สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาและควบคุมต้นแบบหุ่นยนต์ห้าขา

DEVELOPMENT AND CONTROL OF A FIVE-LEGGED ROBOT PROTOTYPE

ธนดิตถ์ สุรังษี
พิศาล สุขสาม
อภิวัฒน์ พลสันติกุล

วพ.
ธ141ก
9649

เลขหมู่.....
เลขทะเบียน 72848
วัน,เดือน,ปี 23 มิ.ย. 2550

b 1117325x
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาและควบคุมต้นแบบหุ่นยนต์ห้าขา

DEVELOPMENT AND CONTROL OF A FIVE-LEGGED ROBOT PROTOTYPE

โดย

นายธนดิตถ์ สุรังษี

นายพิศาล สุขสาม

นายอภิวัฒน์ พลสันติกุล

อาจารย์ที่ปรึกษา

ดร.อำนาจ คณะรัฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชา วิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาและควบคุมต้นแบบหุ่นยนต์ห้าขา

DEVELOPMENT AND CONTROL OF A FIVE-LEGGED ROBOT PROTOTYPE

ผู้จัดทำ

1. นาย ธนดิตถ์ สุรังษี รหัสประจำตัว 46010271
2. นาย พิศาล สุขสาม รหัสประจำตัว 46010531
3. นาย อภิวัฒน์ พลสันติกุล รหัสประจำตัว 46010923

อาจารย์ที่ปรึกษา

(ดร.อำนาจ คณะรัฐ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาและควบคุมต้นแบบหุ่นยนต์ห้าขา

นาย ธนดิตถ์ สุรังษี 46010271
นาย พิศาล สุขสาม 46010531
นาย อภิวัฒน์ พลสันติกุล 46010923
ดร.อำนาจ คณะรัฐ อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

ในปัจจุบันมนุษย์เราได้นำเอาหุ่นยนต์มาใช้งานอย่างแพร่หลาย ในงานหลายๆ ประเภท เช่น การประกอบรถยนต์ซึ่งต้องอาศัยความถูกต้องแม่นยำ หรืองานที่มีความเสี่ยงภัย เช่น การสืบข่าวกรองทางการทหาร หรือไม่ว่าจะเป็นการเข้าไปช่วยเหลือคนที่กำลังประสบภัยต่างๆ ในพื้นที่ที่ยากต่อการเข้าถึง ในกรณีหลังนี้หุ่นยนต์จำเป็นจะต้องใช้ขาในการเคลื่อนที่ในพื้นที่เหล่านั้น ดังนั้นบทความฉบับนี้ กล่าวถึงการออกแบบและสร้างต้นแบบหุ่นยนต์ห้าขาขนาดเล็ก เพื่อนำมาศึกษาและออกแบบลักษณะการเดินแบบสถิตย์ของหุ่นห้าขา โดยผลที่ได้จากการศึกษานี้ จะถูกนำไปประยุกต์ใช้ในการออกแบบและพัฒนาหุ่นยนต์กู้ภัยห้าขาในอนาคต โดยในขณะที่ทำงานมีลักษณะการยืนแบบสแตติก 3 ขา และเปลี่ยนขา 2 ข้างไปเป็นแขนกล ที่มีอุปรณ์ในการทำงานต่างๆ เก็บอยู่ที่ปลายขาเพื่อใช้ในการกู้ภัย ส่วนในขณะเดินจะใช้ขาทั้ง 5 ตามปรกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEVELOPMENT AND CONTROL OF A FIVE-LEGGED ROBOT PROTOTYPE

Tanadit Surungsee

Pisarn Suksam

Apiwat Polsuntikul

Dr.Amnart Kanarat Advisor

ABSTRACT

In the present, we use robots in many arrlications such as assembling parts of automobile, working as a spy for military, or helping officer in rescue mission. Therefore this thesis shows you about designing and building a scaled prototype of a five-legged robot for studying and designing the characteristic of static walking of the five-legged robot. The result of this study can be applied to design and develop a rescue of five-legged robot in the future. When the robot works, it stands on three legs, and turns two legs into two arms with end effectors, which are normally folded inside the legs. But, when it walks, it uses all legs to walk.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง หุ่นยนต์ห้าขา ไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และความร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึง เพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์อำนาจ คณะรัฐ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก ขอขอบคุณ อาจารย์ธวัชชัย คำศรี ที่ให้คำแนะนำเกี่ยวกับไมโครคอลโทรลเลอร์พิค คุณสถาพร สายใจ ที่ให้คำแนะนำเกี่ยวกับการเขียนโปรแกรมภาษา ซี คุณนครินทร์ คำเขียว ที่ให้คำแนะนำเกี่ยวกับวงจรอิเล็คทรอนิค และชุมนุมโรบอทมี่ให้ยืมบอร์ดทดลองและอุปกรณ์ต่างๆ

และต้องขอขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจเอาใจใส่เสมอมาในทุกๆ ด้านอันหาที่เปรียบมิได้ รวมไปถึงคณาจารย์ และบุคคลากร ของภาคเครื่องกล ที่คอยให้คำปรึกษาและความช่วยเหลือมาโดยตลอด ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

ธนดิตถ์ สุรังษี

พิศาล สุขสาม

อภิวัฒน์ พลสันดิกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตของโครงงาน	2
1.4 วิธีการดำเนินโครงงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 แนวคิดในการสร้างและพัฒนาหุ่นยนต์	4
2.1 ความสำคัญของหุ่นยนต์	4
2.2 การเคลื่อนที่ของหุ่นยนต์โดยใช้ขา	5
2.3 รูปแบบการเดินของหุ่นยนต์โดยใช้ขา	6
2.3.1 การเดินแบบสถิต	6
2.3.2 การเดินแบบจลน์	6
2.4 การพิจารณาเลือกจำนวนขาของหุ่นยนต์	6
2.4.1 ข้อดีของการเดินหลายขา	6
2.5 แนวคิดการพัฒนาหุ่นยนต์ห้าขาในอนาคต	7
บทที่ 3 ทฤษฎีและหลักการ	8
3.1 ทฤษฎีโมเมนต์ของแรง	8
3.1.1 โมเมนต์ของแรงคู่ควบ	9
3.2 แรงเสียดทาน	11
3.2.1 แรงเสียดทานสถิตย์	11
3.2.2 แรงเสียดทานจลน์	11
3.2.3 สัมประสิทธิ์ความเสียดทาน	12
3.3 สมดุล	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.4 หลักการทำงานของเซอร์โวมอเตอร์	15
3.5 บอร์ดขับมอเตอร์ ZX-SERVO 16	17
3.5.1 การทำงานของวงจร	19
3.5.2 ขั้นตอนการใช้งาน	20
3.5.3 การเขียนโปรแกรมติดต่อกับบอร์ด ZX-SERVO16	21
3.5.3.1 รูปแบบการสื่อสารข้อมูลอนุกรม	21
3.5.3.2 การกำหนดค่าบอดเรตสำหรับบอร์ด ZX-SERVO16	21
3.5.3.3 คำสั่งสำหรับกำหนดตำแหน่งของเซอร์โวมอเตอร์	21
3.5.3.4 การสร้างคำสั่งเพื่อระบุตำแหน่งของเซอร์โวมอเตอร์	22
3.6 ไมโครคอนโทรเลอร์ (Microcontroller)	22
3.6.1 โครงสร้างของไมโครคอนโทรลเลอร์ (PIC)	23
3.7 บอร์ดไมโครคอนโทรเลอร์ CP-PIC V4.0	25
3.7.1 ลักษณะโดยทั่วไป	26
3.7.2 แหล่งจ่ายไฟ	28
3.7.3 สัญญาณนาฬิกา	28
3.7.4 โหมดการทำงานของบอร์ด	30
3.7.5 การทำงานในโหมดการโปรแกรม	30
3.7.6 การทำงานใน USER MODE หรือ RUN MODE	30
3.8. การควบคุมการเดินของหุ่นยนต์	31
3.8.1. ควบคุมผ่านพอร์ตอนุกรม โดยใช้โปรแกรม MATLAB สั่งการ	31
3.8.2 ควบคุมโดยไมโครคอนโทรลเลอร์ PIC 18F458 ผ่านบอร์ด CP-PIC V4.0	31
3.8.3. ควบคุมผ่านคอมพิวเตอร์ โดยใช้ภาษาซีผ่านพอร์ตอนุกรม	31
บทที่ 4 การออกแบบและการสร้าง	32
4.1 วัสดุและอุปกรณ์	32
4.1.1 วัสดุที่ใช้ทำหุ่นยนต์	32
4.1.2 ต้นกำลัง	33
4.1.2.1 การเลือกต้นกำลัง	33
4.1.2.2 เซอร์โวมอเตอร์	33
4.2 การออกแบบหุ่นยนต์	34
4.2.1 การออกแบบลักษณะการเดิน	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.2.1.1 เดินแบบ 6 ขั้นตอน 34

4.2.1.2. การพัฒนาการเดิน 35

4.2.2 การออกแบบขนาดของหุ่นยนต์ 36

4.2.2.1 ปัจจัยที่มีผลต่อการออกแบบขนาดของขาหุ่นยนต์และขนาดตัวหุ่น 36

4.2.2.2 การคำนวณหาขนาดของหุ่นยนต์ 37

4.3 การสร้างหุ่น 44

4.3.1 การสร้างต้นแบบขนาดเล็กของหุ่นยนต์ 44

4.4 วิธีการและโปรแกรมที่ใช้ควบคุมหุ่น 49

4.4.1 ทำการเทียบคำสั่งในโปรแกรม กับค่ามุมจริงที่เซอร์โวมอเตอร์หมุนไป 49

4.4.2 การคำนวณ จลนศาสตร์ผกผันแบบกลใกสองก้านโยง 50

4.4.3 การหาสมการคณิตศาสตร์ เพื่อใช้ในการเขียนโปรแกรมควบคุมให้หุ่นหมุนตัว 52

บทที่ 5 การเชื่อมต่อและวิธีการใช้งานอุปกรณ์ต่างๆ 54

5.1 ส่วนประกอบหลักของหุ่นยนต์ห้าขา 54

5.1.1 ตัวหุ่นยนต์ 54

5.1.2 แหล่งจ่ายพลังงาน 55

5.1.3 คอมพิวเตอร์และโปรแกรม 55

5.1.4 การเชื่อมต่ออุปกรณ์และการสั่งการ 55

บทที่ 6 การทดลอง 59

6.1 อุปกรณ์การทดลอง 59

6.2 การอ่านค่าผลการทดลอง 63

6.3 การทดลองเดินเป็นเส้นตรง 64

6.4 การทดลองการเลี้ยว 65

6.5 การทดลองการเดินเป็นสี่เหลี่ยมจัตุรัส 69

6.6 วิจารณ์ผลการทดลอง 71

บทที่ 7 สรุปผลและวิจารณ์ 72

7.1 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข 72

7.2 แนวทางในการพัฒนาต่อ 72

บรรณานุกรม 74

ภาคผนวก ก การทำงานของโปรแกรม 75

ภาคผนวก ข โปรแกรมที่ใช้ควบคุม 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2-1 ความต้องการใช้หุ่นยนต์ในการปฏิบัติงานต่างๆ	4
ตารางที่ 2-2 ประเภทของกลไกการเคลื่อนที่ที่ใช้กับหุ่นยนต์	5
ตารางที่ 2-3 ตารางเปรียบเทียบจำนวนขาเพื่อทำการสร้างหุ่นยนต์	7
ตารางที่ 3-1 สัมประสิทธิ์ความเสียดทานสถิต μ_s และสัมประสิทธิ์ความเสียดทานจลน์ μ_k	13
ตารางที่ 3.2 ไมโครคอนโทรลเลอร์พิค	24
ตารางที่ 3.3 ตารางแสดงคุณสมบัติ CPU ของไมโครคอนโทรลเลอร์พิคแต่ละเบอร์	26
ตารางที่ 4.1 แสดงคุณสมบัติที่เราต้องการของโลหะชนิดต่างๆ ที่เรานำมาพิจารณาเลือก	32
ตารางที่ 4.2 แสดงคุณสมบัติจำเป็นที่เราต้องการ ของต้นกำลัง	33
ตารางที่ 4.3 แสดงคุณสมบัติของเซอร์โวมอเตอร์	34
ตารางที่ 4.4 คุณสมบัติเซอร์โวมอเตอร์ขนาดต่างๆ	36
ตารางที่ 6.1 แสดงผลการทดลองของการเดินเป็นเส้นตรง	64
ตารางที่ 6.2 แสดงผลการทดลองของการเลี้ยว	66
ตารางที่ 6.3 แสดงผลตารางการทดลองการเดินเป็นสี่เหลี่ยมจัตุรัส	70

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่ 3.1 โมเมนต์ลัพธ์ของระบบแรงที่อยู่ในระนาบเดียวกัน 8
รูปที่ 3.2 แรงคู่ควบ 9
รูปที่ 3.3 โมเมนต์ของแรงคู่ควบ 10
รูปที่ 3.4 แรง F และ f_s กระทำต่อวัตถุ 11
รูปที่ 3.5 กราฟระหว่างแรงเสียดทานกับแรงภายนอกกระทำต่อวัตถุ 12
รูปที่ 3.6 ตัวอย่างมาตรฐานสัญญาณพัลส์ 16
รูปที่ 3.7 บอร์ดขับมอเตอร์ ZX-SERVO 16 17
รูปที่ 3.8 แสดงวงจรสมบูรณ์ของ ZX-SERVO16 บอร์ดควบคุมเซอร์โวมอเตอร์ 16 ช่อง 18
รูปที่ 3.9 แสดงส่วนประกอบและการจัดขาของจุดต่อต่าง ๆ บนบอร์ด ZX-SERVO16 20
รูปที่ 3.10 ขั้วต่อเซอร์โวมอเตอร์ 20
รูปที่ 3.11 การจัดขาของไมโครคอนโทรลเลอร์เบอร์ต่างๆ 24
รูปที่ 3.12 บอร์ด CP-PIC V4.0 25
รูปที่ 3.13 แสดงโครงสร้างของบอร์ด CP-PIC V4.0 28
รูปที่ 3.14 แสดงสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์พิค 29
รูปที่ 3.15 สวิตช์เลือกโหมด RUN/PROG 30
รูปที่ 4.1 แสดงส่วนประกอบของเซอร์โวมอเตอร์ 33
รูปที่ 4.2 แสดงรูปแบบการเดินของรุ่นพี่ 35
รูปที่ 4.3 แสดงรูปแบบการเดินที่ปรับปรุงใหม่ 36
รูปที่ 4.4 แสดงการจำลองขาและตัวแปรต่างๆ 37
รูปที่ 4.5 แสดงการจำลองขาและตัวแปรต่างๆ 38
รูปที่ 4.6 แสดงการจำลองขาและตัวแปรต่างๆ 39
รูปที่ 4.7 กราฟที่ใช้พิจารณาขาส่วนที่ 3 40
รูปที่ 4.8 แสดงการหาขนาดตัวหุ่นและตัวแปรต่างๆ 40
รูปที่ 4.9 แสดงการหาขนาดตัวหุ่นและตัวแปรต่างๆ 41
รูปที่ 4.10 แสดงการทดลองควบคุมขาหุ่นหนึ่งข้างให้เขียนรูปตัว A 42
รูปที่ 4.11 แสดงรูปตัว A ที่เขียนโดยขาของหุ่นยนต์ที่สร้างขึ้น 43
รูปที่ 4.12 แสดงส่วนประกอบสำคัญที่ใช้ในการสร้างหุ่น 44
รูปที่ 4.13 แสดงการใช้จิ๊กช่วยในการเจาะรูในชิ้นส่วนของข้อต่อขา 45
รูปที่ 4.14 แสดงรูปข้อต่อที่ใช้ 45
รูปที่ 4.15 แสดงขาที่สมบูรณ์หนึ่งข้าง 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

รูปที่ 4.16 แสดงการใช้จิ๊กช่วยในการเจาะรูตัวหุ่น 46
รูปที่ 4.17 แสดงส่วนตัวที่พร้อมประกอบเข้ากับขาทั้งห้า 47
รูปที่ 4.18 แสดงส่วนประกอบทั้งหมดของตัวหุ่น 47
รูปที่ 4.19 แสดงการประกอบตัวหุ่น 48
รูปที่ 4.20 ตัวหุ่นที่เสร็จสมบูรณ์ 48
รูปที่ 4.21 แสดงอุปกรณ์ที่ใช้ในการสอบเทียบการตอบสนอง 49
รูปที่ 4.22 แสดงการสอบเทียบที่ 0 องศา 50
รูปที่ 4.23 แสดงการกำหนดมุม และตัวแปรต่างๆ 50
รูปที่ 4.24 แสดงมุมที่เราต้องการเพื่อใช้ในการหมุนตัว 52
รูปที่ 4.25 แสดงการหามุม β 52
รูปที่ 5.1 แสดงวงจรของบอร์ดควบคุมเซอร์โวมอเตอร์รุ่น ZX-SERVO16 54
รูปที่ 5.2 บอร์ต ZX-SERVO16 55
รูปที่ 5.3 แสดงลำดับของเซอร์โวมอเตอร์ที่ต้องต่อเข้าบอร์ด 56
รูปที่ 5.4 แสดงบอร์ดเร็กกูเลเตอร์ 56
รูปที่ 5.5 แสดงการเชื่อมต่อบนบอร์ด ZX-SERVO16 56
รูปที่ 5.6 การเชื่อมต่อไปยังพอร์ตอนุกรม ของคอมพิวเตอร์ 57
รูปที่ 5.7 หน้าต่างโปรแกรม Line80 57
รูปที่ 5.8 หน้าต่างการรันโปรแกรม 58
รูปที่ 6.1 แสดงหุ่นยนต์ต้นแบบห้าขาขนาดเล็กที่ใช้ทดสอบ 59
รูปที่ 6.2 แสดงคอมพิวเตอร์ที่ใช้ควบคุมหุ่น 60
รูปที่ 6.3 แสดงสายต่อพ่วงที่ใช้ควบคุมหุ่น และจ่ายไฟเลี้ยง 60
รูปที่ 6.4 แสดงพาวเวอร์ ซัพพลาย 61
รูปที่ 6.5 แสดงเรกกูเลเตอร์แปลงไฟจาก 12 โวล์ท เป็น 5 โวล์ท 61
รูปที่ 6.6 แสดงแบทเตอร์รี่ที่ใช้จ่ายไฟให้กับเซอร์โวมอเตอร์ 62
รูปที่ 6.7 แสดงสนามที่ใช้ในการทดสอบ 62
รูปที่ 6.8 แสดงการอ่านค่าผลการทดลอง 63
รูปที่ 6.9 แสดงเส้นทางการเดินเป็นเส้นตรงของหุ่นยนต์ 64
รูปที่ 6.10 แสดงผลการเดินของหุ่น 65
รูปที่ 6.11 แสดงผลจากการใช้โปรแกรม MINITAB หาสมการเส้นโค้ง 65
รูปที่ 6.12 แสดงเส้นทางการลักษณะการหมุนตัว 66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

รูปที่ 6.13 แสดงจุดศูนย์กลางของตัวหุ่นจากการหมุนรอบตัวเอง 68

รูปที่ 6.14 แสดงองศาที่หมุนได้เทียบกับค่าทางทฤษฎี 69

รูปที่ 6.15 แสดงรูปแบบการเดินเป็น สี่เหลี่ยมของหุ่นยนต์ 70

รูปที่ 6.16 แสดงการเปรียบเทียบตำแหน่งสั่งการ และ ตำแหน่งที่เดินได้จริง ของหุ่นยนต์ 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันหุ่นยนต์ได้เข้ามามีบทบาทสำคัญในชีวิตของเราไม่ว่าจะเป็นทางการศึกษา การแพทย์ งานบริการ หรือเทคโนโลยี เป็นต้น และที่สำคัญหุ่นยนต์ยังสามารถทำหน้าที่แทนมนุษย์ได้ในหลายๆเรื่อง ไม่ว่าจะเป็นพื้นที่ที่เสี่ยงต่ออันตราย ที่ที่เล็กเกินไปสำหรับมนุษย์ที่จะเข้าถึง ใช้ในการเฝ้าสำรวจตรวจตราในที่ที่ต้องดูแลตลอดเวลา เช่น ในพื้นสามจังหวังชายแดนภาคใต้ และพื้นที่เสี่ยงภัยในกรุงเทพฯ ในส่วนของการผลิตก็จะใช้หุ่นยนต์ในเรื่องของความแม่นยำ และความรวดเร็วในการผลิต ร่วมถึงการแก้ปัญหาเรื่องการขาดแรงงานที่มีฝีมือด้วย

และหากจะพูดถึงการเคลื่อนที่ของหุ่นยนต์ที่ทำงานบนพื้นดินก็จะสามารถแบ่งออกได้เป็น 2 แบบคือ

1. หุ่นยนต์ที่ใช้ล้อหรือตีนตะขาบในการเคลื่อนที่ ซึ่งมีความสามารถจำกัดในการเคลื่อนที่เข้าไปในบริเวณที่มีความลำบากมาก ไม่ค่อยได้

2. หุ่นยนต์ที่ใช้ขาในการเคลื่อนที่ ซึ่งมีความยืดหยุ่นมากกว่า ทั้งเรื่องรูปแบบ และลักษณะท่าทางของการเคลื่อนที่มากกว่าหุ่นยนต์ที่ใช้ล้อในการเคลื่อนที่ ดังนั้นจึงมีความสามารถที่ดีกว่าในการเคลื่อนที่เข้าไปในบริเวณที่มีความยากลำบาก

โครงงานสร้างต้นแบบและควบคุมหุ่นห้าขาจึงเป็นโครงงานที่มีความน่าสนใจเป็นอย่างมาก ทั้งในเรื่องรูปลักษณ์ และลักษณะการเดินของหุ่น เพื่อใช้เป็นต้นแบบสำหรับการศึกษา และพัฒนาให้สามารถใช้งานในด้านอื่นๆ ต่อไปในอนาคต

ในปัจจุบันเทคโนโลยีของการสร้างหุ่นยนต์ที่ใช้ขาในการเคลื่อนที่ นั้นหากแบ่งตามจำนวนขาสามารถแบ่งได้เป็น

1. หุ่นยนต์สองขา มีลักษณะการเคลื่อนที่แบบไดนามิค
2. หุ่นยนต์สี่ขา มีลักษณะการเคลื่อนที่ได้ 2 แบบ คือ แบบสะแตติค และแบบไดนามิค
3. หุ่นยตน์ห้าขา มีลักษณะการเคลื่อนที่ได้ 2 แบบ คือ แบบสะแตติค และแบบไดนามิค
4. หุ่นยนต์หกขาส่วนใหญ่จะเดินแบบสแตติค

1.2 วัตถุประสงค์ของโครงงาน

ในการสร้างหุ่นยนต์นั้น เราสามารถแบ่งเป็นส่วนที่สำคัญและน่าสนใจได้หลายส่วน เช่น การเคลื่อนที่ ระบบต้นกำลัง โครงสร้าง เป็นต้น และเนื่องจากโครงงานสร้าง และพัฒนาหุ่นยนต์ห้าขาตัวต้นแบบนี้ เป็นรุ่นที่ 2 ดังนั้นเราจึงให้ความสนใจกับ

1 เพื่อศึกษาหลักการทำงานเบื้องต้นของหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์

3 เพื่อศึกษาการเคลื่อนที่ของหุ่นยนต์โดยใช้เซอร์โวมอเตอร์เป็นต้นกำลัง

4 เพื่อศึกษาโปรแกรมควบคุมการทำงานของหุ่นยนต์

5 นำความรู้ทางด้านวิศวกรรมเครื่องกลและด้านต่างๆ มาประยุกต์กับโครงงาน

6 สร้างต้นแบบขนาดเล็กของหุ่นยนต์ห้าขา

7 เขียนโปรแกรมควบคุมท่วงท่า จังหวะการเดินและทิศทางของหุ่นยนต์

1.3 ขอบเขตของโครางงาน

เนื่องจากกลุ่มเรามีความสนใจในการศึกษา และออกแบบหุ่นยนต์ ซึ่งในศึกษา และการสร้างหุ่นนั้นสามารถแบ่งแยกย่อยออกไปได้หลายส่วน ซึ่งในโครงงานนี้ เราสนใจในส่วนของรูปแบบ และลักษณะการเดินของหุ่นดังนั้น โครงงานของเราจึงมีขอบเขตของโครงงานดังนี้

1 ออกแบบและสร้างหุ่น

2 ศึกษารูปแบบและท่วงท่าการเดินของหุ่นยนต์เพื่อนนำมาใช้ในการออกแบบลักษณะการเดินของหุ่นยนต์

3 สามารถควบคุมหุ่นยนต์ให้สามารถเดินไปในทิศทางตามที่ต้องการได้

1.4 วิธีการดำเนินโครงงาน

เพื่อใหเราทำงานได้อย่างมีประสิทธิภาพดังนี้เราจึงได้กำหนดวิธีและขั้นตอนการดำเนินงานเพื่อให้ การทำงานทเป็นไปได้ด้วยความราบรื่น

1 ศึกษาข้อมูลที่ได้มาจากรุ่นพี่และทำการปรับปรุงแบบ

2 ศึกษาเรื่องวัสดุเพื่อหาว่าวัสดุชนิดใดเหมาะสมที่จะนำมาใช้ทำหุ่นยนต์

3 ศึกษาข้อมูลของของมอเตอร์ที่จะนำมาใช้เป็นต้นกำลัง

4 ศึกษาจังหวะ ลักษณะ และท่าทางที่เหมาะสมสำหรับการเดินของหุ่น

5 สร้างหุ่นยนต์ตัวต้นแบบโดยมีวิธีการ จากการที่กลุ่มได้ทำการศึกษาโปรเจค การสร้างหุ่นยนต์ 5 ขา เราได้ทำการศึกษาในหัวข้อต่างๆ ดังนี้

