

การออกแบบและสร้างหุ่นยนต์ทรงกลม



นายศิวศิลป์ พรจำศิลป์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมเมคคาทรอนิกส์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2560

DESIGN AND PROTOTYPING OF SPHERICAL ROBOT



Siwasilp Pornjumsilp

**A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Engineering in Mechatronics Engineering**

Suranaree University of Technology

Academic Year 2017

การออกแบบและสร้างหุ่นยนต์ทรงกลม

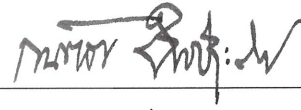
มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นักศึกษานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา
ตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์



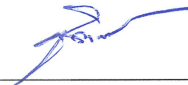
(อ. ดร. โสรกฐา แจ็งการ)

ประธานกรรมการ



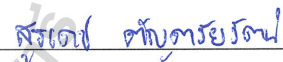
(รศ. ร.อ. ดร. กนต์ธร ชำนิประศาสน์)

กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)



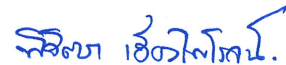
(อ. ดร. ชิตพงษ์ เวชไชสงค์)

กรรมการ



(อ. ดร. สุรเดช ตัญตวรรษรัตน์)

กรรมการ



(อ. ดร. พิจิตรา เอื่องไพโรจน์)

กรรมการ



(ศ. ดร. สันติ แม้นศิริ)

รองอธิการบดีฝ่ายวิชาการและพัฒนาความเป็นสากล



(รศ. ร.อ. ดร. กนต์ธร ชำนิประศาสน์)

คณบดีสำนักวิชาวิศวกรรมศาสตร์


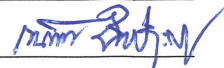
ศิวศิลป์ พรจำศิลป์ : การออกแบบและสร้างหุ่นยนต์ทรงกลม (DESIGN AND
PROTOTYPING OF SPHERICAL ROBOT) อาจารย์ที่ปรึกษา : รองศาสตราจารย์
เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 128 หน้า.

ปัจจุบันเทคโนโลยีเกี่ยวกับหุ่นยนต์ได้เข้ามามีบทบาทสำคัญกับการดำเนินชีวิตของมนุษย์
มากขึ้นเรื่อย ๆ ไม่ว่าจะเป็นสื่อการเรียนรู้ในแวดวงวิทยาศาสตร์เทคโนโลยี ภาคการเกษตร
ภาคอุตสาหกรรม รวมไปถึงจนถึงวงการภาพยนตร์ ซึ่งสามารถพบเห็นได้อย่างแพร่หลาย
อันเป็นแรงผลักดันให้เกิดจินตนาการกับเหล่านักนักเรียน นักวิทยาศาสตร์ ตลอดจนถึงวิศวกร
ผู้ออกแบบสร้างสรรค์สิ่งประดิษฐ์ใหม่ ๆ เช่น หุ่นยนต์ทรงกลมที่เคลื่อนที่ด้วยการกลิ้งมีวงตัว
ดังที่ปรากฏในมหากาพย์ภาพยนตร์ นิยายวิทยาศาสตร์เรื่อง STAR WARS VII หรือที่รู้จักกัน
ในชื่อหุ่นยนต์ BB8 (ASTROMECH DROID) กล่าวคือเป็นหุ่นยนต์ที่มีความน่าสนใจเป็นอย่างมาก
เนื่องจากต้องใช้จินตนาการ และองค์ความรู้หลากหลายแขนง ในการสร้างสรรค์หุ่นยนต์นี้ขึ้น

หุ่นยนต์ BB8 ในอนาคตนั่น จะเป็นแรงบันดาลใจในการสร้างสรรค์ผลงานการประดิษฐ์
ให้กับเหล่าวิศวกร นักออกแบบรุ่นใหม่ ซึ่งงานวิจัยนี้เป็นการสร้างหุ่นยนต์ทรงกลมที่ใช้วิทยาการ
และเทคนิคเชิงวิศวกรรมในการออกแบบและสร้างหุ่นยนต์ ให้สามารถควบคุมการทำงาน
ตามภารกิจที่กำหนดได้ อีกทั้งในส่วนของ การออกแบบได้มีการออกแบบพื้นที่ภายใน
เพื่อใช้สำหรับติดตั้งอุปกรณ์เสริม ไม่ว่าจะเป็นส่วน Interface กับมนุษย์หรือ Module ต่าง ๆ
เพื่อเพิ่มลูกเล่นให้กับหุ่นยนต์นี้ ได้อย่างหลากหลาย กล่าวคือหุ่นนี้สามารถนำไปประยุกต์ใช้งาน
ในด้านต่าง ๆ เช่น การนำเอาไปเป็นหุ่นยนต์นำทางคนพิการ ไก่นำทัวร์ มาสคอตเปิดงานต่าง ๆ
รวมไปถึงงานวิจัยที่เกี่ยวข้อง แต่ทว่าต้นทุนในการสร้างหุ่นยนต์ BB8 ดังในภาพยนต์นั้น มีค่าใช้จ่าย
ในการลงทุนสร้างสูงมาก จึงทำให้หุ่นยนต์นี้ยังไม่เป็นที่รู้จักในเชิงลึก ซึ่งส่วนใหญ่จะรู้จักหุ่นยนต์นี้
เพียงในภาพยนต์เท่านั้น ซึ่งถ้าสามารถแก้ไขปัญหานั้นได้ นักออกแบบจะสามารถพัฒนา
เทคโนโลยีด้านหุ่นยนต์ให้ทัดเทียมกับเทคโนโลยีที่มีราคาสูงจากต่างประเทศได้

งานวิจัยนี้จะเป็นการออกแบบโครงสร้างภายใน เปลือกหุ้ม ระบบควบคุมการทำงาน
ของหุ่นยนต์ ให้สามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ ผลที่ได้จากงานวิจัยนี้จะเป็น
จุดเริ่มต้นที่ทำให้สามารถสร้างหุ่นยนต์ที่มีราคาถูกกว่าการสร้างในต่างประเทศ และสามารถผลิต
ขึ้นเองได้ ในมหาวิทยาลัยรวมถึงการเพิ่มขีดความสามารถของหุ่นยนต์ คือสามารถเพิ่มเติม
รูปแบบการทำงานพัฒนานวัตกรรม และระบบฝังในตัวหุ่นยนต์นี้ต่อไปได้อย่างไม่สิ้นสุด

สาขาวิชา วิศวกรรมเครื่องกล
ปีการศึกษา 2560

ลายมือชื่อนักศึกษา 
ลายมือชื่ออาจารย์ที่ปรึกษา 

SIWASILP PORNJUMSILP : DESIGN AND PROTOTYPING OF
SPHERICAL ROBOT. THESIS ADVISOR : ASSOC. PROF. FLT. LT.
KONTRON CHAMNIPRASART, Ph.D., 128 PP.

BB8/SPHERICAL ROBOT/ROLLING MOTION

Nowadays, robotic technology has become increasingly important to human life. Robots are applied in several industries, e.g. agricultural industry, or even the film industry. Different kinds of robot could be seen in the movies. One of the most popular robot is the "BB-8" (ASTROMECH DROID) in the famous science-fiction film named STAR WARS VII, this robot is very interesting because it requires cognitive and imagination skills for creating this robot. The "BB-8" robot in the future will inspire the creation, and invention of the new generation of engineers. The advantages of this robot, besides the common movement of the robot, is the control technology that allows robots to interact with humans, including operating system modules, and feature enhancements. This robot that can be used with many applications. For example robot for guiding people with disabilities, robot for guiding location or giving placement tips, etc. However, the cost of producing this robot is very high, making it hard for the public to access it. People know the "BB-8" only in movies.

The details of the "BB-8" production are confidential, resulting in the difficulty of developing this type of robot to other applications. This robot is expected to emerge with embedded system in order to enhance the capability of the robots for a variety of applications.

School of Mechanical Engineering

Academic Year 2017

Student's Signature

Advisor's Signature

Sinasilp Pornjumsilp.
Kont Chant

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี ทั้งนี้ผู้วิจัยขอขอบพระคุณบุคคลและหน่วยงานต่าง ๆ ที่ได้ให้คำปรึกษา ชี้แนะแนวทาง และให้ความช่วยเหลืออย่างดียิ่งเสมอมา ดังนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี รวมถึงบุคลากรประจำศูนย์เครื่องมือ ที่ได้ให้ความช่วยเหลือในการใช้อุปกรณ์ และเครื่องมือต่าง ๆ ในการทำงานวิจัยนี้

รองศาสตราจารย์เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ และ ดร.สุรเดช ตัญญูทรัพย์รัตน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ถ่ายทอดความรู้ อบรมสั่งสอน ให้คำปรึกษา ชี้แนะข้อมูลต่าง ๆ ที่เป็นประโยชน์ในงานวิจัย และแนวทางการแก้ปัญหาต่าง ๆ ด้วยความเมตตากรุณาเสมอมา

อาจารย์ ดร.พิจิตรา เอื้องไพโรจน์ อาจารย์ประจำสาขาวิชาวิศวกรรมเครื่องกล ที่ได้ให้คำปรึกษาทางด้านภาษาต่างประเทศ แนวคิดในการออกแบบการทดลองด้วยความเมตตากรุณา

อาจารย์วิชัย ศรีสุรภักดิ์ ที่มอบโอกาสประสิทธิประสาทวิชาความรู้ ทักษะและประสบการณ์ ในการดำเนินชีวิตตลอดจนท่านเป็นต้นแบบในการดำเนินชีวิต ด้วยความเมตตากรุณาเมตตาอุเบกขา

คุณสิทธิพนธ์ จุฬนธิ และอาจารย์อภิสิทธิ์ หล่อนกลาง ผู้ร่วมในงานวิจัยนี้ ที่ได้ช่วยเหลือ แนะนำแนวทางการแก้ไขปัญหา และให้ข้อมูลในการทำงานวิจัยนี้ จนสำเร็จลุล่วงไปด้วยดี

คุณอดิศักดิ์ แดงบุตร และเพื่อนบัณฑิตศึกษาทุกท่านที่คอยให้ความร่วมมือสนับสนุนผู้วิจัย สละเวลา แรงกาย แรงใจ ให้ความช่วยเหลือเกื้อกูลและเป็นกำลังใจที่ดีโดยตลอดมา

สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์เล่มนี้ ผู้วิจัยขอมอบให้กับบิดา มารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ และถ่ายทอดประสบการณ์ที่ดีให้แก่ผู้วิจัยตลอดมา ตลอดจนให้การสนับสนุนและเป็นกำลังใจ ในการดำเนินงานวิจัยในครั้งนี้ จนประสบความสำเร็จ

ศิวศิลป์ พรจำศิลป์

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย).....	ก
บทคัดย่อ (ภาษาอังกฤษ).....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูปภาพ.....	ช
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของงานวิจัย.....	1
1.2 วัตถุประสงค์การวิจัย.....	2
1.3 ขอบเขตการวิจัย.....	2
1.4 วิธีดำเนินงานวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
1.6 สถานที่ทำงานวิจัย.....	3
1.7 ต้นแบบหุ่นยนต์.....	3
1.8 แผนการดำเนินการ.....	4
2 บริหารศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง	5
2.1 กล่าวนำ.....	5
2.2 ไมโครคอนโทรเลอร์ (Microcontroller).....	9
2.3 อาดูโน่ (Arduino).....	10
2.4 รีโมทคอนโทรล (Radio Control).....	15
2.5 ตัวขับเคลื่อนมอเตอร์ (Driver Motor).....	17
2.6 แม่เหล็กนีโอไดเมียม (Neodymium magnets).....	18
2.7 ล้อโอมนิ (Omni wheel).....	21
2.8 แหล่งพลังงาน (Battery).....	27

สารบัญ (ต่อ)

หน้า

3	วิธีการดำเนินการวิจัย	31
3.1	กล่าวนำ	31
3.2	การออกแบบขั้นตอนการวิจัย	31
3.3	การออกแบบขั้นตอนการสร้าง	32
3.4	ศึกษาข้อมูลโครงสร้างก่อนดำเนินการ	33
3.5	ขั้นตอนการปฏิบัติงาน	34
4	ผลการทดลอง	48
4.1	กล่าวนำ	48
4.2	ศึกษาความสามารถของอุปกรณ์	48
4.3	วิเคราะห์ข้อมูล	60
4.4	ปัญหาที่พบ	61
4.5	แนวทางแก้ไข	61
5	สรุปและข้อเสนอแนะ	75
5.1	สรุปผลงานวิจัย	75
5.2	ข้อเสนอแนะ	76
	รายการอ้างอิง	77
	ภาคผนวก	
	ภาคผนวก ก Hardware & design	80
	ภาคผนวก ข รายละเอียดโปรแกรมควบคุม	102
	ภาคผนวก ค บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่	121
	ประวัติผู้เขียน	128

สารบัญตาราง

ตารางที่	หน้า
1.1 ตารางแผนการดำเนินงาน.....	4
2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ Arduino Atmega 168.....	18
2.2 ตัวอย่างคุณสมบัติเฉพาะของแม่เหล็กนีโอไดเมียม.....	20
2.3 ตัวอย่างรูปร่างแม่เหล็กนีโอไดเมียม.....	20
2.4 แม่เหล็กนีโอไดเมียม (Nd-Fe-B) CP แบบจาน รูปวงแหวน).....	30
3.1 ขนาดของ Robot.....	35

สารบัญรูป

รูปที่	หน้า
1.1 โครงสร้างเปลือกนอกของหุ่นยนต์ BB8.....	3
1.2 แบบจำลองโครงสร้างภายในหุ่นยนต์.....	3
1.3 ถอด Dimensions หุ่นยนต์ BB8 จากภาพยนต์.....	4
2.1 BB8 – Master [1].....	5
2.2 Hamster Ball [2].....	6
2.3 Universal Wheel [3].....	7
2.4 Shape Memory Alloys[4].....	7
2.5 New Spherical robot with wind-driven [5].....	7
2.6 ไมโครคอนโทรลเลอร์.....	10
2.7 Arduino Duemilanove.....	12
2.8 หน่วยความจำและขาสัญญาณของ Arduino Atmega 168.....	13
2.9 ตำแหน่งพอร์ตของ Arduino ATmega 168.....	14
2.10 วิทยุบังคับ RadioLink AT9S.....	15
2.11 Receiver RadioLink AT9S.....	16
2.12 MONSTER MOTO SHIELD VNH2SP30.....	17
2.13 ลักษณะการวางตัวขั้วเหนือ-ใต้ Nd.....	18
2.14 แม่เหล็กนีโอไดเมียม.....	19
2.15 กราฟเปรียบเทียบคุณสมบัติของแม่เหล็ก.....	19
2.16 Mecanum wheel.....	21
2.17 Mecanum wheel Forward.....	22
2.18 Mecanum wheel Move Left.....	22
2.19 Mecanum wheel Sprint.....	23
2.20 Omni wheel.....	23
2.21 Omni wheel แบบสี่ล้อ.....	24
2.22 Omni สี่ล้อ wheel Forward.....	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.23 Omni ^{สี่} ล้อ wheel Move Left.....	24
2.24 Omni wheel สามล้อ Rota caster.....	25
2.25 Omni wheel สามล้อ Forward.....	25
2.26 Omni wheel สามล้อ Move Left.....	26
2.27 Omni wheel สามล้อ Sprint.....	26
2.28 ตัวอย่างแบตเตอรี่หลายชนิด.....	28
2.29 โครงสร้างภายในของแบตเตอรี่.....	28
2.30 Battery Charger Unit.....	30
3.1 แผนผังแสดงขั้นตอนการดำเนินงานวิจัย.....	31
3.2 แผนผังแสดงขั้นตอนการดำเนินงานวิจัย (ต่อ).....	32
3.3 แบบ Robot ที่ศึกษาข้อมูลโครงสร้างภายใน.....	33
3.4 ถอด Scale BB8.....	34
3.5 เทียบ Scale BB8 ด้วยหน่วย pixel.....	34
3.6 Head BB8 Scale.....	35
3.7 Shell Robot.....	35
3.8 จัดตบแต่งรอยแผล.....	36
3.9 Head - Body Mask #I.....	36
3.10 Robot mask #I (top view).....	37
3.11 Robot mask #I (side view).....	37
3.12 Robot mask # I (front view).....	38
3.13 Robot mask # II.....	38
3.14 Robot mask # III.....	39
3.15 Robot mask # IV.....	39
3.16 Circuit Monster Shield Drive.....	40
3.17 Test Drive Motor.....	40
3.18 Test Connect Receiver.....	41
3.19 Test Blink LED by RC Radio.....	41

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20 Power module.....	42
3.21 Test Drive and Battery.....	42
3.22 Stepdown.....	42
3.23 Test Controller Mask #II.....	43
3.24 Test Controller Mask #IV.....	43
3.25 Head Robot New.....	44
3.26 Magnate in Head.....	44
3.27 Damping.....	44
3.28 Magnate in Shell (top view).....	45
3.29 Test Install Head & Body.....	45
3.30 Head & Body #II.....	46
3.31 Head Ready Use.....	46
3.32 เจาะหัวกลวงลดน้ำหนัก.....	46
3.33 เปลี่ยน damping เป็นฟองน้ำ.....	47
3.34 มาร์คส่วนที่พื้นผิวขรุขระ.....	47
3.35 แก้ไขหัวน็อตที่มีปัญหา.....	47
4.1 Monster Moto VNH2SP3.....	48
4.2 Block Diagram VNH2SP3.....	49
4.3 Circuit Diagram VNH2SP3.....	49
4.4 โปรแกรมทดสอบ Motor Shield.....	50
4.5 ทดสอบวัดแรงดันและกระแสไฟฟ้า.....	50
4.6 ตัวอย่าง Waveform Operation.....	51
4.7 ชุด Adiolink-at9s-10ch.....	51
4.8 เชื่อมต่อ Arduino กับ Receiver.....	52
4.9 ทดสอบรับค่าด้วย Interrupt.....	52
4.10 วงจร Arduino & Receiver.....	53
4.11 ค่าจาก Remote ที่ Arduino รับได้.....	53

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.12 วงจร RC Low Pass.....	54
4.13 สัญญาณเมื่อผ่านวงจร RC Low Pass.....	54
4.14 Signal Analysis.....	55
4.15 Signal from Oscilloscope PWM.....	55
4.16 วงจรทดสอบด้วยหลอด LED.....	56
4.17 โปรแกรมอ่านค่าจาก Remote.....	56
4.18 โปรแกรมทดลองใช้งาน Data ที่รับเข้ามาจาก Remote.....	57
4.19 การวางแม่เหล็ก.....	57
4.20 Head robot Mask #IV.....	58
4.21 Neodymium Magnate.....	58
4.22 เตรียมพื้นผิวติดแม่เหล็ก.....	58
4.23 Neodymium Test.....	59
4.24 กราฟการทดสอบระยะห่างกับแรงดึงดูด.....	59
4.25 ความเข้ากันได้ของส่วนประกอบ.....	59
4.26 สมการการกลิ้ง.....	60
4.27 ทิศทางการเคลื่อนที่.....	60
4.28 สมการการหมุน.....	60
4.29 การเคลื่อนที่จากข้างใน.....	61
4.30 Record Voltage to SD Module.....	61
4.31 วงจรแบ่งแรงดันไฟฟ้า.....	62
4.32 การต่อวงจรแบ่งแรงดันไฟฟ้า.....	63
4.33 แอมพลิฟายเออร์แบบ Inverting LM358.....	63
4.34 โปรแกรมอ่านค่าแรงดันไฟฟ้ากระแสตรง.....	64
4.35 ค่าแรงดันไฟฟ้ากระแสตรง.....	64
4.36 วงจรการวัดกระแสโดยใช้ตัวต้านทานแบบทแยงมุม.....	65
4.37 โปรแกรมวัดกระแสโดยใช้ตัวต้านทานแบบทแยงมุม.....	65
4.38 ค่าการวัดกระแสไฟฟ้า.....	66

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.39 วงจร SD Card & Arduino.....	66
4.40 โปรแกรมทดสอบการทำงาน SD Card.....	67
4.41 ผลการทดสอบ Module SD Card.....	67
4.42 Wiring วงจร Data logger.....	68
4.43 ผลการอ่านค่า Analog ทั้ง 3 Pin จากไฟล์ใน SD Card.....	68
4.44 กราฟผลของ Volt ที่จากไฟล์ใน SD Card.....	68
4.45 ผลการอ่านค่า Amp จากไฟล์ใน SD Card.....	69
4.46 ผลการบันทึกข้อมูลและแสดงในหน้าต่าง Excel.....	69
4.47 กราฟกระแส Motor 2 ตัว ควบคุมการไต่อยู่ที่ 50%.....	70
4.48 กราฟกระแส Motor L ควบคุมการไต่อยู่ที่ 100%.....	70
4.49 กราฟกระแส Motor R ควบคุมการไต่อยู่ที่ 100%.....	71
4.50 กราฟกระแส Motor Mission Test.....	71
4.51 กราฟกระแส Motor 2 ตัว ควบคุมการไต่อยู่ที่ 100%.....	72
4.52 กราฟที่ระยะเวลาการใช้งาน.....	72
4.53 กราฟระยะเวลาการชาร์ตแบตเตอรี่.....	73
4.54 กราฟ Charge Vole ,Time.....	73
ก.1 Design Robot Concept Mask #IV (A).....	80
ก.2 Design Robot Concept Mask #IV (B).....	81
ก.3 Dimension Head Damping.....	82
ก.4 Dimension Swing Arm.....	82
ก.5 Dimension Base.....	83
ก.6 Dimension Structure.....	83
ก.7 Structure Assembly.....	84
ก.8 Assembly Model Mask #IV.....	85
ก.9 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านหน้า.....	85
ก.10 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านหลัง.....	86
ก.11 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านบน.....	86

สารบัญรูป (ต่อ)

รูปที่	หน้า
ก.12 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านข้าง.....	87
ก.13 หุ่นยนต์ภายใน Shell Mask #II มุมมองด้านหน้า.....	87
ก.14 หุ่นยนต์ภายใน Shell Mask #II มุมมองด้านข้าง.....	88
ก.15 วงจรรวมหุ่นยนต์ Shell Mask #II.....	88
ก.16 ชุดแม่เหล็กคูดภายใน Shell Mask #II.....	89
ก.17 โครงหุ่นยนต์ Mask #III ด้านหน้า.....	89
ก.18 โครงหุ่นยนต์ Mask #III ด้านข้าง.....	90
ก.19 ชุดแม่เหล็กคูดภายใน Shell Mask #III.....	90
ก.20 ชุด Damping Mask #III.....	91
ก.21 ทดสอบความยืดหยุ่น Damping.....	91
ก.22 วงจรรวม Mask # IV.....	92
ก.23 หุ่นยนต์ภายใน Shell Mask #IV.....	93
ก.24 วงจรลดแรงดัน Plugin Mask #IV.....	93
ก.25 บรรจุ Robot เข้าสู่ Shell.....	94
ก.26 ทดสอบระบบการทำงานก่อนปิด Shell.....	95
ก.27 Shell ที่ใช้ในการทำงานวิจัย.....	96
ก.28 บรรจุหุ่นยนต์และทดสอบการเคลื่อนที่.....	96
ก.29 มาร์คตำแหน่งที่ Shell ขรุขระ.....	97
ก.30 แก้ไขตำแหน่งมาร์คที่มีปัญหา.....	97
ก.31 ทดสอบการติดตั้งหัวของหุ่นยนต์.....	98
ก.32 ส่วนด้านล่างของหัวหุ่นยนต์ที่ติดแม่เหล็ก.....	98
ก.33 ออกแบบและสร้างส่วนหัวชิ้นใหม่.....	99
ก.34 ทำหัวให้กลวงเพื่อลดน้ำหนัก.....	99
ก.35 ตกแต่งส่วนหัวและประกอบหุ่นยนต์.....	100
ก.36 นำไปทดสอบและบันทึกผลการทดลอง.....	100

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของงานวิจัย

เทคโนโลยีเกี่ยวกับหุ่นยนต์ได้เข้ามามีบทบาทสำคัญกับการดำเนินชีวิตประจำวันของมนุษย์มากขึ้นเรื่อย ๆ ไม่ว่าจะเป็นสื่อวิทยาศาสตร์เทคโนโลยีเกษตรอุตสาหกรรม รวมไปถึงงานวงการภาพยนตร์ ซึ่งมีอยู่อย่างแพร่หลาย อันเป็นแรงผลักดันให้เกิดจินตนาการกับเหล่านักเรียนนักศึกษา ตลอดจนไปถึงเหล่าวิศวกรผู้ออกแบบทดลองและสร้างสรรค์สิ่งประดิษฐ์ใหม่ ๆ ดังเช่น หุ่นยนต์ทรงกลมเคลื่อนที่ด้วยการกลิ้งเป็นทรงกลม ดังที่ปรากฏในมหากาพย์ภาพยนตร์วิทยาศาสตร์เรื่อง STAR WARS VII หรือที่รู้จักกันในชื่อหุ่นยนต์ BB8 (ASTROMECH DROID) ในอนาคตจะเป็นแรงบันดาลใจในการสร้างสรรค์ผลงานกับเหล่าวิศวกรนักออกแบบในอนาคต

หุ่นยนต์นี้สามารถเพิ่มเทคโนโลยีการควบคุมที่ทำให้หุ่นยนต์สามารถ Interface กับมนุษย์ รวมไปถึงสามารถต่อรวมเข้ากับ module ระบบการทำงานต่าง ๆ หรือเพิ่มลูกเล่นเข้าไปได้ กล่าวคือหุ่นนี้ สามารถนำเอาไปประยุกต์ใช้งานได้อย่างหลากหลาย หรือแม้แต่นำไปใช้งานในแขนงอื่นๆ อาทิเช่น การนำเอาไปเป็นหุ่นยนต์นำทางคนพิการ ไก่นำทัวร์ หรืองานวิจัยที่เกี่ยวข้องอีกมากมาย แต่ทว่าต้นทุนการผลิตหุ่นยนต์ Model นี้ค่อนข้างที่จะมีราคาสูงมากจึงทำให้หุ่นยนต์นี้ยังไม่เป็นที่รู้จักในเชิงลึกซึ่งส่วนใหญ่จะรู้จักเพียงในภาพยนตร์ดังกล่าวมาข้างต้นเท่านั้น ซึ่งถ้าสามารถแก้ไขปัญหานั้นที่พบได้นักออกแบบก็จะสามารถพัฒนาเทคโนโลยีด้านหุ่นยนต์ให้ทัดเทียมกับเทคโนโลยีที่มีราคาสูงในต่างประเทศได้

กล่าวโดยรวมคือผู้พัฒนาจะสามารถเพิ่มนวัตกรรมที่ต้องการฝังในตัวหุ่นยนต์ได้อย่างไม่สิ้นสุดด้วยเหตุผลที่ได้กล่าวมาข้างต้น งานวิจัยนี้จึงมีวัตถุประสงค์เพื่อการออกแบบและสร้างหุ่นยนต์ลักษณะทรงกลมสามารถแสดงอิริยาบถได้ เคลื่อนที่ด้วยการกลิ้งม้วนตัวควบคุมผ่านทางรีโมทคอนโทรล โดยหุ่นยนต์ต้นแบบนี้เบื้องต้นจะสร้างมาเพื่อนำไปใช้ประโยชน์ในมหาวิทยาลัย อาทิเช่น ใช้สำหรับเป็นสื่อการเรียนการสอน หุ่นยนต์นำทางค้นหาหนังสือในห้องสมุด หุ่นยนต์แนะนำมหาวิทยาลัย ใช้ต้อนรับแขกคนสำคัญ ตลอดจนไปถึงเพื่อให้นักศึกษารุ่นใหม่ได้นำไปศึกษาพัฒนาประสบการณ์และสร้างองค์ความรู้ โดยจะทำให้เข้าใจถึงหลักการทำงานการออกแบบซึ่งหุ่นยนต์ต้นแบบจะสามารถนำไปพัฒนาต่อยอดในอนาคตข้างหน้าได้เป็นอย่างดี อันเป็นการช่วยเพิ่มขีดความสามารถทางด้านวิศวกรรมและเทคโนโลยีอย่างไร้ขีดจำกัด เป็นการพัฒนาประเทศไทยให้ทัดเทียมกับนานาประเทศและทั่วโลกอีกต่อไป

1.2 วัตถุประสงค์ของงานวิจัย

- 1.2.1 ออกแบบและสร้างหุ่นยนต์ต้นแบบ SPHERICAL ROBOT
- 1.2.2 เพื่อศึกษาการออกแบบ Controller ที่ใช้ควบคุม SPHERICAL ROBOT
- 1.2.3 เพื่อศึกษาและจำลองการทำงานระบบการเคลื่อนที่ภายในทรงกลม

1.3 ขอบเขตของงานวิจัย

- 1.3.1 สร้างแบบจำลองของ SPHERICAL ROBOT
- 1.3.2 ออกแบบโครงสร้างหลักทั้งหุ่นยนต์ภายใน และเปลือกหุ้มภายนอก
- 1.3.3 ออกแบบระบบกลไกการยึดส่วนหัวกับส่วนตัวของหุ่นเข้าด้วยกัน
- 1.3.4 ออกแบบระบบคอนโทรลเลอร์โดยใช้ Remote RC [A9] + Arduino

1.4 วิธีดำเนินงานวิจัย

- 1.4.1 ศึกษาค้นคว้าองค์ความรู้ ทฤษฎี และเอกสารที่เกี่ยวข้อง
- 1.4.2 วิเคราะห์หลักการทางจลศาสตร์เพื่อใช้ในการ
- 1.4.3 ออกแบบโครงสร้างหลักของ SPHERICAL ROBOT
- 1.4.4 ออกแบบโครงสร้างภายนอก (เปลือกหุ้มหุ่นยนต์)
- 1.4.5 ออกแบบระบบควบคุมการทำงานของคอนโทรลเลอร์
- 1.4.6 จัดหาวัสดุและอุปกรณ์ในการจัดสร้าง
- 1.4.7 ดำเนินการสร้างต้นแบบ
- 1.4.8 ทดสอบและทำการปรับปรุงแก้ไข
- 1.4.9 ทำการวิเคราะห์ข้อมูลและสรุปผลที่ได้
- 1.4.10 เขียนรายงานฉบับสมบูรณ์

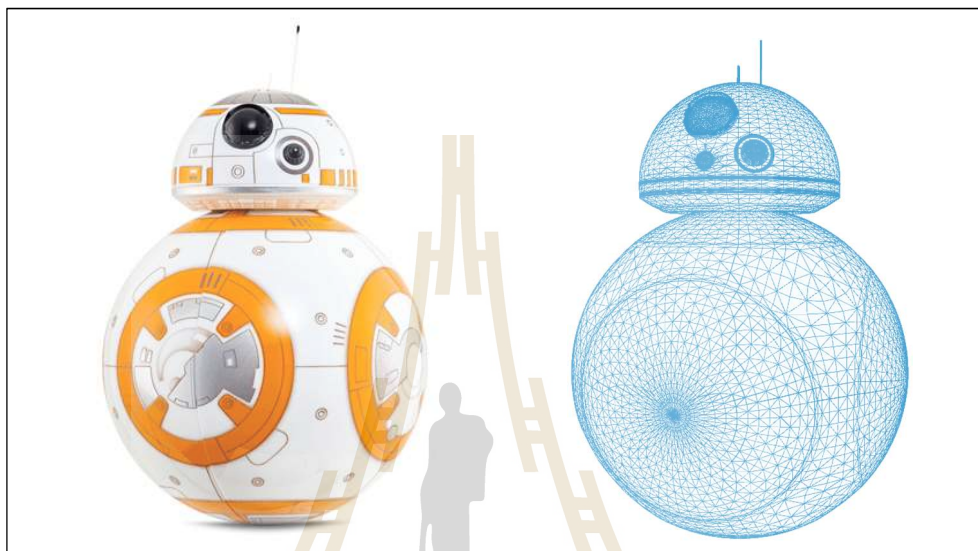
1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 ได้เรียนรู้วิธีการออกแบบและสร้างหุ่นยนต์ทรงกลม
- 1.5.2 ได้รู้วิธีการเขียนโปรแกรมและสร้าง Controller สำหรับควบคุมหุ่นยนต์
- 1.5.3 ได้แนวทางการพัฒนาหุ่นยนต์เพื่อยกระดับขีดความสามารถจากหุ่นยนต์ต้นแบบระดับห้องปฏิบัติการไปสู่ระดับที่สามารถใช้งานจริงเป็น Mobile Robot ได้

1.6 สถานที่ทำงานวิจัย

1.6.1 อาคารเครื่องมือวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีสุรนารี

1.7 ต้นแบบหุ่นยนต์



รูปที่ 1.1 โครงสร้างเปลือกนอกของหุ่นยนต์ BB8



รูปที่ 1.2 แบบจำลองโครงสร้างภายในหุ่นยนต์

บทที่ 2

ปริทรรศน์วรรณกรรมและทฤษฎีที่เกี่ยวข้อง

2.1 กล่าวนำ

ในส่วนแรกจะเป็นส่วนที่ผู้วิจัยได้ทำการค้นคว้าเอกสาร วิทยานิพนธ์ต่าง ๆ ที่เกี่ยวข้องกับงานวิจัยที่ได้ดำเนินการอยู่ พบว่ามีบทความและเอกสารต่างๆที่เกี่ยวข้อง ผู้วิจัยจึงเลือกนำเสนองานวิจัยที่เกี่ยวข้อง และสอดคล้องกับงานวิจัย ที่ได้ดำเนินการอยู่นี้เท่านั้น

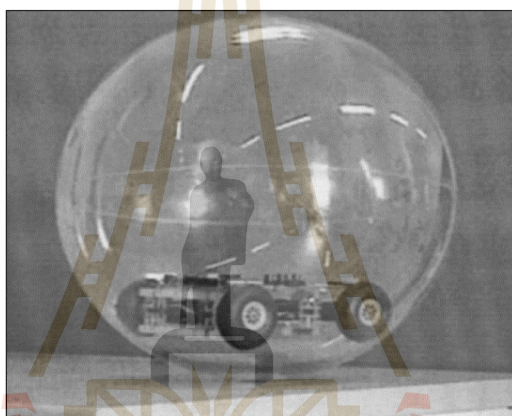
หุ่นยนต์ทรงกลมเป็นข้อมูลการวิจัยที่เกิดขึ้นอันเนื่องมาจากลักษณะของหุ่นยนต์ที่อยู่ภายในของลูกบอลเคลื่อนที่ในลักษณะ Holonomic ที่สภาพแวดล้อมภายในปิดสนิทและสามารถดึงออกจากการชนได้อย่างง่ายดาย เป็นงานวิจัยกลุ่มที่ได้เริ่มที่จะพัฒนาวิธีการที่เป็นเอกลักษณ์ของการขับเคลื่อน โดยดึงเอาแต่ละเอกลักษณ์ทางวิศวกรรมที่โดดเด่นมาใช้ในการประยุกต์ สิ่งสำคัญ ที่ต้องทำการควบคุมคือ น้ำหนัก ทิศทาง ความเร็ว ให้สัมพันธ์กัน เพื่อความถูกต้องในการควบคุม ฯลฯ หุ่นยนต์ทรงกลมในช่วงต้นการดำเนินการวิจัย มีลักษณะคล้ายกับลูกบอลหนูแฮมสเตอร์และมีแรงบิดจำกัด การสูญเสียพลังงานสูงเป็นผลอันเนื่องมาจากแรงเสียดทานภายใน นักวิจัยได้เริ่มพัฒนาและหาแนวคิดต่างๆ ตลอดจนการนำหลักการพื้นฐานทางฟิสิกส์มาช่วยออกแบบ เพื่อที่จะทำให้สามารถควบคุมหุ่นยนต์ตระกูลนี้ได้ ซึ่งในที่นี้จะขอกกล่าวถึงภาพรวมและทิศทางการวิจัยในปัจจุบันที่กลุ่มต่าง ๆ พยายามออกแบบที่เกี่ยวข้องกับงานวิจัย

เจ.เจ. แอบริมส์ (2015) [1] บีบีเอท (BB8) ผู้สร้างและออกแบบตัวละครหุ่นยนต์ ประเภทดรอยด์ในสตาร์วอร์ส ปรากฏตัวครั้งแรกในภาพยนตร์ สตาร์วอร์ส อูบิตการณัแห่งพลัง บีบีเอทได้ปรากฏครั้งแรกในตัวอย่าง 90 วินาทีของ อูบิตการณัแห่งพลัง เผยแพร่โดย ลูคัสฟิล์ม เมื่อวันที่ 28 พฤศจิกายน ค.ศ. 2014 โดยเป็นหุ่นยนต์ที่ใช้นักแสดงเซด หรือสวมชุดในภาพยนตร์



รูปที่ 2.1 BB8 – Master [1]

Abhilash Pandya (2012) [2] ได้ออกแบบสิ่งที่เรียกกันทั่วไปว่าการออกแบบ Hamster Ball ที่ได้ชื่อนี้เพราะมันเลียนแบบหนูแฮมสเตอร์ใน ลูกบอลของเล่น หุ่นยนต์ขนาดเล็กมีล้อขับเคลื่อนภายในลูกบอลซึ่งผู้ออกแบบได้เปลี่ยนมาเป็น รถควบคุมระบบไกลขนาดเล็ก ซึ่งมีน้ำหนักมากพอที่จะขับเคลื่อนหุ่นยนต์ที่ถูกหุ้มด้วยเปลือกได้ ส่วนของหัวหุ่นยนต์ภายในเมื่อมีการเปลี่ยนแปลงขณะเคลื่อนที่จะทำให้เปลือกนอกเปลี่ยนทิศทางการเดินทาง ขานพาหนะที่ใช้ขับเคลื่อนเป็นลักษณะล้อจะสร้างเส้นโค้งวิถีการเคลื่อนที่ที่แตกต่าง นอกจากนี้การออกแบบค่อนข้างง่ายที่จะจำลองลักษณะการควบคุม กล่าวคือเป็นลักษณะที่ควบคุมง่ายถูกต้องและเป็นธรรมชาติเช่นเดียวกับประเภทของรถควบคุมระยะไกลพื้นฐานใด ๆ ซึ่งค่อนข้างง่าย

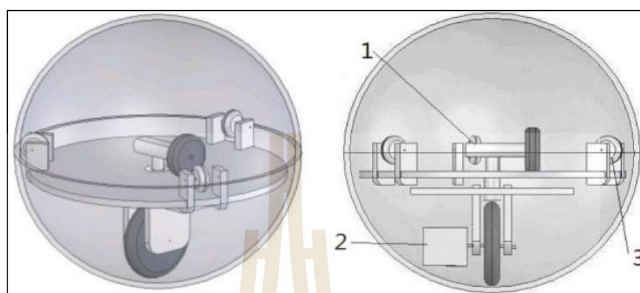


รูปที่ 2.2 Hamster Ball [2]

หนึ่งในอุปสรรคที่สำคัญคืออาการสั่นไถลของหุ่นยนต์ภายใน หรือระบบขับเคลื่อนกลไกจะเกิดขึ้นระหว่างการขับเคลื่อน ซึ่งทำให้เกิดการสูญเสียพลังงานและเกิดภาวะแทรกซ้อนระหว่างการควบคุมเป็นผลเนื่องมาจากแรงเสียดทาน เป็นหลุมพรางของการออกแบบอีกประการหนึ่งของการทำงานเสมือนหุ่นยนต์วิ่งอยู่บนอากาศไม่สามารถจุดเปลือกหุ้มให้เคลื่อนที่ไปข้างหน้าได้ ซึ่งถือเป็นปัญหาใหญ่ในการออกแบบหุ่นยนต์ลักษณะนี้

Richard Chase (2012) [3] ได้ออกแบบที่รวมเอาหลักการแบบจำลองวงล้อหนูแฮมสเตอร์ Universal Wheel มาประยุกต์ โดยโมเดลนี้นำเอาแนวคิดที่เป็นระบบกลไกการไถลที่ภายในสามารถหมุนได้อย่างอิสระสังเกตได้จากเพทที่ยึดล้อ ระบบนี้ใช้ DC Motor 2 ตัวควบคุมหุ่นยนต์ประกอบด้วย มอเตอร์ที่ใช้ควบคุมทิศทาง และ มอเตอร์ที่ใช้ควบคุมความเร็ว ระบบนี้จะช่วยให้หุ่นยนต์เปลี่ยนทิศทางของ Holonomic ได้ง่ายกว่าหุ่นยนต์ที่อธิบายไว้ก่อนหน้านี้ [2] กล่าวคือค่อนข้างง่ายต่อการควบคุมโมเมนตัมของหุ่นยนต์เพราะน้ำหนักถูกถ่ายลงจุดๆเดียว

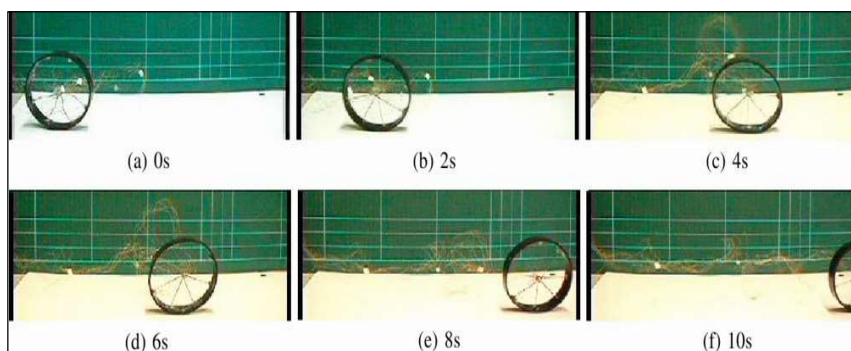
โดยปัญหาของการออกแบบลักษณะนี้คือ ไม่สามารถขับเคลื่อนหุ่นยนต์ในภูมิประเทศที่เป็นหลุมเป็นบ่อได้ สามารถเดินทางในน้ำและเนินทรายเล็กๆ ได้ ซึ่ง โมเดลนี้ มีการสูญเสียพลังงานอันเกิดขึ้นจากแรงเสียดทานของล้อฟองน้ำซึ่งถูกติดตั้งไว้ด้านข้างของหุ่นยนต์ ทำให้เมื่อลงเนินเขา หุ่นยนต์มีอาการม้วนตัว และไม่สามารถควบคุมทิศทางเมื่ออยู่ในกรณีพื้นที่ลาดเอียง และการบังคับ



รูปที่ 2.3 Universal Wheel [3]

ข้อดีของการออกแบบนี้ ลักษณะโดยทั่วไปคล้ายพวงมาลัยและการขับเคลื่อนด้วยกลไกเป็นอิสระต่อกัน โดยหุ่นยนต์จะสามารถย้ายไปในทิศทางใดๆ ได้โดยไม่ต้องคำนึงถึงการวางแผน Holonomic โดยที่ 1. Motor บังคับเลี้ยว 2. Motor ขับเคลื่อน และ 3. ฟองน้ำ

ซีเกีย เจจ (2014) [4] ได้ออกแบบและนำเสนอการเปลี่ยนรูปของเปลือกนอก Shape Memory Alloys ในลักษณะที่ช่วยให้หุ่นยนต์เคลื่อนที่ด้วยวิธีที่ไม่ซ้ำกันของการยึดหดของ Alloys เมื่อมีการจ่ายกระแสไปเขาไปเลี้ยงขดลวด ผลลัพธ์ที่ได้คือทำให้เกิดการสปริงผนวกกับคุณสมบัติการยึดหยุ่นของเปลือกนอกหลักการในการสร้างคือ เปลือกนอกมีการวาง โรเตอร์ใน โครงสร้าง หุ่นยนต์และควบคุมการเคลื่อนที่โดยการจ่ายไฟเข้าไปในแต่ละขดลวดที่วางเอาไว้ที่ละจุดสลับกัน โดยรูปภาพที่ 2.4 แสดงให้เห็นถึงการเคลื่อนไหวทั้งหมด โดยวัสดุที่ใช้เป็น โลหะผสม SMA ซึ่งคุณสมบัติคือเมื่อจ่ายไฟจะหดตัว และเมื่อปลดไฟออกจะขยายกลับเข้าสู่สภาวะเดิมทันที ซึ่งการเคลื่อนที่ของหุ่นยนต์อาศัยหลักการนี้ทำให้มันสามารถเคลื่อนที่ได้ เมื่อควบคุมได้อย่างถูกต้อง หุ่นยนต์จะสามารถเคลื่อนไหวได้อย่างราบรื่น ข้อเสียหลักของการออกแบบลักษณะนี้คือ ใช้พลังงานสูงมาก และขึ้นรูปได้ค่อนข้างยาก



รูปที่ 2.4 Shape Memory Alloys [4]

