



หุ่นยนต์เอไอจับน้องไก่

AI&Robot Poultry Predator

ชนกร หนองหารพิทักษ์

ชนกฤต เจนธนุรเวท

หลักสูตรวิศวกรรมการวัดและการควบคุม

ภาควิชาแมคคาทรอนิกส์และอัตโนมัติ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2566

ภาควิชาแมคคาทรอนิกส์และอัตโนมัติ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง หุ่นยนต์เอไอจับน้องไก่

AI&Robot Poultry Predator

ผู้จัดทำ ธนกร หนองหารพิทักษ์ 63010391

นายชนกฤต เจริญรเวท 63010393

อาจารย์ที่ปรึกษา



(ผศ.เทพจิตรี เชาโยภา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโปรเจค หุ่นยนต์เอไอจับน้องไก่
นักศึกษา นายชนกร หนองหารพิทักษ์
 นายชนกฤต เจริญรเวท
ภาควิชา แมคคาทรอนิกส์และอโตเมชัน
คณะ วิศวกรรมศาสตร์
อาจารย์ที่ปรึกษา ผศ.เทพจิตร์ เชยโกคา
ปีการศึกษา 2566

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ นำเสนอเกี่ยวกับการออกแบบตัวแขนกลโดยมีต้นแบบมาจาก AR4 การออกแบบโครงสร้างต่างๆ ถูกออกแบบด้วยโปรแกรม SOLIDWORKS ในส่วนของการควบคุมแขนกลมีการนำระบบปฏิบัติการหุ่นยนต์หรือ ROS เข้ามาใช้ในการควบคุม ซึ่งเขียนโดยภาษา Python ในการเขียนโปรแกรมการทำงานของแขนกล อีกทั้งยังมีการติดตั้งกล้อง Xbox Kinect และใช้ AI ประยุกต์ใช้เทคนิคการรู้จำวัตถุ (Object Detection) ในการตรวจจับวัตถุ ซึ่งวัตถุที่เรากำหนดให้ AI ตรวจจับก็คือสะโพกไก่ จุดมุ่งหมายของโครงการนี้คือสามารถควบคุมแขนกลและใช้ในการหยิบวัตถุที่ AI ตรวจจับได้ โดยมีการสั่งการจากคอมพิวเตอร์ และสามารถควบคุมแขนกลให้สามารถทำงานได้อย่างเป็นระบบตามที่วางแผน

Project title : AI&Robot Poultry Predator
Student name : Thanakorn Nonghanphitak
Thanakit Jenthanurawate
Department : Mechatronics and Automation
Faculty : School of Engineering
Mentor name : Asst. Prof. Thepjit Cheypoca

Year 2566

ABSTRACT

The thesis presents the design of a robotic arm modeled after the AR4. Various structural elements Designed with SOLIDWORKS program. As for controlling the robotic arm, Robot Operating System or ROS is used to control it using the Python language. In the robotic arm control system, there is an Xbox Kinect camera and AI with object detection which the object of AI is a chicken. The highlight of the project is a robotic arm and collection of AI-generated objects by commanding the computer. And the robotic arm as the main planning principles.

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้จนสำเร็จได้ด้วยดี ทางคณะผู้จัดทำต้องขอขอบพระคุณอาจารย์ที่ปรึกษา ศศ.เทพจิตร เขยโกคา เป็นอย่างสูง ที่ได้ให้คำปรึกษาแนวทางการแก้ไขปัญหา ตลอดจนการทำวิจัย ทำให้ทางคณะผู้จัดทำสามารถทำวิจัยได้อย่างมีประสิทธิภาพ รวมถึงความเอาใจใส่ดูแล และมีการสอบถามถึงความคืบหน้าของงานวิจัยอย่างสม่ำเสมอ ทำให้ผู้จัดทำสามารถทำงานได้อย่างเป็นระบบและใช้เวลาได้อย่างมีประสิทธิภาพ

ขอขอบคุณอาจารย์ประจำภาควิชาทุกท่านที่ให้ความรู้ทั้งทางทฤษฎี และปฏิบัติที่สามารถนำมาประยุกต์ใช้ในการทำงานวิจัยครั้งนี้

ขอขอบคุณนางแมคคาทรอนิกส์ ที่เอื้อเฟื้ออุปกรณ์และสถานที่ในการทำวิจัย จนผู้จัดทำสามารถทำวิทยานิพนธ์ฉบับนี้เสร็จสิ้น

ขอขอบคุณเพื่อนๆ ทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ คอยถามความคืบหน้าของโครงการอยู่เสมอ จนสามารถสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ตลอด และเป็นแรงบันดาลใจที่ดีที่สุดจนทำให้โครงการนี้สำเร็จสมบูรณ์ลงได้

ผู้จัดทำ

นายชนกร หนองหารพิทักษ์

นายชนกฤต เจนธนุรเวท

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ข
ABSTRACT	ค
กิตติกรรมประกาศ	ง
สารบัญ	จ
สารบัญตาราง	ช
สารบัญรูป	ซ
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง	
2.1 แขนกล	4
2.2 ROS (Robot Operating System)	4
2.3 Library ที่ใช้ภายใน ROS	6
2.4 ไมโครคอนโทรลเลอร์	7
2.5 การรู้จำวัตถุ (Object Detection)	10
2.6 โพรโตคอล SPI (Serial Peripheral Interface)	11
2.7 สเต็ปเปอร์มอเตอร์	12
2.8 สเต็ปเปอร์มอเตอร์ไครฟ์เวอร์	13
2.9 ซอฟต์แวร์ที่เกี่ยวข้อง	14

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 3 วิธีดำเนินงาน	
3.1 กำหนดหัวข้อและขอบเขตการจัดทำโครงการ	18
3.2 ศึกษาทฤษฎีเบื้องต้น	18
3.3 ศึกษาการทำงานของแขนกลจากต้นแบบที่เรานำมาใช้	19
3.4 ออกแบบแขนกลขึ้นมาใหม่โดยอ้างอิงต้นแบบจากแขนกล AR4	19
3.5 เก็บ Dataset รูปภาพสำหรับการเทรน Image Detection	20
3.6 เขียน โปรแกรมการทำงานของแขนกล	20
3.7 ทดสอบการทำงานของแขนกลที่ออกแบบและตรวจสอบอีกครั้ง	27
3.8 จัดซื้ออุปกรณ์ต่าง ๆ	27
3.9 ประกอบแขน	38
3.10 ทดสอบการทำงานและแก้ไขข้อผิดพลาด	40
3.11 สรุปผลและจัดทำรายงาน	40
บทที่ 4 ผลการดำเนินงาน	
4.1 ผลการจัดทำแขนกล	41
4.2 ผลการเขียน โปรแกรมควบคุมแขนกล	43
4.3 ผลการเขียนบอร์ด STM32 และ ESP32	45
4.4 ผลการเขียน โปรแกรม Image Detection	46
บทที่ 5 สรุปผลและข้อเสนอแนะ	
5.1 สรุปผล	47
5.2 ปัญหาและอุปสรรคที่พบ	48
5.3 แนวทางการแก้ไข	49
5.4 ข้อเสนอแนะ	49
เอกสารอ้างอิง	50
ประวัติผู้วิจัย	51

สารบัญตาราง

ตารางที่

หน้า

3.1 ตารางรายการสั่งซื้อนี้้อต

32



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปร่างภาพ

รูปร่างภาพที่	หน้า
1.1 แบบจำลองแขนกล	1
2.1 โมเดลแขนกล AR4	4
2.2 LOGO ของ ROS 1 noetic	5
2.3 การส่ง message จาก publisher node ไปยัง subscriber node	5
2.4 โลโก้ของ Moveit	6
2.5 โลโก้ของ Rviz	6
2.6 การส่งข้อมูลของ roserial	7
2.7 บอร์ด STM32F103C8T6	8
2.8 บอร์ด ESP32	9
2.9 Image Detection โดย YOLO	10
2.10 การทำงานของ SPI	11
2.11 สเต็ปเปอร์มอเตอร์	12
2.12 สเต็ปเปอร์มอเตอร์ไครฟ์เวอร์	13
2.13 โปรแกรม SOLIDWORKS 2022	14
2.14 หน้าต่างการใช้งานโปรแกรม SOLIDWORKS 2022	15
2.15 โปรแกรม Visual Studio Code	15
2.16 หน้าต่างการใช้งานโปรแกรม Visual Studio Code	16
2.17 โปรแกรม Visual Studio 2022	17
2.18 หน้าต่างการใช้งานโปรแกรม Visual Studio 2022	17
2.19 หน้าต่างการใช้งานโปรแกรม Arduino IDE	18
2.20 หน้าต่างการใช้งานโปรแกรม Arduino IDE	18
3.1 ตัวอย่างการออกแบบแขนกลโดยโปรแกรม SOLIDWORKS	20
3.2 ตัวอย่างร่องไก่อปรีน 3 มิติ	21
3.3 ตัวอย่างไฟล์ URDF	21

สารบัญรูปภาพ (ต่อ)

รูปภาพที่	หน้า
3.4 Workflow ของโปรแกรม	22
3.5 หน้าต่างโปรแกรม Moveit Setup Assistant ที่โหลดไฟล์ URDF เข้ามาแล้ว	23
3.6 ตัวอย่างไฟล์ SRDF	23
3.7 ตัวอย่างโปรแกรม interface ในการควบคุมแขนกลโดย Moveit	23
3.8 ตัวอย่างโปรแกรมในบอร์ด STM32	24
3.9 ตัวอย่างโปรแกรมในบอร์ด ESP32	25
3.10 ตัวอย่างโปรแกรม Image Detection	26
3.11 หน้าต่างโปรแกรม Rviz ที่จำลองและแสดงโมเดลแขนกลแบบ 3 มิติ	27
3.12 32009 (45x75x20mm) taper roller bearing	28
3.13 30206 (30x62x17.25mm) taper roller bearing	28
3.14 30204 (20x47x15.25mm) taper roller bearing	28
3.15 AXK3552/AS3552 (35x52x4mm) thrust bearing with washers	29
3.16 NTA1625 (1.00x1.5625x0.0781 inch) thrust bearing	29
3.17 TRA1625 (1.000x1.5625x0.0312 inch) thrust washers	29
3.18 TRD1625 (1.000x1.5625x0.125 inch) thrust washers	29
3.19 B1616 (1x1-1/4x1 inch) needle roller bearing	30
3.20 HK1612 (16x22x12mm) needle roller bearing	30
3.21 LM3UU 3mm. linear rod bearing	30
3.22 688Z (8x16x5mm) groove ball bearing	30
3.23 30203 (17x40x13.25mm) taper roller bearing	31
3.24 3mm. x 85mm. shaft	31
3.25 2mm. x 2mm. keystick (length 50mm.)	31
3.26 8mm. keyed rotary shaft (length 50mm.)	32
3.27 214-L Timing belt	33
3.28 150XL037 belt	34
3.29 180XL037 belt	34

สารบัญรูปภาพ (ต่อ)

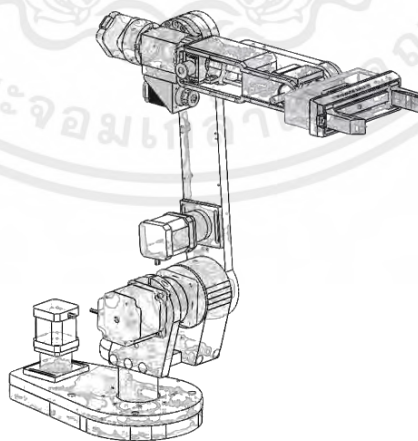
รูปภาพที่	หน้า
3.30 84XL037 belt	34
3.31 60T XL pulley	35
3.32 L10 Pulley	35
3.33 XL 15 tooth 8mm. Bore	35
3.34 XL 10 tooth 6mm. bore pulley	35
3.35 J1 motor SKU: NEMA17	36
3.36 J2 motor SKU: NEMA23	36
3.37 J3 motor SKU: NEMA17	36
3.38 J4 motor SKU: NEMA11	37
3.39 J5 motor : NEMA17 with lead screw	37
3.40 J6 motor SKU: NEMA17	37
3.41 STM 32 – STM32F103C8T6	38
3.42 ESP 32	38
3.43 กลไกของแขนกล	39
3.44 การ Wiring สายไฟ	40
3.45 ประกอบบอร์ด STM32 เข้ากับบอร์ด ESP32	40
4.1 การออกแบบแขนกลด้วยโปรแกรม SOLIDWORKS	42
4.2 แขนกลที่เสร็จสมบูรณ์	43
4.3 โปรแกรมการควบคุมแขนกลโดยใช้ ROS Moveit	44
4.4 ควบคุมและจำลองการเคลื่อนไหวของแขนกลผ่านโปรแกรม Rviz	45
4.5 บอร์ด STM32 และ ESP32 ที่ต่อกันสมบูรณ์	46
4.6 โปรแกรม Image Detection ตรวจจับน้องไก่	47

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโรงงาน

เนื่องด้วยในปัจจุบัน โรงงานอุตสาหกรรมต่างๆ มีการนำเทคโนโลยีใหม่ๆ เข้ามาใช้ในการกระบวนการผลิตสินค้าต่างๆ ทำให้มีการลดทรัพยากรมนุษย์ หรือแรงงานมนุษย์ลง เนื่องจากว่าการใช้แรงงานมนุษย์นั้นมีความแปรปรวนในหลายๆเรื่องไม่ว่าจะเป็น เจ็บป่วย หรือลาออก ส่งผลให้การทำงานเกิดความล่าช้า หรือสูญเสียทรัพยากรบางส่วนโดยไม่จำเป็น เช่น ค่าใช้จ่ายในด้านแรงงานทำให้ปัจจุบันมีการใช้เครื่องจักรในโรงงานอุตสาหกรรมมากขึ้นเรื่อยๆ แทนการใช้มนุษย์ เราจึงคิดตัวแขนกลที่สามารถหยิบวัตถุที่เราใช้ AI ในการประมวลผลขึ้นมาได้ สามารถนำไปประยุกต์ใช้กับการนำแขนกลแทนมนุษย์ในการจัดเรียงลายผลิตสินค้าภายในโรงงานอุตสาหกรรมได้ ทำให้ลายการผลิตมีความเสถียรขึ้นและสามารถคำนวณจำนวนสินค้าได้แม่นยำยิ่งขึ้นอีกด้วย แต่การใช้งานก็ยังมีข้อจำกัดอยู่บ้างในบางส่วน เช่น หากวัตถุที่มาตามสายพานกึ่งของแขนกล AI ไม่สามารถจับได้ว่าเป็นวัตถุที่เราต้องการทำให้เกิดข้อผิดพลาดได้ ฉะนั้นทางผู้จัดทำจึงได้เทรนดี AI ซ้ำๆ และใช้สูตรคำนวณต่างๆนำมาประยุกต์ใช้ในการวิเคราะห์ และประมวลผลทำให้เกิดความผิดพลาดน้อยที่สุด และทำให้มีประสิทธิภาพในการทำงานสูงที่สุด