5.1 หาข้อมูลและปัจจัยต่างที่มีผลต่อการออกแบบ แล้วทำการออกแบบและสร้างหุ่นยนต์ห้าขาขนาดเล็กที่สมบูรณ์

5.2 คำนวณหาองศาของเซอร์โวมอเตอร์แต่ละตัว จากตำแหน่งปลายขาของหุ่นยนต์

5.3 ทำการเทียบคำสั่งในโปรแกรม กับค่ามุมจริงที่เซอร์โวมอเดอร์หมุนไปแล้วเขียนโปรแกรมเพื่อควบคุมตำแหน่งของเซอร์โวให้ไปในองศาที่ต้องการ

5.4 สร้างขาต้นแบบมาหนึ่งข้างแล้วทดลองเขียนโปรแกรมควบคุม และเก็บข้อมูล

5.5 นำข้อมูลที่ได้จากต้นแบบมาพัฒนาและสร้างหุ่นยนต์จริง โดยใช้เซอร์โวมอเตอร์เป็นต้นกำลัง และใช้อลูมิเนียมเป็นโครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.6 ศึกษาหาวิธีการควบคุมที่เหมาะสม และเขียนโปรแกรมเพื่อใช้ในการควบคุมหุ่นยนต์

5.6.1 ควบคุมผ่านพอร์ตซีเรียลของคอมพิวเตอร์ โดยใช้โปรแกรม MATLAB

5.6.2 ควบคุมผ่านไมโครคอนโทรลเลอร์ PIC 18F458 ผ่านบอร์ด CP-PIC V4.0

5.6.3 ควบคุมผ่านพอร์ตซีเรียลของคอมพิวเตอร์ โดยใช้ภาษาซี

6 ศึกษาระบบควบคุม แล้วทำการเขียนโปรแกรมควบคุมหุ่น

7 ทดลองโปรแกรมและตัวหุ่นยนต์ แล้วทำการปรับปรุง

8 สรุปผลและวิจารณ์ผล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ในการทำโครงงานร่วมกันเป็นกลุ่ม ก่อให้เกิดการเรียนรู้ที่เป็นประโยชน์ต่อตัวของนักศึกษาในหลายๆ เรื่อง ดังนั้นผลที่คาดว่าจะได้รับมีดังนี้

1 เพิ่มพูลความรู้ทางวิชาการให้กับนักศึกษา

2 ฝึกให้นักศึกษาใช้เวลาว่างให้เป็นประโยชน์ โดยการศึกษาหาความรู้ด้วยตัวเอง

3 เรียนรู้วิธีการ และฝึกการใช้เครื่องมือต่างๆ ในการทำงานจริง

4 เพิ่มพูลประสบการณ์ และฝึกการทำงานร่วมกันเป็นกลุ่มของนักศึกษาภายในกลุ่ม

5 ได้หุ่นยนต์ต้นแบบตามจุดประสงค์ของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
แนวคิดในการสร้างและพัฒนาหุ่นยนต์

2.1 ความสำคัญของหุ่นยนต์

ในปัจจุบันหุ่นยนต์อุตสาหกรรมในประเทศต่างๆ ในโลกนี้เพิ่มจำนวนขึ้นมาอย่างรวดเร็ว สาเหตุที่มีการนำหุ่นยนต์มาใช้งานเป็นจำนวนมาก เพราะว่า หุ่นยนต์สามารถทำหน้าที่แทนมนุษย์ได้ในหลายๆ เรื่อง ไม่ว่าจะเป็นพื้นที่ ที่เสี่ยงต่ออันตราย พื้นที่เล็กเกินไป สำหรับมนุษย์ที่จะเข้าถึง ใช้ในการเฝ้าสำรวจตรวจตราในพื้นที่ ที่ต้องดูแลตลอดเวลา เช่น ในพื้นที่สามจังหวังชายแดนภาคใต้ และพื้นที่เสี่ยงภัยในกรุงเทพฯ ในส่วนของการผลิตที่ต้องใช้ความถูกต้องแม่นยำ และความรวดเร็วในการผลิตก็จะใช้หุ่นยนต์ รวมถึง การแก้ปัญหาเรื่องการขาดแคลนแรงงานที่มีฝีมือ จากข้อมูลในประเทศสหรัฐอเมริกา ยุโรป และญี่ปุ่น อัตราการเพิ่มของพลเมืองจะลดลง และประชากรที่อยู่ในวัยทำงานจะลดลงไปเรื่อยๆ ทำให้ขาดแคลนแรงงาน จำเป็นต้องใช้หุ่นยนต์อุตสาหกรรมมาช่วยในการทำงานในโรงงานอุตสาหกรรม และในยุคสมัยนี้ เทคโนโลยีที่ใช้ในการผลิดหุ่นยนต์มีการพัฒนาไปอย่างมาก ทำให้ประสิทธิภาพการผลิตของหุ่นยนต์ดีขึ้นอย่างมาก เป็นผลให้ขีดความสามารถของหุ่นยนต์ มีหดความหลากหลาย และมีประสิทธิภาพในการทำงานมากขึ้น พร้อมทั้งมีเทคโนโลยีสมองกล ที่ทันสมัยอยู่ภายใน สามารถทำงานได้หลายงานแทนมนุษย์ ร่วมทั้งคิด และวิเคราะห์สิ่งต่างๆ ได้เอง ซึ่งนับว่า เป็นสิ่งที่สามารถอำนวยความสะดวก และเพิ่มความปลอดภัยในการทำงานอย่างมีประสิทธิภาพของมนุษย์เรานั่นเอง และเป็นสิ่งจำเป็นที่มนุษย์เรา ต้องมีหุ่นยนต์ไว้ใช้งาน และในปัจจุบันมีหุ่นยนต์ใหม่ๆ เกิดขึ้นตลอดเวลาส่วนใหญ่จะเป็นหุ่นยนต์ที่ตอบสนองต่อความต้องการความจำเป็นของมนุษย์นั้นเอง

ภาระกิจ	ความสามารถของหุ่นยนต์
การสำรวจ	ช่วยในงานที่มนุษย์เข้าถึงได้ยาก เช่น ในป่าทึบ หุบเหว ฯลฯ
อวกาศ	สำรวจ และทำงานบนดาวเคราะห์ รวมถึงอวกาศด้วย
การทหาร	ช่วยในการตรวจตราในพื้นที่อันตราย ร่วมถึงการเฝ้าระวังภัย
ดับเพลิง	ค้นหาคน และทำงานในที่อันตราย
ไฟฟ้า	ค้นหาและทำงานในที่อันตราย หรือในสถานที่คนเข้าไปไม่ได้
สื่อสาร	ทำงานในพื้นที่อันตราย
ก่อสร้าง โยธา	ทดแทนแรงงานฝีมือที่ขาดแคลน ทำงานในที่อันตราย
การแพทย์	ช่วยงานพยาบาล ทดแทนแรงงานพยาบาล ให้กำลังใจผู้ป่วย
กำจัดขยะ	ทำงานในที่สกปรกและอันตราย
การศึกษาการวิจัย	ช่วยงานวิจัย หรือทำงานแทนคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขนส่งสินค้า	เพิ่มจำนวนผลผลิตของงาน และความรวดเร็วในการขนส่ง
ก๊าซ น้ำประปา	ทำงานโดยอัตโนมัติ ในที่ที่ไม่เหมาะต่องานของมนุษย์
การค้าขาย	เพิ่มผลผลิตในการค้า
การเกษตร	เพิ่มผลผลิต ทดแทนแรงงานที่ขาดแคลน ช่วยคนทำงาน
การบริการต่างๆ	ทดแทนแรงงานที่ขาดแคลน และช่วยในการเรียกลูกค้า

ตารางที่ 2-1 ความต้องการใช้หุ่นยนต์ในการปฏิบัติงานต่างๆ

เราคาดหวังที่จะใช้หุ่นยนต์ในสถานที่อันตรายมากๆ ในสภาพแวดล้อมที่เลวร้าย ที่คนเราไม่สามารถทนได้ เช่น ในสามจังหวัดชายแดนภาคใต้ เฝ้าระวังในพื้นที่เสี่ยงภัยตามจุดต่างๆในกรุงเทพฯ ในอวกาศ ทะเลลึก นอกจากนั้นในสถานที่อันตรายที่ไม่เหมาะกับมนุษย์เข้าไปทำงาน เช่น การตรวจสอบเตาปฏิกรณ์ปรมาณู โรงงานเคมี หรือสถานที่บางแห่งที่คนเข้าไปไม่ได้ หรือมีขนาดเล็ก เช่น ในท่อ

2.2 การเคลื่อนที่ของหุ่นยนต์โดยใช้ขา

กลไกการเคลื่อนไหวกลายมาเป็นสิ่งที่ขาดไม่ได้ ในการขยายพื้นที่การทำงานของหุ่นยนต์ การเคลื่อนที่เดินสองขาเหมือนคน การเดินสองขาจะต้องใช้เทคโนโลยีการควบคุมขั้นสูง ถ้าเป็นบริเวณที่ขรุขระหรือมีสิ่งกีดขวางขนาดใหญ่อยู่มาก การเดินแบบหลายขาเป็นวิธีที่เหมาะสม ยิ่งขาที่ใช้ยาวเท่าไรก็สามารถข้ามสิ่งกีดขวาง หลุมบ่อ หรือพื้นเป็นขั้นได้อย่างสบาย การรักษาสมดุลตลอดเวลาการเดินด้วยขาสองขา หรือสามขาไม่ใช่เรื่องง่าย แต่ถ้ามีจำนวนขามากกว่านั้น แม้ขาบางขาจะยกลอยอยู่แต่ถ้ามีขามากกว่าสามขาแตะอยู่กับพื้นก็จะสามารถรักษาสมดุลได้

การเดินโดยใช้ขา	การเดิน 2 ขา
	การเดิน 3 ขา
	การเดิน 4 ขา
	การเดิน 5 ขา (กำลังศึกษาและพัฒนาอยู่ขณะนี้)
	การเดิน 6 ขา
	การเดิน 8 ขา
การเดินไม่ใช้ขา	เช่น หุ่นยนต์มีล้อ หุ่นยนต์งู
แบบผสม	เช่น ล้อ+ขาหลายขา

ตารางที่ 2-2 ประเภทของกลไกการเคลื่อนที่ที่ใช้กับหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รูปแบบการเดินของหุ่นยนต์โดยใช้ขา

การเคลื่อนที่ด้วยการเดินโดยใช้ขานั้น กลไกที่ใช้สำหรับการเคลื่อนที่มักจะมีองศาความอิสระ (Degree of Freedoms) จำนวนมากเพื่อให้มีอิสระในการเคลื่อนที่สูง และสามารถเคลื่อนที่ไปบนพื้นที่ขรุขระมากๆ ได้ โดยทั่วไปต้องมีอุปกรณ์ในการตรวจสภาพพื้นล่วงหน้าเพื่อช่วยในการวางเท้าลงสัมผัสพื้น ระบบควบคุมจะมีความซับซ้อนมากตามไปด้วย เนื่องจากต้องควบคุมข้อต่อและอุปกรณ์ตรวจรู้ต่างๆ จุดเด่นของการเคลื่อนที่แบบนี้คือสามารถเคลื่อนที่บนพื้นที่ขรุขระมากๆ ได้ดีมาก รูปแบบการเดินของหุ่นยนต์โดยใช้ขามี 2 ชนิด คือ

2.3.1 การเดินแบบสถิตย์

การเดินลักษณะนี้จะรักษาสมดุลไว้ตลอดเวลา เห็นได้ทั่วไปจากการเดินสองขาของหุ่นยนต์ของเล่น ลักษณะเด่นคือ แม้จะหยุดการเคลื่อนไหวในขณะที่ยกขาขึ้นขาหนึ่ง ก็ยังสามารถยืนอยู่ได้โดยไม่เสียการทรงตัว

2.3.2 การเดินแบบจลน์

วิธีการเดินนี้มีเจตนาที่ให้เสียสมดุลไปในระหว่างการถ่ายน้ำหนัก เมื่อเสียสมดุลไปจะใช้การก้าวขาออกไปข้างหนึ่งทำให้กลับมามีสมดุล ซึ่งในระหว่างที่ก้าวขาออกไปจะไม่มีการรักษาสมดุล ดังนั้น จึงไม่สามารถหยุดนิ่งระหว่างที่ก้าวขาได้อย่างการเดินแบบสถิต

2.4 การพิจารณาเลือกจำนวนขาของหุ่นยนต์

เมื่อเปรียบเทียบกับการเดินสองขาแล้ว การเดินหลายขาโดยเฉพาะด้วยขามากกว่าสี่ขา จะมีความมั่งคงแตกต่างกันมาก สำหรับการเดินสองขาหากไม่ใช้การเดินแบบจลน์ จะนำหุ่นยนต์ไปใช้งานจริงได้ยาก แต่การเดินด้วยขามากกว่าสี่ขา แม้จะเป็นการเดินแบบสถิตก็สามารถทำให้เคลื่อนที่ได้รวดเร็วพอที่จะใช้งานได้จริง ที่เป็นเช่นนี้เพราะว่าการทรงตัวสามารถทำได้ด้วยขาสามขา ในการเดินด้วยขาหลายขาจึงสามารถใช้ขาอื่นนอกจากสามขานั้นในการก้าวเคลื่อนที่ได้อย่างง่ายดาย นั้นคือ ใช้ขาสามขาวางกับพื้นเพื่อรักษาสมดุลไว้ แล้วใช้ขาที่เหลือในการขยับเคลื่อนที่ การเดินเป็นการเดินแบบสถิต ดังนั้น การควบคุมจึงทำได้ง่ายกว่าเมื่อเทียบกับการเดินแบบจลน์

2.4.1 ข้อดีของการเดินหลายขา

การเดินด้วยขาเมื่อเปรียบเทียบกับการวิ่งด้วยล้อ นั้นมีความซับซ้อนเพิ่มขึ้นจึงควบคุมยากขึ้นด้วย แต่ในพื้นที่ทุรกันดาร หรือในกองซากปรักหักพัง ในการเคลื่อนที่ด้วยล้อดินตะขาบทำได้ยากหุ่นยนต์ที่เดินหลายขาถูกคิดค้นมาเพื่อใช้ทำงานในที่อันตรายที่ยากที่มนุษย์จะเข้าไปถึง นอกจากนี้ในพื้นที่ที่ไม่มีส่วนพื้นราบเรียบเลยก็ตาม หุ่นยนต์ที่เดินหลายขาสามารถที่จะเคลื่อนที่ไปได้ โดยใช้ขาก้าวข้ามสิ่งกีดขวางนั้นไป จึงเคลื่อนที่ผ่านสิ่งกีดขวางนั้นได้ นอกจากนี้ยังมีความสามารถหลายอย่างแฝงอยู่ในการเดินหลายขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนขา	ปัจจัยแสดงการเลือกขา					รวม
	สมดุล	น้ำหนัก	พลังงาน	ราคา	พลังงาน	
4	x	3	3	3	3	x
5	2	2	2	2	2	10
6	3	1	1	1	1	7

หมายเหตุ3 = ดี 2 = ปานกลาง 1 = ไม่ดี x = ยอมรับไม่ได้

ตารางที่ 2-3 ตารางเปรียบเทียบจำนวนขาเพื่อทำการสร้างหุ่นยนต์ [1]

ในการเปรียบเทียบจำนวนขา เพื่อทำการสร้างหุ่นยนต์นั้น ปัจจัยในการเลือกขาเพื่อการปฏิบัติภารกิจนี้ต้องการหุ่นยนต์ที่มีการเดินแบบสถิตเพื่อรับประกันความมั่นคงในการก้าวเดิน แม้ว่าหุ่นยนต์จะเกิดอุบัติเหตุจนถึงกับต้องหยุดเดินก็จะไม่เกิดความเสียหาย ดังนั้นหุ่นยนต์ที่มีขา 4 ขาซึ่งมีการเดินเป็นแบบจลน์นั้นจึงไม่เหมาะสมกับภารกิจนี้ แม้ว่าความเหมาะสมในด้านอื่นๆจะมีมากกว่าหุ่นยนต์ 5 ขา และ6 ขา เมื่อพิจารณาความเหมาะสมจากหุ่นยนต์ที่เหลือนั้นพบว่า หุ่นยนต์ที่มีขา 5 ขานั้นมีความเหมาะสมในการสร้างมากที่สุด

2.5 แนวคิดการพัฒนาหุ่นยนต์ห้าขาในอนาคต

จากการที่เราได้สังเคราะห์การเดินของหุ่นยนต์ห้าขาได้แล้ว (ที่จะกล่าวในปริญญานิพนธ์เล่มนี้) เราได้มีแนวคิดในการพัฒนาหุ่นยนต์ต่อไป ดังต่อไปนี้

แนวคิดที่ 1

หุ่นยนต์อาจจะเปลี่ยนระบบต้นกำลังใหม่ เช่น เป็นระบบไฮโดรริก เพื่อที่หุ่นยนต์จะได้มีความสามารถในด้านของกำลังมากขึ้น โดยใช้ประโยชน์จากรูปแบบการเดินที่สร้างขึ้นในปริญญานิพนธ์เล่มนี้

แนวคิดที่ 2

เขียนโปรแกรมเพิ่มทักษะความสามารถในการเคลื่อนที่ให้มีรูปแบบการเดินที่หลากหลายเพื่อให้เหมาะกับสถานะการณ์ต่างๆ

แนวคิดที่ 3

การติดตั้งอุปกรณ์เพิ่มเติม เพื่อเพิ่มประสิทธิภาพของหุ่น เช่น กล้องอินฟาเรด หรืออุปกรณ์รับ-ส่งคลื่นโซน่า บนตัวของหุ่นยนต์ เพื่อให้หุ่นยนต์มีความสามารถในการสำรวจและรับรู้สภาพพื้นที่รอบตัวหุ่น

แนวคิดที่ 4

เพิ่มขนาดของโซโวมอเตอร์ให้มีทอร์กมากขึ้นเพื่อที่หุ่นยนต์จะได้มีความสามารถในการขึ้นทางต่างระดับ ขึ้นพื้นเอียง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ทฤษฎีและหลักการ

3.1 ทฤษฎีโมเมนต์ของแรง (Moment)

โมเมต์ (Moment) ของแนวรอบจุดหรือแกนเป็นการวัดแนวโน้มของวัตถุว่าจะหมุนรอบจุดหรือหมุนรอบแกนไปทิศทางใด ผลคูณระหว่างขนาดของระยะทางจากจุดหมุนไปตั้งฉากกับแนวของแรง เรียกกับว่า โมเมนต์ของแรง (Moment) หรือ ทอร์ก (Torque) และเรียกจุดที่ตรึงอยู่กับที่ว่า จุดหมุน (Fulcum)

โมเมนต์ลัพธ์ของระบบแรงอยู่ในระนาบ $X-Y$ ดังนั้น โมเมนต์ที่เกิดจากแรงรอบจุดหมุน O ที่มีทิศตามแกน Z ดังรูปที่ 3.1 ดังนั้น โมเมนต์ลัพธ์ M_{RO} ของระบบหาได้โดยการบวกหรือลบโมเมนต์ของแต่ละแรงตามทิศทางการหมุนรอบแกน เนื่องจากเวคเตอร์โมเมนต์ทั้งหมดอยู่ในแนวเดียวกัน ผลรวมเวคเตอร์สามารถเขียนในรูป

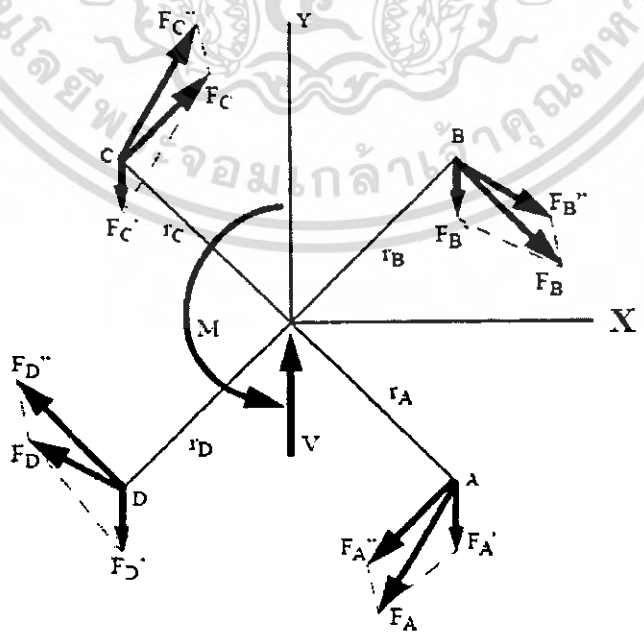
$$M_{RO} = \sum Fd \qquad (3\text{-}1)$$

เมื่อ

M เป็นโมเมนต์ของแรง

F หน่วยเป็นนิวตัน – เมตร

d เป็นระยะทางจากจุดหมุนไปตั้งฉากกับแนวแรง F

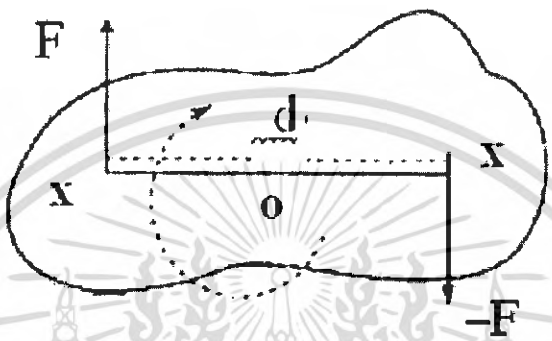


รูปที่ 3.1 โมเมนต์ลัพธ์ของระบบแรงที่อยู่ในระนาบเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1. โมเมนต์ของแรงคู่ควบ (Moment of a Couple)

แรงคู่ควบ (Couple) คือ แรงสองขนานกันและมีขนาดเท่ากัน โดยมีทิศทางตรงข้ามกันและห่างกัน โดยระยะตั้งฉาก d ดังรูปที่ 3.2 เนื่องจากแรงลัพธ์ของแรงสองแรงที่ประกอบเป็นแรงคู่ควบมีค่าเท่ากับศูนย์ ผลของแรงคู่ควบมีค่าเท่ากับศูนย์ แต่ผลของแรงคู่ควบจะทำให้เกิดการหมุนในทิศทางตามแนวของแรงคู่ควบ



รูปที่ 3.2 แรงคู่ควบ

โมเมนต์ที่เกิดจากแรงคู่ควบ เรียกว่า โมเมนต์ของแรงคู่ควบ (Couple Moment) มีค่าเท่ากับผลรวมของโมเมนต์ของแรงคู่ควบทั้งคู่รอบจุด O ในอาณาบริเวณใด ๆ สำหรับการพิสูจน์เวคเตอร์ระบุตำแหน่ง r_A และ r_B ทิศทาง O ไปยังจุด A และ B ที่อยู่บนแนวการกระทำของ –F และ F ดังรูปที่ 3.3 โมเมนต์คู่ควบรอบ O ดังนั้น

$$M = r_A(-F) + r_B(F) \tag{3-2}$$

$$M = (r_B - r_A)F \tag{3-3}$$

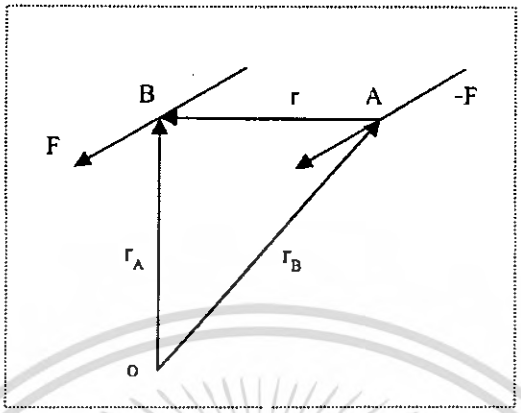
จากกฎรูปสามเหลี่ยมของการบวกเวคเตอร์ $r_A + r = r_B$ หรือ $r = r_B - r_A$ จะได้ว่า

$$M = rxF \tag{3-4}$$

ผลดังกล่าวบ่งบอกว่าโมเมนต์แรงคู่ควบเป็นเวคเตอร์อิสระ (Free Vector) นั่นคือ เป็นเวคเตอร์กกระทำที่จุดใด ๆ ก็ได้และสามารถเคลื่อนย้ายไปที่จุดใดก็ได้เนื่องจาก M ขึ้นอยู่กับเวคเตอร์ระบุตำแหน่งที่มีทิศระหว่างแรงทั้งสอง ซึ่งไม่ใช่ว่าเวคเตอร์ระบุตำแหน่ง และที่มีทิศทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากจุด O ไปยังแรง หลักการนี้ไม่เหมือนโมเมนต์ของแรงที่มีจุดหรือ แกนของโมเมนต์แน่นอนที่จะต้องคำนวณหาโมเมนต์รอบจุดหรือแกนที่แน่นอนนั้น



รูปที่ 3.3 โมเมนต์ของแรงคู่ควบ

รูปสมการสเกลาร์ (Scalar Formulation) โมเมนต์ของแรงคู่ควบ M ถูกนิยามให้มีขนาด

$$M = Fd \tag{3-5}$$

เมื่อ F คือ ขนาดของแรง

d คือ ระยะตั้งฉาก หรือแขนโมเมนต์ระหว่างแรง

ทิศทางของโมเมนต์แรงคู่ควบหาได้จากกฎมือขวา โดยนิ้วหัวแม่มือแสดงทิศทาง ในขณะที่นิ้วมือทั้งสี่ที่กำเป็นทิศการหมุนที่เกิดจากแรงทั้งสอง ในทุกกรณี M กระทำตั้งฉากกับระนาบที่มีแรงคู่ควบบรรจุอยู่

