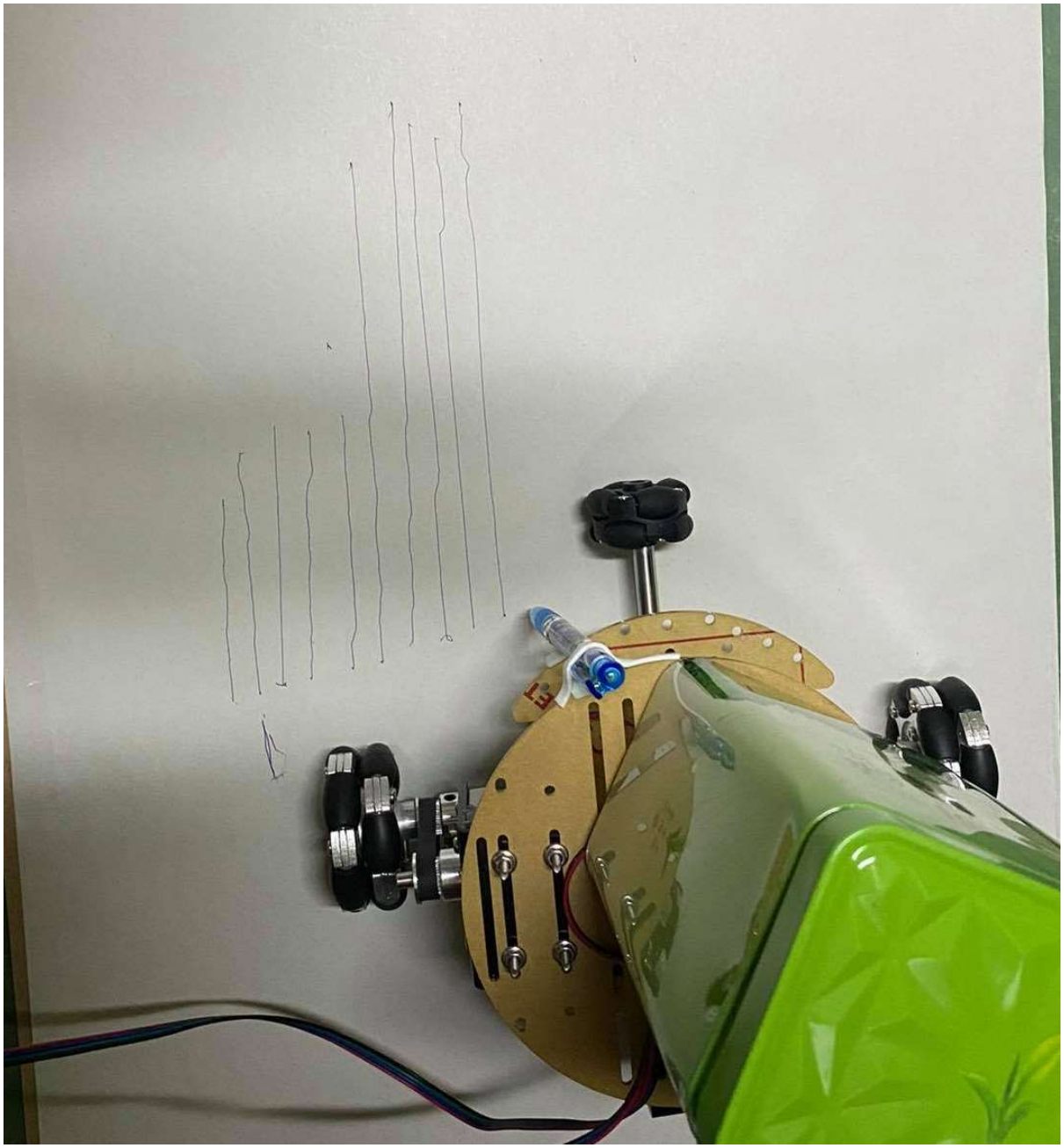
ภาคผนวก ก

การวัดความยาวเส้นตรงของทั้งสองแกน




แกน x


แกน y



ภาคผนวก ข

```
/*
 Mini CNC Plotter firmware, based in TinyCNC https://github.com/MakerBlock/TinyCNC-Sketches
 Send GCODE to this Sketch using gctrl.pde https://github.com/damellis/gctrl
 Convert SVG to GCODE with MakerBot Unicorn plugin for Inkscape available here
https://github.com/martymcguire/inkscape-unicorn

 More information about the Mini CNC Plotter here (german, sorry):
http://www.makerblog.at/2015/02/projekt-mini-cnc-plotter-aus-alten-cddvd-laufwerken/
  */

#include <Servo.h>
#include <Stepper.h>

#define LINE_BUFFER_LENGTH 512

// Servo position for Up and Down
const int penZUp = 80;
const int penZDown = 40;

// Servo on PWM pin 6
const int penServoPin = 6;

// Should be right for DVD steppers, but is not too important here
const int stepsPerRevolution = 20;

// create servo object to control a servo
Servo penServo;

// Initialize steppers for X- and Y-axis using this Arduino pins for the L293D H-bridge
Stepper myStepperY(stepsPerRevolution, 2,3,4,5);
Stepper myStepperX(stepsPerRevolution, 8,9,10,11);

/* Structures, global variables    */
struct point {
```

```
  float x;
  float y;
  float z;
};

// Current position of plothead
struct point actuatorPos;

//  Drawing settings, should be OK
float StepInc = 1;
int StepDelay = 0;
int LineDelay = 50;
int penDelay = 50;

// Motor steps to go 1 millimeter.
// Use test sketch to go 100 steps. Measure the length of line.
// Calculate steps per mm. Enter here.
float StepsPerMillimeterX = 11.42;
float StepsPerMillimeterY = 17.45;

// Drawing robot limits, in mm
// OK to start with. Could go up to 50 mm if calibrated well.
float Xmin = 0;
float Xmax = 500;
float Ymin = 0;
float Ymax = 500;
float Zmin = 0;
float Zmax = 1;

float Xpos = Xmin;
float Ypos = Ymin;
float Zpos = Zmax;

// Set to true to get debug output.
boolean verbose = false;
```

```
//  Needs to interpret
//  G1 for moving
//  G4 P100 (wait 100ms)
//  M3 S30 (pen down)
//  M3 S50 (pen up)
//  Discard anything with a (
//  Discard any other command!

/**********************
 * void setup() - Initialisations
 ***********************/
void setup() {
  //  Setup
  Serial.begin( 11500 );

  penServo.attach(penServoPin);
  penServo.write(penZUp);
  delay(200);

  // Decrease if necessary
  myStepperX.setSpeed(250);
  myStepperY.setSpeed(250);

  //  Set & move to initial default position
  // TBD

  //  Notifications!!!
  Serial.println("Mini CNC Plotter alive and kicking!");
  Serial.print("X range is from ");
  Serial.print(Xmin);
  Serial.print(" to ");
  Serial.print(Xmax);
  Serial.println(" mm.");
  Serial.print("Y range is from ");
  Serial.print(Ymin);
  Serial.print(" to ");
```

```
  Serial.print(Ymax);
  Serial.println(" mm.");
}

/***********************
 * void loop() - Main loop
 ***********************/
void loop()
{
  delay(200);
  char line[ LINE_BUFFER_LENGTH ];
  char c;
  int lineIndex;
  bool lineIsComment, lineSemiColon;

  lineIndex = 0;
  lineSemiColon = false;
  lineIsComment = false;

  while (1) {

    // Serial reception - Mostly from Grbl, added semicolon support
    while ( Serial.available()>0 ) {
      c = Serial.read();
      if (( c == '\n') || (c == '\r') ) {              // End of line reached
        if ( lineIndex > 0 ) {                         // Line is complete. Then execute!
          line[ lineIndex ] = '\0';                    // Terminate string
          if (verbose) {
            Serial.print( "Received : ");
            Serial.println( line );
          }
          processIncomingLine( line, lineIndex );
          lineIndex = 0;
        }
        else {
          // Empty or comment line. Skip block.
```

```
        }
        lineIsComment = false;
        lineSemiColon = false;
        Serial.println("ok");
      }
      else {
        if ( (lineIsComment) || (lineSemiColon) ) {   // Throw away all comment characters
          if ( c == ')' )  lineIsComment = false;     // End of comment. Resume line.
        }
        else {
          if ( c <= ' ' ) {                           // Throw away whitepace and control characters
          }
          else if ( c == '/' ) {                      // Block delete not supported. Ignore character.