รูปที่ 1.1 แบบจำลองแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อศึกษากระบวนการทำงานของแขนกล และใช้กล้องจับภาพคัดแยกวัตถุ
- 1.2.2 เพื่อศึกษาการเขียนคำสั่งการทำงานควบคุมแขนกลด้วยระบบปฏิบัติการแขนกล หรือที่เราเรียกว่า Robot Operating System (ROS)
- 1.2.3 เพื่อเป็นโครงการต้นแบบและสามารถนำไปพัฒนาต่อยอด และขยายผลสำหรับเป็นอุปกรณ์ที่สามารถนำมาใช้ได้จริงๆ ในโรงงานอุตสาหกรรม
- 1.2.4 เพื่อเป็นการฝึกฝนให้นักศึกษาในภาควิชาแมคคาทรอนิกส์และอัตโนมัติชั้น ได้รู้จักกันคว่าหาความรู้และฝึกฝนทักษะในการทำงานร่วมกันเป็นทีม อีกทั้งยังเพิ่มประสบการณ์ในการทำงานร่วมกับองค์กรต่างๆ ที่จะต้องทำงานร่วมกับผู้อื่นได้อีกด้วย
- 1.2.5 เพื่อนำองค์ความรู้ที่ได้ไปประยุกต์ใช้กับงานด้านอื่นๆ ที่เกี่ยวข้อง

1.3 ขอบเขตของโครงการ

- 1.3.1 สามารถควบคุมการทำงานของแขนกลจากระบบปฏิบัติการแขนกลได้
- 1.3.2 สามารถควบคุมการทำงานของมอเตอร์ในแต่ละตัวที่ติดตั้งบนแขนกลได้
- 1.3.3 สามารถเขียนโปรแกรมในการควบคุมแขนกลได้
- 1.3.4 สามารถเทรน AI จนสามารถตรวจจับวัตถุได้อย่างแม่นยำ

1.4 วิธีการดำเนินงาน

- 1.4.1 กำหนดหัวข้อและขอบเขตการจัดทำโครงการ
- 1.4.2 ศึกษาทฤษฎีเบื้องต้น
- 1.4.3 ศึกษาการทำงานของแขนกลจากต้นแบบที่เรานำมาใช้
- 1.4.4 ออกแบบแขนกลขึ้นมาใหม่โดยอ้างอิงต้นแบบจากแขนกล AR4
- 1.4.5 เขียนโปรแกรมการทำงานของชิ้นงานและตรวจสอบความถูกต้อง
- 1.4.6 ทดสอบการทำงานของแขนกลที่ออกแบบและตรวจสอบอีกครั้ง
- 1.4.7 จัดซื้ออุปกรณ์และสร้างชิ้นงานตามที่ออกแบบและวางแผนไว้
- 1.4.8 ปรับปรุงประสิทธิภาพและแก้ไขข้อผิดพลาดของอุปกรณ์
- 1.4.9 สรุปผลและจัดทำรายงาน

บทที่ 2

ทฤษฎีและความรู้ที่เกี่ยวข้อง

สำหรับบทนี้จะกล่าวเกี่ยวกับทฤษฎีและความรู้ต่าง ๆ ที่มีความสำคัญในการออกแบบแขนกล และซอฟต์แวร์ที่ใช้ในการควบคุมแขนกล ซึ่งมีองค์ประกอบดังต่อไปนี้

- 1.) แขนกล
- 2.) ROS (Robot Operating System)
- 3.) Library ที่ใช้ภายใน ROS
- 4.) ไมโครคอนโทรลเลอร์
- 5.) การรู้จำวัตถุ (Object Detection)
- 6.) โพรโตคอล SPI (Serial Peripheral Interface)
- 7.) สเต็ปเปอร์มอเตอร์
- 8.) สเต็ปเปอร์มอเตอร์ไร้เฟือง
- 9.) ซอฟต์แวร์ที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 แขนกล

แขนกลใช้ต้นแบบจากแขนกล AR4 ซึ่งเป็นโมเดล Open Source โดยนำมาปรับปรุงให้เข้า กับงบประมาณและอุปกรณ์ที่มี แขนกล AR4 เป็นแขนกลที่มี 6 degrees of freedom หมายความว่า สามารถตั้ง Input มอเตอร์ได้ทั้ง 6 มอเตอร์โดยอิสระซึ่งกันและกัน ซึ่งทำให้แขนกลมีความอิสระใน การเคลื่อนที่มากขึ้น



รูปที่ 2.1 โมเดลแขนกล AR4

2.2 ROS (Robot Operating System)

2.2.1 ความหมายของ ROS (Robot Operating System)

ROS หรือ Robot Operating System เป็นหนึ่งในระบบซอฟต์แวร์ที่ใช้ในการ พัฒนาและควบคุมหุ่นยนต์และระบบหุ่นยนต์ต่าง ๆ อย่างมีประสิทธิภาพและยืดหยุ่นที่สุด ROS สามารถสร้างและควบคุมหุ่นยนต์ที่มีความซับซ้อนต่าง ๆ ได้ง่ายขึ้น โดย ROS มีชุด ของเครื่องมือและไลบรารีที่มีอยู่ให้ใช้งานอย่างมากมายสำหรับการแสดงผลข้อมูล ควบคุม การเคลื่อนไหว การนำทาง การสื่อสารระหว่างหุ่นยนต์ และการแบ่งปันข้อมูลระหว่าง หน่วยประมวลผลของหุ่นยนต์ นอกจากนี้ ROS ยังรองรับการพัฒนาและการใช้งานใน ภาษา C++ และ Python

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 เวอร์ชันของ ROS (Robot Operating System)

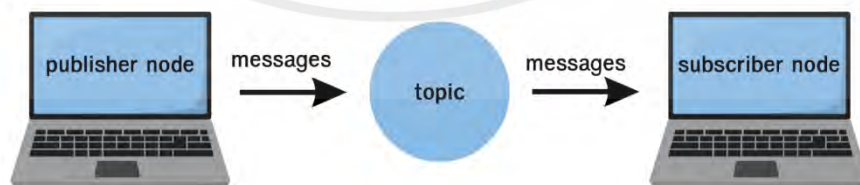
ROS ในปัจจุบันมีด้วยกันหลายเวอร์ชัน และทุกเวอร์ชันในปัจจุบันนั้นทำงานบน Unix-based platforms ปัจจุบัน ROS 2 เป็นเวอร์ชันที่ใหม่ที่สุด ซึ่งสำหรับโปรเจกต์นี้จะเลือกเป็น ROS 1 noetic ซึ่งเป็นเวอร์ชันสุดท้ายของ ROS 1 ซึ่งจะใช้ได้กับ ubuntu 20.04 และยังคงมีความนิยมอยู่



รูปที่ 2.2 LOGO ของ ROS 1 noetic

2.2.3 การทำงานของ ROS

ROS ใช้การสื่อสารแบบ master/slave ซึ่งจะมี ROS master คอยจัดการ node ต่าง ๆ ในระบบ ROS ให้สามารถสื่อสารกันได้ โดยแต่ละ node จะแยกเป็น publisher และ subscriber โดย publisher จะ publish หรือส่ง messages หรือข้อมูลไปยัง topic และ subscriber ที่สนใจจะ subscribe หรือรับข้อมูลนั้นมาจาก topic อีกที



รูปที่ 2.3 การส่ง message จาก publisher node ไปยัง subscriber node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Library ที่ใช้ภายใน ROS

ไลบรารีสำเร็จรูปของ ROS ที่ใช้จะแบ่งออกเป็น 3 ส่วนคือ Moveit, Rviz และ roserial ซึ่งสามารถดาวน์โหลดและติดตั้งได้ทันทีหากมีการติดตั้ง ROS ไว้อยู่แล้ว

2.3.1 Moveit

Moveit คือ framework แบบ motion planning ซึ่งใช้ในการควบคุมแขนกล โดยสามารถสั่งให้แขนกลขยับที่ละ joint หรือจะสั่งให้แขนกลไปยังตำแหน่งใด ๆ ในพิกัดฉาก 3 มิติ $x y z$ ได้ โดยระยะจะไม่เกินระยะยืดสุดของแขนกล ซึ่งสามารถนำไปประยุกต์ใช้ในระบบอุตสาหกรรมที่ใช้การควบคุมแขนกลได้



รูปที่ 2.4 โลโก้ของ Moveit

2.3.2 Rviz

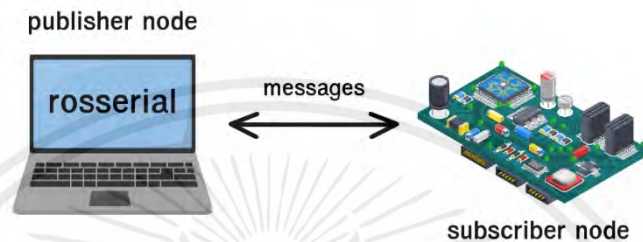
Rviz คือ เครื่องมือที่ช่วยในการจำลองและแสดงผลในรูปแบบ 3 มิติที่เชื่อมต่อได้ทั้งหุ่นยนต์ในโลกจริงและหุ่นยนต์ในโลกเสมือน



รูปที่ 2.5 โลโก้ของ Rviz

2.3.3 roserial

roserial คือ ทำหน้าที่เป็นตัวกลางที่ทำหน้าที่จัดการให้บอร์ดไมโครคอนโทรลเลอร์สามารถที่จะทำการส่งข้อมูลและสื่อสารกับระบบ ROS ที่อยู่บนเครื่องคอมพิวเตอร์ได้



รูปที่ 2.6 การส่งข้อมูลของ roserial

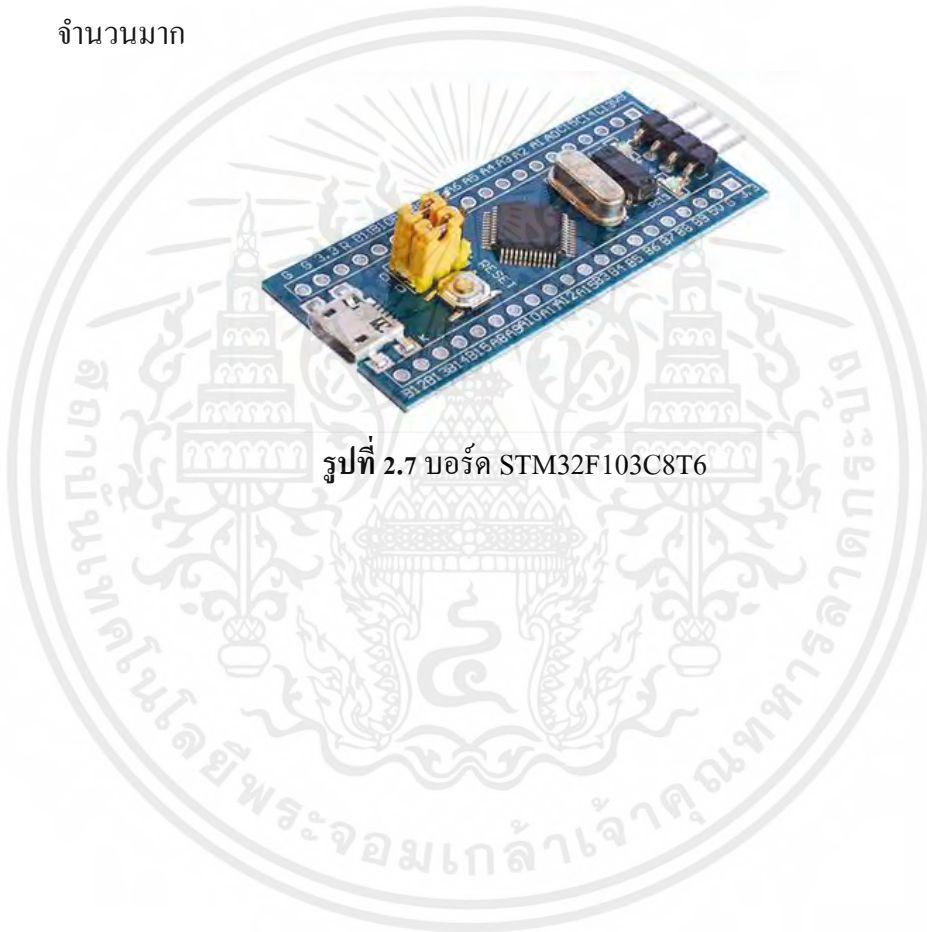
2.4 ไมโครคอนโทรลเลอร์

2.4.1 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller หรือ MCU) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน ซึ่งจะเลือกใช้บอร์ด STM32F103C8T6 และบอร์ด ESP32

2.4.2 STM32

STM32F103C8T6 เป็นบอร์ดที่ใช้ MCU เบอร์ STM32F103RCBT6 ARM Cortex-M3 processor ซึ่งรองรับการทำงานแบบ 32 Bit ซึ่งเร็วกว่า ไมโครคอนโทรลเลอร์ เบอร์ Atmega328P ที่มีอยู่ใน Arduino Uno โดยบอร์ด STM32F103C8T6 มีความเร็ว Clock 72 Mhz มีแรม 20K และหน่วยความจำ Flash 128K รองรับแรงดัน VDC 3.3V และรองรับการทำงานกับ Arduino IDE เหมาะกับงานประมวลผลที่ต้องการใช้ความเร็วสูงๆ และแรมจำนวนมาก



รูปที่ 2.7 บอร์ด STM32F103C8T6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ESP 32

ESP32 เป็น ไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi และ Bluetooth Low Energy (BLE) ในตัว ใช้ภาษา C หรือ Python ในการพัฒนาโปรแกรม ESP32 มาพร้อมกับ touch-sensitive pins ที่สามารถใช้ปลุก ESP32 จากโหมด deep sleep และมี hall effect sensor และ temperature sensor ในตัว

ESP32 ทำงานแบบ Dual Core ที่ความเร็ว 160Mhz มี SRAM 512K หน่วยความจำ Flash 16M มีขา GPIO 36 ขา ความละเอียดในการอ่านค่า ADC 12Bit สามารถเขียนโปรแกรมผ่าน Arduino IDE เหมือนเขียน Arduino ได้