ยีน จุง ลี (2015) [5] ได้นำเสนอหุ่นยนต์รูปแบบหุ่นยนต์ทรงกลมที่ใช้การขับเคลื่อนแบบใหม่ ด้วยกลไกการขับเคลื่อนด้วยลม โดยใช้พัดลม New Spherical Robot With Wind-Driven Driving Mechanism สำหรับการเคลื่อนที่ทำได้โดย หุ่นยนต์มีใบพัดถูกติดตั้งอยู่ ณ ศูนย์กลางของทรงกลม และหุ่นยนต์จะเคลื่อนที่ตามทิศทางที่พัดลมเป่าน้ำหลังได้ เปลือกนอกเป็นกรอบเปิด ทำด้วยแท่งคาร์บอนสปริง แข็งแรงไม่เสียรูปง่าย มีขาที่สามารถพับเก็บเมื่อหยุดใช้งาน หุ่นยนต์สำหรับการหมุนและควบคุมสมดุลอยู่ที่ลูกตุ้มที่อยู่ใต้กรอบ Ducted-Fan การออกแบบนี้ กลไกที่ใช้ลมในการขับเคลื่อน และต้องมีแรงยกมากเพียงพอสำหรับการขับเคลื่อน บนพื้นที่ราบเรียบขรุขระลาดชันได้ การทดลองยืนยันประสิทธิภาพของหุ่นยนต์สามารถทำงานได้ในสภาวะปิด ไม่มีลม



รูปที่ 2.5 New Spherical robot with wind-driven [5]

2.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์คืออุปกรณ์ที่สามารถสร้างระบบควบคุมได้ โดยอุปกรณ์มีขนาดเล็ก และเป็นอุปกรณ์ประเภทสารกึ่งตัวนำที่มีการรวมเอาฟังก์ชันการทำงานต่างๆไว้ในตัวเอง ลักษณะคล้ายกับคอมพิวเตอร์ ซึ่งในที่นี้หมายถึงอุปกรณ์ภายใน ที่ประกอบด้วย หน่วยประมวลผลกลาง พอร์ตในการเชื่อมต่อแบบต่างๆ

2.2.1 ส่วนประกอบทั่วไปของไมโครคอนโทรลเลอร์

1. หน่วยประมวลผลกลาง (Control Processing Unit)
2. หน่วยความจำ ประกอบไปด้วย RAM (Random Access Memory)
EEPROM/EPROM/PROM/ROM
3. หน่วยรับและแสดงผลข้อมูล (Input / Output) ซึ่งมีพอร์ตขยายแบบขนาน (Parallel) และอนุกรม (Serial)
4. ตัวนับเวลา (Timer)
5. หน่วยควบคุมการอินเตอร์รัปต์ (Interrupt Controller)

ส่วนประกอบเหล่านี้เป็นเพียงส่วนประกอบพื้นฐานของไมโครคอนโทรลเลอร์ นอกจากนี้ยังมีส่วนประกอบอื่นเพื่อเพิ่มเติมความสามารถ เช่น

- ADC (Analog to Digital) ส่วนภาครับสัญญาณอนาล็อกแปลงไปเป็นดิจิทัล
- DAC (Digital to Analog) ส่วนภาคส่งสัญญาณดิจิทัลแปลงไปเป็นอนาล็อก
- I2C (Inter Integrate Circuit Bus) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสารระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอกซึ่งถูกพัฒนาขึ้น โดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้นคือ serial data (SDA) , serial clock (SCL)
- SPI (Serial Peripheral Interface) การเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส (Synchronize) มีสัญญาณนาฬิกาเข้ามาเกี่ยวข้องกับระหว่าง ไมโครคอนโทรลเลอร์ จะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI
- PWM (Pulse Width Modulation) การสร้างสัญญาณพัลส์แบบสแควร์เวฟ ที่สามารถปรับเปลี่ยนความถี่และ Duty Cycle ได้เพื่อนำไปควบคุมอุปกรณ์ต่างๆ เช่น มอเตอร์ เซอร์โวมอเตอร์
- UART (Universal Asynchronous Receiver Transmitter) หน้าที่รับส่งข้อมูลแบบอะซิงโครนัสสำหรับมาตรฐานการรับส่งข้อมูลแบบ RS-232

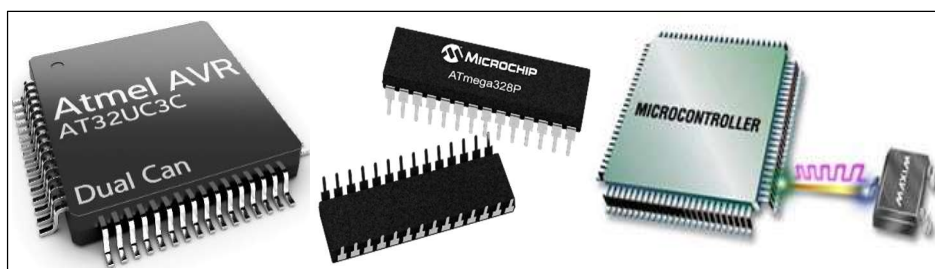
2.2.2 ประเภทของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ มีด้วยกันหลายประเภทแบ่งตามสถาปัตยกรรม การผลิต และกระบวนการทำงานระบบการประมวลผล ที่มีใช้ในปัจุบัน ดังนี้

1. ไมโครคอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip)
2. ไมโครคอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Philips)
3. ไมโครคอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโครคอนโทรลเลอร์ตระกูล Arduino (บริษัทผู้ผลิต Atmel) เป็นการนำเอา AVR มาเขียน firmware Bootloader คำสั่งใช้งานใหม่ ข้อแตกต่างของบอร์ด Arduino กับบอร์ดทดลองทั่วไปคือเป็น Open source หมายถึงการเปิดเผยวิธีการสร้างทั้งในส่วนของ Hardware ไม่ว่าจะเป็นวงจรต่างๆ ภายในบอร์ด และตัว Tool Software ที่เป็นเครื่องมือใช้ในการเขียนคำสั่งต่างๆ ลงบอร์ด ส่งผลให้นักวิจัยมีสิทธิ์ดาวน์โหลดโปรแกรมต่างๆ ที่เกี่ยวกับการพัฒนา Arduino ไปใช้ฟรีและสามารถสร้างวงจรจากต้นแบบใช้ฟรีได้เช่นกัน ข้อดีคือเป็นการสร้างมาตรฐานในการพัฒนาจะเป็นไปในทิศทางเดียวกันเนื่องจากนักพัฒนาทั่วโลกนิยมใช้งานบอร์ดชนิดนี้

5. ไมโครคอนโทรลเลอร์ตระกูล ARM7, ARM9
(บริษัทผู้ผลิต Atmel, Phillips, Analog Device, Sumsung, STMicroelectronics)
6. ไมโครคอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
7. ไมโครคอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
8. ไมโครคอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instrument)
9. ไมโครคอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
10. ไมโครคอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
11. ไมโครคอนโทรลเลอร์ตระกูล (RABBIT RABBIT - SEMICONDUCTOR)
12. ไมโครคอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

(อ้างอิงจาก : ธนภัทร พรหมวัฒน์กักดี, 2008)



รูปที่ 2.6 ไมโครคอนโทรลเลอร์

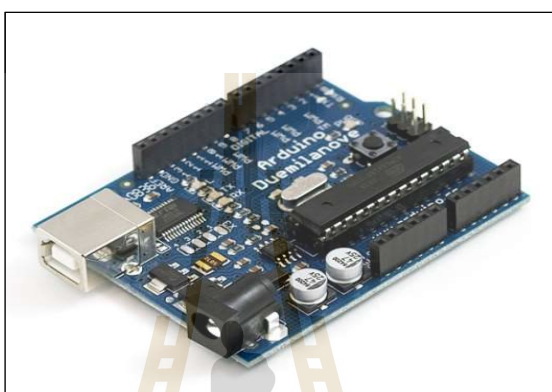
2.3 อาดูโน (Arduino)

Arduino เป็นภาษาอิตาลี โดยเป็นโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR แบบ Open Source ที่ได้รับการปรับปรุงมาจากโครงการพัฒนา Open Source ของ AVR อีกโครงการหนึ่ง โดยใช้ชื่อว่า Wiring แต่เนื่องจากโครงการ Wiring เลือกใช้ AVR เบอร์ ATmega128 เป็นชิพที่มีตัวถังแบบ SMD จึงทำให้อุปสรรคสำคัญ สำหรับผู้ที่เริ่มต้นใช้ ในการสร้างบอร์ดและการต่อวงจรที่มีขนาดใหญ่เกินความจำเป็น สำหรับผู้ที่เริ่มต้น จึงไม่ค่อยเป็นที่นิยม แต่หลังจากที่ Arduino ได้นำ Source code ของ Wiring มาทำการปรับปรุงและพัฒนาใหม่ ทำให้ มีขนาดเล็กกลง เช่น Mega8 และ Mega168 จึงทำให้วงจรของบอร์ดมีขนาดเล็กกลงกว่า Wiring มากและยังใช้อุปกรณ์น้อยชิ้น ทำให้ง่ายต่อการต่อวงจร ประหยัดต้นทุนในการสร้างบอร์ด ด้วยเหตุผลนี้ทำให้ Arduino ได้รับความนิยมนอกจากผู้คนทั่วโลกเป็นอย่างมากในระยะเวลาอันรวดเร็ว

Arduino มีจุดเด่นในเรื่องความง่ายในการเรียนรู้และการใช้งาน เนื่องจากมีคำสั่งที่สนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ถึงแม้ว่ารูปแบบการใช้งานจะคล้ายกับไมโครคอนโทรลเลอร์อย่าง Basic Stamp ของ Parallax, BX-24 ของ Net medias และ Handy Board ของ MIT แต่มีจุดเด่นกว่าของรายอื่น ๆ หลายอย่าง เป็นต้นว่า

- ราคาไม่แพงเนื่องจากเป็น Open Source หมายถึง โปรแกรมคอมพิวเตอร์ที่มีลักษณะเข้าถึงได้โดยสาธารณะ เพื่อใช้หรือเปลี่ยนแปลงแก้ไขจากต้นฉบับ โค้ดโอเพนซอร์ซ ถูกสร้างมา เพื่อให้ให้นักเขียนโปรแกรมสามารถร่วมมือกันพัฒนาซอร์ซโค้ด และแบ่งปันการเปลี่ยนแปลงกับเหล่านักพัฒนา ซึ่งโค้ดที่ถูกเผยแพร่อยู่ภายใต้เงื่อนไขของสัญญาอนุญาตซอฟต์แวร์ คนอื่นสามารถดาวน์โหลด เผยแพร่ และสร้างวงจรขึ้นมาใช้ได้เอง
- โปรแกรมที่ใช้พัฒนาของ Arduino IDE รองรับระบบปฏิบัติการทำงานทั้ง Windows Linux , Macintosh OSX รวมถึง Android บางรุ่นก็สามารถเชื่อมต่อเขียนโปรแกรมได้
- มีรูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานจริง ๆ ที่มีความซับซ้อนได้ และยังสามารถสร้างคำสั่งและ Library ใหม่ ๆ ขึ้นมาใช้งานได้เอง
- มีการเปิดเผยวงจร ทั้งหมดทำให้สามารถนำไปพัฒนาต่อยอดเพิ่มเติมได้ตามความต้องการทั้ง Hardware และ Software

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์โดยใช้ AVR ขนาดเล็กมาเป็นตัวประมวลผล และสั่งงาน เหมาะกับการนำไปประยุกต์ในการควบคุม INPUT และ OUTPUT ต่าง ๆ ได้มากมาย ทั้งเป็นการเชื่อมต่อในรูปแบบอิสระเดียวหรือการเชื่อมต่อร่วมกับอุปกรณ์อื่นยกตัวอย่าง เช่น เครื่องคอมพิวเตอร์ PC หรือการเชื่อมต่อแบบ Digital และแบบ Analog เช่นการรับค่าจากสวิตช์ หรืออุปกรณ์ตรวจจับ (Sensor) แบบต่าง ๆ รวมถึงการควบคุมอุปกรณ์แบบ OUTPUT



รูปที่ 2.7 Arduino Duemilanove

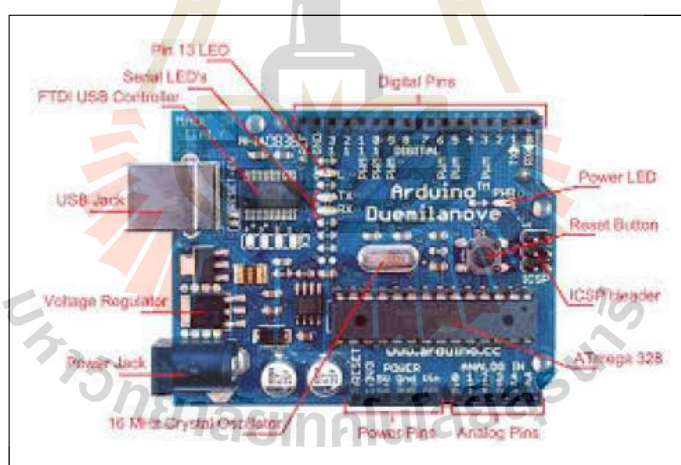
การเปรียบเทียบระหว่างภาษาซีกับ Arduino โดยในภาษาของ Arduino จะใช้ภาษา C++ ซึ่งเป็นรูปแบบโปรแกรมภาษาซีประยุกต์ ที่มีโครงสร้างของตัวภาษาโดยใกล้เคียงกับกับภาษาซีมาตรฐาน ANSI-C และได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อใช้ในการลดความยุ่งยากในการเขียนโปรแกรม โดยที่สามารถเขียนโปรแกรมได้ง่ายขึ้นและสะดวกมากยิ่งขึ้นกว่าการเขียนภาษาซีแบบมาตรฐาน ANSI-C โดยตรงภาษาซี ซึ่งมีความได้เปรียบและมีความอ่อนตัวในการใช้งาน เหนือกว่าการใช้งานเหนือกว่าภาษาอื่นๆ

กล่าวคือ ภาษาซี สามารถนำไปประยุกต์ใช้งานในระบบฮาร์ดแวร์ที่มีความแตกต่างกันได้หลากหลาย โดยผู้ใช้เพียงแต่เลือกใช้ตัวแปลคำสั่ง (C-Compiler) ให้ตรงกับระบบฮาร์ดแวร์ ที่ใช้งาน ส่วนรูปแบบการเขียนการเขียนจะเป็นมาตรฐานเดียวกัน แต่รูปแบบภาษาซีที่มีรูปแบบที่ใช้งานง่าย แต่มีข้อกำหนดในการใช้งานหรือ Syntax แต่ไม่มีฟังก์ชันสำเร็จรูป (Built-in Function) ใดๆ รวมอยู่ในตัวภาษาด้วย โดยส่วนที่เป็นฟังก์ชันการใช้งานต่าง ๆ เช่นการดำเนินการเกี่ยวกับ Input / Output การจองหน่วยความจำ (Memory Allocation) เป็นหน้าที่ของผู้ใช้ที่จะต้องสร้างขึ้นมาเอง หรือในบางครั้งจะต้องใช้วิธีการเรียกใช้ฟังก์ชันที่ ผู้ผลิตตัวแปลคำสั่ง (C-Compiler) ซึ่งได้สร้างเตรียมไว้ในรูปแบบคำสั่ง ในรูปแบบของ Library Function

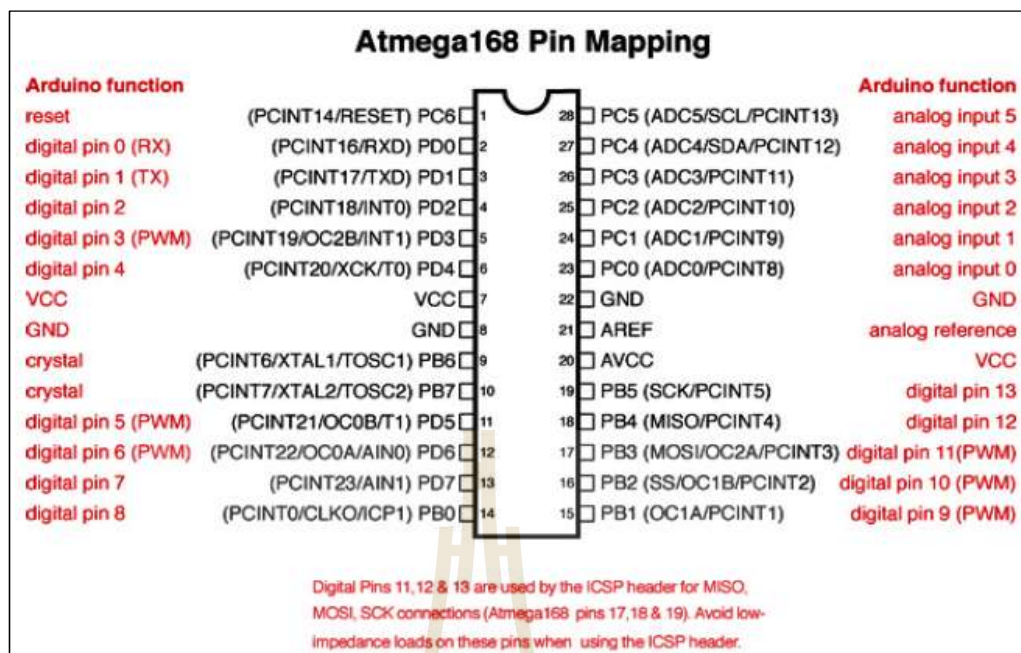
โดยคำสั่งหรือฟังก์ชันนี้เองเป็นสิ่งที่ทำให้ภาษาซีมีความแตกต่าง จนบางครั้งทำให้ผู้ใช้ไม่รู้ว่่าอันไหนคำสั่งมาตรฐาน อันไหนผู้ใช้สร้างขึ้น สำหรับการเขียนโปรแกรม Arduino นั้นจะใช้ภาษา C++ ซึ่งเป็นรูปแบบภาษาซีประยุกต์แบบหนึ่งแต่มีโครงสร้างการทำงานของตัวภาษาโดยรวมจะคล้ายกับภาษาซีมาตรฐาน ANSI-C เพียงแต่มีการปรับปรุงในส่วนความยุ่งยากให้น้อยลงเพื่อสามารถให้ผู้ใช้สามารถใช้งานและเขียนโปรแกรมได้ง่ายขึ้นสะดวกกว่าภาษาซีแบบมาตรฐาน

ในการต่ออุปกรณ์เสริมต่างๆ คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ด หรือเพื่อความสะดวกและยังสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ เช่น XBeeShield, Relay Shield, Wireless Shield, GPRS Shield

Arduino Duemilaove บอร์ดรุ่นใหญ่ในของตระกูล Arduino มีคุณสมบัติ หน่วยความจำแฟลช 16 KB แรม 1 KB ใช้ไฟเลี้ยง 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input / Output 14 ขา (which 6 provide PWM output) มี Analog Input 6 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบน Arduino IDE ผ่าน USB เหมาะสำหรับผู้พัฒนาไมโครคอนโทรลเลอร์ที่ต้องการ บอร์ด Arduino ที่มีหน่วยความจำและขาสัญญาณต่างๆ ให้ต่อใช้งานได้อย่างหลากหลาย



รูปที่ 2.8 หน่วยความจำและขาสัญญาณของ Arduino ATmega 168



รูปที่ 2.9 ตำแหน่งพอร์ตของ Arduino ATmega 168

ตารางที่ 2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ Arduino ATmega 168

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6 Port
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB
SRAM	1 KB
EEPROM	512 Bsyte
Clock Speed	16 Hz

2.4 รีโมทคอนโทรลเลอร์ (Radio Control)

รีโมทคอนโทรล (remote control) คือเครื่องมืออิเล็กทรอนิกส์ชนิดหนึ่ง ใช้สำหรับควบคุมการดำเนินการของสิ่งประดิษฐ์หรือเครื่องจักรต่างๆ โดยเฉพาะเครื่องใช้ไฟฟ้าภายในบ้านเช่น โทรทัศน์ เครื่องเสียง เครื่องเล่นดีวีดี จากระยะไกล โดยไม่ใช้สายไฟ มีขนาดเหมาะสม มีปุ่มฟังก์ชัน รีโมทคอนโทรล เป็นการเรียกย่อมาจาก รีโมทคอนโทรลเลอร์ (remote controller) และสามารถเรียกย่อลงได้อีกเหลือเพียง รีโมท แทนได้ว่า เครื่องควบคุมระยะไกล หรืออุปกรณ์ควบคุมระยะไกล รีโมทคอนโทรล จะสามารถสั่งงานได้ ต้องประกอบด้วย 2 สิ่ง คือ 1. สัญญาณ เป็นรูปแบบของคลื่นความถี่ ซึ่งจะนำออกได้ต้องมีตัวคลื่นพานำออกไป 2.4 ภาคส่งและภาครับสัญญาณ ใช้สำหรับการถอดหรือรับรหัสที่ถูกส่งเข้ามาเพื่อใช้ควบคุมอุปกรณ์ต่างๆ

2.4.1 รีโมท RadioLink AT9-S



รูปที่ 2.10 วิทยุบังคับ RadioLink AT9S

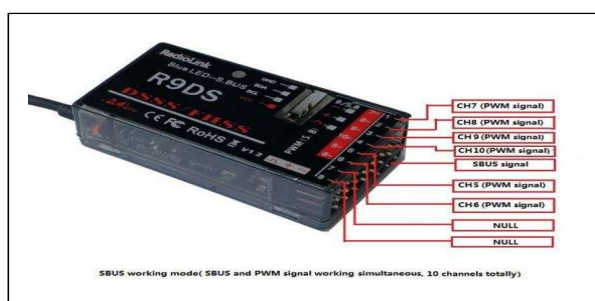
คุณสมบัติเบื้องต้น

- ใช้ได้ทั้งเครื่องบิน เครื่องร่อน มัลติโรเตอร์ เฮลิคอปเตอร์ 120 degree และ 90 degree
- ระบบ DSSS spread spectrum มีความเสถียรของสัญญาณสูง
- วิทยุจอบเป็นสีหน้าจอขนาด 2.8 นิ้วระบบ 2.4 GHz ใช้ Firmware V 1.1.8
- มีระบบแจ้งเตือนผู้เล่นแบบ Vibration alarm สามารถตั้งเสียงเตือนและสั่นได้
- สามารถตั้งเวลาถอยหลังมีเสียงเตือนเมื่อครบเวลาที่ตั้งไว้

- มีระบบ Telemetry สามารถมอนิเตอร์รับข้อมูลต่างๆ จากระยะไกลได้ เช่น Sensor แรงดันไฟของแบตเตอรี่, ความเร็วรอบมอเตอร์, GPS บนหน้าจอวิทยุได้
- ไฟเลี้ยง 7.4 V. ถึง 18 V. มี Anti-reverse ป้องกันการเสียบแบตเตอรี่ผิดขั้ว
- มีช่อง USB สามารถ update ข้อมูลล่าสุดจากทางผู้ผลิตแบบ Online ได้
- ระยะส่งบนพื้นประมาณ 900 เมตร บนอากาศประมาณ 1.5 กิโลเมตร

Features : Receiver RadioLink AT9S

- 1). Size : 183*193*100 mm
- 2). Weight : 0.88 kg
- 3). Frequency : 2.4 GHz ISM band (2400 MHz to 2485 MHz)
- 4). Modulation mode : QPSK
- 5). Channel bandwidth : 5.0 MHz
- 6). Spread spectrum : DSSS
- 7). Adjacent channel rejection : > 38 dbm
- 8). Transmitter power : < 100 mW (PCB testing), < 20 dbm
- 9). Operating Current : < 105 mA
- 10). Operating Voltage : 7.4 ~ 15V
- 11). Control distance : more than 900 meters ground, 1.5 kms air
- 12). Channel : 9 channels, 5~9 channels are customizable
- 13). Simulator model: under the simulator model the transmitter action turn off
- 14). Screen : 2.8 inches 16 colorful screen , 240*320 pixels
- 15). Compatible model : Include all 120 degree and 90 degree



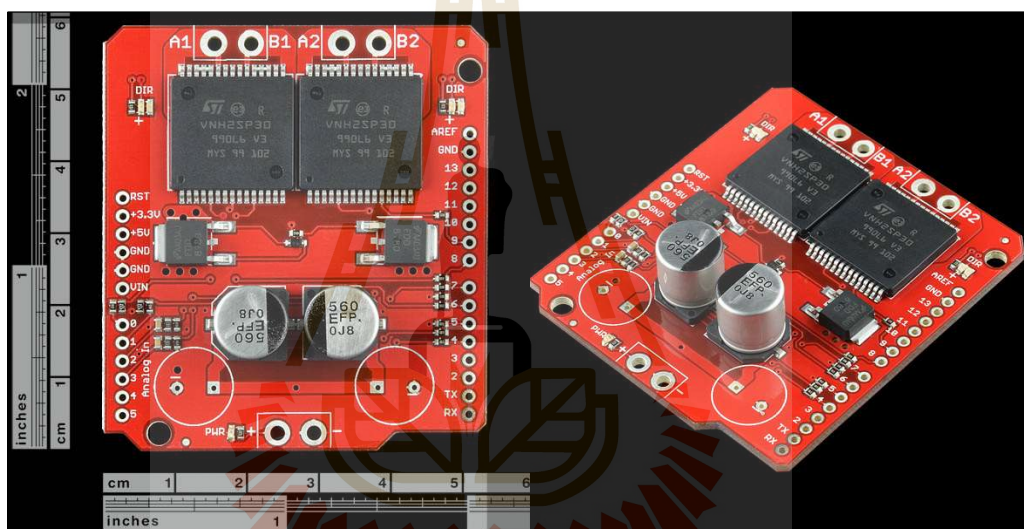
รูปที่ 2.11 Receiver RadioLink AT9S

2.5 ตัวขับมอเตอร์ (Driver Motor)

สำหรับหุ่นยนต์ (Robot) สิ่งที่เป็นพลังขับเคลื่อนหลักให้กับกลไกต่างๆคือมอเตอร์ ซึ่งมอเตอร์ ต้องการการควบคุมจากชุดไดร์เวอร์มอเตอร์ (Motor Driver) เพื่อที่ ให้พลังงาน รวมถึงการกำหนด ทิศทางการหมุน ความเร็วรอบ ของชุดมอเตอร์ ฯลฯ

2.5.1 MONSTER MOTO SHIELD VN2SP30

ตัวขับมอเตอร์ MONSTER MOTO SHIELD VN2SP30 กระแสสูงสุด 30 A ทำงานต่อเนื่องที่ 14 A ความละเอียดสูง โมดูลสำหรับ Arduino ใช้ชิพไอซีเบอร์ VN2SP30 ขณะใช้งานความร้อนสะสมค่อนข้างสูง เพิ่มประสิทธิภาพด้วยการระบายความร้อน



รูปที่ 2.12 MONSTER MOTO SHIELD VN2SP30

คุณสมบัติ

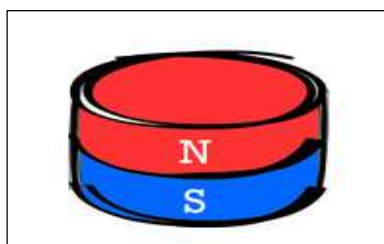
- แรงดันไฟฟ้าสูงสุด : 16V
- กระแสสุดในปัจจุบัน : 30 A
- ทำงานต่อเนื่องที่ : 14 A
- การตรวจจับกระแสที่มีให้กับพินอะนาล็อกของ Arduino
- MOSFET บนความต้านทาน : 19 mΩ (ต่อขา)
- ความถี่ PWM สูงสุด : 20 kHz
- ปิดระบบเมื่อความร้อนเกิน
- การปิดกั้นแรงดันและกระแสเกิน

2.6 แม่เหล็กนีโอไดเมียม (Neodymium magnets)

นีโอไดเมียม (Neodymium) เป็นธาตุโลหะลักษณะเงินมันวาวหายาก เมื่อสัมผัสอากาศ สีจะหมอง เพราะเกิดสนิมสารประกอบออกไซด์ หมายเลขอะตอมคือ 60 สัญลักษณ์ Nd จัดอยู่ในกลุ่มแลนทาไนด์ มีปริมาณบนพื้นโลกมากเป็นอันดับ 2 ในกลุ่มเดียวกันรองจากซีเรียม นีโอไดเมียมเป็นธาตุที่ไม่ได้พบในรูปแบบโลหะบริสุทธิ์เหมือนกับธาตุอื่นๆ ในกลุ่มแลนทาไนด์ นีโอไดเมียมใช้การกลั่นแบบปกติ แม่ว่านีโอไดเมียมถูกจัดว่าเป็นแม่เหล็กหายาก มีการกระจายกันอย่างแพร่หลายในโลกอยู่ในชั้นเปลือกโลก ส่วนใหญ่นีโอไดเมียมในโลกจะขุดได้ที่ในประเทศจีน

แม่เหล็กนีโอไดเมียม (ที่จริงเป็นส่วนผสม, Nd₂Fe₁₄B) เป็นแม่เหล็กถาวรที่แข็งแกร่ง แม่เหล็ก ที่รู้จักกัน แม่เหล็กนีโอไดเมียมแค่ไม่กี่กรัมสามารถยกน้ำหนักพันเท่าน้ำหนักของมันเองได้ แม่เหล็กนีโอไดเมียมมีราคาถูกกว่า เบาและแข็งแกร่งกว่า แม่เหล็กซาแมเรียม โคบอลต์ แต่แม่เหล็กนีโอไดเมียมไม่ได้เหนือกว่าในทุกด้าน แม่เหล็กนีโอไดเมียมก็มีการสูญเสียอำนาจแม่เหล็กของมันเองได้ที่อุณหภูมิสูงและอาจที่จะเกิดสนิม ในขณะที่แม่เหล็กซาแมเรียม โคบอลต์ไม่สามารถเกิดได้ การใช้ประโยชน์นีโอไดเมียมนี้ไม่พ้นเรื่องแม่เหล็กถาวร ปัจจุบันถือได้ว่า แม่เหล็กนีโอไดเมียมมีคุณสมบัติโดดเด่นที่สุดและแพร่หลายที่สุด แม่เหล็กนี้พัฒนาขึ้นเมื่อ ค.ศ.1984 โดยทีมงานของ มะซะ โตะ ซะกะวะ (Masato Sagawa) แห่งบริษัทซุมิโตโม เมทัลส์ (Sumitomo Special Metals) ปัจจุบันคือ บริษัท ฮิตาชิเมทัลส์ (Hitachi Metals) แม่เหล็กนี้เป็นส่วนผสมของนีโอไดเมียมเหล็ก Fe และ โบรอน B

เมื่อเปรียบเทียบแม่เหล็กนีโอไดเมียมกับแม่เหล็กอื่น พบว่ามีพลังสูงกว่าแม่เหล็กเฟอร์ไรต์ (ferrite magnet) ถึง 10 เท่า สูงกว่าแม่เหล็กซาแมเรียม โคบอลต์ (samarium-cobalt magnet) 1.7 เท่า ยิ่งไปกว่านั้น นีโอไดเมียมยังเป็นแลนทาไนด์ที่หายาก เมื่อนำมาผสมกับเหล็กซึ่งมีราคาถูก จึงมีจุดเด่นในเรื่องต้นทุนต่ำ อย่างไรก็ตาม แม่เหล็กนีโอไดเมียมยังมีข้อด้อยคือ ขึ้นสนิมได้ง่ายและอ่อนไหวง่ายต่ออุณหภูมิที่สูงขึ้น ซึ่งกำลังพัฒนาเพื่อปรับปรุงจุดนี้ นิยมใช้แม่เหล็กนีโอไดเมียมกับลำโพง เสดโฟน มอเตอร์เล็กๆ ในโทรศัพท์มือถือ HDD ฯลฯ

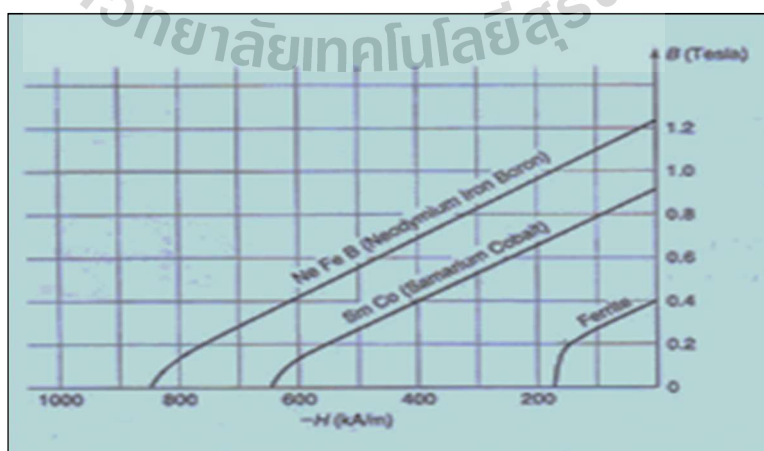


รูปที่ 2.13 ลักษณะการวางตัวขั้วเหนือ-ใต้ Nd



รูปที่ 2.14 แม่เหล็กนีโอไดเมียม

แม่เหล็กนีโอไดเมียมเป็นแม่เหล็กที่มีส่วนประกอบหลักคือ นีโอไดเมียม (Neodymium) เหล็ก โบรอน และมีคุณสมบัติเฉพาะของแม่เหล็กสูงสุดในกลุ่มแม่เหล็กที่มีอยู่ในปัจจุบัน แม่เหล็กนีโอไดเมียมเป็นแม่เหล็กที่ยกขึ้นมาเป็นตัวอย่างจากแม่เหล็กที่มีส่วนประกอบของแร่ธาตุที่หายาก (Rare earth) ต่างๆ ซึ่งมีแรงดึงดูดสูงกว่าแม่เหล็กอื่นๆ แม้ว่าจะทำให้มีขนาดเล็กก็ตาม แต่ก็ยังมีแรงดูดที่น่าพอใจ ดังนั้นจึงสามารถตอบสนองความต้องการได้เป็นอย่างดีในด้าน การลดขนาด (Down sizing) แม่เหล็กนีโอไดเมียมมีความแข็งแรงทนทานด้านกลไกที่เยี่ยมยอด แต่ไม่เหมาะที่จะนำมาใช้ในอุณหภูมิสูง มีการแปรรูปเพลาตึงเพื่อป้องกันสนิม และนำมาใช้กับมอเตอร์หรือเซนเซอร์ขนาดเล็ก เครื่องจักรที่มีความละเอียดสูง ฯลฯ



รูปที่ 2.15 กราฟเปรียบเทียบคุณสมบัติของแม่เหล็ก

ตารางที่ 2.2 ตัวอย่างคุณสมบัติเฉพาะของแม่เหล็กนีโอไดเมียม

ความหนาแน่นของฟลักซ์แม่เหล็กที่คงค้าง (Br)	แรงบังคับ (bHc)	แรงบังคับ (iHc)	Maximum Energy product (BH) _{max}
11.7 ~ 12.5 (kG)	10.8 ~ 12.2 (kOe)	>12.08 (kOe)	35 (MGOe)
1.17 ~ 1.25 (T)	859 ~ 970 (kA/m)	> 955 (kA/m)	278.5 (kJ/m ³)

ตารางที่ 2.3 ตัวอย่างรูปร่างแม่เหล็กนีโอไดเมียม

 เส้นผ่านศูนย์กลางภายนอก M ความหนา	 เส้นผ่านศูนย์กลางภายนอก เส้นผ่านศูนย์กลางภายใน M ความหนา	 ความกว้าง M ความหนา
แบบทรงกระบอก	แบบวงแหวน	แบบเหลี่ยม

ตารางที่ 2.4 แม่เหล็กนีโอไดเมียม (Nd-Fe-B) CP แบบจาน รูปร่างแหวน

ชนิด	ขนาด(mm) ลำดับขนาด	ผิวแม่เหล็ก	เกรด	ฟลักซ์	แรงดูด
CP	Φ16mm×Φ3.5mm×5mm(ชนิดวงแหวน M3)	ชุบนิกเกิล	N35	-	5kgf
CP	Φ20mm×Φ4.5mm×7mm(ชนิดวงแหวน M4)	ชุบนิกเกิล	N35	-	6kgf
CP	Φ25mm×Φ5.5mm×8mm(ชนิดวงแหวน M5)	ชุบนิกเกิล	N35	-	14kgf
CP	Φ32mm×Φ5.5mm×8mm(ชนิดวงแหวน M5)	ชุบนิกเกิล	N35	-	25kgf
CP	Φ36mm×Φ6.5mm×9mm(ชนิดวงแหวน M6)	ชุบนิกเกิล	N35	-	29kgf
CP	Φ42mm×Φ6.5mm×9mm(ชนิดวงแหวน M6)	ชุบนิกเกิล	N35	-	37kgf

2.7 ล้อโอมนิ (Omni-wheel)

ล้อ Omni หรือล้อโพลีคล้ายกับล้อ Mecanum เป็นล้อที่มีแผ่นขนาดเล็กกรอบ ๆ เส้นรอบวง ซึ่งตั้งฉากกับทิศทางการหมุน ผลก็คือล้อสามารถขับเคลื่อนด้วยแรงเต็มที่ แต่ยังสามารถเลื่อนไปทางด้านข้างได้ง่ายมาก ล้อเหล่านี้มักใช้ในระบบไครฟ์แบบ holonomic แพลตฟอรม์การ สามล้อ รอบในการกำหนดค่าสามเหลี่ยมเรียกว่าโดยทั่วไปกีวีไครฟ์ แพลตฟอรม์ Killough

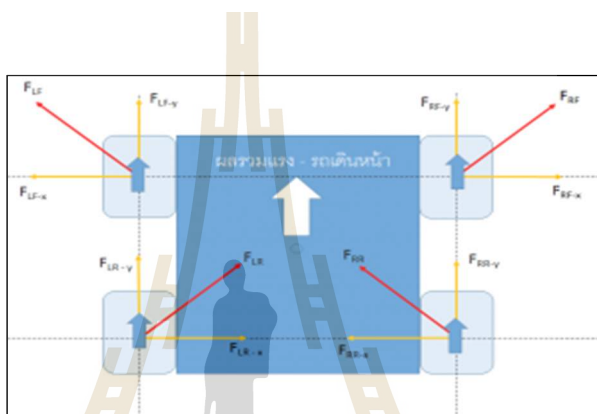
หลังจากที่สตีเฟน Killough ทำงานร่วมกับแพลตฟอรม์รอบทิศทางที่ Oak Ridge National Laboratory การออกแบบของ Killough ในปี 1994 ใช้ล้อคู่ที่ติดตั้งอยู่ในกรงที่มุมขวาและกันและกันและทำให้เกิดการเคลื่อนไหวตามรูปทรงกลมโดยไม่ใช้ล้อ Omni พวกเขาใช้ในการวิจัยหุ่นยนต์อัจฉริยะสำหรับหุ่นยนต์อิสระที่เป็นอิสระ ในโครงการต่างๆ เช่น VEX Robotics , Robocup และ FIRST Robotics หุ่นยนต์จำนวนมากใช้ล้อเหล่านี้เพื่อให้สามารถเคลื่อนที่ได้ทุกทิศทาง ล้อ Omni นอกจากนี้ยังมีบางครั้งใช้เป็นล้อขับเคลื่อนสำหรับหุ่นยนต์ไครฟ์ที่แตกต่างกันเพื่อให้เปลี่ยนทิศทางได้เร็วขึ้น Omni-wheels มักใช้เพื่อให้สามารถเคลื่อนที่บนแกนแนวนอนบนระบบขับเคลื่อน เช่นเดียวกับการเคลื่อนไหวไปข้างหน้าและย้อนกลับ



รูปที่ 2.16 Mecanum wheel

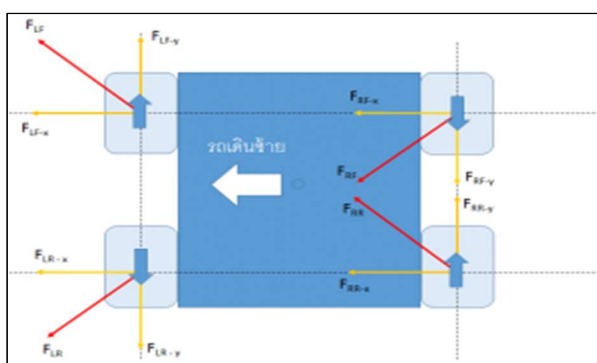
ล้อประเภท Mecanum เป็นล้อที่ทำให้รถหุ่นยนต์เคลื่อนที่ได้ในหลายๆทิศทางโดยไม่ต้องคอบหมุนไปมา รถจะสามารถเคลื่อนที่ ซ้าย ขวา หน้า หลัง ได้ทันที โดยที่ไม่ต้องหมุนรถ สามารถให้รถหุ่นยนต์มีอิสระในการเคลื่อนไหวที่สูงกว่าการใช้ล้อแบบปกติ และสามารถเคลื่อนที่ในพื้นที่แคบๆได้อย่างมีประสิทธิภาพ ต่อไปนี้จะยกตัวอย่างการเคลื่อนที่จากภาพการมองรถจากมุมบนสำหรับรถที่ใช้ล้อ Mecanum ล้อประเภทนี้แต่ละตัวนั้นจะต้องขับเคลื่อนด้วยมอเตอร์

ลูกศรสีฟ้าแสดงถึงทิศทางและแรงการหมุนของแต่ละล้อ เส้นสีแดงนั้นแสดงถึงแรงที่เกิดขึ้นจากการหมุนของแต่ละล้อ เราจะเห็นได้ว่าแรงสีแดงนั้นจะอยู่ในทิศทาง 45 องศาจากแกนหมุนของล้อ ล้อบนขวาและล่างซ้ายจะมีลูกกลิ้งที่หันไปทิศทางเดียวกัน ส่วนล้อบนซ้ายและล่างขวาจะมีลูกกลิ้งที่หันไปทิศทางเดียวกันแรงสีแดงที่เกิดจากการหมุนของล้อนั้นสามารถแบ่งออกเป็นแรงสี่มุมได้ แรงสี่มุมในแกน x และ y เมื่อรวมกันแล้วก็จะเท่าแรงสีแดง ที่นี้เพื่อดูว่ารถจะเคลื่อนที่อย่างไรก็เพียงเอาแรงสี่มุมทั้งหมดมารวมกัน จะเห็นได้ว่าแรงในแกน x นั้นหักล้างกันหมด ขณะที่แรงในแกน y รวมกัน ให้รถไปข้างหน้า เพื่อให้รถเคลื่อนที่ไปข้างหน้าทั้งสี่ข้างหมุนด้วยแรงที่เท่า ๆ กัน



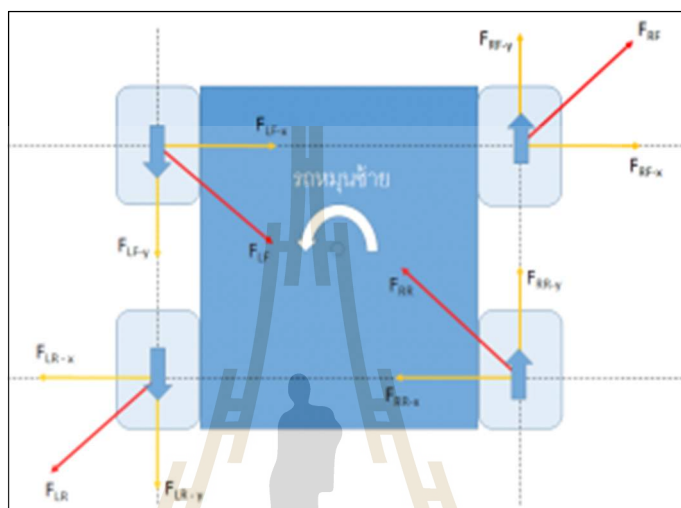
รูปที่ 2.17 Mecanum wheel Forward

เพื่อให้รถเคลื่อนที่ไปทางซ้าย ล้อหน้าซ้ายและหลังขวาจะเคลื่อนหมุนไปข้างหน้า ส่วนล้อบนขวาและล่างซ้ายจะหมุนไปข้างหลัง เมื่อรวมแรงที่เกิดจากแต่ละล้อแล้วจะเห็นว่าสุดท้ายแล้วมีแต่แรงที่ทำให้รถเคลื่อนไปทางซ้ายเท่านั้น



รูปที่ 2.18 Mecanum wheel Move Left

เพื่อให้รถนั้นหมุนไปด้านซ้าย ด้านซ้ายล้อจะหมุนไปด้านหลัง ส่วนด้านขวาล้อจะหมุนไปข้างหน้า จะเห็นได้ว่าสุดท้ายแล้วแรงรวมฝั่งซ้ายนั้นจะเป็นแรงที่กระทำไปทางด้านหลัง ส่วนแรงรวมฝั่งขวานั้นจะทำให้รถไปข้างหน้า สุดท้ายแล้วการที่ฝั่งซ้ายไปด้านหลัง และฝั่งขวาไปด้านหน้า จะทำให้รถหมุนไปด้านซ้าย โดยที่จุดหมุนของรถนั้นคือจุดกึ่งกลางระหว่างทั้งสี่ล้อ



