การศึกษาและประกอบหุ่นยนต์หกขา

STUDY AND CONSTRUCTION HEXAPOD ROBOT

นฤทัศน์ โตธนะคุณ
นัฐพส สุภาจารธรรมคุณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การศึกษาและประกอบหุ่นยนต์หกขา

STUDY AND CONSTRUCTION HEXAPOD ROBOT

นฤทัศน์ โตธนะคุณ
นัฐพส สุภาจารธรรมคุณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STUDY AND CONSTRUCTION HEXAPOD ROBOT

NARUTAS TOTANAKUN

NUTTROSE SUPARJARRATHAMMAKHUN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	การศึกษาและประกอบหุ่นยนต์หกขา STUDY AND CONSTRUCTION HEXAPOD ROBOT		
นักศึกษาผู้จัดทำ	นายนฤทัศน์ โตธนะคุณ	**รหัสนักศึกษา**	53010813
	นายนัฐพส สุภาจารธรรมคุณ	**รหัสนักศึกษา**	53010831
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมการวัดคุม		
ปีการศึกษา	2556		

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
ผศ.ดร.พงษ์ชัย นิลาศ	พงษ์ชัย 27 มิ.ย. 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การศึกษาและประกอบหุ่นยนต์หกขา			
	STUDY AND CONSTRUCTION HEXAPOD ROBOT			
นักศึกษาผู้จัดทำ	นายนฤทัศน์	โตธนะคุณ	รหัสนักศึกษา	53010813
	นายนัฐพส	สุภาจารธรรมคุณ	รหัสนักศึกษา	53010831
อาจารย์ที่ปรึกษา	อาจารย์พงษ์ชัย	นิลาศ		
ปีการศึกษา	2556			

บทคัดย่อ

โครงงานฉบับนี้มีวัตถุประสงค์เพื่อศึกษา และสร้างหุ่นยนต์หกขา โดยหุ่นยนต์นี้สามารถเดินบนพื้นผิวขรุขระได้ ในโครงงานนี้ประกอบด้วยการออกแบบและสร้างโครงของหุ่นยนต์หกขา รวมถึงการศึกษาลักษณะการเดินของหุ่นยนต์หกขา โดยควบคุมการเคลื่อนที่ของหุ่นยนต์ผ่าน 32 Channel Servo Motor Control Driver Board for Arduino ซึ่งทำหน้าที่ควบคุมการทำงานของเซอร์โวมอเตอร์ และใช้เซอร์โวมอเตอร์เป็นส่วนประกอบหลักในการเคลื่อนที่ของข้อต่อขาและชิ้นส่วนขา นอกจากนี้ศึกษาแนวทางการเพิ่มประสิทธิภาพการเดินของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	STUDY AND CONSTRUCTION HEXAPOD ROBOT
Authors	Mr. Narutas Totanakun
	Mr. Nuttrose Suparjarrathammakhun
Thesis Advisor	Assit.Prof.Dr. Phongchai Nilas
Year	2013

ABSTRACT

This Hexapod Robot project focuses on study, design as well as hardware assembly the spider-like robot. The project employs a 32-channels control board, Arduino UNO R3 microcontroller, 18 servo motors and a robot behavior program. The program is written in C to receive parameter values form computer and transmits to controller to activate the robot movement. The Hexapod Robot is based on pre-modeled behaviors that observed from experiments. These pre-modeled behaviors include the movement joint angles, the collaboration of legs, and the walking patterns. The experiment results could show the robot's ability to walk forward, walk backward, turn, and across small obstacles.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง การศึกษาและประกอบหุ่นยนต์หกขา (STUDY AND CONSTRUCTION HEXAPOD ROBOT) สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอุปถัมภ์จากบุคคลหลายฝ่ายที่ให้คำปรึกษา และชี้แนะแนวทาง ทำให้ปริญญานิพนธ์ฉบับนี้บรรลุเป้าหมายตามวัตถุประสงค์ได้เป็นอย่างดี

คณะผู้จัดทำขอขอบพระคุณ ผศ.ดร.พงษ์ชัย นิลาศ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้คำปรึกษา ชี้แนะ ให้ข้อคิดในการแก้ปัญหา รวมถึงช่วยตรวจทานแก้ไขข้อบกพร่องต่างๆ ทำให้ปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์ นอกจากนี้คณะผู้จัดทำขอขอบพระคุณ คณะอาจารย์ประจำสาขาวิชาวิศวกรรมการวัดคุม ที่ให้คำปรึกษา

ขอขอบพระคุณชุมนุมโรบอท ที่ให้คำปรึกษาในด้านเทคนิคการทำหุ่นยนต์ อีกทั้งยังเอื้อเฟื้ออุปกรณ์ และเครื่องมือในการทำงานต่างๆ

และสุดท้ายนี้คณะผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา และครอบครัว เป็นอย่างสูงสำหรับความรัก กำลังใจ คำปรึกษา และการสนับสนุนในด้านต่างๆ ที่มอบให้อยู่อย่างสม่ำเสมอจนเกิดเป็นแรงผลักดัน ทำให้โครงงานนี้ประสบความสำเร็จและผ่านลุล่วงมาได้อย่างสมบูรณ์

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VII

สารบัญตาราง IX

สารบัญกราฟ X

บทที่ 1 บทนำ 1

1.1 ความสำคัญของปริญญานิพนธ์ 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1

1.3 ขอบเขตของปริญญานิพนธ์ 1

1.4 ขั้นตอนการศึกษา 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 3

2.1 เซอร์โวมอเตอร์ 3

2.1.1 หลักการทำงานของเซอร์โวมอเตอร์ 4

2.1.2 สายเชื่อมต่อของเซอร์โวมอเตอร์ 5

2.2 โครงสร้างและส่วนประกอบของ 32 Channel Servo Motor Control Driver Board for Arduino Robot 5

2.2.1 คุณสมบัติของบอร์ด 5

2.2.2 คำแนะนำในการควบคุม Servo Motor 6

2.2.3 การเชื่อมต่อ Servo ไปยังบอร์ด 7

2.2.4 ซอฟแวร์ที่ใช้ควบคุมบอร์ด 7

2.2.5 Secondary development 9

2.3 Arduino 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.3.1 บอร์ด Arduino รุ่น Uno R3 12
2.3.2 การลงโปรแกรม Arduino 12
2.4 Universal Battery Elimination Circuit 14
2.5 โปรแกรม SolidWorks 14
2.5.1 ขั้นตอนการสร้างชิ้นงาน 3 มิติและแบบสั่งงานในโปรแกรม SolidWorks 15
2.5.2 หลักการเขียนแบบพื้นฐานที่ควรรู้ 16
2.6 โปรแกรม Altium Design 18
2.7 ลักษณะการเดินของหุ่นยนต์หกขา 18
2.7.1 การเดินทีละ 3 ขา(Tripod Gait) 18
2.7.2 การเดินทีละ 2 ขา(Dual-leg Gait) 20
2.7.3 การเดินทีละขา(Single-Leg Gait) 21

บทที่ 3 การออกแบบและการสร้าง 22
3.1 กล่าวนำ 22
3.2 การออกแบบโครงสร้าง 22
3.3 การเลือกวัสดุและตัดชิ้นส่วนที่ได้ทำการออกแบบ 25
3.4 การประกอบชิ้นส่วน 25
3.5 การติดต่อสื่อสารและการส่งข้อมูลระหว่างอุปกรณ์ 28
3.5.1 การติดต่อสื่อสารระหว่างบอร์ด Arduino UNO R3 กับบอร์ด Torobot 32 channel serial servo controller 28
3.5.2 การเชื่อมต่อสายสัญญาณระหว่าง Servo Motor 18 ตัว เข้ากับบอร์ด Torobot 32 channel serial servo controller 29
3.5.3 การปรับมุมของ Servo Motor 30
3.6 การโปรแกรมข้อมูลบนบอร์ด Arduino UNO R3 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 การทดลอง **33**

4.1 กล่าวนำ 33

4.2 ขั้นตอนการทดลอง 33

4.3 ผลการทดลอง 34

4.3.1 การทดลองหาค่ามุมของรูปแบบการยืน 34

4.3.2 การทดลองหาค่ามุมของรูปแบบการเดิน 35

4.3.3 การทดลองหาค่ามุมของรูปแบบการเลี้ยวซ้าย 38

4.3.4 การทดลองหาค่ามุมของรูปแบบการเลี้ยวขวา 40

4.3.5 การทดลองหาค่ามุมในรูปแบบการเดินข้ามสิ่งกีดขวาง 42

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ **44**

5.1 สรุปผลการทดลอง 44

5.2 ข้อเสนอแนะ 44

บรรณานุกรม **45**

ภาคผนวก **46**

ภาคผนวก ก.แบบร่างชิ้นส่วนของโครง hexapod robot 47

ภาคผนวก ข.โปรแกรม 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	แสดงเซอร์โวมอเตอร์ (Servo Motor)	3
2.2	แสดงการทำงานของเซอร์โวมอเตอร์	4
2.3	สายเชื่อมต่อของเซอร์โวมอเตอร์	5
2.4	แสดงบอร์ด 32 Channel Servo Motor Control Driver Board for Arduino Robot	6
2.5	แสดงแหล่งจ่ายไฟของ Servo และแหล่งจ่ายไฟของ Chip	6
2.6	แสดงการเชื่อมต่อ Servo กับบอร์ด	7
2.7	แสดงการควบคุม Servo ตัวเดียว	8
2.8	การควบคุม Servo หลายตัวพร้อมกัน	8
2.9	แสดงรูปบอร์ด Arduino ที่ถูกพัฒนาและเผยแพร่ออกมา	11
2.10	แสดงรูปบอร์ด Arduino รุ่น Uno R3	12
2.11	แสดง icon โปรแกรม Arduino IDE	13
2.12	แสดงหน้าต่างของโปรแกรม Arduino IDE	13
2.13	แสดงการใช้งาน UBEC	14
2.14	แบบจำลอง 3 มิติ (ก) แบบ Wireframe และ (ข) แบบ Solid Modeling	14
2.15	ชิ้นงานหรือชิ้นส่วนแบบ Part ประกอบด้วยหลายๆ Features	15
2.16	ชิ้นงานประกอบแบบ Assembly ประกอบด้วย Parts และ Sub-assembly	15
2.17	มุมมอง 3 มิติที่นิยมในการเขียนแบบ	16
2.18	การเขียนภาพฉาย orthographic แบบมุมที่หนึ่ง และสัญลักษณ์	17
2.19	การเขียนภาพฉาย orthographic แบบมุมที่สาม และสัญลักษณ์	17
2.20	แสดงโปรแกรม Altium Design	18
2.21	การเดินทีละ 3 ขา แบบที่ 1	19
2.22	การเดินทีละ 3 ขา แบบที่ 2	19
2.23	การเดินทีละ 3 ขา แบบที่ 3	19
2.24	การเดินทีละ 2 ขา แบบที่ 1	20
2.25	การเดินทีละ 2 ขา แบบที่ 2	20
2.26	การเดินที่ละขา	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.1	แบบร่างของหุ่นยนต์หกขาเมื่อนำแต่ละชิ้นส่วนมาประกอบกัน	22
3.2	แบบร่างชิ้นส่วนด้านบนของหุ่นยนต์หกขา	23
3.3	แบบร่างชิ้นส่วนด้านล่างของหุ่นยนต์หกขา	23
3.4	แบบร่างชิ้นส่วนข้อต่อที่ยึดกับ Servo Motor	23
3.5	แบบร่างชิ้นส่วนที่ยึดเฟืองของ Servo Motor	24
3.6	แบบร่างชิ้นส่วนขา	24
3.7	แบบร่าง Servo Motor จำนวน 18 ชิ้น	24
3.8	แสดงโครงของหุ่นยนต์หกขาจากแผ่นอะลูมิเนียมตอนประกอบเสร็จแล้ว	25
3.9	แสดงการยึดชิ้นส่วนต่างๆเข้ากันด้วย Servo Motor	25
3.10	แสดงโครงสร้างของหุ่นยนต์หกขา	26
3.11	แสดงการติดตั้งบอร์ด Arduino UNO R3	26
3.12	แสดงอุปกรณ์ UBEC	27
3.13	การเชื่อมต่อแบตเตอรี่(Lipo) เข้ากับบอร์ดจ่ายไฟ	27
3.14	หุ่นยนต์หกขาที่ประกอบเสร็จแล้ว	27
3.15	แผนภาพการติดต่อสื่อสารและการส่งข้อมูลระหว่างอุปกรณ์	28
3.16	แผนภาพการเชื่อมต่อระหว่างบอร์ด Arduino กับบอร์ด Torobot	28
3.17	แผนภาพการเชื่อมต่อสัญญาณระหว่าง Servo Motor กับบอร์ด Torobot	29
3.18	แสดงสาย USB Port เชื่อมต่อ Computer กับบอร์ด Arduino UNO R3	31
3.19	แสดงการเลือกรุ่นของบอร์ดในโปรแกรม Arduino IDE	31
3.20	แสดงการตรวจสอบค่าที่ถูกส่งจาก Computer ไปยังบอร์ด Arduino UNO R3	32
4.1	รูปแบบการยืน	35
4.2	รูปแบบการเดิน	37
4.3	รูปแบบการเลี้ยวซ้าย	39
4.4	รูปแบบการเลี้ยวขวา	40
4.5	รูปแบบการเดินข้ามสิ่งกีดขวาง	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงคำสั่งพื้นฐาน 9
3.1 แสดงการหาค่ามุมที่จะใช้โปรแกรม เทียบจากมุม 90 องศา 30
4.1 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการยืน 34
4.2 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการเดิน 35
4.3 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการเลี้ยวซ้าย 38
4.4 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการเลี้ยวขวา 40
4.5 ผลการทดลองหาค่ามุมของ Servo Motor ในการเดินข้ามสิ่งกีดขวาง 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญกราฟ

กราฟที่ **หน้า**

4.1 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเดิน 36
4.2 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเลี้ยวซ้าย 39
4.3 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเลี้ยวขวา 41
4.4 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเดินข้ามสิ่งกีดขวาง . 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

เนื่องจากหุ่นยนต์เริ่มเข้ามามีบทบาทกับชีวิตประจำวันของมนุษย์มากขึ้น เทคโนโลยีที่ได้รับการพัฒนาอย่างต่อเนื่องในปัจจุบัน ทำให้ความสามารถของหุ่นยนต์พัฒนาขึ้นอย่างรวดเร็ว สามารถทำงานต่าง ๆ ที่มนุษย์ไม่สามารถทำได้ ซึ่งสามารถแบ่งประเภทตามความสามารถของหุ่นยนต์ได้แก่ ความสามารถในด้านการแพทย์ ความสามารถในงานวิจัย ความสามารถในงานอุตสาหกรรม ความสามารถในด้านความมั่นคง และความสามารถในด้านบันเทิง สำหรับหุ่นยนต์หกขาจัดอยู่ในประเภทความสามารถในงานวิจัย ใช้สำรวจงานวิจัยทางด้านวิทยาศาสตร์ร่วมกับมนุษย์ เพื่อเก็บบันทึกข้อมูลการเปลี่ยนแปลงต่าง ๆ ซึ่งเป็นงานเสี่ยงอันตรายที่เกินขอบเขตความสามารถของมนุษย์ที่ไม่สามารถปฏิบัติงานสำรวจได้ “หุ่นยนต์หกขา” หรือเรียกอีกชื่อว่า Hexapod Robot จะประกอบด้วยตัวโครงของหุ่นยนต์ มีข้อต่อของชิ้นส่วนขาจำนวน 6 ชุด ข้อต่อแต่ละส่วนสามารถเคลื่อนที่ได้ด้วยเซอร์โวมอเตอร์ควบคุมการทำงานด้วย 32 Channel Servo Motor Control Driver Board for Arduino และ บอร์ด Arduino UNO R3 มีการศึกษาโปรแกรมที่เป็นคำสั่งในการควบคุมการเคลื่อนที่ในรูปแบบต่างๆ ทำให้หุ่นยนต์หกขาสามารถเคลื่อนที่ในพื้นผิวที่ขรุขระและพื้นผิวที่ต่างระดับได้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาค้นคว้าการทำงานของหุ่นยนต์หกขา
2. ทำการรวบรวมข้อมูล โดยการศึกษาวิเคราะห์การออกแบบและรูปแบบการทำงานของหุ่นยนต์หกขา
3. สามารถเลือกอุปกรณ์ต่างๆ รวมทั้งไมโครโปรเซสเซอร์มาใช้ในโครงงานได้
4. สามารถเข้าใจลักษณะการทำงานของ 32 Channel Servo Motor Control Driver Board
5. สามารถเขียนโปรแกรมควบคุมเซอร์โวมอเตอร์
6. ทำการประกอบ ทดสอบ และทำการทดลอง แล้วนำผลที่ได้จากการทดลองมาวิเคราะห์

1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถควบคุมทิศทางการเคลื่อนที่ของเซอร์โวมอเตอร์ได้
2. ทำให้หุ่นยนต์หกขาเคลื่อนที่ตามโปรแกรมที่สั่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษารูปแบบของหุ่นยนต์หกขา
2. ศึกษาการทำงานของเซอร์โวมอเตอร์ และ 32 Channel Servo Motor Control Driver Board
3. ศึกษาการทำงานของไมโครคอนโทรลเลอร์ Arduino UNO R3
4. ศึกษาการเขียนโปรแกรม SolidWorks เพื่อออกแบบตัวโครงและชิ้นส่วนขาของหุ่นยนต์หกขา
5. ศึกษาการเขียนโปรแกรม Altium Designer เพื่อออกแบบวงจรจ่ายไฟ
6. ศึกษาลักษณะการเดินของหุ่นยนต์หกขา
7. ศึกษาการเขียนโปรแกรม Arduino เพื่อควบคุมการเคลื่อนที่ของหุ่นยนต์หกขา

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานของเซอร์โวมอเตอร์และการเขียนโปรแกรมเพื่อควบคุมไมโครคอนโทรลเลอร์ Arduino UNO R3
2. ได้เรียนรู้การออกแบบตัวโครงและชิ้นส่วนขาของหุ่นยนต์หกขา โดยใช้โปรแกรม SolidWorks
3. ได้เรียนรู้การออกแบบวงจรจ่ายไฟ โดยใช้โปรแกรม Altium Designer
4. เข้าใจรูปแบบการเคลื่อนที่ของหุ่นยนต์หกขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 เซอร์โวมอเตอร์

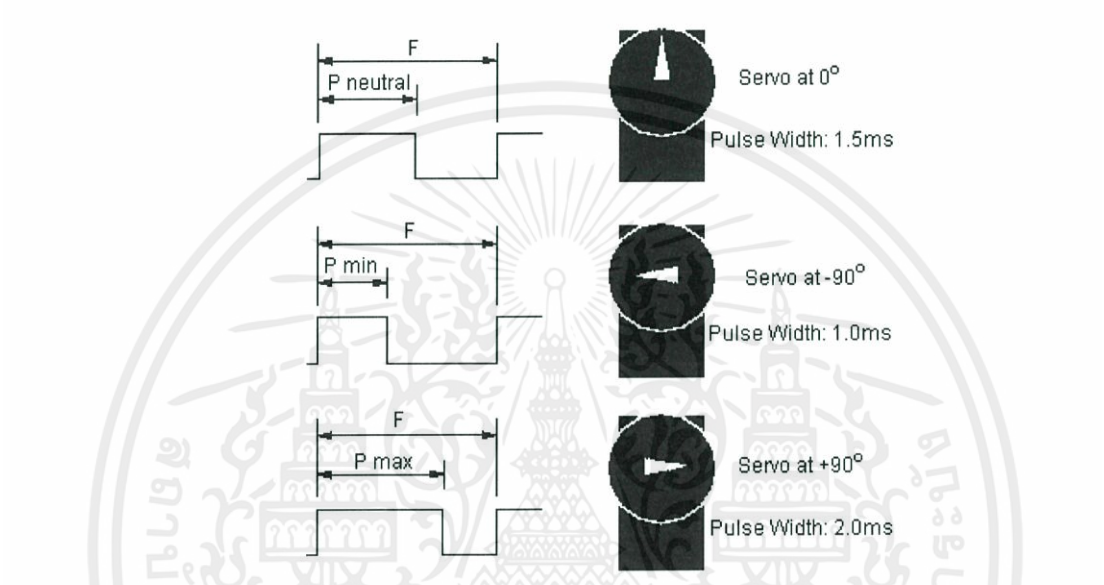
เซอร์โวมอเตอร์(Servo motor) คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์ และ ส่วนควบคุม ต่างๆ ไว้ในโมดูลเดียวกัน หรือ ภายในกล่องพลาสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC,GNDและ สายสัญญาณควบคุม(Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือ ขวาได้จากสายสัญญาณเพียงเส้นเดียวโดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณ พัลส์วิดมอด (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 โวลต์ ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือ จะมีขนาดเล็กน้ำหนักเบา,ให้แรงบิตสูง ,กินพลังงานน้อย และ สามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ(Driver) อื่นๆ เพราะ มอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่ง หรือ ทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้แค่เพียงในช่วงประมาณ 180° หรือ ครึ่งรอบเท่านั้น หรือ บางรุ่นอาจหมุนได้ถึง 210° แต่จะไม่สามารถหมุนเป็นวงรอบได้เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์ และ ตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้น เซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ประมาณ 180 องศา

รูปที่ 2.1 แสดงเซอร์โวมอเตอร์ (Servo Motor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 หลักการทำงานของเซอร์โวมอเตอร์

การควบคุมการทำงานของ เซอร์โวมอเตอร์ ทำได้โดย การป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆโดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูปที่ 2.2 คือ



รูปที่ 2.2 แสดงการทำงานของเซอร์โวมอเตอร์

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์
- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม -90 องศา หรือในทิศทางทวนเข็มนาฬิกา
- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม +90 องศา หรือในทิศทางตามเข็มนาฬิกา