รูปสมการเวคเตอร์ (Vector Formulation) โมเมนต์ของแรงคู่ควบแสดงเป็นผลคูณเวคเตอร์ (Vector Cross Product) โดยใช้สมการ

$$M = rxF \tag{3-6}$$

การประยุกต์ใช้นี้ง่ายต่อการจดจำ ถ้ามีแนวคิดที่จะหาโมเมนต์ของแรงทั้งคู่รอบจุดที่อยู่บนแนวแกนการกระทำของแรงใดแรงหนึ่งของทั้งคู่ เช่น ถ้าต้องการคำนวณหาโมเมนต์รอบจุด A โมเมนต์ของ $-F$ มีค่าเป็นศูนย์รอบจุดนั้น และโมเมนต์ของ F นิยามจากสมการ $M = rxF$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แรงเสียดทาน (Friction Force)

แรงเสียดทานแบ่งออกเป็น 2 แบบ คือ

3.2.1 แรงเสียดทานสถิตย์ (Static friction)

แรงเสียดทานสถิตย์เขียนแทนด้วยสัญลักษณ์ f_s เป็นแรงเสียดทานที่เกิดขึ้นระหว่าง พื้นผิวสัมผัส ขณะที่วัตถุหยุดนิ่ง หรือวัตถุจะเริ่มเคลื่อนที่ โดยขณะที่วัตถุจะเริ่มเคลื่อนที่จะเรียกแรงเสียดทานขณะนั้นว่า แรงเสียดทานที่มากที่สุด (Limiting Friction) เขียนแทนด้วยสัญลักษณ์ $f_{s(\max)}$ จากการทดลองพบว่าแรงเสียดทานสถิดจะมีค่าไม่คงที่

3.2.2 แรงเสียดทานจลน์ (Kinetic friction)

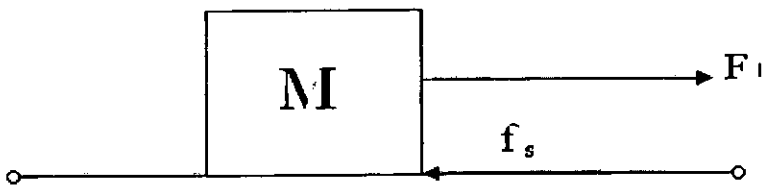
แรงเสียดทานจลน์เขียนแทนด้วยสัญลักษณ์ f_k เป็นแรงเสียดทานที่เกิดขึ้นระหว่างพื้นผิวสัมผัสในขณะที่วัตถุเคลื่อนที่ด้วยความเร็วคงที่ แต่ในการคำนวณ อนุโลมว่า วัตถุเคลื่อนที่ด้วยความเร่งก็ได้ ซึ่งจากการทดลอง พบว่าแรงเสียดทานจลน์จะมีค่าคงที่เสมอ

แรงเสียดทานสถิต และแรงเสียดทานจลน์ เป็นแรงเสียดทานระหว่างผิววัตถุที่สัมผัสกัน เช่น ผิวแท่งเหล็กกับผิวพื้นโต๊ะ จากการสังเกตในชีวิตประจำวันจะพบว่า **แรงเสียดทานจะขึ้นอยู่กับลักษณะ และชนิดของผิวสัมผัสกัน โดยไม่ขึ้นอยู่กับขนาด และรูปร่างของพื้นที่ผิว ที่สัมผัสกัน และแรงเสียดทานจลน์จะไม่ขึ้นกับความเร็วสัมพัทธ์ ของวัตถุที่เลื่อนผ่านกัน** เช่น ถ้าผิวสัมผัสชนิดเดียวกันเป็นผิวหยาบหรือขรุขระ แรงเสียดทานจะมีค่ามาก แต่ถ้าเป็นผิวเกลี้ยง หรือลื่น แรงเสียดทานมักจะมีค่าน้อย ส่วนผิวสัมผัสชนิดเดียวกันซึ่งหยาบมาก จะมีแรงเสียดทานมาก เป็นต้น

เมื่อวัตถุอยู่นิ่งขนาดของแรงเสียดทานสถิตจะมีค่าเท่ากับขนาดของแรงเสียดทานฉุดขณะนั้น แต่มีทิศทางตรงกันข้าม นอกจากนั้นจะพบว่า ขนาดของแรงเสียดทานสถิตจะมีค่าเพิ่มขึ้นตามแรงที่ฉุดเพิ่มขึ้น ซึ่งขนาดของแรงเสียดทานสถิตจะเพิ่มขึ้นมากที่สุด ก็ต่อเมื่อวัตถุเริ่มจะเคลื่อนที่ และขณะที่แรงฉุด มีค่าเป็นศูนย์ จะทำให้แรงเสียดทานสถิดมีค่าเป็นศูนย์ด้วย

จึงสรุปได้ว่าแรงเสียดทานสถิตมีค่าไม่คงที่ โดยมีค่าเปลี่ยนแปลงได้ตั้งแต่ศูนย์กระทั่งถึงแรงเสียดทานสถิตที่มีค่ามากที่สุด

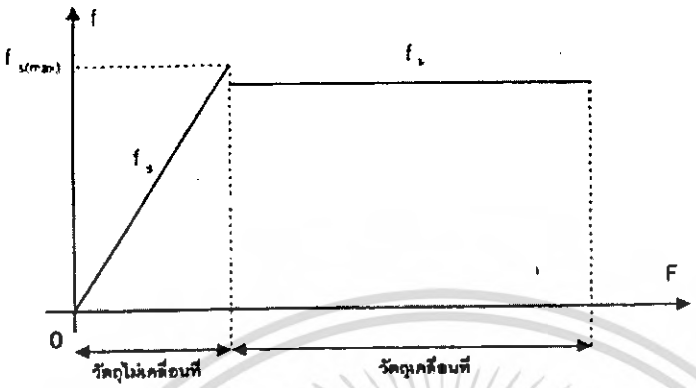
ซึ่งเขียนสมการได้ว่า $f_s = F$ ที่ฉุดขณะนั้น



รูปที่ 3.4 แรง F และ f_s กระทำต่อวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองนี้ เมื่อนำขนาดของแรงที่ฉุดกับขนาดของแรงเสียดทานที่เกิดขึ้นมาเขียนกราฟ จะได้ดังรูปที่ 3.4



รูปที่ 3.5 กราฟระหว่างแรงเสียดทานกับแรงภายนอกกระทำต่อวัตถุ

จากกราฟ จะสรุปได้ว่า

1 แรงเสียดทานสถิตจะมีค่าไม่คงที่ โดยมีค่าแปรเปลี่ยนได้ตั้งแต่ศูนย์ถึงแรงเสียดทานสถิตสูงสุด

2 แรงเสียดทานสถิตจะมีค่าสูงสุด (Limiting friction) เมื่อวัตถุเริ่มไถล หรือเริ่มเคลื่อนที่

3 แรงเสียดทานสถิตอาจมีค่าน้อยกว่า เท่ากับ หรือมากกว่าแรงเสียดทานจลน์ก็ได้ แต่แรงเสียดทานสถิตสูงสุดจะมากกว่าแรงเสียดทานจลน์เสมอ

4 แรงเสียดทานจลน์เกิดขณะที่วัตถุเคลื่อนที่ และจะมีค่าคงที่เสมอ

3.2.3 สัมประสิทธิ์ความเสียดทาน (Coefficient of friction)

จากการทดลองดึงวัตถุซึ่งวางอยู่บนพื้น ซึ่งมีแรงเสียดทาน จะพบจากการทดลองว่า สำหรับวัตถุคู่หนึ่งแรงเสียดทานที่มากที่สุด ($f_{s(\max)}$) และแรงเสียดทานจลน์ (f_k) จะเป็นส่วนโดยตรงกับแรงที่พื้นผิวสัมผัสกระทำต่อวัตถุในแนวตั้งฉากกับวัตถุ (Normal force) ซึ่งจะเท่ากับน้ำหนักของวัตถุที่วางบนพื้นราบ

ซึ่งจากการทดลองจะเขียนเป็นความสัมพันธ์ได้ดังนี้

$$f \alpha N \tag{3-7}$$

หรือ

$$f = \mu N \tag{3-8}$$

เมื่อ μ เป็นค่าคงตัวของการแปรผัน และเรียกว่า **สัมประสิทธิ์ความเสียดทาน** ซึ่งมีค่าคงที่สำหรับวัตถุคู่ใด ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีของแรงเสียดทานที่เกิดขึ้นขณะที่วัตถุเริ่มจะเคลื่อนที่ ซึ่งเป็นแรงเสียดทานสถิตที่มีค่าสูงสุด คือ $f_{s(\max)}$ ค่าสัมประสิทธิ์ที่ได้ จะเรียกว่า สัมประสิทธิ์ความเสียดทานสถิต เขียนแทนด้วยสัญลักษณ์ μ_s ซึ่งจะได้ความสัมพันธ์ว่าว่า

$$f_{s(\max)} = \mu_s N \tag{3-9}$$

ส่วนแรงเสียดทานที่เกิดขึ้นขณะที่วัตถุเคลื่อนที่ ซึ่งเรียกว่า แรงสียดทานจลน์ ค่าสัมประสิทธิ์ที่ได้ จะเรียกว่า สัมประสิทธิ์ความเสียดทานจลน์ เขียนแทนด้วยสัญลักษณ์ μ_s ซึ่งจะได้ความสัมพันธ์ว่า

$$f_k = \mu_k N \tag{3-10}$$

จากสมการที่ได้จะเห็นว่า ถ้าทราบแรงเสียดทานสถิต f_s แรงเสียดทานจลน์ f_k และแรงที่พื้นกระทำต่อวัตถุในแนวตั้งฉาก N เราจะทราบสัมประสิทธิ์ความเสียดทานสถิต μ_s และสัมประสิทธิ์ความสียดทานจลน์ μ_k ได้ โดยสัมประสิทธิ์ความเสียดทานนี้จะไม่มีหน่วย เพราะเป็นอัตราส่วนของแรง

ค่าสัมประสิทธิ์ความเสียดทานสถิต μ_s และสัมประสิทธิ์ความเสียดทานจลน์ μ_k หาได้จากการทดลอง ซึ่งจะมีค่าขึ้นอยู่กับชนิดของผิวสัมผัส อุณหภูมิ และสารปนเปื้อน (Contaminant) ระหว่างผิวสมผัส ดังตารางที่ 3.1

ผิวสัมผัสระหว่าง	สัมประสิทธิ์ ความเสียดทานสถิต μ_s	สัมประสิทธิ์ ความเสียดทานจลน์ μ_k
ยางกับคอนกรีต (แห้ง)	1.00	0.80
ยางกับคอนกรีต (เปียก)	0.30	0.25
ล้อยางกับถนน (แห้ง)	0.90	0.65
ล้อยางกับถนน (เปียก)	0.70	0.55

ตารางที่ 3-1 สัมประสิทธิ์ความเสียดทานสถิต μ_s และสัมประสิทธิ์ความเสียดทานจลน์ μ_k

จากตารางพบว่า สำหรับผิวสัมผัสคู่หนึ่ง สัมประสิทธิ์ความเสียดทานสถิต μ_s จะมากกว่าสัมประสิทธิ์ความเสียดทานจลน์ μ_k เสมอ ($\mu_s > \mu_k$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 สมดุล

จากกฎการเคลื่อนที่ข้อที่ 1 ของนิวตัน เราทราบแล้วว่าวัตถุ จะไม่มีการเปลี่ยนสภาพการเคลื่อนที่ คือ วัตถุจะรักษาสภาพหยุดนิ่ง หรือยังคงรักษาสภาพเคลื่อนที่ด้วยความเร็วคงที่ต่อไปได้ นอกจากจะมีแรงลัพธ์ ที่ไม่เป็นศูนย์มากระทำต่อวัตถุ จึงทำให้วัตถุเคลื่อนที่ด้วยความเร่ง

ในกรณีที่วัตถุเดิมหยุดนิ่ง และยังคงหยุดยิ่งต่อไปไม่มีการเปลี่ยนสภาพการเคลื่อนที่ เราเรียกว่า วัตถุนั้นใน สภาพสมดุลสถิต เช่น หนังสือวางบนโต๊ะ รถจอดอยู่ข้างถนน เป็นต้น

สำหรับกรณีที่มีวัตถุที่มีการเคลื่อนที่เป็นเส้นตรงด้วยความเร็วคงที่ รถยนต์แล่นบนพื้นถนนด้วยอัตราเร็วคงที่เท่าเดิม เราจะเรียกว่าวัตถุนั้นอยู่ในสภาพ สภาพสมดุลจลน์ เช่น เรือลอยไปตามด้วยอัตราเร็วคงที่ รถยนต์แล่นบนพื้นถนนด้วยอัตราเร็วคงที่ เป็นต้น นอกจากนี้สมดุลจลน์ยังรวมถึงสมดุลของวัตถุเคลื่อนที่ โดยมีสภาพการหมุน คือ มีการหมุนรอบแกนซึ่งวางตัวในแนวเดิมด้วยอัตราการหมุนคงที่ เช่น รอก กว้าน ล้อ ที่หมุนรอบแกนหมุนซึ่งวางตัวในแนวเดิมด้วยอัตราการหมุนคงที่

ดังนั้น เงื่อนไขที่ทำให้วัตถุสมดุลต่อการเคลื่อนที่ คือ แรงลัพธ์ที่กระทำต่อวัตถุต้องเป็นศูนย์ ซึ่งเขียนสมการได้ คือ

$\sum F = 0$ ซึ่งเมื่อแยกพิจารณาแต่ละแนวจะได้ว่า

แนวราบ $\sum F_x = 0$ คือ แรงซ้าย เท่ากับ แรงขวา

แนวดิ่ง $\sum F_y = 0$ คือ แรงขึ้น เท่ากับ แรงลง

และ การหมุน $\sum M = 0$ คือ วัตถุไม่มีการหมุน หรือหมุนคงที่

การคำนวณสมดุลต่อการเคลื่อนที่เนื่องจากมีแรงภายนอกมากระทำต่อวัตถุ แยกพิจารณาเงื่อนไขของแรงต่าง ๆ ที่กระทำต่อวัตถุได้ 3 แบบ ดังนี้

1 การสมดุลของแรงตัดที่จุดเดียวกัน ส่วนใหญ่ได้แก่ การสมดุลของแรงสามแรง จะได้ว่า ถ้าแรงสามแรงกระทำต่อวัตถุ แล้ววัตถุอยู่ในสภาวะสมดุล เมื่อต่อแนวแรงทั้งสามมันจะตัดกันที่จุดเดียวกันเสมอ

2 การสมดุลของวัตถุในระบบที่ประกอบด้วยวัตถุหลายก้อน มีหลักการพิจารณาดังนี้

2.1 ถ้าวัตถุในระบบวิ่งด้วยอัตราเร็วเท่ากัน หาค่าที่ต้องการจากสมการผลรวมแรงในระบบเท่ากับศูนย์

2.2 ถ้าวัตถุในระบบที่วิ่งด้วยอัตราเร็วไม่เท่ากัน หาค่าที่ต้องการโดยการตัดวัตถุออกมาคิดแรงทีละก้อนตามสมการ

$$\sum F = 0 \qquad (3\text{-}11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 การสมดุลของแรงที่ตัดกันคนละจุด เพื่อความสะดวกและความรวดเร็วในการคำนวณได้ดังนี้

3.1 หยิบวัตถุที่ต้องการหาแรงออกมาเขียนแรงกระทำต่าง ๆ ให้ครบ

3.2 เลือกตรงจุดที่ไม่ทราบค่าแรงมากที่สุด

3.3 คำนวณหาค่าที่ต้องการจากสมการ

แนวราบ $\sum F_X = 0$ (3-10)

แนวดิ่ง $\sum F_Y = 0$ (3-11)

และ การหมุน $\sum M = 0$ (3-12)

4. ต้องการหาแรงปฏิกริยาที่ใด ให้รวมกันแบบเวคเตอร์จะได้

ขนาดของแรง $R = \sqrt{\left(R_X^2 + R_Y^2\right)}$ (3-13)

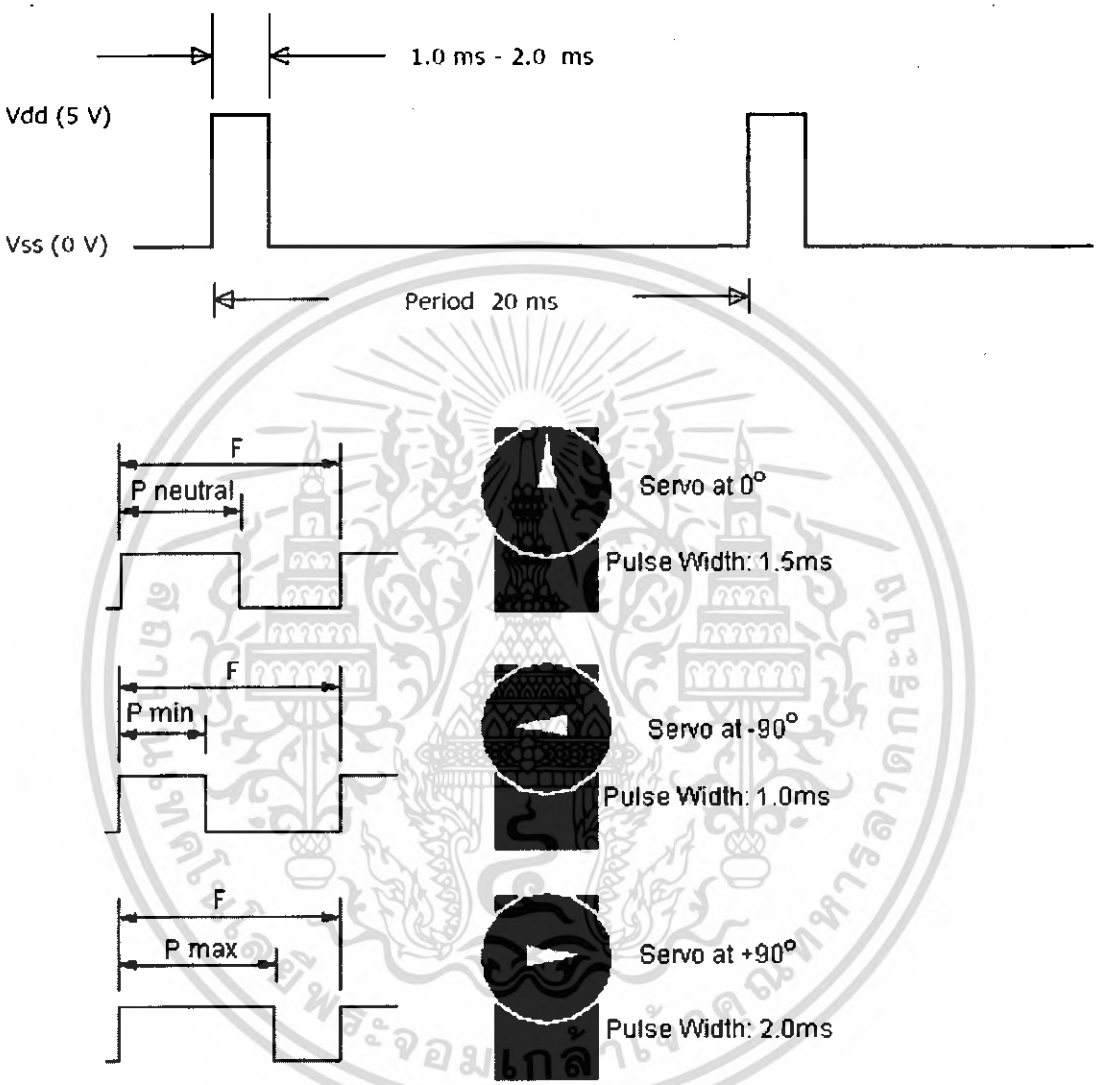
ทิศทาง $\tan\alpha = \frac{R_Y}{R_X}$ (3-14)

3.4 หลักการทำงานของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์คือ มอเตอร์ไฟฟ้ากระแสตรง (DC motor) ที่ถูกประกอบรวมกับ ชุดเกียร์ และส่วน ควบคุม ต่างๆ ไว้ในโมดูลเดียวกัน หรือ ภายในกล่องพลาสสติกเดียวกัน โดยมอเตอร์ชนิดนี้จะมีสายต่อเพื่อใช้งานเพียง 3 เส้นเท่านั้น คือ VCC, GND และสายสัญญาณควบคุม (Control Line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือขวาได้จากสายสัญญาณเพียงเส้นเดียว โดยสัญญาณที่ใช้ควบคุมนี้จะเป็นสัญญาณ พัลส์วิดมอดคูเลเตอร์ (PWM) แบบ TTL Level ระดับแรงดันที่จ่ายให้มอเตอร์นี้จะอยู่ในช่วงประมาณ 4 ถึง 6 V ขึ้นอยู่กับคุณสมบัติของมอเตอร์แต่ละตัว ข้อดีของมอเตอร์ชนิดนี้ก็คือ จะมีขนาดเล็กน้ำหนักเบา, ให้แรงบิดสูง, กินพลังงานน้อย และสามารถควบคุม ด้วยแรงดันลอจิกที่เป็น TTL ได้โดยตรงไม่จำเป็นต้องต่อวงจรขับ (Driver) อื่นๆ เพราะมอเตอร์ชนิดนี้จะมีวงจรควบคุมบรรจุไว้ภายในอยู่แล้ว ซึ่งมอเตอร์ชนิดนี้สามารถควบคุมให้หมุนไปในตำแหน่ง หรือทิศทางองศาที่ต้องการได้ โดยอาศัยสัญญาณความกว้างพัลส์ ที่ป้อนให้มอเตอร์ แต่เซอร์โวมอเตอร์นี้จะหมุนได้เพียงในช่วงประมาณ 180 องศา หรือ ครึ่งรอบเท่านั้น หรือบางรุ่นอาจหมุนได้ถึง 210 องศา แต่จะไม่สามารถหมุนเป็นวงรอบได้เนื่องจากโครงสร้างภายในจะประกอบด้วย ตัวต้านทานชนิดปรับค่าได้ (VR) ที่ทำหน้าที่ตรวจสอบตำแหน่งการหมุนของมอเตอร์ และ ตัวต้านทานนี้จะถูกยึดติดกับแกนหมุนของมอเตอร์ ซึ่งจากการที่ตัวต้านทานปรับค่านี้ไม่สามารถหมุนเป็นวงรอบได้ ดังนั้นเซอร์โวมอเตอร์จึงถูกออกแบบให้หมุนได้เพียงแค่ประมาณ 180 องศา หรือครึ่งรอบเท่านั้น เพื่อป้องกันความเสียหายที่จะเกิดกับตัวต้านทานปรับค่าได้ แต่ถ้าหากเราต้องการให้มอเตอร์หมุนเป็นวงรอบ (360 องศา) นั้นก็สามารถทำได้ โดยจะต้องทำการปรับแต่ง (Modify) ดัดแปลงชิ้นส่วนบางอย่างของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการทำงานของ เซอร์โวมอเตอร์ ทำได้โดย การป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ซึ่งตำแหน่งและทิศทางการหมุนของมอเตอร์นี้จะขึ้นอยู่กับขนาดของความกว้างของพัลส์นั้นๆ โดยทั่วไปแล้วความกว้างของสัญญาณพัลส์จะมีจุดให้อ้างอิง 3 จุด ดังรูป คือ



รูปที่ 3.6 ตัวอย่างมาตรฐานสัญญาณพัลส์ [2]

- สัญญาณความกว้างพัลส์ขนาด 1.5 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม 0 องศา หรือ จุดกึ่งกลางของมอเตอร์

- สัญญาณความกว้างพัลส์ขนาด 1 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม - 90 องศา หรือในทิศทางทวนเข็มนาฬิกา

- สัญญาณความกว้างพัลส์ขนาด 2 ms จะควบคุมให้เซอร์โวมอเตอร์หมุนไปอยู่ที่ตำแหน่งมุม + 90 องศา หรือในทิศทางตามเข็มนาฬิกา

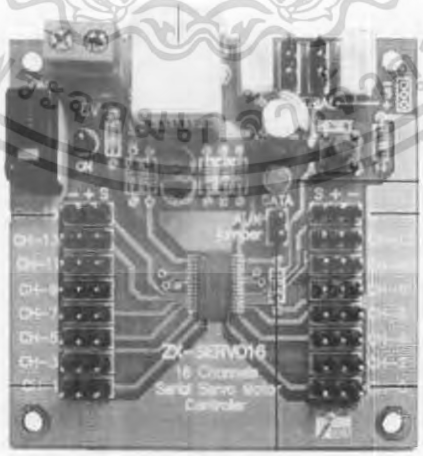
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หมายเหตุ ค่าความกว้างพัลส์ และ ระยะองศาการหมุนของมอเตอร์ที่อธิบายด้านบน นั้นเป็นเพียงค่าประมาณเท่านั้น ทั้งนี้ระยะการหมุน และขนาดของพัลส์ที่ควบคุมการทำงานของมอเตอร์ในแต่ละยี่ห้ออาจจะไม่เท่ากัน ดังนั้นในการใช้งาน จึงควรศึกษารายละเอียดของมอเตอร์ในแต่ละรุ่นที่นำมาใช้ ซึ่งโดยปกติแล้วรายละเอียดต่างๆ ของมอเตอร์มักจะมีติดมากับตัวมอเตอร์นั้นๆ อยู่แล้ว