รูปที่ 2.19 Mecanum wheel Sprint

ในส่วนของล้อ Omni นั้นเป็นล้อพิเศษที่มีลูกกลิ้งติดอยู่รอบล้อคล้ายๆล้อ Mecanum ข้อแตกต่างก็คือแทนที่ลูกกลิ้งเหล่านี้จะทำมุมเป็น 45 องศา จากแกนหมุนแบบล้อ Mecanum ล้อ Omin จะมีลูกกลิ้งที่ทำมุมเป็น 90 องศา กับตัวล้อตามที่แสดงในรูปด้านล่าง ล้อ Omni นั้นสามารถใช้ได้กับรถหุ่นยนต์สองประเภท รถสี่ล้อ และรถสามล้อ



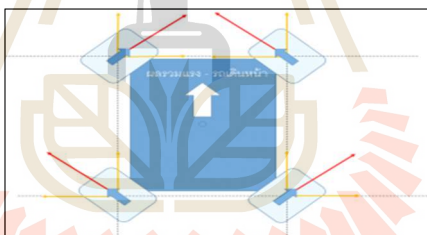
รูปที่ 2.20 Omni wheel

ตัวอย่างหุ่นยนต์ Omni แบบสี่ล้อ



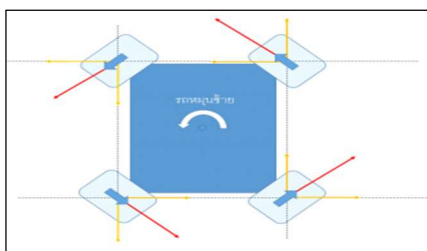
รูปที่ 2.21 Omni wheel แบบสี่ล้อ

สำหรับรถ Omni แบบที่ใช้สี่ล้อ นั้น ล้อจะต้องตั้งเป็นมุม 45 องศา แบบในรูปด้านล่างนี้ การเคลื่อนที่ของรถจะเกิดจากผลรวมแรงของแต่ละล้อ ในแบบเดียวกันกับรถที่ใช้ล้อ Mecanum เพื่อให้รถวิ่งไปข้างหน้าเราจะหมุนทุกๆล้อไปข้างหน้าเหมือนในรูปข้างล่างนี้



รูปที่ 2.22 Omni wheel สี่ล้อ Forward

ถ้าจะทำให้รถหมุนอยู่กับที่ เราจะต้องให้ล้อฝั่งซ้ายหมุนไปในทิศทางที่ตรงกันข้ามกับล้อฝั่งขวา



รูปที่ 2.23 Omni สี่ล้อ wheel Left

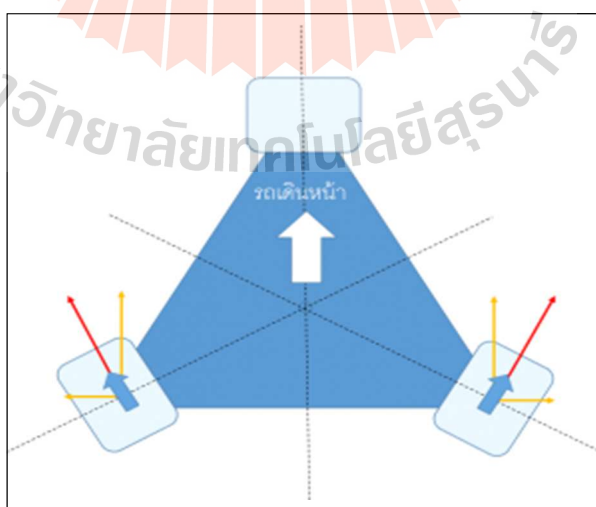
ตัวอย่าง Omni แบบสามล้อ

รถที่ใช้ล้อแบบ Omni นั้นจริงๆแล้วสามารถเคลื่อนไหวได้ครบทุกทิศทางโดยการใช้เพียงแค่สามล้อเท่านั้น



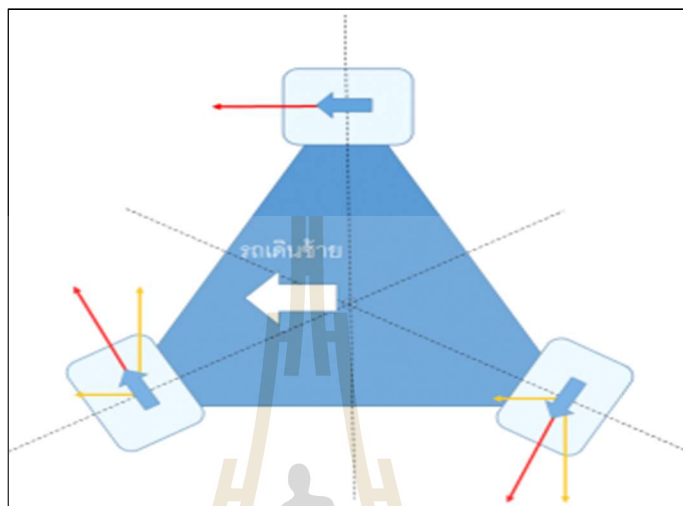
รูปที่ 2.24 Omni wheel สามล้อ Rota caster

เพื่อให้รถ Omni สามล้อเดินไปข้างหน้าเราจะขับเคลื่อนสองล้อหลังไปด้านหน้าเช่นในตัวอย่างภาพด้านล่าง ล้อหน้านั้นไม่ต้องมีการขับเคลื่อน



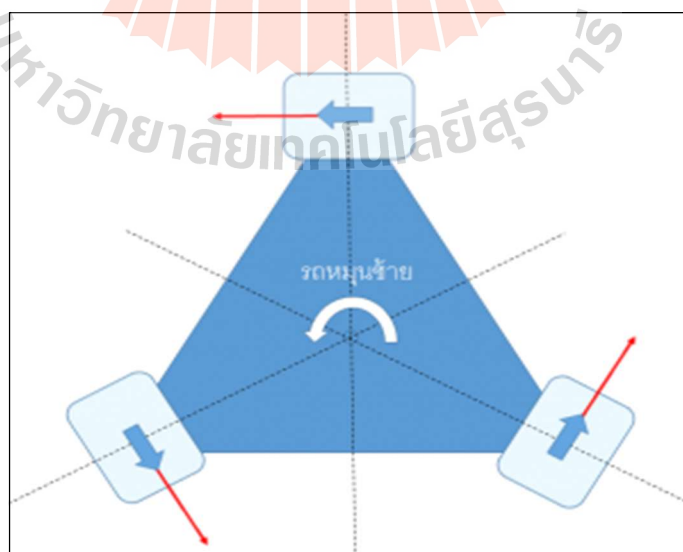
รูปที่ 2.25 Omni wheel สามล้อ Forward

จากรูปที่ 2.25 เพื่อให้รถ Omni สามล้อเคลื่อนไปด้านซ้าย สองล้อหลังนั้นจะขับเคลื่อนตรงข้ามกัน ส่วนล้อหน้านั้นจะขับเคลื่อนไปด้านซ้ายเช่นกัน



รูปที่ 2.26 Omni wheel สามล้อ Move Left

เพื่อให้รถ Omni หมุนเป็นวงกลมอยู่กับที่ ทุกล้อจะขับเคลื่อนเพื่อให้เกิดแรงหมุนที่กึ่งกลางหุ่นยนต์ด้วยแรงเท่าๆกัน ส่งผลให้เวกเตอร์ลัพธ์ส่งให้หุ่นยนต์หมุนตัว



รูปที่ 2.27 Omni wheel สามล้อ Sprint

2.8 แหล่งพลังงาน (Battery)

เป็นอุปกรณ์ที่ประกอบด้วย เซลล์ไฟฟ้าเคมี หนึ่งเซลล์หรือมากกว่า ที่มีการเชื่อมต่อภายนอกเพื่อให้กำลังงานกับอุปกรณ์ไฟฟ้า แบตเตอรี่มีเครื่องหมายบอกจะมีพลังงานศักย์ไฟฟ้าสูงกว่าขั้วที่มีเครื่องหมายลบ ขั้วที่มีเครื่องหมายลบคือแหล่งที่มาของอิเล็กตรอนที่เมื่อเชื่อมต่อกับวงจรภายนอก แล้วอิเล็กตรอนเหล่านี้จะไหล และส่งมอบพลังงานให้กับอุปกรณ์ภายนอก เมื่อแบตเตอรี่เชื่อมต่อกับวงจรภายนอก สารอิเล็กโทรไลต์ มีความสามารถที่จะเคลื่อนที่โดยทำตัวเป็นไอออน ขอมให้ปฏิกิริยาทางเคมีทำงานแล้วเสร็จในขั้วไฟฟ้าที่อยู่ห่างกัน เป็นการส่งมอบพลังงาน ให้กับวงจรภายนอก การเคลื่อนที่ไหลของไอออนเหล่านี้ที่อยู่ในแบตเตอรี่ที่ทำให้เกิดกระแสไหลออกจากแบตเตอรี่เพื่อปฏิบัติงานในอดีตคำว่า แบตเตอรี่ หมายถึงเฉพาะอุปกรณ์ที่ประกอบด้วยเซลล์หลายเซลล์ แต่การใช้งานได้มีการพัฒนาให้รวมถึงอุปกรณ์ที่ประกอบด้วยเซลล์เพียงเซลล์เดียวแบตเตอรี่แปลงพลังงานเคมีให้เป็นพลังงานไฟฟ้าโดยตรง แบตเตอรี่ประกอบด้วยเซลล์แบบโวลตาได้มากกว่าหนึ่งเซลล์ แต่ละเซลล์ประกอบด้วยสอง ครึ่งเซลล์ ที่เชื่อมต่อเรียงกันเป็นแถวโดยสารอิเล็กโทรไลต์ที่เป็นสื่อกระแสไฟฟ้าที่มีไอออนที่มีประจุลบ (anion) และ ไอออนที่มีประจุบวก (cation) ครึ่งเซลล์หนึ่งตัวจะมีอิเล็กโทรไลต์และขั้วลบ (อิเล็กโทรดที่แอนไอออนวิ่งเข้าหา) อีกครึ่งเซลล์หนึ่งจะมีอิเล็กโทรไลต์และขั้วบวก (อิเล็กโทรดที่แคทไอออนวิ่งเข้าหา Redox ปฏิกิริยา Redox เป็นตัวให้พลังงานกับแบตเตอรี่ แคทไอออนจะลดลง (อิเล็กตรอนมีการเพิ่ม) ที่แคโทดระหว่างการชาร์จประจุ ในขณะที่แอนไอออนจะถูกออกซิไดซ์ (อิเล็กตรอนจะถูกปล่อยออก) ที่ขั้วบวกระหว่างการชาร์จ ในระหว่างการดีสชาร์จกระบวนการจะเป็นตรงกันข้าม ขั้วไฟฟ้าทั้งสองไม่ได้สัมผัสกัน แต่เชื่อมต่อทางไฟฟ้าโดย อิเล็กโทรไลต์ เซลล์บางตัวใช้อิเล็กโทรไลต์แตกต่างกันสำหรับแต่ละครึ่งเซลล์ ตัวคั่นช่วยให้ไอออนไหลระหว่างครึ่งเซลล์ แต่จะช่วยป้องกันการผสมของอิเล็กโทรไลต์ทั้งสองด้าน แต่ละครึ่งเซลล์มี แรงเคลื่อนไฟฟ้า (หรือ EMF) ที่กำหนดโดยความสามารถของการจับกระแสไฟฟ้าจากภายในสู่ภายนอกของเซลล์

ชนิดเซลล์ของแบตเตอรี่

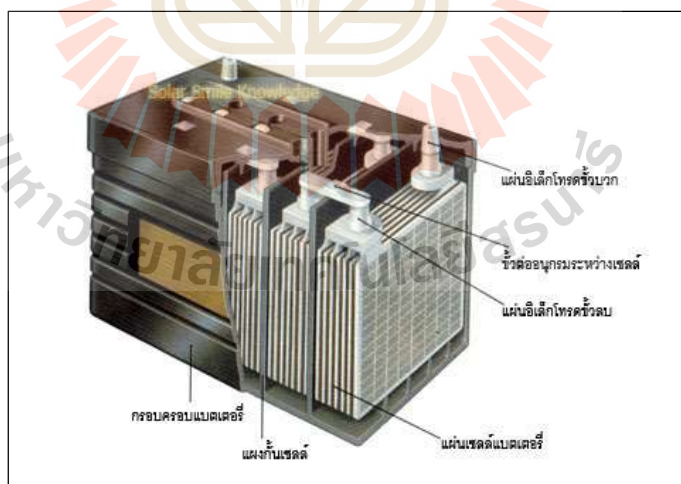
- นิกเกิล-แคดเมียม (NiCd)
- นิกเกิลเมตทัลไฮไดรด์ (NiMH)
- ลิเทียม-ไอออน (Li-Ion)



รูปที่ 2.28 ตัวอย่างแบตเตอรี่หลายชนิด

2.8.1 แบตเตอรี่แบบลิธเอซิด (Lead-Acid Battery)

ภายในลิธเอซิดแบตเตอรี่จะประกอบด้วยเซลล์อยู่ภายในโดยต่อกันแบบอนุกรม ซึ่งจำนวนเซลล์ จะขึ้นอยู่กับการออกแบบแบตเตอรี่นั้นๆ เพื่อให้มีค่าแรงดันใช้งานที่เท่าไร โดยทั่วไปหนึ่งเซลล์ มีแรงดันประมาณ 2 โวลต์ ตัวอย่างเช่นแบตเตอรี่รถยนต์มีแรงดันใช้งานที่ 12 โวลต์ ดังนั้นข้างในแบตเตอรี่จะประกอบด้วยเซลล์ 6 เซลล์ต่ออนุกรมกันอยู่



รูปที่ 2.29 โครงสร้างภายในของแบตเตอรี่

ลักษณะของการปล่อยประจุไฟฟ้าของแบตเตอรี่ จะแบ่งออกเป็น 2 แบบด้วยกัน

1. แบตเตอรี่ที่สามารถปล่อยประจุ (กระแส) ไฟฟ้าได้น้อย (Shallow-Cycle Battery) แบตเตอรี่ที่ออกแบบมาเพื่อปล่อยประจุไฟฟ้าได้ประมาณ 10-20 เปอร์เซ็นต์ของประจุไฟฟ้ารวมก่อน จะทำการนำไปชาร์จประจุใหม่ ในส่วนของการปล่อยประจุไฟฟ้าจะมีหน่วยเป็นแอมมอวาร์ด (Ahr) เช่น 100 Ahr หมายถึงแบตเตอรี่สามารถปล่อยประจุกระแสไฟฟ้า 100 หน่วย ได้ 1 ชั่วโมง

ตัวอย่าง ถ้ามีแบตเตอรี่แบบปล่อยประจุได้น้อย (Shallow cycle battery) ที่สามารถปล่อยประจุไฟฟ้าได้ 100 แอมมอวาร์ด แบตเตอรี่ตัวนี้ควรที่จะปล่อยประจุไฟฟ้าหรือใช้กระแสไฟฟ้าเพียง 10-20 แอมมอวาร์ด หลังจากนั้นจะต้องทำการชาร์จประจุให้เต็มก่อนการคลายประจุครั้งต่อไป ถ้าการปล่อยประจุมากเกินไปที่กำหนดไว้ เช่นทำการปล่อยประจุที่ 50 แอมมอวาร์ด ซึ่งจะทำให้แบตเตอรี่มีอายุการใช้งานที่สั้นลง (เสื่อมเร็ว) อย่างมากเช่นตามสเปคอายุการใช้งานของแบตเตอรี่สามารถชาร์จได้ 3000 ครั้งอาจจะลดเหลือเพียงแค่ 1000 ครั้ง ดังนั้นการออกแบบระบบโดยรวมควรคำนึงถึงลักษณะการปล่อยประจุไฟฟ้าของแบตเตอรี่ด้วย

2. แบตเตอรี่ที่สามารถปล่อยประจุ (กระแส) ไฟฟ้าได้มาก (Deep-Cycle Battery) คือแบตเตอรี่ที่สามารถปล่อยประจุได้ถึง 60-80 เปอร์เซ็นต์ ของประจุรวมก่อนที่จะทำการชาร์จประจุ ส่วนมากแล้วจะนำมาใช้กับระบบผลิตพลังงานไฟฟ้าในบ้านพักอาศัย แบตเตอรี่ชนิดนี้จะมีราคาที่สูงกว่าแบบแรกมาก แต่ใช้เพียงไม่กี่ตัวก็สามารถทดแทนประจุไฟฟ้ารวมจากแบตเตอรี่แบบแรก แบตเตอรี่แบบนี้จะมีความคุ้มค่าในระยะยาว

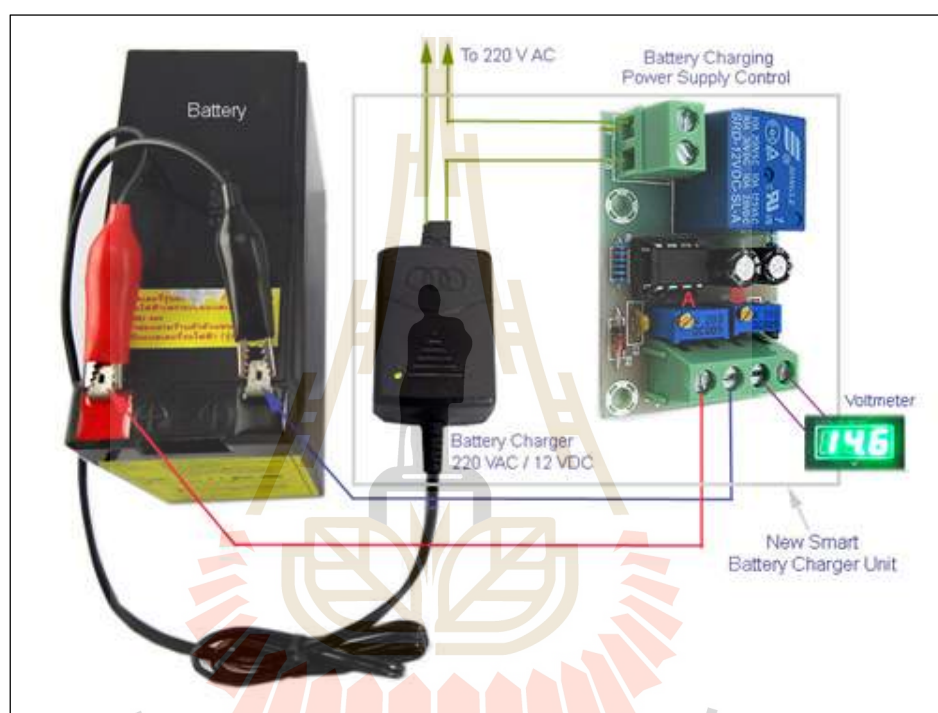
ความสัมพันธ์ในการทำงานของแบตเตอรี่กับเครื่องควบคุมการชาร์จ

เครื่องควบคุมการชาร์จ แบตเตอรี่ จะต่อกับเครื่องควบคุมการชาร์จซึ่งทำหน้าที่ปรับแรงดันให้เหมาะสม ไม่ให้สูงไปเพราะอาจทำให้แบตเตอรี่เสียหายได้ ถ้าแบตเตอรี่มีแรงดันที่ต่ำมากกว่าค่าที่ตั้งไว้ในเครื่องควบคุมการชาร์จ เครื่องควบคุมการชาร์จจะปลดโหลดออกไปทันที เพราะถ้าไม่ทำอย่างนี้แล้ว ประจุที่เก็บไว้ในแบตเตอรี่จะถูกปล่อยไปจนหมด ซึ่งไม่เป็นผลดีกับแบตเตอรี่ เพราะจะส่งผลทำให้เซลล์ที่อยู่ภายใน ไม่สามารถกลับมาชาร์จประจุได้อีก

ข้อควรระวัง! ไม่ควรปล่อยให้แบตเตอรี่ปล่อยประจุ (กระแสไฟ) จนหมด เพราะจะทำให้ประสิทธิภาพในการเก็บประจุของแบตเตอรี่ลดลงไปอย่างมาก และบางครั้งจะไม่สามารถนำกลับมาชาร์จประจุได้อีกต่อไป

- ควรติดตั้งแบตเตอรี่ที่อุณหภูมิที่กำหนดไว้ในสเปค โดยส่วนใหญ่แล้วแบตเตอรี่จะทำงานได้ดีที่อุณหภูมิ 25 องศาเซลเซียส ถ้าอุณหภูมิสูงกว่านี้จะทำให้อายุการใช้งานของแบตเตอรี่ลดลง ถ้าอุณหภูมิต่ำ จะทำให้ประสิทธิภาพในการเก็บประจุลดลง

- ควรเลือกขนาดความจุของแบตเตอรี่ให้มีการชาร์จประจุเต็มทุกวัน เพราะถ้าแบตเตอรี่แบบลิเธียมไม่เลยชาร์จเต็มเลย จะทำให้อายุการใช้งานของแบตเตอรี่สั้นลง และสิ้นเปลืองค่าใช้จ่ายโดยใช่เหตุ
- การติดตั้งอุปกรณ์ที่ใช้งาน ต้องมีความเหมาะสมกับขนาดของแบตเตอรี่ มิฉะนั้นอาจทำให้แบตเตอรี่เสื่อมสภาพเร็ว และทำให้เราเสียค่าใช้จ่ายในการเปลี่ยนแบตเตอรี่ชุดใหม่



รูปที่ 2.30 Battery Charger Unit

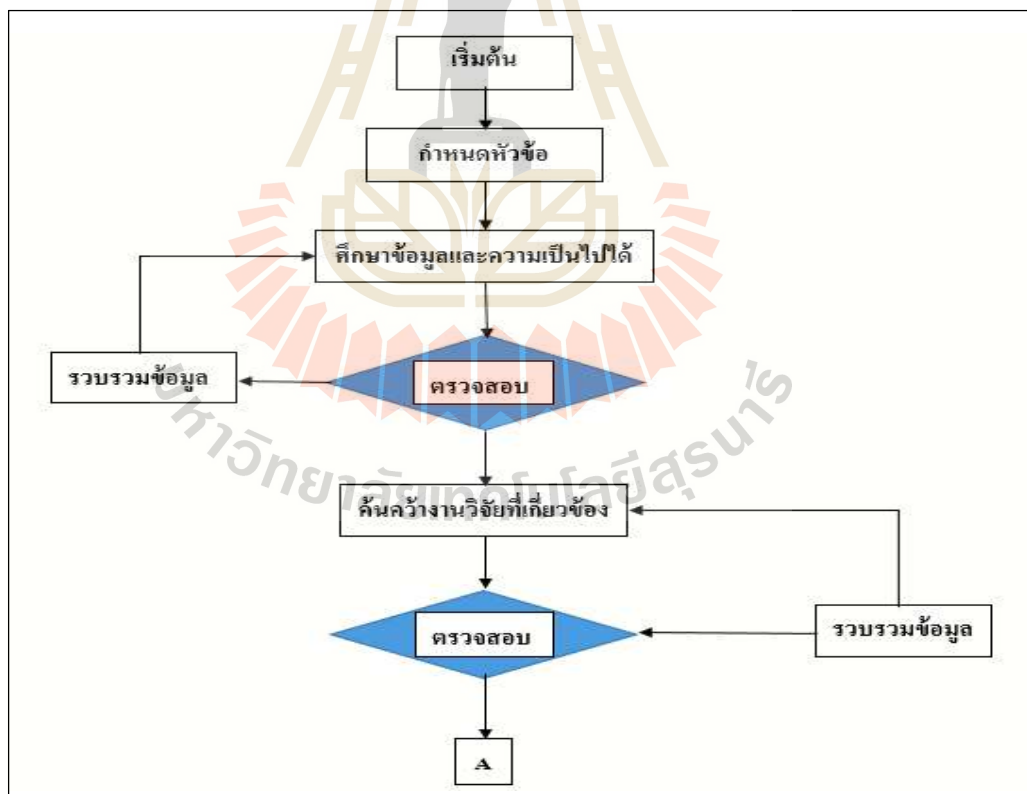
ส่วนใหญ่ในหุ่นยนต์ ทั่วไปจะมีแหล่งจ่ายไฟอย่าง 2 ชุดใหญ่ๆ ด้วยกัน คือ ภาคจ่ายไฟให้กับภาคคอนโทรล 5V และภาคจ่ายไฟให้กับชุดขับเคลื่อน ในภาคนี้จะเป็นภาคสำคัญอีกภาคหนึ่งที่จะทำให้ระบบมีความทนต่อการรบกวนต่าง ๆ ถ้าเราออกแบบภาคจ่ายไฟไม่ดีแล้ว จะทำให้วงจรของเราถูกรบกวนจากสัญญาณต่างๆ ได้ง่าย เรานิยมใช้แบตเตอรี่ขนาดเล็กที่พบเห็นกันโดยทั่วไปซึ่งแบตเตอรี่แบบนี้จะให้แรงดัน 6V หรือ 12V ก็ขึ้นอยู่กับผู้ผลิตจะออกแบบไว้ แบตเตอรี่แบบนี้จะสามารถจ่ายกระแสได้สูงกว่าแบบถ่านไฟฉาย เรานิยมใช้แบตเตอรี่แบบนี้ในวงจรที่ต้องการกระแสสูงหรือต้องการทำงานอย่างต่อเนื่องเป็นเวลานาน เช่น ใช้เป็นแหล่งพลังงานในการขับเคลื่อนสำหรับการแข่งขัน เป็นต้น แบตเตอรี่แบบนี้จะมีราคาสูง และ ตัวชาร์จก็จะมีราคาสูงตาม

บทที่ 3 วิธีดำเนินงานวิจัย

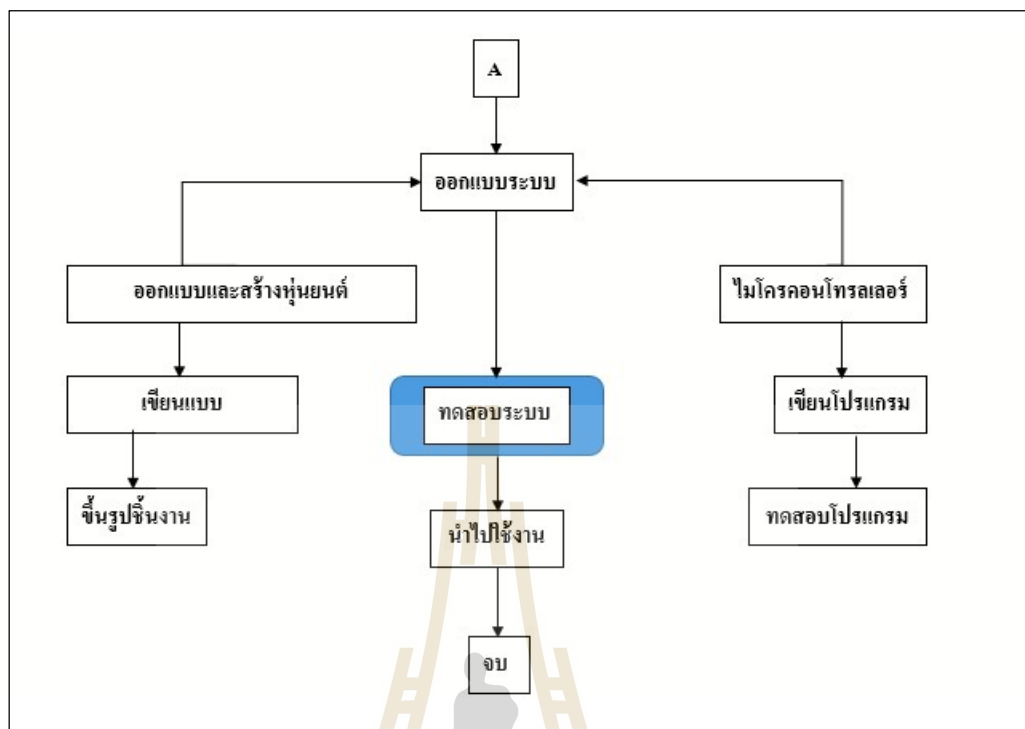
3.1 กล่าวนำ

งานวิจัยนี้เป็นส่วนหนึ่งในการศึกษามีความต้องการออกแบบและสร้างหุ่นยนต์ SPHERICAL ROBOT นำไปพัฒนาและประยุกต์ใช้งานตามความต้องการด้านอื่น ๆ ต่อไปกล่าวคือในส่วนของวิธีการดำเนินการซึ่งมีด้วยกัน 2 ส่วน โดยแบ่งเป็นส่วนที่ 1 คือการออกแบบขั้นตอนการวิจัย และส่วนที่ 2 คือการออกแบบขั้นตอนการสร้าง ซึ่งขั้นตอนในการดำเนินงานจะมีดังต่อไปนี้

3.2 การออกแบบขั้นตอนการวิจัย



รูปที่ 3.1 แผนผังแสดงขั้นตอนการดำเนินงานวิจัย



รูปที่ 3.2 แผนผังแสดงขั้นตอนการดำเนินงานวิจัย (ต่อ)

3.3 การออกแบบขั้นตอนการสร้าง

ในส่วนของขั้นตอนการสร้างหุ่นยนต์นั้น ได้แบ่งส่วนการดำเนินงานออกเป็น 2 ส่วน ดังนี้

3.3.1 สร้าง Hardware

ในการออกแบบจะเลือกใช้วัสดุที่เป็น Aluminum เนื่องจากมีน้ำหนักที่เบาและแข็งแรง ภายหลังจากการออกแบบด้วยโปรแกรมแล้ว ทำให้ทราบขนาดของโครงสร้างที่แน่นอน เพื่อจัดเตรียมอุปกรณ์สำหรับโครงสร้าง และทำการประกอบให้ตรงตามแบบที่วางแผนไว้ โดยในงานวิจัยนี้ ผู้พัฒนาได้สร้างหุ่นยนต์ออกมาหลายโมเดล เพื่อใช้ทดสอบความสามารถในการขับเคลื่อนของหุ่นยนต์ว่าโมเดลไหนมีประสิทธิภาพที่สุด หลังจากนั้นทำการติดตั้งอุปกรณ์อิเล็กทรอนิกส์ ที่ใช้ในการควบคุมซึ่งในการติดตั้งอุปกรณ์อิเล็กทรอนิกส์นี้ จะต้องคำนึงถึงพื้นที่ที่เหมาะสมและสามารถปรับปรุงแก้ไขได้ง่าย รวมไปถึงลดการเชื่อมต่อของสายไฟให้น้อยและสั้นที่สุด เพื่อเพิ่มประสิทธิภาพการทำงาน

3.3.2 เขียน Software

ในส่วนนี้เป็นขั้นตอนการออกแบบโปรแกรมเพื่อใช้ในการควบคุมระบบการทำงาน โดยผู้พัฒนา จะเรียงลำดับการทดสอบการทำงานแยกออกจากกัน โมดูลย่อย เมื่อสมบูรณ์ทีละส่วน แล้วจึงนำเข้ามารวมกัน

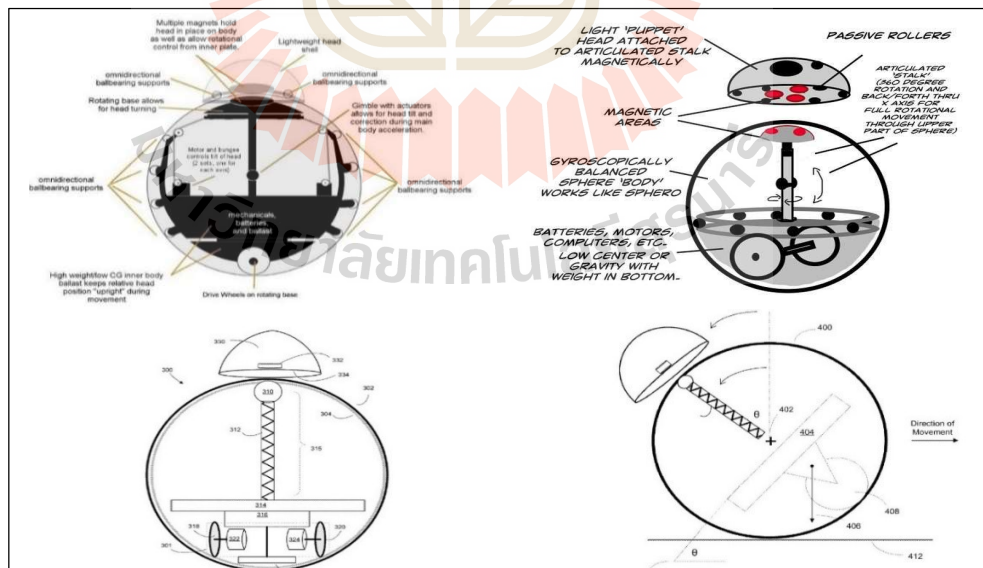
3.4 ศึกษาข้อมูลโครงสร้างก่อนดำเนินการ

ในส่วนนี้เป็นดำเนินการค้นคว้าหาตัวอย่างที่เหมาะสมที่สุด เพื่อเป็นแม่แบบในการใช้สร้าง โดยมีการกำหนดพารามิเตอร์ต่างๆ ว่าด้วยขนาดของ Robot ความเร็วที่ต้องการ ระยะเวลาการใช้งาน ในแต่ละภารกิจ รวมถึงระยะเวลาในการชาร์ตแบต

หลังจากนั้นดำเนินการเก็บบันทึกข้อมูล ตรวจสอบ และสรุปผลแนวทางการสร้างต้นแบบ โดยข้อมูลที่ได้ทำการเก็บและนำมาวิเคราะห์สรุปผล ใช้เป็นประโยชน์ในการสร้างหุ่นยนต์ตลอดจน การแก้ไขข้อบกพร่องที่เกิดขึ้น ระหว่างการสร้างตัวชิ้นงานและทดสอบ จากนั้นทำการบันทึกและสรุปผลในขั้นตอนถัดไป

3.4.1 ข้อมูลโครงสร้าง ต้นแบบ Robot

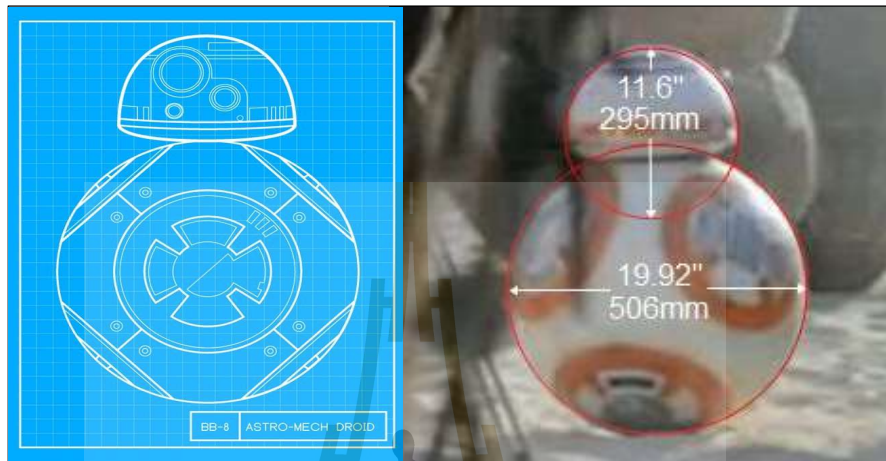
ผู้พัฒนาศึกษาข้อมูลหุ่นยนต์หลากหลายโมเดล เพื่อประกอบการออกแบบที่ดีที่สุด



รูปที่ 3.3 แบบ Robot ที่ศึกษาข้อมูล โครงสร้างภายใน

3.5 ขั้นตอนการปฏิบัติงาน

Step 1/6 : ศึกษารายละเอียดโครงสร้างภายในและภายนอกของหุ่นยนต์แม่แบบ



รูปที่ 3.4 ถอด Scale BB8



รูปที่ 3.5 เทียบ Scale BB8 ด้วยหน่วย pixel



รูปที่ 3.6 Head BB8 Scale

ตารางที่ 3.1 ขนาดของ Robot

PART	Size Master	Re_side	Use
Body	$(295+310)/2 = 515.1 \text{ mm}$	500 mm	900
Head	$(506+525)/2 = 302.5 \text{ mm}$	300 mm	$(500 \times 300) / 900 = 500$

*ในส่วนของหัวกับลำตัวมีขนาดประมาณ 5:3 ในการสร้างหุ่นยนต์นี้ มีการเพิ่มขนาดของหุ่นยนต์ขึ้น

Step 2/6 : สร้างเปลือกห่อหุ้ม หุ่นยนต์ภายใน Shell ด้วยไฟเบอร์กราส



รูปที่ 3.7 Shell Robot



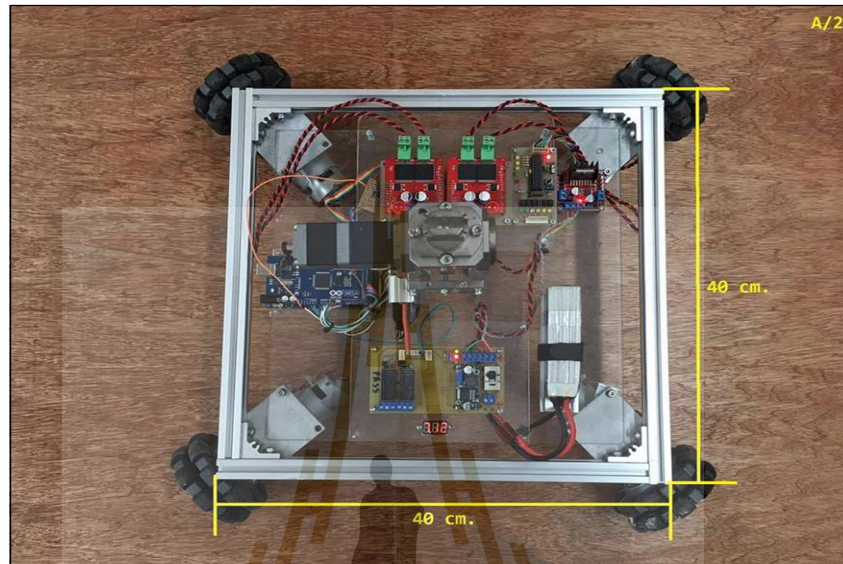
รูปที่ 3.8 จัดตบแต่งรอยแผล



รูปที่ 3.9 Head - Body Mask # I

ขั้นตอน ทดสอบความเข้ากันได้ระหว่าง Shell ที่สร้างขึ้นกับส่วนหัวจำลอง Head & Body Mask # I

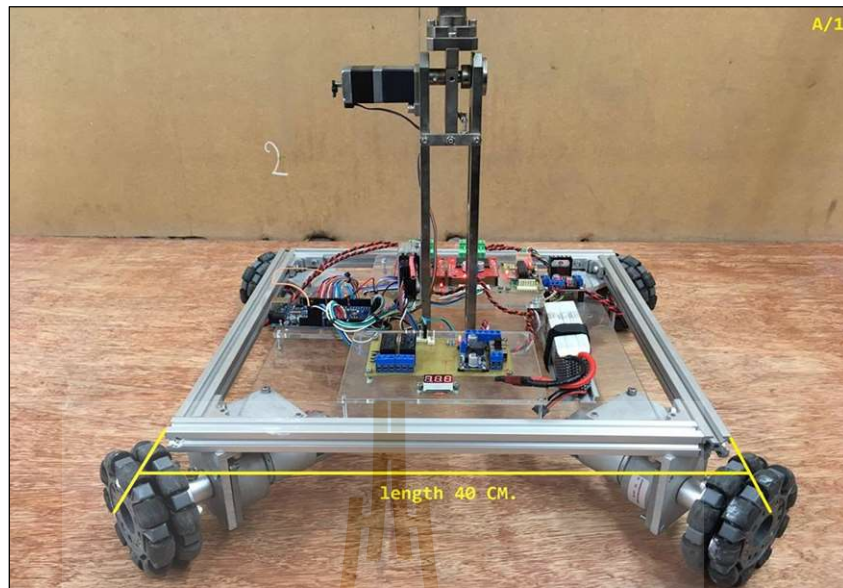
Step 3/6 : สร้างตัวหุ่นยนต์ขับเคลื่อนภายใน



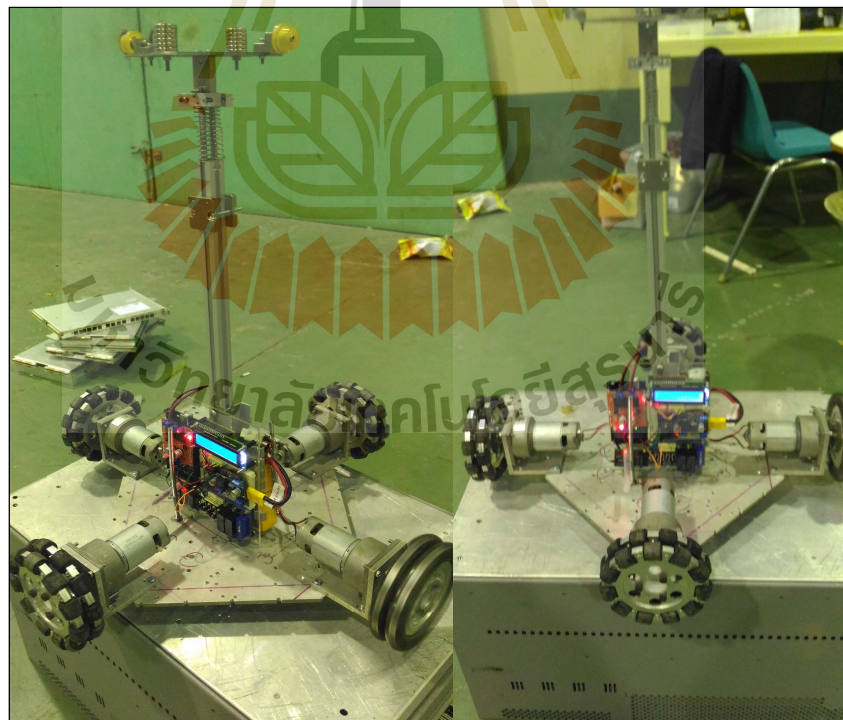
รูปที่ 3.10 Robot mask # I (top view)



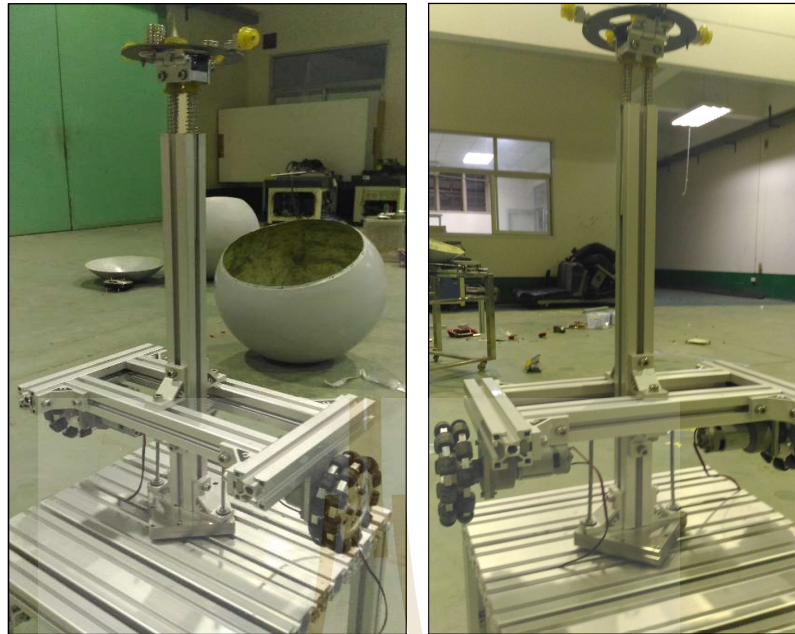
รูปที่ 3.11 Robot mask # I (side view)