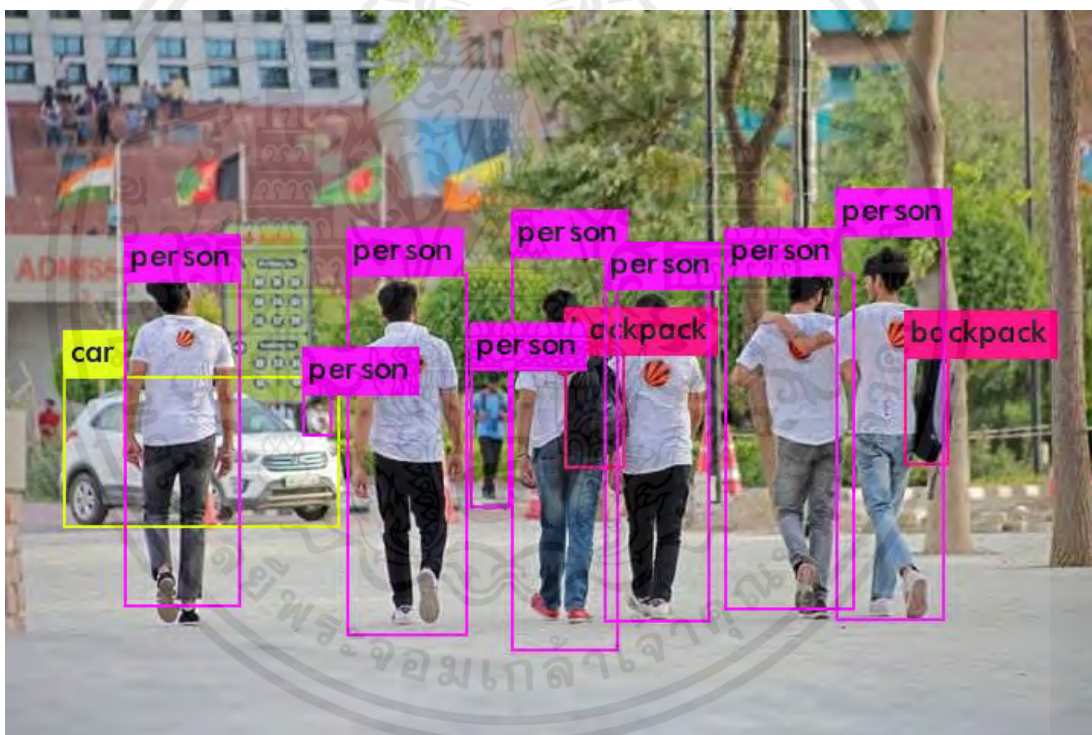


รูปที่ 2.8 บอร์ด ESP32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การรู้จำวัตถุ (Object Detection)

การรู้จำวัตถุ คือ AI ที่ใช้ในงาน Computer Vision ที่สามารถจดจำจำแนก และตรวจจับวัตถุใด ๆ จากในภาพได้ ในการตรวจจับวัตถุสามารถทำได้หลายวิธี วิธีที่นิยมที่สุดคือการวาดกล่องรอบวัตถุ (Bounding Box) โมเดลอัลกอริทึมที่ใช้ในการตรวจจับวัตถุในภาพมีหลากหลายโมเดล ซึ่งในที่นี้จะใช้เป็น โมเดลอัลกอริทึม YOLO หรือ You Only Look Once พัฒนาโดย ultralytics ซึ่งทำงานโดยการนำแบ่งภาพออกเป็น $N*N$ ช่อง แล้วคำนวณความน่าจะเป็นของช่องนั้น ๆ ว่ามีค่าความเป็นไปได้ว่าจะเป็นวัตถุที่อยู่ใน Dataset หรือไม่



รูปที่ 2.9 Image Detection โดย YOLO

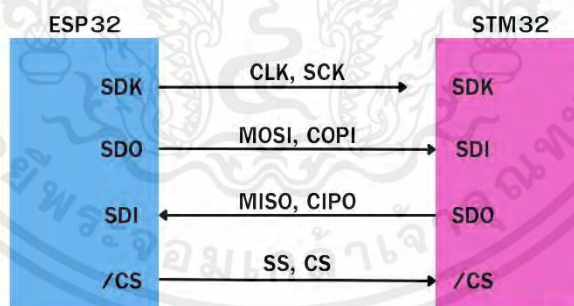
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 โปรโตคอล SPI (Serial Peripheral Interface)

SPI หรือ Serial Peripheral Interface คือ รูปของแบบการสื่อสารข้อมูลแบบอนุกรมแบบซิงโครนัสรูปแบบหนึ่งการสื่อสารอนุกรมแบบ SPI จะอาศัยสัญญาณนาฬิกาเป็นตัวกำหนดจังหวะการรับส่งข้อมูล สามารถส่งข้อมูลไปยังปลายทางและรับข้อมูลจากปลายทางกลับมาในครั้งเดียวกัน (Full Duplex) การสื่อสารอนุกรมแบบ SPI จะแบ่งอุปกรณ์ออกเป็น 2 ฝั่ง คือ Master เป็นตัวควบคุมการรับส่งข้อมูล และ Slave เป็นอุปกรณ์ที่รอรับคำสั่งจาก Master ในบัสการสื่อสารแบบอนุกรมแบบ SPI สามารถมี Slave มีได้มากกว่า 1 ตัว

การสื่อสารอนุกรมแบบ SPI ใช้สายสัญญาณทั้งหมดจำนวน 4 เส้น มีรายละเอียดดังนี้

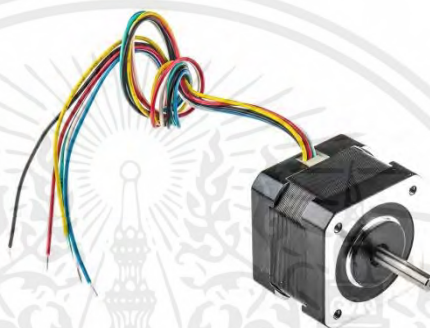
- SCK (Clock Data) : ใช้สำหรับส่งสัญญาณนาฬิกาจาก Master ไปยัง Slave
- MISO (Master In Slave Out) : ใช้สำหรับรับข้อมูลจาก Slave
- MOSI (Master Out Slave In) : ใช้สำหรับส่งข้อมูลจาก Master ไปยัง Slave
- SS/CS (Slave Select/Chip Select) : ใช้สำหรับเลือก Slave ที่ต้องการใช้งาน



รูปที่ 2.10 การทำงานของ SPI

2.7 สเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์ คือ มอเตอร์ไฟฟ้าที่ขับเคลื่อนด้วยคลื่นสี่เหลี่ยม (Pulse) โดยโครงสร้างภายในนั้นจะประกอบไปด้วยขั้วแม่เหล็กบนสเตเตอร์ (Stator) ที่ทำมาจากแผ่นเหล็กวงแหวน จะมีซี่ยื่นออกมาประกอบกันเป็นชั้น ๆ โดยแต่ละซี่ที่ยื่นออกมานั้นจะมีขดลวดพันอยู่ เมื่อมีกระแสผ่านขดลวดจะเกิดสนามแม่เหล็กไฟฟ้าขึ้น



รูปที่ 2.11 สเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์ทำงานโดยแบ่งการหมุนทั้งหมดเป็นชุดของสเต็ปย่อยๆ โดยแต่ละสเต็ปจะถูกขับเคลื่อนโดยการจ่ายสัญญาณไฟฟ้าให้กับขดลวดมอเตอร์ตามลำดับที่กำหนด โดยแกนของสเต็ปเปอร์มอเตอร์จะหมุน 1 สเต็ปต่อสัญญาณไฟฟ้า 1 ลูกคลื่น ข้อดีของสเต็ปเปอร์มอเตอร์คือความแม่นยำสูง ต้นทุนต่ำ และใช้งานง่าย อย่างไรก็ตาม สเต็ปเปอร์มอเตอร์ไม่ได้ให้แรงบิดสูงและความเร็วของสเต็ปเปอร์จะไม่ได้สูงมาก

2.8 สเต็ปเปอร์มอเตอร์ไครฟ์เวอร์

สเต็ปเปอร์มอเตอร์ไครฟ์เวอร์ คือ อุปกรณ์ที่ใช้ขับสเต็ปเปอร์มอเตอร์ ให้สามารถขับไปยังตำแหน่งต่าง ๆ ได้ โดยจะอาศัยสัญญาณไฟฟ้าแบบคลื่นสี่เหลี่ยม (Pulse) ในการขับเคลื่อนมอเตอร์ เพื่อให้เกนเกิดการหมุนไปที่ละขั้น โดยสามารถหมุนรอบแกนได้ถึง 360 องศา



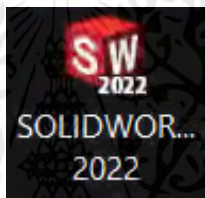
รูปที่ 2.12 สเต็ปเปอร์มอเตอร์ไครฟ์เวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 ซอฟต์แวร์ที่เกี่ยวข้อง

2.9.1 SOLIDWORKS 2022

โปรแกรมสำหรับออกแบบงานด้านวิศวกรรม เช่น ชิ้นส่วน อุปกรณ์ เครื่องจักร เป็นต้น โดยเป็นโปรแกรมในตระกูล CAD (Computer Aided Design and Drafting) ซึ่งสามารถจำลองเป็นโมเดลสามมิติได้ และมีความละเอียดกว่า CAD ทั่วไปในการวิเคราะห์ ส่วนต่าง ๆ เช่น ความแข็งแรง อุณหภูมิ อายุการใช้งาน เป็นต้น SOLIDWORKS ได้เสนอฟังก์ชันทางการออกแบบที่มีประสิทธิภาพ ใช้งานง่ายด้วย User Interface ที่ใช้ไอคอนในการทำงาน รวมถึงการใช้คีย์ลัดเข้ามาช่วย เพื่อเพิ่มความรวดเร็วในกระบวนการออกแบบและทำให้ออกแบบงานได้สะดวกยิ่งขึ้น



รูปที่ 2.13 โปรแกรม SOLIDWORKS 2022

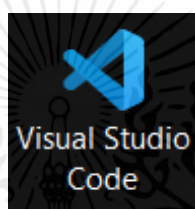


รูปที่ 2.14 หน้าต่างการใช้งานโปรแกรม SOLIDWORKS 2022

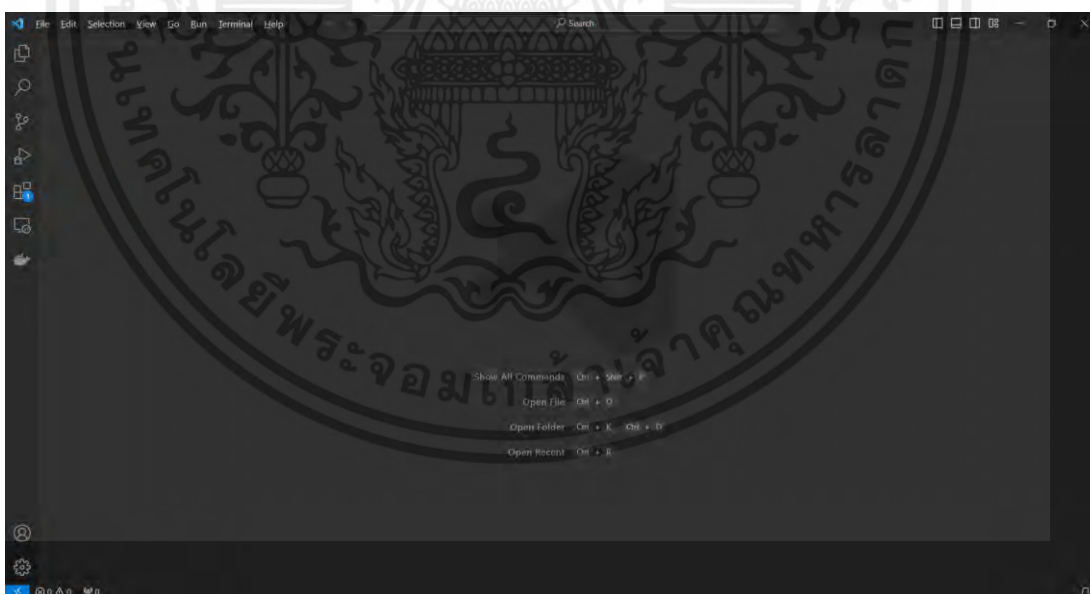
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 Visual Studio Code

Visual Studio Code หรือ VS Code เป็นโปรแกรมประเภท Text Editor ใช้ในการแก้ไขโค้ดที่มีขนาดเล็ก แต่มีประสิทธิภาพสูง เป็น Open Source จึงสามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย เหมาะสำหรับนักพัฒนาโปรแกรมที่ต้องการใช้งานหลายแพลตฟอร์มรองรับการใช้งานทั้งบน Windows, macOS และ Linux ใช้งานได้หลายภาษาและสามารถเชื่อมต่อกับ Git ได้ง่าย สามารถนำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือและส่วนขยายต่าง ๆ ให้เลือกใช้มากมาย สามารถปรับเปลี่ยน Themes ได้หลากหลาย



รูปที่ 2.15 โปรแกรม Visual Studio Code



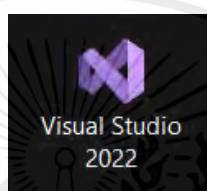
รูปที่ 2.16 หน้าต่างการใช้งานโปรแกรม Visual Studio Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

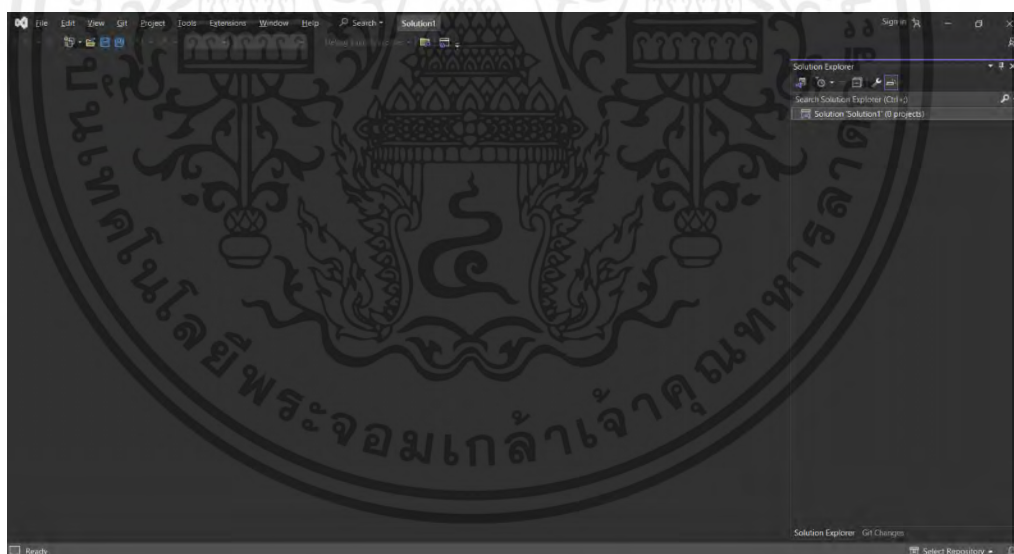
2.9.3 Visual Studio 2022

Visual Studio คือ โปรแกรมที่เป็นเครื่องมือในการพัฒนาหรือสร้างโปรแกรม ระบบต่าง ๆ บนระบบปฏิบัติการ Window หรือ Mobile