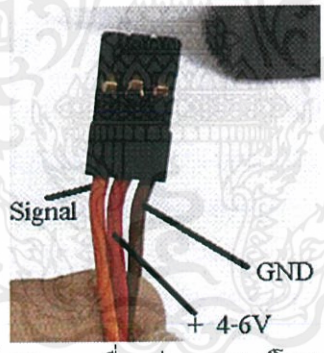
ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมอื่นๆ นั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆ โดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 ms เป็นต้น และสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยหลักการก็คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ที่จ่ายให้กับมอเตอร์ทางขาสัญญาณควบคุมกับค่าเวลาของวงจร RC ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากันมอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของ วงจร RC เปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control line) มอเตอร์จึงจะหยุดหมุน

2.1.2 สายเชื่อมต่อของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์จะมีสายเชื่อมต่อทั้งหมด 3 เส้น ได้แก่ เส้นสีส้ม จะเป็นสายสัญญาณที่ใช้ควบคุมตำแหน่งแกนของ Servo Motor, เส้นสีแดง จะเป็นสายไฟเลี้ยง โดยสามารถต่อแรงดันอยู่ระหว่าง 4-6V และเส้นสีดำ จะเป็นสายกราวด์



รูปที่ 2.3 สายเชื่อมต่อของเซอร์โวมอเตอร์

2.2 โครงสร้างและส่วนประกอบของ 32 Channel Servo Motor Control Driver Board for Arduino Robot

2.2.1 คุณสมบัติของบอร์ด

1) แหล่งจ่ายไฟ: 6.5 -12 V
2) อัตราการส่งข้อมูล: 9600,19200,28400,57600, 115200,128000
3) หน่วยความจำแฟลช: รอม 512K

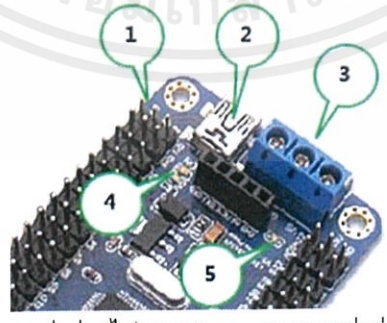
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) อินพุตการสื่อสาร: USB หรือ UART (TTL หรือ USART)

รูปที่ 2.4 แสดงบอร์ด 32 Channel Servo Motor Control Driver Board for Arduino Robot

2.2.2 คำแนะนำในการควบคุม Servo Motor

1) การควบคุม Servo ต้องการสองแหล่งจ่าย: แหล่งจ่ายไฟของ Servo และแหล่งจ่ายไฟของ Chip
2) แหล่งจ่ายไฟ Servo (+): VS (ซ้ายของกล่องช่องต่อสีน้ำเงิน ตามตำแหน่งในรูปที่ 2.5)
3) แหล่งจ่ายไฟ Servo (+): GND (กลางของกล่องช่องต่อสีน้ำเงิน ตามตำแหน่งในรูปที่ 2.5)
4) ค่าพารามิเตอร์แหล่งจ่ายไฟ Servo ขึ้นค่าที่กำหนดมาของ Servo เช่น ถ้า TR213 Servo มีแหล่งจ่าย 4.8-7.2 V แหล่งจ่ายไฟ Servo สามารถใช้แหล่งจ่าย 4.8-7.2 V ได้
5) แหล่งจ่ายไฟ Chip (+) : VSS (ขวาของกล่องช่องต่อสีน้ำเงิน ตามตำแหน่งในรูปที่ 2.5)
6) แหล่งจ่ายไฟ Chip (-) : GND (กลางของกล่องช่องต่อสีน้ำเงิน ตามตำแหน่งในรูปที่ 2.5)
7) ความต้องการ VS ในช่วง 6.5-12 V กำลังไฟของ Chip คือ การป้อนข้อมูลผ่านพอร์ต VSS ต้องการแหล่งจ่ายใน 6.5-12 V



รูปที่ 2.5 แสดงแหล่งจ่ายไฟของ Servo และแหล่งจ่ายไฟของ Chip

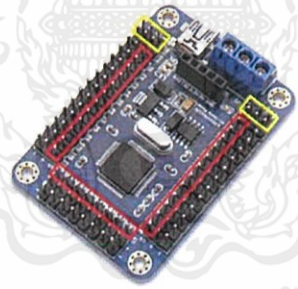
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

1. พอร์ต USB ตำแหน่งที่ 2 ตามรูป สามารถจ่ายไฟให้กับ Chip โดยสามารถเลือกพอร์ต USB หรือ พอร์ต VSS ก็ได้
2. ตำแหน่งที่ 1 ในรูป สามารถจ่ายไฟให้กับ Chip เช่นกัน คือ 5V และ GND ซึ่งที่5V คือ ขั้วบวกและ GND คือ ขั้วลบ Chip ต้องการแหล่งจ่าย 5V
3. ตำแหน่งที่ 1,2 และ 3 สามารถจ่ายไฟให้กับ Chip ได้เช่นกัน ขึ้นอยู่กับที่เราจะเลือกใช้
4. ไฟ LED สีเขียวที่ตำแหน่งที่ 4 ในรูป เป็นสถานะในการทำงานของ Chip ถ้าไฟสีเขียวติด Chip ทำงานอย่างถูกต้อง ถ้าไฟไม่ติดแสดงว่า Chip ทำงานผิดปกติ
5. ไฟ LED สีเขียวที่ตำแหน่งที่ 5 ในรูป เป็นสถานะในการทำงานของไฟ Servo ถ้าไฟสีเขียวติดแสดงว่าไฟที่จ่ายให้กับ Servo ถูกต้อง ถ้าไฟไม่ติดแสดงว่าเกิดความผิดพลาดของไฟที่จ่ายให้ Servo

2.2.3 การเชื่อมต่อ Servo ไปยังบอร์ด

1) ที่มีการทำกรอบสีแดง ในรูปที่ 2.6 คือส่วนของการเชื่อมต่อสัญญาณของ Servo
2) ที่มีการทำกรอบสีเหลือง ในรูปที่ 2.6 คือส่วนที่ไม่มีการเชื่อมต่อกับ Servo
3) ในการเชื่อมต่อ Servo จะต่อเข้ากับช่องตามสัญลักษณ์ เช่น S1,S2,S3 เป็นช่องของสัญญาณในการเชื่อมต่อ Servo



รูปที่ 2.6 แสดงการเชื่อมต่อ Servo กับบอร์ด

2.2.4 ซอฟแวร์ที่ใช้ควบคุมบอร์ด

1. การดาว์นโหลดซอฟแวร์ สามารถดาว์นโหลดซอฟแวร์ได้ที่ http://www.torobot.com/down/rios_usc.exe
2. การควบคุม Servo แบบทิศทางเดียว Run RIOS_USC.exe เลือกหมายเลขพอร์ตให้ถูกต้อง จากนั้นคลิกที่ปุ่ม “open”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การควบคุม Servo ตัวเดียว ใช้เมาส์ในการลากแผงควบคุม Servo ต้องเลือกช่องควบคุมให้ตรงกับตัว Servo ที่ได้ต่อไว้ เช่น ในภาพด้านล่าง Servo เชื่อมต่อกับช่อง S1

รูปที่ 2.7 แสดงการควบคุม Servo ตัวเดียว

4. การควบคุม Servo หลายตัวพร้อมกัน Servo หลายตัวจะถูกควบคุมพร้อมๆกัน โดยทำตามขั้นตอนดังรูป ซึ่งจะต้องมีการกำหนดเวลา เช่น ในตัวอย่างเรากำหนดเวลาในการหมุนของ Servo ไว้ที่ 1000ms แล้วคลิกที่ปุ่ม Add ที่ด้านล่าง จากนั้นโปรแกรมจะแปลงข้อมูลที่เราสั่งเพื่อควบคุม ถ้ามี Servo 10 ตัว ก็จะสามารถควบคุมพร้อมกันทั้ง 10 ตัว ได้เลย

5. การดาว์นโหลดกลุ่มคำสั่งในการควบคุม

- ถ้าเรามีคำสั่งหลายคำสั่ง ดังตัวอย่างภาพด้านล่าง เราจะทำการคลิกที่ปุ่มคำสั่ง run เพื่อทดสอบคำสั่งเหล่านั้น
- ถ้าผลการทดสอบของคำสั่งนั้นผ่าน คลิกที่ปุ่ม Download เพื่อดาว์นโหลดกลุ่มคำสั่งนั้น
- เมื่อเสร็จสิ้นการดาวน์โหลด ซอฟต์แวร์จะแจ้งว่า “ดาวน์โหลดเสร็จสมบูรณ์” หมายเลข = 1 หมายถึงเลขของกลุ่มคำสั่ง

รูปที่ 2.8 การควบคุม Servo หลายตัวพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. การสั่งทำงานพร้อมกันทั้งกลุ่ม อันดับแรกเราต้องคลิกที่ปุ่ม read เพื่อให้เลขประจำเครื่องทุกเครื่องได้รับข้อมูล หลังจากที่ทำการใส่ค่าตัวเลขเสร็จ เราจะทำการคลิกที่ปุ่ม run และเลือกสั่งว่าจะให้กลุ่มไหนทำงาน

7. การใช้งานแบบ Off-Line อันดับแรกเราจะคลิกที่ปุ่ม read เพื่อให้เลขประจำเครื่องทุกเครื่องได้รับข้อมูล หลังจากที่ใส่ค่าตัวเลขเสร็จ ให้คลิกที่ปุ่ม off-line และทำการเลือกว่าจะให้กลุ่มไหนทำงานแบบ off-line (กลุ่มที่มีการ off-line จะไม่ทำงาน จนกว่ามีการเปิดการทำงาน)

2.2.5 Secondary development

- Servo Control นั้นจะทำงานตามคำสั่ง โดยมันไม่สามารถคิดเองได้ทั้งหมด
- การติดต่อสื่อสารของ protocol จะมีหมายเลขการติดต่อสื่อสาร , baud rate 9600, 8 data bits , 1 stop bit

Name	Command	Description
Control one single servo	#1P1500T100\r\n	Data 1 refers to the servo's channel Data 1500 refers to the servo' location, in the range 500-2500 Data 100 refers to the time of execution and represents the speed, in the range 100-9999
Control multiple servos	#1P600#2P900#8P2500T100\r\n	Data 1, 2, and 8 refer to the servo's channels Data 600, 900, and 2500 refer to the locations of the servos that correspond to three channels Data 100 refers to the time of execution and represents the speed of three servos. Regardless of the number of servos, there is only one time, or one T. The command is executed at the same time; that is, all servos operate simultaneously.
Execute one single action group	#1GC2\r\n	Data 1 refers to the serial number of the action group Data 2 refers to the number of cycles
Execute multiple action groups	#1G#3G#1GC2\r\n	Execute the first, third and first action group,The number of cycles is 2. One particular group of action can appear repeatedly. There can be only one number of cycles or C. The command is executed in sequence; that is, the action groups are executed in sequence.

ตารางที่ 2.1 แสดงคำสั่งพื้นฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Arduino

Arduino เป็นภาษา อิตาลี ซึ่งใช้เป็นชื่อของโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR แบบ Open Source ที่ได้รับการปรับปรุงมาจากโครงการพัฒนา Open Source ของ AVR อีกโครงการหนึ่งที่ชื่อว่า “Wiring” แต่เนื่องจากโครงการของ “Wiring” เลือกใช้ AVR เบอร์ ATmega128 ซึ่งเป็นไมโครคอนโทรลเลอร์ ที่มีจำนวนของหน่วยความจำ และ I/O ค่อนข้างมาก และที่สำคัญ ATmega128 เป็นชิพที่มีตัวถังแบบ SMD จึงทำให้เป็นอุปสรรคสำหรับผู้เริ่มต้นในการสร้างบอร์ดและต่อวงจรขึ้นมาใช้งานกันเอง และบอร์ดจะมีขนาดค่อนข้างใหญ่ ซึ่งอาจดูว่าเกินความจำเป็นสำหรับผู้เริ่มต้น จึงไม่ค่อยได้รับความนิยมเท่าที่ควร แต่หลังจากที่ทางทีมงาน Arduino นำ Source Code ของ “Wiring” มาพัฒนาปรับปรุงใหม่ โดยให้สามารถใช้งานกับไมโครคอนโทรลเลอร์ AVR ขนาดเล็ก อย่าง Mega8 และ Mega168 ได้ จึงทำให้ระบบวงจรของบอร์ดมีขนาดเล็กลงกว่า “Wiring” มากและยังใช้อุปกรณ์น้อยชิ้น ทำให้ง่ายต่อการต่อวงจรใช้งานกันเอง และยังประหยัดต้นทุนในการสร้างบอร์ดไปได้มาก ด้วยเหตุนี้เองที่ทำให้ “Arduino” ได้รับความนิยมจากผู้ใช้งานทั่วโลกเป็นอย่างมาก ในระยะเวลาอันรวดเร็ว

เนื่องจาก Arduino เป็นภาษา อิตาลี ซึ่งมีสำเนียงการอ่านออกเสียงที่เป็นรูปแบบเฉพาะ และยังไม่มีการกำหนดเป็นคำภาษาไทยขึ้นมาอย่างเป็นทางการ ถึงแม้ว่า Arduino จะเป็นที่รู้จักของคนไทยมาระยะเวลาหนึ่งแล้วก็ตามที แต่ก็ยังไม่มีคำอ่านที่เป็นภาษาไทย อย่างเป็นทางการว่า คำๆนี้ ควรอ่านออกเสียงเป็นไทยว่าอย่างไร บางคนอ่านว่า อาร์-ดู-วี-โน่ บางคนอ่านว่า อา-เดีย-โน บางคนอ่าน เอ-อา-ดู-ไอ-โนและอื่นๆอีกมากมาก ซึ่งผู้เขียนเองก็ไม่มีความรู้เรื่องภาษา อิตาลี เลย จึงไม่ทราบเหมือนกันว่า จะเรียกชื่อของ Arduino เป็นภาษาไทยว่าอย่างไรจึงจะถูกต้อง ดังนั้นเพื่อไม่ให้เกิดความสับสน จึงขอใช้การทับศัพท์ตามชื่อเรียกที่เขียนเป็นภาษาอังกฤษเป็น Arduino ไปเลย

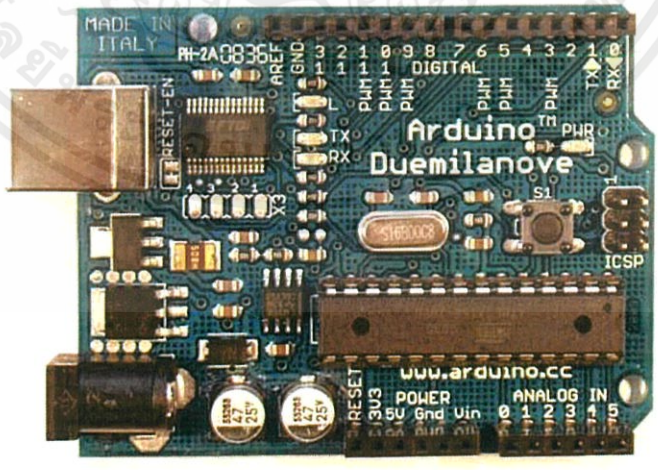
Arduino มีจุดเด่นในเรื่องของ ความง่ายในการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่างๆขึ้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ซึ่งถึงแม้ว่า Arduino เองจะมีรูปแบบการใช้งาน คล้ายๆกับกันไมโครคอนโทรเลอร์อย่าง Basic Stamp ของ Parallax, BX-24 ของ Netmedias และ Handy Board ของ MIT แต่ก็มีจุดเด่นกว่าของรายอื่นๆหลายอย่าง เป็นต้นว่า

- ราคาไม่แพง เนื่องจากมี Source Code และวงจร แจกให้ฟรี สามารถต่อวงจรขึ้นมาใช้งานได้เอง
- โปรแกรมที่ใช้พัฒนาของ Arduino รองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX
- มีรูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานจริงๆที่มีความซับซ้อนมากๆได้ และยังสามารถสร้างคำสั่งและ Library ใหม่ๆ ขึ้นมาใช้งานเองได้ เมื่อมีความชำนาญมากขึ้นแล้ว
- การเปิดเผยวงจรและ Source Code ทั้งหมดทำให้สามารถนำไปพัฒนาต่อยอดเพิ่มเติมได้ตามความต้องการทั้ง Hardware และ Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ โดยใช้ AVR ขนาดเล็กเป็นตัวประมวลผลและสั่งงาน เหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และ นำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input / Output ต่างๆ ได้มากมาย ทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่นๆ เช่น คอมพิวเตอร์ PC ทั้งนี้ก็เนื่องมาจากว่า Arduino สนับสนุนการเชื่อมต่อกับอุปกรณ์ Input / Output ต่างๆได้มากมาย ทั้งแบบ Digital และ Analog เช่น การรับค่าจากสวิตช์ หรืออุปกรณ์ตรวจจับ (Sensor) แบบต่างๆ รวมไปถึง การควบคุมอุปกรณ์ Output ต่างๆตั้งแต่ LED, หลอดไฟ,มอเตอร์, รีเลย์ ฯลฯ โดยระบบฮาร์ดแวร์ของ Arduino สามารถสร้างและประกอบขึ้นใช้งานได้เอง ในกรณีที่ผู้ใช้พอมีความรู้ด้านอิเล็กทรอนิกส์อยู่บ้าง หรือ สามารถซื้อแผงวงจรสำเร็จรูปที่มีการผลิตออกจำหน่ายกัน ในราคาที่ไม่แพง สำหรับเรื่องของโปรแกรมที่จะใช้เป็นเครื่องมือในการพัฒนานั้น สามารถ Download มาใช้งานกันได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆ โดย Arduino มีจุดเด่น ในเรื่องของความง่ายต่อการพัฒนาโปรแกรม และมีเอกสารข้อมูลรวมทั้งตัวอย่างต่างๆให้ใช้เป็นแนวทางในการศึกษาเรียนรู้เป็นจำนวนมาก

เนื่องจากArduino เป็นระบบการพัฒนาไมโครคอนโทรลเลอร์แบบ Open Source ซึ่งมีการตีพิมพ์เอกสารต่างๆที่เกี่ยวข้องออกมาเผยแพร่ให้ได้รับรู้เป็นระยะๆ รวมทั้งการเปิดเผย Source Code และตัวอย่างต่างๆ ให้ผู้ใช้นำไปใช้งาน หรือ พัฒนาดัดแปลงต่อยอดได้โดยไม่เสียค่าใช้จ่าย ด้วยเหตุนี้จึงมีผู้คนทั่วไปให้ความสนใจและนำไปศึกษาทดลองใช้งานกันมากมาย มีการนำไปดัดแปลงและสร้างเป็นโครงงาน แบบต่างๆ กันเป็นจำนวนมาก จึงเป็นประโยชน์อย่างยิ่งสำหรับผู้เริ่มต้นที่สามารถนำเอาตัวอย่างและโครงงานต่างๆที่คนอื่นทำไว้แล้ว มาใช้อ้างอิงเป็นแนวทางในการศึกษาเรียนรู้ได้โดยง่ายและที่สำคัญคือ ฟรี ไม่เสียค่าใช้จ่าย



รูปที่ 2.9 แสดงรูปบอร์ด Arduino ที่ถูกพัฒนาและเผยแพร่ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 บอร์ด Arduino รุ่น Uno R3

เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย คุณสมบัติของบอร์ดคือ

- ใช้ไมโครคอนโทรลเลอร์ ATmega328
- แรงดันใช้งาน 7-12 โวลต์
- หน่วยความจำแฟลชสำหรับโปรแกรมขนาด 32 กิโลไบต์
- Analog Input 6 ช่อง
- Digital Input 14 ช่อง

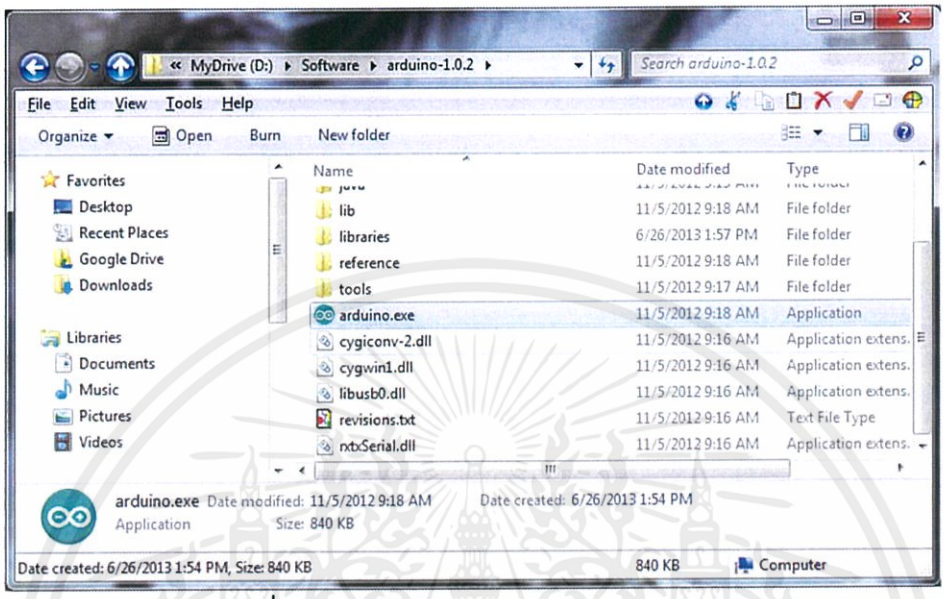
รูปที่ 2.10 แสดงรูปบอร์ด Arduino รุ่น Uno R3