ส่วนการที่จะควบคุมให้มอเตอร์หมุนเป็นมุมขนาดต่างๆ นั้นก็สามารถทำได้โดยการป้อนสัญญาณพัลส์เป็นระดับความกว้างต่างๆ โดยอ้างอิงจากจุด ทั้ง 3 จุดที่กล่าวมานี้ ตัวอย่างเช่น ถ้าต้องการให้มอเตอร์หมุนไปที่มุม -45 องศา เราก็จะต้องป้อนสัญญาณพัลส์ที่มีความกว้าง 1.25 ms เป็นต้น และสัญญาณพัลส์นี้จะต้องจ่ายให้มอเตอร์ทุกๆ 20 ms (Period) เพื่อรักษาสภาพตำแหน่งของมอเตอร์ไว้โดยหลักการก็คือ จะอาศัยการเปรียบเทียบช่วงเวลาของความกว้างพัลส์ ที่จ่ายให้กับมอเตอร์ทางขาสัญญาณควบคุม กับค่าเวลาของวงจร RC ภายในบอร์ดควบคุมในตัวของมอเตอร์ ซึ่งค่าเวลาของวงจร RC นี้ จะมีการเปลี่ยนแปลงตามการหมุนของมอเตอร์ เนื่องจากตัวต้านทานปรับค่าจะถูกยึดติดอยู่กับแกนหมุนของมอเตอร์ ซึ่งการหมุนของมอเตอร์จะทำให้ค่าความต้านทานของตัวต้านทานปรับค่า (VR) เปลี่ยนแปลงไป เป็นผลทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงตามไปด้วย โดยในขณะที่เราป้อนสัญญาณความกว้างพัลส์ให้กับมอเตอร์ทางขาสัญญาณควบคุม สัญญาณนี้จะถูกนำไปเปรียบเทียบกับค่าเวลาของวงจร RC หากค่าทั้ง 2 ไม่เท่ากัน มอเตอร์ก็จะหมุนทำให้ค่าเวลาของวงจร RC เปลี่ยนแปลงจนกระทั่งค่าเวลาความกว้างพัลส์ของ วงจร RC เปลี่ยนแปลงจนเท่ากับสัญญาณพัลส์ทางขาควบคุม (Control line) มอเตอร์จึงจะหยุดหมุน

3.5. บอร์ดขับมอเตอร์ ZX-SERVO 16 [3]



รูปที่ 3.7 บอร์ดขับมอเตอร์ ZX-SERVO 16

ZX-SERVO 16 เป็นบอร์ดควบคุมเซอร์โวมอเตอร์ 16 ตัวได้ในเวลาเดียวกัน ซึ่งมีคุณสมบัติดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
72848

1. ติดต่อกับไมโครคอนโทรลเลอร์ได้ทุกตระกูลด้วยการสื่อสารข้อมูลแบบอนุกรม โดยใช้สายสัญญาณเพียงเส้นเดียว

2. สามารถเลือกและเปลี่ยนบอดเรต 2400 บิตต่อวินาที และ 38400 บิตต่อวินาทีได้ โดยผ่านซอฟต์แวร์

3. สามารถควบคุมผ่านคอมพิวเตอร์ ผ่านพอรต์อนุกรม RS- 232 ได้ โดยใช้ซอฟต์แวร์ PSCI :Parallax Servo Controller Interface[4] โดยสามารถ ดาวน์โลดได้ที่ www.inex.co.th หรือ www.parallax.com

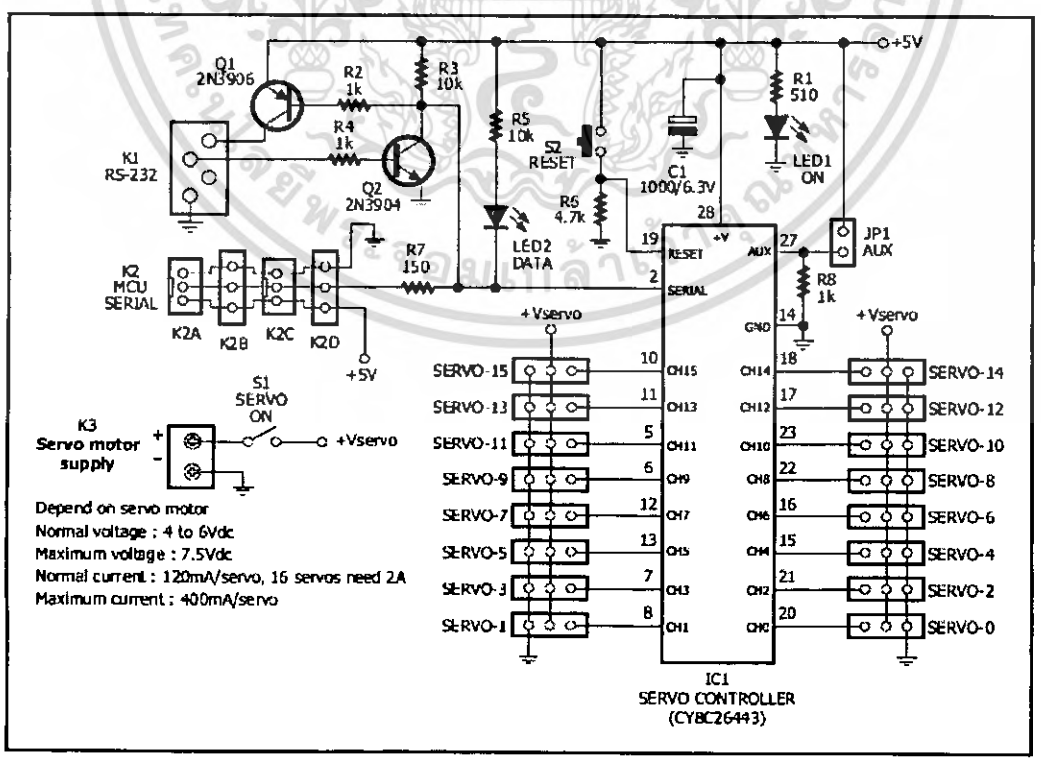
4. ควบคุมเซอร์โวมอเตอร์ทั้ง 16 ตัว ได้ในเวลาเดียวกัน และกำหนดองศาในการหมุนของเซอร์โวมอเตอร์ได้ระหว่าง 0 ถึง 180 องศา

5. กำหนดความเร็วในการเคลื่อนที่ของเซอร์โวมอเตอร์แต่ละตัวได้ โดยกำหนดการไล่ลำดับเวลา (Ramping) ตั้งแต่ 0.75 ถึง 60 วินาที

6. แจ้งตำแหน่งปัจจุบันของเซอร์โวมอเตอร์ในแต่ละตัว เมื่อส่งคำสั่งร้องขอจากผู้ใช้

7. ความละเอียดในการควบคุมตำแหน่งสูง โดยใช้วงจรพัลส์วิดธ์มอดดูเลเตอร์ (PWM) ความละเอียด 16 บิต เพื่อควบคุมเซอร์โวมอเตอร์ให้เคลื่อนที่ 0 ถึง 180 องศา โดยมีความละเอียด 2 ไมโครวินาทีต่อสเต็ป

8. สามารถติดต่อพวงกันเพื่อขยายการควบคุมเซอร์โวมอเตอร์สูงสุด 32 ตัวได้ในเวลาเดียวกัน



รูปที่ 3.8 แสดงวงจรสมบูรณ์ของ ZX-SERVO16 บอร์ดควบคุมเซอร์โวมอเตอร์ 16 ช่อง [3]

เอกสารนี้ได้มาจากแหล่งที่ให้ใช้ได้ทั้งไม่ได้ให้ใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 การทำงานของวงจร

บอร์ดควบคุมเซอร์โวมอเตอร์ ZX-SERVO16 มีไมโครคอนโทรลเลอร์ PSoC เบอร์ CY8C26443 ที่บรรจุโปรแกรมควบคุมพิเศษอยู่ภายในหลักการทำงาน สื่อสารรับส่งข้อมูลกับไมโครคอนโทรเลอร์ภายนอกหรือคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม RS-232 โดยใช้สายสัญญาณเพียงเส้นเดียว ซึ่งสายสัญญาณนี้ทำหน้าที่ทั้งรับและส่งข้อมูล ดังนั้นเพื่อไม่ให้เกิดการชนกันของข้อมูลจะต้องมีการขับสถานะเอาต์พุตพียงสถานะเดียวเท่านั้น คือ สถานะลอจิก "0"

ในสภาวะปกติ ขา Serial (ขา 2) จะมีสถานะลอจิก "1" จากตัวต้านทาน R3 ที่ต่อพูลอัปเอาไว้ LED2 DATA จะติดสว่างก็ต่อเมื่อ การรับส่งข้อมูลลอจิก "0" ระหว่างไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์ภายนอกกับบอร์ด ZX-SERVO16

<u>ข้อควรระวัง</u> การเชื่อมต่อ ขาสัญญาณอนุกรมเข้ากับไมโครคอนโทรลเลอร์ ต้องระมัดระวังในการเขียนโปรแกรมจะต้องกำหนดไม่ให้มีการป้อนลอจิก "1" จากไมโครคอนโทรลเลอร์ภายนอกเข้ามาเป็นอันขาด เนื่องจากอาจทำให้ไมโครคอนโทรลเลอร์ภายนอกที่นำมาต่อร่วมด้วย และไมโครคอนโทรลเลอร์ CY8C26443 บนบอร์ด ZX-SERVO16 เสียหายได้ เนื่องจากเกิดการชนของข้อมูลที่เป็นเอาต์พุตทั้งคู่ ถึงแม้ว่า จะมีการต่อตัวต้านทานอนุกรมกับสายสัญญาณ เพื่อป้องกันในระดับหนึ่งแล้วก็ตาม

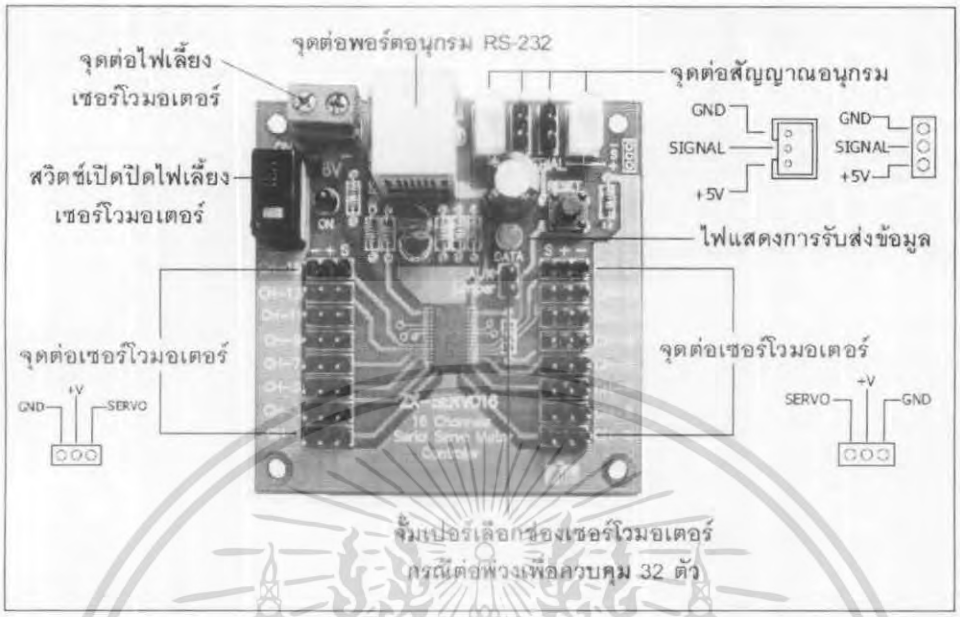
บอร์ด ZX-SERVO16 สามารถต่อพ่วงกันได้ 2 บอร์ด เพื่อขับเซอร์โวมอเตอร์รวม 32 ตัว ผ่านทางจุดต่อ K2 ซึ่งทำไว้ 2 แบบคือ แบบ PCB 2 มม. และ IDC 2.5 มม. จึงต่อกับบอร์ดควบคุมหุ่นยนต์มชในรุ่น Stamp - BOX, RBX – 877, RBX – 51AC2 และ AX – 11 ของบริษัทบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์จำกัดได้ทันที หรือจะต่อกับบอร์ดไมโครคอนโทรเลอร์ของผู้ใช้งานเองก็ได้ เมื่อต่อพ่วงบอร์ ZX SERVO – 16 เข้าด้วยกัน จะต้องมีการตำแหน่งช่องของเซอร์โวมอเตอร์ให้ต่างกัน โดยต่อจั๊มเปอร์ที่ตำแหน่ง AUX ถ้าเอาจั๊มเปอร์ออก ตำแหน่งช่องบนบอร์ด ZX SERVO – 16 ตัวนั้นๆ จะเป็นช่อง CH0 ถึง CH15 ถ้าต่อจั๊มเปอร์ลงไป ตำแหน่งช่องจะเป็น CH16 ถึง CH31 จะต้องกำหนดให้แตกต่างกันจึงจะสามารถใช้คำสั่งระบุตัวเซอร์โวมอเตอร์ได้ตั้งแต่ CH0 ถึง CH15

บอร์ด ZX SERVO – 16 จะแยกแหล่งจ่ายไฟให้กับเซอร์โววมอเตอร์และไมโครคอนโทรเลอร์ CY8C26443 ออกจากกัน โดยแหล่งจ่ายไฟ +5 V สำหรับ CY8C26443 สามารถใช้ร่วมกับบอร์ดไมโครคอนโทรลเลอร์ที่นำมาต่อใช้งานด้วย ส่วนแหล่งจ่ายไปสำหรับเซอร์โวมอเตอร์ต้องต่อเข้าที่เทอร์มินอลตำแหน่ง K3 บนบอร์ด แรงดันที่ใช้ขับมอเตอร์ควรมีค่าระหว่าง 4 ถึง 7.5 V ส่วนความสามารถในการจ่ายกระแสไฟฟ้าขึ้นอยู่กับจำนวนมอเตอร์ที่นำมาต่อด้วย ปกติเซอร์โวมอเตอร์ 1 ตัวต้องการกระแสไฟฟ้าประมาณ 100 mA ในขณะทำงาน โดยแรงดันที่ใช้ขับมอเตอร์จะมีสวิตช์สำหรับตัดต่อด้วย

บนบอร์ด ZX SERVO – 16 มีไฟแสดงสถานะ 2 ตัวโดย LED สีแดงใช้แสดงสถานะการจ่ายไฟเลี้ยง +5V ให้กับวงจรส่วน LED2 DATA ใช้แสดงการรับส่งข้อมูลกับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 ขั้นตอนการใช้งาน



รูปที่ 3.9 แสดงส่วนประกอบและการจัดขาของจุดต่อต่าง ๆ บนบอร์ด ZX-SERVO16

1. เลื่อนตำแหน่งสวิตช์ ON บนบอร์ด ZX-SERVO-16 ไปที่ตำแหน่ง OFF

2. คลายสกรูของเทอร์มินอลบล็อกสีฟ้าในตำแหน่ง +V motor แล้วต่อสายไฟบวกและสายไฟลบจากแหล่งจ่ายไฟภายนอกเพื่อเป็นไฟเลี้ยงให้กับเซอร์โวมอเตอร์เข้าไป ต้องระมัดระวังการต่อสายไฟ จะต้องต่อให้ถูกต้องตรงขั้ว

3. เสียบสายเซอร์โวมอเตอร์ที่ต้องการควบคุมเข้ากับจุดต่อเซอร์โวมอเตอร์ จะต้องสังเกตทิศทางของสายให้ถูกต้องตรงขั้ว

สายสีดำเป็นไฟลบ (-) หรือกราวด์

สายีแดงเป็นไฟบวก (+)

สายสีขาวเป็นสายสัญญาณ (s)

รูปที่ 3.10 ขั้วต่อเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ต่อสายสัญญาณ และไฟเลี้ยงเข้ากับไมโครคอนโทรเลอร์ที่นำมาควบคุมบอร์ด ZX-SERVO16 โดยต้องไล่ลำดับสายให้ถูกต้อง ซึ่งต่างจากสายของเซอร์โวมอเตอร์

5. จ่ายไฟให้กับไมโครคอนโทรเลอร์ที่ใช้ควบคุม จะเห็น LED ที่ตำแหน่ง ON บนบอร์ด ZX-SERVO16 ติดไฟสว่างทันทีที่จ่ายไฟ

3.5.3 การเขียนโปรแกรมติดต่อกับบอร์ด ZX-SERVO16

3.5.3.1 รูปแบบการสื่อสารข้อมูลอนุกรม เนื่องจากบอร์ด ZX-SERVO16 ใช้สายสัญญาณเพียงเส้นเดียวในการรับและส่งข้อมูล ดังนั้นจะต้องกำหนดให้ไมโครคอนโทรลเลอร์ใช้โหมดในการสื่อสารอนุกรมแบบคอลเล็กเตอร์เปิด โดยปกติการส่งข้อมูลลอจิก "1" จะเป็นการส่งระดับแรงดัน +5 V ออกไป ส่วนที่เป็นลอจิก "0" จะส่งระดับแรงดัน 0 V ออกไป <u>แต่สำหรับการส่งข้อมูลแบบคอลเล็กเตอร์เปิด</u> ในสภาวะลอจิก "1" จะเป็นการปล่อยสถานะขาให้ลอยไว้ และให้เกิดสภาวะลอจิก "1" จากตัวต้านทานพูลอัปแทน เพื่อป้องกันการชนกันของข้อมูล

3.5.3.2 การกำหนดค่าบอดเรตสำหรับบอร์ด ZX-SERVO16 บอร์ด ZX-SERVO16 สามารถกำหนดค่าบอดเรตได้ 2 ค่าคือ 2,400 และ 38,400 บิตต่อวินาที แต่เมื่อเปิดจ่าไฟครั้งแรก ค่าบอดเรตของบอร์ด ZX-SERVO16 มีค่าเท่ากับ2,400 บิตต่อวินาที ซึ่งเป็นค่าตั้งต้น ผู้ให้งานสามารถเปลี่ยนค่าบอดเรต และตรวจสอบค่าบอดเรตปัจจุบันได้จากการส่งคำสั่ง "!SCSBR"

การขึ้นต้นคำสั่งเพื่อติดต่อบอร์ด ZX-SERVO16 นั้นจะต้องขึ้นต้นด้วยเครื่องหมาย "!" และตามด้วยอักษร "$OD" หรือ CR หลังจากส่งคำสั่งเรียบร้อยแล้ว คำสั่งที่มีการส่งข้อมูลกลับจะมีการส่งข้อมูลกลับ หลังจากหน่วงเวลาไป 1.5 มิลลิวินาทีแล้ว

การเปลี่ยนค่าบอดเรดกลับไปเป็น 2,400 บิตต่อวินาทีอีกครั้งทำได้โดยการรีเซตบอร์ด ZX-SERVO16 ด้วยการกดสวิตช์ RESET หรือส่งค่า "! SCSBR", O, CR ออกไป

3.5.3.3 คำสั่งสำหรับกำหนดตำแหน่งของเซอร์โวมอเตอร์

<u>รูปแบบคำสั่ง</u> "! SC" C R pw. LOWBYTE, pw.HIGHBYTE, $OD

สำหรับการควบคุมเซอร์โวมอเตอร์ ผู้ใช้งานจะต้องส่งคำส่างระบุตำแหน่งไปยังบอร์ด ZX-SERVO16 แต่ละคำสั่งจะประกอบด้วย

ส่วนหัวของคำสั่ง คือการส่งอักขระ "! SC" ออกไป แล้วตามด้วยค่าพารามิเตอร์ 3 ตัวได้แก่ ค่า C, R และ pw และจบด้วยอักขระปิดท้าย $OD

C คือ ช่องของเซอร์โวมอเตอร์ที่ต้องการติดต่อ มีค่าตั้งแต่ 0 ถึง 31 โดยเซอร์โวมอเตอร์ในตำแหน่ง 0 ถึง 15 เป็นช่องของบอร์ด ZX-SERVO16 ที่ไม่ได้ต่อจั๊มเปอร์ AUX ไว้ ส่วนตำแหน่ง 16 ถึง 31 เป็นช่องของบอร์ด ZX-SERVO16 ที่ต่อจั๊มเปอร์ AUX ไว้

R คือ ค่าความชันของการปรับเปลี่ยนความกว้างพัลส์ที่ป้อนให้กับเซอร์โวมอเตอร์แต่ละตัว เพื่อให้แกนของเซอร์โวมอเตอร์หมุนไปยังตำแหน่งที่กำหนดด้วยเวลาที่แตกต่างกันกำหนดค่าได้ตั้งแต่ 0 ถึง 63 ถ้าเท่ากับ 0 จะไม่มีการกำหนดค่าระดับความชัน เซอร์โวมอเตอร์จะหมุนไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งที่กำหนดทันทีด้วยความเร็วสูงสุด การกำหนดค่า 1 ถึง 63 จะทำให้เซอร์โวมอเตอร์หมุนไปยังตำแหน่งที่ต้องการภายในเวลา 0.75 ถึง 60 วินาที

pw คือค่าข้อมูล 16 บิตสำหรับระบุตำแหน่งของเซอร์โวมอเตอร์ มีค่า 250 ถึง 2,500 สำหรับกำหนดตำแหน่งองศาของเซอร์โวมอเตอร์ 0 ถึง 180 องศาให้ความละเอียดในการกำหนดค่าแต่ละสเต็ปเท่ากับ 2 ไมโครวินาที

หมายเหตุ เซอร์โวมอเตอร์แต่ละตัวมีคุณสมบัติไม่เหมือนกันทุกประการ ถ้ากำหนดค่า 250 แล้ว แกนของเซอร์โวมอเตอร์ไม่เคลื่อนที่ไปยังตำแหน่ง 0 องศา ให้ทดลองปรับเปลี่ยนเป็นค่า 255 หรือ 260 แล้วทดลองดูผลการทำงาน เช่นเดียวกับการกำหนดค่า 1,250 อาปรับเปลี่ยนค่าได้เล็กน้อยเพื่อให้ได้ตำแหน่ง 180 องศาอย่างถูกต้อง

3.5.3.4 การสร้างคำสั่งเพื่อระบุตำแหน่งของเซอร์โวมอเตอร์

รูปแบบคำสั่ง "! SCRSP" ch $OD

ตัวอย่างคำสั่งที่ตอบกลับมา "ch positional position2"

คำสั่งนี้จะคืนค่าที่สั่งอ่านค่า และค่าตำแหน่งปัจจุบันออกมา โดยผลลัพธ์ที่นำมาแสดงจะแจ้งหมายเลขช่องที่ขับเซอร์โวมอเตอร์เป็นตัวเลขฐานสิบหก ซึ่งผู้ใช้สามารถเปลี่ยนค่าให้เป็นค่าตัวเลขฐนสิบเพื่อดูค่าตำแหน่งได้ เช่น 04:CA หมายถึงค่า 1226

3.5.3.5 การควบคุมบอร์ด ZX-SERVO16 ผ่านคอมพิวเตอร์

การสื่อสารระหว่างบอร์ด ZX-SERVO16 กับคอมพิวเตอร์จะทำผ่านพอร์ตอนุกรม โดยต่อสาย CX-4 ดังแสดงในรูปที่ 4 จะต้องจ่ายไฟเลี้ยง +5 V ให้กับบอร์ด ZX-SERVO16 ด้วย จากรูปที่ 4 ใช้ไฟเลี้ยงผ่านมาจากบอร์ด Stamp-BOX ซึ่งไฟเลี้ยง +5 V นี้จะใช้เฉพาะส่วนที่ควบคุมบอร์ด ZX-SERVO16 เท่านั้น สำหรับไผเลี้ยงเซอร์โวมอเตอร์ยังคงต้องจ่ายตามปรกติ

ซอฟแวร์ที่ใช้ในการควบคุมนี้คือ PSCI หรือ Parallax Controller Interface สามารถดาวน์โลดได้ที่ www.inex.co.th หรือ www.parallax.com จะได้เป็นไฟล์สำหรับติดตั้ง ให้ทำหารติดตั้งโปรแกรมให้เรียบร้อยก่อน

จากนั้นทำการเปิดโปรแกรม PSCI เลือกพอร์ตอนุกรมที่เชื่อมต่อ โดยไปที่เมนู File>Select Comm Port แล้วเลือกพอร์ดต่ออนุกรมที่เชื่อมต่อ หน้าต่างแสดงสถานะของโปรแกรมจะแสดงข้อความ Opening Comm Port และ Comm Port Open ตามลำดับ เพื่อแสดงการเปิดพอร์ดอนุกรมตำแหน่งที่กำหนด เพื่อสื่อสารข้อมูล

3.6 ไมโครคอนโทรเลอร์ (Microcontroller)

ปัจจุบันเครื่องใช้ไฟฟ้าอุปกรณ์อิเล็กทรอนิกส์ตลอดจนระบบโรงงานอุตสาหกรรมจะใช้ไมโครคอนโทรลเลอร์ซึ่งติดตั้งอยู่ภายในเป็นตัวควบคุมเกือบทั้งหมด ทำให้ไมโครคอนโทรลเลอร์กลายเป็นอีกหนึ่งอุปกรณ์ที่ผู้ผลิตสารกึ่งตัวนำหลาย ๆ บริษัทให้ความสนใจ และมีการแข่งขันสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาก อุปกรณ์ที่ใช้ไมโครคอนโทรลเลอร์ควบคุมมรมากมายหลายชนิ เช่น อุปกรณ์เครื่องใช้ภายในบ้าน สัญญาณไฟจราจร รถยนต์ ตลอดจนระบบอุตสาหกรรม PLC, CNC, Robot เป็นต้น

ปัจจุบันไมโครคอนโทรลเลอร์ PIC16F877 ซึ่งเป็นไมโครคอมพิวเตอร์แบบชิปเดี่ยว (Single Chip Microcontroller) ถูกใช้งานอย่างแพร่หลาย การศึกษาเกี่ยวกับไมโครคอนโทรลเลอร์จึงมีองค์ประกอบหลายอย่างเช่นสถาปัตยกรรมของไมโครคอนโทรลเลอร์ วงจรอิเล็กทรอนิกส์ ตลอดจนภาษาที่ใช้ในการเขียน ซึ่งจะต้องศึกษาควบคู่กันไป โดยจะนำเสนอลำดับการทำงานดังต่อไปนี้

3.6.1 โครงสร้างของไมโครคอนโทรลเลอร์ (PIC)