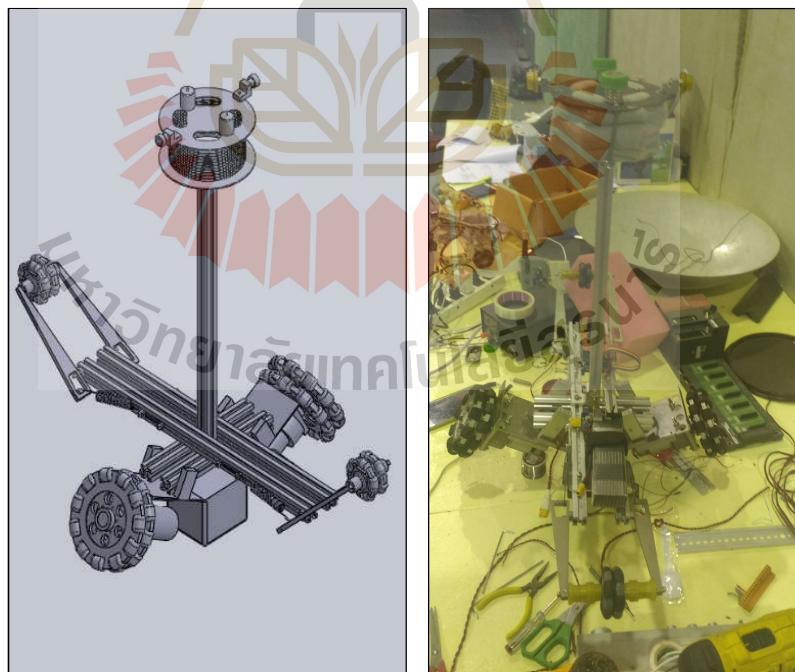
รูปที่ 3.12 Robot mask # I (front view)



รูปที่ 3.13 Robot mask # II



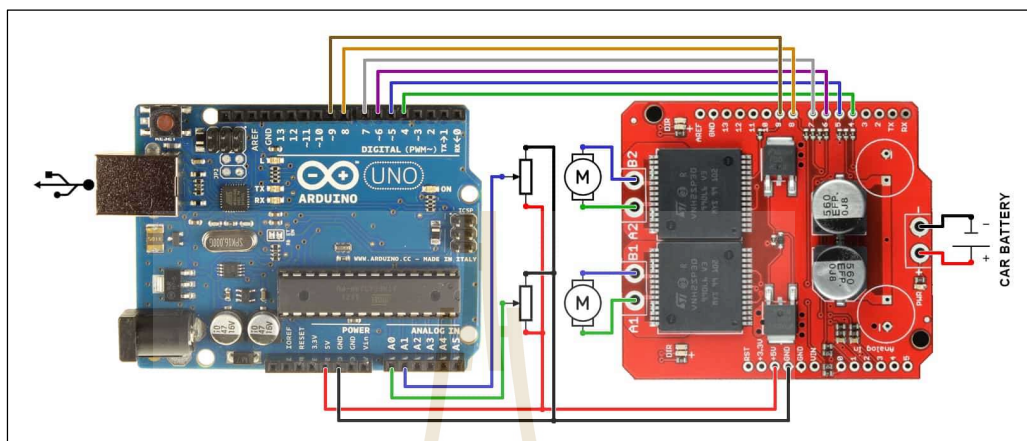
รูปที่ 3.14 Robot mask # III



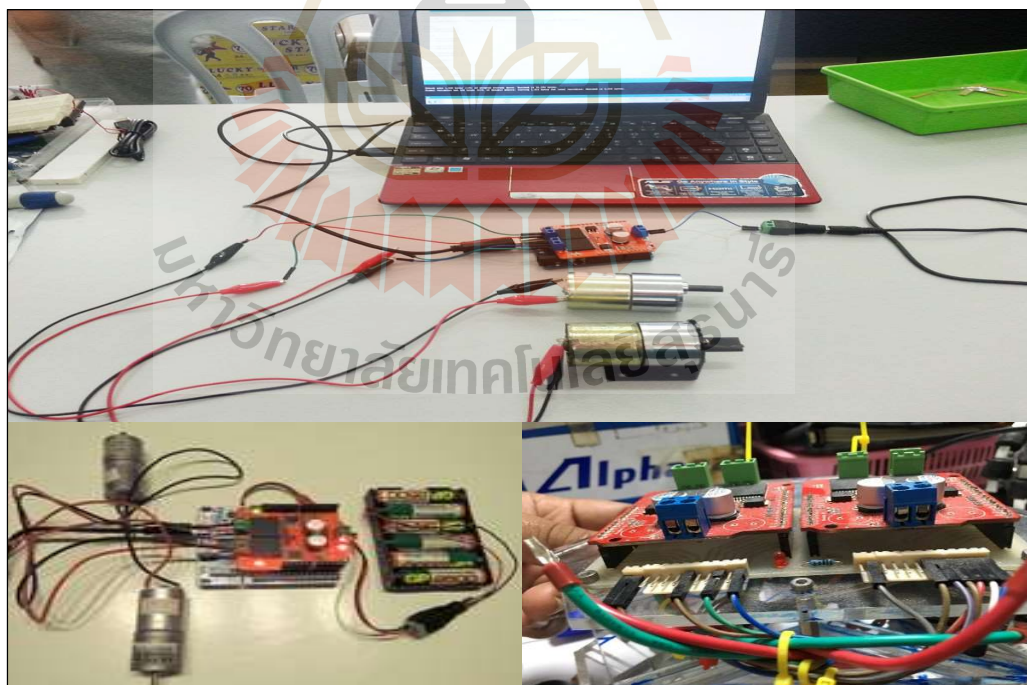
รูปที่ 3.15 Robot mask # IV

Step 4/6 : สร้างระบบควบคุมหุ่นยนต์และทำการเชื่อมต่อกับชุดรีโมทคอนโทรล

Sub 1/4 : ทำการเชื่อมต่อระหว่าง Arduino , Driver , Motor และทดสอบควบคุมการทำงาน

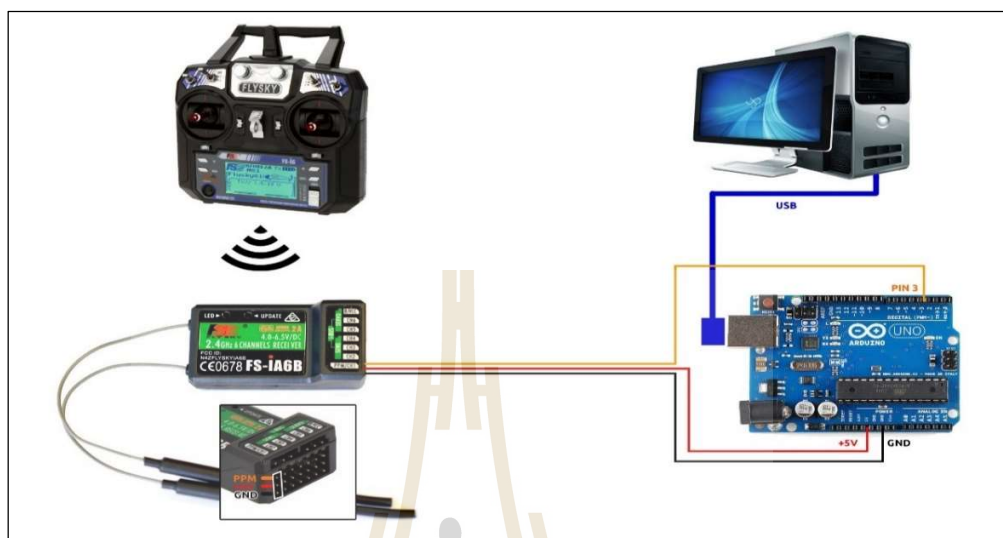


รูปที่ 3.16 Circuit Monster Shield Drive



รูปที่ 3.17 Test Drive Motor

Sub 2/4 : ทดสอบการเชื่อมต่อระหว่าง Arduino กับ Receiver และทดสอบปรับ Parameters



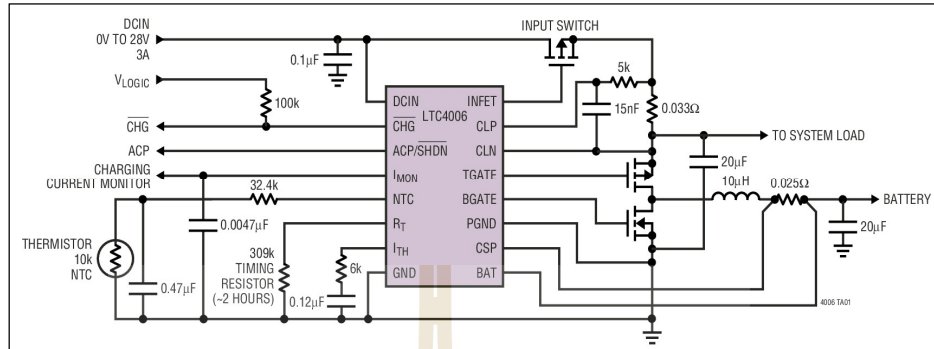
รูปที่ 3.18 Test Connect Receiver



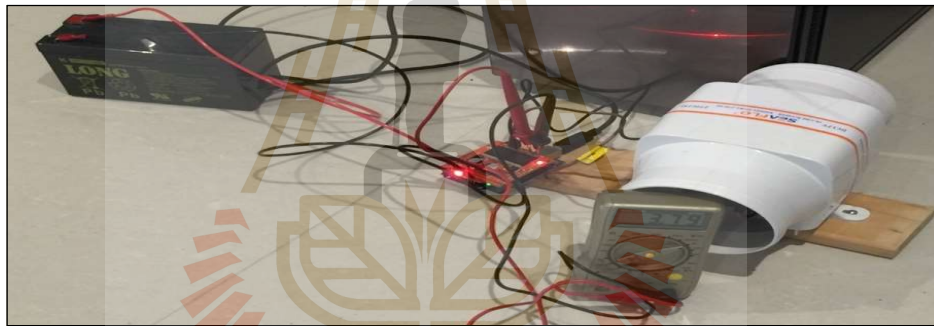
รูปที่ 3.19 Test Blink LED By RC Radio

จากการทดลองขั้นนี้ pulseIn จะส่งค่าความกว้างของ pulse ออกมา หน่วยเป็น millisec ซึ่งอยู่ในช่วง 1,000 ถึง 2,000 ซึ่งค่ากลางของมันจะอยู่ที่ 1,500 ค่าได้ที่ออกมา ซึ่งสามารถแปลงให้อยู่ในช่วง -500 ถึง 500 ได้ด้วยวิธี $\text{map}(x, 1000, 2000, -500, 500)$ ซึ่งจะ map ตัวเลขในช่วง 1000 ถึง 2000 ให้เป็น -500 ถึง 500 หลังจากนั้นทดสอบควบคุม แสดงผลออกทาง LED และ Serial Monitor

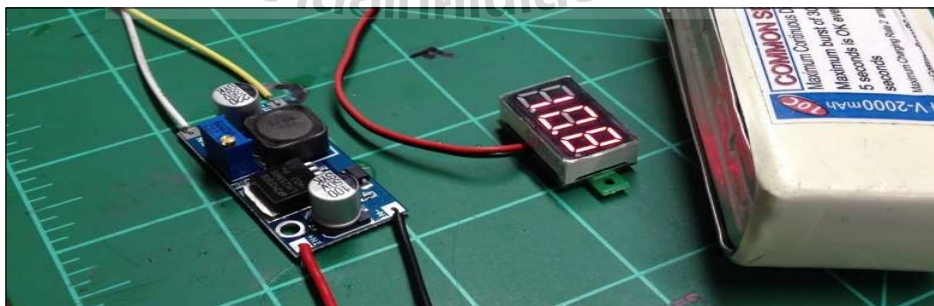
Sub 3/4 : เชื่อมต่อ Battery เข้ากับระบบหลักที่ได้ทดสอบใน Sub 1/4 และ Sub 2/4



รูปที่ 3.20 Power module

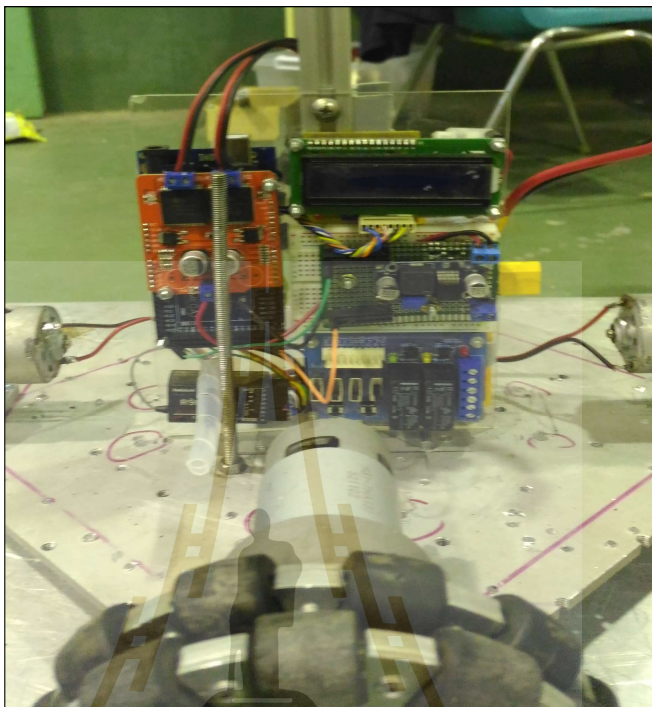


รูปที่ 3.21 Test Drive and Battery

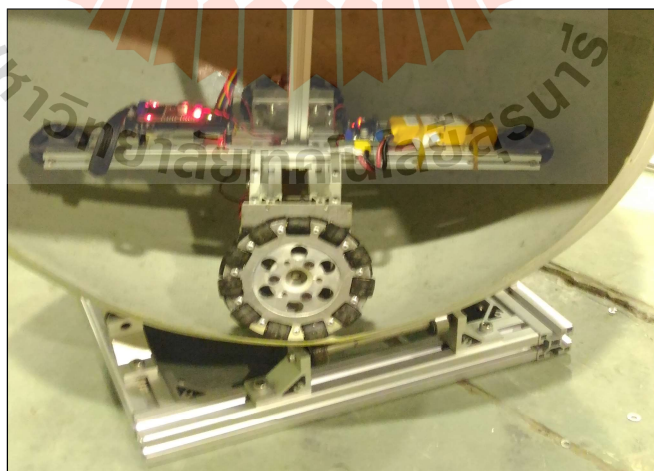


รูปที่ 3.22 Stepdown

Sub 4/4 : ทำการรวม Sub 1,2,3 เข้าด้วยกันและทดสอบการทำงาน

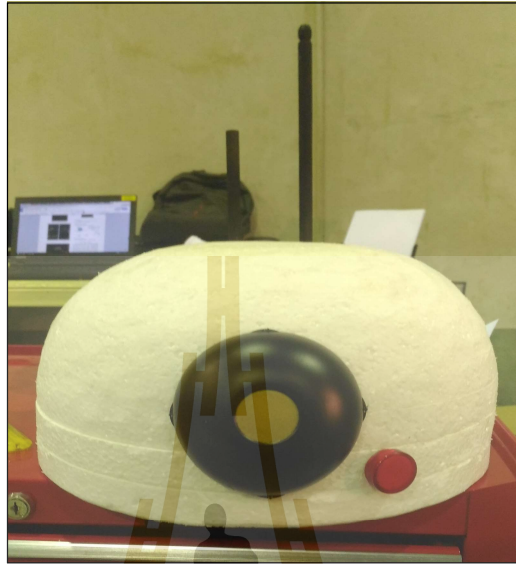


รูปที่ 3.23 Test Controller Mask #II



รูปที่ 3.24 Test Controller Mask #IV

Step 5/6 : สร้างส่วนหัวและติดแม่เหล็กดูส่วนหัว



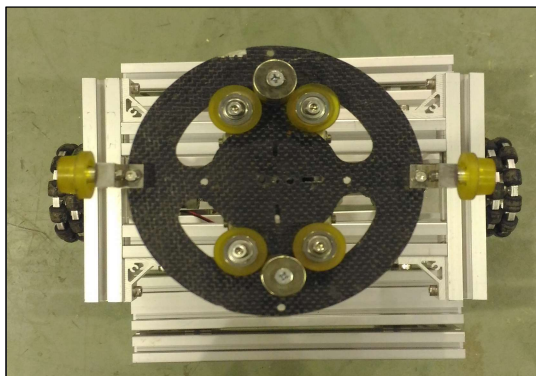
รูปที่ 3.25 Head Robot New



รูปที่ 3.26 Magnate in Head



รูปที่ 3.27 Damping



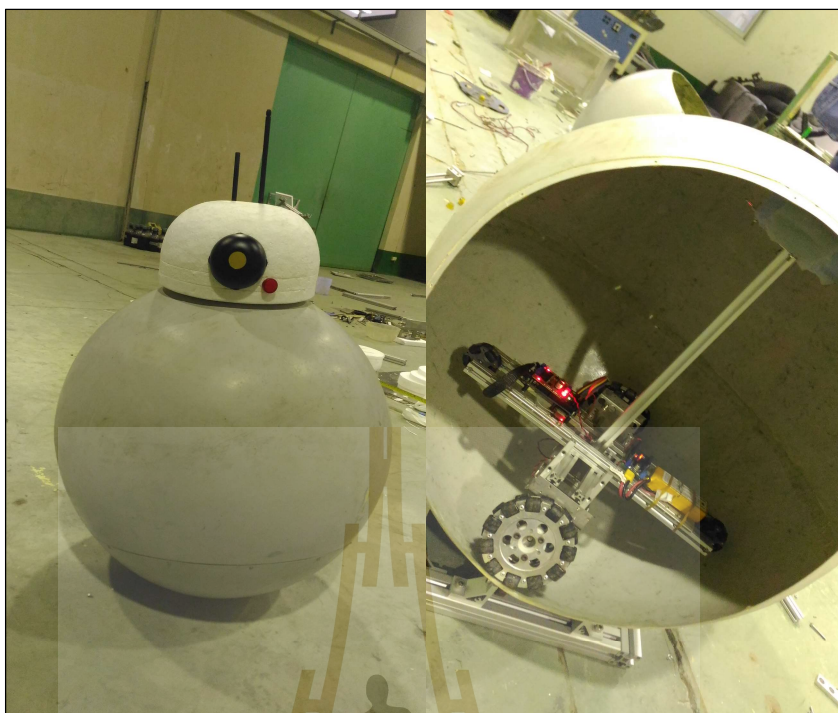
รูปที่ 3.28 Magnate in Shell (top view)



รูปที่ 3.29 Test Install Head & Body

ในส่วนนี้แบ่งงานสำคัญออกเป็น 2 ส่วน ดังนี้

1. ส่วนหัวภายนอก Shell ดังที่ปรากฏในรูปที่ 3.25 Head Robot และ 3.26 Magnate in Head
2. ส่วนที่อยู่ภายใน Shell ดังที่ปรากฏในรูปที่ 3.27 Damping และ 3.28 Magnate top view



รูปที่ 3.30 Head & Body #II

Step 6/6 : ทดสอบและแก้ไขการทำงาน

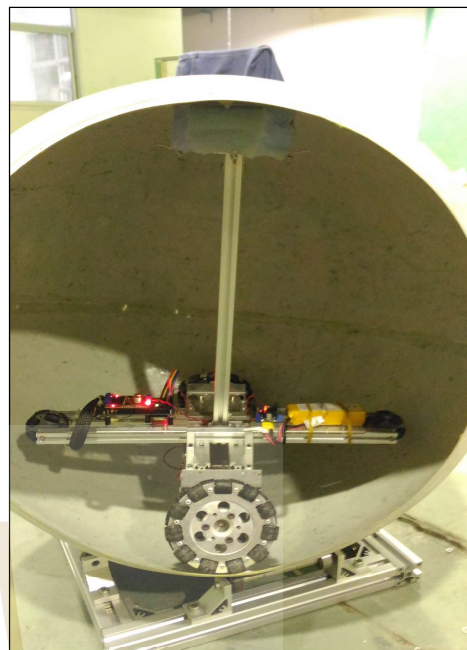
หลังจากทำการติดตั้งอุปกรณ์ทุกชิ้นเข้าด้วยกันแล้วผู้พัฒนา ทดลองเพิ่มมวลลงไปในส่วนของหัว Robot เพื่อที่จะทดสอบแรงดึงดูดของแม่เหล็ก ทำการแก้ไขและเตรียมติดตั้งชุดเก็บข้อมูลส่วนต่อไป



รูปที่ 3.31 Head Ready Use



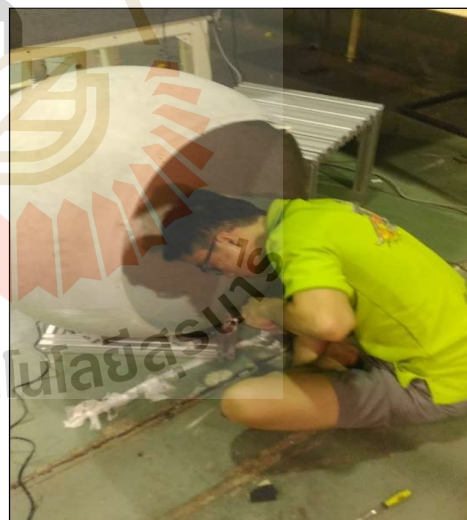
รูปที่ 3.32 เจาะหัวกลวงลดน้ำหนัก



รูปที่ 3.33 เปลี่ยน damping เป็นฟองน้ำ



รูปที่ 3.34 มาร์คส่วนที่พื้นผิวขรุขระ



รูปที่ 3.35 แก้ไขหัวน็อตที่มีปัญหา

Step 0/0 : ปรับปรุงแก้ไขส่วนที่พบปัญหา และประกอบทดสอบประสิทธิภาพต่อไป

บทที่ 4

ผลการศึกษาและวิเคราะห์ผล

4.1 กล่าวนำ

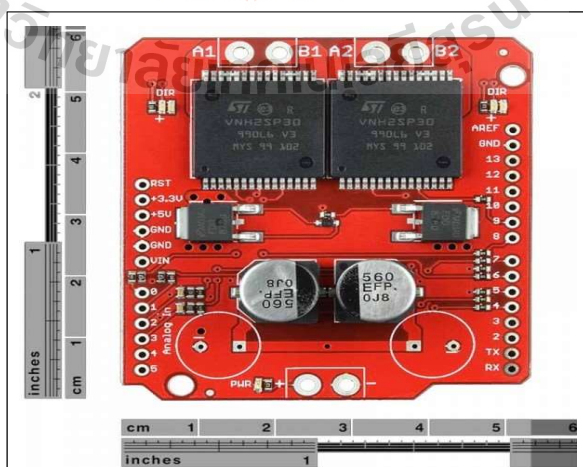
งานวิจัยนี้ได้ทำการสร้างหุ่นยนต์ทรงกลมต้นแบบ เพื่อวิเคราะห์และแก้ไขปัญหการ ออกแบบหุ่นยนต์ประเภทนี้ เพื่อนำไปใช้สำหรับสร้างหุ่นยนต์ โดยทางผู้วิจัยมีแนวคิดที่จะสร้าง ต้นแบบหุ่นยนต์โดยประกอบด้วย การนำข้อมูลที่สังเคราะห์แล้วในบทที่ 3 มาสร้างเพื่อให้ได้มาซึ่งชิ้นงานที่สมบูรณ์ โดยบทนี้ในส่วนแรกผู้วิจัยจะกล่าวถึงประสิทธิภาพของอุปกรณ์ ที่ได้เลือกใช้และในส่วนถัดไป จะเป็นการอภิปรายเกี่ยวกับการทดสอบความเข้ากันได้ ของอุปกรณ์

งานวิจัยในครั้งนี้ ได้ดำเนินการสร้าง และเก็บข้อมูลต่างๆ ภายในอาคารเครื่องมือ 4 ห้อง F4101 มหาวิทยาลัยเทคโนโลยีสุรนารี และได้นำข้อมูลทั้งหมดมาทำการวิเคราะห์ เพื่อหาแนวทางในการสร้างและเพิ่มประสิทธิภาพของหุ่นยนต์ โดยผลการศึกษามีรายละเอียด ดังต่อไปนี้

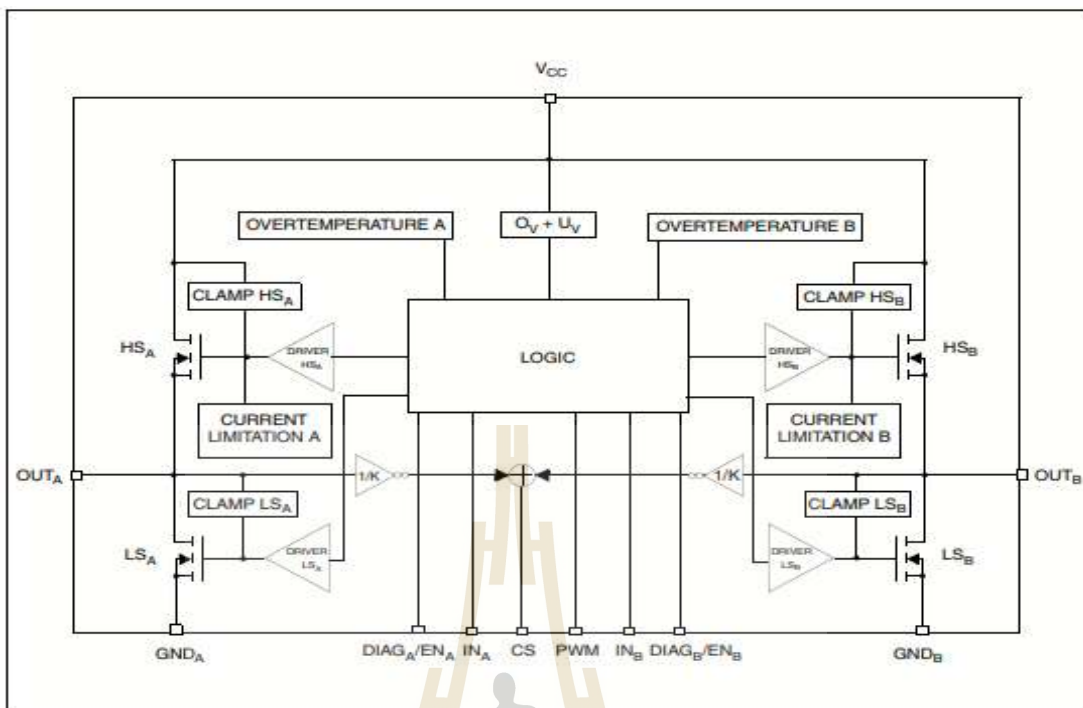
4.2 ศึกษาความสามารถของอุปกรณ์

4.2.1 ทดสอบชุด Drive (Motor Driver Shield Monster Moto VN2SP3)

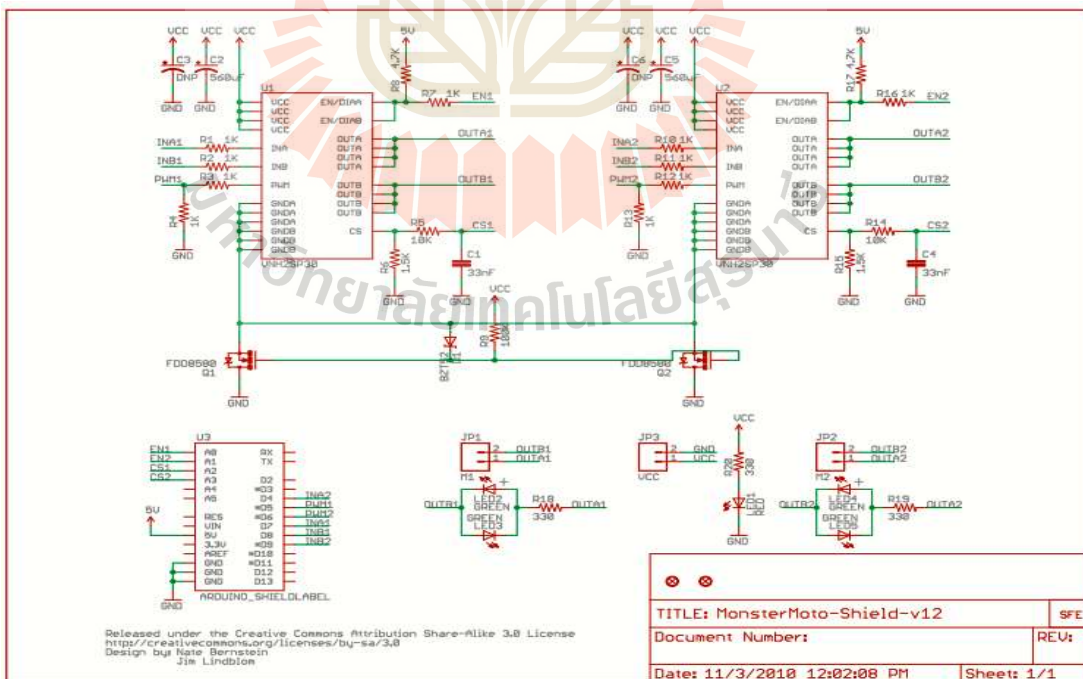
ได้ทำการศึกษาโครงสร้างและระบบการทำงานของบอร์ดไดร์ ดังต่อไปนี้



รูปที่ 4.1 Monster Moto VN2SP3



รูปที่ 4.2 Block Diagram VN12SP3

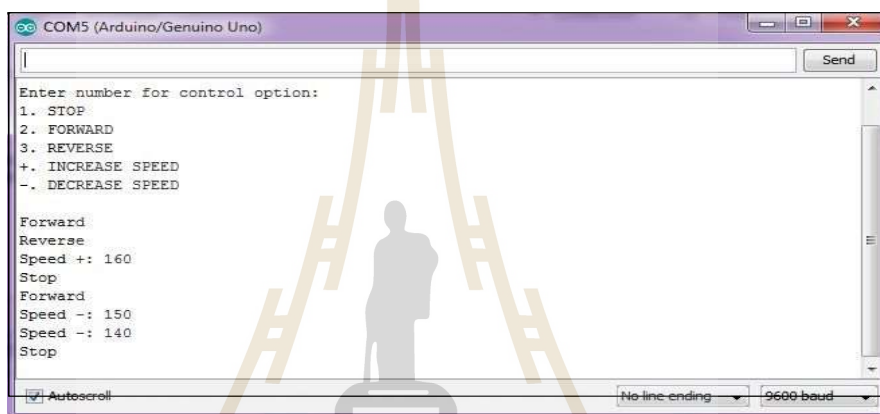


รูปที่ 4.3 Circuit Diagram VN12SP3

4.2.2 กำหนด PIN Motor Drive สำหรับ VNH2SP3

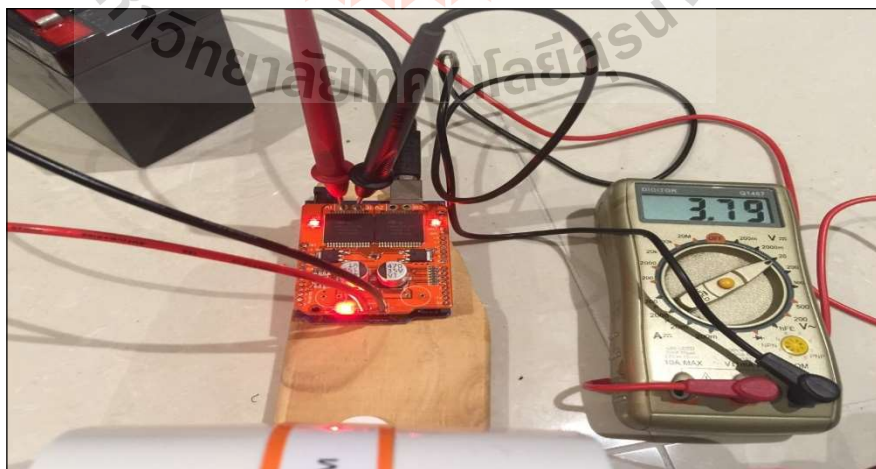
Motor 0	Motor 1
STOP : D7 0, D8 0 & D7 1, D7 1	STOP : D4 0, D9 0 & D4 1, D9 1
CW : D4 1, D9 0	CCW : D4 0, D9 1
CW : D7 1, D8 0	CCW : D4 0, D9 1

4.2.3 เขียนโปรแกรมทดสอบการควบคุม ทิศทาง ความเร็ว

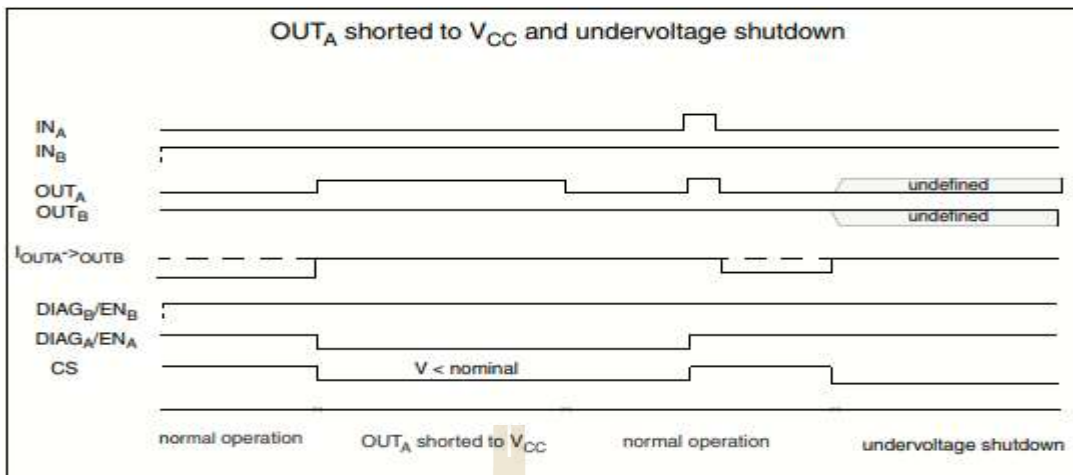


รูปที่ 4.4 โปรแกรมทดสอบ Motor Shield

4.2.4 ทดสอบวัดแรงดันและกระแสไฟฟ้า



รูปที่ 4.5 ทดสอบวัดแรงดันและกระแสไฟฟ้า



รูปที่ 4.6 ตัวอย่าง Waveform Operation

สรุป อุปกรณ์ที่ใช้ทดลอง

- Arduino
- Motor DC 12 V x 2 ตัว
- Monster Moto Shield
- Power Supply

ในการทดลองนี้ สามารถกำหนดทิศทาง ความเร็ว ตามที่ต้องการ ได้อย่างถูกต้องและแม่นยำ

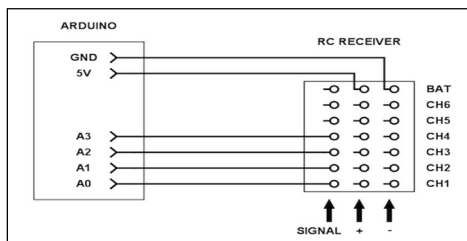
ข้อจำกัด

- Max Voltage is 16.4 V
- Maximum PWM frequency : 20 kHz
- Maximum Current 33 A
- Continuous Current: 14 A

4.2.5 ทดสอบชุด Control



รูปที่ 4.7 ชุด Adiolink-at9s-10ch



รูปที่ 4.8 เชื่อมต่อ Arduino กับ Receiver

ทำการเชื่อมต่อวงจร ดังรูปที่ 4.8 ในส่วนของภาครับสัญญาณและชุดควบคุม (Arduino)

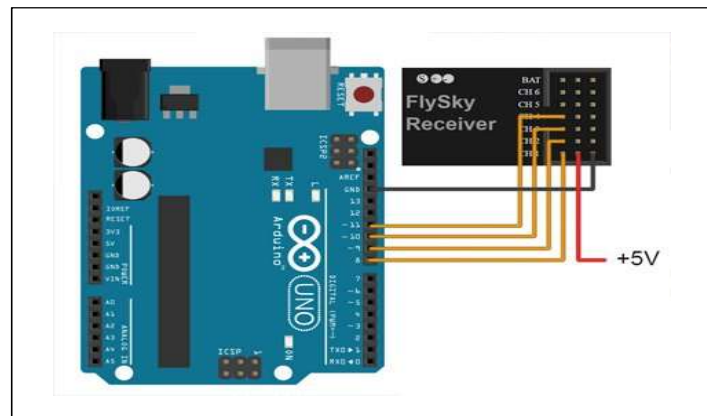
(-) เชื่อมต่อ Arduino GND กับหมุดในแถว GND ของเครื่องรับ

(+) เชื่อมต่อ Arduino V+ กับหมุดในแถว V+ ของเครื่องรับ

(Signal) สายสัญญาณของเครื่องรับ สำหรับป้อนให้ Arduino ใช้สั่งงานต่อไป

```
#include <EnableInterrupt.h> #define SERIAL_PORT_SPEED 57600
#define RC_NUM_CHANNELS 4 #define RC_CH1 0 #define RC_CH2 1
#define RC_CH3 2 #define RC_CH4 3 #define RC_CH1_INPUT A0
#define RC_CH2_INPUT A1 #define RC_CH3_INPUT A2
#define RC_CH4_INPUT A3
uint16_t rc_values[RC_NUM_CHANNELS]; uint32_t rc_start[RC_NUM_CHANNELS];
volatile uint16_t rc_shared[RC_NUM_CHANNELS];
void rc_read_values() { noInterrupts(); memcpy(rc_values, (const void *)
rc_shared, sizeof(rc_shared)); interrupts(); }
void calc_input(uint8_t channel, uint8_t input_pin) {
  if (digitalRead(input_pin) == HIGH) {
    rc_start[channel] = micros(); } else {
  uint16_t rc_compare = (uint16_t)(micros() - rc_start[channel]);
rc_shared[channel] = rc_compare; }}
void calc_ch1() { calc_input(RC_CH1, RC_CH1_INPUT); }
void calc_ch2() { calc_input(RC_CH2, RC_CH2_INPUT); }
void calc_ch3() { calc_input(RC_CH3, RC_CH3_INPUT); }
void calc_ch4() { calc_input(RC_CH4, RC_CH4_INPUT); }
void setup() { Serial.begin(SERIAL_PORT_SPEED);
  pinMode(RC_CH1_INPUT, INPUT); pinMode(RC_CH2_INPUT, INPUT);
  pinMode(RC_CH3_INPUT, INPUT); pinMode(RC_CH4_INPUT, INPUT);
enableInterrupt(RC_CH1_INPUT, calc_ch1, CHANGE);
enableInterrupt(RC_CH2_INPUT, calc_ch2, CHANGE);
enableInterrupt(RC_CH3_INPUT, calc_ch3, CHANGE);
enableInterrupt(RC_CH4_INPUT, calc_ch4, CHANGE);}
void loop() { rc_read_values(); Serial.print("CH1:");
Serial.print(rc_values[RC_CH1]); Serial.print("\t"); Serial.print("CH2:");
Serial.print(rc_values[RC_CH2]); Serial.print("\t"); Serial.print("CH3:");
Serial.print(rc_values[RC_CH3]); Serial.print("\t"); Serial.print("CH4:");
Serial.println(rc_values[RC_CH4]); delay(200);}
```

รูปที่ 4.9 ทดสอบรับค่าด้วย Interrupt



รูปที่ 4.10 วงจร Arduino & Receiver

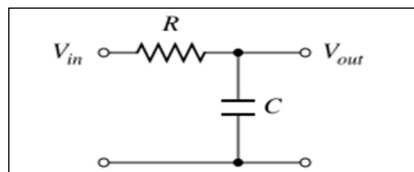
4.2.6 ทดลองการรับค่าจาก Remote

```

Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1496 Channel 3: 1752 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1752 Channel 4: 1496
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1496 Channel 3: 1752 Channel 4: 1500
Channel 1: 1196 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1188 Channel 2: 1500 Channel 3: 1752 Channel 4: 1496
Channel 1: 1184 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1496
Channel 1: 1192 Channel 2: 1496 Channel 3: 1752 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192 Channel 2: 1500 Channel 3: 1756 Channel 4: 1500
Channel 1: 1192

```

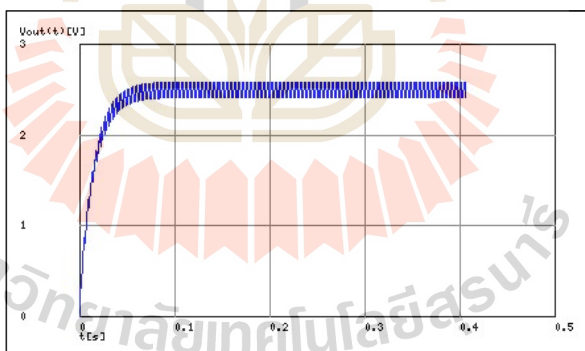
รูปที่ 4.11 ค่าจาก Remote ที่ Arduino รับได้



รูปที่ 4.12 วงจร RC Low Pass

ซึ่งในการทดสอบผู้พัฒนาอยากทราบว่า ถ้าต้องการนำเอาสัญญาณจากชุดรับสัญญาณนำไปใช้งานเลย โดยที่ไม่ผ่านชุดไมโครคอนโทรเลอร์ ด้วยวงจรรูปที่ 4.9 ใช้ capacitor 2.2 μF ตัวต้านทานขนาด 15K โอห์ม ทำการป้อนสัญญาณและวัดค่าที่ได้

จากการทดลองดังกล่าวพบว่าค่าที่ออกมามีการ Swing ขึ้นลงอยู่ตลอดเวลาอันเป็นผลมาจากภาครับวิทยุ มีความถี่และคลื่นรบกวนในอากาศแทรกเข้ามาด้วยซึ่งถ้าเปรียบเทียบกับรูปที่ 4.10 และค่าที่อ่านได้ในรูปที่ 4.9 จะเห็นความคลาดเคลื่อนของช่วงสัญญาณที่มีการเปลี่ยนแปลงซึ่งส่งผลต่อระบบการควบคุมเพียงเล็กน้อย ดังนั้นค่าความคลาดเคลื่อนที่เกิดขึ้นจึงไม่ส่งผลกระทบต่อระบบ ถ้าหากสัญญาณแอมป์ขณะที่ระบบ Standby เราสามารถเขียนโปรแกรมเพื่อรองรับ Gap ได้



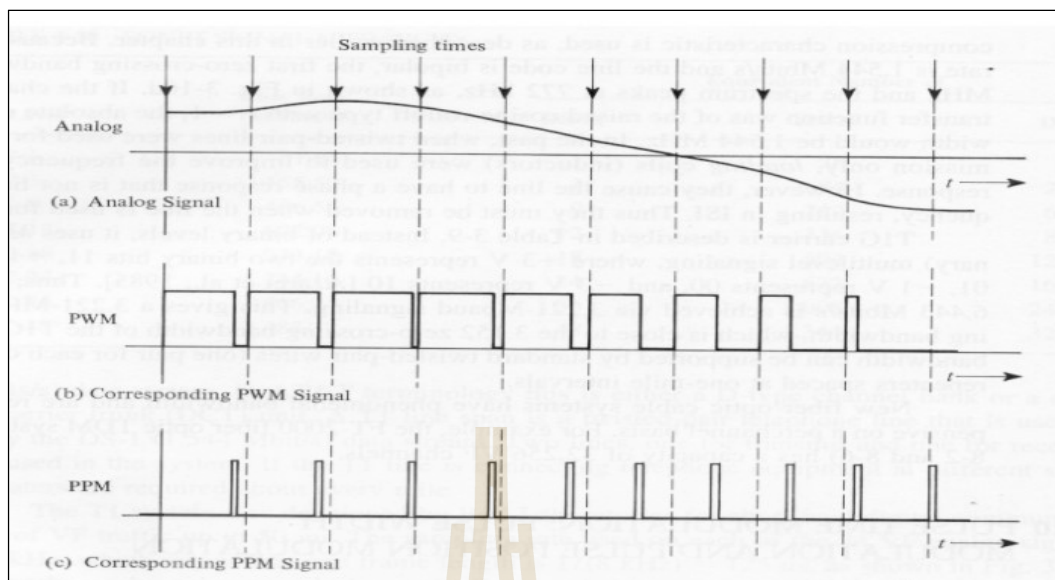
รูปที่ 4.13 สัญญาณเมื่อผ่านวงจร RC Low Pass

รูปแบบการใช้งานคำสั่ง : [pulseIn (pin, value ,timeout)]

โดยที่ pin คือขาดิจิตอลอินพุตที่ต้องการอ่านค่า

value คือลอจิกที่จะอ่านค่าความกว้างพัลส์ออกมา ในที่นี้จะอ่านที่ลอจิก 1

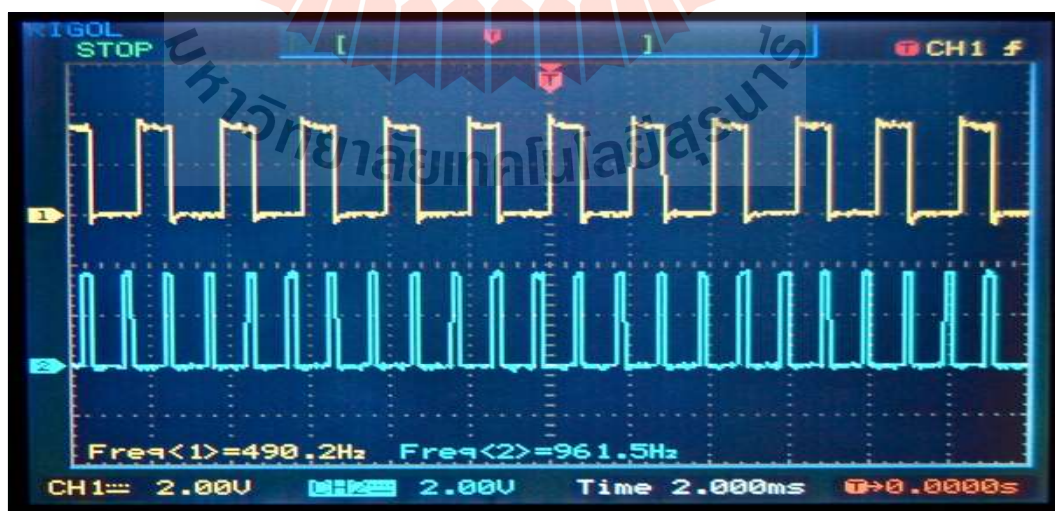
timeout คือช่วงเวลาที่กำหนดว่า ถ้าหากไม่เกิดพัลส์ ขึ้นมาในช่วงเวลาที่ไมโครวินาทีแล้วจะสั่งให้ไปทำงาน ในคำสั่งถัดไป