2.3.2 การลงโปรแกรม Arduino

บอร์ด Arduino ที่มีอยู่หลายแบบนั้นจะใช้โปรแกรมตัวเดียวกันในการเขียนชุดคำสั่งครับ โดยโปรแกรมที่ว่านี้มีชื่อเป็นทางการว่า Arduino IDE (IDE นั้นย่อมาจาก Integrated Developemnt Environment) ซึ่งใช้งานได้ทั้งบนระบบปฏิบัติการ Window (XP Vista 7 8) ทั้ง 32 และ 64 บิต, Mac OS X และ Linux ก็ใช้ได้ เรียกได้ว่าใช้งานได้กับทุกระบบปฏิบัติการเลย แถมเป็นอิสระจากการทำงานของ OS ทุกชนิด ทำให้ไม่ต้องมีการ Install โปรแกรม ให้วุ่นวายแบบโปรแกรมอื่น เพราะแค่ Download มา จากนั้น Unzip ไว้ใน Directory ที่ต้องการก็เป็นอันเสร็จ ที่สำคัญ"ฟรี" แบบไม่มีโฆษณาและไม่ต้อง Crack ก่อนอื่นเรามาดูสิ่งที่จำเป็นต้องใช้ในการลงโปรแกรม และการทดสอบการทำงานของบอร์ดกับ Arduino IDE มีขั้นตอนดังนี้

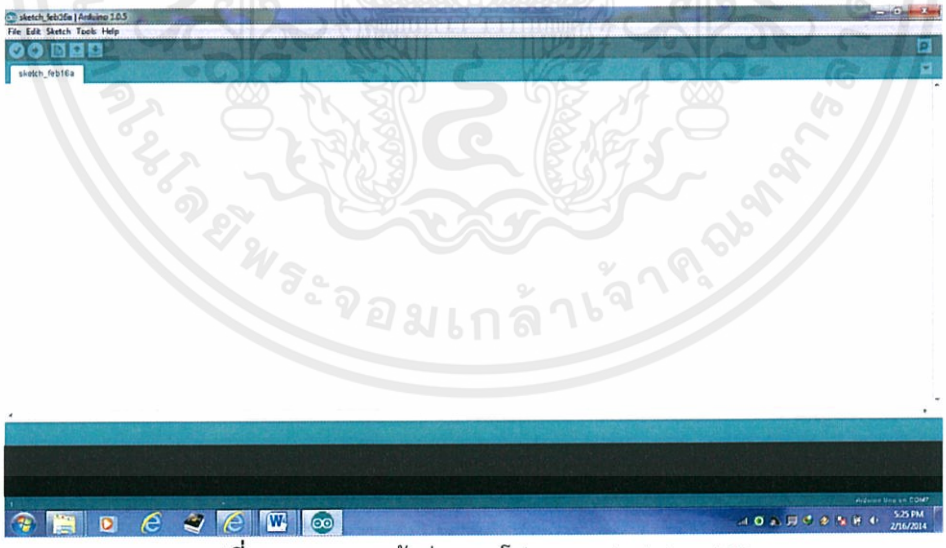
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. unzip เสร็จเรียบร้อย ก็ไปเริ่มโปรแกรมได้ที่ icon ตามรูปที่ 2.11



รูปที่ 2.11 แสดง icon โปรแกรม Arduino IDE

2. เข้าที่โปรแกรม Arduino IDE จะได้



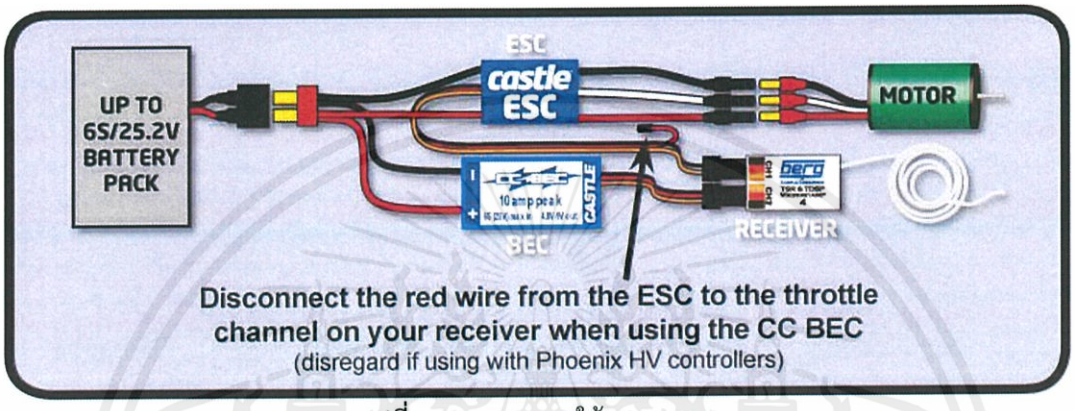
รูปที่ 2.12 แสดงหน้าต่างของโปรแกรม Arduino IDE

3. ก่อนใช้งานต้องตั้งค่า COM Port และเลือกรุ่นของบอร์ด Arduino ที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Universal Battery Elimination Circuit

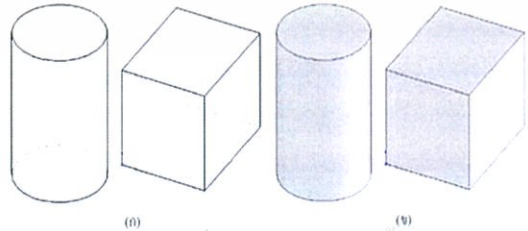
Universal Battery Elimination Circuit (UBEC) ทำหน้าที่จำกัดแรงดัน(V) และกระแสไฟ(A) จากแบตเตอรี่เพื่อที่จะจ่ายไฟให้กับ รีซีฟและเซอร์โวมอเตอร์ต่างๆ ซึ่งจะสามารถจ่ายกระแสได้สูงสุดถึง 10A เหมาะกับเซอร์โวมอเตอร์ตัวใหญ่ๆ ที่ต้องการไฟเลี้ยงมาก



รูปที่ 2.13 แสดงการใช้งาน UBEC

2.5 โปรแกรม SolidWorks

โปรแกรม SolidWorks เป็นโปรแกรมช่วยเขียนแบบที่เน้นการเขียนแบบใน 3 มิติ หรือ แบบ Parametric Solid Models คำว่า Solid Model หมายถึง แบบที่มีทรงตัน ซึ่งหมายถึง แบบใน 3 มิติที่มีเนื้อใน ซึ่งแตกต่างจากแบบ 3 มิติ ชนิด Wireframe ซึ่งสามารถใช้โปรแกรม CAD 2 มิติ บวกกับจิตนาการของคนเขียนแบบก็สามารถเขียนออกมาได้ Wireframe 3-D Model จึงเป็นแบบ 3 มิติที่ประกอบด้วยลายเส้น หรือ line มาต่อๆ กัน ส่วนคำว่า Parametric Solid Model หมายถึง แบบ 3 มิติที่ถูกสร้างขึ้นด้วยความสัมพันธ์ทางคณิตศาสตร์ ซึ่งเป็นการคำนวณภายในตัวโปรแกรม การเขียนแบบ 3 มิติลักษณะนี้จะมีความสะดวกกับผู้เขียนแบบมากกว่า การแสดง 3 มิติ ชนิด Wireframe Model และ Solid Model



รูปที่ 2.14 แบบจำลอง 3 มิติ (ก) แบบ Wireframe และ (ข) แบบ Solid Modeling

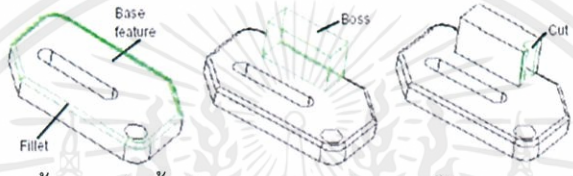
เอกสารนี้เป็นเอกสาร[illegible]ได้[illegible]ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 ขั้นตอนการสร้างชิ้นงาน 3 มิติและแบบสั่งงานในโปรแกรม SolidWorks

กระบวนการสร้างชิ้นงานใน SolidWorks ประกอบด้วย 3 ขั้นตอนหลัก ได้แก่

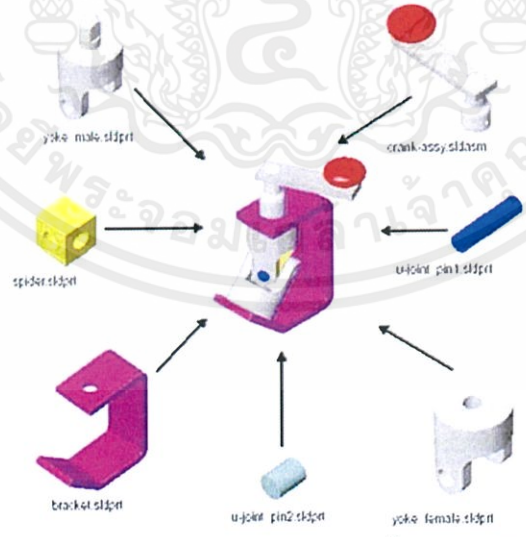
1. การสร้าง Sketch หรือลายเส้นใน 2 มิติ บนระนาบ (plane, flat surface) ใดๆ ลายเส้นที่สร้างขึ้นส่วนใหญ่จะเป็นรูปปิด

2. ใช้คำสั่งใดๆ ใน Features ทำให้ Sketch กลายเป็นชิ้นงาน 3 มิติ ในการสร้างชิ้นงาน 3 มิติที่ไม่ซับซ้อน Feature จะถูกสร้างซ้อนทับต่อๆ กันไป โดย feature แรกสุดโดยทั่วไปจะใช้ feature ที่มีขนาดใหญ่ เป็นหลัก เรียกว่า Base ส่วน feature ที่สร้างต่อๆ มาจะเรียกว่า Boss



รูปที่ 2.15 ชิ้นงานหรือชิ้นส่วนแบบ Part ประกอบด้วยหลายๆ Features

3. นำ Parts หลายๆ ชิ้นมาประกอบกันเป็น Assembly ขอเรียกเป็นภาษาไทยว่า “ชิ้นงานประกอบ” ในกรณีที่ชิ้นงานประกอบไม่ซับซ้อน ก็อาจเรียกว่า Subassembly เราสามารถนำ part และsubassembly หลายๆ ชิ้น มาประกอบเป็นชิ้นงานประกอบขั้นสุดท้าย เรียกว่า Full Assembly



รูปที่ 2.16 ชิ้นงานประกอบแบบ Assembly ประกอบด้วย Parts และ Sub-assembly

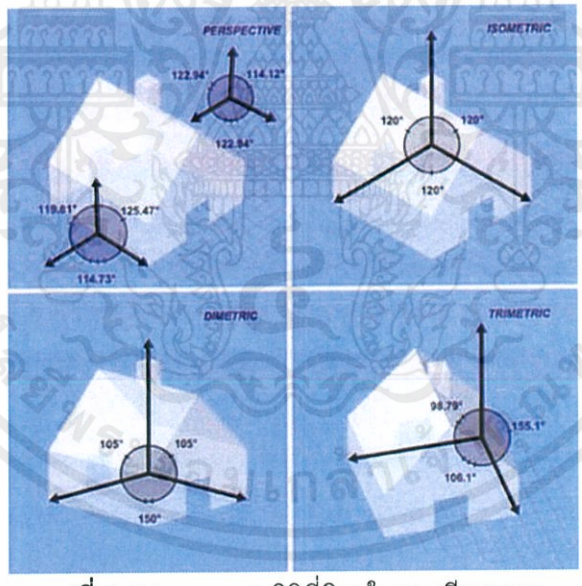
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 หลักการเขียนแบบพื้นฐานที่ควรรู้

แบบทางวิศวกรรม (Engineering Drawing) เป็นแบบจำลอง (Model) ที่สร้างขึ้นบนระนาบ (e.g.กระดาษเขียนแบบ, จอภาพ) เพื่อเป็นตัวแทนของชิ้นส่วนทางกล (Mechanical Part) แบ่งออกเป็น 2 ประเภท ใหญ่ๆ คือ (1) แบบ 3 มิติ และ (2) แบบ 2 มิติ ในบางครั้งจะใช้คำว่า"มุมมอง" หรือ View เพื่อขยายความว่าแบบในแต่ละประเภท เกิดจากการมองวัตถุในลักษณะใด เพื่อให้ได้แบบที่มี ความยาวใน 2 หรือ 3 มิติ สำหรับแบบ 2 มิติ จะเกิดจากการมองตั้งฉาก กับด้านใดด้านหนึ่งของวัตถุ เพื่อให้เห็นขนาดที่แท้จริง (true length) แบบ 2 มิติ เรานิยมเรียกว่า ภาพฉาย (Projection)

มุมมอง 3 มิติ (3-D View) ที่นิยมใช้ในการเขียนแบบเครื่องจักรกล คือ แบบ Axonometric หมายถึง การมองวัตถุให้เยื้องออกจากแกนหลักทั้ง 3 แกน (ไม่มองขนานกับแกนใดๆ เลย) ผลที่ได้ก็คือภาพวัตถุที่มีมิติในทั้ง 3 แกน ซึ่งจะแบ่งออกได้เป็น 3ประเภท คือ

1. Isometric View คือ ภาพที่มีสามแกนหลักทำมุมเท่ากันทั้งหมด ทุกมุมมีขนาด 120 องศา
2. Dimetric View คือ มีมุมระหว่างแกนหลักในภาพ เท่ากัน 2 มุม
3. Trimetric View คือ ทั้งสามมุมระหว่างแกนหลัก ไม่เท่ากันเลย

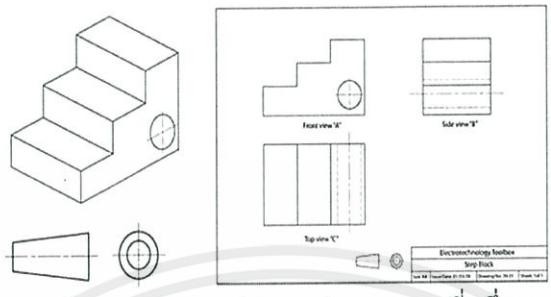


รูปที่ 2.17 มุมมอง 3 มิติที่นิยมในการเขียนแบบ

ภาพฉาย 2 มิติ (2-D Projection) เนื่องจากเป็นการมองตั้งฉากกับด้านใดด้านหนึ่งของวัตถุ จึงนิยมเรียกว่า ภาพฉายแบบ orthographic (Orthographic Projection) แบ่งออกเป็น 2 ประเภท คือ

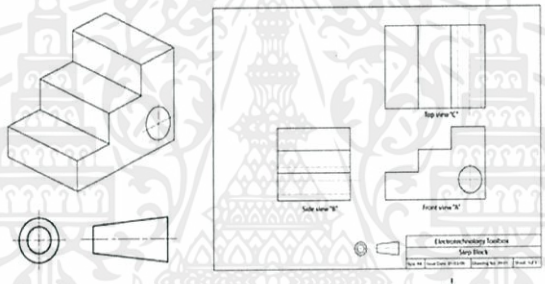
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.แบบมุมที่หนึ่ง (First Angle Projection)



รูปที่ 2.18 การเขียนภาพฉาย orthographic แบบมุมที่หนึ่ง และสัญลักษณ์

2.แบบมุมที่สาม (Third Angle Projection)



รูปที่ 2.19 การเขียนภาพฉาย orthographic แบบมุมที่สาม และสัญลักษณ์

หน่วยในการวัดความยาวของการเขียนแบบทางกล

1. หน่วยเมตทริค (Metric Unit) โดยความยาวจะนิยมใช้เป็นมิลลิเมตร ตัวย่อคือ ม.ม. (mm) ข้อสังเกตปกติความยาวในทางวิศวกรรมจะใช้เป็น เมตร (ม. หรือ m) การที่ใช้เป็นมิลลิเมตร เพราะในงานสร้างชิ้นงานทางกล ซึ่งมีขนาดค่อนข้างเล็ก เราต้องการความความละเอียดในการวัด เพื่อให้ได้ขนาดที่ถูกต้องแม่นยำ หน่วย metric นี้อยู่ในกลุ่มเดียวกับ หน่วย S.I. ซึ่งเป็นที่นิยมใช้แพร่หลายมากที่สุด

2. หน่วยแบบอังกฤษ (English Unit) หรือ หน่วย p แบบ Old English ความยาวจะเป็น นิ้ว (inch) จะใช้สัญลักษณ์ double quote (") ตามหลังตัวเลขความยาวเป็นนิ้ว เช่น 1" คือ ยาว 1 นิ้ว และ 3 ½ คือ ยาวสามนิ้วครึ่ง ประเทศที่ยังใช้ความยาวเป็นนิ้ว ที่สำคัญ คือ สหรัฐอเมริกา ความยาวใน หน่วยนิ้วยังสามารถแบ่งย่อยให้เล็กลงได้อีก หน่วยย่อยของนิ้วในภาษาไทย คือ "หุน" กำหนดให้ 1 นิ้ว = 8 หุน ฉะนั้น ½ นิ้ว จะเท่ากับ 4 หุน จะพบการใช้งานหน่วยนิ้วและหุน ในการบอกความยาว ของน็อตและสกรู และขนาดของดอกสว่าน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 โปรแกรม Altium Design

โปรแกรม Altium Design คือโปรแกรมทางด้านการออกแบบวงจรไฟฟ้าและสร้างลายวงจรลงบนแผ่นวงจรพิมพ์ หรือ PCB (Printed Circuit Board) ที่ไว้ใช้สาหรับลงอุปกรณ์และบัดกรีให้ได้วงจรที่สามารถใช้งานได้ โดยโปรแกรม Altium Design เป็นโปรแกรมที่ทางบริษัท Altium Limited ได้พัฒนาขึ้น โดยมีฟังก์ชันใช้งานที่มีความหลากหลาย ใช้ง่าย และมีความแม่นย่าในการออกแบบวงจรไฟฟ้าสูง

รูปที่ 2.20 แสดงโปรแกรม Altium Design

ขั้นตอนการออกแบบ PCB (โดยรวม) มีดังนี้

1. วาดลายวงจรในหน้า Schematic
2. อัพเดทใส่หน้า PCB
3. จัดอุปกรณ์และลากลายวงจร (ลากด้วยตัวเอง หรือ โปรแกรม)
4. ปริ้นท์ลายวงจร

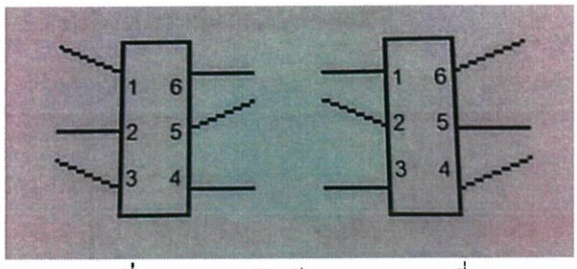
2.7 ลักษณะการเดินของหุ่นยนต์หกขา

ลักษณะการเดินของหุ่นยนต์หกขานั้นมีอยู่ด้วยกัน 3 ลักษณะใหญ่ๆด้วยกันคือ

2.7.1 การเดินทีละ 3 ขา (Tripod Gait)

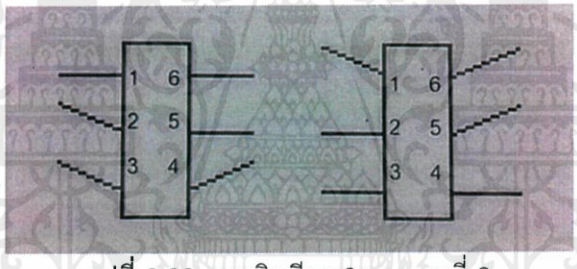
แบบที่ 1 จะแบ่งการทำงานเป็น 2 สเต็ป โดยขาด้านหนึ่งจะยก 2 ขาคือ ขาที่ 1 และขาที่ 3 ส่วนขาที่ 2 จะอยู่กับที่และขาอีกด้านหนึ่งจะยก 1 ขา คือ ขา 5 ส่วนขาที่ 4 และขาที่ 6 จะอยู่กับที่เป็น 1 สเต็ป ส่วนสเต็ปที่ 2 จะทำการยกขาที่ 2 ส่วนอีกด้านหนึ่งจะยกขาที่ 4 และขาที่ 6 เป็นการครบรอบการทำงาน 1 ครั้ง หลังจากนั้นก็กระทำแบบนี้สลับกันไปเรื่อยๆ ดังรูปที่ 2.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



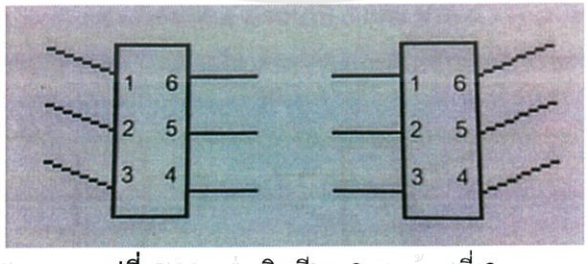
รูปที่ 2.21 การเดินทีละ 3 ขา แบบที่ 1

แบบที่ 2 จะแบ่งการทำงานเป็น 2 สเต็ป โดยขาด้านหนึ่งจะยก 2 ขา คือ ขาที่ 2 และขาที่ 3 ส่วนขาที่ 1 จะอยู่กับที่ และขาอีกด้านหนึ่งจะยก 1 ขา คือ ขา 4 ส่วนขาที่ 5 และขาที่ 6 จะอยู่กับที่เป็น 1 สเต็ป ส่วนสเต็ปที่ 2 จะทำการยกขาที่ 1 ส่วนขาที่ 2 และขาที่ 3 จะอยู่กับที่ อีกด้านหนึ่งจะยกขาที่ 5 และ ขาที่ 6 ส่วนขาที่ 4 จะอยู่กับที่ เป็นการครบรอบการทำงาน 1 ครั้ง หลังจากนั้นก็กระทำแบบนี้สลับกันไปเรื่อยๆ ดังรูปที่ 2.22



รูปที่ 2.22 การเดินทีละ 3 ขา แบบที่ 2

แบบที่ 3 จะแบ่งการทำงานเป็น 2 สเต็ป โดยขาด้านหนึ่งจะยก 3 ขา คือ ขาที่ 1 ขาที่ 2 และขาที่ 3 อีกด้านหนึ่งจะทำการอยู่กับที่ 3 ขา คือ ขาที่ 4 ขาที่ 5 และขาที่ 6 เป็น 1 สเต็ป ส่วนสเต็ปที่ 2 จะทำการยกขา 3 ขา คือ ขาที่ 4 ขาที่ 5 และ ขาที่ 6 ส่วนอีกด้านหนึ่ง จะทำการอยู่กับที่ 3 ขา คือขาที่ 1 ขาที่ 2 และขาที่ 3 เป็นการครบรอบการทำงาน 1 ครั้ง ดังรูปที่ 2.23