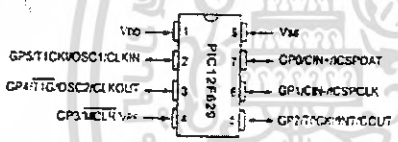
Microcontroller PIC มีโครงสร้างหลายลักษณะ ทั้ง 8 pins , 14 pins, 18 pins และ 40 pins ซึ่งมีให้เลือกใช้มากมายขึ้นอยู่กับว่าจะเลือกใช้ขนาด Memory เท่าใดให้เหมาะสมกับลักษณะงานที่ทำ โดยสามารถแบ่งหน่วยความจำได้ 3 แบบ คือ (1) หน่วยความจำโปรแกรมแฟลช (FLASH Program Memory) มีขนาดความจุ 1 ถึง 32 (2) หน่วยความจำข้อมูล (Data Memory RAM) มีความจุถึง 256 bytes Microcontroller PIC16F877 มีพอร์ตใช้งานทั้งสิ้น 5 พอร์ต คือ PORTA 6 บิท, PORTB 8 บิท, PORTC 8 บิท, PORTD 8 บิท, และ PORTE 3 บิท เป็นพอร์ตแบบมี 2 ทิศทาง คือ สามารถเป็นได้ทั้งอินพุท และเอาท์พุท และยังเป็นพอร์ตที่สามารถแปลงสัญญาณ ADC (Analog to Digital Converter) ได้อีกด้วย Microcontroller PIC มีคุณลักษณะดังต่อไปนี้

- มีพอร์ต I/O ขนาด 3 บิท, 6บิท, 8 บิท จำนวน 5 พอร์ต
- มีหน่วยความจำโปรแกรมแบบแฟลช 1 ถึง 32 kwords
- มีหน่วยความจำข้อมูลแรม RAM 64 ถึง 1536 bytes
- มีหน่วยความจำข้อมูลอีอีพรอม EEPROM 256 bytes
- Timer / Counter
    - 0 ขนาด 8 บิท
    - 1 ขนาด 16 บิท
    - 2 ขนาด 8 บิท
- มีกระแสซิงก์และกระแสซอร์ส (High Sink/Source Current) 25 mA
- มีวงจรแปลงสัญญาณ Analog to Digital Converter 10 บิท
- มีวงจรสื่อสารแบบอนุกรมทั้ง SPI และ I^2C (Master/Slave)
- มีวงจร Pulse Width Modulation (PWM) ความละเอียดสูงถึง 10 บิท
- มีหน่วยความจำแบบ Flash
- สามารถเขียนและลบได้มากกว่า 100,000 ครั้ง

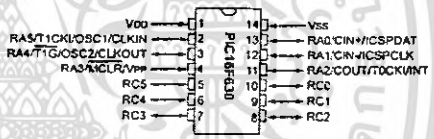
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบอร์ (Device)	หน่วยความจำโปรแกรม (Program Memory)	หน่วยความจำข้อมูล (Data Memory)		อินพุท/เอาต์พุท (I /O Port)	ไทเมอร์ / เคาน์เตอร์ (Timer /Counter)
	Flash Memory (Kwords)	RAM (bytes)	EEPROM (bytes)		
PIC12F629	1	64	128	8	2
PIC16F628	2	224	128	18	3
PIC16F873	4	192	128	28	3
PIC16F877	8	368	256	40	3
PIC18F242	16	768	256	28	4
PIC18F452	32	1,536	256	40	4
PIC18F458	32	1,536	256	40	4

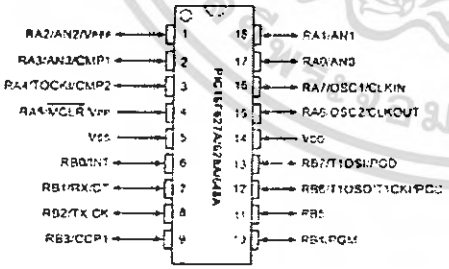
ตารางที่ 3.2 ไมโครคอนโทรลเลอร์พิค



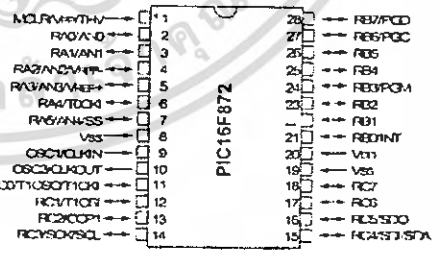
Microcontroller PIC 8 pins



Microcontroller PIC 14 pins



Microcontroller PIC 18 pins



Microcontroller PIC 28 pins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

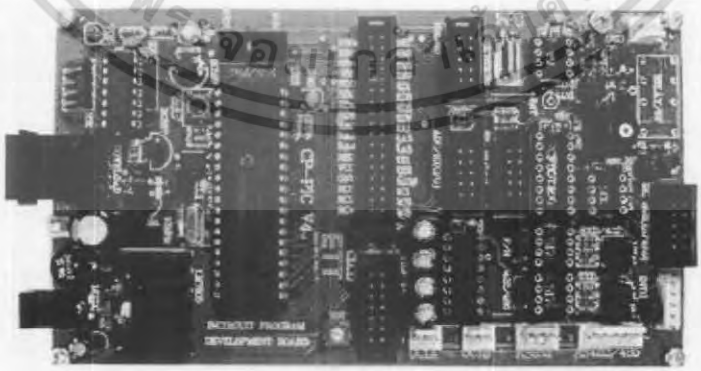
รูปที่ 3.11 การจัดขาของไมโครคอนโทรลเลอร์เบอร์ต่างๆ

ในตัว CPU ของ Microcontroller PIC16F877 ขนาด 40 Pins มีขาสัญญาณที่ใช้ติดต่อ Input/Output Port มีจำนวนขาสัญญาณทั้งสิ้น 33 pins ดังนี้

- PORTA RA0_RA5 จำนวน 6 Pins
- PORTB RA0_RB7 จำนวน 8 Pins
- PORTC RC0_RC7 จำนวน 8 Pins
- PORTD RD0_RD7 จำนวน 8 Pins
- PORTE RE0_RE2 จำนวน 3 Pins

ขาและสัญญาณแต่ละขาของ Microcontroller PIC16F877 มีหน้าที่การทำงานดังนี้

3.7 บอร์ดไมโครคอนโทรเลอร์ CP-PIC V4.0[5]



รูปที่ 3.12 บอร์ด CP-PIC V4.0

ไมโครคอนโทรเลอร์เป็นอุปกรณ์พิเศษที่รวมเอาซีพียู, หน่วยความจำ และพอร์ตเข้าด้วยกัน โดยบรรจุรวมกันอยู่ภายในตัวถังเดียวกัน สำหรับไมโครคอนโทรเลอร์ที่มีความสามารถสูงๆ บาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบอร์จะรวมเอาวงจรสร้างฐานเวลาจริง หรือรีไทม์คล็อก (RTC), วงจรตรวจสอบจังหวะการทำงานหรือวอต์ชด็อก (Watch Dag), วงจรตั้งเวลาหรือไทเมอร์ (Timer), วงจรนับ (Counter) และวงจรแปลงสัญญาณอะนาล็อกเป็นดิจิตอลเข้าไว้ด้วยกัน ไมโครคอนโทรเลอร์จะสามารถทำงานได้เมื่อจ่ายไฟเลี้ยงและต่อวงจรกำเนิดนาฬิกาให้แก่มันจากนั้นซีพียูในไมโครคอนโทรเลอร์จะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านข้อมูลคำสั่งแล้วทำงานตามคำสั่งที่บรรจุอยู่ในหน่วยความจำโปรแกรม

ไมโครคอนโทรเลอร์เปรียบเสมือนสมองที่ใช้ควบคุมอุปกรณ์ต่างๆ ในการเคลื่อนที่ของหุ่นยนต์ซึ่งจะเป็นตัวรับค่าต่างๆ จากมอเตอร์มาประมวลผลและส่งสัญญาณไปควบคุมความเร็วของมอเตอร์ ในการจัดทำโครงการในครั้งนี้ได้เลือกใช้ไมโครคอนโทรเลอร์รุ่น CP-PIC V4.0 เบอร์ 18F458 เนื่องจากมีลักษณะการใช้งานที่ง่ายและภาษาที่ใช้ในการเขียนโปรแกรมก็สามารถเข้าใจได้ง่าย บอร์ดควบคุมในตระกูล PIC มีหลายรุ่นซึ่งแต่ละชนิดมีความสามารถที่แตกต่างกันในการเลือกที่ใช้รุ่นไหนพิจารณาจากขอบเขตของงานที่เราต้องการให้มันทำ และงบประมาณที่เรามีนั่นเอง

3.7.1 ลักษณะโดยทั่วไป

บอร์ดไมโครคอนโทรลเลอร์ CP-PIC V4.0 ที่ได้ออกแบบนี้ เป็นบอร์ดที่ออกแบบไว้ใช้งานกับไมโครคอนโทรลเลอร์ในตระกูล PIC โดยจะสามารถใช้ได้ กับเบอร์ 16F877-20P,18F442 และ 18F458 หรือเบอร์อื่นๆ ที่มีโครงสร้างและตำแหน่งขาสัญญาณเหมือนกันโดย CPU แต่ละเบอร์ก็จะมีคุณสมบัติที่แตกต่างกันซึ่งสามารถสรุปคุณสมบัติของ CPU แต่ละเบอร์อย่างคร่าวๆ ดังตาราง

DEVICE	Program Memory	Data Memory		CAN	I/O	OSC	Timers	PLL
	Flash	RAM (Bytes)	EEPROM (Byte)	Module	(Bit)	max (MHz)		
PIC 16F877	8K (14-Bit Words)	368	256	NO	33	20MHz	3	NO
PIC 18F442	16 Kbyte	768	256	NO	34	40MHz	4	YES
PIC 18F458	32 Kbyte	1536	256	YES	34	40MHz	4	YES

ตารางที่ 3.3 ตารางแสดงคุณสมบัติ CPU ของไมโครคอนโทรลเลอร์พิคแต่ละเบอร์

CPU ดังกล่าวจะบรรจุอยู่ภายในตัวถังแบบ DIP ขนาด 40 ขา และมีทรัพยากรต่างๆบรรจุไว้ภายในตัว CPU อย่างครบถ้วน ไม่ว่าจะเป็น ADC/TIMER/COUNTER/PWM หรือ PORT I/O ต่างๆ ซึ่งมีความเหมาะสมในการนำไปประยุกต์ใช้งานในลักษณะต่างๆได้เป็นอย่างดี

สำหรับอุปกรณ์ I/O ต่างๆ ซึ่งไม่ได้มีบรรจุไว้ในตัวCPU ด้วยทางทีมงานอีทีที ก็ได้จัดหาและทำการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ต่างๆที่มีความจำเป็นไว้ให้ด้วยแล้ว ไม่ว่าจะเป็นระบบฐานเวลา RTC วงจร Line Driver สำหรับการสื่อสารข้อมูลอนุกรมแบบ RS232 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

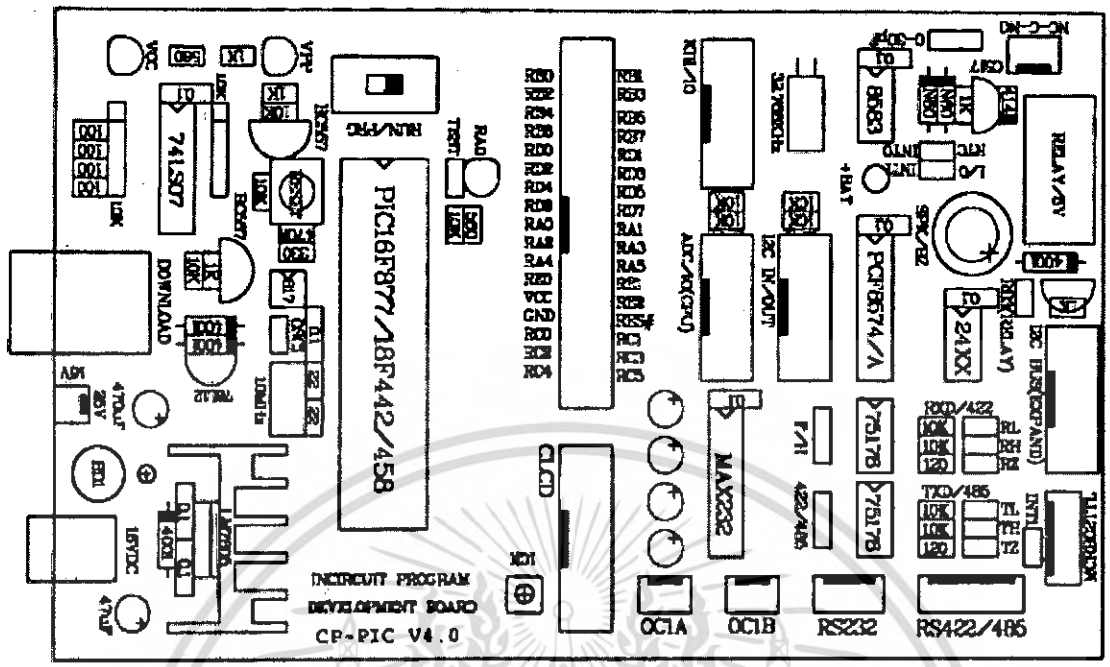
RS422/485 และยังสามารถให้ผู้ใช้ทำการเพิ่มเติมอุปกรณ์ I/O อื่นๆ เข้าไปได้อีกตามความจำเป็นในการใช้งาน

CP-PIC V4.0 เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ I/O ภายนอกอื่นๆ ที่มีความจำเป็นไว้รองรับการใช้งานในลักษณะต่างๆ เพื่อให้ผู้ใช้งานสามารถนำบอร์ดไปใช้งานในลักษณะงานที่แตกต่างกันได้ โดยไม่ต้องดัดแปลงวงจร หรือ อาจดัดแปลงวงจรเพียงเล็กน้อยสำหรับงานบางอย่าง ซึ่งบอร์ดรุ่นนี้เหมาะสำหรับกลุ่มผู้ที่ต้องการนำบอร์ดไมโครคอนโทรลเลอร์ไปใช้งานจริงๆ แต่ไม่สะดวกที่จะสร้างบอร์ดเอง ประกอบด้วยส่วนต่างๆ ดังนี้

- RS – 232 1 แชนเนล
- RS-422/458 1 แชนเนล (IC 75176 เป็น Option)
- ETT CON 34PIN (ET BUS I/O 34PIN)
- 5 Volt Regulator On Board
- วงจรโปรแกรมแบบ High Voltage ภายในบอร์ด
- ใช้ Adaptor 16VDC (Option)
- ADC/IO(CPU) พอร์ตสำหรับต่อ อินพุตอนาลอก 8 Channel
- CLCD 14PIN พอร์ตสำหรับต่อ LCD (4 Bit Data)
- RTC #PCF8583P (Option)
- EEPROM ตั้งแต่เบอร์ #2432 ขึ้นไป (Option)
- I2C IN/OUT เป็นIC ขยายพอร์ต I/O #PCF8574AP (Option)
- KBI/IO 10 Pin สำหรับต่อกับ Keyboard หรือ ใช้เป็น Input /Output Port
- Relay Onboard 5V 1ตัว (Option)
- MCRB02TTL ขั้วต่อ Macnetic Card Reader
- Mini Speaker/Buzzer
- I2C BUS(EXPAND)
- PWM1 ขั้วต่อสำหรับใช้งาน Capture/Compare/PWM ตัวที่หนึ่ง
- PWM2 ขั้วต่อสำหรับใช้งาน Capture/Compare/PWM ตัวที่สอง

หมายเหตุ Option คือ ส่วนที่ออกแบบไว้ให้เป็น Socket เปล่าๆหากต้องการใช้งานต้องหาซื้อเพิ่มเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดง โครงสร้างของบอร์ด CP-PIC V4.0

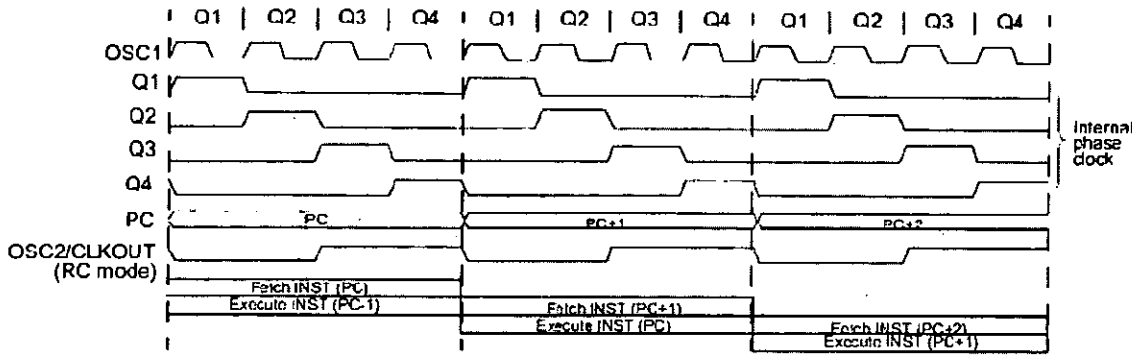
3.7.2 แหล่งจ่ายไฟ (POWER SUPPLY)

สำหรับแหล่งจ่ายไฟของบอร์ด CP-PIC ทั้ง V3.0 และ V4.0 นั้น จะสามารถต่อใช้งานได้ทั้งกับไฟกระแสตรงและกระแสสลับ เนื่องจากในบอร์ดได้จัดเตรียมวงจร RECTIFIER แบบ BRIDGE พร้อมวงจร FILTER และREGULATOR ขนาด +5V ไว้ให้อย่างครบถ้วนอยู่แล้ว โดยผู้ใช้สามารถป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดันไม่ต่ำกว่า 13V เนื่องจากบอร์ดถูกออกแบบให้โปรแกรมข้อมูลด้วย ไฟสูง คือ 13 V จึงต้องใช้แหล่งจ่ายไฟที่มากกว่า 13V อยู่ในช่วงประมาณ 13V ถึง 16V โดยสามารถเลือกต่อกับขั้ว CONNECTOR แบบ CPA ขนาด 2 ขา หรือจะต่อผ่านขั้ว CONNECTOR สำหรับ ADAPTER จ่ายไฟก็ได้เช่นกัน โดยการทำงานของแหล่งจ่ายไฟจะมีหลอดแสดงผลLED “VCC” สำหรับแสดงผลการทำงานให้ทราบด้วย

3.7.3 สัญญาณนาฬิกา CLOCK

ไมโครคอนโทรลเลอร์จะมีการทำงานตามจังหวะสัญญาณนาฬิกาของระบบ ซึ่งใน 1 Cycle (Clock Bus) ของ CPU จะประกอบไปด้วย สัญญาณนาฬิกาจากภายนอกจำนวณ 4 cycle คือ Q1, Q2, Q3 และ Q4 ดังรูปด้านล่างฉนั้นความถี่ที่ CPU ประมวลผลต่อคำสั่งจะเท่ากับ ความถี่จากคริสตอลภายนอก หารด้วย 4 หรือ อาจกล่าวได้ว่าความเร็วในการทำงานของ ไมโครคอนโทรลเลอร์ตระกูล PIC ต่อคำสั่งจะมีค่าเป็น ¼ เท่าของความถี่คริสตอลออสซิลเลเตอร์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์พิค

แหล่งกำเนิดสัญญาณนาฬิกาได้ 2 แบบ คือ

- ใช้ค่าความถี่จากวงจรกำเนิดความถี่แบบ Oscillator จากภายนอก โดยสามารถใช้ได้กับ Oscillator ที่มีค่าความถี่ระหว่าง 0-40MHz ในเบอร์ 18F442 กับ 18F458 ส่วน 16F877-20P จะอยู่ในช่วง 0-20MHz

- ใช้ค่าความถี่จากวงจร PLL(Phase-Lock-Loop) ที่บรรจุไว้ภายในตัว CPU แล้ว โดยวิธีการนี้จะต้องป้อนค่าความถี่ Crystal ขนาด 0-10MHz ให้กับขา OSC1 และ OSC2 ของ CPU ด้วย แล้วจึงทำการกำหนดค่า Configuration ในการโปรแกรมให้อยู่ในโหมดของ เฟสล็อกลูป ซึ่งเมื่อโปรแกรมการทำงานในโหมดนี้ CPU จะใช้วงจร เฟสล็อกลูปภายใน คูณสัญญาณนาฬิกาที่เข้ามาทางขา OSC1 และ OSC2 ด้วยสี่ เช่น ใช้คริสตอลออสซิลเลเตอร์ขนาดความถี่ 10 MHz เมื่อผ่านวงจรเฟสล็อกลูปภายใน สัญญาณที่ได้ออกมาจะมีความถี่ 40 MHz ซึ่งเป็นสี่เท่าของสัญญาณนาฬิกาที่เข้ามา คุณสมบัตินี้จะไม่มีอยู่ใน CPU PIC16F877 สามารถทำได้เฉพาะเบอร์ 18F442 และ 18F458 เท่านั้น

ซึ่งการทำงานของ CPU จะอาศัยสัญญาณนาฬิการะบบ หรือ BUS CLOCK เป็นจุดอ้างอิงการทำงานให้สัมพันธ์และสอดคล้องกับวงจรภายในอื่นๆ โดยที่ค่าความเร็วของสัญญาณนาฬิการะบบ หรือ BUS CLOCK ของ CPU นั้น จะมีค่าเป็น ¼ ของสัญญาณนาฬิกาจาก Oscillator ภายนอก หรือในกรณีที่มีการใช้งานวงจร PLL ด้วย ค่าความถี่ของ BUS CLOCK ก็จะมีค่าเป็น ¼ ของสัญญาณนาฬิกา Output ที่ได้จากวงจร PLL เช่นกัน

ดังนั้นในกรณีที่จะใช้สัญญาณนาฬิกาจาก Oscillator ภายนอก จะต้องเลือกใช้ Oscillator ที่มีค่าความถี่อยู่ระหว่าง 0-40MHz (เฉพาะ 18F442 และ 18F458 ส่วน 16F877 จะอยู่ในช่วง 0-20MHz) หรือในกรณีที่จะใช้ค่าความถี่ของสัญญาณนาฬิกาจาก Crystal ภายนอกร่วมกับวงจร PLL นั้น ค่าคริสตอลที่นำมาต่อจะต้องมีค่าไม่เกิน 10MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

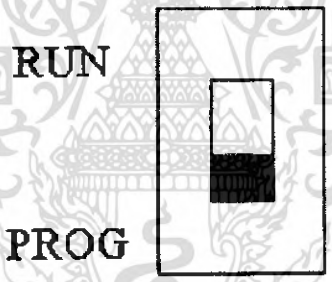
3.7.4 โหมดการทำงานของบอร์ด

การทำงานของบอร์ด CP-PIC V3.0&V4.0 นั้น สามารถกำหนดโหมดการทำงานของบอร์ดได้ 2 โหมดการทำงานด้วยกัน คือ โหมดการโปรแกรม (PROG) และโหมดการทำงานปรกติ (RUN)

3.7.5 การทำงานในโหมดการโปรแกรม (PROG)

ในโหมดนี้จะใช้สำหรับในกรณีที่ต้องการโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ ในบอร์ด CP-PIC V4.0 นี้ ได้ออกแบบในส่วนของการโปรแกรมเป็นแบบ High Voltage Programing ใช้แรงดันในการโปรแกรม 13 V ข้อดีของการโปรแกรมแบบนี้คือสามารถใช้งาน I/O Port ได้ครบทุกขา และในการออกแบบได้ใช้ SLIDE SWITCH เพื่อตัดต่อขาสัญญาณที่ใช้ในการโปรแกรม ดังนั้นเมื่อทำงานใน โหมดปกติขาสัญญาณต่างๆ ก็จะถูกแยกออกจากวงจรส่วนของการโปรแกรม ดังนั้นจึงสามารถใช้งานขาสัญญาณต่างๆ ได้ครบทั้งหมด

การเข้าสู่โหมดของการโปรแกรมทำได้โดยการเลือกตำแหน่ง SLIDE SWITCH (PROG/RUN) มาที่ตำแหน่งPROG ส่วนซอฟว์แวร์ ที่ใช้ในการดาวน์โหลดโปรแกรมจะใช้โปรแกรม EPICwin ซึ่งในรายล่ะเอียดการใช้งานจะกล่าวภายหลัง ในหัวข้อ การใช้งาน EPICwin



รูปที่ 3.15 สวิตช์เลือกโหมด RUN/PROG

3.7.6 การทำงานใน USER MODE หรือ RUN MODE

การทำงานในโหมดนี้ คือ การทำให้ CPU กระทำตามคำสั่งต่างๆ ตามโปรแกรมที่เราได้ออกแบบไว้ซึ่งการเข้าสู่โหมดนี้ ทำได้โดยการเลือกตำแหน่ง SLIDE SWITCH (PROG/RUN) มาที่ตำแหน่ง RUN สวิตช์ Slide ก็จะทำการแยกขาสัญญาณต่างๆ ออกจากวงจรในส่วนของการโปรแกรม ฉนั้นในการใช้งาน I/O Port จึงสามารถนำมาใช้งานได้ครบทั้งหมดและจากการทดลองการใช้ไมโครคอนโทรลเลอร์ PIC 18F485 ผ่านบอร์ด CP-PIC V4.0 ควบคุมเราได้พบปัญหาต่างๆ เช่น การคำนวณ และการสั่งการของไมโครคอนโทรลเลอร์ทำได้ค่อนข้างช้า พื้นที่ภายในไมโครคอนโทรลเลอร์พิคมีอยู่อย่างจำกัด เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8. การควบคุมการเดินของหุ่นยนต์

ในการควบคุมการเดินของหุ่นยนต์เราได้ทำการทดลองอยู่ 3 วิธีด้วยกัน คือ

1. ควบคุมผ่านคอมพิวเตอร์ โดยใช้ภาษาซีผ่านพอร์ตอนุกรม
2. ควบคุมผ่านพอร์ตอนุกรม โดยใช้โปรแกรม MATLAB สั่งการ
3. ควบคุมโดยไมโครคอนโทรลเลอร์ Pic 18F485 ผ่านบอร์ด CP-PIC V4.0

ซึ่งแต่ละวิธีก็จะมีข้อดีและข้อเสียแตกต่างกันไป ในส่วนนี้เราจะทำการอธิบายการใช้โปรแกรมต่างๆในการควบคุมหุ่นยนต์ ดังนี้

3.8.1. ควบคุมผ่านพอร์ตอนุกรม โดยใช้โปรแกรม MATLAB สั่งการ

ในการเขียนโปรแกรมโดยใช้โปรแกรมแมทแลบจากภาคการศึกษาที่ผ่านเราพบว่าแมทแลบฟังค์ชั่นทางคณิตศาสตร์ให้เลือกใช้อย่างสะดวกมากมาย และยังมีโคต (CODE) ที่ใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมไว้พร้อมแล้ว แต่มีข้อเสียตรงที่ว่าตัวโปรแกรมแมทแลบไม่มีความเสถียรในการทำงานเท่าที่ควร และในการขียนโปรแกรมเรายังขาดความรู้ในเชิงลึกในการใช้โปรแกรมทำให้เราเขียนโปรแกรมที่มีขนาดใหญ่ ไม่สามรถทำการย่อโปรแกรมได้นั่นเอง เราได้เสนอวิธีการใช้แมทแลปไว้ที่ภาคผนวก

3.8.2 ควบคุมโดยไมโครคอนโทรลเลอร์ PIC 18F458 ผ่านบอร์ด CP-PIC V4.0

เป็นทางเลือกทางหนึ่งที่เราได้นำมาใช้ เนื่องจากการใช้โปรแกรมแมทแลบมีความไม่เหมาะสมเท่าที่ควรดังที่ได้กล่าวมาแล้ว เราจึงได้ทอลองใช้การขียนโปรแกรมด้วยภาษาซี ซึ่งผู้เขียนมีความชำนาญมากว่า แล้วทำการเบิร์นลงไมโครคอนโทรลเลอร์พิค โดยใช้บอร์ด CP-PIC V4.0 ซึ่งเป็นได้ทั้งบอร์ดเบิร์น และบอร์ดควบคุม (ดังที่ด้อธิบายในข้างต้นแล้ว) แล้วทำการทดสอบพบว่า.