          }
          else if ( c == '(' ) {                      // Enable comments flag and ignore all characters until ')' or
EOL.
            lineIsComment = true;
          }
          else if ( c == ';' ) {
            lineSemiColon = true;
          }
          else if ( lineIndex >= LINE_BUFFER_LENGTH-1 ) {
            Serial.println( "ERROR - lineBuffer overflow" );
            lineIsComment = false;
            lineSemiColon = false;
          }
          else if ( c >= 'a' && c <= 'z' ) {        // Upcase lowercase
            line[ lineIndex++ ] = c-'a'+'A';
          }
          else {
            line[ lineIndex++ ] = c;
          }
        }
      }
    }
  }
```

```
}

void processIncomingLine( char* line, int charNB ) {
  int currentIndex = 0;
  char buffer[ 64 ];                    // Hope that 64 is enough for 1 parameter
  struct point newPos;

  newPos.x = 0.0;
  newPos.y = 0.0;

  //  Needs to interpret
  //  G1 for moving
  //  G4 P150 (wait 150ms)
  //  G1 X60 Y30
  //  G1 X30 Y50
  //  M3 S30 (pen down)
  //  M3 S50 (pen up)
  //  Discard anything with a (
  //  Discard any other command!

  while( currentIndex < charNB ) {
    switch ( line[ currentIndex++ ] ) {        // Select command, if any
    case 'U':
      penUp();
      break;
    case 'D':
      penDown();
      break;
    case 'G':
      buffer[0] = line[ currentIndex++ ];      // /!\ Dirty - Only works with 2 digit commands
      //      buffer[1] = line[ currentIndex++ ];
      //      buffer[2] = '\0';
      buffer[1] = '\0';

      switch ( atoi( buffer ) ){               // Select G command
      case 0:                                  // G00 & G01 - Movement or fast movement. Same here
```

```
  case 1:
    // /!\ Dirty - Suppose that X is before Y
    char* indexX = strchr( line+currentIndex, 'X' );  // Get X/Y position in the string (if any)
    char* indexY = strchr( line+currentIndex, 'Y' );
    if ( indexY <= 0 ) {
      newPos.x = atof( indexX + 1);
      newPos.y = actuatorPos.y;
    }
    else if ( indexX <= 0 ) {
      newPos.y = atof( indexY + 1);
      newPos.x = actuatorPos.x;
    }
    else {
      newPos.y = atof( indexY + 1);
      indexY = '\0';
      newPos.x = atof( indexX + 1);
    }
    drawLine(newPos.x, newPos.y );
    //         Serial.println("ok");
    actuatorPos.x = newPos.x;
    actuatorPos.y = newPos.y;
    break;
  }
  break;
case 'M':
  buffer[0] = line[ currentIndex++ ];         // /!\ Dirty - Only works with 3 digit commands
  buffer[1] = line[ currentIndex++ ];
  buffer[2] = line[ currentIndex++ ];
  buffer[3] = '\0';
  switch ( atoi( buffer ) ){
  case 300:
    {
      char* indexS = strchr( line+currentIndex, 'S' );
      float Spos = atof( indexS + 1);
      //           Serial.println("ok");
      if (Spos == 30) {
```

```
            penDown();
          }
          if (Spos == 50) {
            penUp();
          }
          break;
        }
      case 114:                              // M114 - Repport position
        Serial.print( "Absolute position : X = " );
        Serial.print( actuatorPos.x );
        Serial.print( "  -  Y = " );
        Serial.println( actuatorPos.y );
        break;
      default:
        Serial.print( "Command not recognized : M");
        Serial.println( buffer );
      }
    }
  }



}


/**********************************
 * Draw a line from (x0;y0) to (x1;y1).
 * Bresenham algo from https://www.marginallyclever.com/blog/2013/08/how-to-build-an-2-axis-arduino-cnc-gcode-interpreter/
 * int (x1;y1) : Starting coordinates
 * int (x2;y2) : Ending coordinates
 **********************************/
void drawLine(float x1, float y1) {

  if (verbose)
  {
```

```
  Serial.print("fx1, fy1: ");
  Serial.print(x1);
  Serial.print(",");
  Serial.print(y1);
  Serial.println("");
}

//  Bring instructions within limits
if (x1 >= Xmax) {
  x1 = Xmax;
}
if (x1 <= Xmin) {
  x1 = Xmin;
}
if (y1 >= Ymax) {
  y1 = Ymax;
}
if (y1 <= Ymin) {
  y1 = Ymin;
}

if (verbose)
{
  Serial.print("Xpos, Ypos: ");
  Serial.print(Xpos);
  Serial.print(",");
  Serial.print(Ypos);
  Serial.println("");
}

if (verbose)
{
  Serial.print("x1, y1: ");
  Serial.print(x1);
  Serial.print(",");
  Serial.print(y1);
```