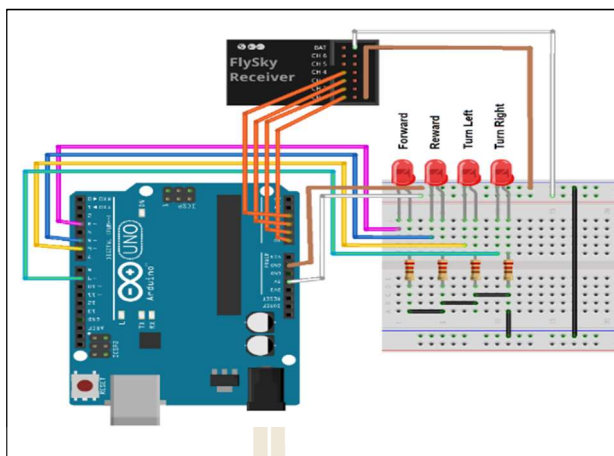
รูปที่ 4.14 Signal Analysis

4.2.7 วิเคราะห์ค่าที่รับเข้ามาได้

รูปแบบสัญญาณที่รับได้เป็นลักษณะของ PWM ซึ่งย่อมาจาก Pulse Width Modulation และ PPM ย่อมาจาก Pulse Position Modulation PWM เป็นเทคนิคที่ใช้ในการส่งข้อมูลในรูปแบบของความกว้างชีพจร (Pulse) ที่แตกต่างกันออกไปในแต่ละช่องสัญญาณ (Channel)



รูปที่ 4.15 Signal from Oscilloscope PWM



รูปที่ 4.16 วงจรทดสอบด้วยหลอด LED

4.2.7.1 ทดลองเขียนโปรแกรมคุม LED

อุปกรณ์ที่ใช้ทดลอง Arduino, Adiolink at9s 10 ch และ LED 4 ดวง

ในการทดลองนี้สามารถควบคุมสถานะ LED จาก PWM ที่รับเข้ามาได้จากการดักค่าโปรแกรมตัวอย่างที่ใช้ในการดักค่า Pulse จากภาครับสัญญาณ

```
int ch1,ch2,ch3;
int count=0;
void setup() {
  pinMode(5, INPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  Serial.begin(9600);
}
void loop() {
  ch1 = pulseIn(5, HIGH, 25000);
  ch2 = pulseIn(6, HIGH, 25000);
  ch3 = pulseIn(7, HIGH, 25000);
  Serial.print("No. ");
  Serial.print(count);
  Serial.print(" Channel 1 : ");
  Serial.print(ch1);
  Serial.print(" Channel 2 : ");
  Serial.print(ch2);
  Serial.print(" Channel 3 : ");
  Serial.print(ch3);
  Serial.println();
  count++;
}
```

รูปที่ 4.17 โปรแกรมอ่านค่าจาก Remote

```

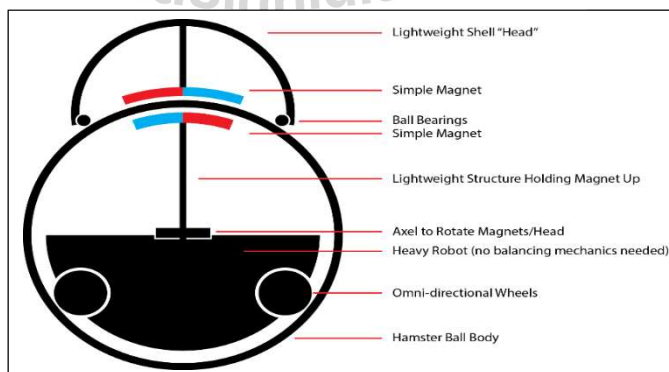
int ch1;
int ch2;
int ch3;
int count = 0;
void setup() {
  pinMode(5, INPUT);
  pinMode(6, INPUT);
  pinMode(7, INPUT);
  Serial.begin(9600);
}
void loop() {
  ch1 = pulseIn(5, HIGH, 25000);
  ch2 = pulseIn(6, HIGH, 25000);
  ch3 = pulseIn(7, HIGH, 25000);
  if(ch1>1000){
    Serial.println("Left Switch: Engaged");
  }
  if(ch1<1000){
    Serial.println("Left Switch:Disengaged");
  }
  Serial.print("Right Stick X:");
  Serial.println(map(ch3, 1000,2000, -500,500));
  Serial.print("Right Stick Y:");
  Serial.println(map(ch2, 1000,2000, -500,500));
  Serial.println();
  delay(100);
}

```

รูปที่ 4.18 โปรแกรมทดลองใช้งาน Data ที่รับเข้ามาจาก Remote

หลังจากที่สามารถเรียกใช้สัญญาณและนำไปควบคุม LED และแสดงออกหน้า Serial Monitor ได้แล้ว ลำดับต่อไปคือการสร้าง Algorithm สำหรับการควบคุมหุ่นยนต์ซึ่งจะแสดงในภาคผนวกต่อไป

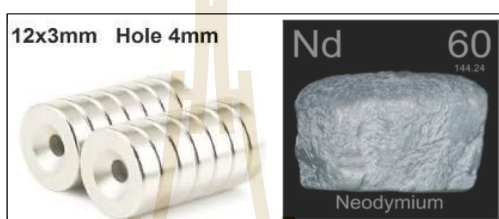
4.2.8 ทดสอบชุดแม่เหล็ก



รูปที่ 4.19 การวางแม่เหล็ก



รูปที่ 4.20 Head robot Mask #IV



รูปที่ 4.21 Neodymium Magnate

ในงานวิจัยนี้ผู้วิจัยเลือกใช้แม่เหล็ก Neodymium Magnate เป็นตัวหลักในการทำให้ส่วนหัว และส่วนตัวของหุ่นยนต์ติดกัน และทำงานได้ทุกสภาวะไม่ว่าจะเป็นขณะหยุดนิ่งหรือมีการเคลื่อนที่ แม่เหล็กนีโอ ดิเมียม ผลิตจาก แร่ นีโอดีเมียม (Neodymium) , แร่เหล็ก (Iron) , แร่ โบรอน (Boron) โดยผลิตให้โครงสร้างของเนื้อโลหะผสม (Alloy) เป็นแบบ Tetragonal Crystalline ค่ากำลังสนามแม่เหล็ก (Remanence) 1 - 1.4 T Coercivity 750-2000 kA/m ค่าความหนาแน่นของพลังงานแม่เหล็ก (Energy Product) 200-400 kJ/cu.m กำลังของแม่เหล็กนีโอดีเมียม มากเพียงพอจะสามารถยกเหล็กที่มีหนักมากกว่าตัวเอง 1,000 เท่า ได้

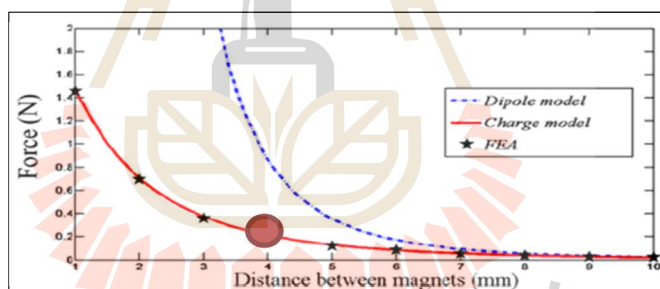


รูปที่ 4.22 เตรียมพื้นผิวติดแม่เหล็ก



รูปที่ 4.23 Neodymium Test

รูปนี้ แสดงให้เห็นถึงกำลังของแม่เหล็กนีโอดิเมียม ได้อย่างชัดเจน ลูกเหล็กทรงกลมขนาดเส้นผ่าศูนย์กลาง 3.63 เซนติเมตร น้ำหนัก 196.1 กรัม ลูกเหล็กทรงกลมทั้งสามลูกเชื่อมต่อกันด้วยแม่เหล็กนีโอดิเมียม มีขนาดเส้นผ่าศูนย์กลาง 4 มิลลิเมตรหนา 1.5 มิลลิเมตร น้ำหนักก้อนละ 0.143 กรัม สามารถแขวนอยู่ได้โดยไม่หลุดออกจากกัน ซึ่งเป็นแม่เหล็กที่ทรงพลังที่สุดในโลก ณ ขณะนี้ จากการศึกษาลงมือทดลองพบว่าระยะที่ดีที่สุดคือช่วง 3.5 – 4.5



รูปที่ 4.24 กราฟการทดสอบระยะห่างกับแรงดึงดูด

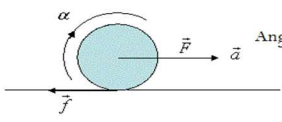


รูปที่ 4.25 ความเข้ากันได้ของส่วนประกอบ

4.3 วิเคราะห์ข้อมูล

ในส่วนนี้จะเป็นการถึงวิธีการเคลื่อนที่ของหุ่นยนต์ ระยะเวลาการทำงาน และแบตเตอรี่

4.3.1 การเคลื่อนที่

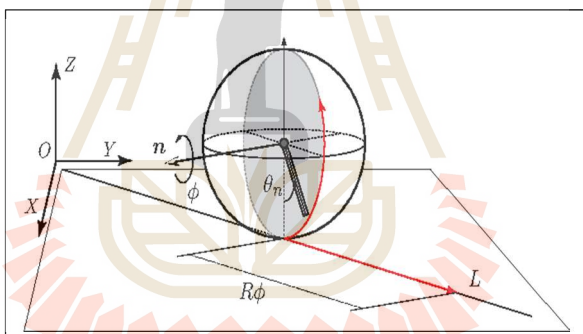


Linear motion $F - f = ma$
 Angular movement $fr = I\alpha$
 if not slip $\alpha = \frac{a}{r}$
 $\frac{fr^2}{I} = a$
 $F - f = \frac{mfr^2}{I}$

$$f = \frac{F}{1 + \frac{mr^2}{I}}, \quad a = \frac{Fr^2}{I + mr^2}$$

If there is no other force $F = 0$
 $\Rightarrow a = 0, f = 0$ Speed will be constant and no friction.

รูปที่ 4.26 สมการการกลิ้ง



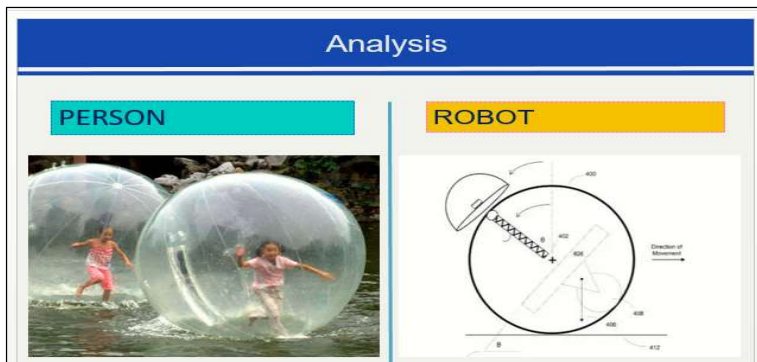
รูปที่ 4.27 ทิศทางการเคลื่อนที่

$$K_{rolling} = \frac{1}{2} I_{CM} \omega^2 + \frac{1}{2} m v_{CM}^2 \quad \text{Kinetic energy of rolling}$$

From $v = \omega R$ That will $K_{rolling} = \frac{1}{2} I_{CM} \left(\frac{v_{CM}}{R} \right)^2 + \frac{1}{2} m v_{CM}^2$

To rewrite $K_{rolling} = \frac{1}{2} \left(\frac{I_{CM}}{R^2} + M \right) v_{CM}^2$ energy of rolling

รูปที่ 4.28 สมการการหมุน



รูปที่ 4.29 การเคลื่อนที่จากข้างใน

มีแรงกระทำต่อ Ball และ Robot 3 แรง คือ

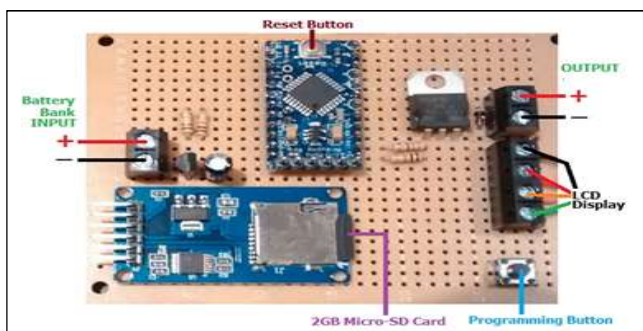
1. แรงเสียดทานระหว่างล้อ Robot กับผนัง Shell
2. แรงเนื่องจากน้ำหนักของ Robot
3. แรงปฏิกิริยาในแนวตั้งจากกับผิวสัมผัส ที่ผนัง Shell กระทำต่อล้อของ Robot

ทั้ง 3 ส่วนนี้ เมื่อนำมารวมกันแล้วผลรวมภายในเอาชนะแรงเสียดทานภายนอก Shell ของลูกบอลได้นั้นสามารถเคลื่อนที่ไปข้างหน้าได้

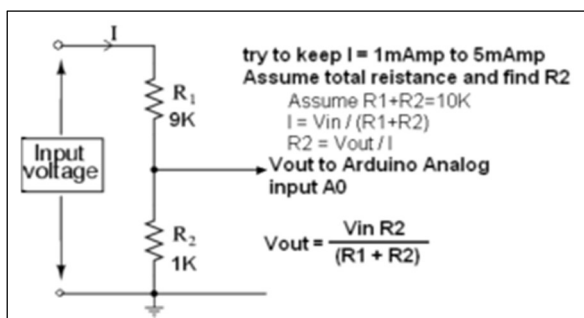
4.3.2 ระยะเวลาการใช้งาน

ภายหลังจากการประกอบทุกส่วนเข้าด้วยกันแล้วผู้พัฒนาได้ทำการทดสอบวัดค่าพลังงานในการใช้งานของหุ่นยนต์ โดยแบ่งการทดสอบในสภาวะเปิดออกเป็น 5 การทดลอง โดยบังคับหุ่นยนต์ และวัดค่าพลังงานที่สูญเสียไปขณะหุ่นยนต์เคลื่อนที่ ดังนี้

4.3.2.1 สร้างอุปกรณ์เก็บค่าพลังงาน โดยบันทึกผลลง SD Card



รูปที่ 4.30 Record Voltage to SD Module



รูปที่ 4.31 วงจรแบ่งแรงดันไฟฟ้า

4.3.2.1.1 การวัดแรงดันไฟฟ้า DC

ในการวัดแรงดันสามารถทำได้โดย การใช้ Arduino internal ADC แรงดันอ้างอิงภายในของ Arduino internal ADC คือ 5V ($V_{ref} = 5V$) ดังนั้นแรงดันไฟฟ้าสูงสุดที่เราสามารถวัดได้โดยไม่ต้องใช้วงจรภายนอกคือ 5V มีความละเอียด 10 บิต คือ $2^{10} = 1024$ ค่าสำหรับระดับ 0 ถึง 5V. ให้ 0V เทียบเท่ากับการอ่านค่า ADC 0 และ 5V เท่ากับ 1023 ค่า ADC เดียวคือ 4.88mV เช่น $1 = 4.88mV$ เพื่อวัดแรงดันไฟฟ้าที่สูงกว่า 5V เราจำเป็นต้องใช้ตัวแปลงแรงดันภายนอกเพื่อให้ตรงกับข้อกำหนดของ ADC ซึ่งจำเป็นต้องใช้แรงดันไฟฟ้าตั้งแต่ 0V ถึง 5V สามารถสร้างได้โดยใช้ตัวต้านทานสองตัวดังแสดงใน รูปที่ 4.31 ในที่นี้เราจะทำการวัดตั้งแต่ 0V ถึง 50V DC ในตัวอย่างนี้ แบตเตอรี่จะใช้เป็นแหล่งแรงดันไฟฟ้าที่จะวัดคุณสมบัตินี้ได้สูงสุด 50V DC เซ็นเซอร์แรงดันไฟฟ้าตามท้องตลาดมีเป็นเพียงวงจรแรงดันไฟฟ้า divider สูตรการอ่านแรงดันไฟฟ้าคือ

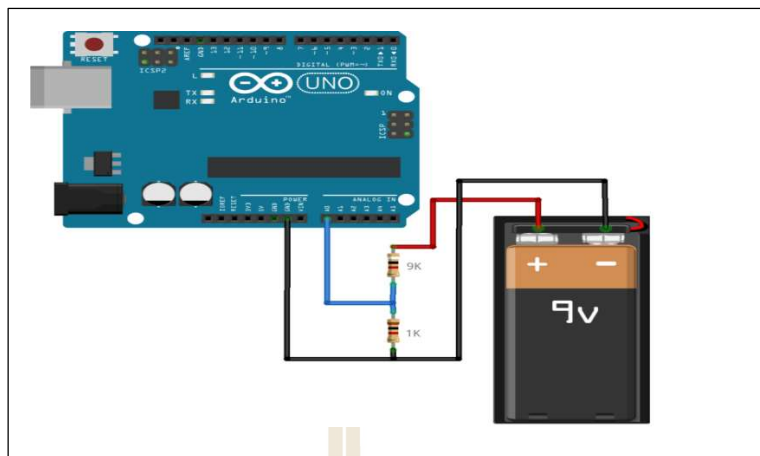
$$\text{แรงดันไฟฟ้า} = \text{ADC_value} \times (V_{ref} / 1024) \times \text{Volt สูงสุดที่แหล่งจ่าย}$$

$$\text{จาก : } V_{REF} = 5V$$

$$\text{ADC_Value เป็นจำนวนเต็ม AnalogRead}$$

$$\text{อัตราส่วนหารคืออัตราส่วนแรงดันไฟฟ้า} = V_{in} / V_{out}$$

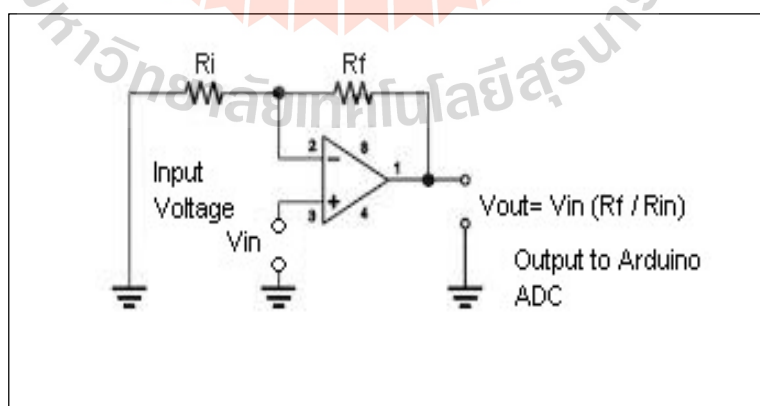
Arduino สามารถวัดแรงดันไฟฟ้าได้โดย อ้างอิงกับขา Gnd ส่วนประกอบที่ใช้สำหรับการวงจรวัดแรงดันไฟฟ้ากระแสตรง 9K, 1K Resistors วงจรวัดแรงดันไฟฟ้า Arduino DC วงจรประกอบด้วยเพียงสองตัวต้านทาน ที่ใช้เป็นตัวหารแรงดันไฟฟ้า เมื่อวัดแรงดันไฟฟ้าตกค้างในตัวต้านทานโอห์ม 1K ซึ่งเป็นค่า 5V เมื่อเราใช้แรงดันไฟฟ้า 50V ผ่านตัวหารแรงดันไฟฟ้ากระแสไฟที่ไหลผ่านตัวต้านทานคือ $I = V / (R1 + R2)$ ในกรณีที่แรงดันไฟฟ้าสูงสุดที่วัดได้คือ 50V ดังนั้นค่า $I = 50$ สูงสุดที่เชื่อมต่อกันระหว่างแบตเตอรี่ $9V / 10K = 5 \text{ mAmp}$



รูปที่ 4.32 การต่อวงจรแบ่งแรงดันไฟฟ้า

4.3.2.1.2 ตัวอย่างการวัดแรงดันไฟฟ้านี้เราจะวัดแรงดันไฟฟ้า

ของแบตเตอรี่ DC 9V คุณสามารถเชื่อมต่อแหล่งสัญญาณ DC ใด ๆ ตัวอย่างการวัดแรงดันไฟฟ้านี้สามารถวัดได้สูงสุด 50 โวลต์ ซึ่งจะเป็ประโยชน์สำหรับการวัดแรงดันไฟฟ้า ในส่วนของอะแดปเตอร์หรือแรงดันไฟฟ้าของแบตเตอรี่ มีวงจรป้องกันแรงดันไฟฟ้าเกิน สามารถสร้างขึ้นได้ โดยการเพิ่ม zener ไดโอด ผ่านตัวต้านทาน 1K สำหรับการวัดมิลลิโวลต์ เครื่องขยายสัญญาณแบบป้องกันการย้อนกลับของกระแสไฟฟ้าดังที่แสดงในรูปที่ 4.33 แอมพลิฟายเออร์ Inverting



รูปที่ 4.33 แอมพลิฟายเออร์แบบ Inverting LM358

```

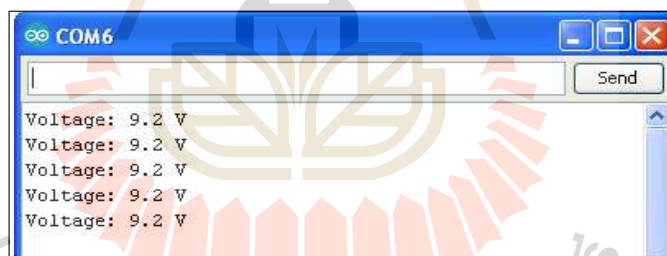
Float VoltIn = A0;
void setup() {
  Serial.begin(9600);
}
void loop() {
  int sensorValue = analogRead(VoltIn);
  float voltage = sensorValue * (5.0 / 1024.0) * 10;
  Serial.print("Voltage: ");
  Serial.print(voltage);
  Serial.println(" V");
  delay(1000);
}

```

รูปที่ 4.34 โปรแกรมอ่านค่าแรงดันไฟฟ้ากระแสตรง

4.3.2.1.3 ผลการวัดแรงดันไฟฟ้ากระแสตรง

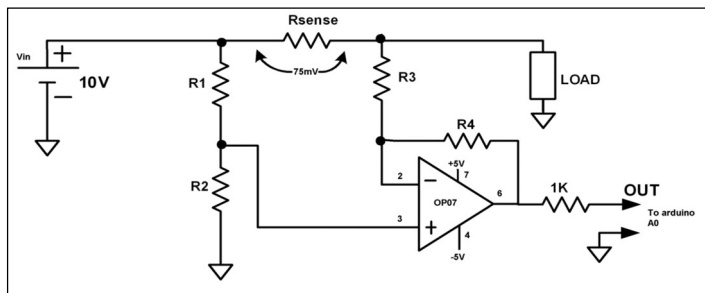
จากรูปที่ 4.34 ทำการเขียนโปรแกรมเพื่อทดสอบการวัดแรงดันไฟฟ้า 0 ถึง 50V DC เปิด หลังจากนั้นเปิด Serial Monitor จาก (Tools >> Serial Monitor) ตั้งอัตราการถ่ายโอนข้อมูลไปที่ 9600 ดูแรงดันไฟฟ้าในจอภาพแบบอนุกรม



รูปที่ 4.35 ค่าแรงดันไฟฟ้ากระแสตรง

4.3.2.2 การวัดกระแสไฟฟ้ากระแสตรง

ในการออกแบบมิเตอร์มิเตอร์แบบดิจิตอลเป็นกระบวนการแปลงโวลต์มิเตอร์เป็น Amp Meter เรารู้ว่า $V = I \times R$ เราสามารถนำเอาค่าจากแรงดันที่ตกคร่อมตัวต้านทาน (shunt) เป็นสัดส่วน โดยตรงกับกระแส ($V = I$) ในแอมมิเตอร์ความต้านทานนี้เรียกว่า Shunt ความต้านทานการหัดตัวมีค่าน้อยมากและจะไม่กระทบต่อแรงดันไฟฟ้าของโหลด โดยทั่วไปแล้วจะอยู่ที่ 75mV shunt ซึ่งมีขายทั่วไปตามท้องตลาด



รูปที่ 4.36 วงจรการวัดกระแสโดยใช้ตัวต้านทานแบบทแยงมุม

ตัวอย่างนี้ วัดกระแส 50Amp โดยใช้ 75mV shunt (Rsense) Shunt ให้แรงดันไฟฟ้า 75mV ข้ามมันเมื่อเราผ่านปัจจุบันของ 50Amp Arduino ใช้เวลา 5V เป็น input สำหรับ ADC ดังนั้นเครื่องขยายเสียง Differential จะแปลง 75mV เป็น 5V การคำนวณสำหรับ 50Amps 75mV Shunt โดยที่ Vo เป็นแรงดันที่กำหนด ADC (A0) ที่ตัวอุปกรณ์ ตัวต้านทานแรงดันขนาด 25Amps / 75mV Shunt Resistor OpAmp OP07 ,66.66K, 1K Resistors ซึ่งจะได้ว่า

$$R_{in} = R1 = R3 \quad R_f = R2 = R4$$

$$V_{in} = 75mV \quad V_o = V_{in} * (R_f / R_{in})$$

จาก : $V_{in} = 75mV$ และ $V_o = 5V$

สมมติ : $R_{in} = 1K$ (สันนิษฐานว่า Rin อยู่ในช่วง 1K ถึง 50K)

ค้นหา : $R_{in} = ?$

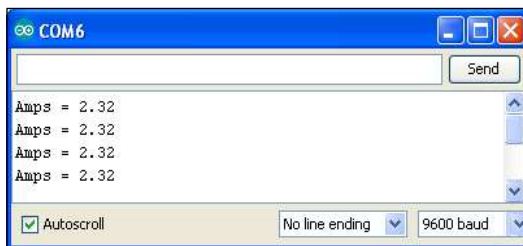
$$R_{in} = R1 = R3 = 66.66K$$

$$R_f = R2 = R4 = 1K$$

ดังนั้น สามารถใช้ตัวต้านทานมาตรฐานและคำนวณย้อนกลับได้ในโค้ดดังนี้

```
const float ShuntAmps = 50
void setup() { Serial.begin(9600); }
void loop() {
  int sensorValue = analogRead(A0);
  float current = sensorValue * (ShuntAmps / 1024.0);
  Serial.print("Current:");
  Serial.print(current);
  Serial.println(" Amps");
  delay(1000);
}
```

รูปที่ 4.37 โปรแกรมวัดกระแสโดยใช้ตัวต้านทานแบบทแยงมุม



รูปที่ 4.38 ค่าการวัดกระแสไฟฟ้า

4.3.2.2.1 ผลการวัดแรงดันไฟฟ้ากระแสตรง

ทำการเปิด Serial Monitor ทำการตั้งอัตราการถ่ายโอนข้อมูลที่ 9600 ดูแรงกระแสไฟฟ้าในจอภาพแบบอนุกรมดังรูปที่ 4.38 หลังจากนั้นแก้ไขโปรแกรมต่อไป

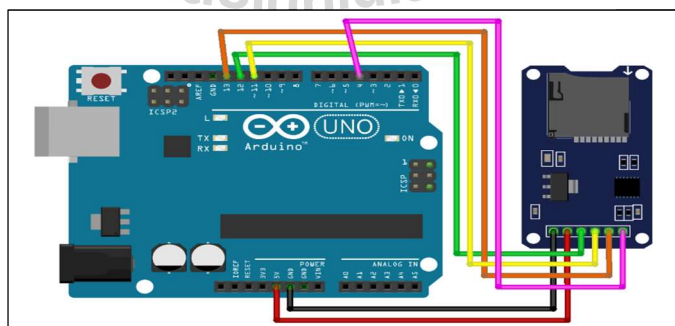
4.3.2.2 การเก็บค่าลง SD Card

Micro SD Card Module โมดูล SD card รุ่นนี้รองรับการใช้งานการ์ดขนาด (Micro SD) ทำหน้าที่บันทึกข้อมูลลง SD Card เก็บ Data loger โดยใช้ร่วมกับ Arduino ซึ่งมี library มาตรฐาน Input DC 3V – 5V ส่งข้อมูลผ่าน SPI Pin MOSI ,SCK ,MISO ,CS

4.3.2.2.2 ต่อวงจร SD Card กับ Arduino

โดยที่มีการเชื่อมต่อดังนี้

GND	=>	GND	MISO	=>	Pin12
SCK	=>	Pin13	MOSI	=>	Pin11
CS/SDSC	=>	Pin4	VCC	=>	5V



รูปที่ 4.39 วงจร SD Card & Arduino

```

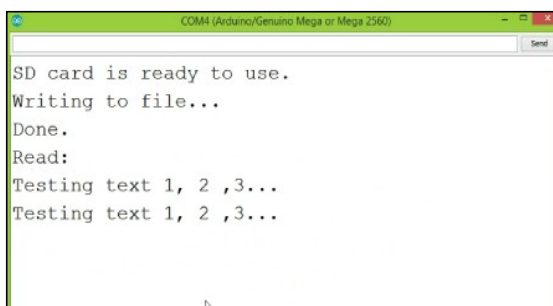
#include <SPI.h>
File myFile;
const int chipSelect = 10;
void setup(){
  Serial.begin(9600);
  while (!Serial) {;}
  Serial.print("Initializing SD card...");
  pinMode(SS, OUTPUT);
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present!");
    return;
  }
  Serial.println("card initialized.");
  myFile = SD.open("datalog.txt", FILE_WRITE);
  if (myFile) {
    Serial.print("Writing to datalog.txt...");
    myFile.println("CodeMobiles.com");
    myFile.close();
    Serial.println("done.");
  } else { Serial.println("error opening datalog.txt"); }
  myFile = SD.open("datalog.txt");
  if (myFile) {
    Serial.println("datalog.txt:");
    while (myFile.available()) { Serial.write(myFile.read()); }
    myFile.close();
  } else { Serial.println("error opening datalog.txt"); }
}
void loop(){ Serial.println("End Commun."); }

```

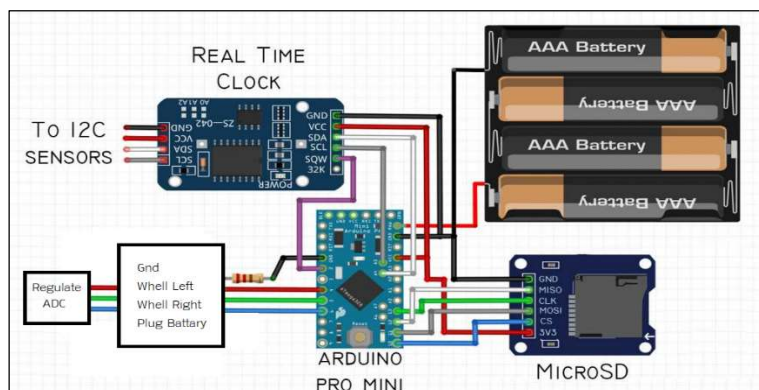
รูปที่ 4.40 โปรแกรมทดสอบการทำงานของ SD Card

4.3.2.2.2 เขียนโปรแกรมทดสอบ SD Card Module ดังนี้

ทำการเขียนโปรแกรมดังรูปที่ 4.40 เพื่อทดสอบว่าสามารถเขียนอ่านโมดูล SD Card หลังจากโหลดโปรแกรมทำการเปิด Serial Monitor ตั้งอัตรา การถ่ายโอนข้อมูลไปที่ 9600 เพื่อดูสถานะการทำงานของ SD Card หลังจากนั้นทำการปรับปรุงแก้ไขโปรแกรมสำหรับทำ Data Logger สำหรับบันทึกข้อมูลลงบน SD Card

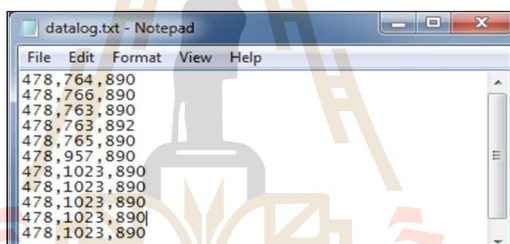


รูปที่ 4.41 ผลการทดสอบ Module SD Card



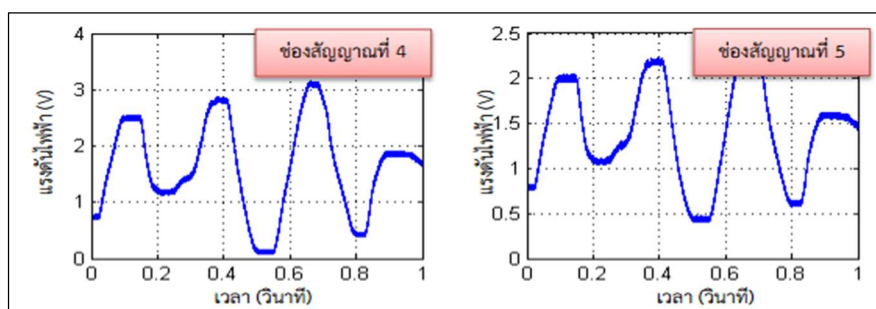
รูปที่ 4.42 Wiring วงจร Data logger

ทดสอบการรับค่าและบันทึกลง SD Card และทดสอบเปิดไฟล์ datalog.txt จาก SD Card

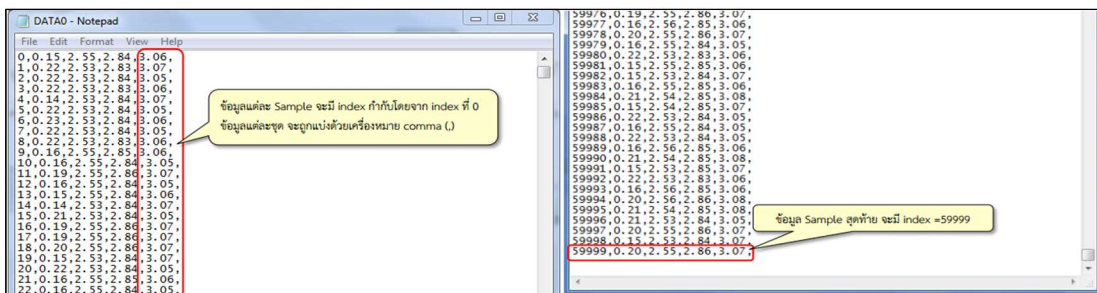


รูปที่ 4.43 ผลการอ่านค่า Analog ทั้ง 3 Pin จากไฟล์ใน SD Card

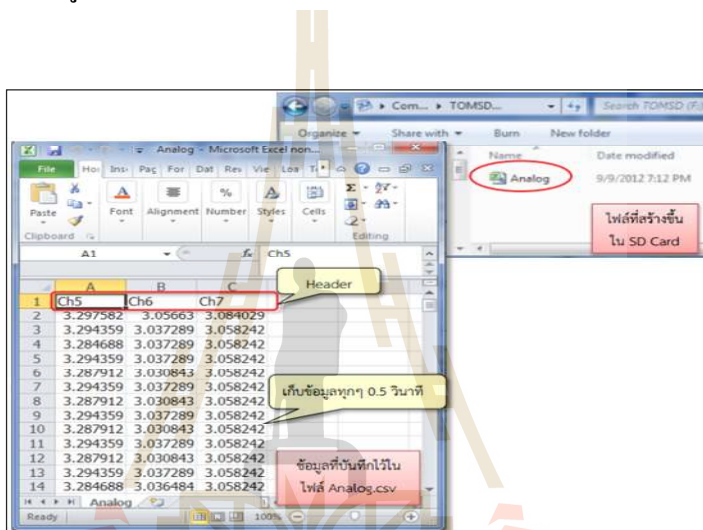
ทำการแก้ไขโปรแกรมให้ทำการ คำนวณก่อนบันทึกค่า Volt ดังรูปที่ 4.35 และทดสอบอ่านไฟล์ DATA.txt และทดสอบนำไปพล็อตกราฟ



รูปที่ 4.44 กราฟผลของ Volt ที่จากไฟล์ใน SD Card



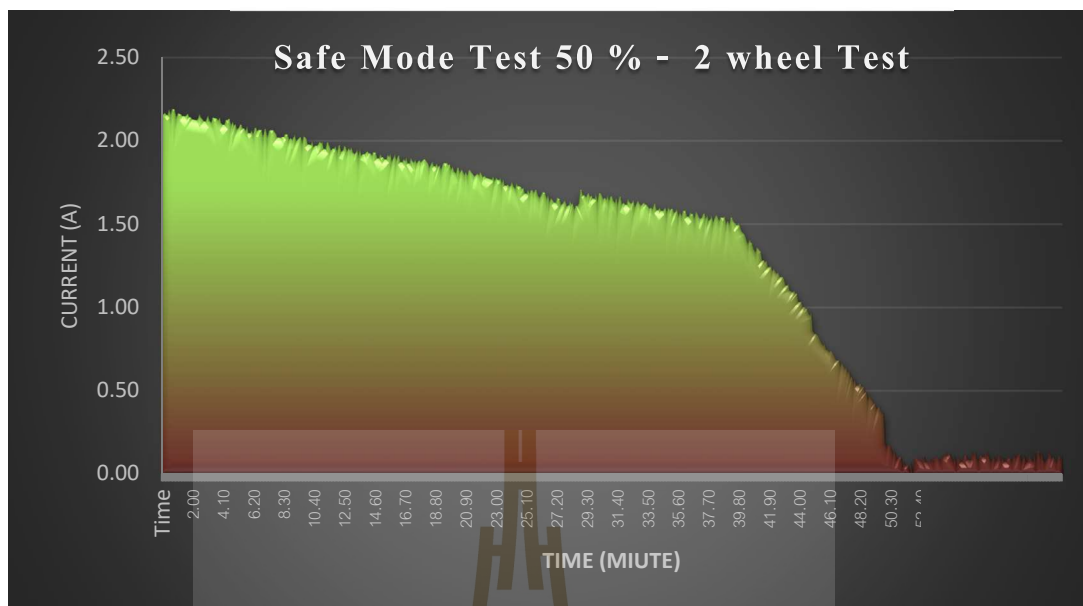
รูปที่ 4.45 ผลการอ่านค่า Amp จากไฟล์ใน SD Card



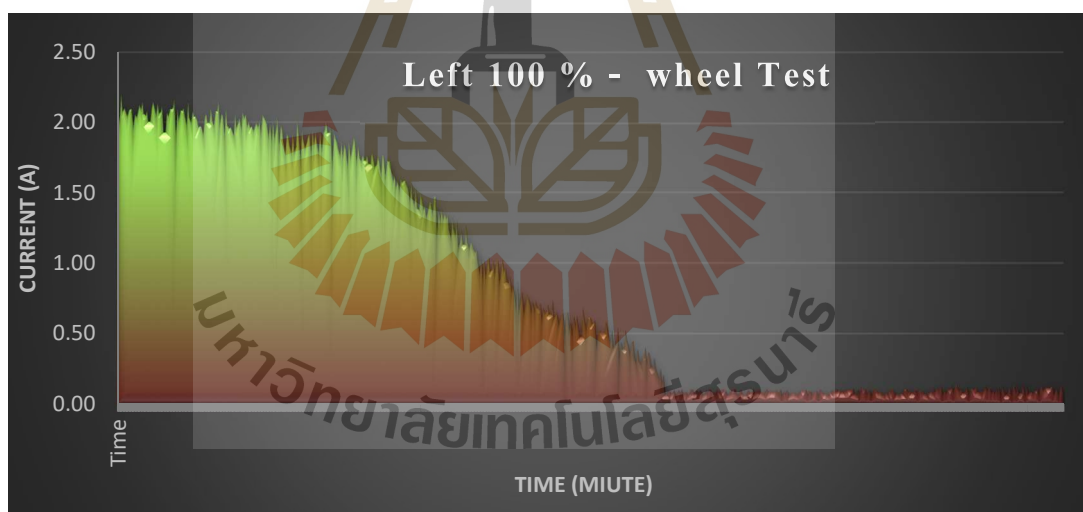
รูปที่ 4.46 ผลการบันทึกข้อมูลและแสดงในหน้าต่าง Excel

พบว่าผลการอ่านค่า Volt ที่ได้ออกมา นั้นจากรูปที่ 4.44 มีการแกว่งขึ้นลงแปรผันตรงตามการควบคุมคันเร่งจากรีโมท ซึ่งช่องสัญญาณที่ 4 คือแรงดันไฟฟ้าตกคร่อมล้อขวา และช่องสัญญาณที่ 5 คือแรงดันไฟฟ้าตกคร่อมล้อซ้าย กราฟที่ได้เกิดจากการบังคับ คันโยกจากรีโมท ไปหน้า – ถอยหลัง สลับกันไปมา สรุปได้ว่าวิธีการวัด Volt ไม่เหมาะกับการนำมาประเมินระยะเวลาช่วงการทำงาน

ทำการแก้ไขโปรแกรมให้ คำนวณก่อนบันทึกค่า Amp ดังรูปที่ 4.38 และทดสอบอ่านไฟล์ DATA0.txt เพื่อใช้สำหรับเก็บผลการทดลอง เนื่องจากพบว่าค่า Volt ที่เก็บได้นั้น นำไปใช้ประเมินหรือคาดการณ์สถานการณ์ ได้ค่อนข้างยากเพราะเวลาที่มีการเร่งแรงดันจะพุ่งขึ้น เมื่อลดคันเร่งลงแรงดันจะลดลงอย่างรวดเร็ว กล่าวคือเหมาะกับการวัดความเร่งของหุ่นยนต์ เทียบกับแรงดัน ดังนั้นค่าที่เหมาะสมที่จะนำมาประเมินประสิทธิภาพระยะเวลาการทำงานของหุ่นยนต์ คือการวัดกระแส (Amp) ดังรูปที่ 4.45 และ 4.46 ที่จะค่อยๆ ลดลงเมื่อมีการใช้งานเทียบกับเวลาการใช้งาน