1. การคำนวณและการสั่งการของไมโครคอนโทรลเลอร์ทำได้ค่อนข้างช้า
2. พื้นที่ภายในไมโครคอนโทรลเลอร์พิคมีอยู่อย่างจำกัด
3. นำไปพัฒนาต่อได้ยากเพราะผู้ที่จะทำต่อต้องมีความรู้ทั้งทางด้านโปรแกรมภาษาซี และบอร์ด PIC อย่างลึกซึ้ง

3.8.3. ควบคุมผ่านคอมพิวเตอร์ โดยใช้ภาษาซีผ่านพอร์ตอนุกรม

ข้อดีของภาษาซี คือ โปรแกรมที่เขียนด้วยภาษาซี สามารถแก้ไขได้ง่ายกว่าภาษาแอสแซมบลี เพราะโครงสร้างง่ายต่อการสร้างโมดูลาร์ (Modular) เช่น ฟังก์ชัน Main, ฟังก์ชัน While ตลอดจนคำสั่ง If, else If, เป็นต้น และยังสามารถสั่งงานได้อย่างรวดเร็ว การคำนวณ และการสั่งงานเร็วกว่าการควบคุมผ่านไมโครคอนโทรลเลอร์ฉนั้นเราจึงได้ทำการเขียนรูปแบบในการควบคุมหุ่นยนต์ พร้อมคำอธิบายต่างๆ จากการทดลองเราพบว่าวิธีนี้เหมาะสมที่สุดเนื่องจาก

1. ต้องทำการสร้างโคดเพื่อใช้ในการติดต่อสือสารผ่านพอร์ตอนุกรมเอง
2. การคำนวณ และการสั่งงานเร็วกว่าการควบคุมผ่านไมโครคอนโทรลเลอร์
3. สามารถใช้ทรัพยากรในการเขียนโปรแกรมอย่างเต็มที่ เพราะเขียนอยู่บนคอมพิวเตอร์
4. สามารถนำไปพัฒนาต่อได้ง่าย และหลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบและการสร้าง

ในหัวข้อนี้ได้กล่าวถึง การออกแบบ โดยเริ่มตั้งแต่การเลือกสรร วัสดุที่จะนำมาใช้ในการสร้างหุ่น ที่มาของต้นกำลัง อุปกรณ์ที่นำมาใช้ในการควบคุม การคำนวณในส่วนต่างๆ ของตัวหุ่น และการเขียนโปรแกรมเพื่อใช้ในการควบคุมการเดินของหุ่น

4.1 วัสดุและอุปกรณ์

4.1.1 วัสดุที่ใช้ทำหุ่นยนต์

ในโลกเรามีวัสดุมากมายที่สามารถนำมาใช้ในการทำหุ่นยนต์ได้ และสามารถจำแนกวัสดุออกเป็น 2 ประเภทใหญ่ได้ดังนี้

1. อโลหะ
2. โลหะ

โดยมีความแตกต่างทางกายภาพที่สามารถบ่งบอกได้ ดังนี้ โลหะจะมีความแข็งแรงมากกว่าอโลหะ มีจุดหลอมเหลวที่ค่อนข้างสูง และโลหะส่วนมากมักเป็นของแข็ง ยกเว้น ปรอท

จากลักษณะต่างๆ ที่กล่าวมา เราพบว่าโลหะมีคุณสมบัติหลายๆ อย่างที่ตรงกับลักษณะของหุ่นของเรา เช่นมีความแข็งแรงทนทาน อีกทั้งยังมีความสามารถในการทนความร้อนได้ดีอีกด้วย ซึ่งต่อไปอาจพัฒนาไปเป็นหุ่นยนต์กู้ภัย

จากนั้นเราได้ทำการพิจารณาคุณสมบัติของโลหะชนิดต่างๆ ดังตารางที่ 3.1 เพื่อทำการเลือกวัสดุที่จะนำมาใช้ในการสร้างหุ่น

วัสดุ	ความแข็งแรง	จุดหลอมเหลว	ความทนทานต่อสนิม	ความยากในการตัดปาด	น้ำหนัก	ราคา
เหล็ก	สูง	สูงมาก	ต่ำ	สูงมาก	สูงมาก	สูง
สแตนเลส	สูงมาก	สูง	สูงมาก	สูงมาก	สูง	สูง
อลูมิเนียม	ปานกลาง	สูง	สูงมาก	ดี	ต่ำ	ปานกลาง
ไททาเนี่ยม	สูงมาก	สูง	สูงมาก	สูง	ต่ำ	สูงมาก

ตารางที่ 4.1 แสดงคุณสมบัติที่เราต้องการของโลหะชนิดต่างๆ ที่เรานำมาพิจารณาเลือก

จากตารางคุณสมบัติ เราพบว่าอลูมิเนียมมีความเหมาะสมที่สุด ในการสร้าหุ่นต้นแบบของเรา เนื่องจากวัตถุประสงค์ของโครงงาน เราสร้างหุ่นยดน์ห้าขาเพื่อออกแบบการเดิน และสร้างหุ่นต้นแบบ ดังนั้นหุ่นยนต์ของเราจึงไม่มีความจำเป็นที่ต้องรับน้ำหนักมาก เราจึงเลือกอลูมิเนียมมาเป็นวัสดุในการทำหุ่น นอกจากจะมีคุณสมบัติในเรื่องของความเบาแล้วก็ยังราคาไม่แพงจนเกินไปนัก และยังง่ายต่อการตัดปาดอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ต้นกำลัง

4.1.2.1 การเลือกต้นกำลัง เนื่องจากโครงงานนี้เป็นโครงงานทำต่อเนื่องมาจากรุ่นพี่ปีที่ผ่านมา จากปีที่ผ่านมานั้นรุ่นพี่ได้ออกแบบเอาไว้เป็นหุ่นยนต์ห้าขาที่ใช้ไฮโดรลิก และลูกสูบเป็นต้นกำลังเพื่อทีจะได้ให้มีกำลังไว้ใช้ในการงาน แต่จากจุดประสงค์ของกลุ่มเราเพื่อสร้างต้นแบบขนาดเล็ก และรูปแบบการเดินของหุ่นยนต์ห้าขา ดังนั้นหุ่นของเราไม่มีความจำเป็นที่จะต้องมีพละกำลังมากมาย เราจึงจำเป็นที่จะต้องเลือกต้นกำลังใหม่เพื่อให้เหมาะสมกับโครงงานของเรา

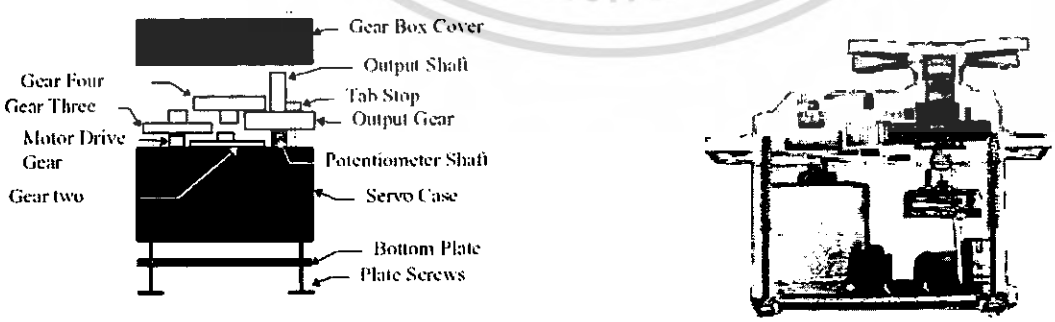
ชนิดของต้นกำลัง	แรง	ความเร็ว	การควบคุมตำแหน่ง	ราคา
นิวแมติกส์	ปานกลาง	ปานกลาง	ต่ำ	ปานกลาง
ไฮดรอลิค	สูงมาก	ต่ำ	ดี	สูงมาก
มอเตอร์กระแสตรง	ต่ำ	สูงมาก	ดี	ต่ำ
เซอร์โวมอเตอร์	กลาง / สูง	ปานกลาง	ดีมาก	ปานกลาง

ตารางที่ 4.2 แสดงคุณสมบัติจำเป็นที่เราต้องการ ของต้นกำลัง

เมื่อพิจารณาจากตารางแล้ว เซอร์โวมอเตอร์มีความเหมาะสมกับโครงงานหุ่นยนต์ห้าขามากที่สุด เนื่องจากเซอร์โวมอเตอร์สามารถควบคุมตำแหน่งได้แม่นยำ ควบคุมความเร็วของการหมุน และองค์ประกอบอื่นอยู่ในเกณฑ์ที่ดีเราจึงได้เลือกเซอร์โวมอเตอร์เป็น ต้นกำลังของหุ่น

4.1.2.2 เซอร์โวมอเตอร์

เซอร์โวมอเตอร์ หรือมอเตอร์ที่มีการประกอบชุดเกียร์ทำให้สามารถยกน้ำหนักได้มากกว่ามอเตอร์ทั่วไป และชุดควบคุมขั้นต้น อยู่ภายในดังรูปที่ 4.1



รูปที่ 4.1 แสดงส่วนประกอบของเซอร์โวมอเตอร์ [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของเซอร์โวมอเตอร์

(1) น้ำหนักเบา

(2) แรงบิดต่อน้ำหนักตัวสูง

(3) มีประสิทธิภาพสูง

(4) ควบคุมด้วยสัญญาณพัลส์โดยตรง

การเลือกเซอร์โวมอเตอร์

1 ปัจจัยที่มีผลต่อการเลือกเซอร์โวมอเตอร์ เนื่องจากเซอร์โวมอเตอร์มีการผลิตออกมามากมายหลายชนิด และแตกต่างกันออกไปตามลักษณะการใช้งานเพื่อให้เหมาะสมกับงาน แต่ละชนิด เราจึงต้องคำนึงถึงปัจจัยดังนี้

- ขนาดของตัวหุ่นและมอเตอร์
- น้ำหนักทั้งของตัวหุ่นยนต์ ภาระที่หุ่นต้องรองรับ และแรงบิดของเซอร์โวมอเตอร์
- ความเร็วที่เราต้องการใช้
- งบประมาณที่มี

2 คุณสมบัติของเซอร์โวมอเตอร์แต่ละรุ่นตามท้องตลาดทั่วไปแล้ว มีเซอร์โวมอเตอร์มากมายหลายรุ่นผลิตออกว่างขาย ซึ่งแต่ละรุ่นในของแต่ละบริษัทนั่นก็จะมีความสามารถแตกต่างกันไปในเรื่องของทั้งคุณภาพ และราคา ดังนั้นเราก็ต้องเลือกเซอร์โวมอเตอร์ให้สอดคล้องกับปัจจัยต่างๆ ตามที่เราได้พิจารณามาแล้วในหัวข้อก่อนหน้านี้

จากการศึกษาหาข้อมูล ของเซอร์โวมอเตอร์ที่มีขายอยู่ตามท้องตลาดทั่วไป เราพบว่ามีรุ่นที่น่าสนใจดังตารางคุณสมบัติของเซอร์โวมอเตอร์

Servo motor	Dimensions L x W x H (mm)	Weight (kg)	Voltage (V)	Speed (sec/60)	Torque (kg-cm)	Price (bath)
Futaba s3003	41 x 20 x 36	0.037	4.8	0.23	3.2	450
			6	0.19	4.1	
s03T/STD/JR	39.5 x 20.0 x 39.6	0.046	4.8	0.33	7.2	640
			6	0.27	8	
S04/BBM/JR	54.4 x 26.5 x 51.5	0.114	4.8	0.25	10	990
			6	0.2	13	

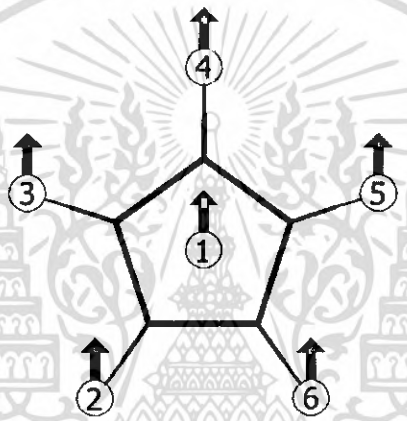
ตาราง 4.3 แสดงคุณสมบัติของเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบหุ่นยนต์

4.2.1 การออกแบบลักษณะการเดิน

4.2.1.1 เดินแบบ 6 ขั้นตอน โดยนำรูปแบบการเดินของรุ่นพี่มาศึกษา ซึ่งเป็นการเดินแบบก้าวทีละขา โดยหุ่นจะโย้ตัวไปข้างหน้าก่อนในจังหวะแรก จากนั้นจึงค่อยก้าวขาไปข้างหน้าทีละก้าวจนครบทุกขา แล้วจึงโย้ตัวไปข้างหน้าอีกครั้งเพื่อเริ่มเดินในรอบใหม่ จากการวิเคราะห์แล้วเราพบว่าการเดินแบบนี้มีข้อดีคือ หุ่นสามารถรับน้ำหนักได้มากเนื่องจากมีขาที่คอยรับน้ำหนักถึง 4 ข้าง ในระหว่างการเติน แต่ข้อเสียคือ หุ่นจะเดินช้ามากเพราะต้องใช้เวลาถึง 6 จังหวะในการเดินแต่ละก้าวและที่สำคัญที่สุดในจังหวะการโย้ตัวไปข้างหน้าจะส่งผลให้เซอร์โวมอเตอร์ของขาหน้าตรงต้องรับภาระที่สูงมากเพื่อหยุดตัวของหุ่นให้อยู่ในตำแหน่งที่ต้องการ



รูปที่ 4.2 แสดงรูปแบบการเดินของรุ่นพี่

4.2.1.2 การพัฒนาการเดิน จากการที่ลักษณะการเดินของรุ่นพี่เป็นการเดินที่ช้ามาก ดังนั้นเราจึงได้พัฒนารูปแบบการเดินให้เร็วกว่าเดิมขึ้น โดยลดจังหวะการเดินจาก 6 จังหวะลงมาเป็น 3 จังหวะโดยแบ่งได้เป็น

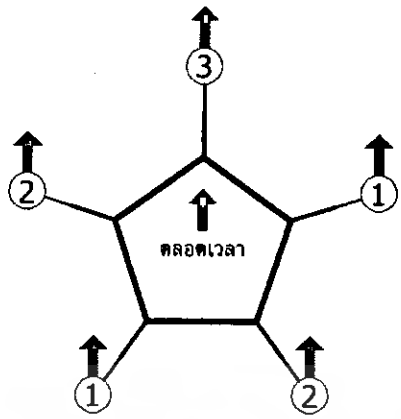
จังหวะที่ 1 เป็นการก้าวของขาหน้าตรงเพียงขาเดียว

จังหวะที่ 2 เป็นการก้าวของ ขาหน้าขวา และขาหลังซ้าย โดยมีขาหน้าตรง ขาหน้าซ้าย และขาหลังขวาเป็นขาที่รองรับน้ำหนัก

จังหวะที่ 3 เป็นการก้าวของ ขาหน้าซ้าย และขาหลังขวา โดยมีขาหน้าตรง ขาหน้าขวา และขาหลังซ้ายเป็นขาที่รองรับน้ำหนัก

โดยที่ลำตัวของหุ่นเคลื่อนที่ไปข้างหน้าตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงรูปแบบการเดินที่ปรับปรุงใหม่

การเดินแบบนี้มีข้อดีคือ สามารถเดินได้เร็วกว่าแบบเก่า ถึงแม้จะมีข้อเสียคือ จะมีขาที่คอยรับน้ำหนัก 3 ข้าง แต่เซอร์โวของขาหน้าตรงไม่ต้องรับภาระสูง เนื่องจากการที่ตัวหุ่นเคลื่อนที่ตลอดเวลา

4.2.2 การออกแบบขนาดของหุ่นยนต์

4.2.2.1 ปัจจัยที่มีผลต่อการออกแบบขนาดของขาหุ่นยนต์และขนาดตัวหุ่น

1 ระยะห่างระหว่างลำตัวกับพื้น ระยะห่างนี้ มีผลต่อความเร็วในการเดินของหุ่น และการเคลื่อนที่ข้ามสิ่งกีดขวางเนื่องจากจุดประสงค์ของการออกแบบหุ่นในครั้งนี้ ต้องการศึกษาถึงลักษณะท่วงท่าของการเดินเป็นหลัก จึงกำหนดให้ระยะห่างระหว่างลำตัว กับพื้นอยู่ที่ 2 เซ็นติเมตร

2 เซฟตี้เฟคเตอร์ในการออกแบบหุ่นครั้งนี้ไม่มีจุดประสงค์ที่จะให้หุ่นรับภาระใดๆจากภายนอก จึงกำหนดให้เซฟตี้เฟคเตอร์เป็น 1.5

3 ขีดจำกัดของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ในท้องตลาดทั่วไปมีมากมายหลายรูปแบบ คุณสมบัติและคุณภาพก็จะขึ้นอยู่กับราคา ยิ่งมีราคาที่สูงประสิทธิภาพของเซอร์โวมอเตอร์ก็จะสูงตาม และจากการศึกษาและออกแบบหุ่นห้าขาในภาคเรียนก่อนหน้านี้ และจากการศึกษาประสิทธิภาพของเซอร์โวมอเตอร์รุ่นต่างๆ ที่มีจำหน่ายอยู่ทั่วไปในท้องตลาดเราพบเซอร์โมอเตอร์รุ่นที่น่าสนใจดังนี้

Servo motor	Dimensions L x W x H (mm)	Weight (kg)	Voltage (V)	Speed (sec/60)	Torque (kg-cm)	Price (bath)
Futaba s3003	41 x 20 x 36	0.037	4.8	0.23	3.2	450
			6	0.19	4.1	
s03T/STD/JR	39.5 x 20.0 x 39.6	0.046	4.8	0.33	7.2	640
			6	0.27	8	

ตารางที่ 4.4 คุณสมบัติเซอร์โวมอเตอร์ขนาดต่างๆ

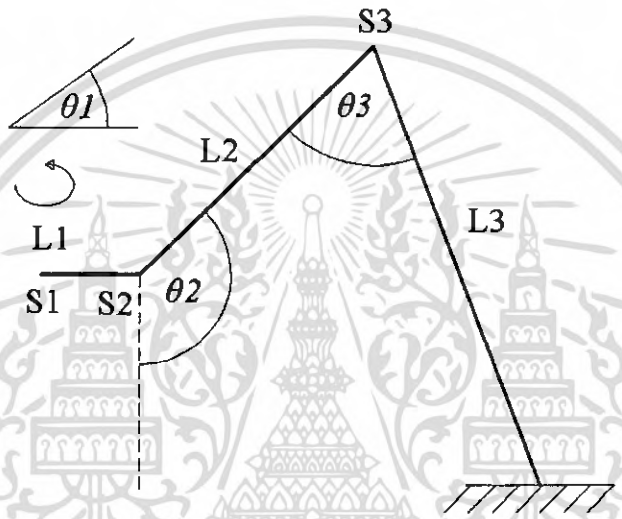
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากคุณสมบัติดังที่ได้แสดงในตารางเราก็จะนำไปใช้ในการออกแบบสร้างหุ่นยนต์ซึ่งเราก็จะได้กล่าวถึงในหัวข้อต่อไป

4.2.2.2 การคำนวณหาขนาดของหุ่นยนต์

1 ขนาดของขาหุ่นยนต์ จากขั้นตอนของการสรุปข้อมูลและปัจจัยต่างๆ ที่มีผลต่อการออกแบบแล้วเราได้ขนาดของความยาวของขาหุ่นในส่วนที่ 1 และส่วนที่ 2 แล้ว โดยเราจะทำให้สั้นที่สุดเท่าที่จะสามารถทำได้ และในขาส่วนที่ 3 นั้นเราเราได้ทำการพิจารณาดังต่อไปนี้

1. การวิเคราะห์หาสูตรเพื่อนำไปใช้ในการเขียนโปรแกรม ดังนี้



รูปที่ 4.4 แสดงการจำลองขาและตัวแปรต่างๆ

ความยาว

$$L_1 = L_2 = 68.9 \text{ mm} \tag{4.1}$$

มาจากการที่ทำขาของหุ่นในส่วนที่ 1 และ2 ให้สั้นที่สุด

ทอร์ค

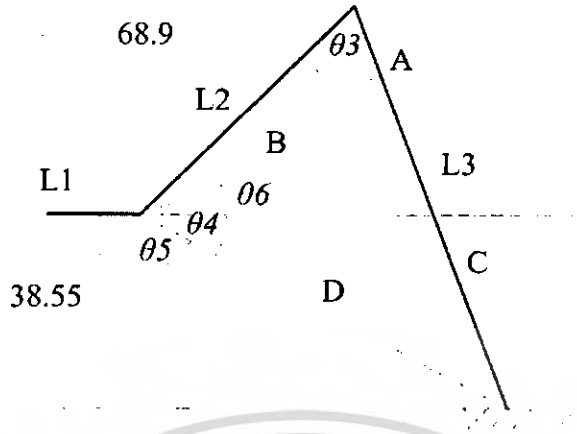
$$S_1 = S_2 = 7.2 \text{ kg cm} \tag{4.2}$$

$$S_3 = 3.2 \text{ kg cm} \tag{4.3}$$

มุมเล็กที่สุดที่ขาส่วนที่ 3 ทำกับส่วนที่ 2 ได้มาจากแบบในโปรแกรม SolidWorks

$$\theta_3 = 43.24 \text{ องศา} \tag{4.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงการจำลองขาและตัวแปรต่างๆ