Visual Studio รองรับโปรเจกต์ที่สร้างด้วยหลาย ๆ ภาษา เช่น C, C++ , C#, Python และ GO โดยมีส่วนเสริมหรือ Extension ที่สามารถดาวน์โหลดเพิ่มเติมภายหลังได้



รูปที่ 2.17 โปรแกรม Visual Studio 2022



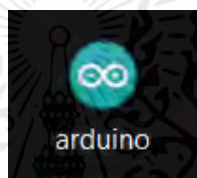
รูปที่ 2.18 หน้าต่างการใช้งาน โปรแกรม Visual Studio 2022

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.4 Arduino IDE

Arduino IDE คือ โปรแกรม Open Source ทำหน้าที่ ติดต่อระหว่าง คอมพิวเตอร์ ไม่ว่าจะ เป็นระบบปฏิบัติการ Windows, Mac OS X หรือ Linux กับบอร์ด Arduino ซึ่ง โปรแกรมนี้ถูกออกแบบมาให้ช่วยต่อการเขียนโค้ดและ อัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino

นอกจากบอร์ด Arduino แล้ว บอร์ดไมโครคอนโทรลเลอร์ตัวอื่น ๆ ก็ยังสามารถพัฒนาและอัปโหลดโปรแกรมที่เราเขียนลงบอร์ดไมโครคอนโทรลเลอร์ นั้น ๆ โดยใช้โปรแกรม Arduino IDE ได้อีกด้วย



รูปที่ 2.19 หน้าต่างการใช้งานโปรแกรม Arduino IDE



รูปที่ 2.20 หน้าต่างการใช้งานโปรแกรม Arduino IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

วิธีดำเนินงาน

โปรเจกต์หุ่นยนต์เอไอจับน้องไก่ มีวัตถุประสงค์เพื่อทำการศึกษาและออกแบบสร้างแขนกลต้นแบบที่ใช้ในการระบุและจับน้อง สำหรับบทนี้จะกล่าวถึงวัสดุ อุปกรณ์และขั้นตอนการดำเนินงาน ดังต่อไปนี้

3.1 กำหนดหัวข้อและขอบเขตการจัดทำโครงการ

ก่อนที่จะทำการเริ่มงานได้นั้น ต้องมีการกำหนดหัวข้อที่จะทำ เพื่อที่จะได้ทราบถึงจุดมุ่งหมายและขอบเขตของงาน จากนั้นจึงมาประชุมวางแผนงาน เพื่อให้สามารถทำงานจนสำเร็จลุล่วงได้ตามเป้าหมายที่กำหนด

3.2 ศึกษาทฤษฎีเบื้องต้น

ทฤษฎีและความรู้ต่าง ๆ ที่มีความสำคัญในการออกแบบแขนกล และซอฟต์แวร์ที่ใช้ในการควบคุมแขนกล ซึ่งมีองค์ประกอบดังต่อไปนี้

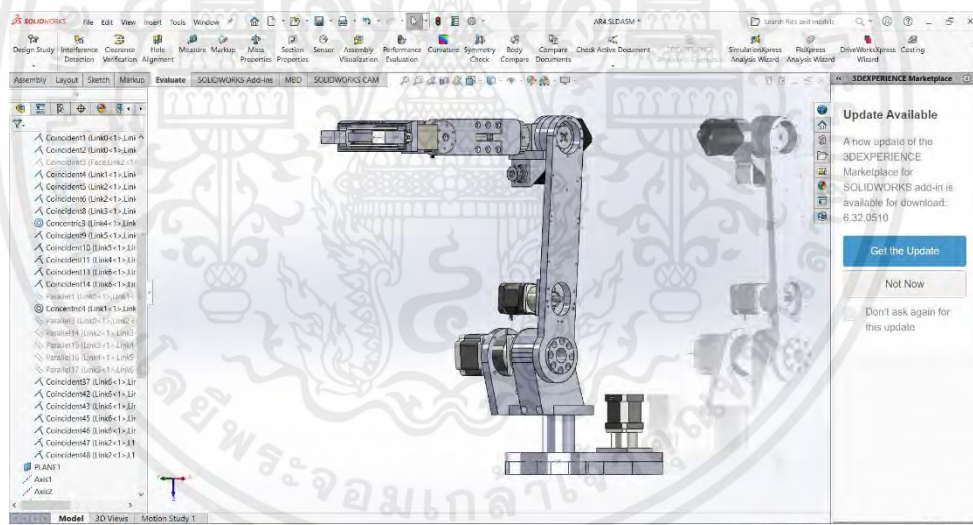
- 1.) แขนกล
- 2.) ROS (Robot Operating System)
- 3.) Library ที่ใช้ร่วมกับ ROS
- 4.) ไมโครคอนโทรลเลอร์
- 5.) การรู้จำวัตถุ (Object Detection)
- 6.) โปรโตคอล SPI (Serial Peripheral Interface)
- 7.) สเต็ปเปอร์มอเตอร์
- 8.) สเต็ปเปอร์มอเตอร์ไครฟ์เวอร์
- 9.) ซอฟต์แวร์ที่เกี่ยวข้อง

3.3 ศึกษาการทำงานของแขนกลจากต้นแบบที่เรานำมาใช้

จากการศึกษาการทำงานของแขนกล แขนกลต้นแบบที่เรานำมาใช้ การเคลื่อนที่ของแขนกลจะเคลื่อนที่ในส่วนต่างๆด้วยมอเตอร์จำนวน 6 จุดเพื่อหมุนสายพานในส่วนต่างๆของแขนกล หรืออาจจะใช้เป็นโซ่แทนได้เพื่อที่จะได้มีความแข็งแรงและทนทานมากยิ่งขึ้น แต่ชิ้นงานที่เราทำนั้นจะใช้เป็นแค่สายพานธรรมดา เพื่อลดค่าใช้จ่ายในส่วนนี้

3.4 ออกแบบแขนกลขึ้นมาใหม่โดยอ้างอิงต้นแบบจากแขนกล AR4

การออกแบบแขนกล ใช้ต้นแบบของ AR4 ซึ่งเป็น โมเดล Open Source ของแขนกล 6 degrees of freedom ในการออกแบบจะออกแบบในที่ละส่วน โดยจะมีแบ่งเป็น อะลูมิเนียมและพลาสติก ส่วนที่เป็นอะลูมิเนียมจะเป็นส่วนที่ต้องรับโหลดหนัก และส่วนที่เป็นพลาสติกเป็นส่วนที่ไม่ต้องรับโหลดหนักเพื่อลดค่าใช้จ่ายในส่วนนี้



รูปที่ 3.1 ตัวอย่างการออกแบบแขนกลโดยโปรแกรม SOLIDWORKS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 เก็บ Dataset รูปภาพสำหรับการเทรน Image Detection

ทำการปรี้น 3 มิติของไถ่จำลองและถ่ายภาพในหลาย ๆ มุม เพื่อเก็บเป็น Dataset สำหรับการเทรน Image Detection ซึ่งต้องการ Dataset จำนวนหนึ่งเพื่อให้ AI มีความแม่นยำมากยิ่งขึ้น

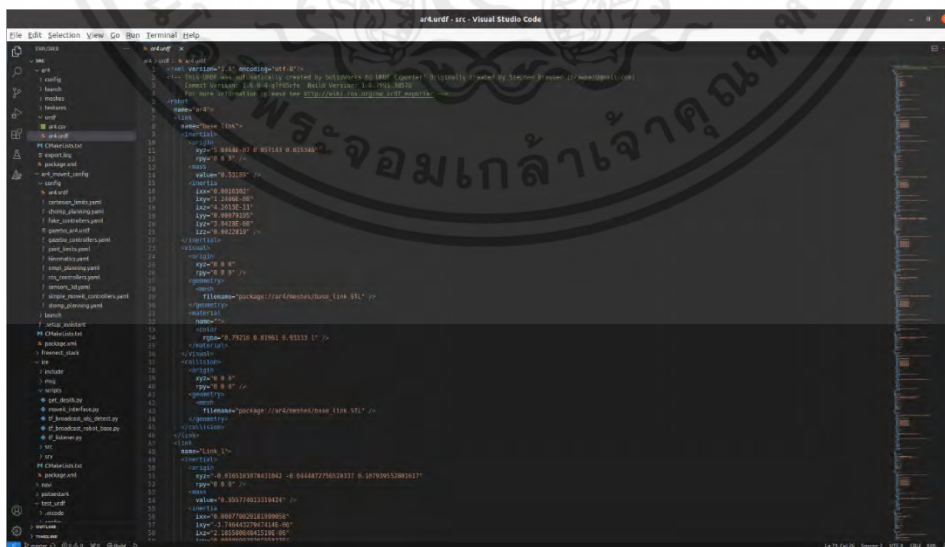


รูปที่ 3.2 ตัวอย่างร่องไถ่ปรี้น 3 มิติ

3.6 เขียนโปรแกรมการทำงานของแขนกล

3.6.1 export ไฟล์โมเดลแขนกลจาก SOLIDWORKS

เมื่อออกแบบแขนกลเสร็จสิ้นในโปรแกรม SOLIDWORKS แล้วทำการ export ไฟล์ออกมาเป็นไฟล์ URDF เพื่อนำไปใช้กับ Moveit

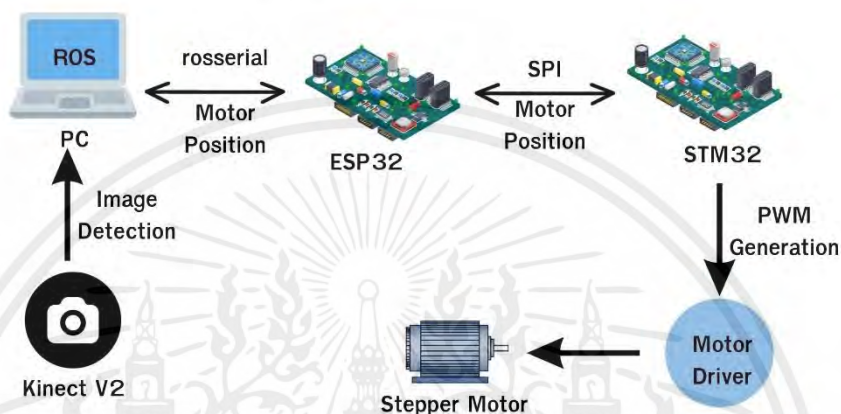


รูปที่ 3.3 ตัวอย่างไฟล์ URDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 ออกแบบ Workflow ของโปรแกรม

ก่อนเริ่มการเขียน โปรแกรมจริง จำเป็นต้องมีการออกแบบ Workflow ของโปรแกรมที่จะทำงาน เพื่อสะดวกให้การพัฒนาและแก้ไข โปรแกรม และเพื่อให้การทำงานเป็นระบบมากขึ้น



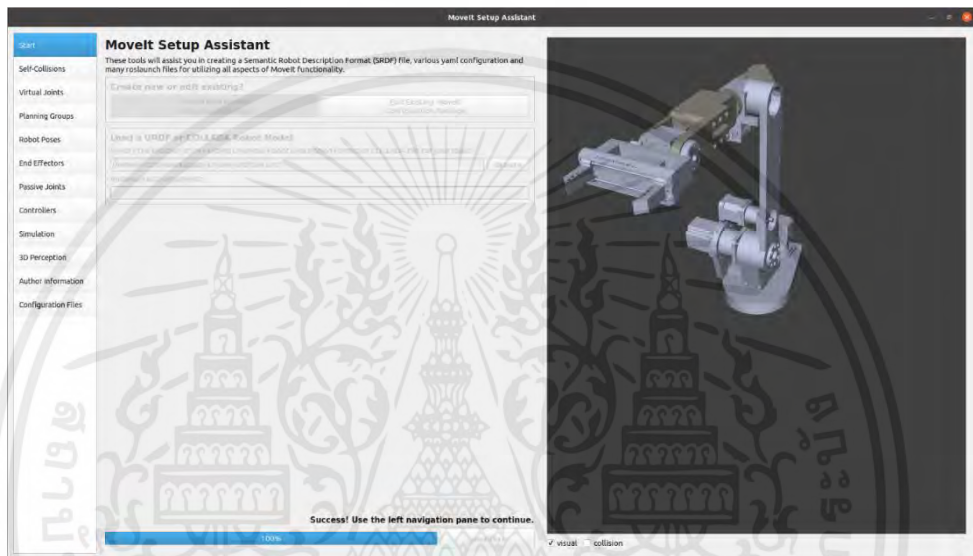
รูปที่ 3.4 Workflow ของโปรแกรม

3.6.3 ใช้ Moveit Assistant Setup เพื่อตั้งค่าต่าง ๆ ของแขนกล

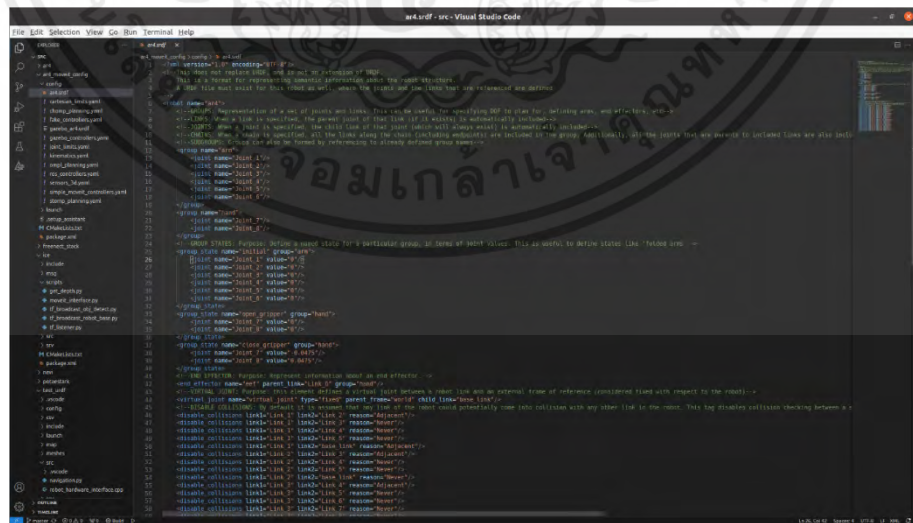
สำหรับการเขียน โปรแกรมเพื่อควบคุมแขนกลด้วยการใช้ ROS ในการควบคุม จะใช้ ROS 1 Noetic บนระบบปฏิบัติการ Ubuntu 20.04 ซึ่งส่วนของโค้ดจะเขียนบนคอมพิวเตอร์และใช้โปรแกรม Visual Studio Code ในการออกแบบโค้ด