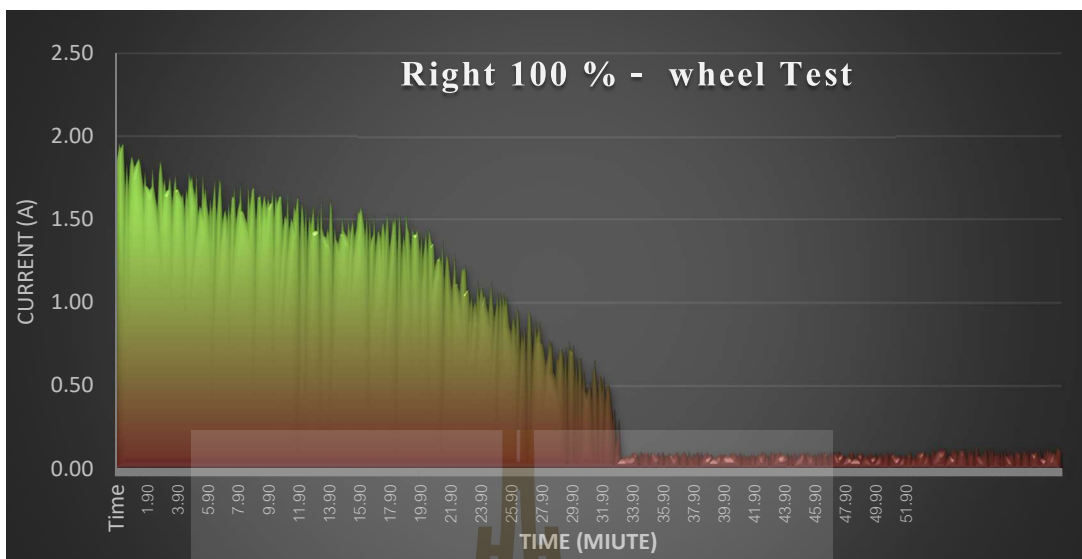


รูปที่ 4.47 กราฟกระแส Motor 2 ตัว ควบคุมการไต่อยู่ที่ 50%

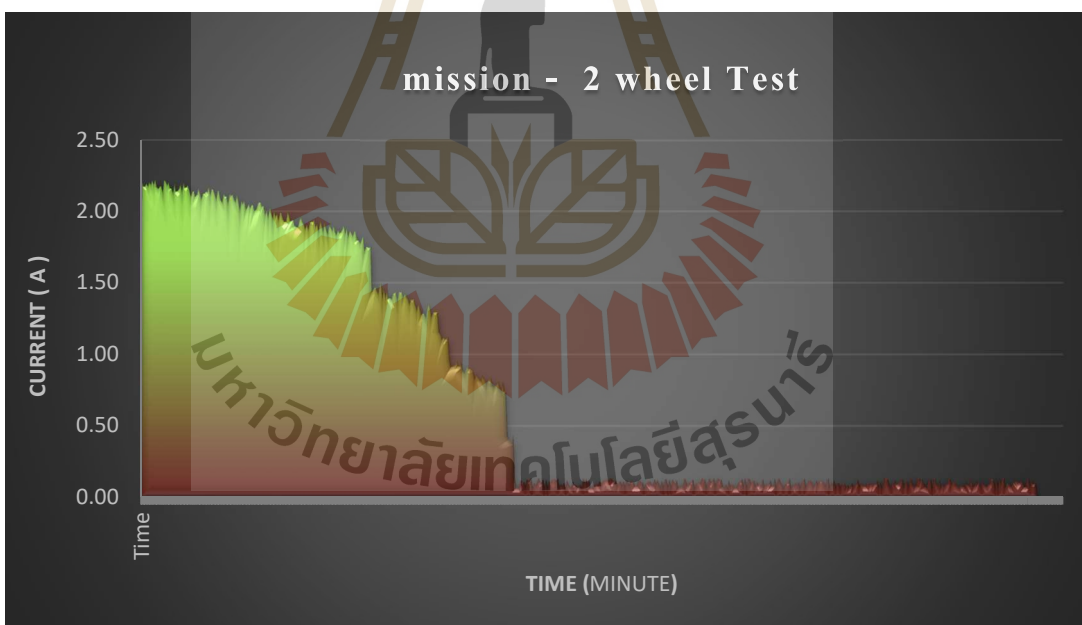


รูปที่ 4.48 กราฟกระแส Motor L ควบคุมการไต่อยู่ที่ 100%

สรุป ระยะเวลาในการเก็บค่า 1 ชั่วโมง รูปที่ 4.47 Motor L ควบคุมการไต่อยู่ที่ 100% ทำงานได้ระยะเวลารวม 35 นาที และรูปที่ 4.48 Motor 2 ตัว ควบคุมการไต่อยู่ที่ 50% แลระยะเวลาเก็บค่า 1 ชั่วโมง Motor 2 ตัว ควบคุมการไต่ที่ 50% สามารถทำงานได้ระยะเวลารวม 48 นาที

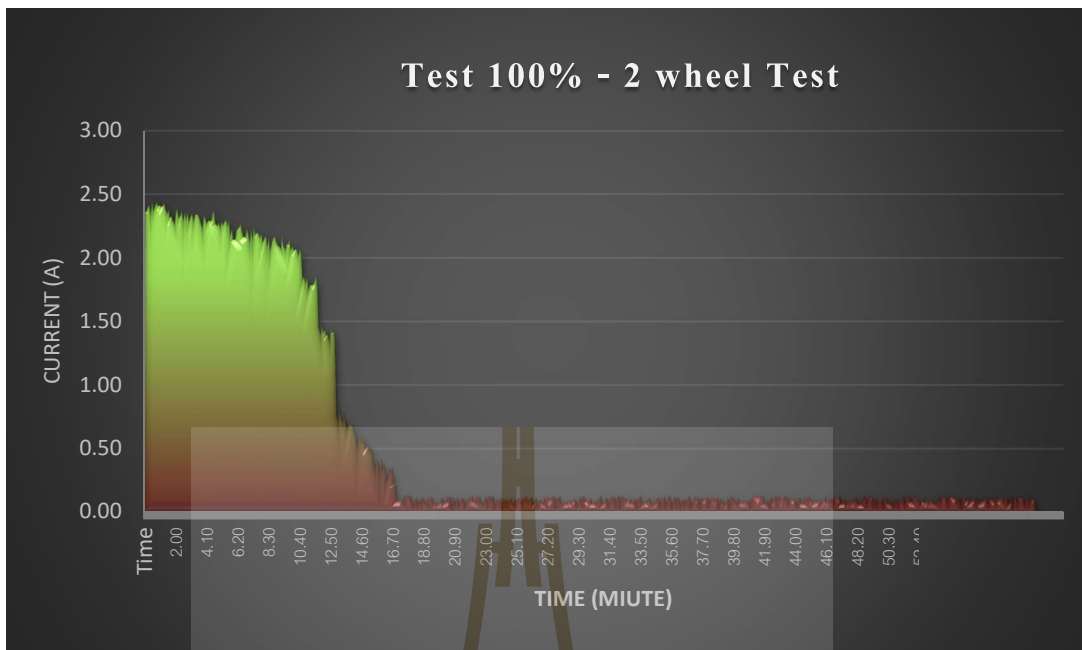


รูปที่ 4.49 กราฟกระแส Motor R ควบคุมการไต่อยู่ที่ 100%



รูปที่ 4.50 กราฟกระแส Motor Mission Test

สรุป ระยะเวลาในการเก็บค่า 1 ชั่วโมง รูปที่ 4.49 Motor R ควบคุมการไต่อยู่ที่ 100% ทำงานได้ 32 นาที รูปที่ 4.50 กระแสที่ Motor 2 ตัว วิ่งตามภารกิจที่กำหนด ระยะเวลาในการเก็บค่ารวม 1 ชั่วโมง Motor 2 ตัว วิ่งตามภารกิจที่กำหนด ทำงานได้ที่ 24 นาที



รูปที่ 4.51 กราฟกระแส Motor 2 ตัว ควบคุมการไต่อยู่ที่ 50%

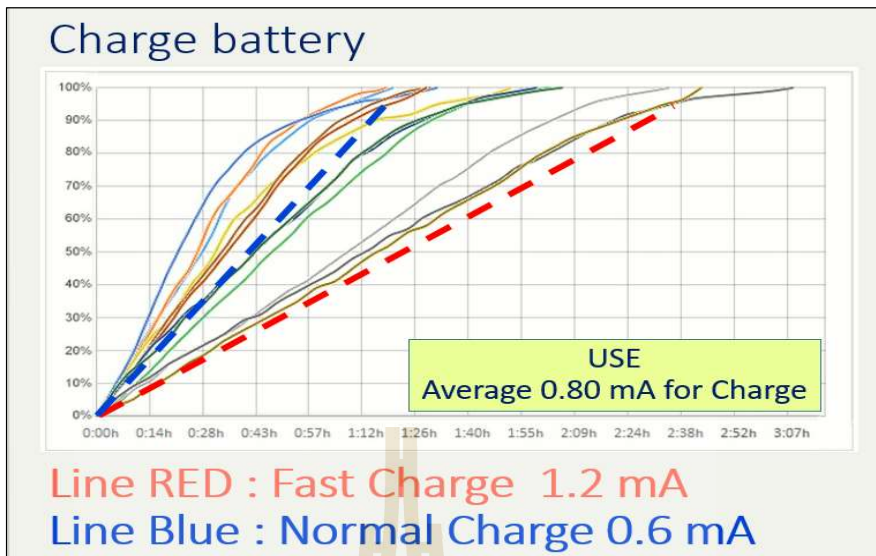
สรุป ระยะเวลาในการทดสอบ 1 ชั่วโมง Motor 2 ตัวควบคุมการไต่อยู่ที่ 100% ทำงานได้ที่ 16 นาที

4.3.2.3 ผลการบันทึกค่าแรงดันไฟฟ้า

หลังจากที่ได้ทำการเก็บค่าพลังงานไฟฟ้า ผู้พัฒนาได้นำข้อมูลการทดสอบทั้ง 5 รูปแบบ ซึ่งใช้สำหรับการประเมินผลการการทำงาน สรุปเป็นชุดข้อมูลรวมดังนี้



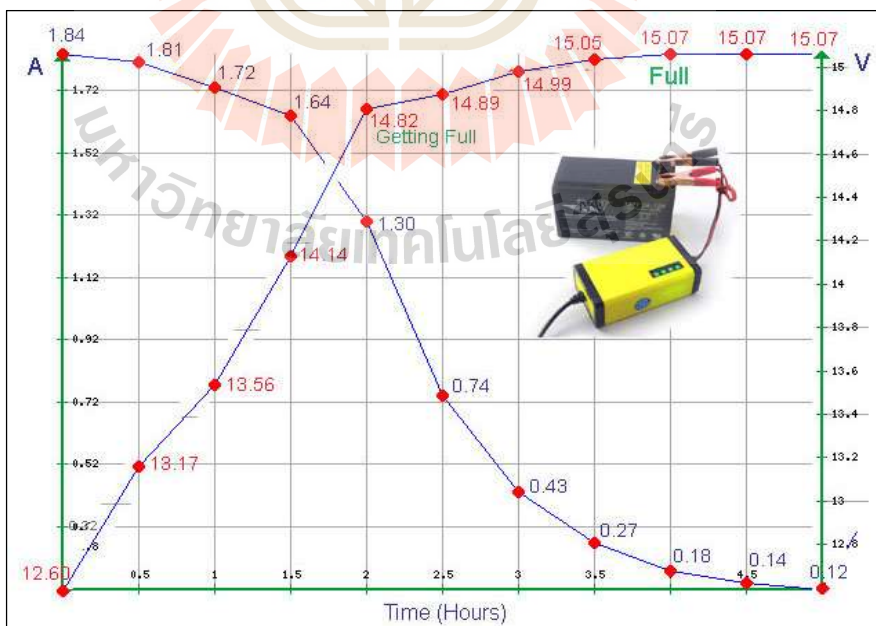
ภาพที่ 4.52 กราฟที่ระยะเวลาการใช้งาน



ภาพที่ 4.53 กราฟระยะเวลาการชาร์จแบตเตอรี่

4.3.2 เวลาในการชาร์จแบตเตอรี่

ในส่วนนี้เป็นส่วนของการเก็บข้อมูลระยะเวลาในการชาร์จแบตเตอรี่ ดังรูปที่ 4.53 เพื่อใช้สำหรับประเมินระยะเวลาในการชาร์จ ให้ได้ทันถ่วงทีต่อการใช้งานในแต่ละภารกิจ



ภาพที่ 4.54 กราฟ Charge Vole ,Time

4.4 ปัญหาที่พบ

1. หาโมลที่ใช้สร้าง Shell ค่อนข้างยากเพราะ Design ที่สร้างนั้นมีขนาดใหญ่
2. น้ำหนักของ Shell ไม่เท่ากันทำให้ วิ่งได้ไม่สมบรูณ์ ควบคุมทิศทางได้ยาก
3. ผิวขรุขระภายใน Shell ของ Robot ทำให้เกิดการติดขัดขณะวิ่ง
4. หัวฉีดสลดลื้อ และแม่เหล็กดูดติดหัวฉีดที่ใช้เย็บ Shell
5. น้ำหนักของหัวมีผลต่อการยึดติดระหว่างหัวกับตัวมาก
6. ถ้าระยะห่างระหว่างแม่เหล็กไม่เท่ากันจะมีผลต่อการเคลื่อนที่ คือ
 - ถ้าแม่เหล็กแนบโมลมากเกินไป Robot จะยกตัวลอยขึ้น จะส่งผลหุ่นยนต์วิ่งไม่ได้
 - ถ้าจังหวะที่ล้อหุ่นยนต์วิ่งไปเจอขอบยางที่เย็บ Shell นั้นจะเกิดการยุบตัวส่งผลให้แม่เหล็กจะห่างจากกันมากเกินไป ส่งผลให้ทำให้หัวอาจหลุดได้
7. หากผิวของ Shell นั้นถ้ามีหัวฉีดเล็ก ๆ หรือหินฝังอยู่ ระหว่างทำภารกิจ หัวอาจหลุดได้

4.5 แนวทางการแก้ไข

- ข้อที่ 1-5 มีการดำเนินการปรับปรุงแก้ไขเรียบร้อยแล้ว
- ข้อที่ 6 ปัจจุบันแก้ไขโดยการเจาะรูที่ โมล เพิ่มจุดยึดหัวฉีดเพื่อไม่ให้มีช่องว่างในการยุบตัว รวมไปถึงมีความพยายามขัดผิวด้านในคาน้ำหนักของ Robot ตัวในลงอีก
- ข้อที่ 7 ปัจจุบันแก้ไขปัญหาโดยการลง WAX ที่ผิวของ Shell ด้านนอกให้มันวาวเพื่อเวลาที่กลิ้งไปสะดุดของมีคมหรือฉีดจะได้หลุดออกง่าย หรือไม่ติดโมล เพราะถ้ามีแผลขรุขระแล้วจะทำให้ส่วนหัวเคลื่อนที่ได้ค่อนข้างยาก

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปงานวิจัย

งานวิจัยนี้เป็นการออกแบบหุ่นยนต์ทรงกลม สามารถเคลื่อนที่ได้ (Mobile Robot) ตามคำสั่งที่ป้อนให้กับหุ่นยนต์ด้วยรีโมทคอนโทรล และสามารถทำงานได้ตามที่ผู้วิจัยต้องการ ในส่วนของการดำเนินการตรวจวัดและเก็บข้อมูลนำมาวิเคราะห์เพื่อหาความสามารถ ทำให้ได้ทราบถึงสมรรถนะและประสิทธิภาพการใช้งานหุ่นยนต์ที่สร้างขึ้น ทำให้รู้ว่าสิ่งใดควรปรับปรุง สิ่งใดควรใส่ใจเป็นพิเศษ ตลอดจนการวัดค่าพลังงานที่ใช้ในแต่ละภารกิจเพื่อทราบเวลาที่ชัดเจน ในการใช้งานเพื่อนำไปใช้งานที่มีประสิทธิภาพ รวมไปถึงการเลือกใช้วัสดุอุปกรณ์ที่เหมาะสม อีกทั้งยังใช้ความรู้ที่เรียนมาได้ตรงตามวัตถุประสงค์ที่ตั้งไว้ได้อย่างชัดเจน

เมื่อทำการวิเคราะห์ข้อมูลและเปรียบเทียบผลการทดสอบหุ่นยนต์รุ่นเก่า และได้พัฒนาปรับปรุงมาเป็นรุ่นปัจจุบัน สรุปได้ว่าหุ่นยนต์ที่สร้างนั้นสามารถที่ใช้ในการปฏิบัติการกิจ (Operation time) โดยไม่พักจะอยู่ที่ประมาณ 25 นาที ส่วนถ้าบังคับโดยใช้ความเร็วเป็นครั้งหนึ่งของความเร็วสูงสุดตลอดภารกิจ ความสามารถสามารถปฏิบัติการกิจระยะเวลาจะอยู่ที่ประมาณ 50 นาที ซึ่งทั้งนี้ขึ้นอยู่กับการบังคับหุ่นยนต์ของผู้ใช้งานและภารกิจที่แตกต่างกันออกไป สำหรับระยะเวลาในการชาร์ตแบตเตอรี่ในแต่ละครั้ง จะใช้ระยะเวลาในการชาร์ต 2 ชั่วโมง ใช้กระแสสำหรับการชาร์ตที่ดี สมควรอยู่ที่ 0.8 mA 15 V และมีแบตเตอรี่สำรอง ชนิด Lipo ขนาด 14.8 V สำรองภายในตัวหุ่นยนต์ ซึ่งสามารถเรียกใช้ได้ในกรณีฉุกเฉิน สำหรับชุดแบตเตอรี่สำรองระยะเวลาการปฏิบัติการกิจโดยไม่พักจะอยู่ที่ประมาณ 10 นาที ในส่วนของ ความสมบูรณ์ในการควบคุมทิศทางอิสระอยู่ในเกณฑ์ดี และส่วนหัวและเปลือกหุ้มมีโอกาสในการหลุดออกจากกัน ลดลงในเกณฑ์ที่ยอมรับได้ ซึ่งเหตุการณ์ที่ทำให้ทั้ง 2 ส่วนนี้หลุดออกจากกันนั้น เป็นกรณีที่มีการเปลี่ยนแปลงทิศทางสวนกันอย่างฉับพลัน และหุ่นยนต์พุ่งชนสิ่งกีดขวางอย่างรุนแรง ทั้งนี้ขึ้นอยู่กับปัจจัยอื่นด้วย ในส่วนของระยะทางการควบคุมหุ่นยนต์ ที่ได้ทำการทดสอบ ระยะควบคุมสูงสุดอยู่ที่ 700 – 900 เมตร ขึ้นอยู่กับสภาพแวดล้อมและสิ่งกีดขวาง

จากการที่ได้ทำการออกแบบและทดลองสร้างหุ่นยนต์ลักษณะนี้ทั้ง 4 รุ่น ผู้วิจัยพบว่า ยังมีได้อีกหลากหลายวิธี ที่จะสามารถออกแบบและสร้างหุ่นยนต์ลักษณะนี้ได้ อีก ซึ่งขึ้นอยู่กับ การเลือกใช้อุปกรณ์ที่มีประสิทธิภาพ การติดตั้งชุดขับเคลื่อนในตำแหน่งใหม่ๆ รวมถึงการเพิ่ม ลูกเล่นให้หุ่นยนต์นี้มีความน่าสนใจมากยิ่งขึ้น ตลอดจนชุดอุปกรณ์เสริมที่สามารถอ่านค่า และป้อนกลับมายังผู้ควบคุม (Monitor) ก็มีผลสำคัญที่จะช่วยประเมินการควบคุม ได้ดียิ่งขึ้น ซึ่งจะช่วยให้เพิ่มประสิทธิภาพในการทำงานของหุ่นยนต์นี้ รวมไปถึงสามารถประยุกต์ใช้กับงานอื่น ๆ ที่เกี่ยวข้องได้ต่อไป

5.2 ข้อเสนอแนะ

จากวิจัยในครั้งนี้ ทำให้ได้ทราบถึงวิธีการเลือกหาอุปกรณ์ที่เหมาะสมที่สุดเพื่อใช้สร้าง หุ่นยนต์ทรงกลม ว่าผลลัพธ์ที่ได้นั้นจะมีรูปแบบเป็นไปตามที่ผู้วิจัยได้ออกแบบไว้ก่อนสร้างเพียงใด และมีข้อเสนอแนะเพื่อนำไปศึกษาต่อไปดังนี้

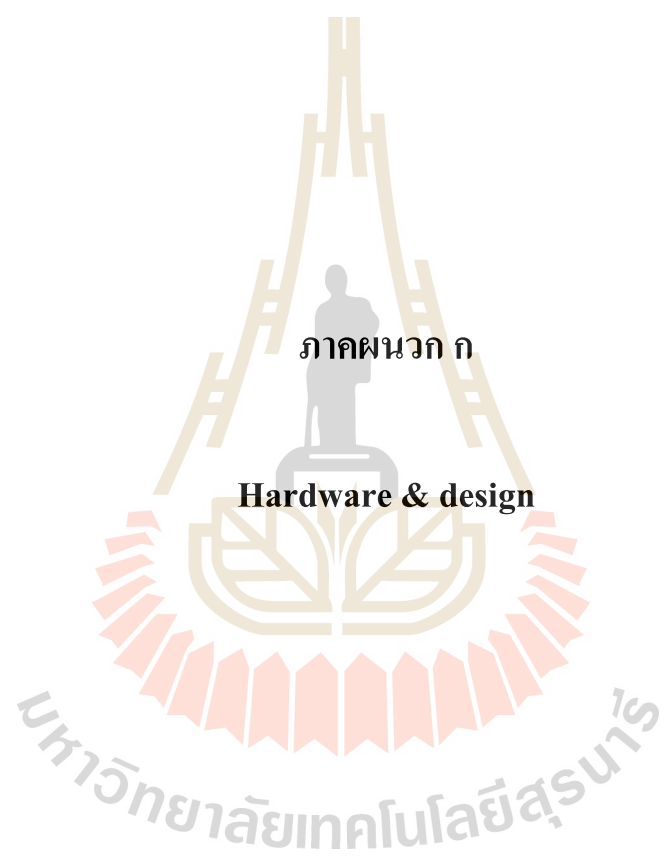
- ในอนาคตต้องสร้างระบบควบคุมอัตโนมัติสามารถทำงานได้เองโดยไม่ต้องมีผู้ควบคุม
- ในอนาคตต้องสร้างระบบตอบโต้กับมนุษย์ให้สามารถสื่อสารและแสดงกิริยาท่าทางได้

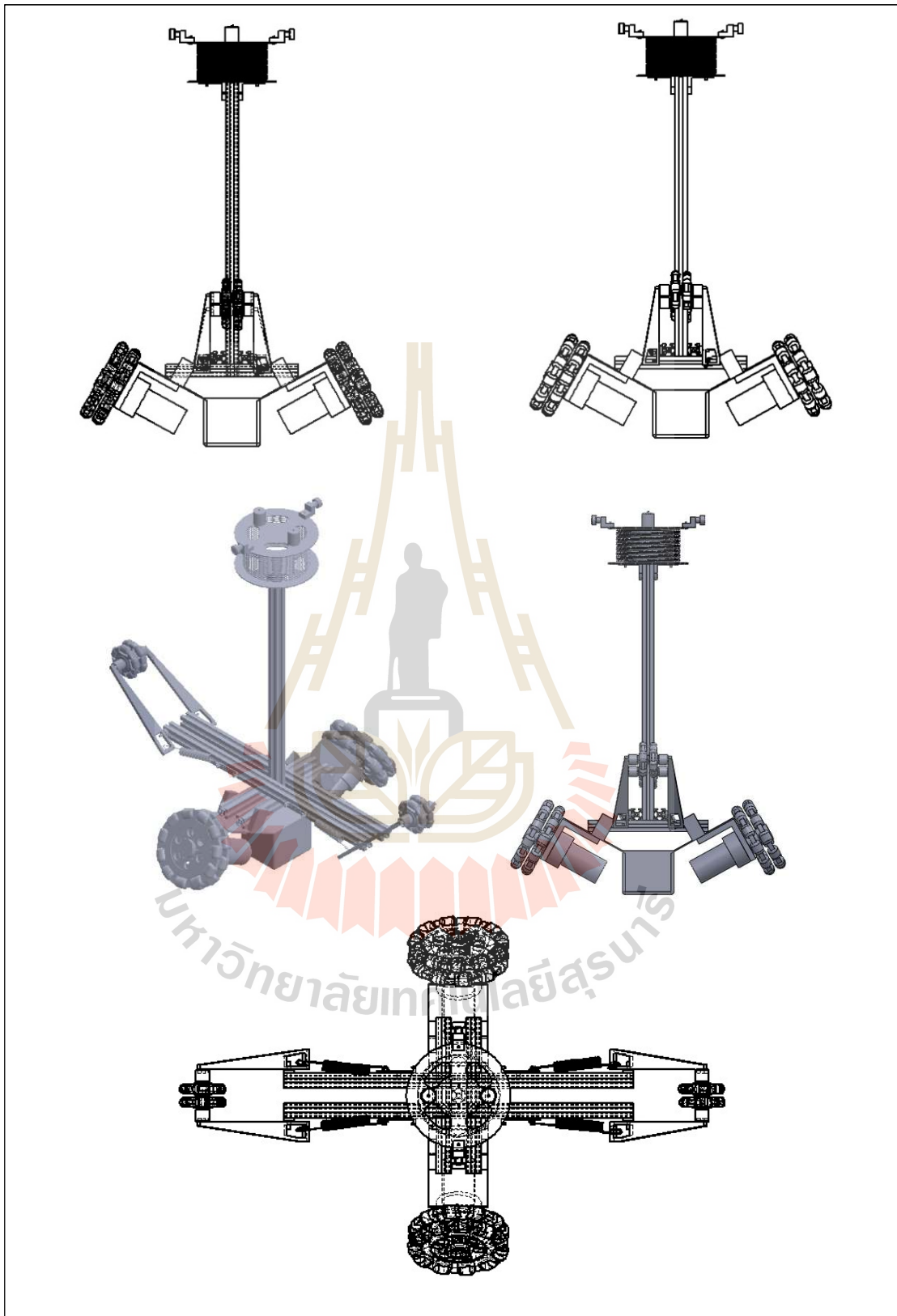
การศึกษาเกี่ยวกับการสร้างหุ่นยนต์ทรงกลมต้นแบบนี้ ผู้วิจัยหวังเป็นอย่างยิ่งว่าจะเป็น ประโยชน์ต่อการพัฒนาวงการหุ่นยนต์ และอุตสาหกรรมในประเทศไทยตลอดจน นักเรียนนิสิต นักศึกษาและผู้ที่มีความสนใจ เนื่องจากจะเป็นอีกแนวทางหนึ่งที่สำคัญ ซึ่งจะมีส่วนในการช่วย ในการพัฒนาไอเดียร์การออกแบบ และการเลือกใช้อุปกรณ์ที่เหมาะสม ในการสร้างชิ้นงาน สร้างหุ่นยนต์ หรือสิ่งประดิษฐ์อื่นๆ และเป็นประโยชน์ต่อการพัฒนาต่อยอดที่ดีได้ อีกต่อไป

รายการอ้างอิง

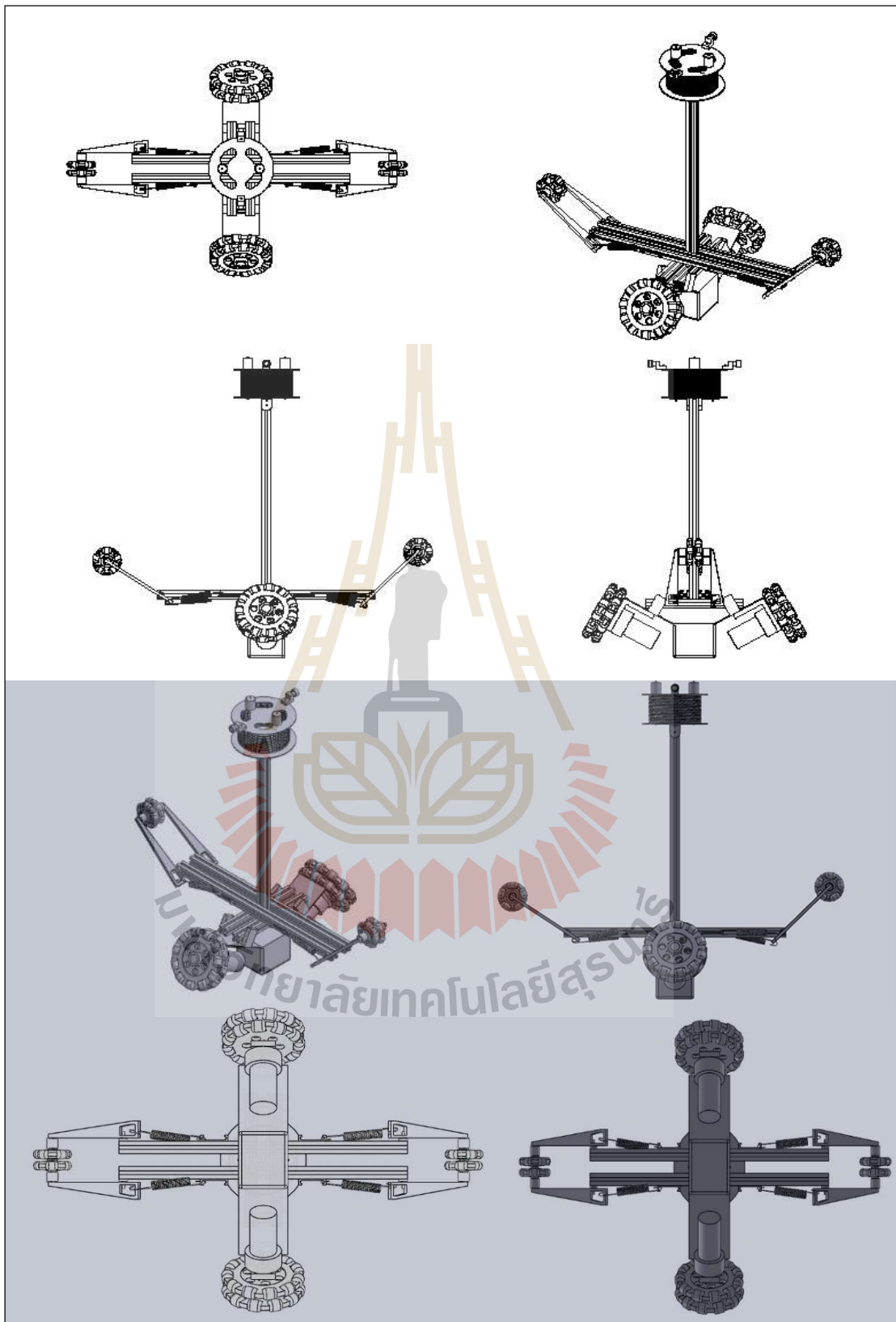
- Halme. A., Schonberg. T., and Wang. Y.(1996). Motion Control of a Spherical Mobile Robot. In **Proceedings of the 4th International Workshop on Advanced Motion Control Tsu-City Japan**. 1: 259–264.
- Lin, X., Guo, S., and Tanaka Hata S.(2011). Development of a Spherical Underwater Robot. In **Proceedings of 2011 IEEE/ICME International Conference on Complex Medical Engineering (CME) Harbin China**. 4: 662–665.
- Michaud, F., Laplante, J., Larouche, H., Duquette, S., Létourneau, and Masson, D.(2005). Autonomous spherical mobile robot for Child Development studies. **IEEE Trans. Syst. Man Cybern.** 3: 471–480.
- Kenyon, S., Creary, D., and Maynard, J.(2005). small cheap and portable reconnaissance robot. In **Proc SPIE London**. 5: 434–443.
- Seeman, M., Broxvall, M., Saffiotti, A., and Wide, P.(2006). Autonomous Spherical Robot for Security Tasks. In **Proceedings of 2006 IEEE International Conference on Computational Intelligence for Homeland Security Personal Safety**. 12(2): 51–55.
- Hanxu, S., Qingxuan, J., Chenkun, S., and Kailiang, Z.(2008.) An Omni-Directional Rolling Spherical Robot with Telescopic Manipulator. In **Proceedings of the 2nd International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA 2008) Shenzhen, China**. 1: 1–6.
- Liu, D., Sun, H., and Jia, Q.(2008). Spherical Mobile Robot Driving Ahead Motion Control by Feedback Linearization. In **Proceedings of the 2nd International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA 2008) China**. 143(3): 1–6.
- Alves, J., and Dias, J.(2003) Design and control of a spherical robot. In **Proc IME J. Syst. Contr. Eng.** 1: 457–467.

- Zhan, Q., Cai, Y., and Yan, C.(2011). Analysis and Experiments of an Omni-Directional Spherical Robot. In **Proceedings of 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai**. 1: 4921–4926.
- Yue, M., Deng, Z., and Yu, W.(2006). Introducing HIT Spherical Robot: Dynamic Modeling and Analysis Based on Decoupled Subsystem. In **Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO'06) Kunming**. 19: 181–186.
- Zhao, B., Wang, P., Li, M., and Sun, L.(2009). Study on Turning in Place of a Spherical Robot Based on Stick-Slip Principle. In **Proceedings of 2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)**. 3: 771–775.
- Kim, Y., Ahn, S., and Lee, Y.(2010). New A Spherical Robot with Arms. In **Proceedings of the 110th WSEAS International Conference on Robotics Control and Manufacturing Technology**. 1: 63–67.
- Artusi, M., Potz, M., Aristizabal, J., Menon, C., Cocuzza, S., and Debei, S.(2011). Electroactive elastomeric actuators for the implementation of a deformable spherical rover. **IEEE/ASME Trans. Mechatron**. 2: 50–57.
- Yamanaka, T., Nakaura, S., and Sampei, M.(2013). Hopping Motion Analysis of Superball Spherical Robot Based on Feedback Control. In **Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)**, Las Vegas NV, USA, 4: 3805–3810.
- Brown, H., and Xu, Y.(1996). Single-Wheel, Gyroscopically Stabilized Robot. In **Proceedings of 1996 IEEE International Conference on Robotics and Automation**, Minneapolis MN, USA. 1: 3658–3663.
- Shu, G., Zhan, Q., and Cai, Y.(2009). Motion Control of Spherical Robot Based on Conservation of Angular Momentum. **Proceedings of International Conference on Mechatronics and Automation (ICMA 2009)**, 12(1): 599–604.
- Peck, M.(2005). Low-Power, High-Agility Space Robotics. In **Proceedings of 2005 AIAA Guidance Navigation and Control Conference and Exhibit**, 1: 1–12.
- Schroll, G.(2010). Angular Momentum Torque Enhancement for Spherical Vehicles Robot. In **WO Patent WO**, 1: 117-129.

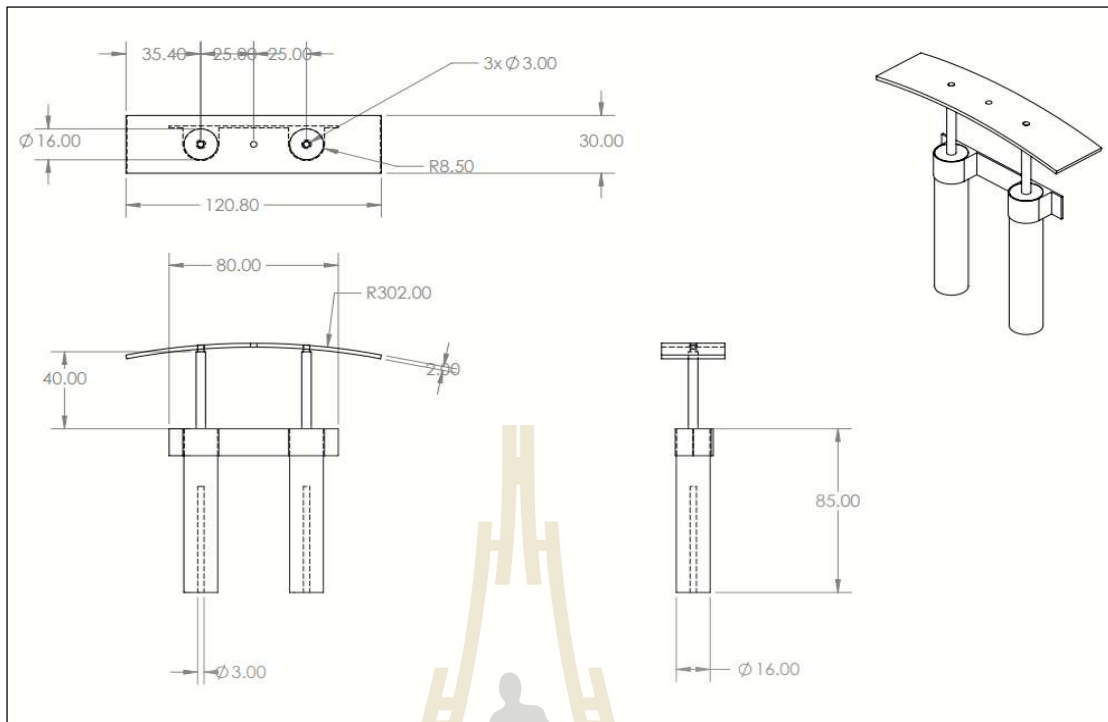




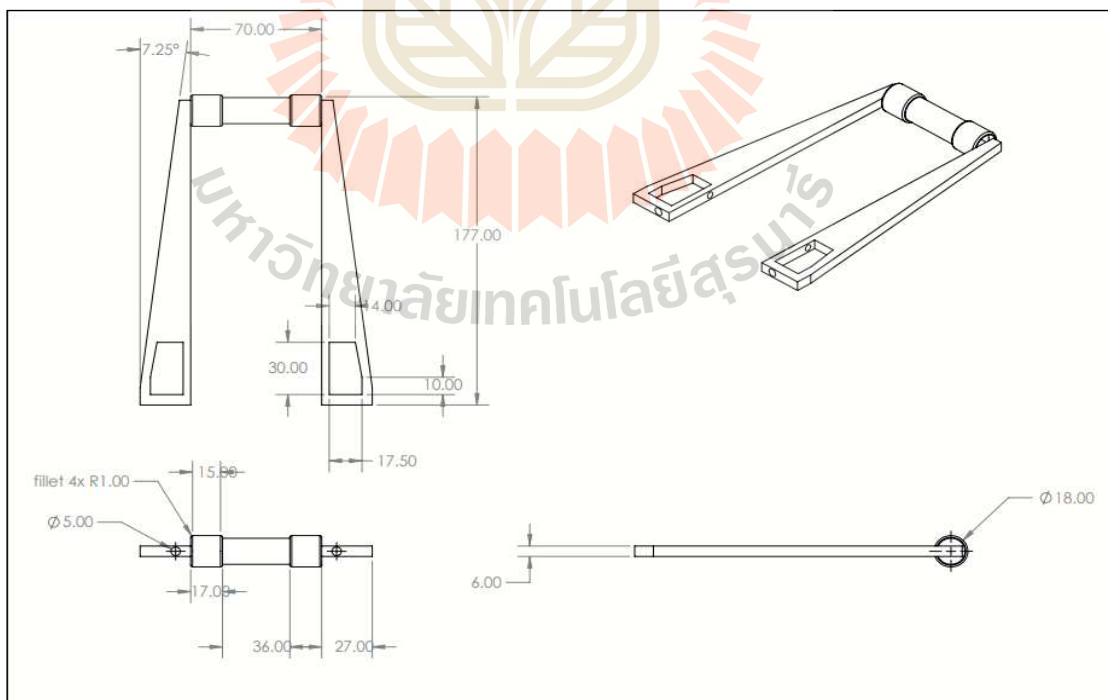
ก.1 Design Robot Concept Mask #IV (A)



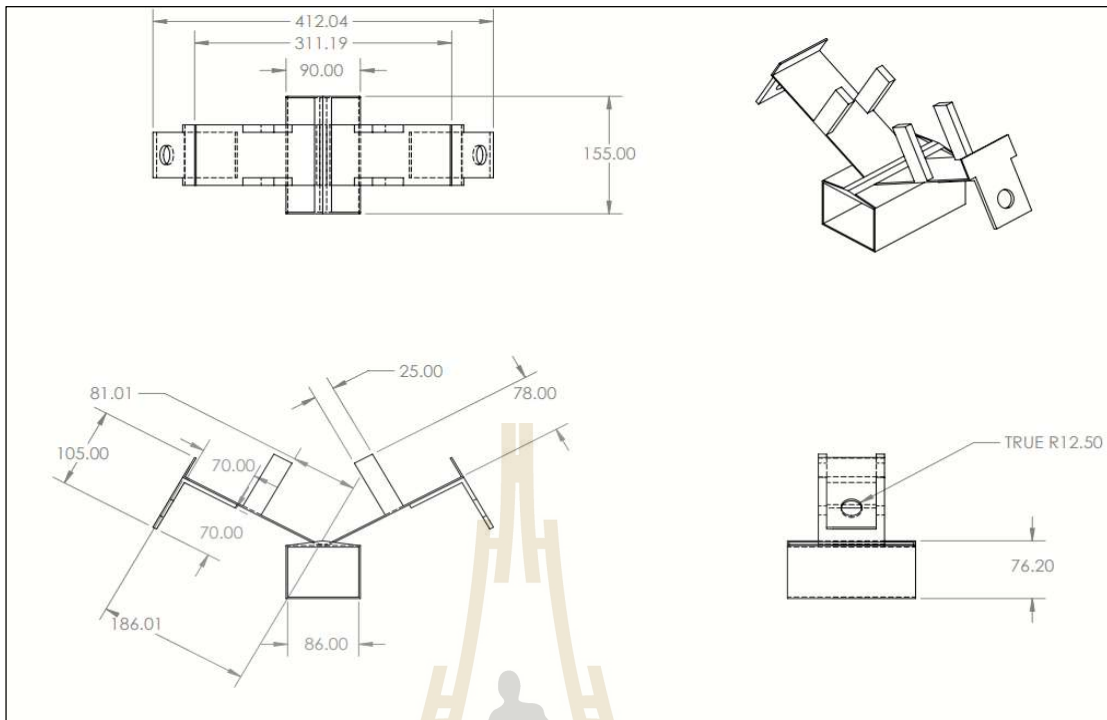
รูปที่ ๓.2 Design Robot Concept Mask #IV (B)