ต่อมาสิ่งที่เราต้องการรู้คือค่าของ θ_2 และเราใช้ทฤษฎีของสามเหลี่ยมเข้ามาช่วยในการหาค่าของมุม โดยเริ่มจากลากเส้นจากจุดหมุนของเซอร์โวมอเตอร์ตัวที่ 2 มาตั้งฉากขาส่วนที่ 3 แล้วเราสามารถหาค่าความยาวของ A , B ,C และ D ได้โดยการใช้สูตรของตรีโกณมิติ ได้

$$A = L_2 COS\theta_3 \tag{4.5}$$

$$B = L_2 SIN\theta_3 \tag{4.6}$$

$$C = L_3 - A \tag{4.7}$$

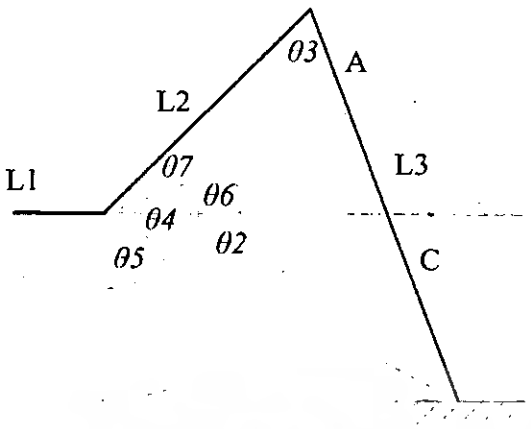
$$D = \sqrt{B^2 + C^2} \tag{4.8}$$

จากนั้นเราก็จะสามารถหาขนาดของ θ_4, θ_5 และ θ_6

$$\theta_4 = \tan^{-1}\left(\frac{C}{B}\right) \tag{4.9}$$

$$\theta_5 = COS^{-1}\left(\frac{38.55}{D}\right) \tag{4.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงการจำลองขาและตัวแปรต่างๆ

$$\theta_6 = 90 - \theta_4 - \theta_5 \tag{4.11}$$

ส่วน θ_7 เราหาได้จากกฎของสามเหลี่ยมมุมฉาก ได้

$$\theta_7 = 90 - \theta_3 \tag{4.12}$$

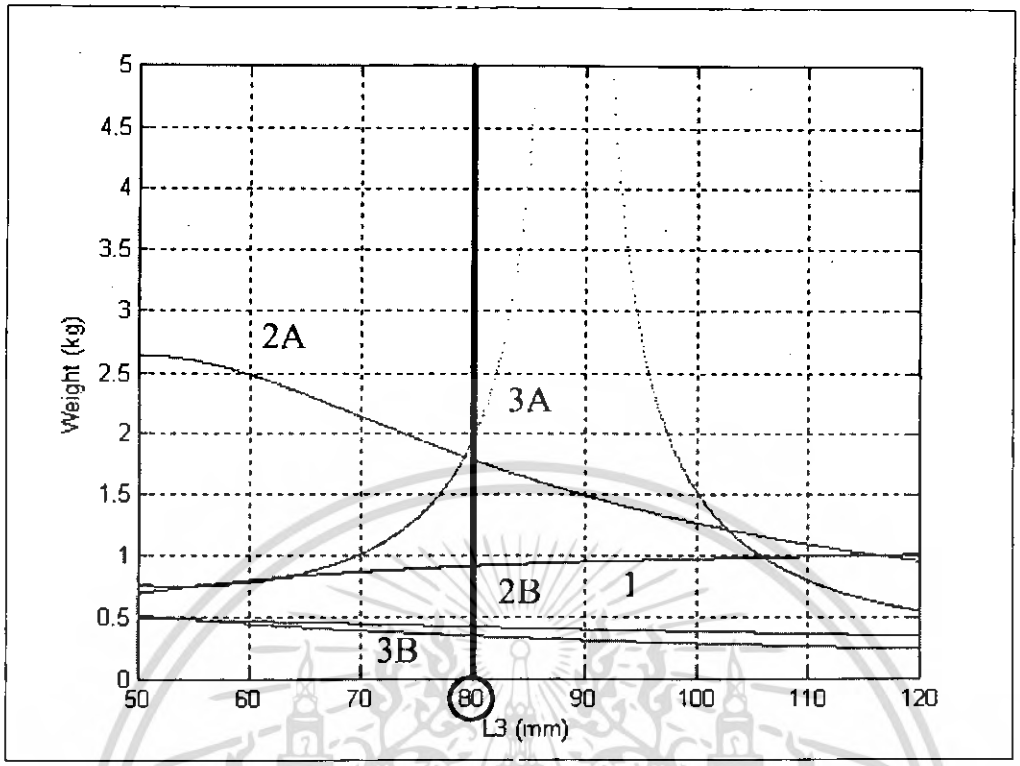
ดังนั้นเราก็จะสามารถหาค่า θ_2 ได้ตามต้องการ

$$\theta_2 = \theta_5 + \theta_4 + \theta_7 - \theta_6 \tag{4.13}$$

2 การวิเคราะห์กราฟจากโปรแกรมแมทแลบ จากนั้นนำสูตรที่ได้จากการวิเคราะห์มาเขียนโปรแกรมลงในแมทแลบ เพื่อพิจารณาหาขนาดของขาในส่วนที่ 3 โดย ทำการกำหนดค่าความยาวให้กับขาส่วนที่ 3 เริ่มตั้งแต่ 50 มิลลิเมตร จากนั้นเพิ่มขนาดความยาวขึ้น จนถึง 120 มิลลิเมตร แล้ววิเคราะห์หากราฟระหว่าง

1. ค่าทอร์ค ของเซอร์โวมอเตอร์ตัวที่ 2 เมื่อขามีความยาว สั้นที่สุด
2. ค่าทอร์ค ของเซอร์โวมอเตอร์ตัวที่ 3 เมื่อขามีความยาว สั้นที่สุด
3. ค่าทอร์ค ของเซอร์โวมอเตอร์ตัวที่ 2 เมื่อขามีความยาว ยาวที่สุด
4. ค่าทอร์ค ของเซอร์โวมอเตอร์ตัวที่ 3 เมื่อขามีความยาว ยาวที่สุด
5. ความสามารถในการก้าวขาเมื่อขาส่วนที่ 3 มีขนาดความยาวที่เปลี่ยนแปลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

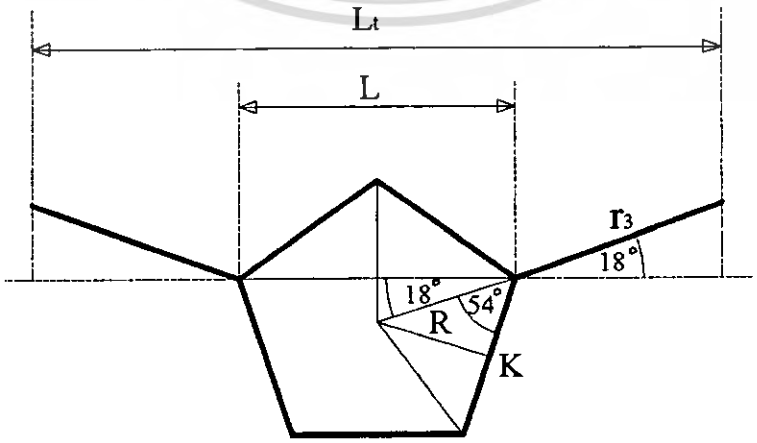


รูปที่ 4.7 กราฟที่ใช้พิจารณาขาส่วนที่ 3

จากกราฟเราได้ตัดสินใจเลือกช่วงความยาวของขาส่วนที่ 3 มีค่า 80 มิลลิเมตร เพาะที่ตำแหน่งดังกล่าวเป็นช่วงที่ตัดกันของเส้น กับเส้น ทำให้ได้แรงของเซอร์โวมอเตอร์ทั้งสองที่สมดุลย์เพราะถ้าหากขา L3 ยาวมากกว่านี้จะทำให้เซอร์โวตัวที่ 2 ที่จังหวะสั้นสุดรับแรงได้น้อยลง หรือถ้าขนาดของ L3 สั้นกว่านี้ก็จะทำให้เซอร์โวมอเตอร์ตัวที่ 3 รับแรงได้น้อยลงเป็นอย่างมาก

3 การวิเคราะห์หาขนาดของลำตัว การหาขนาดของลำตัว

กำหนดให้ความกว้างของตัวหุ่นทั้งหมดไม่เกิน 400 มิลลิเมตร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.8 แสดงการหาขนาดตัวหุ่นและตัวแปรต่างๆ

ในหัวข้อที่ผ่านมาเราได้ขนาดความยาวขา คือ

$$r_3 = 150.46\,\text{mm} \tag{4.14}$$

$$r_3 \cos 18 = 143.1 \tag{4.15}$$

จากนั้นใช้คุณสมบัติของตรีโกณหาความยาวของ K สมมุติให้ L_t มีค่าไม่เกิน 450 มิลลิเมตร

L มีค่าเท่ากับ 150 มิลลิเมตร

ดังนั้น

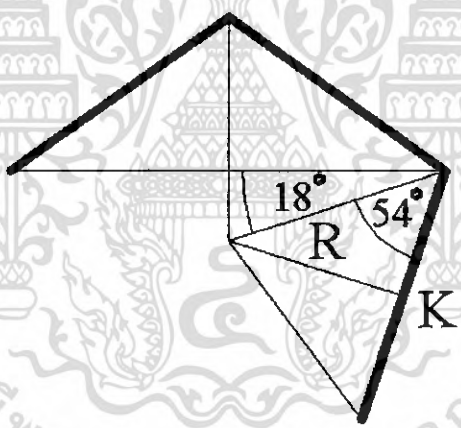
$$2r_3 \cos 18 + L < L_t \tag{4.16}$$

$$2\text{x}143.1+150=436.2<\text{L}_\text{t} \tag{4.17}$$

ค่าที่ได้มีค่าน้อยกว่า 450 แสดงว่าใช้ได้

ดังนั้น

$$\text{R}=\frac{\text{L}/2}{\cos 18}=78.86 \tag{4.18}$$



รูปที่ 4.9 แสดงการหาขนาดตัวหุ่นและตัวแปรต่างๆ

จากรูปเราสามารถหาขนาดของ K ได้ดังนี้

$$K = 2R\cos 54 \tag{4.19}$$

$$\text{K}=92.71\ \text{mm} \tag{4.22}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 การพิจารณาเลือกเซอร์โวมอเตอร์ ในการเลือกเซอร์โวมอเตอร์นั้นเราจำเป็นต้องเลือกให้เหมาะสมกับสิ่งที่เราจะเอาไปใช้ โดยมีปัจจัยที่สำคัญที่เป็นตัวกำหนด อยู่ 2 อย่าง คือ ทอร์ค ณ ข้อต่อที่เซอร์โวมอเตอร์แต่ละตัวต้องรับ และงบประมาณที่เรามี

จากการพิจารณา(ดังที่จะแสดงการคำนวณต่อไป) เราพบว่าจุดที่จะรับน้ำหนักมากที่สุดของตัวหุ่นยนต์คือจุดที่ 2 ดังนั้นเซอร์โวมอเตอร์ในจุดนี้ จึงควรมีทอร์คที่สูง ส่วนจุดที่ 3 รับโลดน้อยที่สุด และจุดที่ 1 นั้นแรงลงตามแนวแกนของมอเตอร์ ดังนั้นไม่จำเป็นที่จะต้องใช้มอเตอร์ที่มีทอร์คสูงมากนัก

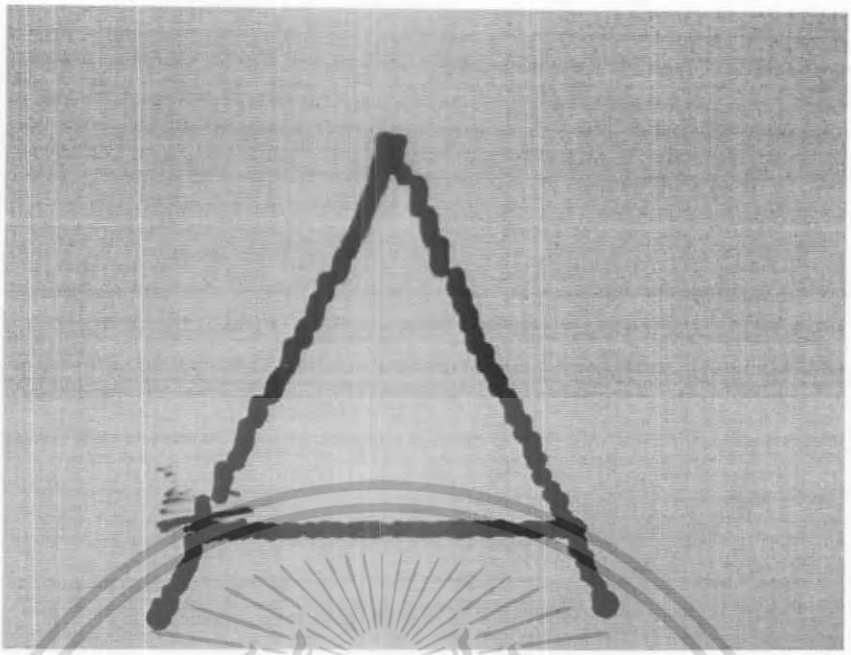
เลือก จุดที่ 1 ใช้ S04/BBM/JR, จุดที่ 2 ใช้ S04/BBM/JR, จุดที่ 3 ใช้ Futaba s3003

จากการที่เรามีการทดลองจำลองขา 1 ข้างขึ้นมาเพื่อทดลองควบคุมให้เขียนรูปตัว A โดยใช้เซอร์โวมอเตอร์ Futaba s3003 ในข้อต่อทั้ง 3 จุด และสร้างให้แต่ละข้อมีขนาดสั้นที่สุดเท่าที่จะทำได้ ดังรูป

รูปที่ 4.10 แสดงการทดลองควบคุมขาหุ่นหนึ่งข้างให้เขียนรูปตัว A

เราพบว่า ณ จุดที่ 1 เกิดการส่ายเนื่องจากเซอร์โวมอเตอร์พยายามจะรักษาตำแหน่งของมันเอาไว้ แต่มีแรงไม่เพียงพอ ดังนั้นในจุดที่ 1 นี้ควรจะเพิ่มทอร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงรูปตัว A ที่เขียนโดยขาของหุ่นยนต์ที่สร้างขึ้น

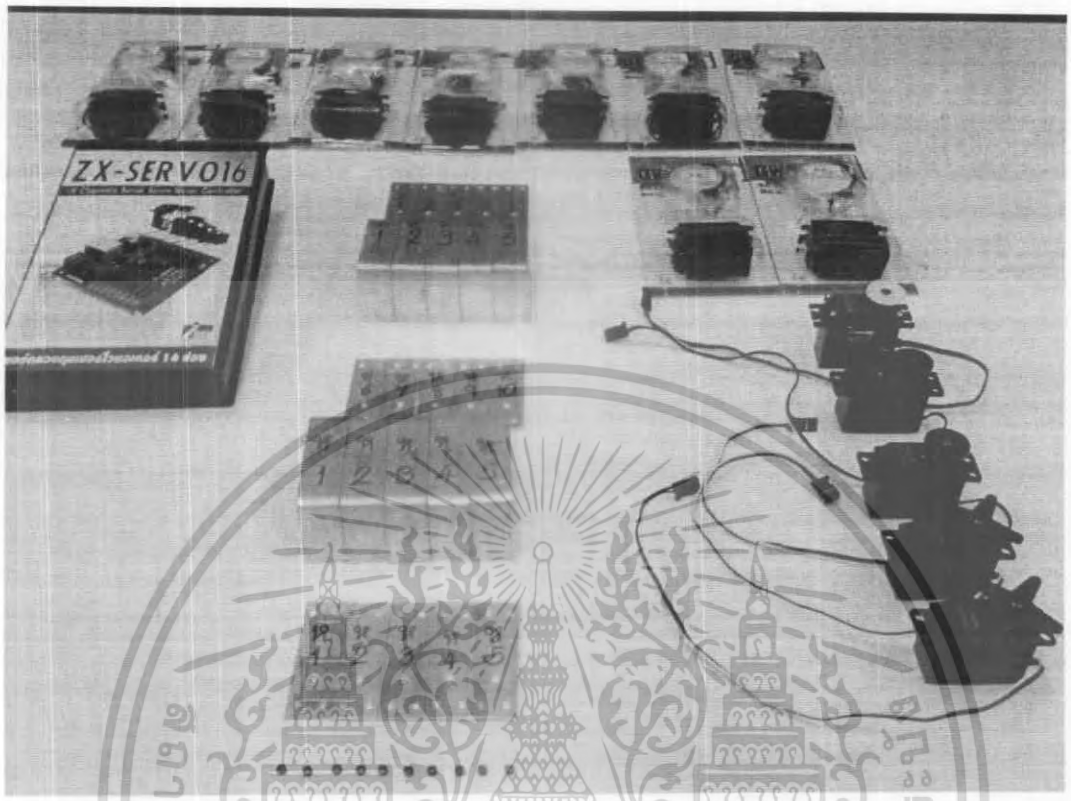
จากรูปเราพบว่าเราสามารถควบคุมการทำงานของขาของหุ่นยนต์ได้ด้วยโปรแกรมแมทแลปซึ่งค่อนข้างสมบูรณ์ แต่เนื่องจากเซอร์โวมอเตอร์ไม่มีทอร์กที่เพียงพอทำให้รูปตัว A ที่เขียนมาไม่นิ่งเท่าใดนัก

เมื่อเราได้ขนาดของเซอร์โวมอเตอร์แล้ว ต่อไปเราก็จะมาพิจารณาปัจจัยในเรื่องของงบประมาณ จากขนาดของเซอร์โวมอเตอร์ที่กล่าวมา ขา 1 ข้างต้องใช้เงินซื้อเซอร์โวมอเตอร์เป็นเงิน 2080 บาท ดังนั้น ขา 5 ข้างเป็นเงิน 10400 บาท ซึ่งงบประมาณของเรามีไม่เพียงพอ เราจึงต้องทำการลดขนาดของเซอร์โวมอเตอร์ที่จุดที่ 2 ลงมาใช้ s03R/STD/JR

สรุป จุดที่ 1 ใช้ s03T/STD/JR จุดที่ 2 ใช้ s03T/STD/JR จุดที่ 3 ใช้ Futaba s3003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การสร้างหุ่น



รูปที่ 4.12 แสดงส่วนประกอบสำคัญที่ใช้ในการสร้างหุ่น

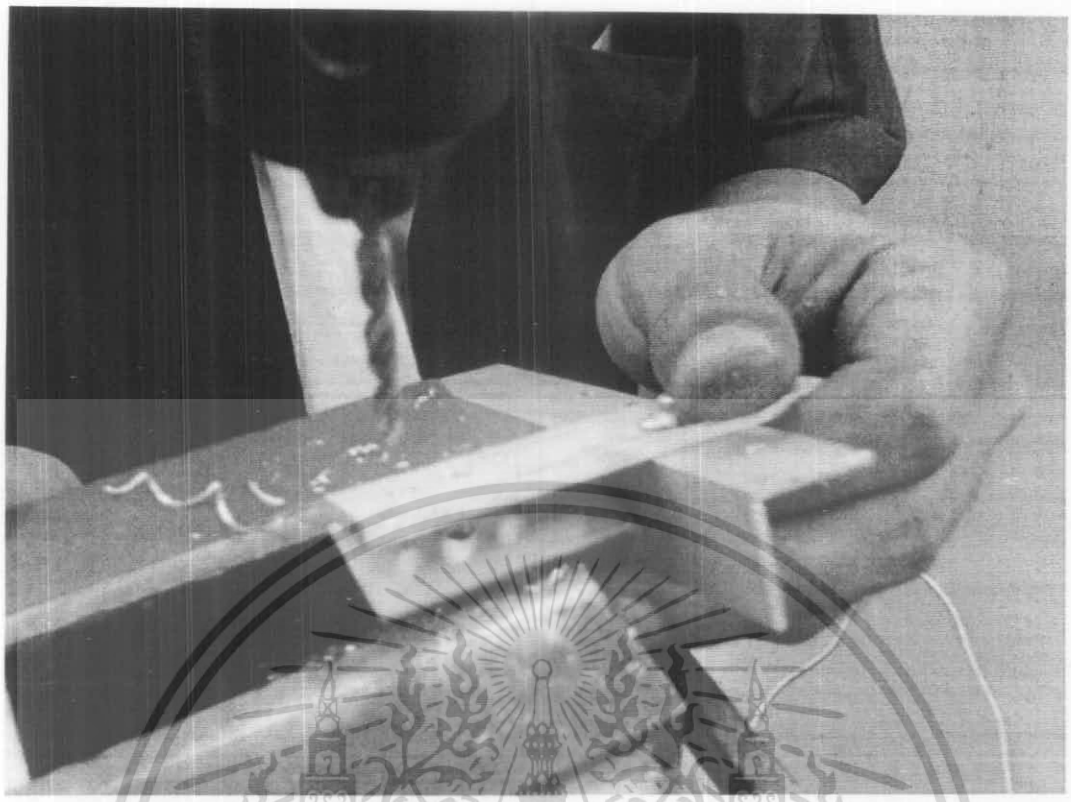
4.3.1 การสร้างต้นแบบขนาดเล็กของหุ่นยนต์

ในการสร้างหุ่นนั้นมีส่วนที่เราต้องให้ความระมัดระวังและใช้ความพิถีพิถันในการทำงาน ได้แก่

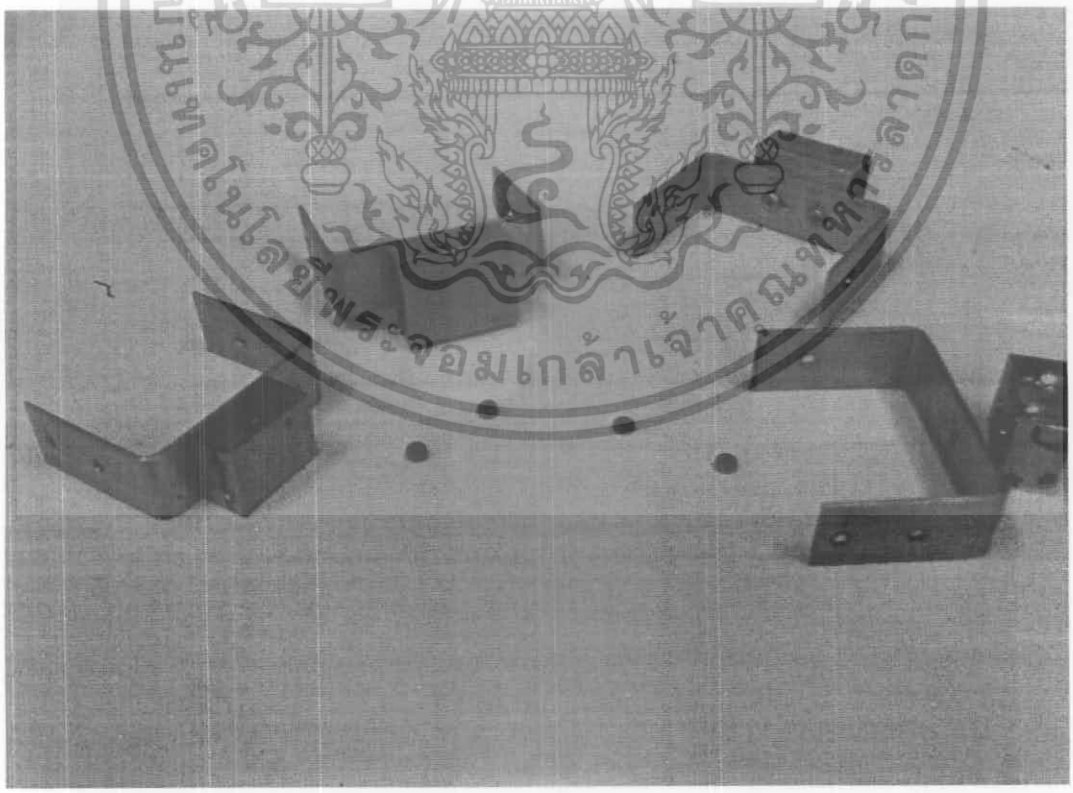
1. การตัดปาด เราได้ใช้เครื่องตัดในการทำการตัดในส่วนของอลูมิเนียมแผ่น และอลูมิเนียมเส้น และส่วนอลูมิเนียมกล่องเราได้ใช้เครื่องตัด แล้วจึงตกแต่ด้วยเครื่องกัด เพื่อให้ได้รอยตัดที่เรียบเนียนสวยงาม

2. การเจาะรูให้ได้ตรงตำแหน่งที่เราต้องการ เราได้ทำการสร้างจิ๊กขึ้นมาเพื่อใช้ในการเจาะรูให้เที่ยงตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