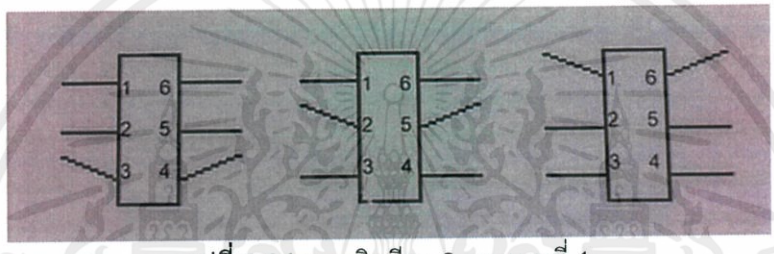


รูปที่ 2.23 การเดินทีละ 3 ขา แบบที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

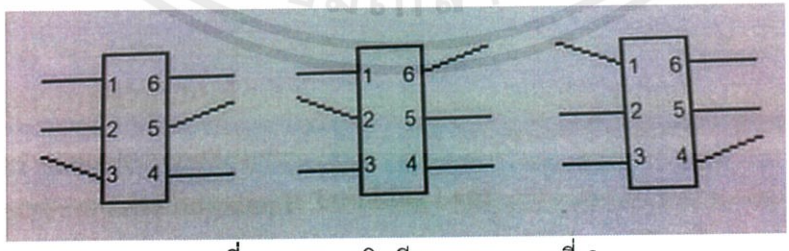
2.7.2 การเดินทีละ 2 ขา(Dual-leg Gait)

แบบที่ 1 จะแบ่งการทำงานเป็น 3 สเต็ป โดยขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 3 ส่วนขาที่ 1 และขาที่ 2 จะอยู่กับที่ และอีกด้านหนึ่งจะยกขา 1 ขา คือ ขาที่ 4 ส่วนขาที่ 5 และขาที่ 6 จะอยู่กับที่เป็น 1 สเต็ป ส่วนสเต็ปที่ 2 ทำได้โดยให้ขาด้านหนึ่งยก 1 ขา คือ ขาที่ 2 ส่วนขาที่ 1 และขาที่ 3 จะอยู่กับที่ และอีกด้านหนึ่งจะยกขา 1 ขา คือ ขาที่ 5 ส่วนขาที่ 4 และขาที่ 6 จะอยู่กับที่เป็นอีก 1 สเต็ป ส่วนสเต็ปที่ 3 ทำได้โดยให้ขาด้านหนึ่งยก 1 ขา คือ ขาที่ 1 ส่วนขาที่ 2 และขาที่ 3 จะอยู่กับที่ และ อีกด้านหนึ่งจะยกขา 1 ขา คือ ขาที่ 6 ส่วนขาที่ 4 และขาที่ 5 จะอยู่กับที่ เป็นอีก 1 สเต็ป เป็นการครบรอบการทำงาน 1 ครั้ง หลังจากนั้นก็กระทำแบบนี้สลับกันไปเรื่อยๆ ดังรูปที่ 2.24



รูปที่ 2.24 การเดินทีละ 2 ขา แบบที่ 1

แบบที่ 2 จะแบ่งการทำงานเป็น 3 สเต็ป โดยขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 3 ส่วนขาที่ 1 และขาที่ 2 จะอยู่กับที่ และอีกด้านหนึ่งจะยกขา 1 ขา คือ ขาที่ 5 ส่วนขาที่ 4 และ ขาที่ 6 จะอยู่กับที่ เป็น 1 สเต็ป ส่วนสเต็ปที่ 2 ทำได้โดยให้ขาด้านหนึ่งยก 1 ขา คือ ขาที่ 2 ส่วนขาที่ 1 และขาที่ 3 จะอยู่กับที่ และอีกด้านหนึ่งจะยกขา 1 ขา คือ ขาที่ 6 ส่วนขาที่ 4 และขาที่ 5 จะอยู่กับที่ เป็นอีก 1 สเต็ป ส่วนสเต็ปที่ 3 ทำได้โดยให้ขาด้านหนึ่งยก 1 ขา คือ ขาที่ 1 ส่วนขาที่ 2 และขาที่ 3 จะอยู่กับที่ และอีกด้านหนึ่งจะยกขา 1 ขา คือ ขาที่ 4 ส่วนขาที่ 5 และ ขาที่ 6 จะอยู่กับที่ เป็นอีก 1 สเต็ป เป็นการครบรอบการทำงาน 1 ครั้ง หลังจากนั้นก็กระทำแบบนี้สลับกันไปเรื่อยๆ ดังรูปที่ 2.25

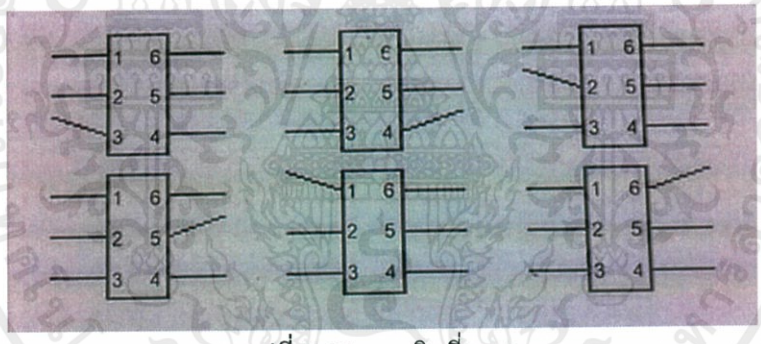


รูปที่ 2.25 การเดินทีละ 2 ขา แบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 การเดินทีละขา (Single-Leg Gait)

การเดินในลักษณะนี้จะแบ่งการทำงานเป็น 6 สเต็ป โดยขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 3 ส่วนขาที่ 1 และขาที่ 2 จะอยู่กับที่ ส่วนขาอีกด้านหนึ่ง คือ ขาที่ 4 ขาที่ 5 และ ขาที่ 6 จะอยู่กับที่ เป็น 1 สเต็ป ส่วนสเต็ปที่ 2 ทำได้โดยให้ขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 4 ส่วนขาที่ 5 และขาที่ 6 จะอยู่กับที่ ส่วนขาอีกด้านหนึ่ง คือ ขาที่ 1 ขาที่ 2 และ ขาที่ 6 จะอยู่กับที่ เป็นอีก 1 สเต็ป ส่วนสเต็ปที่ 3 ทำได้โดยให้ขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 2 ส่วนขาที่ 1 และขาที่ 3 จะอยู่กับที่ ส่วนขาด้านหนึ่ง คือ ขาที่ 4 ขาที่ 5 และขาที่ 6 จะอยู่กับที่ เป็นอีก 1 สเต็ป ส่วนสเต็ปที่ 4 ทำได้โดยให้ขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 5 ส่วนขาที่ 4 และขาที่ 6 จะอยู่กับที่ ส่วนขาอีกด้านหนึ่ง คือ ขาที่ 1 ขาที่ 2 และ ขาที่ 6 จะอยู่กับที่ เป็นอีก 1 สเต็ป ส่วนสเต็ปที่ 5 ทำได้โดยให้ขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 1 ส่วนขาที่ 2 และขาที่ 3 จะอยู่กับที่ ส่วนขาอีกด้านหนึ่ง คือ ขาที่ 4 ขาที่ 5 และขาที่ 6 จะอยู่กับที่ เป็นอีก 1 สเต็ป ส่วนสเต็ป ที่ 6 ทำได้โดยให้ขาด้านหนึ่งจะยก 1 ขา คือ ขาที่ 6 ส่วนขาที่ 4 และ ขาที่ 5 จะอยู่กับที่ ส่วนขาอีกด้านหนึ่ง คือ ขาที่ 1 ขาที่ 2 และ ขาที่ 6 จะอยู่กับที่ เป็นอีก 1 สเต็ปเป็นการครบรอบการทำงาน 1 ครั้ง หลังจากนั้นก็กระทำแบบนี้สลับกันไปเรื่อยๆ ดังรูปที่ 2.26



รูปที่ 2.26 การเดินทีละขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

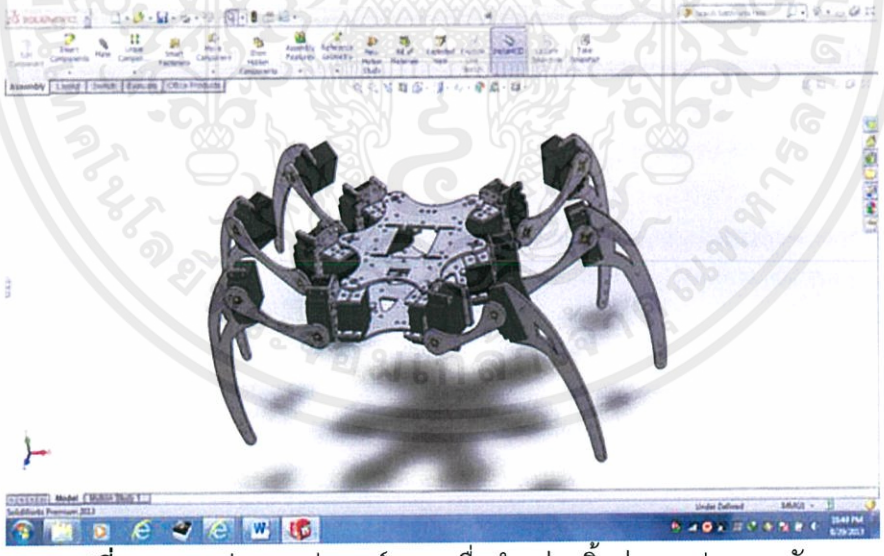
บทที่ 3
การออกแบบและการสร้าง

3.1 กล่าวนำ

ในส่วนของการดำเนินงานทางคณะผู้จัดทำได้แบ่งออกเป็น 4 ส่วนหลัก คือ การออกแบบโครงสร้างโดยใช้โปรแกรม Solidworks, การเลือกวัสดุและตัดชิ้นส่วนที่ได้ทำการออกแบบ, ทำการประกอบชิ้นส่วนต่างๆเข้าด้วยกัน และการติดต่อสื่อสารและการส่งข้อมูลระหว่างอุปกรณ์

3.2 การออกแบบโครงสร้าง

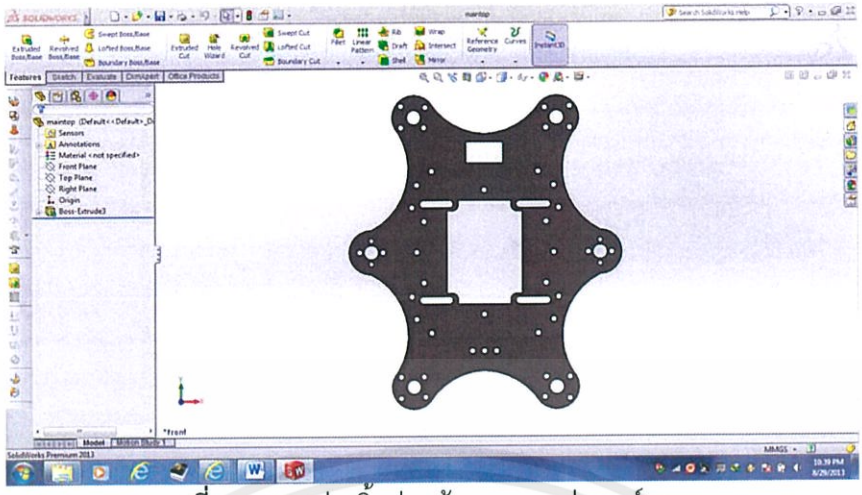
การออกแบบโครงสร้างหุ่นยนต์หกขานี้ใช้โปรแกรม Solidworks ในการออกแบบ โดยทำการร่างแบบของอุปกรณ์ต่างๆ ที่เป็นส่วนประกอบของหุ่นยนต์หกขา ซึ่งประกอบด้วยโครงชิ้นส่วนด้านบนของหุ่นยนต์หกขาจำนวน 1 ชิ้น, โครงชิ้นส่วนด้านล่างของหุ่นยนต์หกขา จำนวน 1 ชิ้น, ชิ้นส่วนข้อต่อที่ยึดกับ Servo Motor จำนวน 12 ชิ้น, ชิ้นส่วนที่ยึดกับเฟืองของ Servo Motor จำนวน 6 ชิ้น,ชิ้นส่วนขา จำนวน 6 ชิ้น และ Servo Motor จำนวน 18 ชิ้น เป็นส่วนประกอบหลัก จากนั้นนำชิ้นส่วนต่างๆมาประกอบเข้าด้วยกันก่อนทำการสร้างจริง



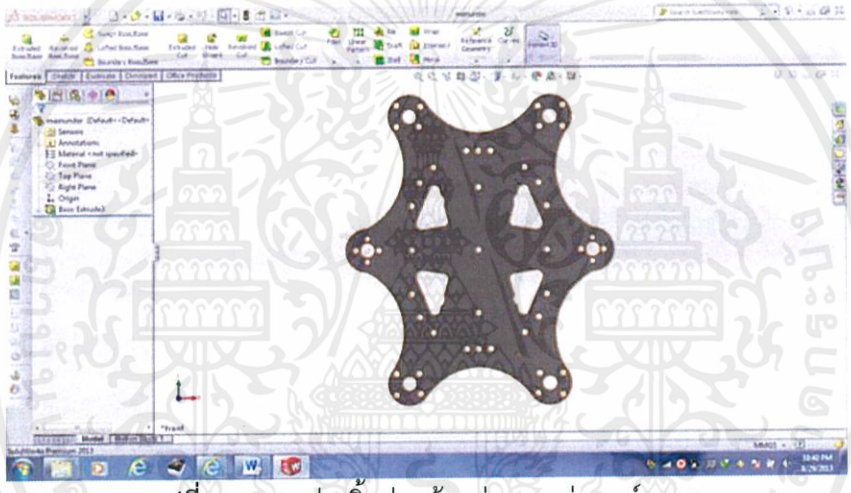
รูปที่ 3.1 แบบร่างของหุ่นยนต์หกขาเมื่อนำแต่ละชิ้นส่วนมาประกอบกัน

ในการออกแบบโครงสร้างของหุ่นยนต์หกขา จะใช้ความหนาของชิ้นวัสดุที่ 3 มิลลิเมตร ในการออกแบบทุกชิ้นส่วน ต่อไปเป็นรูปแสดงแบบร่างของชิ้นส่วนต่างๆ ที่ใช้โปรแกรม Solidworks

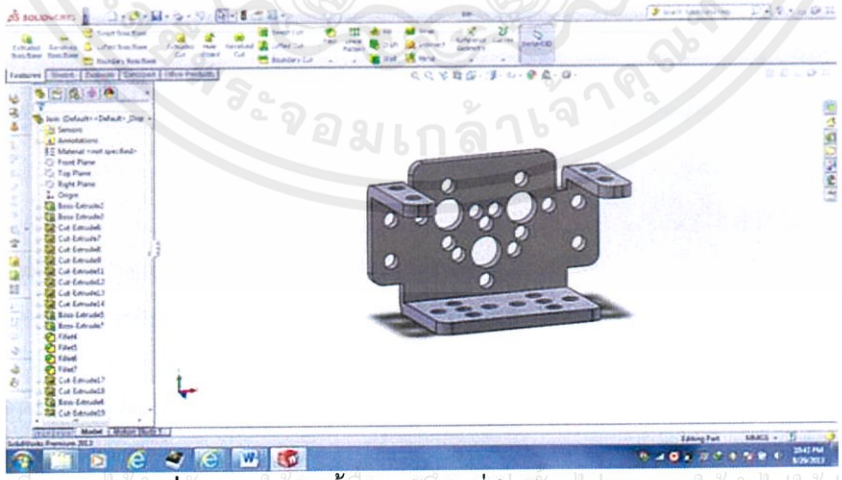
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แบบร่างชิ้นส่วนด้านบนของหุ่นยนต์หกขา

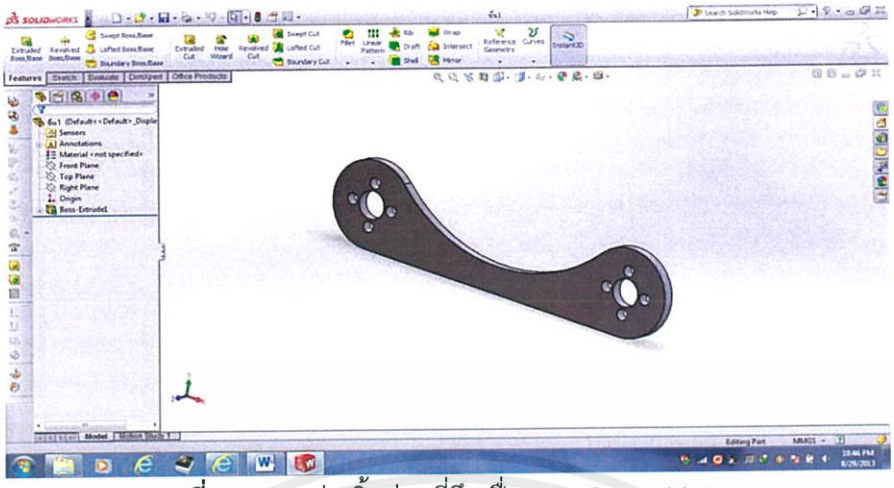


รูปที่ 3.3 แบบร่างชิ้นส่วนด้านล่างของหุ่นยนต์หกขา

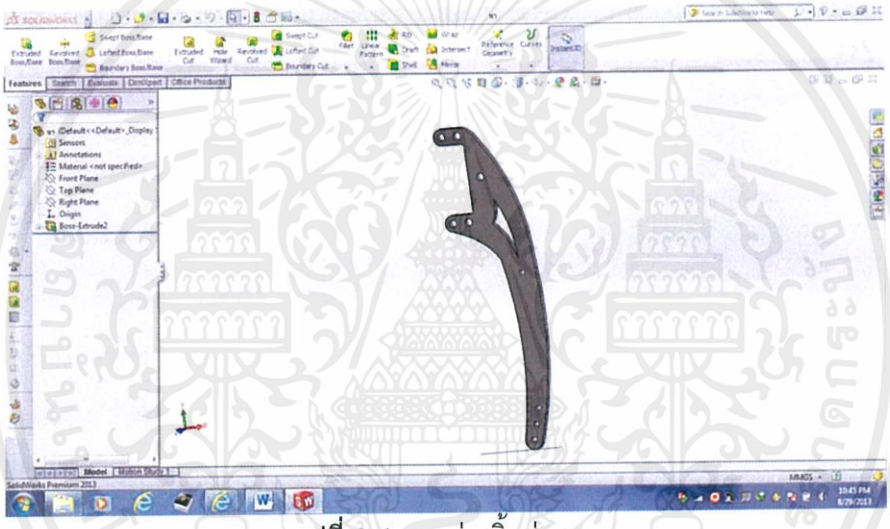


รูปที่ 3.4 แบบร่างชิ้นส่วนข้อต่อที่ยึดกับ Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แบบร่างชิ้นส่วนที่ยึดเฟืองของ Servo Motor



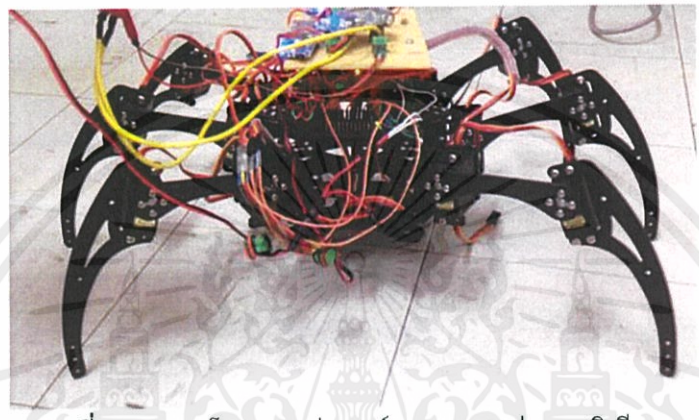
รูปที่ 3.6 แบบร่างชิ้นส่วนขา

รูปที่ 3.7 แบบร่าง Servo Motor จำนวน 18 ชิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเลือกวัสดุและตัดชิ้นส่วนที่ได้ทำการออกแบบ

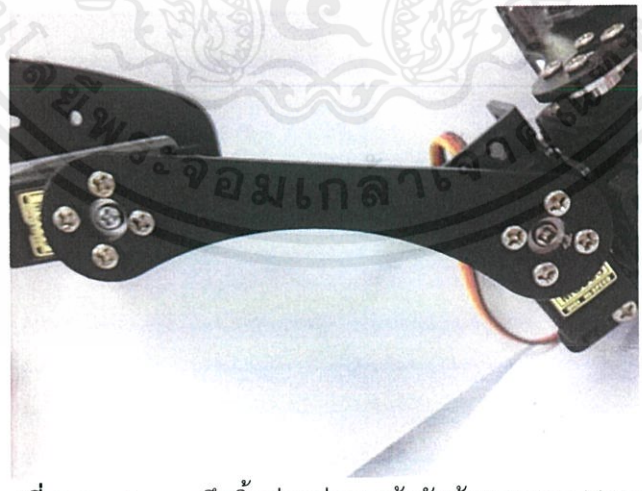
การเลือกใช้วัสดุที่เป็นโครงของหุ่นยนต์หกขา จะเลือกใช้แผ่นอะลูมิเนียมหนา 3 มิลลิเมตร ในการตัดชิ้นส่วนต่างๆ ที่ได้ทำการออกแบบไว้ เนื่องจากแผ่นอะลูมิเนียมมีน้ำหนักเบาและมีความแข็งแรง เมื่อเทียบกับวัสดุอื่นที่หนา 3 มิลลิเมตร ส่วนในเรื่องการตัดชิ้นส่วนที่ได้ทำการออกแบบจะส่งแบบร่างที่ได้จากโปรแกรม Solidworks ไปตัดโครงของหุ่นยนต์หกขาจากแผ่นอะลูมิเนียม ด้วยเครื่อง CNC (Computerize Numerical Control)