การควบคุมแขนกล จะเป็นไปโดยไลบรารีของ ROS ชื่อว่า Moveit ซึ่งช่วยในการคำนวณ Inversed Kinematic เพื่อหาค่าองศา ความเร่ง และความเร็วมอเตอร์ต่าง ๆ ซึ่งจะนำไปป้อนให้มอเตอร์อีกทีเพื่อควบคุมตำแหน่งของมอเตอร์

ขั้นแรกใช้ Moveit Assistant Setup โดยการนำไฟล์ URDF ของแขนกลที่ได้มาจากโปรแกรม SOLIDWORKS เพื่อตั้งค่าต่าง ๆ สำหรับการใช้งานกับ Moveit โดย Moveit Assistant Setup จะสร้างไฟล์ Semantic Robot Description Format (SRDF) ซึ่งเป็นไฟล์ที่บอกคุณลักษณะของแขนกลนั้น ๆ ทั้ง แกนหมุนของมอเตอร์ ค่าความเฉื่อยของแขนกล และอื่น ๆ



รูปที่ 3.5 หน้าต่างโปรแกรม Moveit Setup Assistant ที่โหลดไฟล์ URDF เข้ามาแล้ว

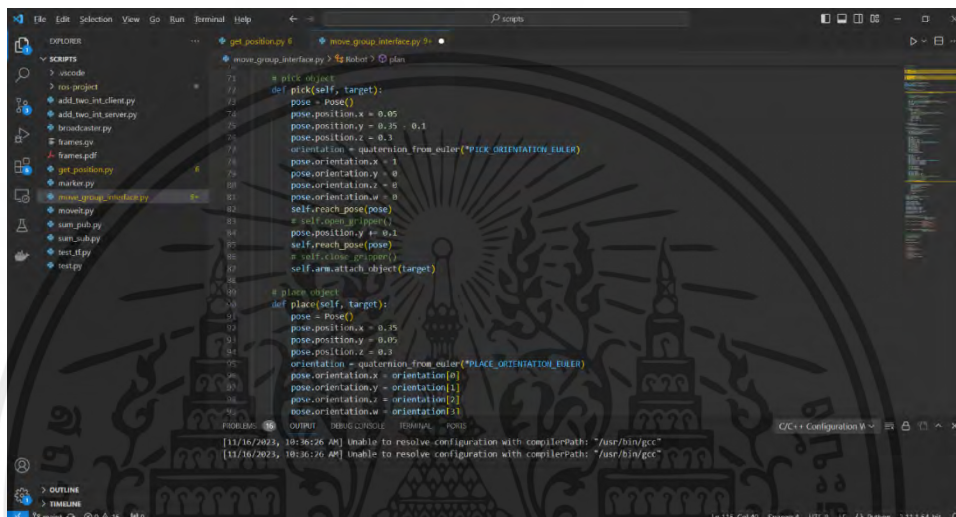


รูปที่ 3.6 ตัวอย่างไฟล์ SRDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.4 การเขียน โปรแกรม Interface ควบคุมแขนกล

การเขียน โปรแกรม เริ่มเขียน โปรแกรม interface ด้วยภาษา Python เพื่อใช้ในการเชื่อมต่อเข้ากับ Moveit เพื่อคำนวณ Inversed Kinematic จากตำแหน่งในพิกัดฉาก 3 มิติ x y z ที่ป้อนเข้าไป และไฟล์ SRDF ที่ได้จาก Moveit Assistant Setup เมื่อได้ค่าองศา ความเร่ง และความเร็วมอเตอร์ จึงนำไปสั่งให้มอเตอร์ไปยังตำแหน่งที่ต้องการ



```

71 # pick object
72 def pick(self, target):
73     pose = Pose()
74     pose.position.x = 0.05
75     pose.position.y = 0.25 - 0.1
76     pose.position.z = 0.1
77     orientation = quaternion.from_euler("PICK_ORIENTATION_EULER")
78     pose.orientation.x = 0
79     pose.orientation.y = 0
80     pose.orientation.z = 0
81     pose.orientation.w = 0
82     self.reach_pose(pose)
83     # self.pick_group(target)
84     pose.position.y += 0.1
85     self.reach_pose(pose)
86     # self.pick_group(target)
87     self.attach_object(target)
88
89 # place object
90 def place(self, target):
91     pose = Pose()
92     pose.position.x = 0.25
93     pose.position.y = 0.05
94     pose.position.z = 0.2
95     orientation = quaternion.from_euler("PLACE_ORIENTATION_EULER")
96     pose.orientation.x = orientation[0]
97     pose.orientation.y = orientation[1]
98     pose.orientation.z = orientation[2]
99     pose.orientation.w = orientation[3]
100
101 # main
102 if __name__ == '__main__':
103     # pick object
104     # place object
105     # test
106     # test
107     # test
108     # test
109     # test
110     # test
111     # test
112     # test
113     # test
114     # test
115     # test
116     # test
117     # test
118     # test
119     # test
120     # test
121     # test
122     # test
123     # test
124     # test
125     # test
126     # test
127     # test
128     # test
129     # test
130     # test
131     # test
132     # test
133     # test
134     # test
135     # test
136     # test
137     # test
138     # test
139     # test
140     # test
141     # test
142     # test
143     # test
144     # test
145     # test
146     # test
147     # test
148     # test
149     # test
150     # test
151     # test
152     # test
153     # test
154     # test
155     # test
156     # test
157     # test
158     # test
159     # test
160     # test
161     # test
162     # test
163     # test
164     # test
165     # test
166     # test
167     # test
168     # test
169     # test
170     # test
171     # test
172     # test
173     # test
174     # test
175     # test
176     # test
177     # test
178     # test
179     # test
180     # test
181     # test
182     # test
183     # test
184     # test
185     # test
186     # test
187     # test
188     # test
189     # test
190     # test
191     # test
192     # test
193     # test
194     # test
195     # test
196     # test
197     # test
198     # test
199     # test
200     # test
201     # test
202     # test
203     # test
204     # test
205     # test
206     # test
207     # test
208     # test
209     # test
210     # test
211     # test
212     # test
213     # test
214     # test
215     # test
216     # test
217     # test
218     # test
219     # test
220     # test
221     # test
222     # test
223     # test
224     # test
225     # test
226     # test
227     # test
228     # test
229     # test
230     # test
231     # test
232     # test
233     # test
234     # test
235     # test
236     # test
237     # test
238     # test
239     # test
240     # test
241     # test
242     # test
243     # test
244     # test
245     # test
246     # test
247     # test
248     # test
249     # test
250     # test
251     # test
252     # test
253     # test
254     # test
255     # test
256     # test
257     # test
258     # test
259     # test
260     # test
261     # test
262     # test
263     # test
264     # test
265     # test
266     # test
267     # test
268     # test
269     # test
270     # test
271     # test
272     # test
273     # test
274     # test
275     # test
276     # test
277     # test
278     # test
279     # test
280     # test
281     # test
282     # test
283     # test
284     # test
285     # test
286     # test
287     # test
288     # test
289     # test
290     # test
291     # test
292     # test
293     # test
294     # test
295     # test
296     # test
297     # test
298     # test
299     # test
300     # test
301     # test
302     # test
303     # test
304     # test
305     # test
306     # test
307     # test
308     # test
309     # test
310     # test
311     # test
312     # test
313     # test
314     # test
315     # test
316     # test
317     # test
318     # test
319     # test
320     # test
321     # test
322     # test
323     # test
324     # test
325     # test
326     # test
327     # test
328     # test
329     # test
330     # test
331     # test
332     # test
333     # test
334     # test
335     # test
336     # test
337     # test
338     # test
339     # test
340     # test
341     # test
342     # test
343     # test
344     # test
345     # test
346     # test
347     # test
348     # test
349     # test
350     # test
351     # test
352     # test
353     # test
354     # test
355     # test
356     # test
357     # test
358     # test
359     # test
360     # test
361     # test
362     # test
363     # test
364     # test
365     # test
366     # test
367     # test
368     # test
369     # test
370     # test
371     # test
372     # test
373     # test
374     # test
375     # test
376     # test
377     # test
378     # test
379     # test
380     # test
381     # test
382     # test
383     # test
384     # test
385     # test
386     # test
387     # test
388     # test
389     # test
390     # test
391     # test
392     # test
393     # test
394     # test
395     # test
396     # test
397     # test
398     # test
399     # test
400     # test
401     # test
402     # test
403     # test
404     # test
405     # test
406     # test
407     # test
408     # test
409     # test
410     # test
411     # test
412     # test
413     # test
414     # test
415     # test
416     # test
417     # test
418     # test
419     # test
420     # test
421     # test
422     # test
423     # test
424     # test
425     # test
426     # test
427     # test
428     # test
429     # test
430     # test
431     # test
432     # test
433     # test
434     # test
435     # test
436     # test
437     # test
438     # test
439     # test
440     # test
441     # test
442     # test
443     # test
444     # test
445     # test
446     # test
447     # test
448     # test
449     # test
450     # test
451     # test
452     # test
453     # test
454     # test
455     # test
456     # test
457     # test
458     # test
459     # test
460     # test
461     # test
462     # test
463     # test
464     # test
465     # test
466     # test
467     # test
468     # test
469     # test
470     # test
471     # test
472     # test
473     # test
474     # test
475     # test
476     # test
477     # test
478     # test
479     # test
480     # test
481     # test
482     # test
483     # test
484     # test
485     # test
486     # test
487     # test
488     # test
489     # test
490     # test
491     # test
492     # test
493     # test
494     # test
495     # test
496     # test
497     # test
498     # test
499     # test
500     # test
501     # test
502     # test
503     # test
504     # test
505     # test
506     # test
507     # test
508     # test
509     # test
510     # test
511     # test
512     # test
513     # test
514     # test
515     # test
516     # test
517     # test
518     # test
519     # test
520     # test
521     # test
522     # test
523     # test
524     # test
525     # test
526     # test
527     # test
528     # test
529     # test
530     # test
531     # test
532     # test
533     # test
534     # test
535     # test
536     # test
537     # test
538     # test
539     # test
540     # test
541     # test
542     # test
543     # test
544     # test
545     # test
546     # test
547     # test
548     # test
549     # test
550     # test
551     # test
552     # test
553     # test
554     # test
555     # test
556     # test
557     # test
558     # test
559     # test
560     # test
561     # test
562     # test
563     # test
564     # test
565     # test
566     # test
567     # test
568     # test
569     # test
570     # test
571     # test
572     # test
573     # test
574     # test
575     # test
576     # test
577     # test
578     # test
579     # test
580     # test
581     # test
582     # test
583     # test
584     # test
585     # test
586     # test
587     # test
588     # test
589     # test
590     # test
591     # test
592     # test
593     # test
594     # test
595     # test
596     # test
597     # test
598     # test
599     # test
600     # test
601     # test
602     # test
603     # test
604     # test
605     # test
606     # test
607     # test
608     # test
609     # test
610     # test
611     # test
612     # test
613     # test
614     # test
615     # test
616     # test
617     # test
618     # test
619     # test
620     # test
621     # test
622     # test
623     # test
624     # test
625     # test
626     # test
627     # test
628     # test
629     # test
630     # test
631     # test
632     # test
633     # test
634     # test
635     # test
636     # test
637     # test
638     # test
639     # test
640     # test
641     # test
642     # test
643     # test
644     # test
645     # test
646     # test
647     # test
648     # test
649     # test
650     # test
651     # test
652     # test
653     # test
654     # test
655     # test
656     # test
657     # test
658     # test
659     # test
660     # test
661     # test
662     # test
663     # test
664     # test
665     # test
666     # test
667     # test
668     # test
669     # test
670     # test
671     # test
672     # test
673     # test
674     # test
675     # test
676     # test
677     # test
678     # test
679     # test
680     # test
681     # test
682     # test
683     # test
684     # test
685     # test
686     # test
687     # test
688     # test
689     # test
690     # test
691     # test
692     # test
693     # test
694     # test
695     # test
696     # test
697     # test
698     # test
699     # test
700     # test
701     # test
702     # test
703     # test
704     # test
705     # test
706     # test
707     # test
708     # test
709     # test
710     # test
711     # test
712     # test
713     # test
714     # test
715     # test
716     # test
717     # test
718     # test
719     # test
720     # test
721     # test
722     # test
723     # test
724     # test
725     # test
726     # test
727     # test
728     # test
729     # test
730     # test
731     # test
732     # test
733     # test
734     # test
735     # test
736     # test
737     # test
738     # test
739     # test
740     # test
741     # test
742     # test
743     # test
744     # test
745     # test
746     # test
747     # test
748     # test
749     # test
750     # test
751     # test
752     # test
753     # test
754     # test
755     # test
756     # test
757     # test
758     # test
759     # test
760     # test
761     # test
762     # test
763     # test
764     # test
765     # test
766     # test
767     # test
768     # test
769     # test
770     # test
771     # test
772     # test
773     # test
774     # test
775     # test
776     # test
777     # test
778     # test
779     # test
780     # test
781     # test
782     # test
783     # test
784     # test
785     # test
786     # test
787     # test
788     # test
789     # test
790     # test
791     # test
792     # test
793     # test
794     # test
795     # test
796     # test
797     # test
798     # test
799     # test
800     # test
801     # test
802     # test
803     # test
804     # test
805     # test
806     # test
807     # test
808     # test
809     # test
810     # test
811     # test
812     # test
813     # test
814     # test
815     # test
816     # test
817     # test
818     # test
819     # test
820     # test
821     # test
822     # test
823     # test
824     # test
825     # test
826     # test
827     # test
828     # test
829     # test
830     # test
831     # test
832     # test
833     # test
834     # test
835     # test
836     # test
837     # test
838     # test
839     # test
840     # test
841     # test
842     # test
843     # test
844     # test
845     # test
846     # test
847     # test
848     # test
849     # test
850     # test
851     # test
852     # test
853     # test
854     # test
855     # test
856     # test
857     # test
858     # test
859     # test
860     # test
861     # test
862     # test
863     # test
864     # test
865     # test
866     # test
867     # test
868     # test
869     # test
870     # test
871     # test
872     # test
873     # test
874     # test
875     # test
876     # test
877     # test
878     # test
879     # test
880     # test
881     # test
882     # test
883     # test
884     # test
885     # test
886     # test
887     # test
888     # test
889     # test
890     # test
891     # test
892     # test
893     # test
894     # test
895     # test
896     # test
897     # test
898     # test
899     # test
900     # test
901     # test
902     # test
903     # test
904     # test
905     # test
906     # test
907     # test
908     # test
909     # test
910     # test
911     # test
912     # test
913     # test
914     # test
915     # test
916     # test
917     # test
918     # test
919     # test
920     # test
921     # test
922     # test
923     # test
924     # test
925     # test
926     # test
927     # test
928     # test
929     # test
930     # test
931     # test
932     # test
933     # test
934     # test
935     # test
936     # test
937     # test
938     # test
939     # test
940     # test
941     # test
942     # test
943     # test
944     # test
945     # test
946     # test
947     # test
948     # test
949     # test
950     # test
951     # test
952     # test
953     # test
954     # test
955     # test
956     # test
957     # test
958     # test
959     # test
960     # test
961     # test
962     # test
963     # test
964     # test
965     # test
966     # test
967     # test
968     # test
969     # test
970     # test
971     # test
972     # test
973     # test
974     # test
975     # test
976     # test
977     # test
978     # test
979     # test
980     # test
981     # test
982     # test
983     # test
984     # test
985     # test
986     # test
987     # test
988     # test
989     # test
990     # test
991     # test
992     # test
993     # test
994     # test
995     # test
996     # test
997     # test
998     # test
999     # test
1000    # test