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

//  Convert coordinates to steps
x1 = (int)(x1*StepsPerMillimeterX);
y1 = (int)(y1*StepsPerMillimeterY);
float x0 = Xpos;
float y0 = Ypos;

//  Let's find out the change for the coordinates
long dx = abs(x1-x0);
long dy = abs(y1-y0);
int sx = x0<x1 ? StepInc : -StepInc;
int sy = y0<y1 ? StepInc : -StepInc;

long i;
long over = 0;

if (dx > dy) {
  for (i=0; i<dx; ++i) {
    myStepperX.step(sx);
    over+=dy;
    if (over>=dx) {
      over-=dx;
      myStepperY.step(sy);
    }
    delay(StepDelay);
  }
}
else {
  for (i=0; i<dy; ++i) {
    myStepperY.step(sy);
    over+=dx;
    if (over>=dy) {
      over-=dy;
      myStepperX.step(sx);
```

```
      }
      delay(StepDelay);
    }
  }

  if (verbose)
  {
    Serial.print("dx, dy:");
    Serial.print(dx);
    Serial.print(",");
    Serial.print(dy);
    Serial.println("");
  }

  if (verbose)
  {
    Serial.print("Going to (");
    Serial.print(x0);
    Serial.print(",");
    Serial.print(y0);
    Serial.println(")");
  }

  //  Delay before any next lines are submitted
  delay(LineDelay);
  //  Update the positions
  Xpos = x1;
  Ypos = y1;
}

//  Raises pen
void penUp() {
  penServo.write(penZUp);
  delay(LineDelay);
  Zpos=Zmax;
  if (verbose) {
```

```
    Serial.println("Pen up!");
  }
}
//  Lowers pen
void penDown() {
  penServo.write(penZDown);
  delay(LineDelay);
  Zpos=Zmin;
  if (verbose) {
    Serial.println("Pen down.");
  }
}
```