รูปที่ 3.8 แสดงโครงของหุ่นยนต์หกขาจากแผ่นอะลูมิเนียม

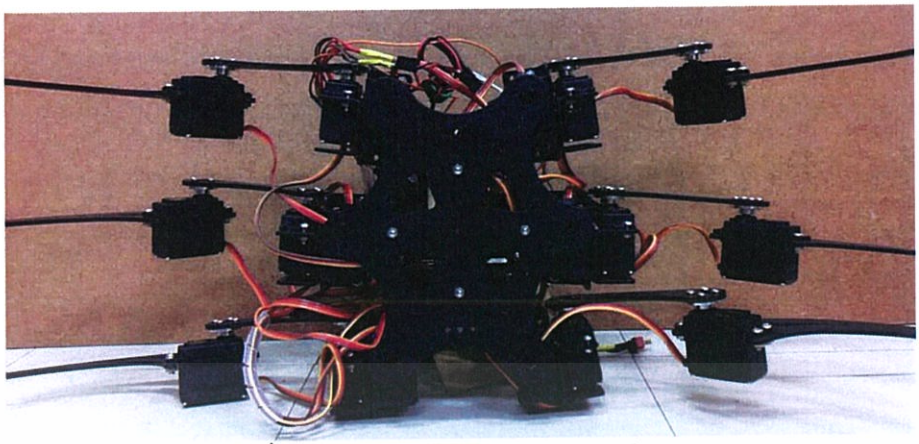
3.4 การประกอบชิ้นส่วน

ขั้นตอนแรกเป็นการประกอบชิ้นส่วนอะลูมิเนียม ที่ผ่านการตัดด้วย CNC เข้าด้วยกัน โดยแต่ละข้อต่อของชิ้นส่วนจะยึดเข้ากันด้วย Servo Motor



รูปที่ 3.9 แสดงการยึดชิ้นส่วนต่างๆ เข้ากันด้วย Servo Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



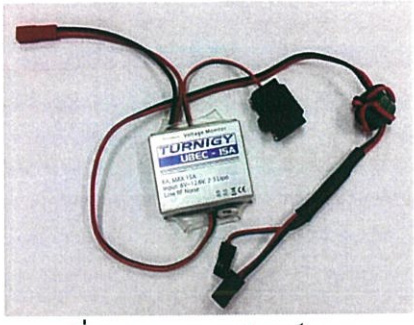
รูปที่ 3.10 แสดงโครงสร้างของหุ่นยนต์หกขา

ขั้นตอนถัดมา เป็นการติดตั้งบอร์ด Arduino UNO R3 ,บอร์ด Torobot 32 channel serial servo controller และ Servo Motor เข้าด้วยกัน

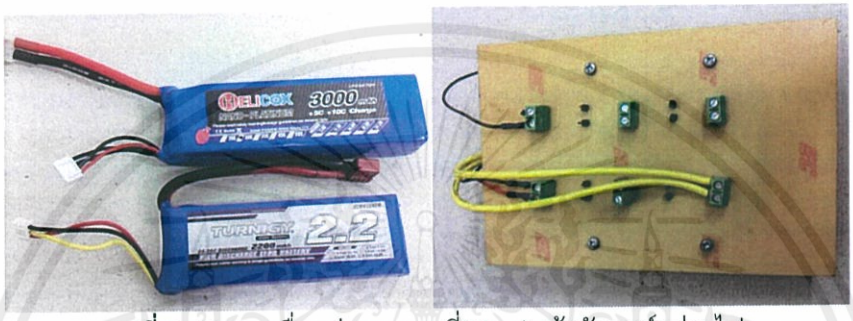
รูปที่ 3.11 แสดงการติดตั้งบอร์ด Arduino UNO R3

ทำการเชื่อมต่อแบตเตอรี่(Lipo) เข้ากับบอร์ดจ่ายไฟ จากนั้นเชื่อมต่อไปยังอุปกรณ์ UBEC (Universal Battery Elimination Circuit) โดย UBEC ทำหน้าที่จำกัดแรงดัน(V) และกระแสไฟ(A) จากแบตเตอรี่เพื่อที่จะจ่ายให้กับ Servo Motor ซึ่งโครงสร้างของหุ่นยนต์หกขาประกอบด้วย Servo Motor จำนวน 18 ตัว จึงต้องการไฟเลี้ยงมาก ดังนั้นเมื่อผ่านอุปกรณ์ UBEC จะสามารถจ่ายกระแสได้สูงเพียงพอกับที่ Servo Motor ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

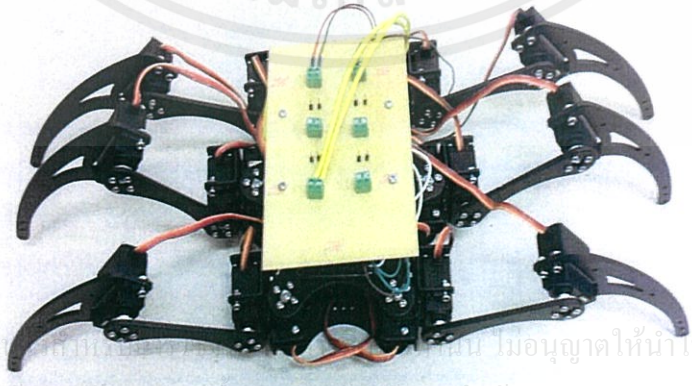


รูปที่ 3.12 แสดงอุปกรณ์ UBEC



รูปที่ 3.13 การเชื่อมต่อแบตเตอรี่(Lipo) เข้ากับบอร์ดจ่ายไฟ

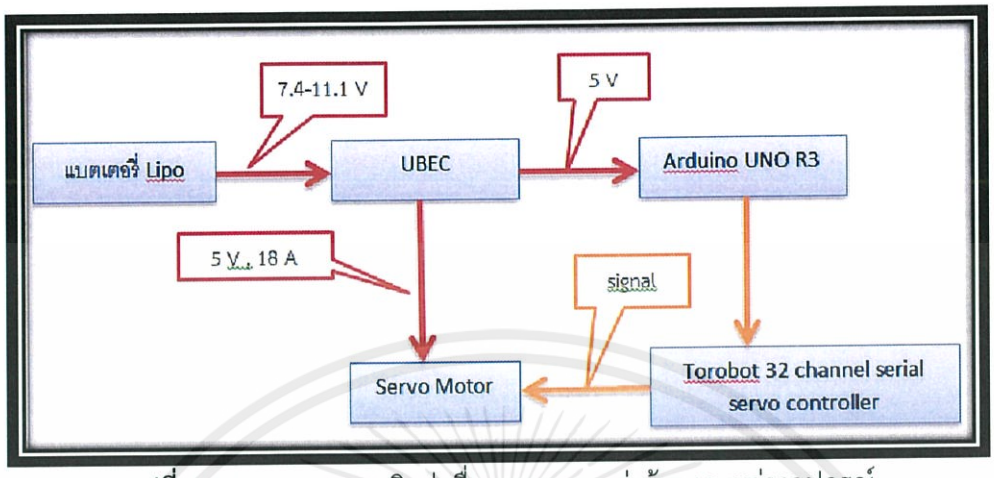
เมื่อประกอบชิ้นส่วนและอุปกรณ์ต่างๆ เข้าด้วยกันเรียบร้อยแล้ว จะได้หุ่นยนต์หกขาที่สมบูรณ์ โดยหุ่นยนต์จะมีหกขา ใช้ Servo Motor จำนวน 18 ตัว มีการเชื่อมต่อแบตเตอรี่(Lipo) ไปยังบอร์ดจ่ายไฟ จากนั้นเชื่อมต่อไปยังอุปกรณ์ UBEC เพื่อแปลงกระแส แล้วจึงต่อเข้ากับ Servo Motor จำนวน 18 ตัว สามารถควบคุมการเคลื่อนที่ของหุ่นยนต์ ด้วยการโปรแกรมคำสั่งเข้าที่บอร์ด Arduino UNO R3 จากนั้นค่าที่สั่งจะถูกส่งต่อไปยัง บอร์ด Torobot 32 channel serial servo controller ซึ่งบอร์ดนี้จะเป็นตัวควบคุมสัญญาณให้ Servo Motor หมุนไปอยู่ที่ตำแหน่งมุมที่ได้โปรแกรมไว้



รูปที่ 3.14 หุ่นยนต์หกขาที่ประกอบเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

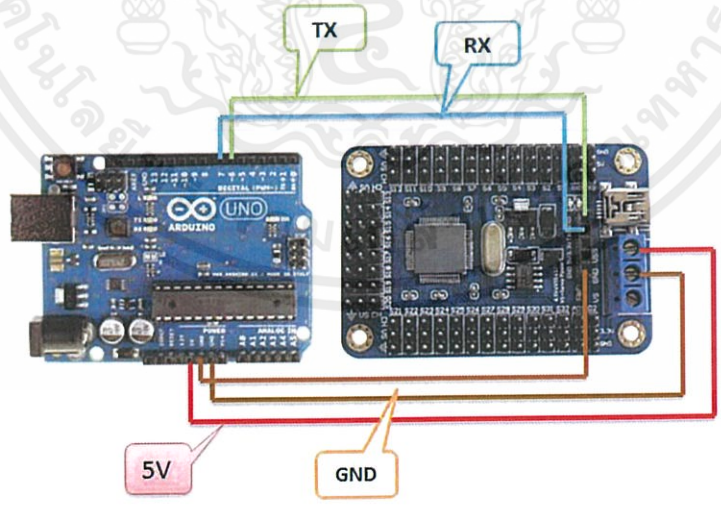
3.5 การติดต่อสื่อสารและการส่งข้อมูลระหว่างอุปกรณ์



รูปที่ 3.15 แผนภาพการติดต่อสื่อสารและการส่งข้อมูลระหว่างอุปกรณ์

3.5.1 การติดต่อสื่อสารระหว่างบอร์ด Arduino UNO R3 กับบอร์ด Torobot 32 channel serial servo controller

การติดต่อสื่อสารระหว่างบอร์ดจะมีการส่งข้อมูลผ่านช่อง TX กับ RX ซึ่งได้เขียนโปรแกรมใน Arduino ให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX จากนั้นเชื่อมต่อสัญญาณ TX กับ RX จากบอร์ด Arduino มายังบอร์ด Torobot และมีการเชื่อมต่อสายไฟ 5 V กับ GND ระหว่างบอร์ดทั้งสอง



รูปที่ 3.16 แผนภาพการเชื่อมต่อระหว่างบอร์ด Arduino กับบอร์ด Torobot

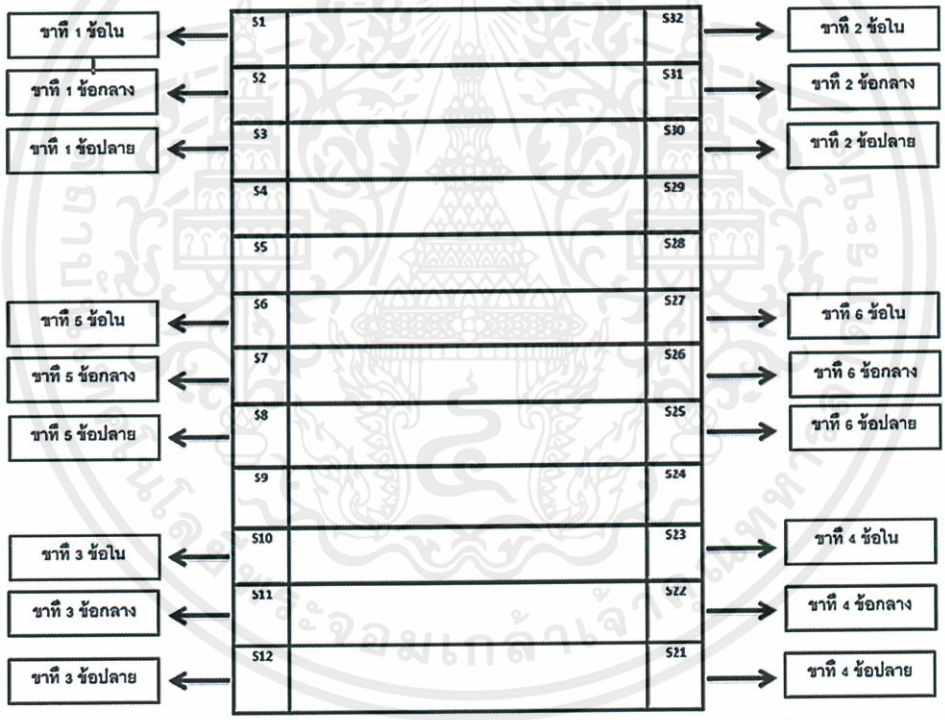
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรมในการควบคุมนั้น จะใช้โปรแกรม Arduino IDE แต่เนื่องจากเป็นการติดต่อสื่อสารและส่งข้อมูลระหว่างบอร์ด Arduino UNO R3 กับบอร์ด Torobot 32 channel serial servo controller จึงจำเป็นจะต้องมีชุดคำสั่ง ที่ทั้งสองบอร์ดมีความเข้าใจตรงกัน ดังนี้

<u>ชุดคำสั่ง</u>

Control one single servo :	#1P1500T100\r\n
Control multiple servos :	#1P600#2P900#8P2500T100\r\n
Execute one single action group :	#1GC2\r\n
Execute multiple action groups :	#1G#3G#1GC2\r\n

3.5.2 การเชื่อมต่อสายสัญญาณระหว่าง Servo Motor 18 ตัว เข้ากับบอร์ด Torobot 32 channel serial servo controller



รูป 3.17 แผนภาพการเชื่อมต่อสัญญาณระหว่าง Servo Motor กับบอร์ด Torobot

จากรูปที่ 3.17 เป็นการเชื่อมต่อสัญญาณจากบอร์ด Torobot 32 channel serial servo controller กับ Servo Motor 18 ตัว เพื่อใช้ในการควบคุมตำแหน่งมุมของ Servo Motor แต่ละตัว ซึ่งได้เลือกช่องของการเชื่อมต่อ(channel) โดยแบ่งเป็นชิ้นส่วนแต่ละข้อของขา ประกอบด้วยข้อใน, ข้อกลาง และข้อปลาย จะให้ขาเดียวกันใช้ช่องของการเชื่อมต่อสัญญาณ ของชิ้นส่วนแต่ละข้ออยู่ติดกัน ทำให้ง่ายต่อการจำและรู้ว่าเป็นขาใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 การปรับมุมของ Servo Motor

เนื่องจาก Servo Motor บางตัวค่าของมุมผิดไป จึงต้องหาค่ามุมที่ถูกต้อง ซึ่งปกติมุมของ Servo Motor จะอยู่ในช่วง 0 ถึง 180 องศา แต่ถ้าจะโปรแกรมต้องป้อนค่าในช่วง 500 ถึง 2500 เมื่อนำค่าทั้งสองช่วงมาเทียบกันจะได้สูตร คือ

$$\text{ค่ามุม (องศา)} = \left(\frac{180(A-500)}{2000}\right) \text{ โดย A เป็นค่าในช่วง 500 ถึง 2500}$$

legs	Part 1	Part 2	Part 3
1	adjust 1500 = angle 90 degree	adjust 1200= angle 90 degree	adjust 1500= angle 90 degree
2	adjust 1500= angle 90 degree	adjust 1400= angle 90 degree	adjust 1500 = angle 90 degree
3	adjust 1500= angle 90 degree	adjust 1650= angle 90 degree	adjust 1500= angle 90 degree
4	adjust 1450= angle 90 degree	adjust 1500= angle 90 degree	adjust 1500 = angle 90 degree
5	adjust 1500= angle 90 degree	adjust 1500= angle 90 degree	adjust 1500= angle 90 degree
6	adjust 1450= angle 90 degree	adjust 1500= angle 90 degree	adjust 1500= angle 90 degree

ตารางที่ 3.1 แสดงการหาค่ามุมที่จะใช้โปรแกรม เทียบจากมุม 90 องศา

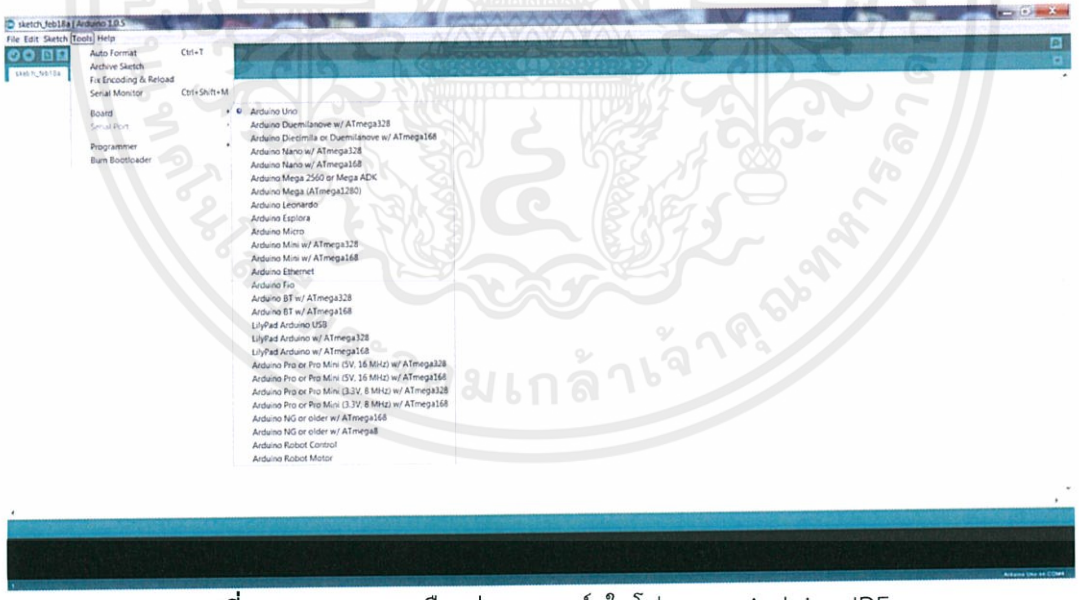
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การโปรแกรมข้อมูลบนบอร์ด Arduino UNO R3

1. จะควบคุมการเคลื่อนที่ของหุ่นยนต์หกขาด้วยบอร์ด Arduino UNO R3 โดยทำการเชื่อมต่อกับ Computer ผ่านสาย USB Port

รูปที่ 3.18 แสดงสาย USB Port เชื่อมต่อ Computer กับบอร์ด Arduino UNO R3

2. เปิดโปรแกรม Arduino IDE ซึ่งโปรแกรมนี้จะใช้กับบอร์ด Arduino หลายรุ่น ดังนั้นจะต้องกำหนดบอร์ดที่ใช้ในโปรแกรมนี้ โดยเข้าที่ Tool ตามด้วย Board แล้วเลือก Arduino UNO



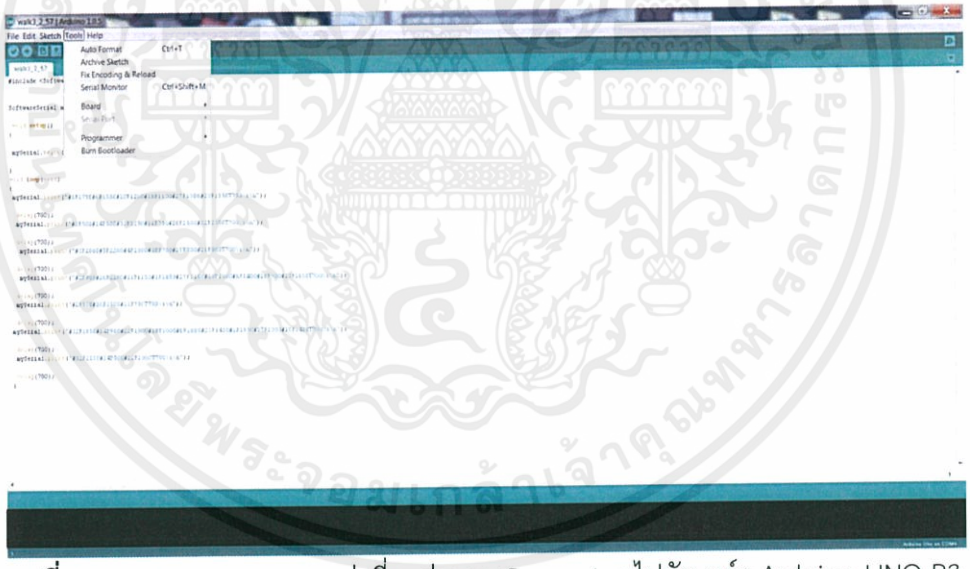
รูปที่ 3.19 แสดงการเลือกรุ่นของบอร์ดในโปรแกรม Arduino IDE