```

รูปที่ 3.7 ตัวอย่างโปรแกรม interface ในการควบคุมแขนกลโดย Moveit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.5 การเขียน โปรแกรมที่ใช้ในบอร์ด STM32

บอร์ด STM32 ทำหน้าที่ในการควบคุมมอเตอร์ในแต่ละ Joint ของแขนกล โดย บอร์ด STM32 นั้นจะใช้ภาษา C ในการเขียน และพัฒนาในโปรแกรม Visual Studio การทำงานของ STM32 จะทำหน้าที่สั่งสตีปเปอร์มอเตอร์ให้ไปในตำแหน่งที่ต้องการ โดย ตำแหน่งของมอเตอร์จะได้รับมาจากบอร์ด ESP32 อีกที โดยสื่อสารผ่าน โปรโตคอล SPI

```

STM32_control.cpp
#include "stm32103C8T6.h"

// Number of Stepper Motors
#define STEPPER_UNITS 6

// Stepper Timing
#define STEPPER_TIMER_CN 1
#define STEPPER_TIMER TIM15
#define STEPPER_TIMER_CLK 80
#define STEPPER_TIMER_BASE 1

// Stepper Motor Configurations
#define HALF_STEP_DRIVE 0
#define FULL_STEP_DRIVE 1
#define HALF_STEP_DIVIDE 2
#define STEPPER_ORIGEN 0
#define STEPPER_BIPOLAR 1
#define DIR_CM 0
#define DIR_CCW 1

// Stepper Configuration Parameter Structure
typedef struct
{
    GPIO_TypeDef* TIM_GPIO[4];
    uint16_t IN_PIN[4];
    uint16_t STEPS_PER_REV;
    STEPPER_CFG;
    uint8_t STEPPING_Mode;
}STEPPER_CFGType;

void STEPERS_Init(void);
void STEPERS_Init_TIMER(HandleTypeDef* TIM_Handle);
void STEPPER_SetSpeed(uint8_t sub_STEPPER_Instance, uint16_t sub_REV);
void STEPPER_Stop_BlockImpairment(HandleTypeDef* sub_STEPPER_Instance, uint32_t sub_Steps, uint8_t sub_DIV);
void STEPPER_Stop_SpeedImpairment(HandleTypeDef* sub_STEPPER_Instance, uint32_t sub_Steps, uint8_t sub_DIV);
void STEPPER_Stop(uint8_t sub_STEPPER_Instance);
void STEPPER_Restart(void);
void STEPPER_Stop_Delay(HandleTypeDef* HTA);
  
```


รูปที่ 3.8 ตัวอย่าง โปรแกรมในบอร์ด STM32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.6 การเขียน โปรแกรมที่ใช้ในบอร์ด ESP32

เพื่อแบ่งการทำงานออกมาเป็นระบบ จึงได้แบ่งส่วนที่สื่อสารกับคอมพิวเตอรืทำหน้าที่ในบอร์ด ESP32 และส่งบอร์ด STM32 อีกที โดยโปรแกรมในบอร์ด ESP32 จะใช้ภาษา C ในการเขียนและเขียนบนโปรแกรม Arduino IDE ซึ่งสามารถใช้ในการพัฒนาโปรแกรมและอัปโหลดลงบอร์ดได้สะดวก

หน้าที่ของบอร์ด ESP32 คือการรับคำสั่งตำแหน่งของมอเตอร์แต่ละตัวจากคอมพิวเตอรืและส่งคำสั่งไปยังบอร์ด STM32 ผ่านโปรโตคอล SPI เพื่อให้ STM32 ควบคุมมอเตอร์อีกที



```

esp32 control / Arduino 1.8.16
File Edit Sketch Tools Help

esp32_control
#include <SPI.h>
#include <ESP.h>
#include <esp_d_mega/Int32.h>
#include <esp_d_mega/Float.h>
#include <sensor_mega/Range.h>
#include <esp_d_mega/String.h>

#define SPI_SCK 18
#define SPI_MOSI 23
#define SPI_MISO 19
#define SPI_CS 5
#define SS 5

NodeHandle nh;
esp_d_mega<Int32> motor_position;
nh.subscribe<esp_d_mega<Int32>> sub("motor_position", motorPositionCallback);

void setup() {
  Serial.begin(115200);
  nh.initNode();
  nh.subscribe(sub);

  pinMode(ESP_CS, OUTPUT);
  pinMode(ESP_MOSI, OUTPUT);
  pinMode(ESP_MISO, INPUT);
}

void loop() {
  nh.spinOnce();
  if (nh.available() > 0) {
    digitalWrite(SS, LOW);
  }
}

```

รูปที่ 3.9 ตัวอย่าง โปรแกรมในบอร์ด ESP32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.7 การเขียน โปรแกรม Image Detection

ในการจำแนกและตรวจจับวัตถุในภาพ ต้องใช้อัลกอริทึม

Image Detection ในการทำงานเริ่มทำการเขียน โปรแกรม Image Detection โดยใช้
โปรแกรม Visual Studio Code และใช้ภาษา Python ในการเขียน โปรแกรม

โปรแกรมจะทำหน้าที่นำภาพที่ได้จากกล้องทุกเฟรมมาคำนวณความ
น่าจะเป็นของวัตถุที่อยู่ใน Dataset โดยหากเจอวัตถุที่อยู่ใน Dataset จะทำการ
ตีกรอบ Bounding Box ขึ้นและระบุ (Classification) ว่าวัตถุนั้นคืออะไร โดยวัตถุ
ที่อยู่ใน Dataset คือ น่องไก่



```

1 import rospy
2 import cv2
3 from ultralytics import YOLO
4 import numpy as np
5 from sensor_msgs.msg import Image
6 from cv_bridge import CvBridge
7 from deepolarary import getDepthCommandRequest
8 import torch
9 print(torch.backends.mps.is_available())
10
11 model = YOLO("traints/weights/last.pt")
12
13 class ObjectDetection(object):
14
15     def __init__(self):
16         self.bridge = CvBridge()
17         rospy.init_node("Chicken_detect", anonymous=True)
18         rospy.Subscriber("/camera/rgb/image_raw", Image, self.update_frame_callback)
19         rospy.wait_for_message("/20200401/image_raw", Image)
20         self.get_depth_command = rospy.ServiceProxy("/vision/get_depth", getDepthCommand)
21
22     def update_frame_callback(self, data):
23         self.image = self.bridge.imgmsg_to_cv2(data, desired_encoding="bgr8")
24
25     def main(self):
26
27         while not rospy.is_shutdown():
28             frame = self.image
29             results = model(frame), device="mps"
30             result = results[0]
31             bboxes = np.array(result.bboxes.xyxy.cpu(), dtype="int")
32             classes = np.array(result.cls.cpu(), dtype="int")
33             keypoints = np.array(result.keypoints.xy.cpu(), dtype="int")
34             for keypoint, cls, bbox in zip(keypoints, classes, bboxes):

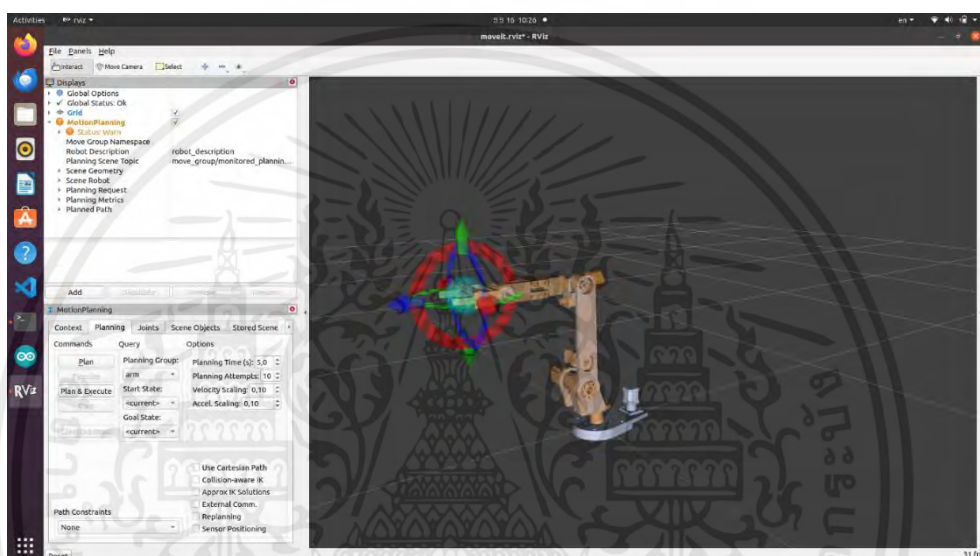
```

รูปที่ 3.10 ตัวอย่างโปรแกรม Image Detection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 ทดสอบการทำงานของแขนกลที่ออกแบบและตรวจสอบอีกครั้ง

จำลองการเคลื่อนที่ของแขนกลที่ได้รับรองศา ความเร็ว และความเร่ง โดยการคำนวณ Inversed Kinematic จาก Moveit โดยใช้ไลบรารีของ ROS ที่ชื่อว่า Rviz ที่จะจำลองและแสดงผลโมเดล 3 มิติของแขนกลและเคลื่อนไหวต่าง ๆ ตั้งแต่เริ่มต้นจนถึงสิ้นสุดเป้าหมาย



รูปที่ 3.11 หน้าต่างโปรแกรม Rviz ที่จำลองและแสดงผลโมเดลแขนกลแบบ 3 มิติ

3.8 จัดซื้ออุปกรณ์ต่าง ๆ

3.8.1 แบริ่ง ใช้สำหรับช่วยในการหมุนส่วนต่างๆของแขนกลที่เป็นจุดหมุน



รูปที่ 3.12 32009 (45x75x20mm) taper roller bearing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 30206 (30x62x17.25mm) taper roller bearing



รูปที่ 3.14 30204 (20x47x15.25mm) taper roller bearing



รูปที่ 3.15 AXK3552/AS3552 (35x52x4mm) thrust bearing with washers

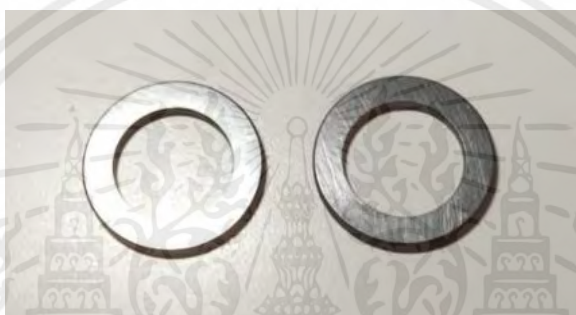


รูปที่ 3.16 NTA1625 (1.00x1.5625x0.0781 inch) thrust bearing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 TRA1625 (1.000x1.5625x0.0312 inch) thrust washers



รูปที่ 3.18 TRD1625 (1.000x1.5625x0.125 inch) thrust washers



รูปที่ 3.19 B1616 (1x1-1/4x1 inch) needle roller bearing

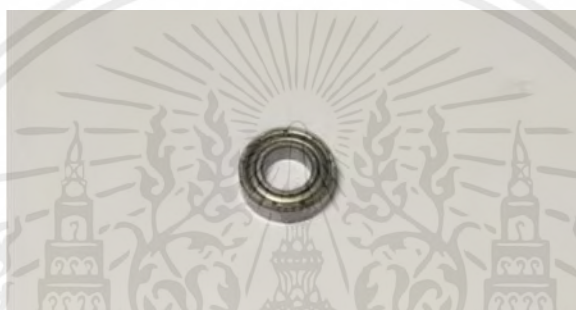


รูปที่ 3.20 HK1612 (16x22x12mm) needle roller bearing

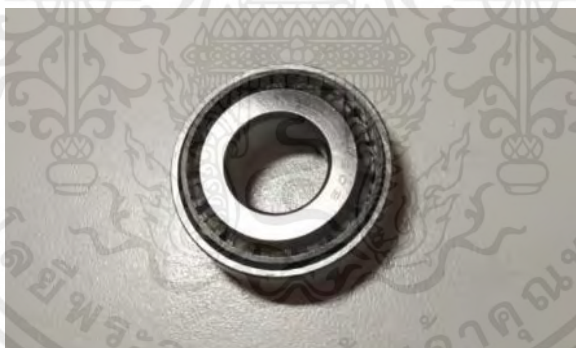
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 LM3UU 3mm. linear rod bearing



รูปที่ 3.22 688Z (8x16x5mm) groove ball bearing



รูปที่ 3.23 30203 (17x40x13.25mm) taper roller bearing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.2 เพลลา ใช้ยึดองค์ประกอบต่าง ๆ ให้สมบูรณ์และแข็งแรง



รูปที่ 3.24 3mm. x 85mm. shaft



รูปที่ 3.25 2mm. x 2mm. keystone (length 50mm.)