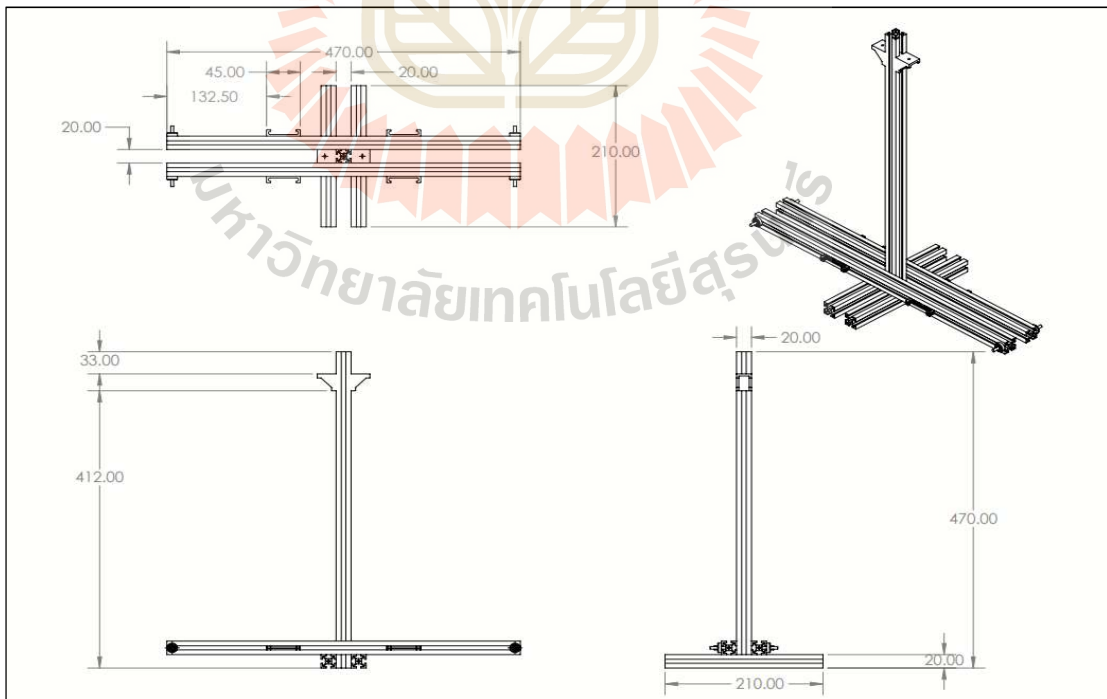
รูปที่ ๓.3 Dimension Head Damping



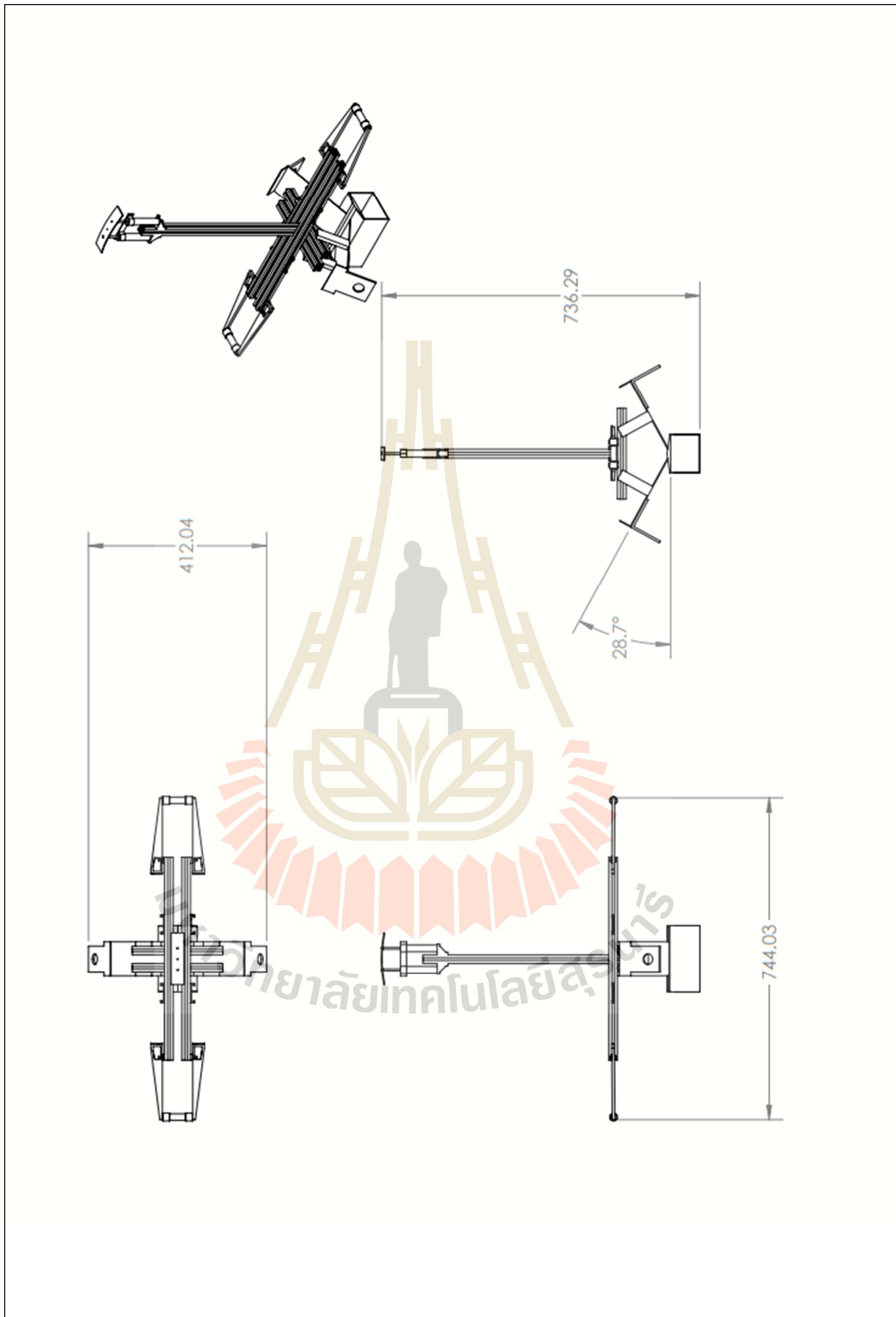
รูปที่ ๓.4 Dimension Swing Arm



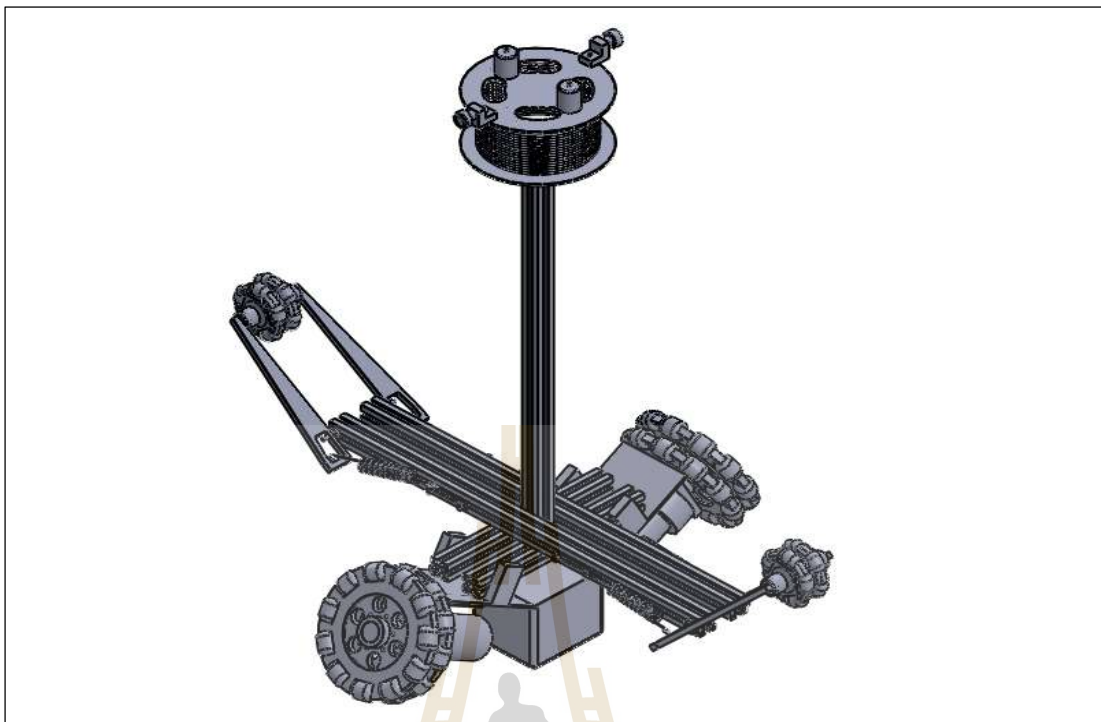
รูปที่ ๓.5 Dimension Base



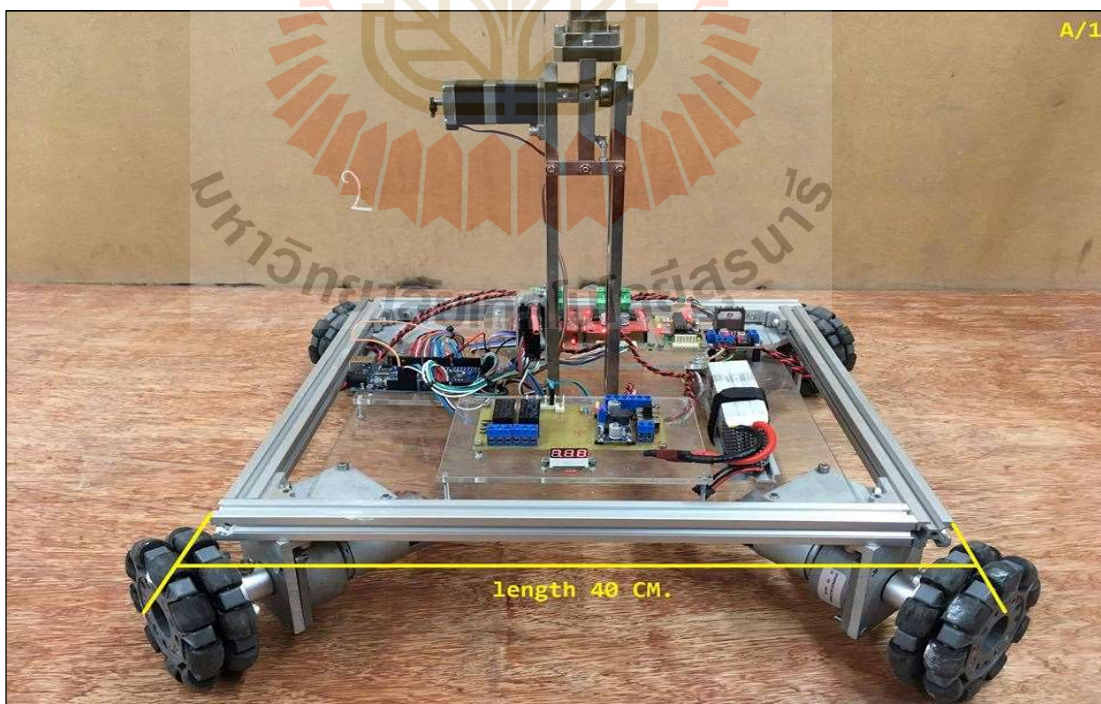
รูปที่ ๓.6 Dimension Structure



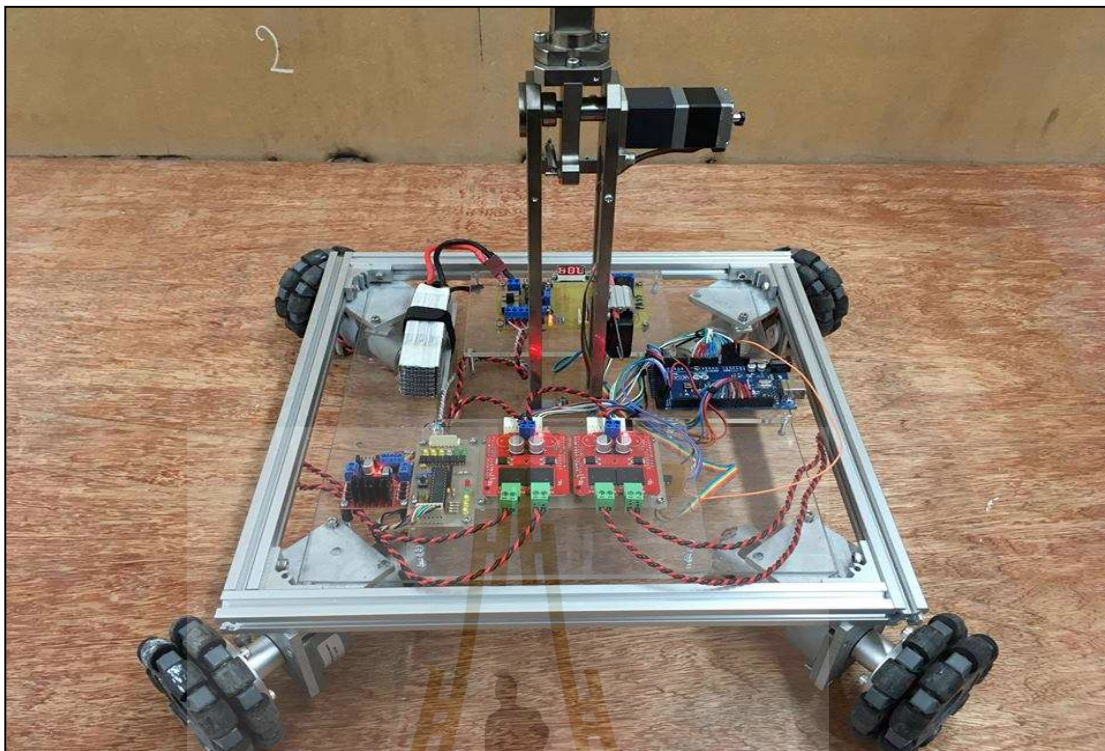
รูปที่ ๓.7 Structure Assembly



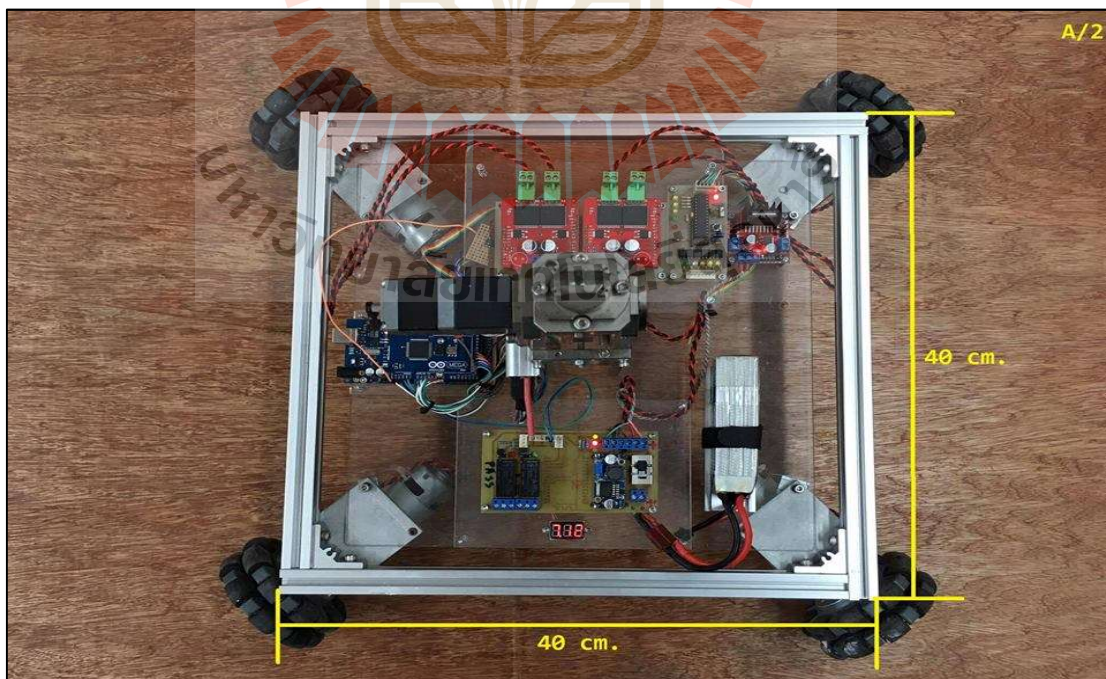
รูปที่ ก.8 Assembly Model Mask #IV



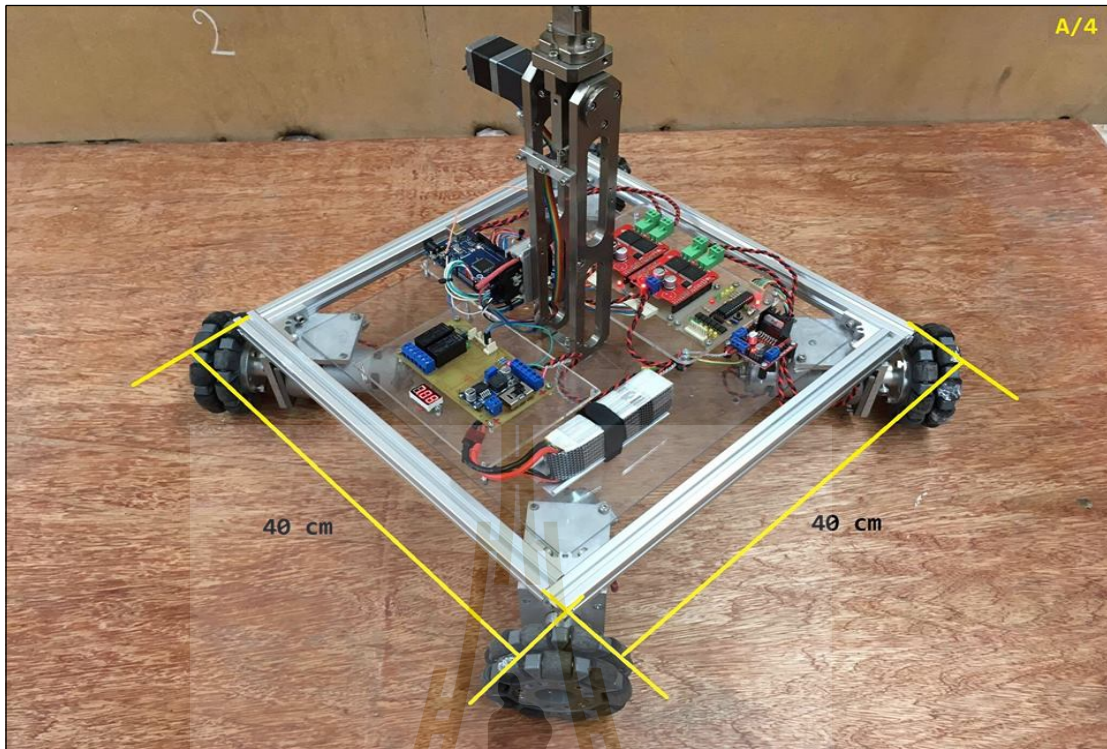
รูปที่ ก.9 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านหน้า



รูปที่ ก.10 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านหลัง



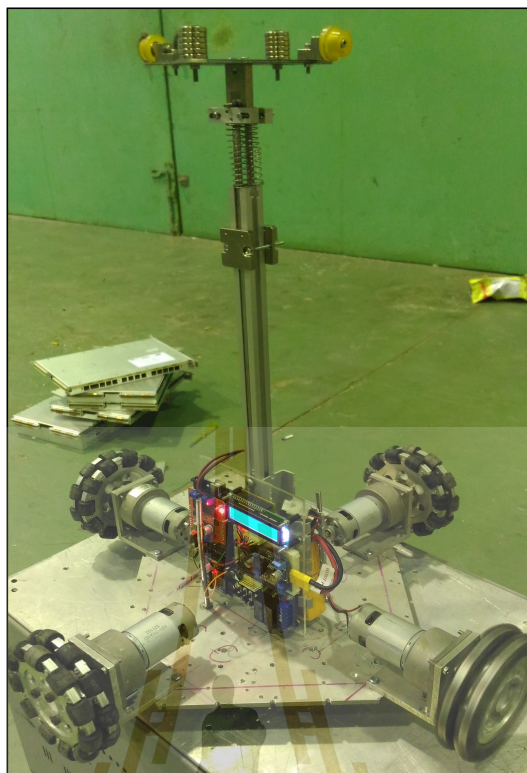
รูปที่ ก.11 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านบน



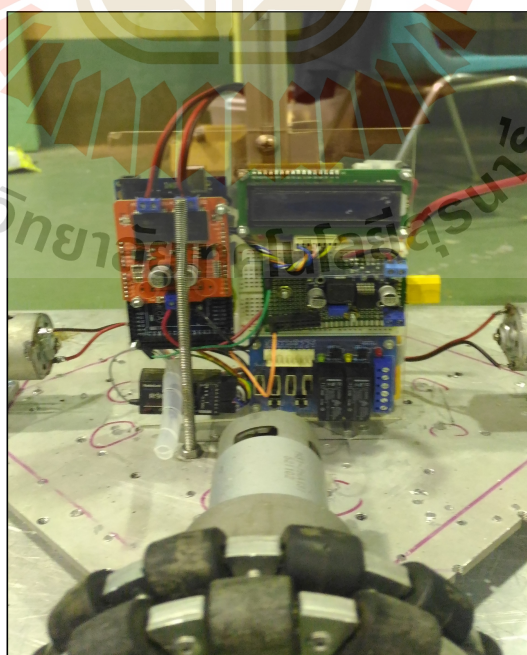
รูปที่ ก.12 หุ่นยนต์ภายใน Shell Mask #I มุมมองด้านข้าง



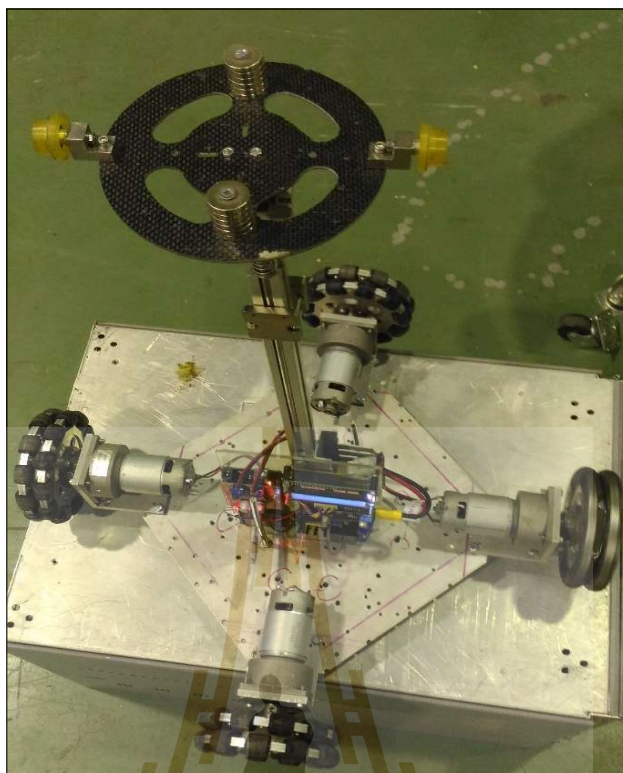
รูปที่ ก.13 หุ่นยนต์ภายใน Shell Mask #II มุมมองด้านหน้า



รูปที่ ก.14 หุ่นยนต์ภายใน Shell Mask #II มุมมองด้านข้าง



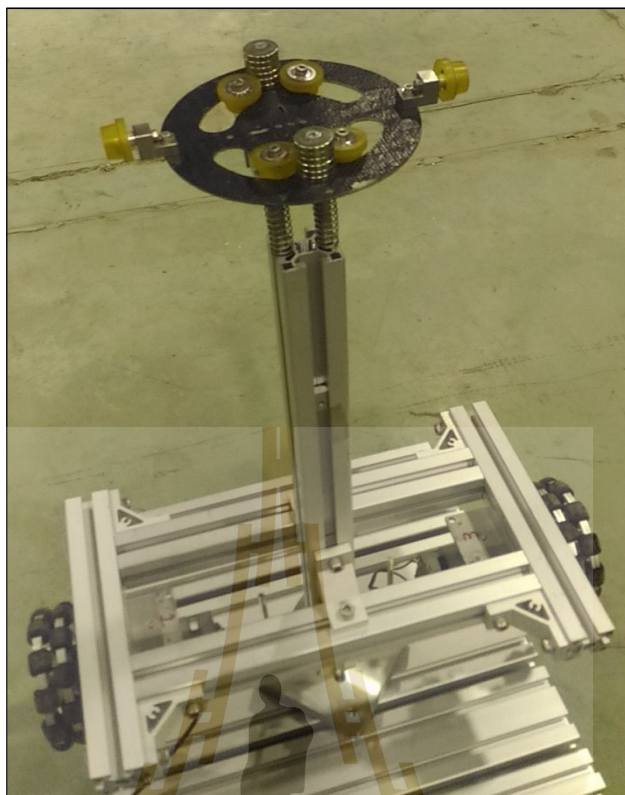
รูปที่ ก.15 วงจรรวมหุ่นยนต์ Shell Mask #II



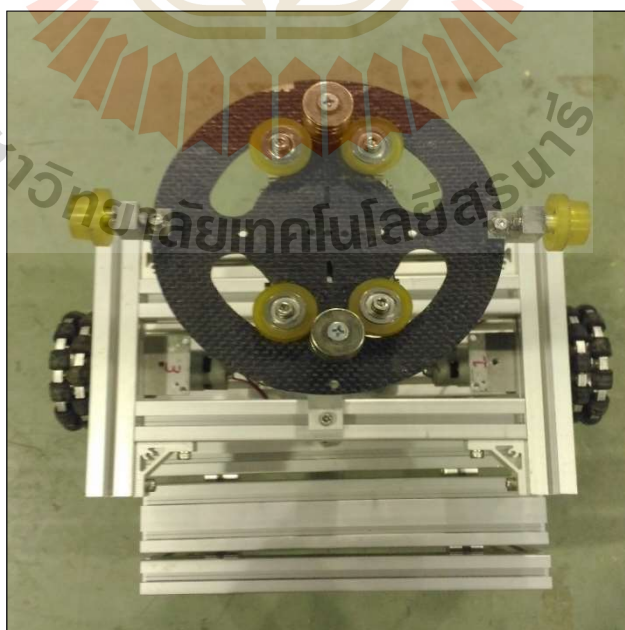
รูปที่ ก.16 ชุดแม่เหล็กดูดภายใน Shell Mask #II



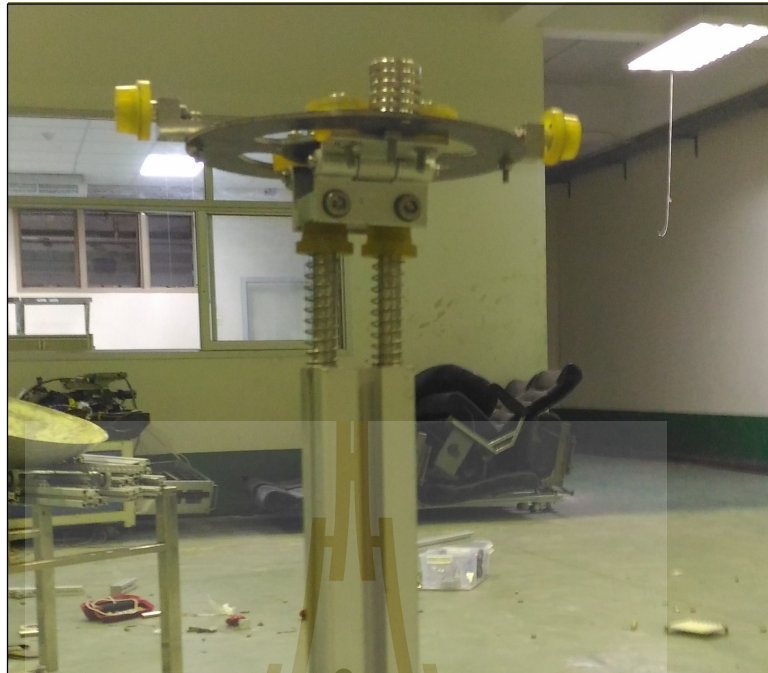
รูปที่ ก.17 โครงหุ่นยนต์ Mask #III ด้านหน้า



รูปที่ ก.18 โครงหุ่นยนต์ Mask #III ด้านข้าง



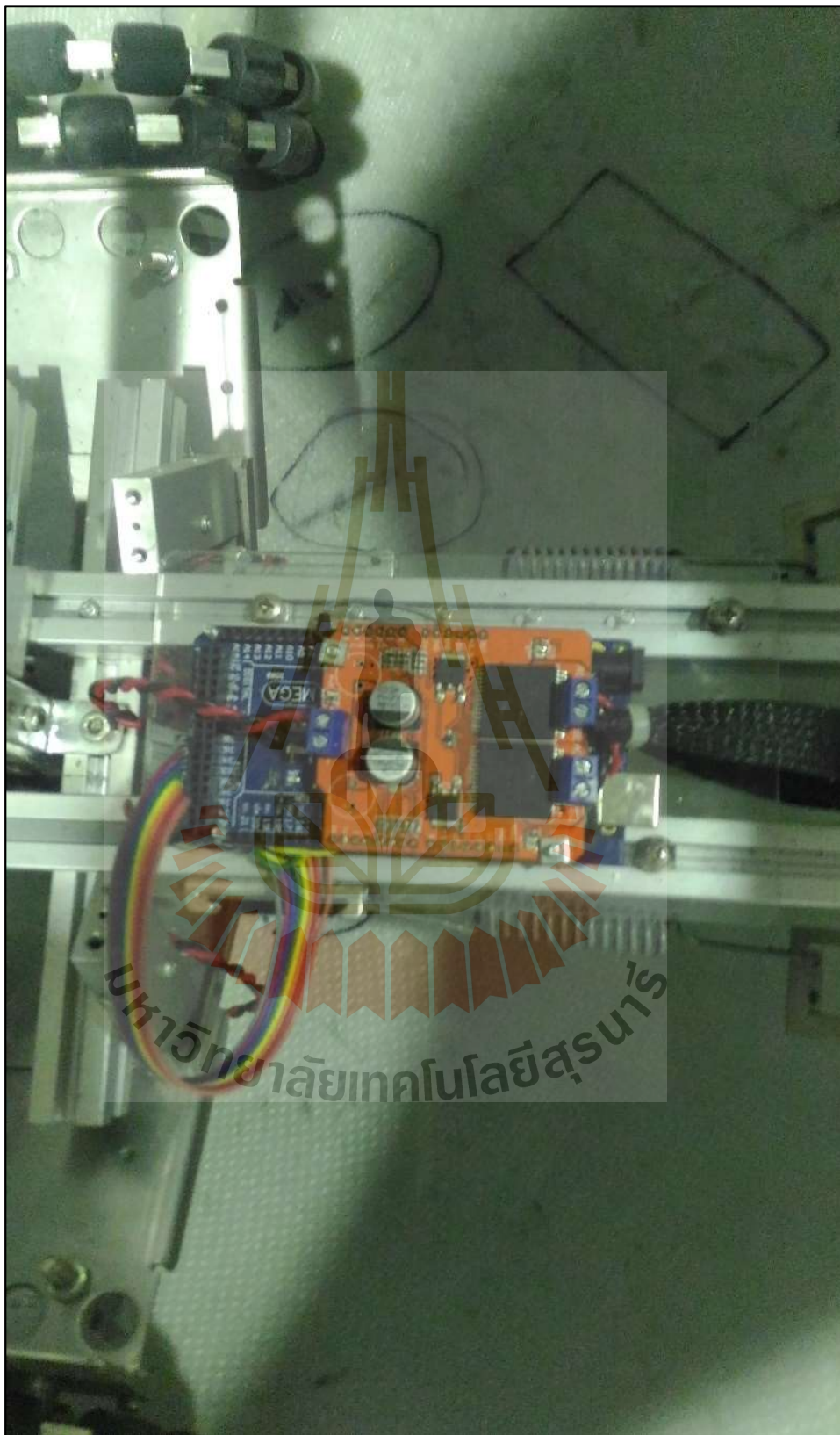
รูปที่ ก.19 ชุดแม่เหล็กดูดภายใน Shell Mask #III



รูปที่ ก.20 ชุด Damping Mask #III



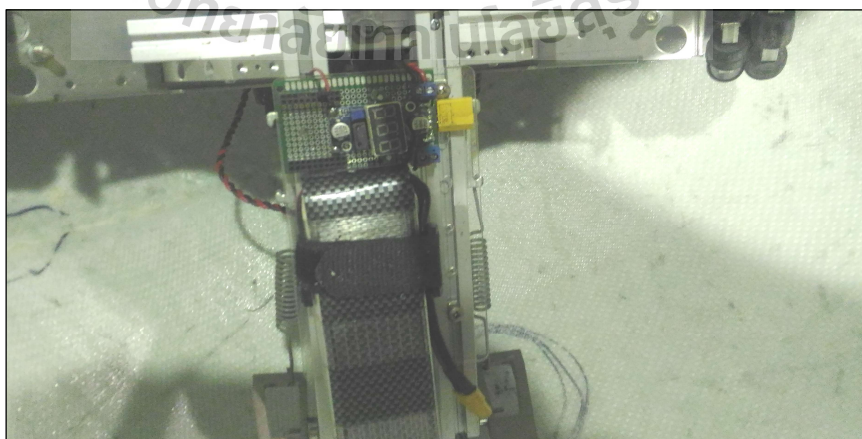
รูปที่ ก.21 ทดสอบความยืดหยุ่น Damping



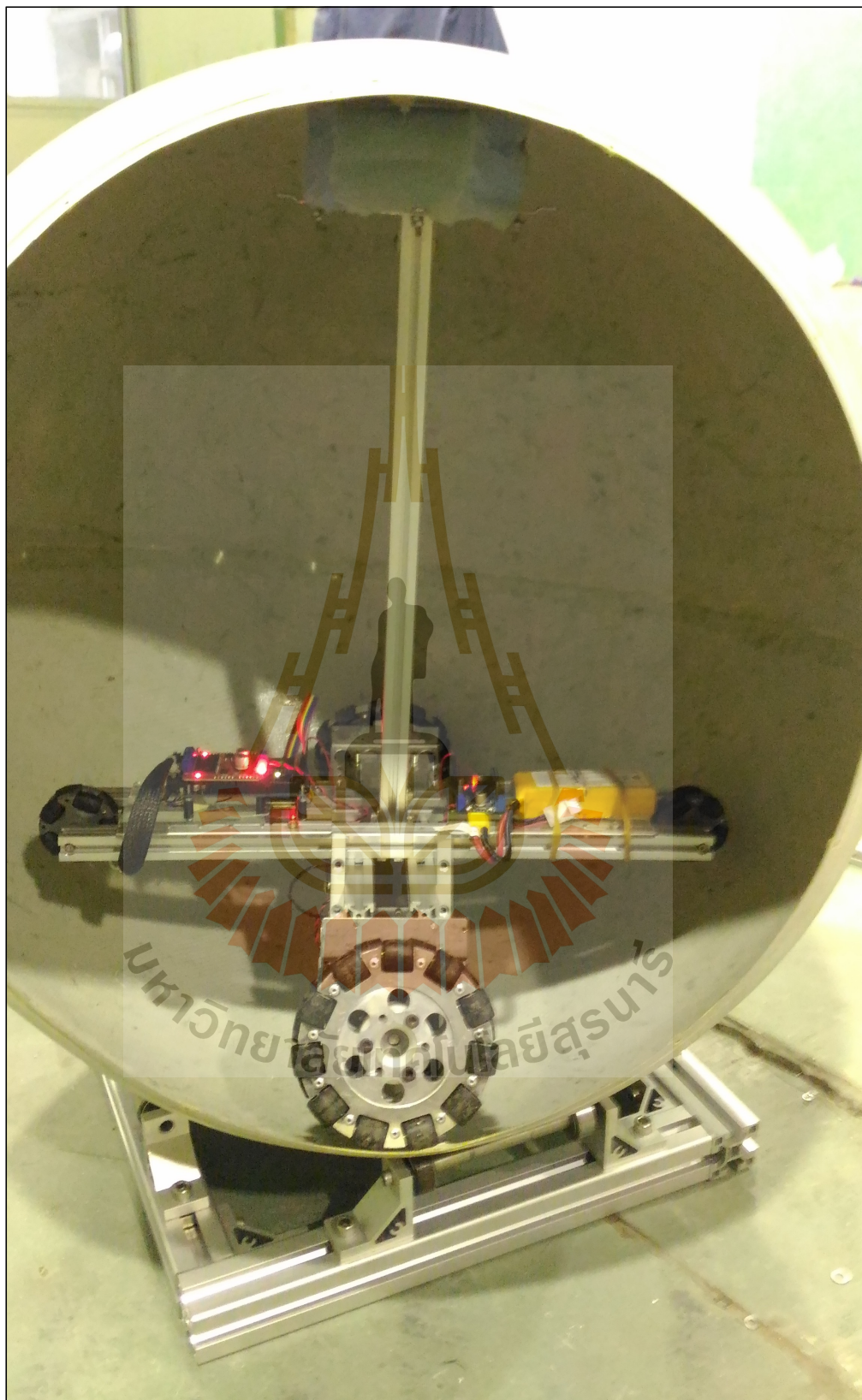
รูปที่ ก.22 วงจรรวม Mask # IV



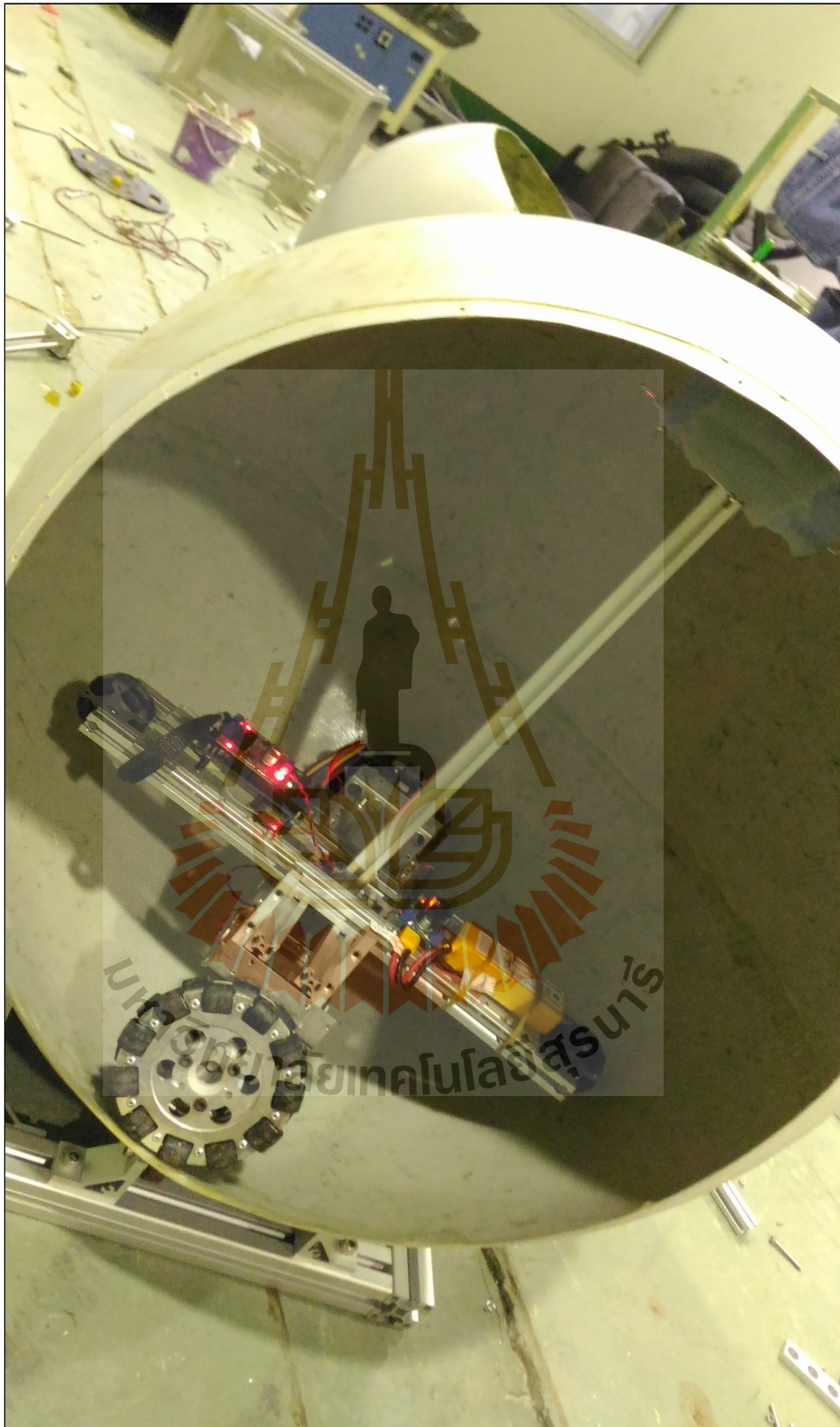
รูปที่ ก.23 หุ่นยนต์ภายใน Shell Mask #IV



รูปที่ ก.24 วงจรลดแรงดัน Plugin Mask #IV



รูปที่ ก.25 บรรจุ Robot เข้าตู้ Shell



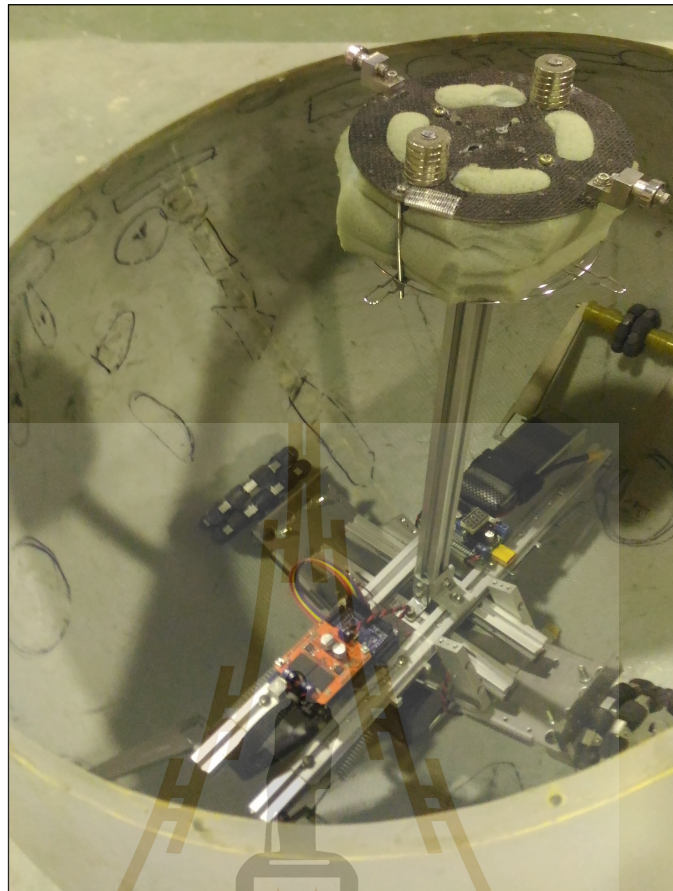
รูปที่ ก.26 ทดสอบระบบการทำงานก่อนปิด Shell



รูปที่ ก.27 Shell ที่ใช้ในการทำงานวิจัย



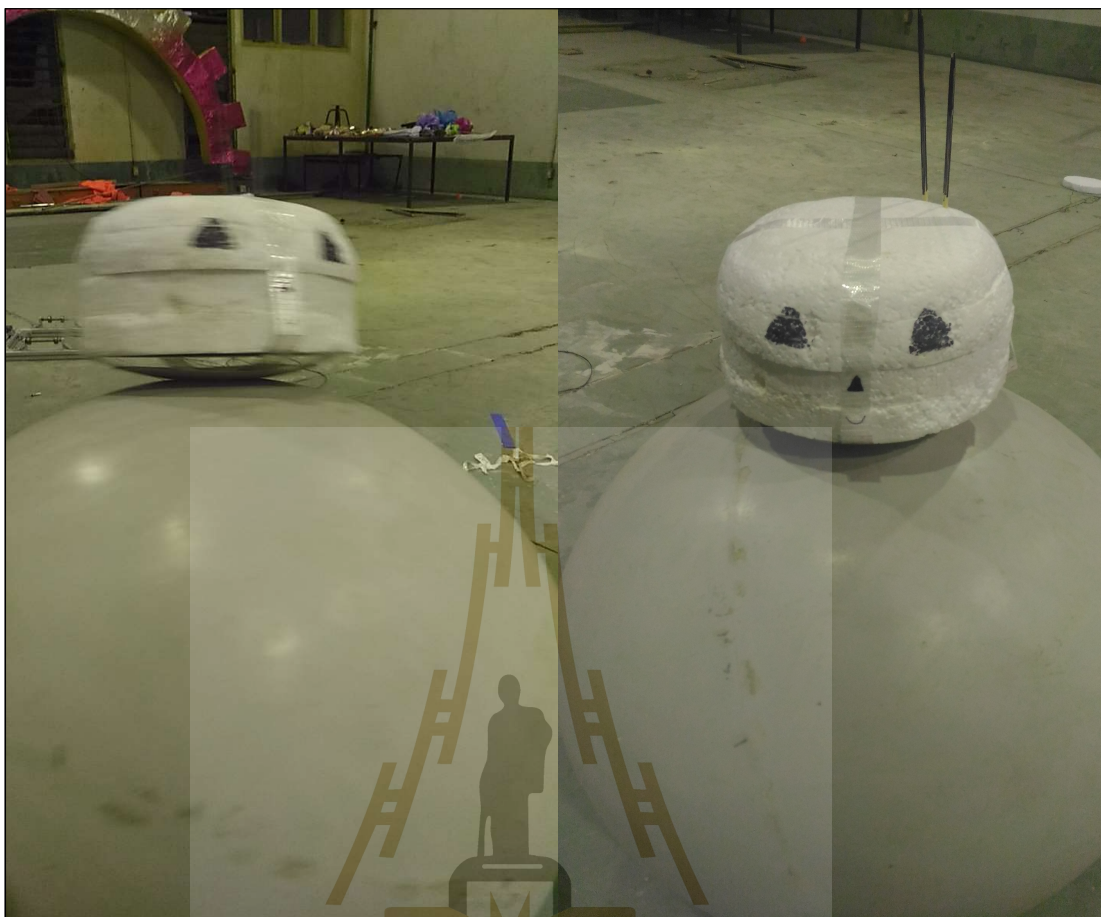
รูปที่ ก.28 บรรจุหุ่นยนต์และทดสอบการเคลื่อนที่