3. การตั้งค่าเริ่มต้นของโปรแกรม เป็นการติดต่อสื่อสารระหว่างบอร์ด Arduino UNO R3 กับบอร์ด Torobot 32 channel serial servo controller ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);        //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
 void setup()
{
 mySerial.begin(9600);               //อัตราการส่งข้อมูล
}
void loop(void)                      //เขียนคำสั่งควบคุมการเคลื่อนที่
{
 }
```

4. เมื่อเขียนคำสั่งควบคุมการเคลื่อนที่แล้ว ให้กด Verify เพื่อเป็นการตรวจสอบพื้นฐาน จากนั้นให้กด Upload ค่าที่ได้โปรแกรมก็จะถูกบันทึกลงบอร์ด Arduino UNO R3 และถ้าต้องการตรวจสอบค่าที่ถูกส่งจาก Computer ไปยังบอร์ด เข้าที่ Tool แล้วเลือก Serial Monitor



รูปที่ 3.20 แสดงการตรวจสอบค่าที่ถูกส่งจาก Computer ไปยังบอร์ด Arduino UNO R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลอง

4.1 กล่าวนำ

การทดลองควบคุมหุ่นยนต์หกขาในลักษณะต่างๆ ประกอบด้วยรูปแบบการยืน, รูปแบบการเดิน, รูปแบบการเลี้ยวซ้าย และรูปแบบการเลี้ยวขวา โดยจะมีการเปรียบเทียบมุมของ Servo Motor จากตำแหน่งเริ่มต้นไปยังตำแหน่งที่เคลื่อนที่ในรูปแบบต่างๆ เพื่อหาค่ามุมของ Servo Motor แต่ละตัว ว่าค่ามุมเท่าไรที่ทำให้การเคลื่อนที่ในรูปแบบต่างๆ นั้นเคลื่อนไหวได้ดีที่สุด

4.2 ขั้นตอนการทดลอง

1.เปิดโปรแกรม Arduino IDE เขียนโปรแกรมคำสั่งลงในวงเล็บของ void loop (void)

<u>รูปแบบการตั้งค่าก่อนโปรแกรม</u>