รูปที่ 3.26 8mm. keyed rotary shaft (length 50mm.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.3 น็อต ใช้สำหรับยึดและประกอบชิ้นส่วนต่าง ๆ ของแกนกล

#6 x .375 Thread Form Screw	6
M3x10 Flat Head Screw	13
M3x10 Set Screw	4
M3x14 Pan Head Screw	10
M3x14 Socket Head Screw	6
M3x18 Socket Head Screw	4
M3x20 Flat Head Screw	6
M3x20 Pan Head Screw	6
M3x25 Pan Head Screw	24
M3x3 Set Screw	4
M3x4 Set Screw	10
M3x5 Socket Head Screw	1
M3x6 Set Screw	7
M3x8 Socket Head Screw	9
M4 Nuts	4
M4 Washers	12
M4x10 Flat Head Screw	16
M4x10 Set Screw	12
M4x10 Socket Head Screw	16
M4x14 Flat Head Screw	2
M4x14 Socket Head Screw	1
M4x18 Flat Head Screw	3
M4x20 Pan Head Screw	2
M4x20 Socket Head Screw	8
M4x35 Pan Head Screw	4
M4x5 Set Screw	20
M6x14 Socket Head Screw	13
M6x18 Flat Head Screw	9
M6x20 Socket Head Screw	3
M8 x 14 Socket Head	1

ตารางที่ 3.1 ตารางรายการสั่งซื้อน็อต

3.8.4 สายพาน ใช้สำหรับทำให้แกนกลเคลื่อนที่ตามการหมุนของมอเตอร์ได้



รูปที่ 3.27 214-L Timing Belt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 150XL037 belt



รูปที่ 3.29 180XL037 belt

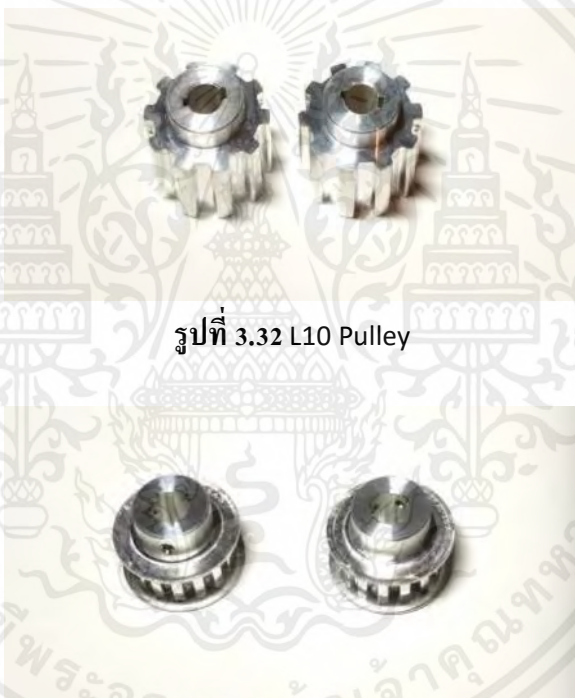
รูปที่ 3.30 84XL037 belt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.5 Pulley ใช้สำหรับยึดสายพานกับส่วนประกอบต่าง ๆ ของแขนกล



รูปที่ 3.31 60T XL pulley



รูปที่ 3.32 L10 Pulley

รูปที่ 3.33 XL 15 tooth 8mm. Bore



รูปที่ 3.34 XL 10 tooth 6mm. bore pulley

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.6 มอเตอร์ ใช้สำหรับการเคลื่อนที่ของแขนกล



รูปที่ 3.35 J1 motor SKU: NEMA17



รูปที่ 3.36 J2 motor SKU: NEMA23

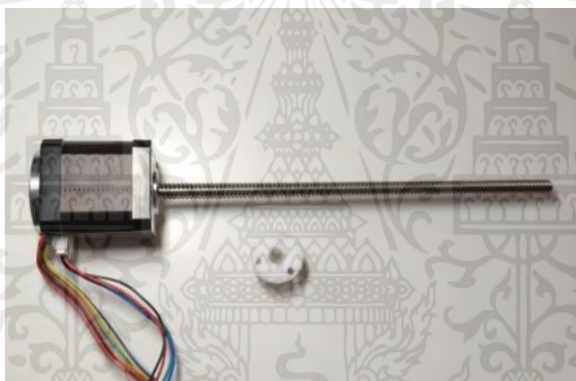


รูปที่ 3.37 J3 motor SKU: NEMA17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 J4 motor SKU: NEMA11



รูปที่ 3.39 J5 motor : NEMA17 with lead screw



รูปที่ 3.40 J6 motor SKU: NEMA17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.7 บอร์ด ใช้เป็นตัวควบคุมการทำงานของแขนกล ผ่านโปรแกรม



รูปที่ 3.41 STM 32 – STM32F103C8T6



รูปที่ 3.42 ESP 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 ประกอบแขนกล

หลังจากที่อุปกรณ์ทุกอย่างที่สั่งซื้อ ได้ถูกจัดส่งอย่างครบถ้วนแล้ว จึงถึงขั้นตอนการประกอบแขนกล และทำการ Wiring อุปกรณ์อิเล็กทรอนิกส์



รูปที่ 3.43 กลไกของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.44 การ Wiring สายไฟ



รูปที่ 3.45 ประกอบบอร์ด STM32 เข้ากับบอร์ด ESP32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 ทดสอบการทำงานและแก้ไขข้อผิดพลาด

เมื่อทำการประกอบแขนกล และเขียน โปรแกรมทุกอย่างเสร็จสิ้นตามแผนที่วางเอาไว้ ก็ถึงขั้นตอนการทดสอบการทำงานของแขนกลและโปรแกรมการควบคุมแขนกล โดยทดสอบว่าแขนกลและโปรแกรมการควบคุมแขนกลนั้นทำงานได้ถูกต้องตามที่กำหนดหรือไม่ ซึ่งพบข้อบกพร่องบางส่วนใน โปรแกรมจึงทำให้แขนกลไม่สามารถทำงานได้ ถูกต้องตามกำหนด ซึ่งก็ได้แก้ไขให้ถูกต้องเป็นที่เรียบร้อยแล้ว

3.11 สรุปผลและจัดทำรายงาน

เมื่อเสร็จสิ้นทุกขั้นตอนดังกล่าวแล้วจึงทำการสรุปผลทุกอย่างที่ได้ และจัดทำ รายงานเพื่อนำส่งอาจารย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

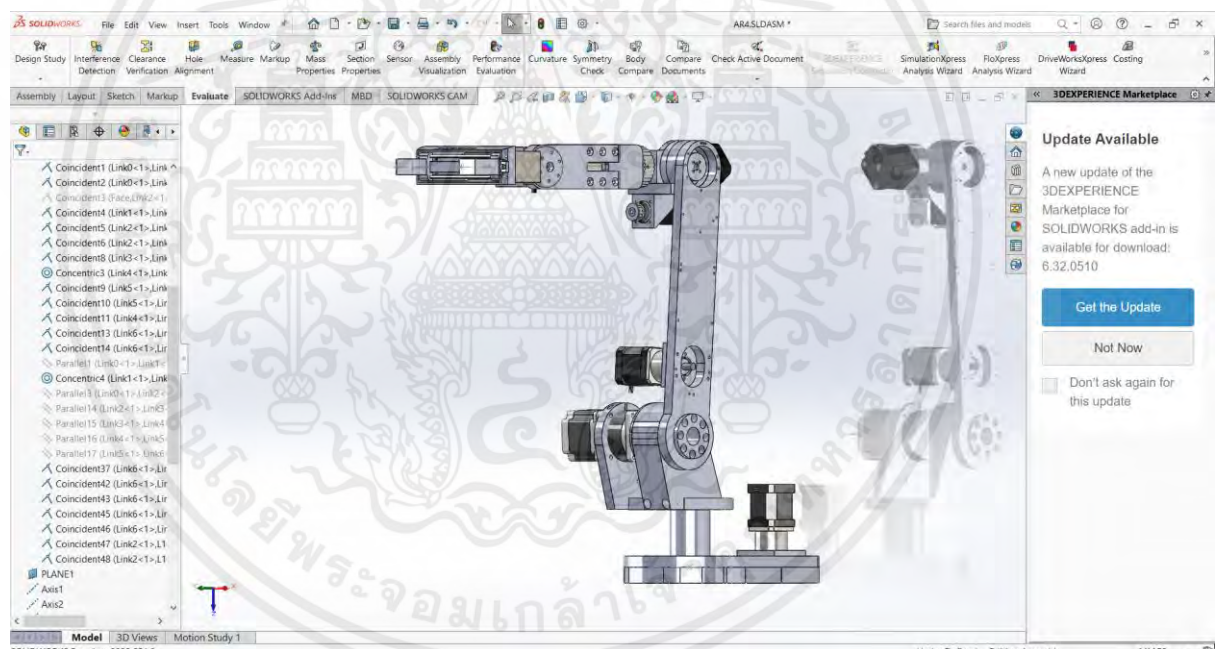
บทที่ 4

ผลการดำเนินงาน

การพัฒนาแขนกลต้นแบบเพื่อใช้สำหรับศึกษาการทำงานของแขนกล และฝึกฝนทักษะการออกแบบ และการเขียนโปรแกรมโดยใช้ ROS มาช่วยในการควบคุมแขนกลผ่านคอมพิวเตอร์ เพื่อต่อยอดและพัฒนาไปในอนาคตต่อไป

4.1 ผลการจัดทำแขนกล

สำหรับการจัดทำแขนกลจะแบ่งเป็น 2 ส่วน คือ ส่วนที่เป็นการออกแบบแขนกลโดยใช้โปรแกรม SOLIDWORKS และส่วนที่เป็นการประกอบแขนกลจริง



รูปที่ 4.1 การออกแบบแขนกลด้วยโปรแกรม SOLIDWORKS

การออกแบบแขนกลนั้นจะต้องออกแบบแต่ละชิ้นส่วน แล้วจึงนำชิ้นส่วนทั้งหมดมาประกอบรวมกัน ซึ่งมีต้นแบบจากแขนกล AR4 โดยคำนึงถึงโหลดสูงสุดที่แขนกลรับได้ และการเคลื่อนที่ของแขนกลจะไม่ขัดกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 แขนกลแต่ละส่วนที่ออกแบบเสร็จสิ้นแล้ว จึงนำมาประกอบเข้าด้วยกัน จนถึงสามารถสามารถจำลองการขยับของแขนกลได้จริง

ในส่วนของการประกอบแขนกล จะเริ่มประกอบแต่ละส่วน ซึ่งรวมไปถึงอุปกรณ์ขับเคลื่อนอย่างมอเตอร์ และอุปกรณ์ส่งกำลังอย่างสายพาน แล้วจึงนำแต่ละส่วนมาประกอบรวมกันอีกที เพื่อความสะดวกในการแก้ไขเมื่อเกิดข้อผิดพลาดขึ้น

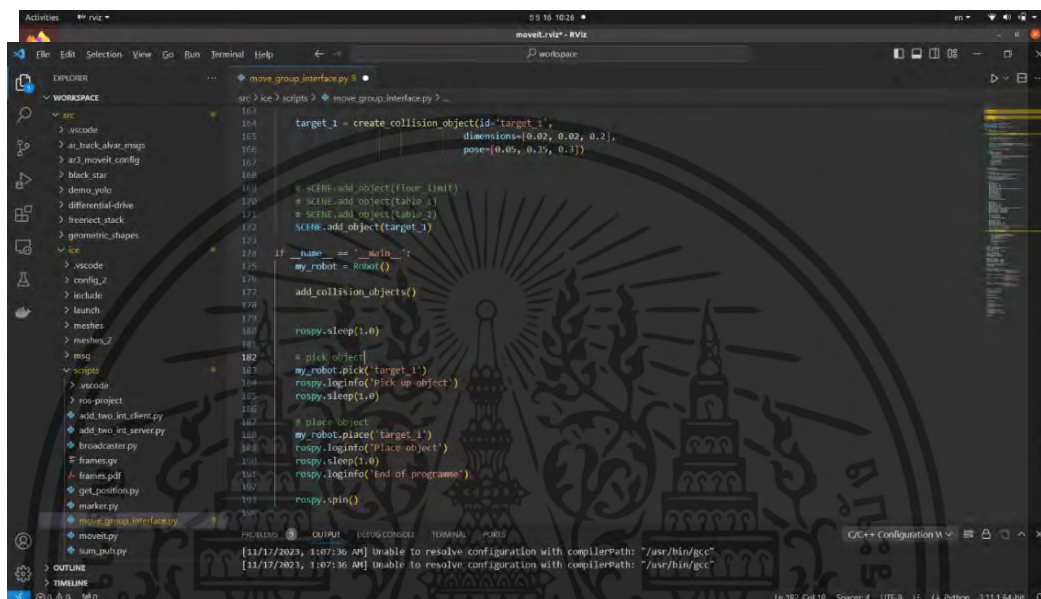


รูปที่ 4.2 แขนกลที่เสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการเขียนโปรแกรมควบคุมแขนกล