รูปที่ ก.29 มาร์คตำแหน่งที่ Shell ขรุขระ



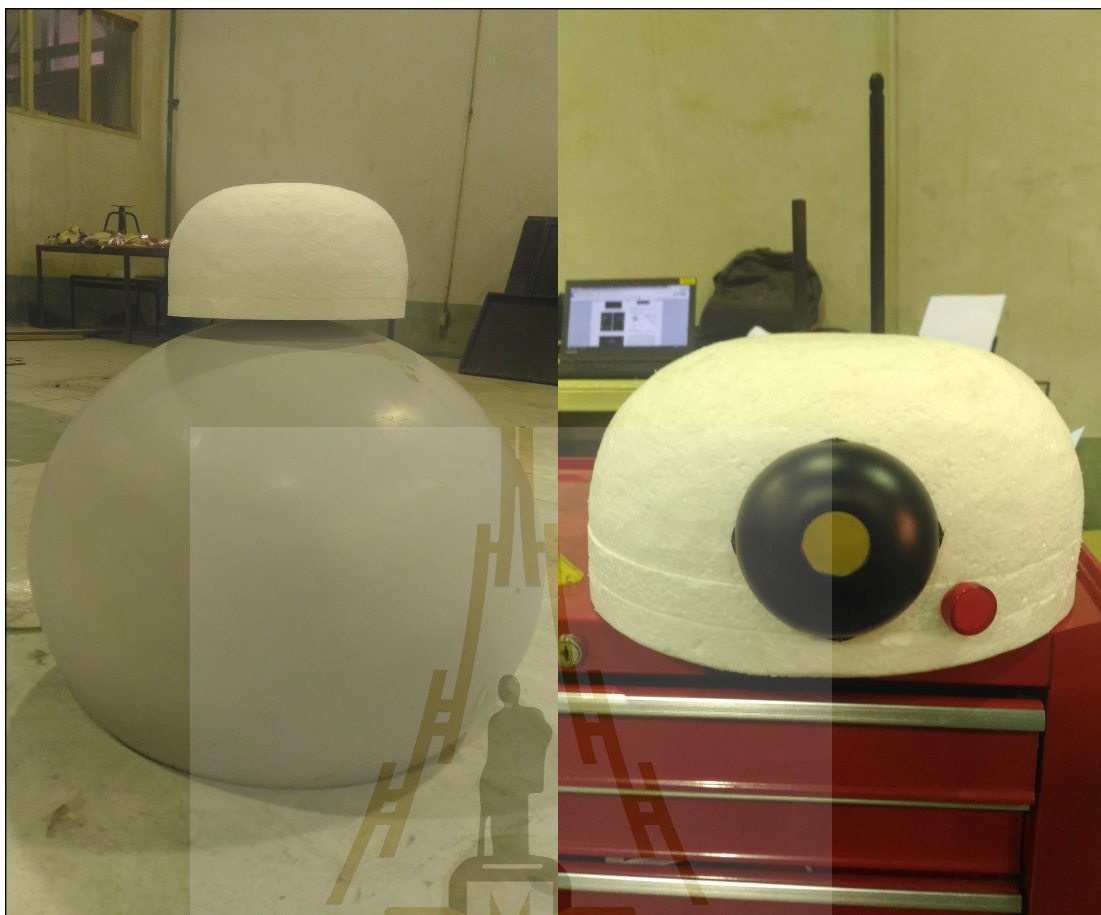
รูปที่ ก.30 แก้ไขตำแหน่งมาร์คที่มีปัญหา



รูปที่ ก.31 ทดสอบการติดตั้งหัวของหุ่นยนต์



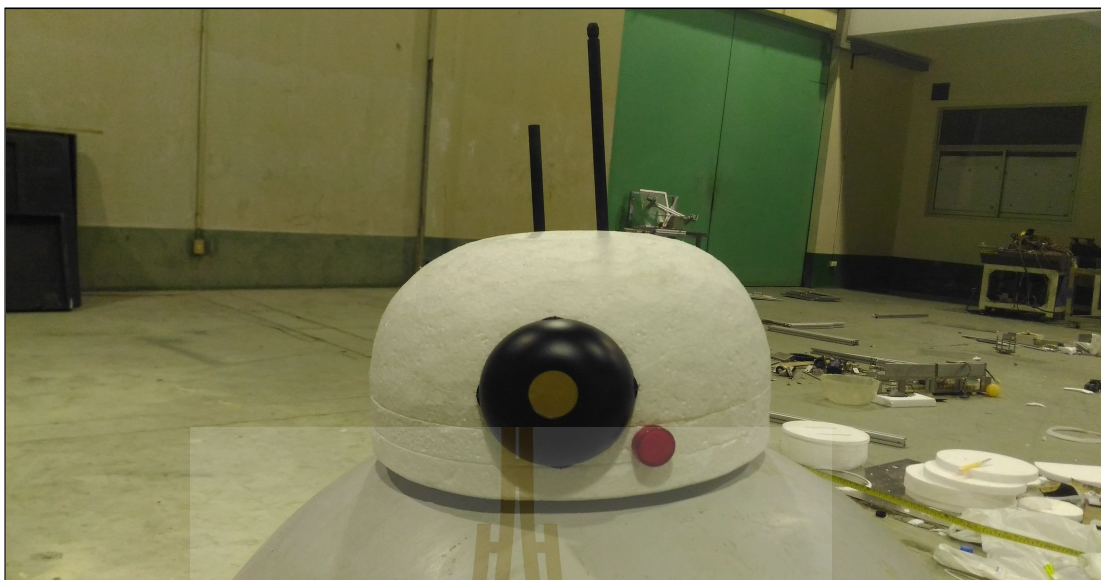
รูปที่ ก.32 ส่วนด้านล่างของหัวหุ่นยนต์ที่ติดแม่เหล็ก



รูปที่ ก.33 ออกแบบและสร้างส่วนหัวขึ้นใหม่



รูปที่ ก.34 ทำหัวให้กลวงเพื่อลดน้ำหนัก



รูปที่ ก.35 ตกแต่งส่วนหัวและประกอบหุ่นยนต์



รูปที่ ก.36 นำไปทดสอบและบันทึกผลการทดลอง



ภาคผนวก ข

รายละเอียดโปรแกรมควบคุม

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
byte CH1_PIN = 53;
byte CH2_PIN = 51;
byte CH3_PIN = 49;
byte CH4_PIN = 47;
byte CH5_PIN = 45;
byte CH6_PIN = 43;

int CH1;
int CH2;
int CH3;
int CH4;
int CH5;
int CH6;

int Relay1=19;
int Relay2=20;
int xDelay=0;

//=====
#define BRAKEVCC 0
#define CW 1
#define CCW 2
#define BRAKEGND 3
#define CS_THRESHOLD 2000
#define StatePin 13
```

รูปที่ ข.1 การประกาศตัวแปรในการควบคุม

```

int count=0;

int june=0; int PWM_Speed=0;

#define Div_Gap 40

#define Max_CH1 1910

#define Min_CH1 1072

#define Center_CH1 1490

#define Max_CH2 1905

#define Min_CH2 1080

#define Center_CH2 1480

#define Max_CH3 1910

#define Min_CH3 1085

#define Center_CH3 1495

#define Max_CH4 1890

#define Min_CH4 1071

#define Center_CH4 1460

```

รูปที่ ข.2 การประกาศตัวแปร Calibrate ค่าที่รับจาก Receiver

```

int Hcount = 0;
int Start = 23;
int Direc = 25;
int Speed = 27;
int inApin[4] = { 35 , 6 , 33 , 43 }; // INA: Clockwise input
int inBpin[4] = { 10 , 11 , 31 , 41 }; // INB: Counter-clockwise input
int pwmpin[4] = { 7 , 8 , 45 , 2 }; // PWM input
int cspin[4] = { A6 , A7 , A2 , A3 }; // CS: Current sense ANALOG input
int enpin[4] = { A1 , A5 , A0 , A4 }; // EN: Status of switches output

```

รูปที่ ข.3 การประกาศตัวแปรติดต่อกับ Monster shield

```
void setup() {  
  
  pinMode(CH1_PIN, INPUT);  
  
  pinMode(CH2_PIN, INPUT);  
  
  pinMode(CH3_PIN, INPUT);  
  
  pinMode(CH4_PIN, INPUT);  
  
  pinMode(CH5_PIN, INPUT);  
  
  pinMode(CH6_PIN, INPUT);  
  
  pinMode(Start, OUTPUT);  
  
  pinMode(Direc, OUTPUT);  
  
  pinMode(Speed, OUTPUT);  
  
  pinMode(Relay1, OUTPUT);  
  
  pinMode(Relay2, OUTPUT);  
  
  Serial.begin(115200);  
  
  Initial_MotorShield();  
  
}
```

รูปที่ ข.4 การกำหนดหน้าที่ตัวแปร

```
void loop() {  
  
  CH1 = pulseIn(CH1_PIN, HIGH);  
  
  CH2 = pulseIn(CH2_PIN, HIGH);  
  
  CH4 = pulseIn(CH3_PIN, HIGH);  
  
  CH3 = pulseIn(CH4_PIN, HIGH);  
  
  CH6 = pulseIn(CH5_PIN, HIGH);  
  
  CH5 = pulseIn(CH6_PIN, HIGH);  
  
  june = (CH5-1000)/7;  
  
  Serial.print("CH1 = ");  
  
  Serial.print(CH1);  
  
  Serial.print("\t CH2 = ");  
  
  Serial.print(CH2);  
  
  Serial.print("\t CH3 = ");  
  
  Serial.print(CH3);  
  
  Serial.print("\t CH4 = ");  
  
  Serial.print(CH4);  
  
  Serial.print("\t CH5 = ");  
  
  Serial.print(CH5);  
  
  Serial.print("\t CH6 = ");  
  
  Serial.print(CH6);  
  
  Serial.print("\t june = ");  
  
  Serial.print(june);  
  
  Serial.println();  
}
```

รูปที่ ข.5 รับค่าและแสดงค่าผ่าน Serial Monitor


```

    if(CH6 < 1000 ) {
count++;
    if(count > 10) {
        Serial.println(count);
        digitalWrite(13, HIGH);
        delay(100);
        digitalWrite(13, LOW);
        delay(100);
        digitalWrite(Relay1, LOW);
    }

} else {
    digitalWrite(Relay1, HIGH);
    Serial.println("Remote on");

    Serial.println(xDelay);
    Check_RemoteCH3Sprint();
    xDelay=150;
    Serial.println(xDelay);
    Arun();
}

Check_head();
}

void Arun() {
    Check_RemoteCH3Sprint();
    Check_RemoteCH1ReftLife();
    Check_RemoteCH2GoBack();
    Check_head();
    Stop();
    count=0;
}

```

รูปที่ ข.6 ฟังก์ชันเช็คควารีโมทได้เปิดอยู่หรือไม่

```

void Initial_MotorShield(void)

{ pinMode(StatePin, OUTPUT); // Fault Monitor

  for (int i = 0; i < 3; i++)

    { pinMode(inApin[i], OUTPUT); // Initialize digital pins as outputs

      pinMode(inBpin[i], OUTPUT);

      pinMode(pwmpin[i], OUTPUT);

      digitalWrite(inApin[i], LOW); // Initialize braked

      digitalWrite(inBpin[i], LOW);
    }
}

void motorOff(int motor)
{ motor--;
  digitalWrite(inApin[motor], LOW);
  digitalWrite(inBpin[motor], LOW);
  analogWrite(pwmpin[motor], 0);
}

void motorGo(uint8_t motor, uint8_t direct, uint8_t pwm)
{
  if (motor == 1) {
    if (direct == CW)
      {
        digitalWrite(inApin[0], HIGH);
        digitalWrite(inBpin[0], LOW);
      }
    else if (direct == CCW)
      { digitalWrite(inApin[0], LOW);
        digitalWrite(inBpin[0], HIGH);
      }
    else
      { digitalWrite(inApin[0], LOW);
        digitalWrite(inBpin[0], LOW);
      }
    analogWrite(pwmpin[0], pwm);
    // delay(xDelay);
  }
}

```

รูปที่ ข.7 ฟังก์ชันการกำหนดค่าของมอเตอร์

```

if (motor == 2) {
    // Serial.print(motor);
    // Serial.print(" ");
    // Serial.print(direct);
    // Serial.print(" ");
    // Serial.println(pwm);

    if (direct == CW)
        { digitalWrite(inApin[1], HIGH);    digitalWrite(inBpin[1], LOW);    }
    else if (direct == CCW)
        { digitalWrite(inApin[1], LOW);    digitalWrite(inBpin[1], HIGH);    }
    else
        { digitalWrite(inApin[1], LOW);    digitalWrite(inBpin[1], LOW);    }
    analogWrite(pwmpin[1], pwm);
    // delay(xDelay);
}

if (motor == 3) {
    if (direct == CW)
        { digitalWrite(inApin[2], HIGH);    digitalWrite(inBpin[2], LOW);    }
    else if (direct == CCW)
        { digitalWrite(inApin[2], LOW);    digitalWrite(inBpin[2], HIGH);    }
    else
        { digitalWrite(inApin[2], LOW);    digitalWrite(inBpin[2], LOW);    }
}

```

รูปที่ ข.8 ฟังก์ชันการกำหนดค่าของมอเตอร์ (ต่อ)

```
if (motor == 4) {  
  // Serial.print(motor);  
  // Serial.print(" ");  
  // Serial.print(direct);  
  // Serial.print(" ");  
  // Serial.println(pwm);  
  if (direct == CW)  
  { digitalWrite(inApin[3], HIGH);  
    digitalWrite(inBpin[3], LOW);  
  }  
  else if (direct == CCW)  
  { digitalWrite(inApin[3], LOW);  
    digitalWrite(inBpin[3], HIGH);  
  }  
  else  
  { digitalWrite(inApin[3], LOW);  
    digitalWrite(inBpin[3], LOW);  
  }  
  analogWrite(pwmpin[3], pwm);  
  // delay(xDelay);  
}  
// Stop();  
}
```

รูปที่ ข.9 ฟังก์ชันการกำหนดค่าของมอเตอร์ (ต่อ)

```

void Check_OverCurrent(void)
{
  bool OverCurrent = false;

  OverCurrent = OverCurrent || (analogRead(cspin[0]) > CS_THRESHOLD);
  OverCurrent = OverCurrent || (analogRead(cspin[1]) > CS_THRESHOLD);
  OverCurrent = OverCurrent || (analogRead(cspin[2]) > CS_THRESHOLD);
  OverCurrent = OverCurrent || (analogRead(cspin[3]) > CS_THRESHOLD);

  if (OverCurrent)
  {
    digitalWrite(StatePin, LOW);

    motorOff(1);
    motorOff(2);
    motorOff(3);
    motorOff(4);
  }
  else
    digitalWrite(StatePin, HIGH);
}

void Stop(void)
{
  motorOff(1);
  motorOff(2);
  motorOff(3);
  motorOff(4);

  digitalWrite(Start, 1);
  digitalWrite(Direc, 1);
  digitalWrite(Speed, 1);
}

```

รูปที่ ข.10 ฟังก์ชันการเช็คและตัดกระแสไฟฟ้าเมื่อเกิด Over load

```

void SprintRight(int xDelay, int xSpeed)
{
  //Serial.println("SprintRight"); motorGo(1, CW, xSpeed);

  motorGo(2, CW, xSpeed);

  motorGo(3, CW, xSpeed);

  motorGo(4, CW, xSpeed);

  delay(200);

  Check_OverCurrent();
}

```

รูปที่ ข.11 ฟังก์ชันการควบคุมการเคลื่อนที่ของหุ่นยนต์

```

void Right(int xSpeed)
{
  Serial.println("Right");

  motorGo(4, CCW, xSpeed);

  motorGo(3, CW, xSpeed);

  motorGo(2, CCW, xSpeed);

  motorGo(1, CW, xSpeed);

  delay(200);

  Check_OverCurrent();
}

```

รูปที่ ข.12 ฟังก์ชันการควบคุมการเคลื่อนที่ไปทางขวา

```

void Go(int xDelay, int xSpeed)
{
  Serial.println("Go");

  motorGo(4, CW, xSpeed);

  motorGo(3, CW, xSpeed);

  motorGo(2, CCW, xSpeed);

  motorGo(1, CCW, xSpeed);

  delay(xDelay);

  Check_OverCurrent();
}

```

รูปที่ ข.14 ฟังก์ชันการควบคุมการเคลื่อนที่ไปข้างหน้า

```

void Back(int xDelay, int xSpeed)
{
  Serial.println("Back");

  motorGo(4, CCW, xSpeed);

  motorGo(3, CCW, xSpeed);

  motorGo(2, CW, xSpeed);

  motorGo(1, CW, xSpeed);

  delay(xDelay);

  Check_OverCurrent();
}

```

รูปที่ ข.15 ฟังก์ชันการควบคุมการเคลื่อนที่ถอยหลัง

```

void Check_RemoteCH1ReftLife(void)
{
  int duration = CH1;
  if (duration > (Center_CH1 + Div_Gap))
  {
    PWM_Speed = map(duration, Center_CH1 + Div_Gap, Max_CH1, 0, 255);
    Left(PWM_Speed);
  }
  else if (duration < (Center_CH1 - Div_Gap))
  {
    PWM_Speed = map(duration, Center_CH1 - Div_Gap, Min_CH1, 0, 255);
    Right(PWM_Speed);
  }
  else
  {
    Stop();
  }
}

```

รูปที่ ข.16 ฟังก์ชันการอ่านค่าจาก Remote CH1

```

void Check_RemoteCH2GoBack(void){
  int duration = CH2;
  if (duration > (Center_CH2 + Div_Gap))
  {
    PWM_Speed = map(duration, Center_CH2 + Div_Gap, Max_CH2, 0, 255);
    Back(xDelay, PWM_Speed);
  }
  else if (duration < (Center_CH2 - Div_Gap))
  {
    PWM_Speed = map(duration, Center_CH2 - Div_Gap, Min_CH2, 0, 255);
    Go(xDelay, PWM_Speed);
  }
  else
  {
    Stop();
  }
}

```

รูปที่ ข.17 ฟังก์ชันการอ่านค่าจาก Remote CH2


```
void Check_RemoteCH2Sprint(void)
{
  int duration = CH2;
  if (duration > (Center_CH2 + Div_Gap ))
  {
    PWM_Speed = map(duration, Center_CH2 + Div_Gap, Max_CH2, 0, 255);
    SprintLeft(xDelay, PWM_Speed);
  }
  else if (duration < (Center_CH2 - Div_Gap ))
  {
    PWM_Speed = map(duration, Center_CH2 - Div_Gap, Min_CH2, 0, 255);
    SprintRight(xDelay, PWM_Speed);
  }
  else
  {
    Stop();
  }
}
```

รูปที่ ข.18 ฟังก์ชัน Map ค่าออกไปเป็น Speed

```
#define BRAKE 0

#define CW 1

#define CCW 2

#define CS_THRESHOLD

//MOTOR 1

#define MOTOR_A1_PIN 7

#define MOTOR_B1_PIN 8

//MOTOR 2

#define MOTOR_A2_PIN 4

#define MOTOR_B2_PIN 9

#define PWM_MOTOR_1 5

#define PWM_MOTOR_2 6

#define CURRENT_SEN_1 A2

#define CURRENT_SEN_2 A3

#define EN_PIN_1 A0

#define EN_PIN_2 A1

#define MOTOR_1 0

#define MOTOR_2 1

short usSpeed = 150; //default motor speed

unsigned short usMotor_Status = BRAKE;
```

รูปที่ ข.19 โปรแกรมทดสอบ Motor

```

void setup()
{
  pinMode(MOTOR_A1_PIN, OUTPUT);
  pinMode(MOTOR_B1_PIN, OUTPUT);
  pinMode(MOTOR_A2_PIN, OUTPUT);
  pinMode(MOTOR_B2_PIN, OUTPUT);
  pinMode(PWM_MOTOR_1, OUTPUT);
  pinMode(PWM_MOTOR_2, OUTPUT);
  pinMode(CURRENT_SEN_1, OUTPUT);
  pinMode(CURRENT_SEN_2, OUTPUT);
  pinMode(EN_PIN_1, OUTPUT);
  pinMode(EN_PIN_2, OUTPUT);

  Serial.begin(9600); // Initiates the serial to do the monitoring
  Serial.println("Begin motor control");
  Serial.println(); //Print function list for user selection
  Serial.println("Enter number for control option:");
  Serial.println("1. STOP");
  Serial.println("2. FORWARD");
  Serial.println("3. REVERSE");

  Serial.println("4. READ CURRENT");
  Serial.println("+. INCREASE SPEED");
  Serial.println("-. DECREASE SPEED");
  Serial.println();
};

```

รูปที่ ข.20 แสดง Menu list เพื่อทดสอบ Motor (ต่อ)

```
void Stop()
{
  Serial.println("Stop");
  usMotor_Status = BRAKE;
  motorGo(MOTOR_1, usMotor_Status, 0);
  motorGo(MOTOR_2, usMotor_Status, 0);
}

void Forward()
{
  Serial.println("Forward");
  usMotor_Status = CW;
  motorGo(MOTOR_1, usMotor_Status, usSpeed);
  motorGo(MOTOR_2, usMotor_Status, usSpeed);
}

void Reverse()
{
  Serial.println("Reverse");
  usMotor_Status = CCW;
  motorGo(MOTOR_1, usMotor_Status, usSpeed);
  motorGo(MOTOR_2, usMotor_Status, usSpeed);
}
```

รูปที่ ข.21 ฟังก์ชันทดสอบ Motor (ต่อ)

```
void IncreaseSpeed()
{
  usSpeed = usSpeed + 10;

  if(usSpeed > 255)
  {
    usSpeed = 255;
  }

  Serial.print("Speed +: ");
  Serial.println(usSpeed);

  motorGo(MOTOR_1, usMotor_Status, usSpeed);
  motorGo(MOTOR_2, usMotor_Status, usSpeed);
}

void DecreaseSpeed()
{
  usSpeed = usSpeed - 10;

  if(usSpeed < 0)
  {
    usSpeed = 0;
  }

  Serial.print("Speed -: ");
  Serial.println(usSpeed);

  motorGo(MOTOR_1, usMotor_Status, usSpeed);
  motorGo(MOTOR_2, usMotor_Status, usSpeed);
}
```

รูปที่ ข.22 ฟังก์ชันกำหนดความเร็ว motor

```

void motorGo (uint8_t motor, uint8_t direct, uint8_t pwm){
  if(motor == MOTOR_1)
  {
    if(direct == CW)
    {
      digitalWrite(MOTOR_A1_PIN, LOW);    digitalWrite(MOTOR_B1_PIN, HIGH);
    }
    else if(direct == CCW)
    {
      digitalWrite(MOTOR_A1_PIN, HIGH);    digitalWrite(MOTOR_B1_PIN, LOW);
    }
    else
    {
      digitalWrite(MOTOR_A1_PIN, LOW);    digitalWrite(MOTOR_B1_PIN, LOW);
    }
    analogWrite(PWM_MOTOR_1, pwm);
  }
  else if(motor == MOTOR_2)
  {
    if(direct == CW)
    {
      digitalWrite(MOTOR_A2_PIN, LOW);    digitalWrite(MOTOR_B2_PIN, HIGH);
    }
    else if(direct == CCW)
    {
      digitalWrite(MOTOR_A2_PIN, HIGH);    digitalWrite(MOTOR_B2_PIN, LOW);
    }
    else
    {
      digitalWrite(MOTOR_A2_PIN, LOW);    digitalWrite(MOTOR_B2_PIN, LOW);
    }
    analogWrite(PWM_MOTOR_2, pwm);
  }
}
{ usSpeed = usSpeed + 10; if(usSpeed > 255) { usSpeed = 255; }
  Serial.print("Speed +: ");
  Serial.println(usSpeed);
  motorGo(MOTOR_1, usMotor_Status, usSpeed);
  motorGo(MOTOR_2, usMotor_Status, usSpeed); }
}
}

```

รูปที่ ข.23 ฟังก์ชันกำหนดทิศทางของ motor

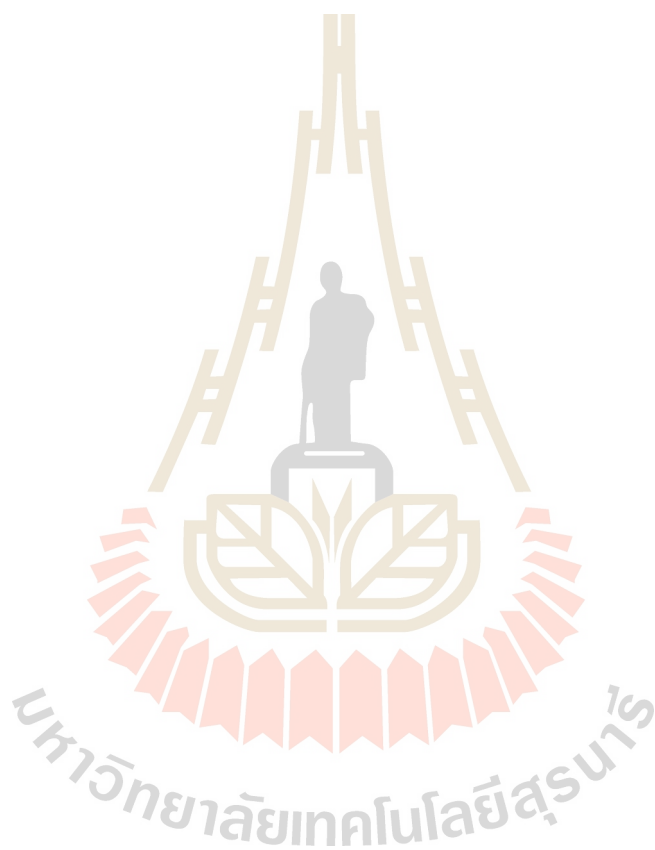


ภาคผนวก ก

บทความทางวิชาการที่ได้รับการตีพิมพ์เผยแพร่

รายชื่อบทความที่ได้รับการตีพิมพ์เผยแพร่ในระหว่างการศึกษา

Pornjumsilp, S., and Chamniprasart, K. (2018). **DESIGN AND PROTOTYPING OF SPHERICAL ROBOT.** The 12th South East Asian Technical University Consortium (SEATUC) Symposium. March 12 – 13, 2018. Yogyakarta, Indonesia. 6 pp.



DESIGN AND PROTOTYPING OF SPHERICAL ROBOT

Siwasilp Pornjumsilp, Kontorn Chamniprasart⁽¹⁾

⁽¹⁾ Insititute of Engineering Suranaree University of Technology, Thailand

Email: kontorn@sut.ac.th

ABSTRACT

Nowadays, robotic technology has become increasingly important to human life. Robots are applied in several industries, e.g. agricultural industry, or even the film industry. Different kinds of robot could be seen in the movies. One of the most popular robot is the "BB-8" (ASTROMECH DROID) in the famous science-fiction film named STAR WARS VII, this robot is very interesting because it requires cognitive and imagination skills for creating this robot. The "BB-8" robot in the future will inspire the creation, and invention of the new generation of engineers. The advantages of this robot, besides the common movement of the robot, is the control technology that allows robots to interact with humans, including operating system modules, and feature enhancements. This robot that can be used with many applications. For example robots for guiding people with disabilities, robot for guiding location or giving placement tips, etc. However, the cost of producing this robot is very high, making it hard for the public to access it. People know the "BB-8" only in movies. The details of the "BB-8" production are confidential, resulting in the difficulty of developing this type of robot to other applications. This robot is expected to emerge with embedded system in order to enhance the capability of the robots for a variety of applications.

KEYWORDS: BB8, Spherical robot, rolling motion.

1. INTRODUCTION

Robot technology has come to play an important role in the daily lives of human. As the well-known "BB-8" robot in the epic science, fiction film will inspire the creation of engineers who designed the experiment and create new inventions for engineers [1], and designers in the future [2]. This robot consists of the control technology that can make a robot interact with a human such as the module system functions, and other features.

The robot is also able to deploy in a wide range of application, such as for navigation, robots for disable people, and robot for guided tours, etc. However the cost of producing robots model is slightly high. The robot production is not publically known in details, it is only known by movie makers. The robot can also increase the capacity of several technology. This research aims to design and build a spherical motion robot [3]. With the rolling motion control via a remote control. The robot prototype is preliminary built to take advantage of navigation which is used to find books in the library. The robot is used in the University to welcome important guests to provide students that can lead to the development of education, experience, and knowledge. Of them it helps engineers to understand the principles of design the spherical motion robot. The prototype robot will be also developed further in the future, which will enhance the ability of the engineering and technology applications.

2. EXPERIMENT

SPHERICAL ROBOT is a robot that moves by sprinting itself to move forward. The user can control the direction of movement they user wants. The robot must be able to hold the head mounted above the sphere. The overall system consists of 1). Drive motor, 2).contorted radio receiver that is used to link to users with the robot including the directional control, 3). Transmitter will transmit a signal to the processing unit inside the robot 4).a set of magnets used to support the head to attach to the outer shell. The "BB-8" is analysis the structure using visual analysis and then increase the robot scale. This wheel, which is put inside the robot, is designed to align on the concave surface of the robot body, helping the robot to more forward, this research is another technique for constructing a robot moving system within the arc of a sphere, with a head which can move freely without falling apart from top shell.

2.1 Apparatus

The experiment was divided into three parts.

Part 1. How to make a robot move within a sphere effectively.

Part 2. How to connect users with robots to control the direction of robot movement as you want.

Part 3. How to install the robot head to the body in Fig 1.

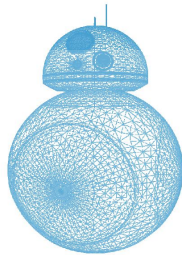


Fig 1. SPHERICAL ROBOT Master

2.2 equipment

2.2.1 Microcontroller

The microcontroller is a device that can create control system. This device is small. It is a semiconductor device with integrated automatic functions. Similar to computer, it is an internal device that contains, processor, and ports for connection.

2.2.1.1 General components of microcontrollers

1. Control Processing Unit
2. Memory consist of RAM and EEPROM/EPROM/PROM/ROM
3. Input / Output With parallel expansion ports and Serial port
4. Timer
5. Interrupt Controller

In this research, we used Arduino mega 2560 in Fig 2 to contact the driver board and get data from the receiver to get the information to control the robot.

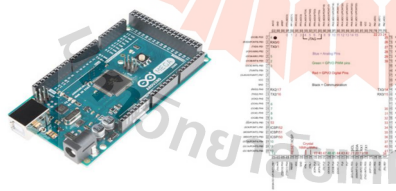


Fig 2. Arduino mega 2560

2.2.2 Remote Controller

Remote control is an electronic device. That is used to control the operation of the invention or the machine, especially home appliances such as TVs, audio, and DVD players from a distance without using of power cord. It consists of two things: the code, and the transmitter. 1).the code is the signal to be removed. 2). carrier receiver used to remove or receive the code then sent to control the device. In this research, used Radio link AT9S in Fig 3.



Fig 3. Radio link AT9S

Property

- The DSSS spread spectrum system has a very high signal-to-noise ratio.
- 2.8-inch color screen, 2.4G system, firmware V 1.1.8
- Vibration alarm system can alert and vibrate.
- The telemetry system can be used to receive information on the aircraft from a distance, such as voltage, battery, motor speed, GPS on the radio screen.
- 7.4V radio. Up to 18V. Anti-reverse anti-pin batting.
- The USB port can update the latest information from the manufacturer online.
- The distance on the ground about 900 meters on the air about 1.5 km.

2.2.2 Driver Motor

Generally, the driver in used to drive a motor for the robot movement, which requires to control the direction and speed of the motor.



Fig 4. Monster Moto Shield VNH2SP30

Description

Monster Moto Shield VNH2SP30 in Fig 4, 30A motor driver Arduino driver module uses IC chip VNH2SP30 to drive current up to 30A. This board, when used in extreme heat, should improve the cooling performance.

Property

- Voltage max: 16V
- Maximum current rating: 30 A
- Practical Continuous Current: 14 A
- Analogue current detection
- MOSFET Resistance: 19 mΩ (per pin)
- Maximum PWM Frequency: 20 kHz
- Thermal Shutdown
- Under voltage and Overvoltage shutdown

2.2.3 Permanent magnet

The neodymium magnet [4] in Fig 5 is the magnetic made of a rare magnetic mineral called neodymium mixed with other minerals. Now it is a permanent magnet. That is the most powerful in the world.

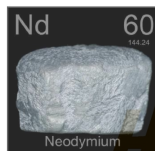


Fig 5. Neodymium [Nd, 60]

Details about neodymium magnet

- Neodymium magnets are produced from neodymium, iron, boron and alloy, Tetragonal Crystalline.
- Magnetic field strength (Remanence) 1 - 1.4 T
- Coercively 750-2000 kA/m
- Density of magnetic energy of 200-400 kJ/cu.m
- Magnetic Loss (Curie temperature) 310-400 Celsius
- Neodymium magnet used to produce equipment that requires high power magnet such as hard disk MRI scanner (magnetic MRI) in the motor, etc.
- The force of the neodymium magnets is enough to lift the iron more than 1,000 times its own iron.



Fig 6. Test Neodymium magnet

Fig 6: Demonstrates the strength of neodymium magnet. Clearly Round ball of 3.63 cm diameter weighs 196.1 g.

Three round steel balls are connected Neodymium magnet with a diameter of 4 mm, thickness 1.5 mm, weight 0.143 g.

2.3 Technique

The robot production is consists of four following steps

2.3.1 A sphere construction

Fig 7: The ball is made of fiberglass casting. The process begins with cutting the foam and then polishing it to get a ball with a radius size 33 cm. After that we wrap it with fiberglass cloth and spray with the resin solution. In this place we can make half of a ball, then put together to get the ball as we need. After completing the process of creating the ball, we cut it into crescents in order to get the niche for maintenance and screw holes to close the nut again. Then, we color it with a shade of gray to reduce scratches and make it easy to clean. Finally, we wax it.



Fig 7. The robot spherical body.

2.3.2 An internal robot construction.

We choose aluminum as the material for the base of the robot because it is light and strong. The wheels are designed According to the concavity of the inner surface of the robot body, to align with the concave surface to push the robot moving forward powerful in Fig 8.



Fig 8. Internal components of the robot.



Fig 9. Robot mark IV.

Previously, we tried to design and build prototypes as shown in Fig 10 and 11.

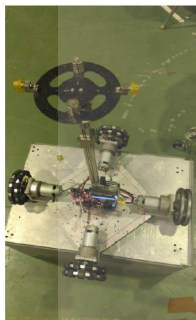


Fig 10. Robot mark II.

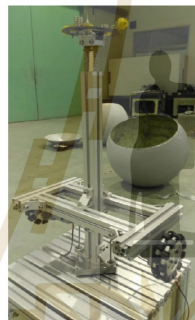


Fig 11. Robot mark III.

We take advantages and disadvantages in each version and been updated to version 4 in Fig 9.

2.3.3 A control system

In this experiment, we use Arduino mega 2560 for the main processor. The controller receives the signal from receiver [pulse Input] and convert a signal pulse format to PWM format. However, it has to conduct a mapping process before converting the signal, as follow in Fig 12:

```
void loop() {
  loopTime = micros() - previousLoopTime;
  if (loopTime >= 10000)
  {
    previousLoopTime = micros();
    CHA = pulseIn(ch1pin, HIGH) - default_pInChA;
    CHB = pulseIn(ch2pin, HIGH) - default_pInChB;
    CHA = Deadband(CHB, Pulse_noiseCh1);
    CHB = Deadband(CHB, Pulse_noiseCh2);
    left_w = -(CHA - CHB);
    right_w = -(CHA + CHB);
  }
}
```

```
currentValue1 = analogRead(CS1) * 0.035; // Motor Driver 1
currentValue2 = analogRead(CS2) * 0.035; // Motor Driver 2
```

```
if (currentValue1 < currentLimit) xxx = OutputToMotor1(left_w);
else xxx = OutputToMotor1(0);
if (currentValue2 < currentLimit) yyy = OutputToMotor2(right_w);
else yyy = OutputToMotor2(0);
analogWrite(pwmApin, xxx);
analogWrite(pwmBpin, yyy);
}
}
double Deadband(double value, double limit)
{
  double temp = 0.0;
  if (value >= limit) temp = value - limit;
  else if (value <= -limit) temp = value + limit;
  else temp = 0.0;
  return temp;
}
int OutputToMotor1(int value)
{
  int temp = 0;
  if (value >= 0)
  {
    digitalWrite(inApin_1, LOW);
    digitalWrite(inBpin_1, HIGH); temp = map(value, 0, 500, 0, 255);
  } else {
    digitalWrite(inApin_1, HIGH);
    digitalWrite(inBpin_1, LOW); temp = map(-value, 0, 500, 0, 255);
  }
  return temp;
}
int OutputToMotor2(int value)
{
  int temp = 0;
  if (value >= 0)
  {
    digitalWrite(inApin_2, LOW);
    digitalWrite(inBpin_2, HIGH); temp = map(value, 0, 500, 0, 255);
  } else {
    digitalWrite(inApin_2, HIGH);
    digitalWrite(inBpin_2, LOW); temp = map(-value, 0, 500, 0, 255);
  }
  return temp;
}
```

Fig 12. Part of the programming

2.3.4 A robot head production and installation.

Fig 13: The demo robot head is a semi-spherical ball with a diameter of 14 in. It is made of aircraft foam and hollow in the head in order to reduce the weight.



Fig 13. Test and install the demo head.



Fig 14. Head ready for next job.

2.4 Design

In this section, we are based on the movie in Fig 1. Clean and modified robot, and enter details, for head ready for next job in Fig 15.

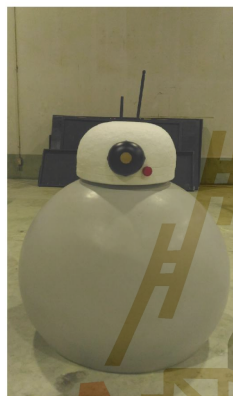


Fig 15. Shell of robot and head.

3. ANALYSIS

The hardware developer is divided into three parts as follows:

3.1 How to make it sprint.

In this section, the robot movement is described based on football movement Fig 16.

Linear motion $F - f = ma$
 Angular movement $f r = I \alpha$
 if not slip $\alpha = \frac{a}{r}$
 $\frac{I \alpha r^2}{I} = a$
 $F - f = \frac{m f r^2}{I}$

$$f = \frac{F}{1 + \frac{m r^2}{I}}, \quad a = \frac{F r^2}{I + m r^2}$$

If there is no other force $F = 0$
 $\Rightarrow a = 0, f = 0$ Speed will be constant and no friction.

Fig 16. Movement equation.

Movement [5] of the object can sometimes be shifted along with rotational motion, as ball motion, golf and bicycle wheel motion. The object rotates about the central axis. In the study of rolling objects, for example, a wheel that runs on a flat road at a constant speed. When considering the kinetic energy of a rolling motion of an object, it is equal to the total of kinetic energy in Fig

$$K_{rolling} = \frac{1}{2} I_{CM} \omega^2 + \frac{1}{2} m v_{CM}^2 \quad \text{Kinetic energy of rolling}$$

From $v = \omega R$ That will $K_{rolling} = \frac{1}{2} I_{CM} \left(\frac{v_{CM}}{R} \right)^2 + \frac{1}{2} m v_{CM}^2$

To rewrite $K_{rolling} = \frac{1}{2} \left(\frac{I_{CM}}{R^2} + M \right) v_{CM}^2$ energy of rolling

Fig 17. Rolling equation.

On the other hand, we can press the straight edge inside the ball to push the ball move forward, shows how the ball can move forward by moving the weight inside the ball in Fig 18.



Sec 1 ===== Sec 2 ===== Sec 3 =====> Time

Fig 18. Pressing Straight edge Ball.

Sec 1 => the ball is stopped.
 Sec 2 => the ball is moving. [Move the red dot (mass)]
 Sec 3 => the ball is moving on for a while and stopped.

3.3 Control variable for robot head.

Form 2.2.3 Permanent magnet, we use Neodymium magnet for stretching the body and head in Fig 19.

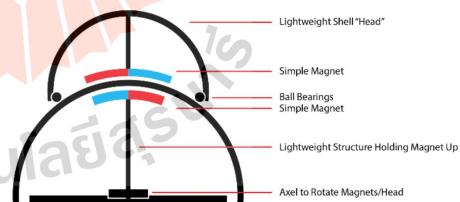


Fig 19. Head section.

Using Neodymium type N52 [7] magnet in Fig 20, the distance between the head magnet and the body magnet need to be maintain at 4 mm. to balance the robot and stabilize the robot hade on the body.

Using Neodymium type N52 size 12x3 mm. Hole 4 mm.

12x3mm Hole 4mm



Fig 20. Use this Magnetic.

We also use the same method to move the robot Fig 21.

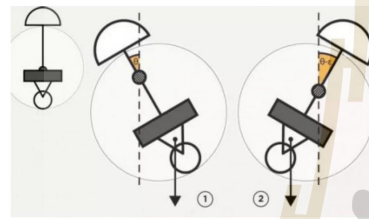


Fig 21. Robot forward, backward.

3.4 Discuss the achievement.

From the above-mentioned modules, we assemble all modules together and test it. As the result of the experiment, robots can move to a specified position as intended and achieve the goals set by the researcher. Although during initial testing and prototyping, problems are encountered throughout the production: the molding of the robes used to cover the robot; the magnetic field is not uniform; batteries are not enough for use; the length of the knitting needle stitches is too long, causing the robot inside to stumble and cannot move forward; non-balanced weight of the head with the selected magnetic set; and speed and direction of the turn is too much, resulting in the filing of the head of the body. After analyzing and solving the above-mentioned problems, all problem is fixed, the robot can move to specified position correctly and performance.

4. CONCLUSION

This paper has proposed the procedure of prototyping a Spherical robot. As the results of robot testing, the movement of the robot can be balanced by controlling movement speed, the distance between the head magnet and body magnet and the Turing speed of the left and right wheel. The capacity of battery should be concerned for the robot complete the tasks. This type of robot will be emerged the human interface system to enhance the capability of the robot for wider range of application in the future.

5. REFERENCES

- [1]. Karavaev Y.L., Kilin A.A. (2015) The dynamics and control of a spherical robot with an internal Omni wheel platform Regular and chaotic dynamics, Vol. 20 (2015), pp.134-152 CrossRefView Record in Scopus
- [2]. Yu et al. (2013) Stabilization and control of a spherical robot, Research Journal of Applied Sciences, Engineering and Technology, 5 (6) (2013), pp. 2289-2296
- [3]. Chase and Pandya (2012) A review of active mechanical driving principles of spherical robots Robotics (2012), pp. 3-23 CrossRefView Record in Scopus
- [4]. D.C. Morton, Astrophys (2000) Neodymium (Nd) D. C. Morton, Astrophys. J. Suppl. Ser., 130, 403 (2000).
- [5]. Svinin et al. (2014) A motion planning strategy for a spherical rolling robot driven by two internal rotors, IEEE transactions on robotics, Vol.30 (2014), and pp.993-1002
- [6]. Borisov et al. (2012) How to control Chaplygin's sphere using rotors Regular and Chaotic Dynamics, 17(3) (2012), pp. 258-272
- [7]. W. C. Martin, R. Zalubas, and L. Hagan, (1978) Atomic Energy Levels - The Rare Earth Elements The Rare Earth Elements, Natl. Stand. Ref. Data Ser., Natl. Bur. Stand. (U.S.) 60 (1978). Wavelengths for which this reference is given are calculated from the energy levels.

PHOTOS AND INFORMATION



Siwasilp Pornjumsilp

Received the Received the B.E. (2015) degree in Computer Engineering from Suranaree University of Technology



Kontorn Champrasart

Received the B.S. (1980) degree in Mechanical Engineering from (First Class Honer) from Royal Thai Air Force Academy, Thailand, M.Sc. (1987) degree in Mechanical Engineering from University of Pittsburgh, and Ph.D. (1992) degree in Mechanical Engineering from University of Pittsburgh, system.

ประวัติผู้เขียน

นายศิวศิลป์ พรจำศิลป์ เกิดเมื่อวันที่ 31 สิงหาคม 2536 ที่อำเภอเมือง จังหวัดนครราชสีมา เริ่มการศึกษาระดับอนุบาลที่โรงเรียนอนุบาลดวงแข ระดับประถมศึกษาปีที่ 1-6 ที่โรงเรียนสุขานารี และมัธยมศึกษาปีที่ 1-6 ที่โรงเรียนราชสีมาวิทยาลัย 2 อำเภอเมือง จังหวัดนครราชสีมา สำเร็จการศึกษาระดับปริญญาตรี (วิศวกรรมคอมพิวเตอร์) สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2559 และได้ศึกษาต่อระดับวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ ขณะศึกษาได้เป็นผู้สอนรายวิชาปฏิบัติการของสาขาวิชาวิศวกรรมเครื่องกล สาขาวิชาวิศวกรรมยานยนต์ สาขาวิชาวิศวกรรมการผลิต สาขาวิชาวิศวกรรมเมคคาทรอนิกส์ และสาขาวิศวกรรมคอมพิวเตอร์ จำนวน 12 รายวิชา ได้แก่

- (1) ปฏิบัติการระบบอัตโนมัติอุตสาหกรรม
- (2) ปฏิบัติการระบบควบคุมและอัตโนมัติ
- (3) ปฏิบัติการวิศวกรรมเมคคาทรอนิกส์ 1
- (4) ปฏิบัติการวิศวกรรมเมคคาทรอนิกส์ 2
- (5) ปฏิบัติการวิศวกรรมเมคคาทรอนิกส์ 3
- (6) ปฏิบัติการฟิสิกส์สำหรับวิศวกร 1
- (7) ปฏิบัติการโปรแกรมคอมพิวเตอร์สำหรับงานประยุกต์ทางวิศวกรรม
- (8) ปฏิบัติการอิเล็กทรอนิกส์สำหรับวิศวกรรมคอมพิวเตอร์
- (9) ปฏิบัติการระบบดิจิทัล
- (10) ปฏิบัติการไมโครโพรเซสเซอร์
- (11) ปฏิบัติการระบบฝังตัว (EMBEDDED SYSTEMS)
- (12) ปฏิบัติการอินเทอร์เน็ตทุกสรรพสิ่ง (Internet of Things)

ระหว่างการศึกษาในระดับปริญญาโทได้นำเสนอผลงานทางวิชาการ เรื่อง DESIGN AND PROTOTYPING OF SPHERICAL ROBOT ในการประชุมนานาชาติ SOUTH EAST ASIAN TECHNICAL UNIVERSITY CONSORTIUM (SEATUC) Symposium 2018 ณ Graduate School Building UGM (Gedung Pascasarjana) เมือง Yogyakarta ประเทศ Indonesia.

รวมทั้งเป็นวิทยากรอบรม นักเรียน นักศึกษา หลักสูตรความรู้ด้านการออกแบบและสร้างหุ่นยนต์ กิจกรรม Robot Design Contest 2017 – 2018 ภาคตะวันออกเฉียงเหนือ