```
#include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);        //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
 void setup()
{
 mySerial.begin(9600);               //อัตราการส่งข้อมูล
}
void loop(void)                      //เขียนคำสั่งควบคุมการเคลื่อนที่
{
}
```

2. ใช้ชุดคำสั่ง Control multiple servos : #1P600#2P900#8P2500T100\r\n โดยที่

ตามด้วย channel ที่เสียบสายสัญญาณของ Servo Motor

P ตามด้วย ค่าของมุมที่โปรแกรมในช่วง 500 ถึง 2500 ซึ่งเกิดจากการเทียบค่ามุมของ servo motor ที่ช่วง 0 องศา ถึง 180 องศา

T ตามด้วย ช่วงของเวลา (ms)

3. ป้อนค่าต่างๆ ตามข้อที่ 2 ของ Servo Motor จำนวน 18 ตัว เพื่อหาค่ามุมที่เหมาะสมในการเคลื่อนที่ของหุ่นยนต์หกขา ในรูปแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

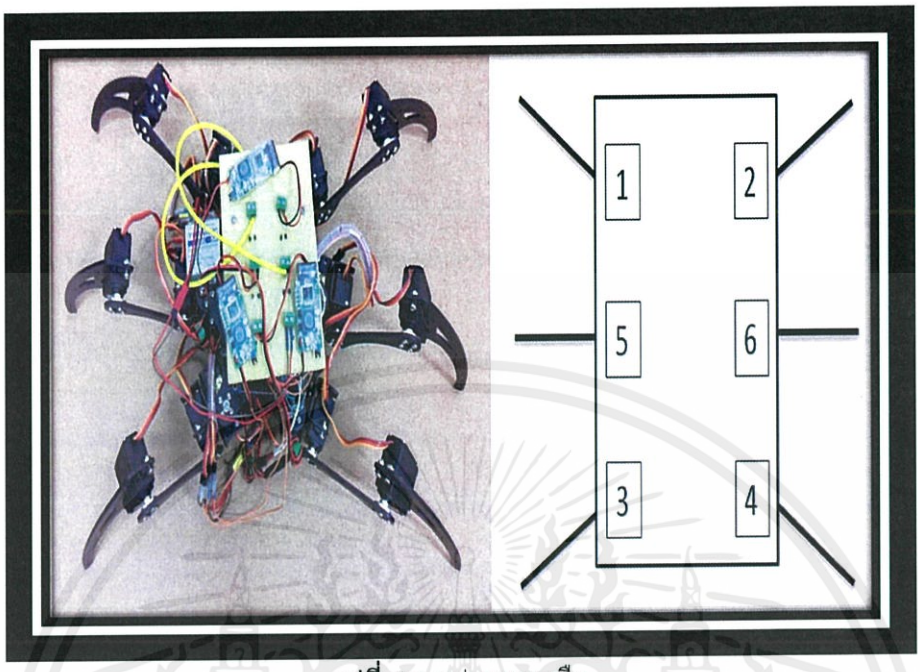
4.3 ผลการทดลอง

4.3.1 การทดลองหาค่ามุมของรูปแบบการยืน

ตารางที่ 4.1 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการยืน

ขาที่	ช่องสัญญาณที่	มุมของ Servo Motor (องศา)	ค่ามุมที่โปรแกรม
ขาที่ 1 ข้อใน	s1	112.5	1750
ขาที่ 1 ข้อกลาง	s2	0	500
ขาที่ 1 ข้อปลาย	s3	135	2000
ขาที่ 2 ข้อใน	s18	54	1100
ขาที่ 2 ข้อกลาง	s32	157.5	2250
ขาที่ 2 ข้อปลาย	s30	18	700
ขาที่ 3 ข้อใน	s10	63	1200
ขาที่ 3 ข้อกลาง	s11	22.5	750
ขาที่ 3 ข้อปลาย	s4	162	2300
ขาที่ 4 ข้อใน	s23	58.5	1150
ขาที่ 4 ข้อกลาง	s22	162	2300
ขาที่ 4 ข้อปลาย	s21	27	800
ขาที่ 5 ข้อใน	s6	90	1500
ขาที่ 5 ข้อกลาง	s14	0	500
ขาที่ 5 ข้อปลาย	s5	153	2200
ขาที่ 6 ข้อใน	s27	72	1300
ขาที่ 6 ข้อกลาง	s26	180	2500
ขาที่ 6 ข้อปลาย	S25	18	700

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 รูปแบบการยืน

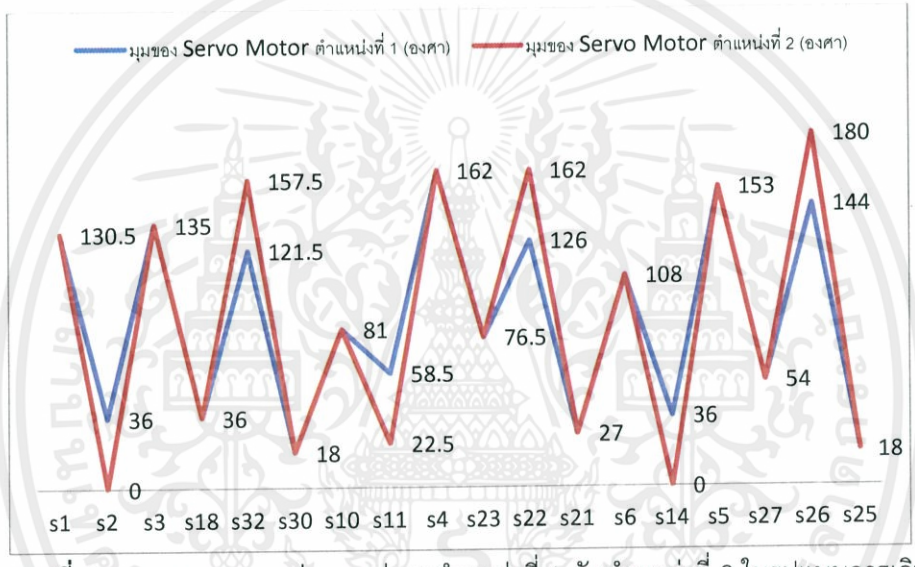
4.3.2 การทดลองหาค่ามุมของรูปแบบการเดิน

ตารางที่ 4.2 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการเดิน

ขาที่	ช่องสัญญาณที่	มุมของ Servo Motor ตำแหน่งที่ 1 (องศา)	ค่ามุมที่โปรแกรม	มุมของ Servo Motor ตำแหน่งที่ 2 (องศา)	ค่ามุมที่โปรแกรม
ขาที่ 1 ข้อใน	s1	130.5	1950	130.5	1950
ขาที่ 1 ข้อกลาง	s2	36	900	0	500
ขาที่ 1 ข้อปลาย	s3	135	2000	135	2000
ขาที่ 2 ข้อใน	s18	36	900	36	900
ขาที่ 2 ข้อกลาง	s32	121.5	1850	157.5	2250
ขาที่ 2 ข้อปลาย	s30	18	700	18	700
ขาที่ 3 ข้อใน	s10	81	1400	81	1400
ขาที่ 3 ข้อกลาง	s11	58.5	1150	22.5	750
ขาที่ 3 ข้อปลาย	s4	162	2300	162	2300
ขาที่ 4 ข้อใน	s23	76.5	1350	76.5	1350
ขาที่ 4 ข้อกลาง	s22	126	1900	162	2300

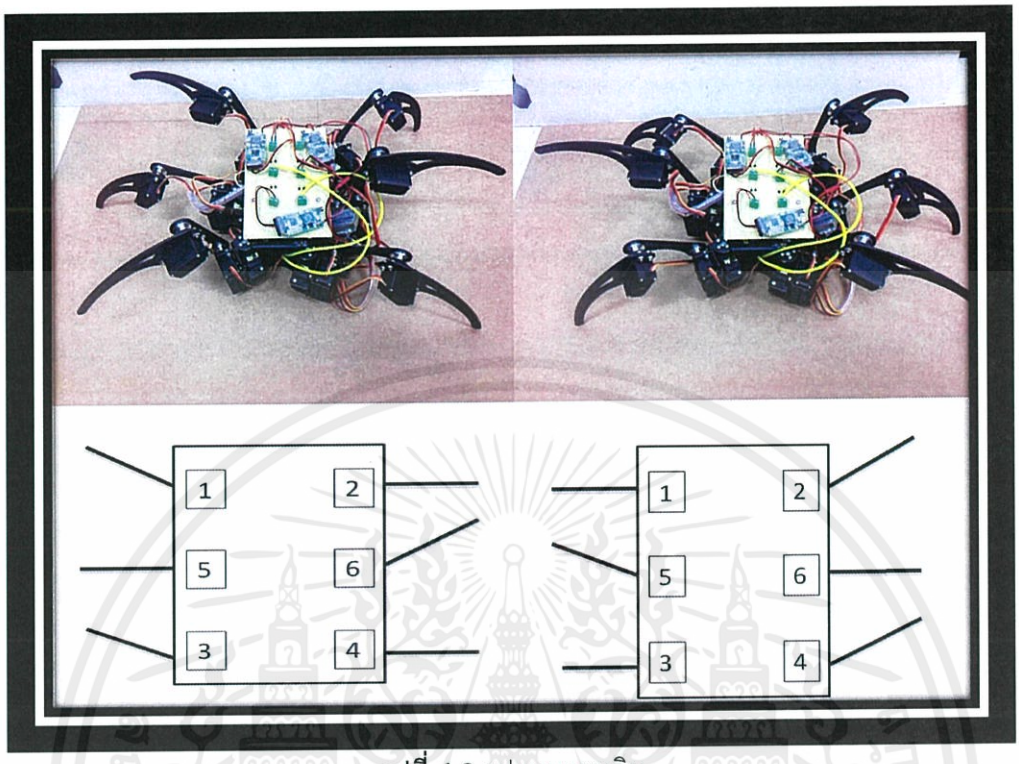
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่ 4 ข้อปลาย	s21	27	800	27	800
ขาที่ 5 ข้อใน	s6	108	1700	108	1700
ขาที่ 5 ข้อกลาง	s14	36	900	0	500
ขาที่ 5 ข้อปลาย	s5	153	2200	153	2200
ขาที่ 6 ข้อใน	s27	54	1100	54	1100
ขาที่ 6 ข้อกลาง	s26	144	2100	180	2500
ขาที่ 6 ข้อปลาย	s25	18	700	18	700



กราฟที่ 4.1 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 รูปแบบการเดิน

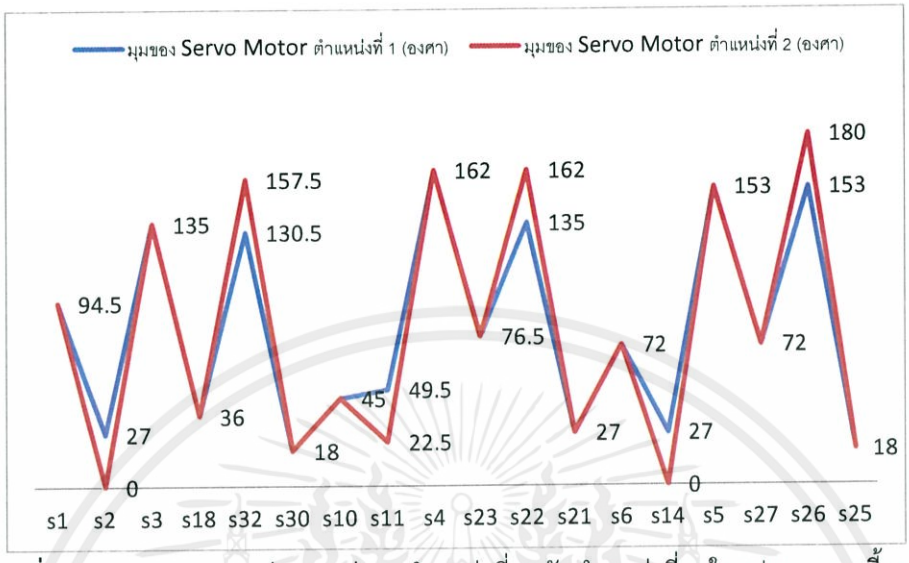
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การทดลองหาค่ามุมของรูปแบบการเลี้ยวซ้าย

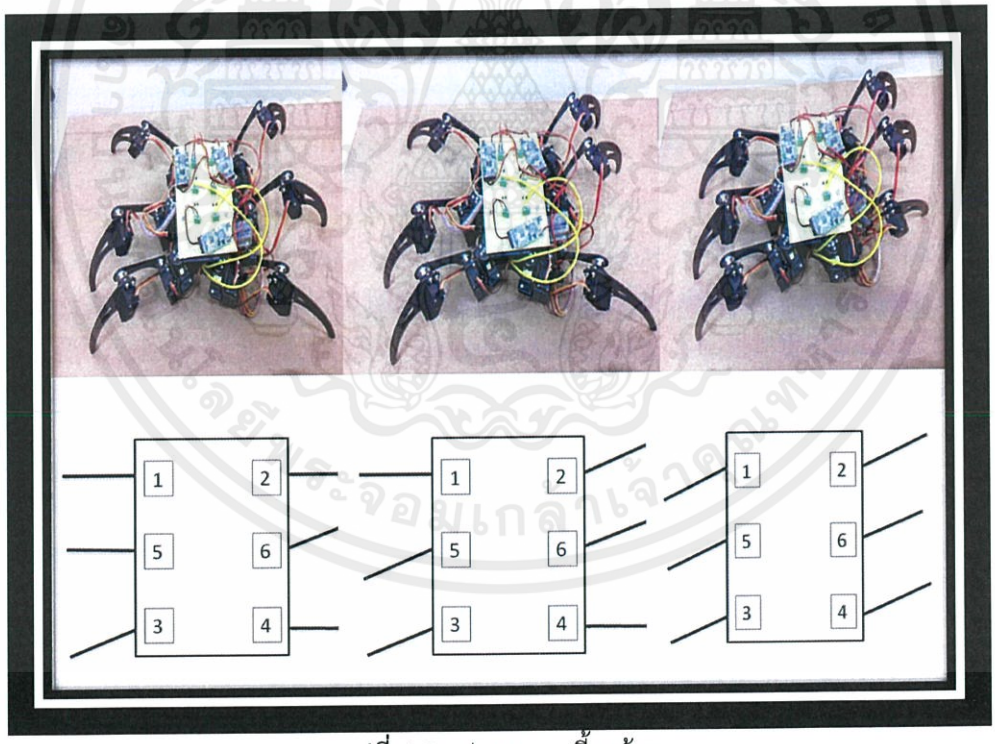
ตารางที่ 4.3 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการเลี้ยวซ้าย

ขาที่	ช่องสัญญาณที่	มุมของ Servo Motor ตำแหน่งที่ 1 (องศา)	ค่ามุมที่โปรแกรม	มุมของ Servo Motor ตำแหน่งที่ 2 (องศา)	ค่ามุมที่โปรแกรม
ขาที่ 1 ข้อใน	s1	94.5	1550	94.5	1550
ขาที่ 1 ข้อกลาง	s2	27	800	0	500
ขาที่ 1 ข้อปลาย	s3	135	2000	135	2000
ขาที่ 2 ข้อใน	s18	36	900	36	900
ขาที่ 2 ข้อกลาง	s32	130.5	1950	157.5	2250
ขาที่ 2 ข้อปลาย	s30	18	700	18	700
ขาที่ 3 ข้อใน	s10	45	1000	45	1000
ขาที่ 3 ข้อกลาง	s11	49.5	1050	22.5	750
ขาที่ 3 ข้อปลาย	s4	162	2300	162	2300
ขาที่ 4 ข้อใน	s23	76.5	1350	76.5	1350
ขาที่ 4 ข้อกลาง	s22	135	2000	162	2300
ขาที่ 4 ข้อปลาย	s21	27	800	27	800
ขาที่ 5 ข้อใน	s6	72	1300	72	1300
ขาที่ 5 ข้อกลาง	s14	27	800	0	500
ขาที่ 5 ข้อปลาย	s5	153	2200	153	2200
ขาที่ 6 ข้อใน	s27	72	1300	72	1300
ขาที่ 6 ข้อกลาง	s26	153	2200	180	2500
ขาที่ 6 ข้อปลาย	s25	18	700	18	700

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟที่ 4.2 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเลี้ยวซ้าย



รูปที่ 4.3 รูปแบบการเลี้ยวซ้าย

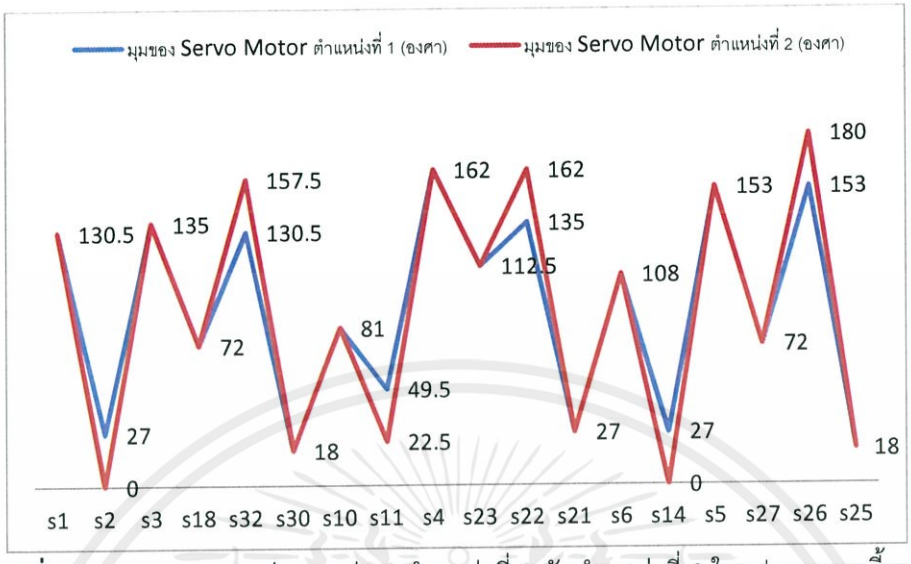
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 การทดลองหาค่ามุมของรูปแบบการเลี้ยวขวา

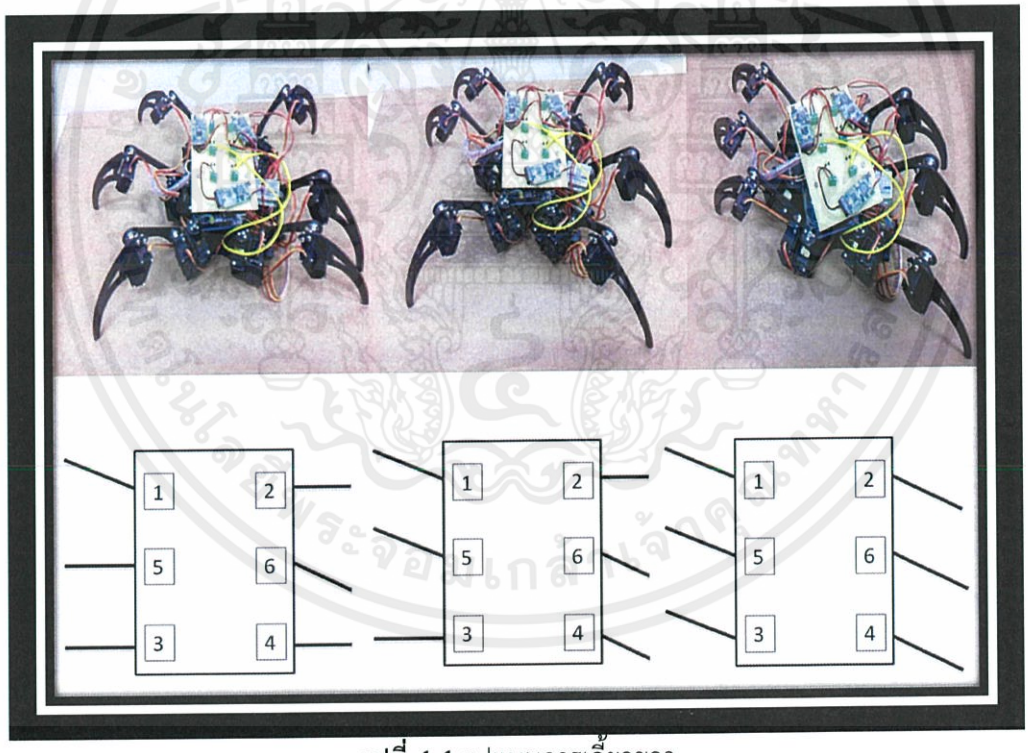
ตารางที่ 4.4 ผลการทดลองหาค่ามุมของ Servo Motor ในรูปแบบการเลี้ยวขวา

ขาที่	ช่องสัญญาณที่	มุมของ Servo Motor ตำแหน่งที่ 1 (องศา)	ค่ามุมที่โปรแกรม	มุมของ Servo Motor ตำแหน่งที่ 2 (องศา)	ค่ามุมที่โปรแกรม
ขาที่ 1 ข้อใน	s1	130.5	1950	130.5	1950
ขาที่ 1 ข้อกลาง	s2	27	800	0	500
ขาที่ 1 ข้อปลาย	s3	135	2000	135	2000
ขาที่ 2 ข้อใน	s18	72	1300	72	1300
ขาที่ 2 ข้อกลาง	s32	130.5	1950	157.5	2250
ขาที่ 2 ข้อปลาย	s30	18	700	18	700
ขาที่ 3 ข้อใน	s10	81	1400	81	1400
ขาที่ 3 ข้อกลาง	s11	49.5	1050	22.5	750
ขาที่ 3 ข้อปลาย	s4	162	2300	162	2300
ขาที่ 4 ข้อใน	s23	112.5	1750	112.5	1750
ขาที่ 4 ข้อกลาง	s22	135	2000	162	2300
ขาที่ 4 ข้อปลาย	s21	27	800	27	800
ขาที่ 5 ข้อใน	s6	108	1700	108	1700
ขาที่ 5 ข้อกลาง	s14	27	800	0	500
ขาที่ 5 ข้อปลาย	s5	153	2200	153	2200
ขาที่ 6 ข้อใน	s27	72	1300	72	1300
ขาที่ 6 ข้อกลาง	s26	153	2200	180	2500
ขาที่ 6 ข้อปลาย	s25	18	700	18	700

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟที่ 4.3 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเลี้ยวขวา



รูปที่ 4.4 รูปแบบการเลี้ยวขวา

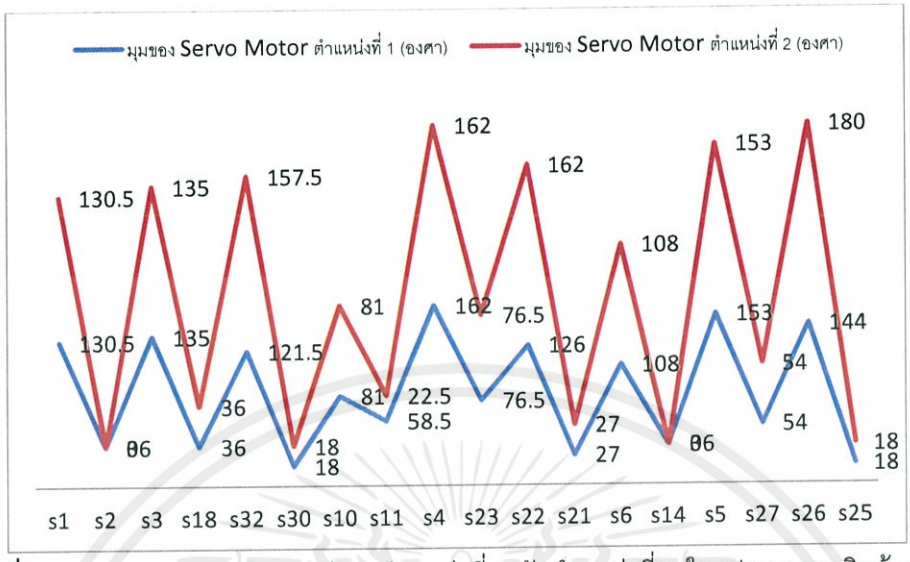
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 การทดลองหาค่ามุมในรูปแบบการเดินข้ามสิ่งกีดขวาง

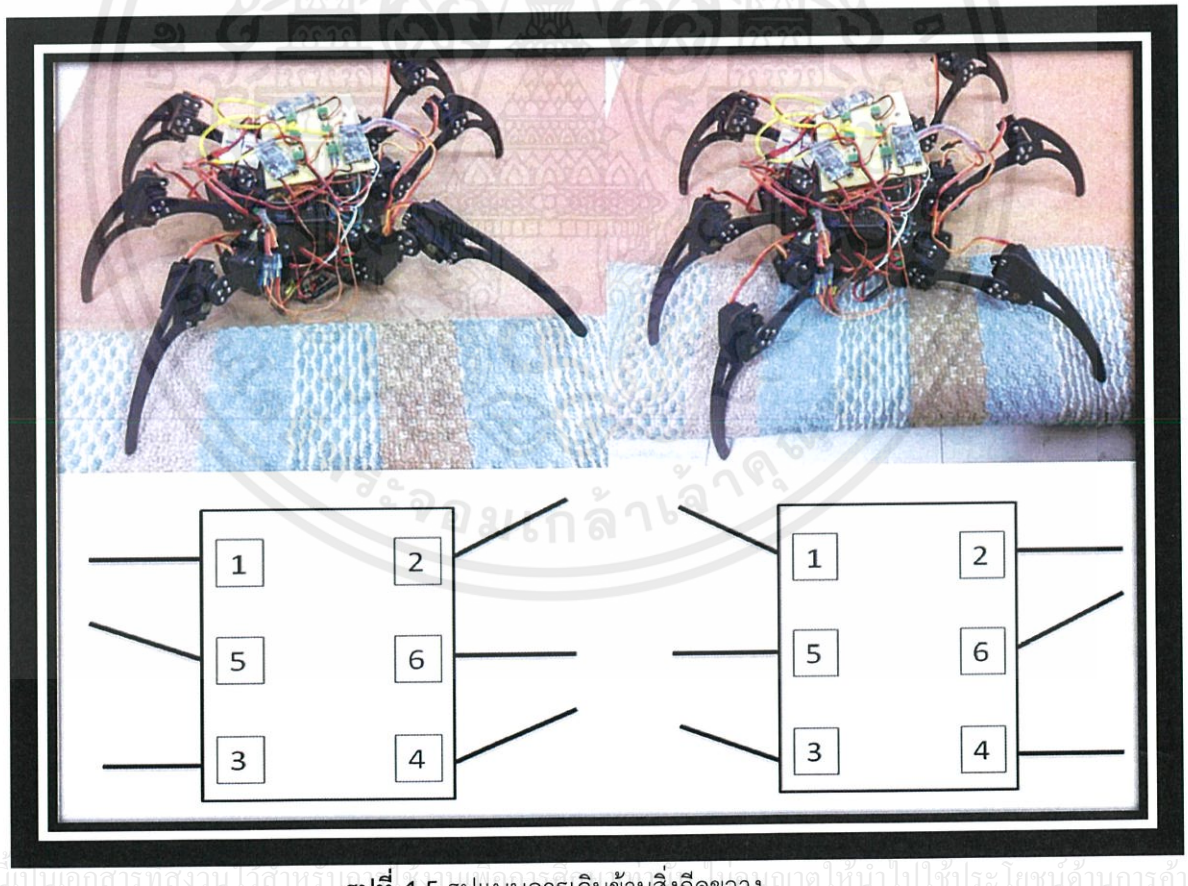
ตารางที่ 4.5 ผลการทดลองหาค่ามุมของ Servo Motor ในการเดินข้ามสิ่งกีดขวาง

ช่องสัญญาณที่	มุมของ Servo Motor ตำแหน่งที่ 1 (องศา)	ค่ามุมที่โปรแกรม	มุมของ Servo Motor ตำแหน่งที่ 2 (องศา)	ค่ามุมที่โปรแกรม
s1	130.5	1950	130.5	1950
s2	36	900	0	500
s3	135	2000	135	2000
s18	36	900	36	900
s32	121.5	1850	157.5	2250
s30	18	700	18	700
s10	81	1400	81	1400
s11	58.5	1150	22.5	750
s4	162	2300	162	2300
s23	76.5	1350	76.5	1350
s22	126	1900	162	2300
s21	27	800	27	800
s6	108	1700	108	1700
s14	36	900	0	500
s5	153	2200	153	2200
s27	54	1100	54	1100
s26	144	2100	180	2500
s25	18	700	18	700

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กราฟที่ 4.4 แสดงความแตกต่างของค่ามุมตำแหน่งที่ 1 กับตำแหน่งที่ 2 ในรูปแบบการเดินข้ามสิ่งกีดขวาง



รูปที่ 4.5 รูปแบบการเดินข้ามสิ่งกีดขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดลองทำหุ่นยนต์หกขา (Hexapod Robot) โดยใช้ Servo Motor จำนวน 18 ตัว ยึดเข้ากับตัวโครงและชิ้นส่วนขา เพื่อใช้ในการเคลื่อนที่ และใช้บอร์ด Arduino UNO R3 กับบอร์ด Torobot 32 channel serial servo controller เป็นตัวควบคุม สามารถสรุปได้ดังนี้

ในด้านฮาร์ดแวร์ของหุ่นยนต์หกขา สามารถเคลื่อนที่ได้ในระดับหนึ่ง แต่ยังไม่เสถียรภาพเท่าที่ควร เพราะคุณภาพของ Servo Motor จะต้องเลือก Servo Motor รุ่นที่ดีกว่านี้ เนื่องจาก Servo Motor แต่ละตัวต้องรับน้ำหนักของตัวโครงและอุปกรณ์ เมื่อมีการเคลื่อนที่ทำให้ไม่สามรถควบคุมมุมได้ตามต้องการ โดยค่าของมุมจะผิดพลาดไปบ้าง

ในด้านซอฟท์แวร์ เมื่อทดสอบการเคลื่อนที่ของหุ่นยนต์หกขา พบว่าสามารถตอบสนองต่อโปรแกรมที่ควบคุมได้เป็นอย่างดี ผู้จัดทำได้ทดลองและเลือกค่ามุมของ Servo Motor ทั้ง 18 ตัว ในรูปแบบการเคลื่อนที่ต่างๆ ได้แก่รูปแบบการยืน, รูปแบบการเดิน,รูปแบบการเลี้ยวซ้าย และรูปแบบการเลี้ยวขวา ทำให้ได้ค่ามุมที่ดีสุดในรูปแบบการเคลื่อนที่ต่างๆ

5.2 ข้อเสนอแนะ

จากการศึกษา ออกแบบ และทำการทดลอง ทำให้ทราบถึงปัญหาที่เกิดขึ้น ดังนี้

1. เสถียรภาพในการเคลื่อนที่ของหุ่นยนต์หกขา มีสาเหตุมาจาก Servo Motor ที่ไม่มีคุณภาพส่งผลทำให้ควบคุมมุมที่ต้องการได้ยาก ดังนั้นควรจะเลือก Servo Motor ที่มีคุณภาพดีกว่านี้

2. Servo Motor ใช้กระแสไฟสูงมาก จึงต้องมีวงจรแปลงกระแสก่อนจ่ายไฟให้ Servo Motor โดยใช้วงจร Universal Battery Elimination Circuit (UBEC)

3. ควรจัดเรียงสายไฟให้เหมาะสม ทำให้ง่ายต่อการตรวจสอบสายเมื่อมีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] วัชรินทร์ เคารพ. **คู่มือการใช้งาน Servo Motor**. กรุงเทพฯ : บริษัท อีทีที จำกัด

[2] Brian Evans. **Beginning Arduino Programming** : Technology in Action

[3] Micbael Margolis. 2011. **Arduino Cookbook**. 3nd ed : O'REILLY

[4] Ann L. Torres. 1996. "**Virtual Model Control of a Hexapod Walking Robot**." Bachelor of Science, Massachusetts Institute of Technology.

[5] SolidWorks. [Online]. Available : www.learn.in.th/training/SolidWorksbyMTEC.pdf

[6] Tates Performance Hobbies. **UBEC**. [Online]. Available : www.rcworld.com.

[7] Torobot. **Servo Controller Instruction for Use**. [Online]. Available : www.torobot.com/down/usc_en.pdf

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

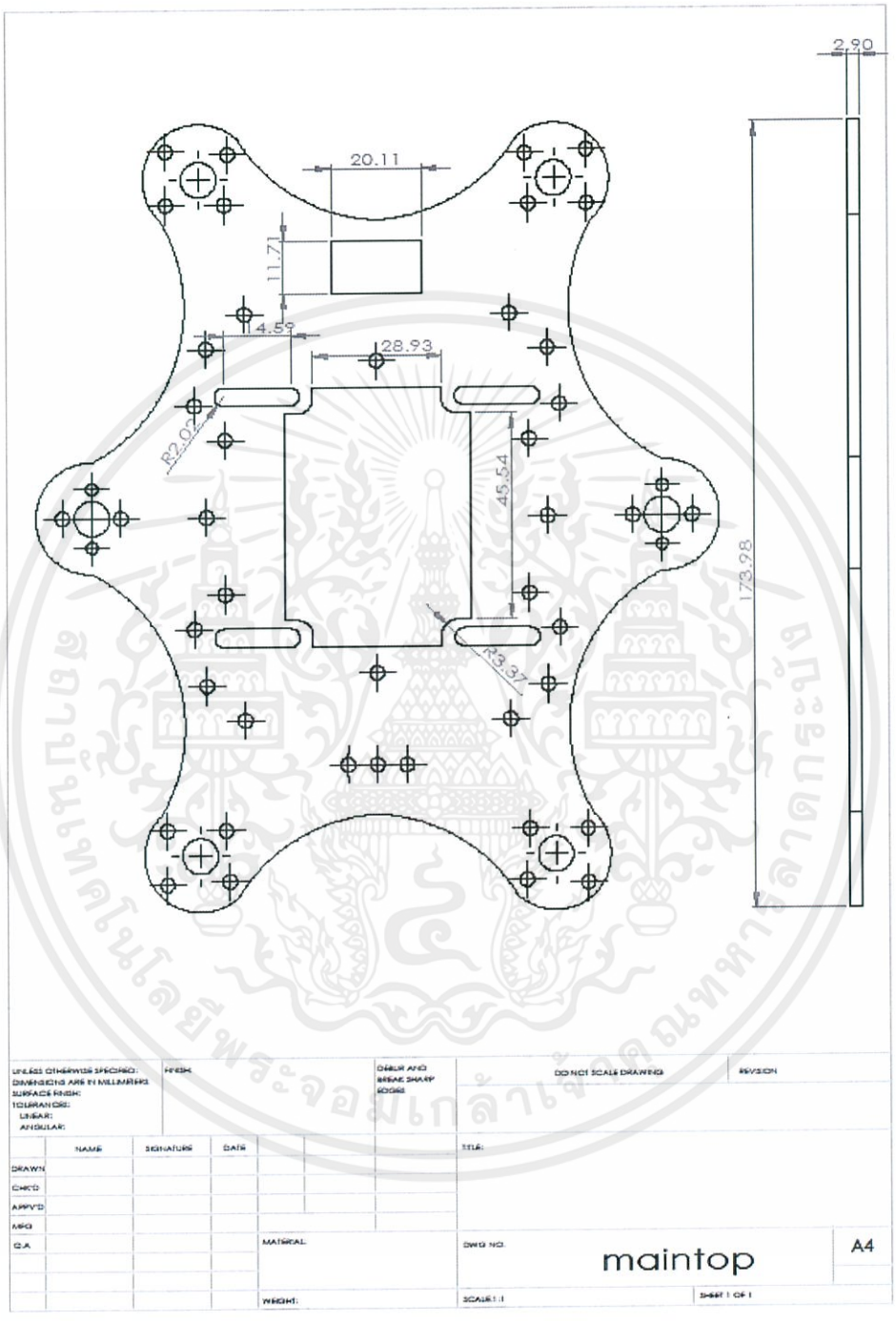
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

แบบร่างชิ้นส่วนของโครง hexapod robot

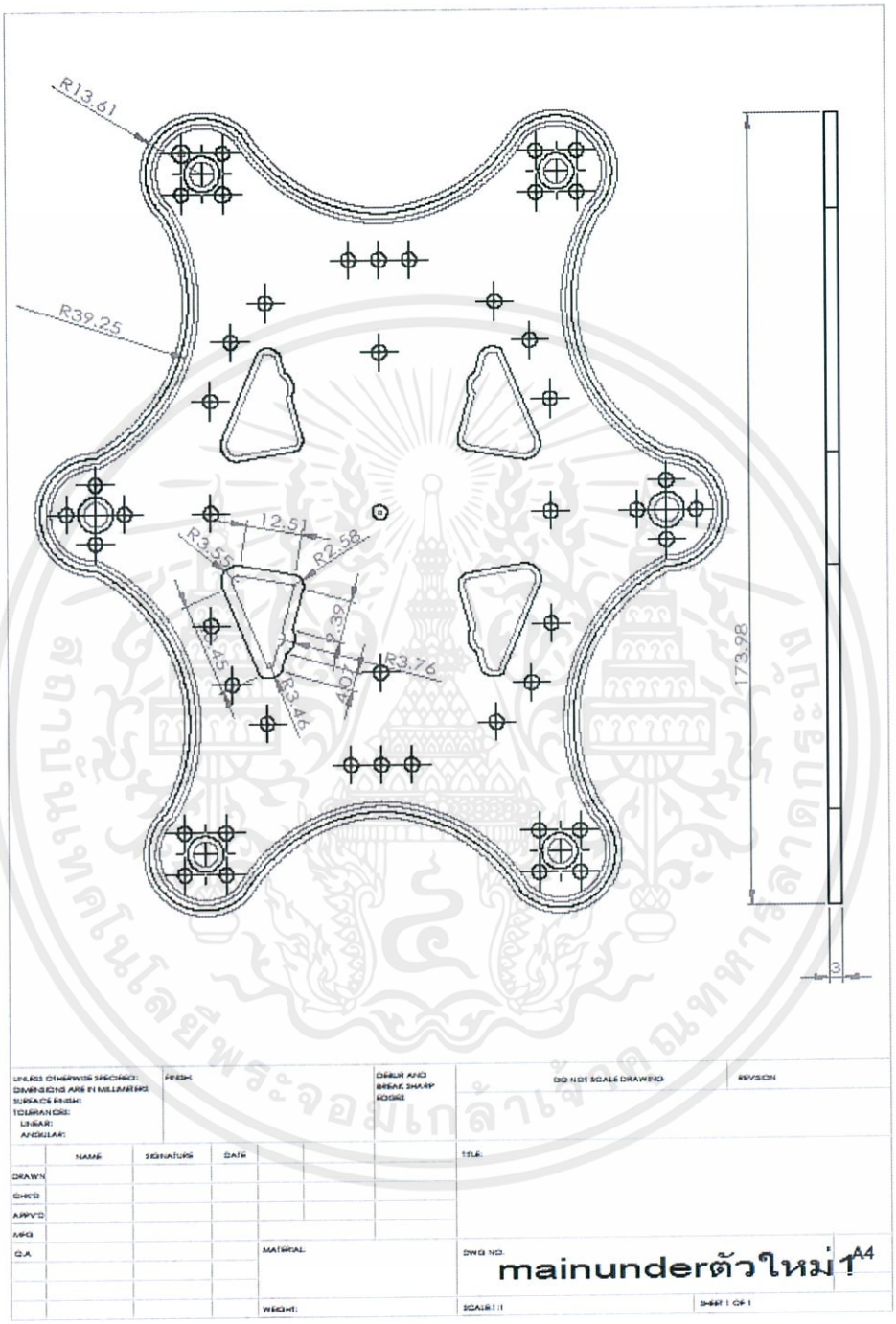
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบร่างชิ้นส่วนด้านบนของหุ่นยนต์หกขา



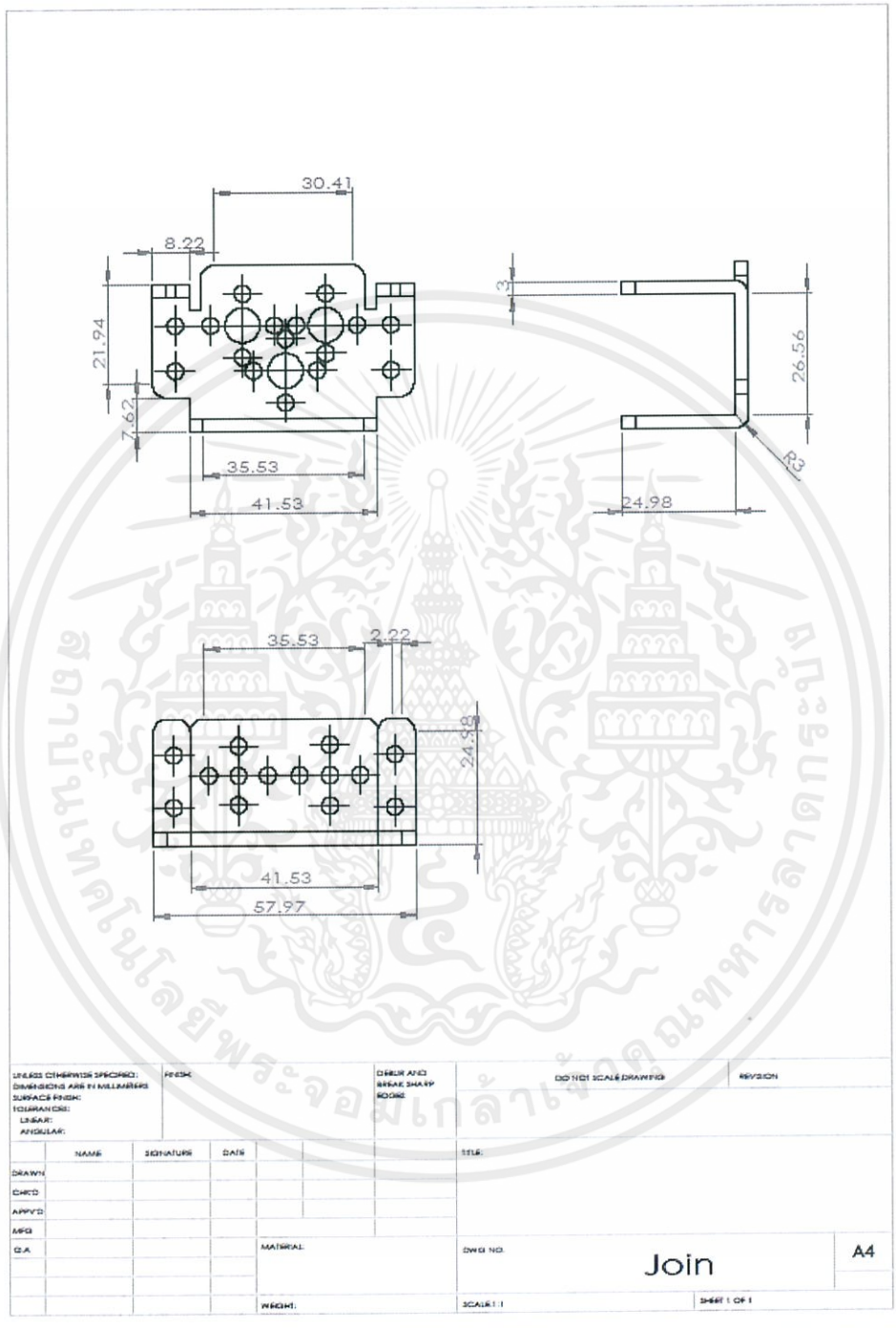
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบร่างชิ้นส่วนด้านล่างของหุ่นยนต์หกขา



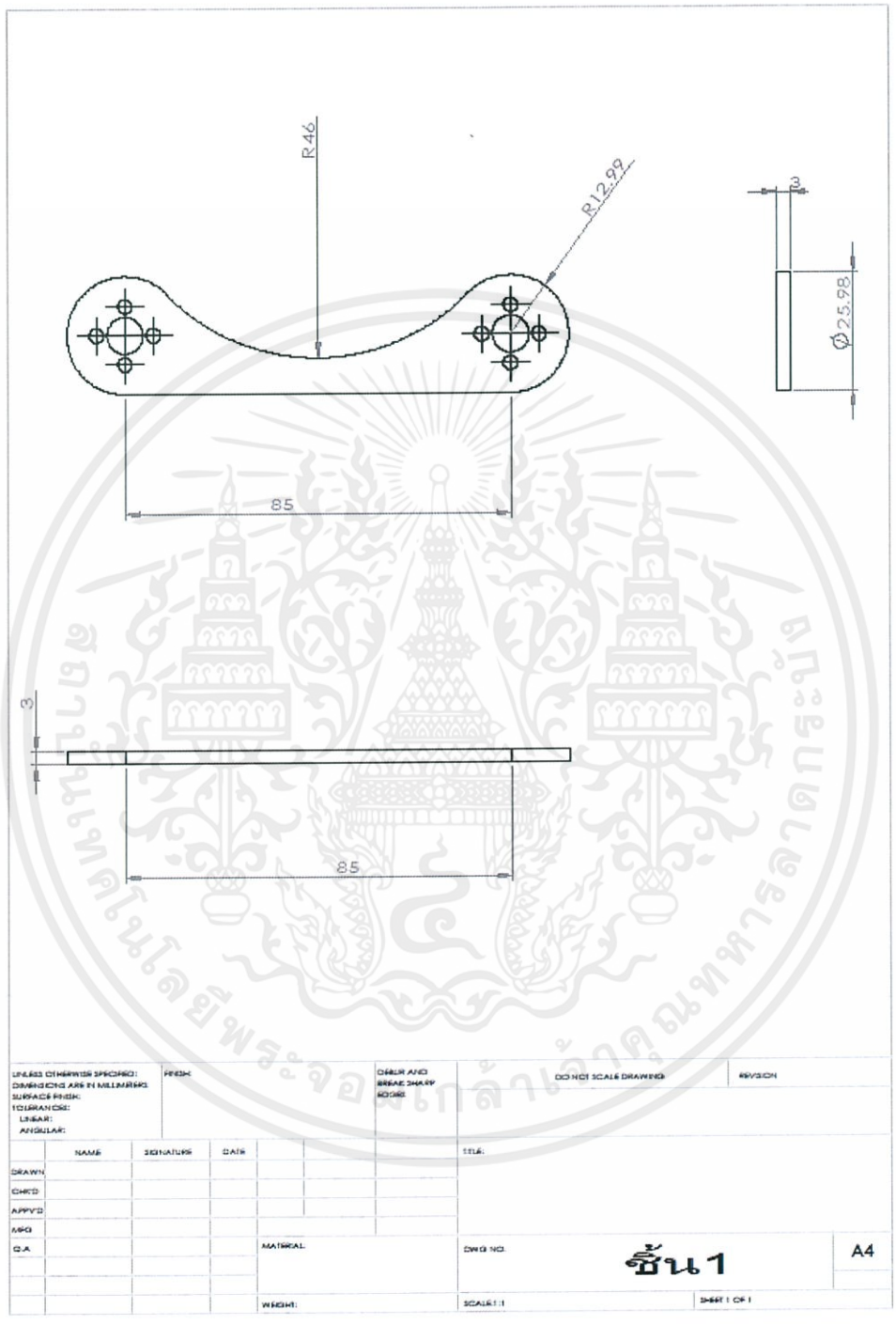
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบร่างชิ้นส่วนข้อต่อที่ยึดกับ servo motor



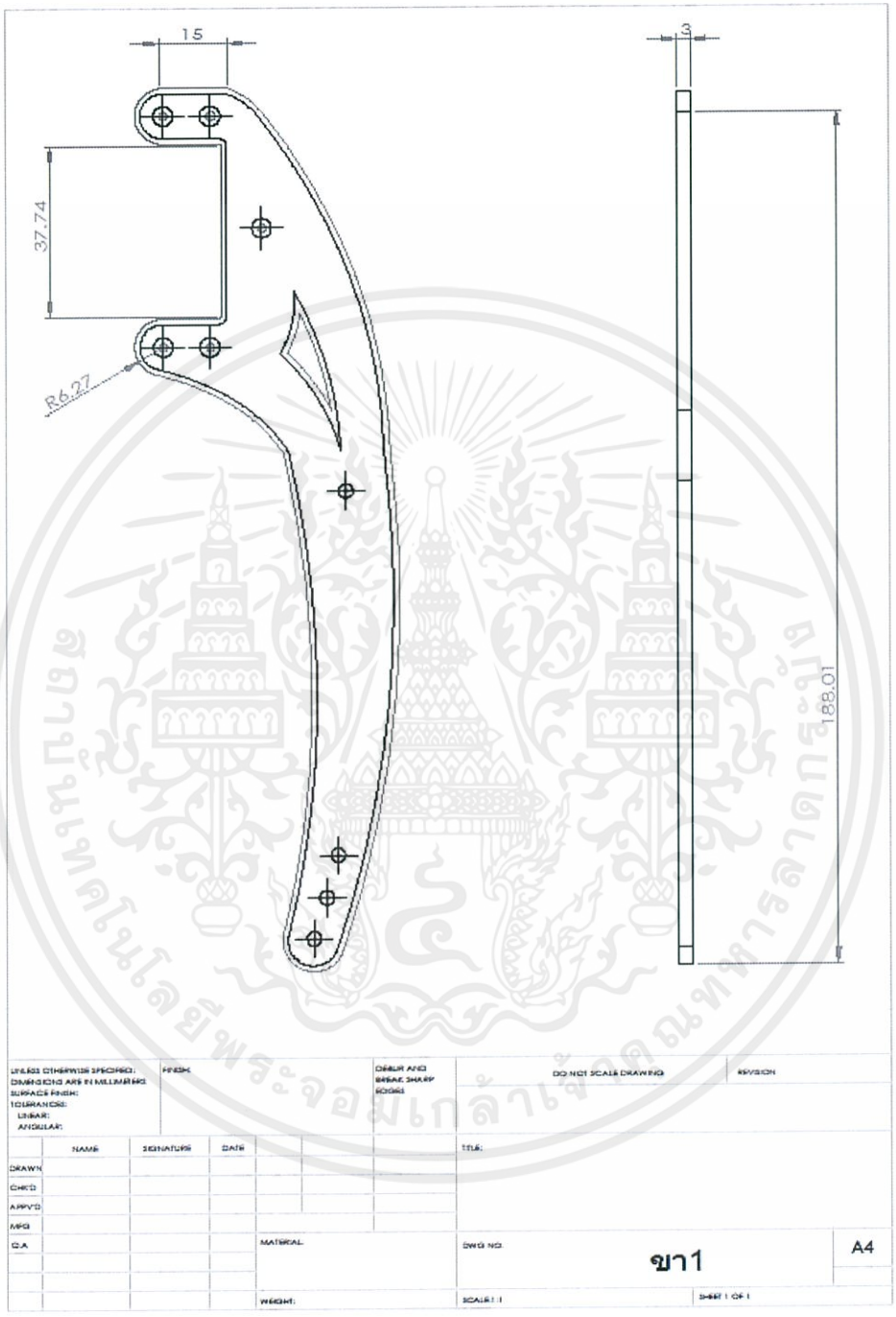
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบร่างชิ้นส่วนที่ยึดเฟืองของ servo motor



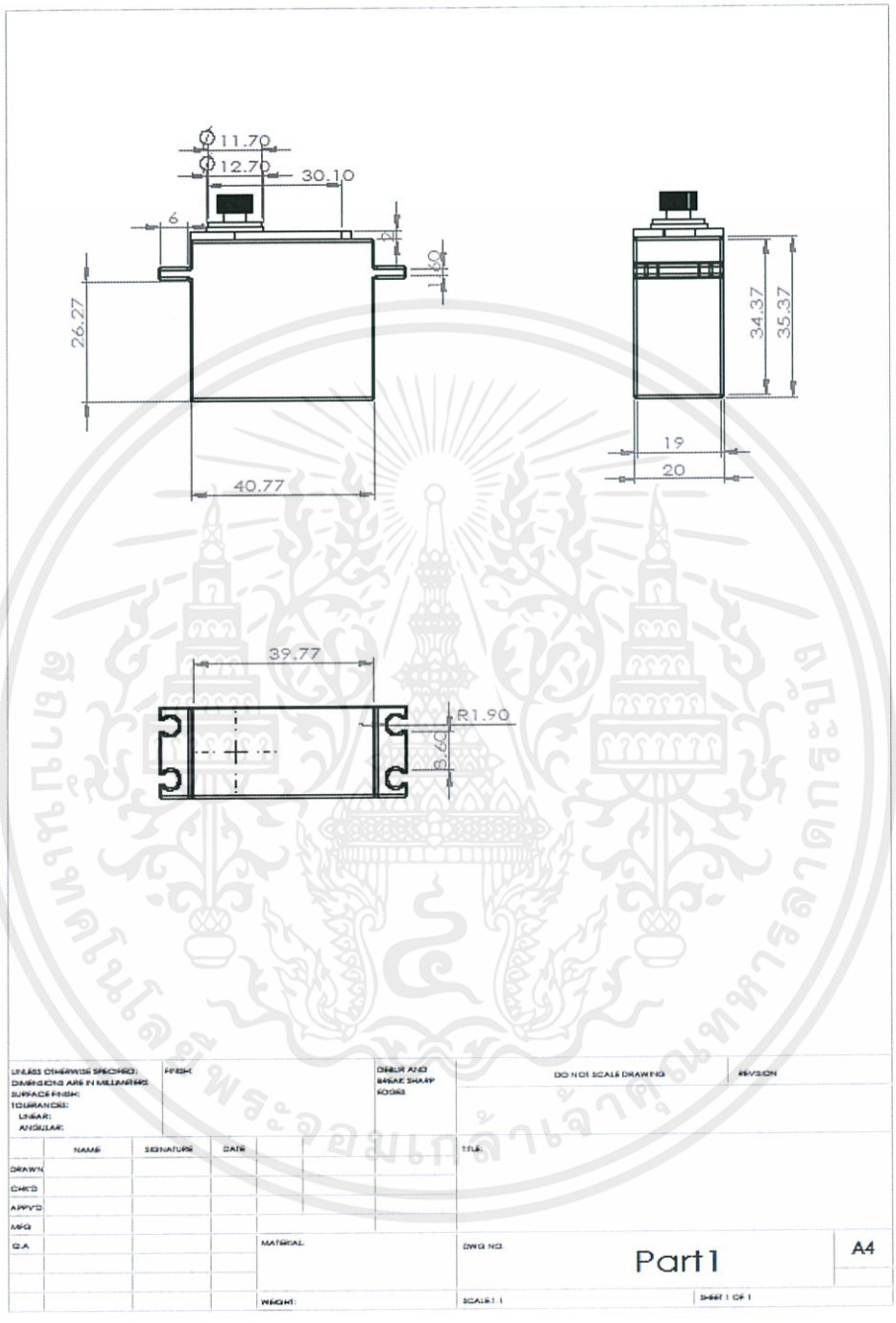
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบร่างชิ้นส่วนขา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบร่าง servo motor



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

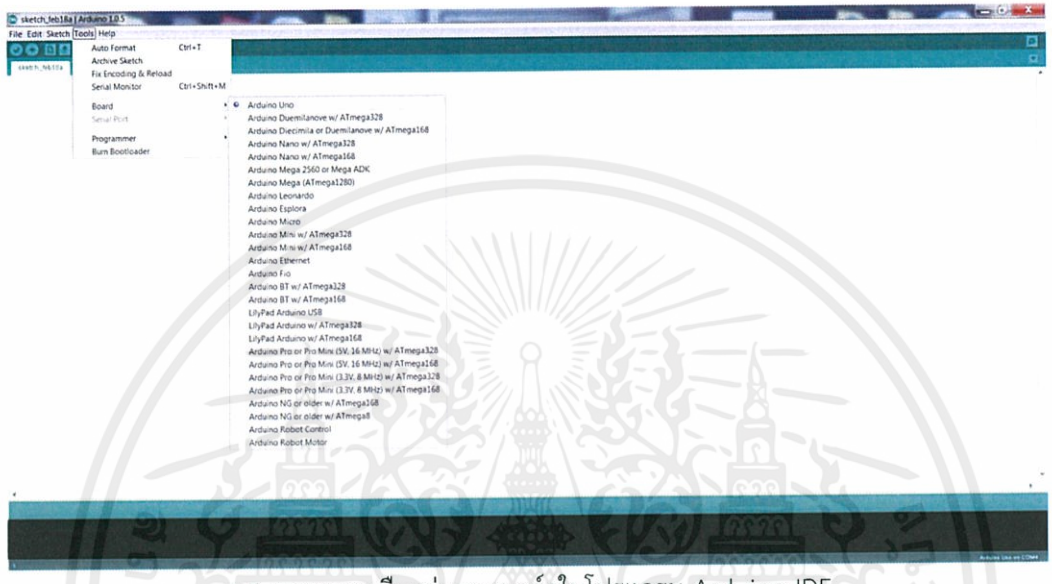
ภาคผนวก ข.

โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้โปรแกรม Arduino IDE

1. เปิดโปรแกรม Arduino IDE ซึ่งโปรแกรมนี้จะใช้กับบอร์ด Arduino หลายรุ่น ดังนั้นจะต้องกำหนดบอร์ดที่ใช้ในโปรแกรมนี้ โดยเข้าที่ Tool ตามด้วย Board แล้วเลือก Arduino UNO



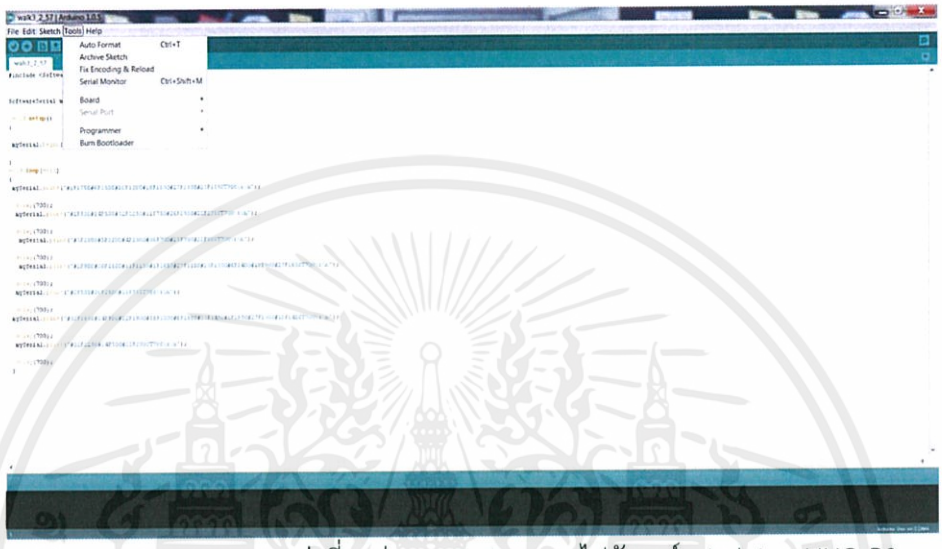
ข.1 แสดงการเลือกรุ่นของบอร์ดในโปรแกรม Arduino IDE

2. การตั้งค่าเริ่มต้นของโปรแกรม เป็นการติดต่อสื่อสารระหว่างบอร์ด Arduino UNO R3 กับบอร์ด Torobot 32 channel serial servo controller ดังนี้