การเขียนโปรแกรมควบคุมแขนกล เพื่อให้สามารถควบคุมแขนกลโดยการทำงานผ่านคอมพิวเตอร์ในการออกแบบโปรแกรมโดยใช้ ROS ในการควบคุม โดย ROS จะมีไลบรารีสำหรับการควบคุมแขนกลคือ Moveit ซึ่งจะเขียนโดยใช้ภาษา Python



```

target_1 = create_collision_object(id="target_1",
                                  dimensions=[0.02, 0.02, 0.2],
                                  pose=[0.05, 0.15, 0.1])

# SCHE.add_object(floor_limit)
# SCHE.add_object(table_1)
# SCHE.add_object(table_2)
SCHE.add_object(target_1)

if __name__ == '__main__':
    my_robot = Robot()
    add_collision_objects()

    rospy.sleep(1.0)

    # pick object
    my_robot.pick("target_1")
    rospy.loginfo("pick up object")
    rospy.sleep(1.0)

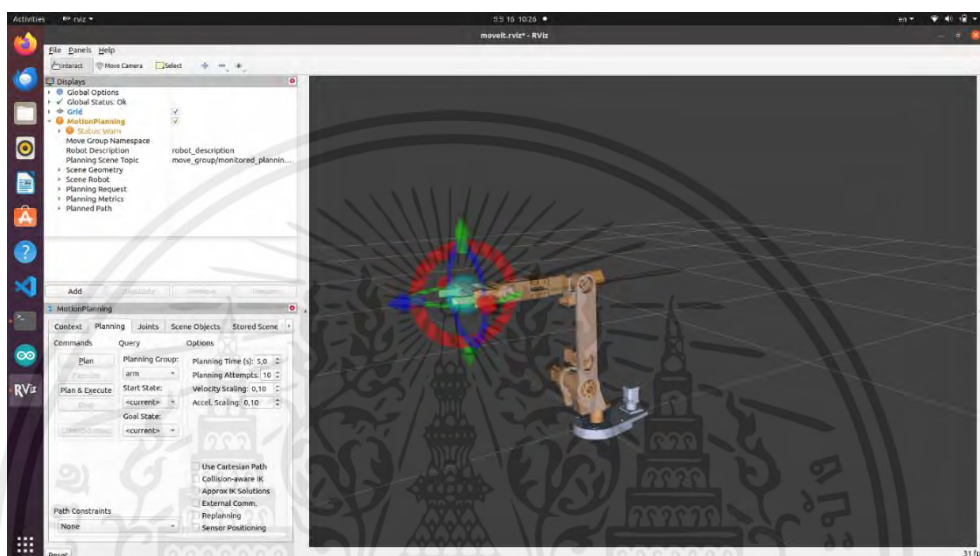
    # place object
    my_robot.place("target_1")
    rospy.loginfo("Place object")
    rospy.sleep(1.0)
    rospy.loginfo("end of programme")

    rospy.spin()
  
```

รูปที่ 4.3 โปรแกรมการควบคุมแขนกล โดยใช้ ROS Moveit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้ป้อนตำแหน่งในพิกัดจาก 3 มิติ x y z ในโลกจริงใส่ในโปรแกรม โปรแกรมจะคำนวณ Inverse Kinematic สำหรับองศา ความเร็ว และความเร่งของแต่ละมอเตอร์ และส่งองศา ความเร็ว และความเร่งต่าง ๆ ของแต่ละมอเตอร์ไปยังบอร์ด ESP32 โดยใช้ไลบรารี roserial เพื่อควบคุมมอเตอร์นั้น ๆ และจะจำลองและแสดงผลการเคลื่อนที่ผ่านโปรแกรม Rviz

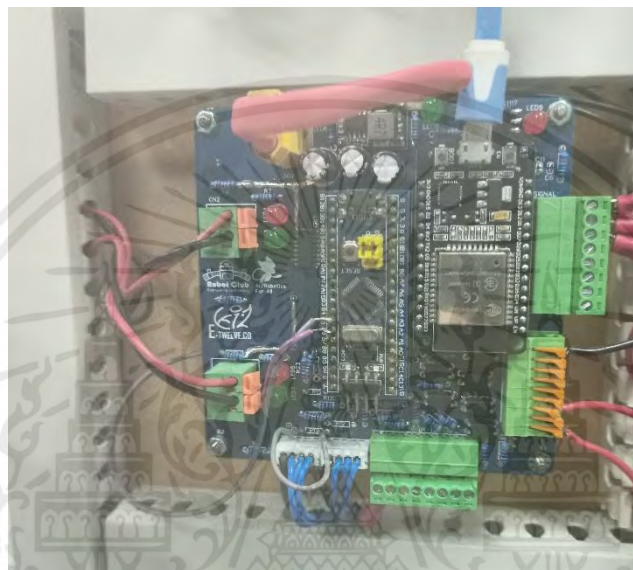


รูปที่ 4.4 ควบคุมและจำลองการเคลื่อนไหวของแขนกลผ่าน โปรแกรม Rviz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการเขียนบอร์ด STM32 และ ESP32

การเขียนบอร์ดควบคุมแขนกลมี 2 บอร์ดคือบอร์ด STM32 และบอร์ด ESP32 โดยบอร์ด STM32 ทำหน้าที่ควบคุมมอเตอร์ของแขนกลในแต่ละ Joint โดยจะรับคำสั่งจากบอร์ด ESP32 และบอร์ด ESP32 ทำหน้าที่ควบคุมบอร์ด STM32 โดยรับคำสั่งจากคอมพิวเตอร์อีกที

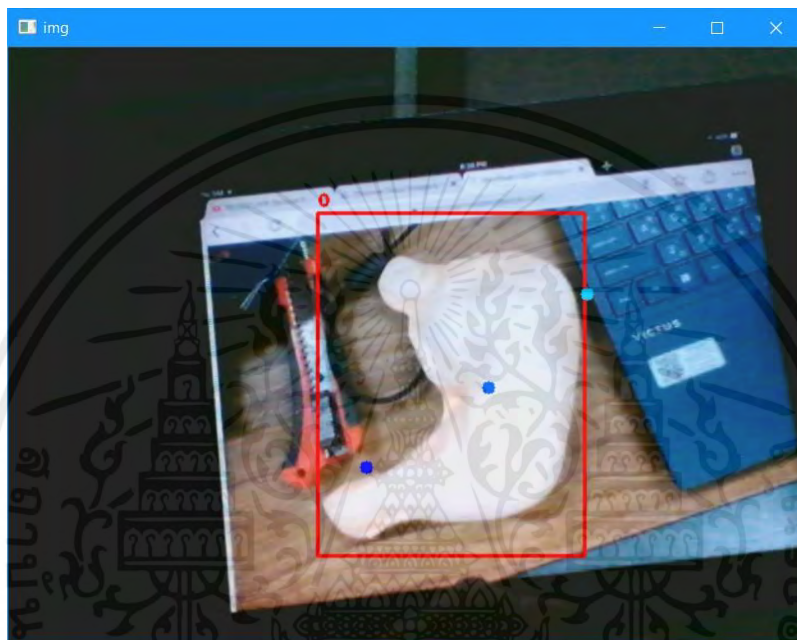


รูปที่ 4.5 บอร์ด STM32 และ ESP32 ที่ต่อกันสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการเขียนโปรแกรม Image Detection

การเขียนโปรแกรม Image Detection ใช้โมเดล YOLO ซึ่งเป็น AI ที่ใช้ในงาน Image Detection และเขียนด้วยภาษา Python จึงง่ายแก่การพัฒนา โดยโปรแกรมจะรับรูปภาพที่กล้องสตรีมมาและตรวจจับและตีกรอบน่องไก่ที่เจอ



รูปที่ 4.6 โปรแกรม Image Detection ตรวจจับน่องไก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การออกแบบแขนกลในโครงการนี้ให้มีความแข็งแรงและประสิทธิภาพที่สูง เพื่อให้สามารถทำงานได้อย่างเสถียรและแม่นยำ นอกจากนี้ยังมีการคำนึงถึงปัญหาต่าง ๆ เช่น การลดค่าใช้จ่ายในส่วนที่ไม่ต้องรับโหลดหนัก และการสร้าง Dataset สำหรับการเทรน Image Detection ด้วยการประมวลผลภาพที่ได้จากการถ่ายภาพ 3 มิติของน้องไก่จำลอง โดยมีขั้นตอนการทำงานดังนี้

5.1.1 การออกแบบแขนกล

ใช้โมเดล Open Source AR4 6 degrees of freedom

เพื่อให้แขนกลมีความยืดหยุ่นและสามารถทำงานในทิศทางหลายทิศทางได้ ในส่วนของชิ้นงานจะมีบางส่วนที่เป็นการกลึงชิ้นงานแบบอคูมิเนียมเฉพาะ บางส่วนที่มีหน้าที่รับโหลดค่อนข้างมาก และบางส่วนเป็นการปรี้น 3 มิติสำหรับ ส่วนที่ไม่ได้รับน้ำหนักมาก เพื่อลดค่าใช้จ่ายที่สูงเกินงบประมาณที่มี

5.1.2 การเทรน Image Detection

ปรี้น 3 มิติของไก่จำลองและถ่ายภาพในหลาย ๆ มุมเพื่อสร้าง Dataset สำหรับการเทรน Image Detection โดยใช้โปรแกรม Visual Studio Code และ Python เพื่อเขียนโปรแกรมที่ใช้ในการจำแนกและตรวจจับวัตถุในภาพ

5.1.3 การควบคุมแขนกลโดยใช้ ROS

ใช้ ROS 1 Noetic บน Ubuntu 20.04 เพื่อควบคุมแขนกล ใช้โปรแกรม Visual Studio Code เพื่อใช้สำหรับควบคุม Library Moveit ด้วยการ ใช้ Moveit Assistant Setup เพื่อตั้งค่าต่าง ๆ สำหรับการใช้งาน Moveit

5.1.4. การควบคุมมอเตอร์โดยใช้บอร์ด STM32

เขียนโปรแกรมบนบอร์ด STM32 โดยใช้ภาษา C และใช้ Inversed Kinematic จาก Moveit เพื่อคำนวณค่าองศา ความเร่ง และความเร็วของมอเตอร์ และมีบอร์ด STM32 ทำหน้าที่ควบคุมมอเตอร์ในแต่ละ Joint ของแขนกล

5.1.5. การทำงานของบอร์ด ESP32

เขียนโปรแกรมในบอร์ด ESP32 เพื่อสื่อสารกับคอมพิวเตอร์ และควบคุมการทำงานของบอร์ด STM32 ใช้ Arduino IDE ในการพัฒนาโปรแกรม

5.1.6. การจำแนกและตรวจจับวัตถุ

ใช้ Image Detection Algorithm เพื่อจำแนกและตรวจจับวัตถุในภาพ และใช้ภาษา Python กับโปรแกรม Visual Studio Code ในการเขียนโปรแกรม

5.1.7. การทดลองเคลื่อนที่

ทดลองการเคลื่อนที่ของแขนกลโดยใช้ค่าองศา ความเร็ว และความเร่งที่ได้จาก Moveit ใช้ Library Rviz เพื่อจำลองและแสดงผลโมเดล 3 มิติของแขนกล

5.2 ปัญหาและอุปสรรคที่พบ

5.2.1 เนื่องจากขั้นตอนการทำ มีขั้นตอนบางขั้นเสียหยา ทำให้ไม่สามารถใช้งานได้

5.2.2 ความรู้และความชำนาญไม่เพียงพอต่อการประกอบแขนกล

5.2.3 เกิดปัญหาอันเนื่องจากการประกอบเครื่องจักรไม่ตรงกับที่ออกแบบไว้

5.2.4 เกิดความล่าช้าจากตารางการทำงาน

ระยะเวลาในการทำโปรเจกต์ค่อนข้างน้อย

5.3 แนวทางการแก้ไข

- 5.3.1 จำเป็นต้องนำสิ่งของอื่นๆที่มีอยู่มาดัดแปลงให้สามารถใช้งานได้
- 5.3.2 เปิดคู่มือต่างๆ ไม่ว่าจะเป็นการใช้งานอุปกรณ์ วิธีการติดตั้งแขนกล
- 5.3.3 ทำการปรับความเข้าใจ และปรับเปลี่ยนอุปกรณ์

เพื่อให้สามารถเชื่อมต่อและทำการทดสอบ

เครื่องจักรให้แล้วเสร็จ

5.4 ข้อเสนอแนะ

- 5.4.1 ควรประเมินชิ้นส่วนว่ามีโอกาสเกิดความเสียหายระหว่างการประกอบ
มากน้อยเพียงใด หากมีโอกาสเสียหายสูง จะทำการซื้อชิ้นส่วนเพิ่มมาเพื่อเก็บสำรองไว้
- 5.4.2 สามารถลดต้นทุนในการซื้อชิ้นส่วนได้มากขึ้น เช่น
ตัวแขนกลบางชิ้นที่สั่งกลึง และบางชิ้นที่สามารถพิมพ์ 3 มิติได้
- 5.4.3 ควรศึกษาวิธีการประกอบแขนกลให้มากขึ้นและรู้วิธีการดัดแปลงชิ้นส่วน
ในกรณีที่ต้องทำการแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] <https://wiki.ros.org/noetic>
- [2] https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- [3] <https://www.st.com/resource/en/datasheet/stm32f205rb.pdf>
- [4] <https://docs.ultralitics.com/>
- [5] <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi/all>
- [6] <https://wiki.ros.org/roserial>
- [7] https://ros-planning.github.io/moveit_tutorials/
- [8] <http://wiki.ros.org/rviz>
- [9] <https://help.ubuntu.com/20.04/ubuntu-help/index.html>
- [10] <https://www.zonemaker.com/product/1069/hanpose-stepper-motor-nema23-%E0%B9%81%E0%B8%A3%E0%B8%87%E0%B8%9A%E0%B8%B4%E0%B8%94-1-89n-m-2-8a-shaft-6-35mm-23hs7628-6-35>
- [11] <https://www.zonemaker.com/product/40/usongshine-stepper-motor-nema17-%E0%B9%81%E0%B8%A3%E0%B8%87%E0%B8%9A%E0%B8%B4%E0%B8%94-42-n-cm-1-5a-17hs4401s-black-end>
- [12] https://openkinect.org/wiki/Main_Page
- [13] <https://www.zonemaker.com/product/154/tb6600-microstep-driver-peak-4a-9-42-vdc-aluminum-cover-2-phase>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้วิจัย

ชื่อ-สกุล	นายชนกฤต เจนธนุรเวท
รหัสประจำตัวนักศึกษา	63010393
สาขาวิชา	แมคคาทรอนิกส์และอโตเมชัน คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วันเดือนปีเกิด	10 กุมภาพันธ์ 2545
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนสารสาสน์วิเทศชลบุรี
มัธยมศึกษา	โรงเรียนสารสาสน์วิเทศชลบุรี
ปีที่เข้ารับการศึกษ	พ.ศ.2563
ที่อยู่ที่สามารถติดต่อได้	167/1 ม.4 ต.หนองซาก อ.บ้านบึง จ.ชลบุรี รหัสไปรษณีย์ 20170
อีเมล	63010393@kmitl.ac.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อ-สกุล	นายธนกร หนองหารพิทักษ์
รหัสประจำตัวนักศึกษา	63010391
สาขาวิชา	แมคคาทรอนิกส์และอโตเมชัน คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
วันเดือนปีเกิด	1 พฤษภาคม 2545
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนศิริพระของ
มัธยมศึกษา	โรงเรียนระยองวิทยาคม
ปีที่เข้ารับการศึกษ	พ.ศ.2563
ที่อยู่ที่สามารถติดต่อได้	9/26 ม.2 ต.ทับมา อ.เมือง จ.ระยอง รหัสไปรษณีย์ 21000
อีเมล	63010391@kmitl.ac.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้