```
 #include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);         //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
 void setup()
{
 mySerial.begin(9600);                //อัตราการส่งข้อมูล
}
void loop(void)                       //เขียนคำสั่งควบคุมการเคลื่อนที่
{
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อเขียนคำสั่งควบคุมการเคลื่อนที่แล้ว ให้กด Verify เพื่อเป็นการตรวจสอบพื้นฐาน จากนั้นให้กด Upload ค่าที่ได้โปรแกรมก็จะถูกบันทึกลงบอร์ด Arduino UNO R3 และถ้าต้องการตรวจสอบค่าที่ถูกส่งจาก Computer ไปยังบอร์ด เข้าที่ Tool แล้วเลือก Serial Monitor



ข.2 แสดงการตรวจสอบค่าที่ถูกส่งจาก Computer ไปยังบอร์ด Arduino UNO R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ควบคุมหุ่นยนต์หกขาในรูปแบบการยืน

```
#include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);        //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
void setup()
{
mySerial.begin(9600);                //อัตราการส่งข้อมูล
}
void loop(void)                      //เขียนคำสั่งควบคุมรูปแบบการยืน
{
mySerial.print("#1P1750#6P1500#10P1200#18P1100#27P1300#23P1550T1000\r\n");
delay(1000);
mySerial.print("#2P500#14P500#32P2250#11P750#26P2500#22P2300T1000\r\n");
delay(1000);
mySerial.print("#3P2000#5P2200#4P2300#30P700#25P700#21P800T1000\r\n");
delay(1000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ควบคุมหุ่นยนต์หกขาในรูปแบบการเดิน

```
#include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);        //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
void setup()
{
mySerial.begin(9600);                //อัตราการส่งข้อมูล
}
void loop(void)                      //เขียนคำสั่งควบคุมรูปแบบการเดิน
{
mySerial.print("#1P1750#6P1500#10P1200#18P1100#27P1300#23P1550T700\r\n");
delay(700);
mySerial.print("#2P500#14P500#32P2250#11P750#26P2500#22P2300T700\r\n");
delay(700);
mySerial.print("#3P2000#5P2200#4P2300#30P700#25P700#21P800T700\r\n");
delay(700);
mySerial.print("#2P900#26P2100#11P1150#1P1850#27P1100#10P1300#6P1400#18P900#
23P1650T700\r\n");
delay(700);
mySerial.print("#2P500#26P2500#11P750T700\r\n");
delay(700);
mySerial.print("#32P1850#14P900#22P1900#18P1000#6P1600#23P1450#1P1950#27P130
0#10P1400T700\r\n");
delay(700);
mySerial.print("#32P2250#14P500#22P2300T700\r\n");
delay(700);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ควบคุมหุ่นยนต์หกขาในรูปแบบการเลี้ยวซ้าย

```
#include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);        //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
void setup()
{
mySerial.begin(9600);                //อัตราการส่งข้อมูล
}
void loop(void)                      //เขียนคำสั่งควบคุมรูปแบบการเลี้ยวซ้าย
{
mySerial.print("#1P1750#6P1500#10P1200#18P1100#27P1300#23P1550T1000\r\n");
delay(1000);
mySerial.print("#2P500#14P500#32P2250#11P750#26P2500#22P2300T1000\r\n");
delay(1000);
mySerial.print("#3P2000#5P2200#4P2300#30P700#25P700#21P800T1000\r\n");
delay(1000);
mySerial.print("#2P800#1P1550#26P2200#23P1350T1000\r\n");
delay(1000);
mySerial.print("#2P500#26P2500T1000\r\n");
delay(1000);
mySerial.print("#14P800#6P1300#22P2000#23P1350T1000\r\n");
delay(1000);
mySerial.print("#14P500#22P2300T1000\r\n");
delay(1000);
mySerial.print("#32P1950#18P900#11P1050#10P1000T1000\r\n");
delay(1000);
mySerial.print("#32P2250#11P750T1000\r\n");
delay(1000);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ควบคุมหุ่นยนต์หกขาในรูปแบบการเลี้ยวขวา

```
include <SoftwareSerial.h>          //เรียกใช้ library ของโปรแกรมในการควบคุม Servo
SoftwareSerial mySerial(6,7);       //กำหนดให้ Pin 6 เป็น TX และให้ Pin 7 เป็น RX
void setup()
{
mySerial.begin(9600);               //อัตราการส่งข้อมูล
}
void loop(void)                     //เขียนคำสั่งควบคุมรูปแบบการเลี้ยวขวา
{
mySerial.print("#1P1750#6P1500#10P1200#18P1100#27P1300#23P1550T1000\r\n");
delay(1000);
mySerial.print("#2P500#14P500#32P2250#11P750#26P2500#22P2300T1000\r\n");
delay(1000);
mySerial.print("#3P2000#5P2200#4P2300#30P700#25P700#21P800T1000\r\n");
delay(1000);
mySerial.print("#2P800#1P1950#26P2200#23P1750T1000\r\n");
delay(1000);
mySerial.print("#2P500#26P2500T1000\r\n");
delay(1000);
mySerial.print("#14P800#6P1700#22P2000#23P1750T1000\r\n");
delay(1000);
mySerial.print("#14P500#22P2300T1000\r\n");
delay(1000);
mySerial.print("#32P1950#18P1300#11P1050#10P1400T1000\r\n");
delay(1000);
mySerial.print("#32P2250#11P750T1000\r\n");
delay(1000)
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้