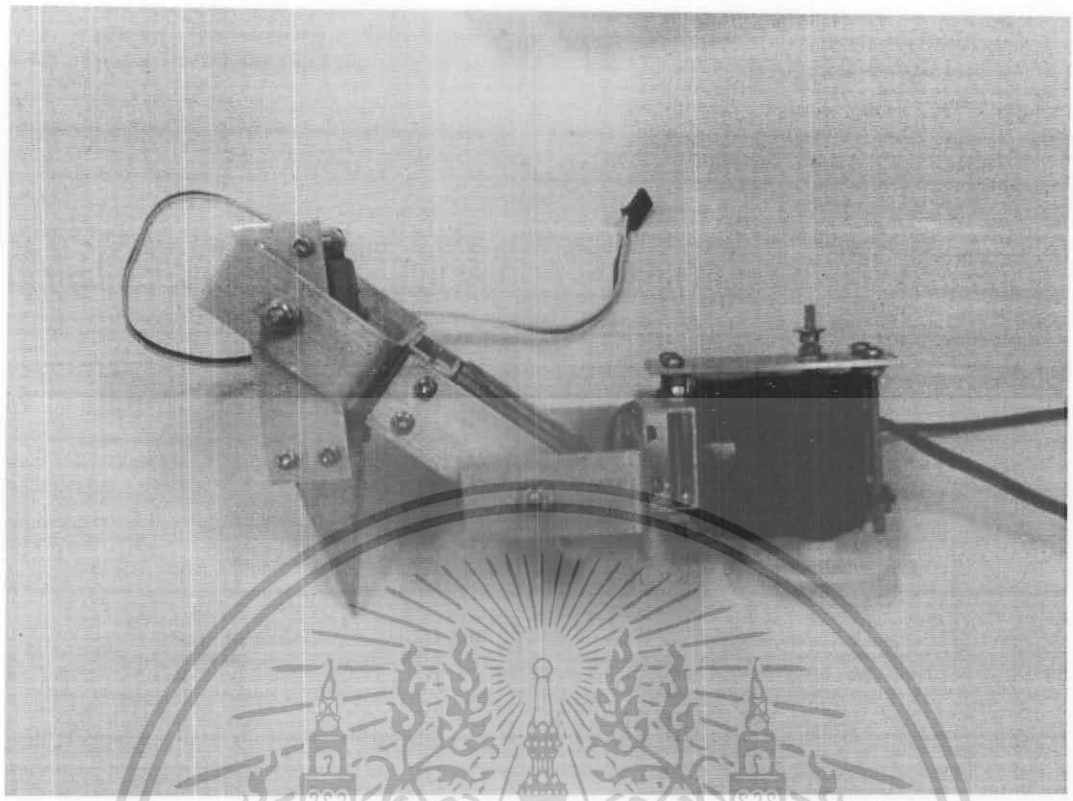


รูปที่ 4.13 แสดงการใช้จิ๊กช่วยในการเจาะรูในชิ้นส่วนของข้อต่อขา

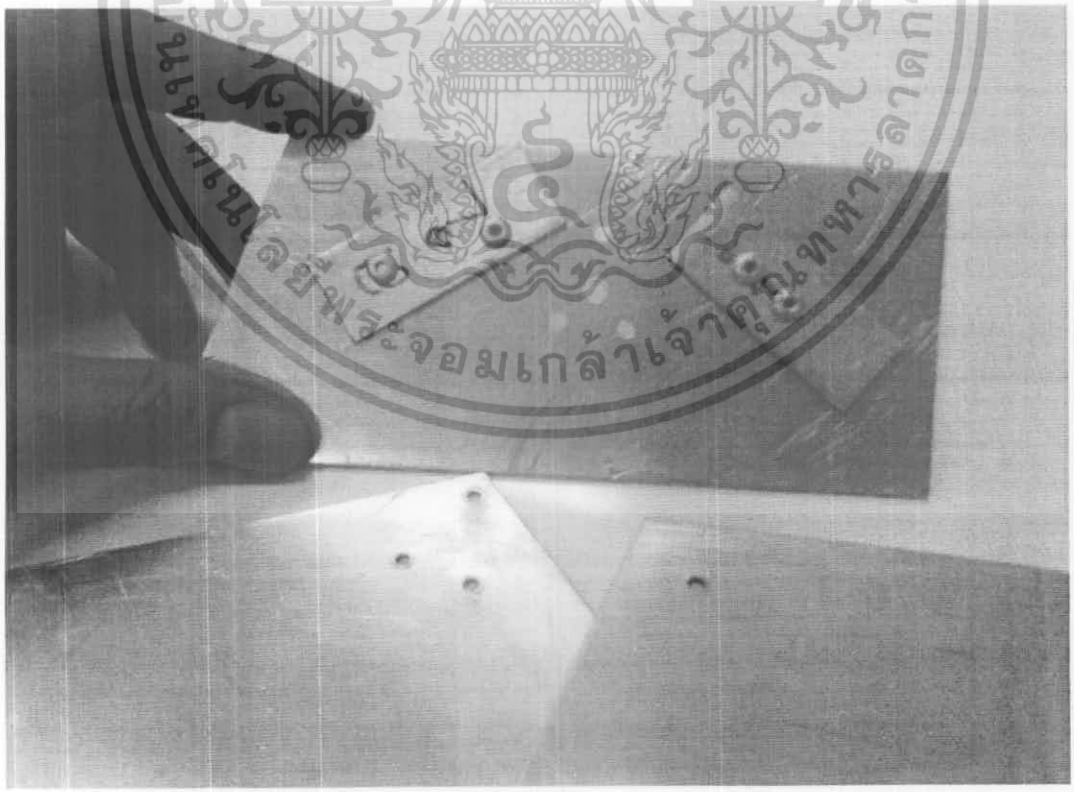


รูปที่ 4.14 แสดงรูปข้อต่อที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

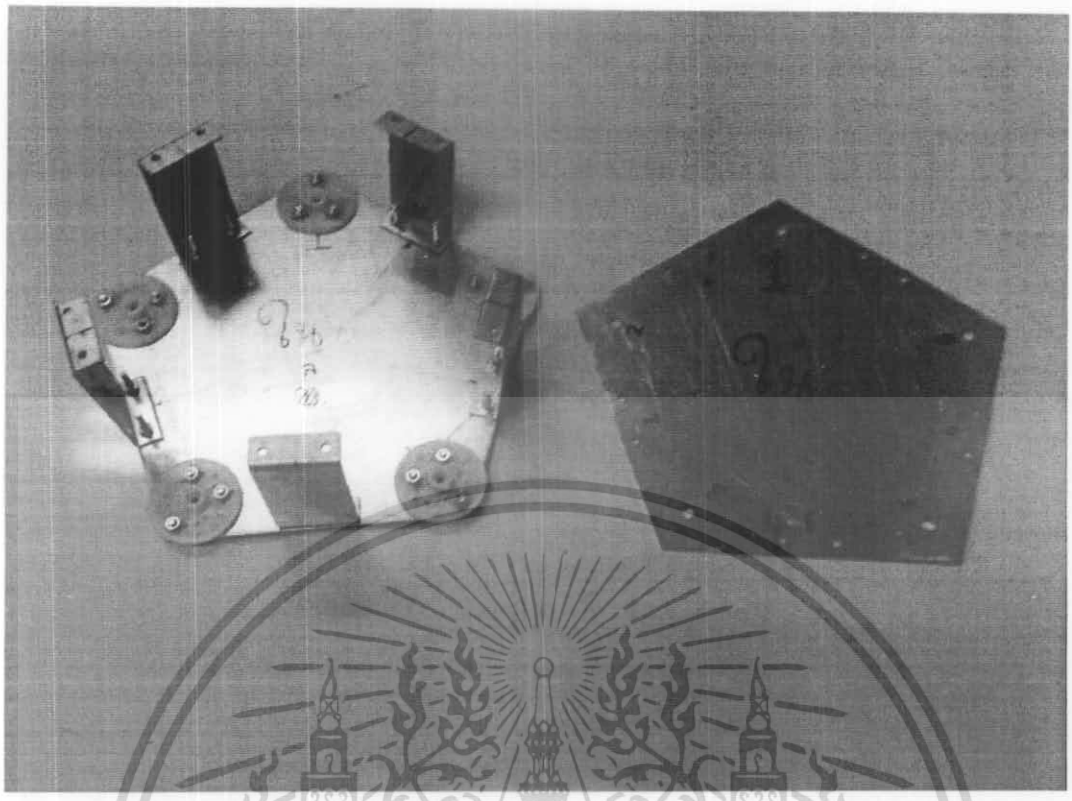


รูปที่ 4.15 แสดงขาที่สมบูรณ์หนึ่งข้าง

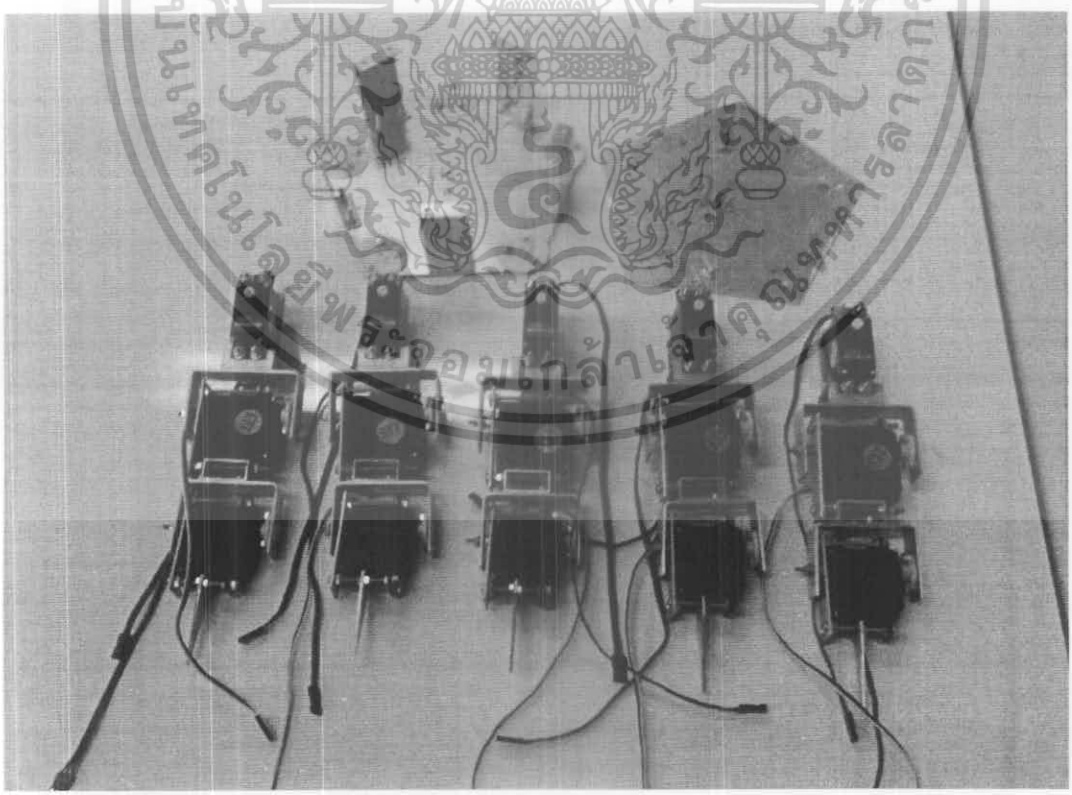


รูปที่ 4.16 แสดงการใช้จิ๊กช่วยในการเจาะรูตัวหุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 แสดงส่วนตัวที่พร้อมประกอบเข้ากับขาทั้งห้า



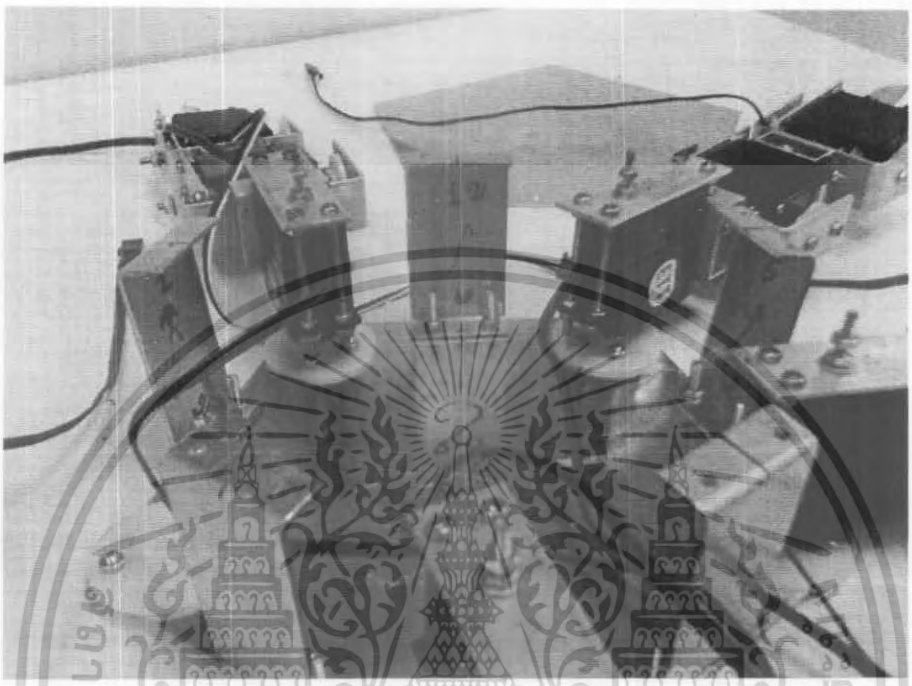
รูปที่ 4.18 แสดงส่วนประกอบทั้งหมดของตัวหุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

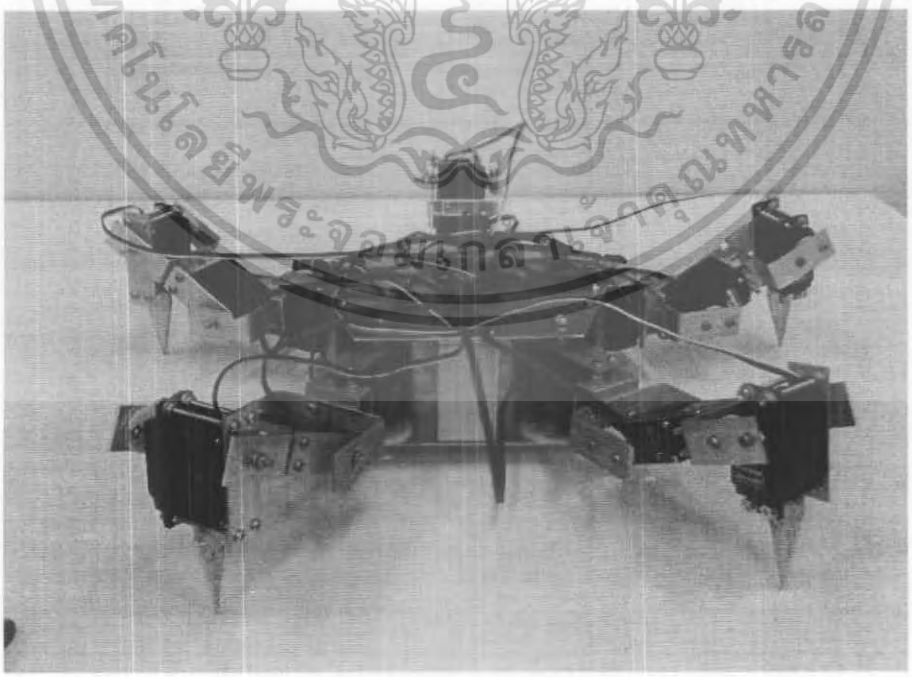
3. การประกอบชิ้นส่วนต่างๆ เข้าด้วยกัน

-ประกอบเซอร์โวมอเตอร์ให้หันไปทางเดียวกันเพื่อให้ง่ายต่อการเขียนโปรแกรม

- ใช้บุชทองเหลือง เพื่อทลดแรงเสียดทานของแต่ละข้อต่อ



รูปที่ 4.19 การประกอบตัวหุ่นเข้ากับขา



รูปที่ 4.20 ตัวหุ่นที่เสร็จสมบูรณ์

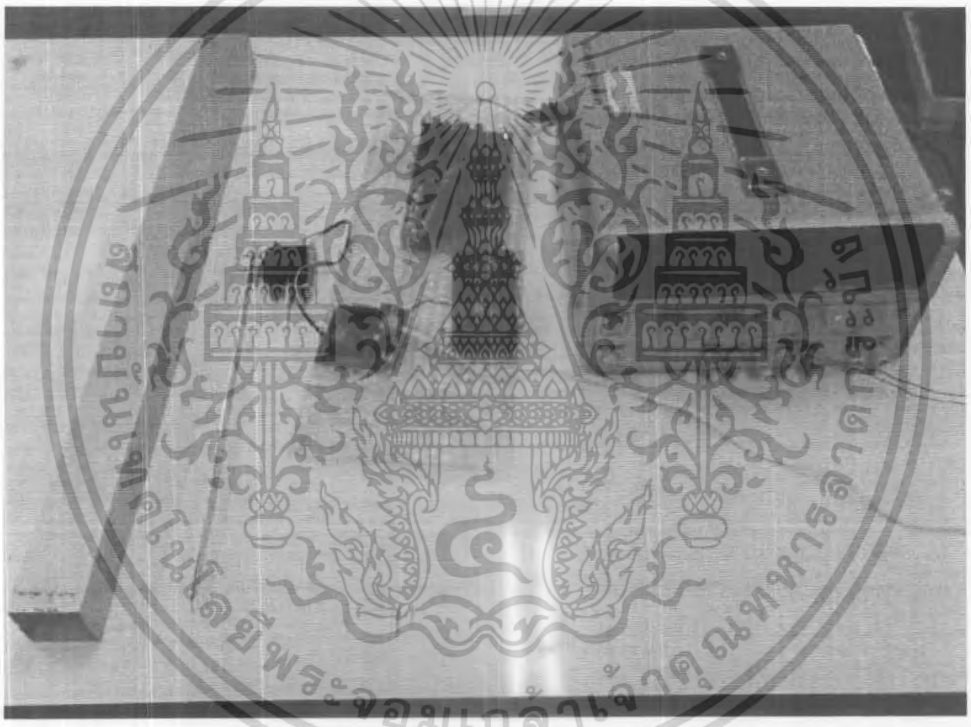
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 วิธีการและโปรแกรมที่ใช้ควบคุมหุ่น

จากบทที่ผ่านมาเราได้ทำการศึกษาถึงวิธีการควบคุม และการเขียนโปรแกรมไปแล้วดังนั้นในบทนี้เราก็จะทำการเขียนโปรแกรมควบคุมหุ่น โดยการใช้ภาษาซีแล้วสั่งผ่านพอร์ตอนุกรมไปยังบอร์ดขับ ZX-SERVO16 เพื่อขับเซอร์โวมอเตอร์ทั้ง 15 ตัว ดังนั้นเราต้องทำการหาตำแหน่งของ 0 และ 180 องศาเทียบกับตัวเลขที่ต้องป้อนให้ เพื่อเป็นการทดสอบการตอบสนองของเซอร์โวมอเตอร์ว่าตรงตามที่คู่มือบอกไว้หรือไม่

4.4.1 ทำการเทียบคำสั่งในโปรแกรม กับค่ามุมจริงที่เซอร์โวมอเตอร์หมุนไป

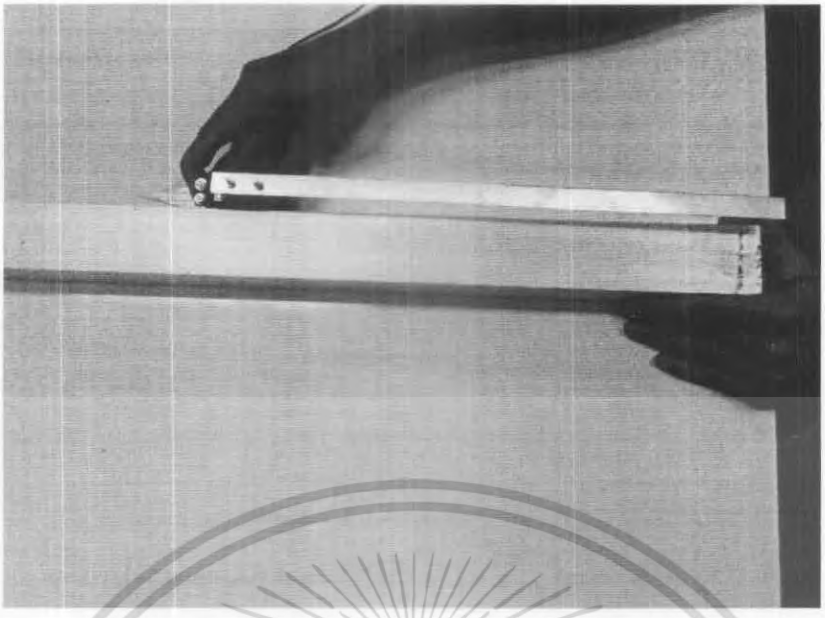
-โดยเริ่มจากการเตรียมแทงอลูมิเนียมยาวหนึ่งฟุต เพื่อใช้เป็นตัวทดสอบ และอลูมิเนียมกล่อง ความประมาณสองฟุต เพื่อใช้เป็นตัวสอบเทียบ ดังรูปที่ 4.23



รูปที่ 4.21 แสดงอุปกรณ์ที่ใช้ในการสอบเทียบการตอบสนอง

- จากนั้นก็ทำการทดสอบ โดยการเขียนโปรแกรมให้เซอร์โวมอเตอร์เริ่มหมุนไปที่ 0 องศา 180 องศา แล้วกลับมาที่ 0 องศา แต่ละที่จะหยุดค้างไวนานประมาณ 10 วินาที ช่วงระหว่างนี้เราก็ใช้สายตาเล็งเปรียบเทียบกับอะลูมิเนียมกล่อง ดังรูปที่ 3.27 แล้วทำการปรับค่าจนได้ค่าตามที่เราต้องการทุกตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

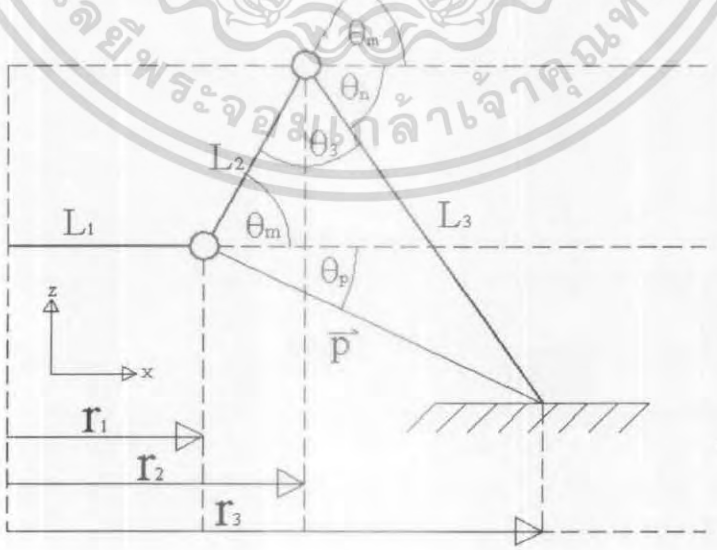


รูปที่ 4.22 แสดงการสอบเทียบที่ 0 องศา

- นำค่าที่ได้ไปคำนวณเพื่อสร้างคำสั่งควบคุมเซอร์โวมอเตอร์โดยกำหนดตำแหน่งที่หมุนเป็นองศา

4.4.2 การคำนวณ จลนศาสตร์ผกผันแบบกลุ่มของก้านโยง [6]

เป็นการคำนวณหามุมระหว่างข้อขาแต่ละข้อ เมื่อเราทราบ ขนาดของขาแต่ละข้อ และตำแหน่งของปลายขาในระบบ 3 มิติ เพื่อที่เราจะได้สามารถเขียนโปรแกรมจัดลักษณะท่าทางของขาของหุ่นให้ปลายขาอยู่ในตำแหน่งที่เราต้องการที่ลักษณะของขาห่นก็ต้องเหมาะสมด้วยเช่นกัน



รูปที่ 4.23 แสดงการกำหนดมุม และตัวแปรต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมมุติฐานในการคำนวณ

1. ชิ้นส่วนทุกส่วนของขาเป็นวัตุแข็งเกร็ง
2. ลำตัวของหุ่นขนานกับพื้นตลอดเวลา

การคำนวณ

จากรูปเราได้

$$r_1 = L_1 \tag{4.28}$$

$$r_3 = \sqrt{x^2 + y^2} \tag{4.29}$$

จากความสัมพันธ์ทางตรีโกณมิติจะได้

$$\theta_1 = \tan^{-1}\left(\frac{y}{x}\right) \tag{4.30}$$

$$\theta_p = \tan^{-1}\left(\frac{z}{r_3 - r_1}\right) \tag{4.31}$$

จะได้ขนาดของ $\bar{p}$

$$\left|\bar{p}\right| = \frac{(r_3 - r_1)}{\cos\theta_p} \tag{4.32}$$

จาก $$\left|\bar{p}\right|^2 + {L_2}^2 - 2\left|\bar{p}\right| L_2 \cos\left(\theta_p + \theta_m\right) = {L_3}^2 \tag{4.33}$$

จะได้ $$\left(\theta_p + \theta_m\right) = \cos^{-1}\left[\frac{-{L_3}^2 + \left|\bar{p}\right|^2 + {L_2}^2}{2x\left|\bar{p}\right| xL_2}\right] \tag{4.34}$$

$$\theta_2 = \left(\theta_p + \theta_m\right) - \theta_p + 90 \tag{4.35}$$

$$r_2 = L_1 + L_2 \cos\theta_m \tag{4.36}$$

จากคาวมสัมพัทธ์แบบสามเหลี่ยมมุมฉาก

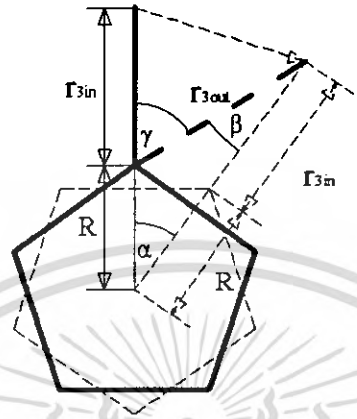
$$\theta_n = \tan^{-1}\left[\frac{\left(\left|\bar{p}\right|\sin\theta_p\right) - \left(L_2 \sin\theta_m\right)}{r_3 - r_2}\right] \tag{4.37}$$

จากรูปจะได้ $$\theta_3 = 180 - \theta_m - \theta_n \tag{4.38}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

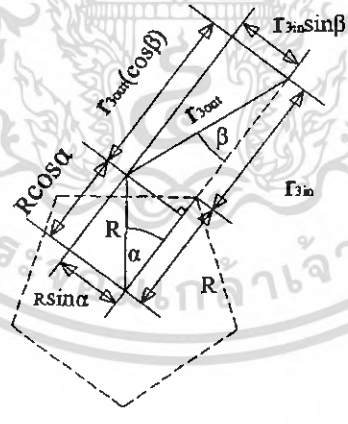
4.4.3 การหาสมการคณิตศาสตร์ เพื่อใช้ในการเขียนโปรแกรมควบคุมให้หุ่นหมุนตัว

เมื่อเราได้วิธีการเขียนโปรแกรมควบคุมหุ่นยนต์ให้เดินที่เหมาะสมแล้ว ต่อมาเราก็ต้องพัฒนาการเดิน เพื่อให้หุ่นสามารถหมุ่นตัวไปตามที่เราต้องการได้



รูปที่ 4.24 แสดงมุมที่เราต้องการเพื่อใช้ในการหมุนตัว

จากรูปมุมที่เราต้องการให้หุ่นหมุนตัวไป α องศาซึ่งเป็นมุมที่เรากำหนดค่าให้ และเราต้องหามุม γ ซึ่งเป็นมุมที่ขาหมุนและ r_{3out} ซึ่งเป็นระยะที่ขาก้าวออกไป โดยเราจะใช้มุม β เป็นมุมเชื่อมเพื่อหาค่า γ



รูปที่ 4.25 แสดงการหามุม β

ดังนั้นมุมที่ฐานของสามของสามเหลี่ยมหน้าจั่วมีค่า

จากรูปเราได้ว่า

$$R\sin\alpha = r_{3out}\sin\beta \tag{4.39}$$

$$R + r_{3in} = R\cos\alpha + r_{3out}\cos\beta \tag{4.40}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดรูปใหม่ได้

$$R(1-\cos\alpha)+r_{3in}=r_{3out}\cos\beta \tag{4.41}$$

(35)/(36) ได้

$$\tan\beta=\frac{R\sin\alpha}{R(1-\cos\alpha)+r_{3in}} \tag{4.42}$$

ดังนั้นเราได้

$$\beta=\tan^{-1}\left[\frac{R\sin\alpha}{R(1-\cos\alpha)+r_{3m}}\right] \tag{4.43}$$

ดังนั้นจากรูปที่ 7 เราได้ค่าของ γ

$$\gamma=\alpha+\beta \tag{4.44}$$

และจากรสมการที่ (4.39) เราจะได้

$$r_{3out}=\frac{R\sin\alpha}{\sin\beta} \tag{4.45}$$

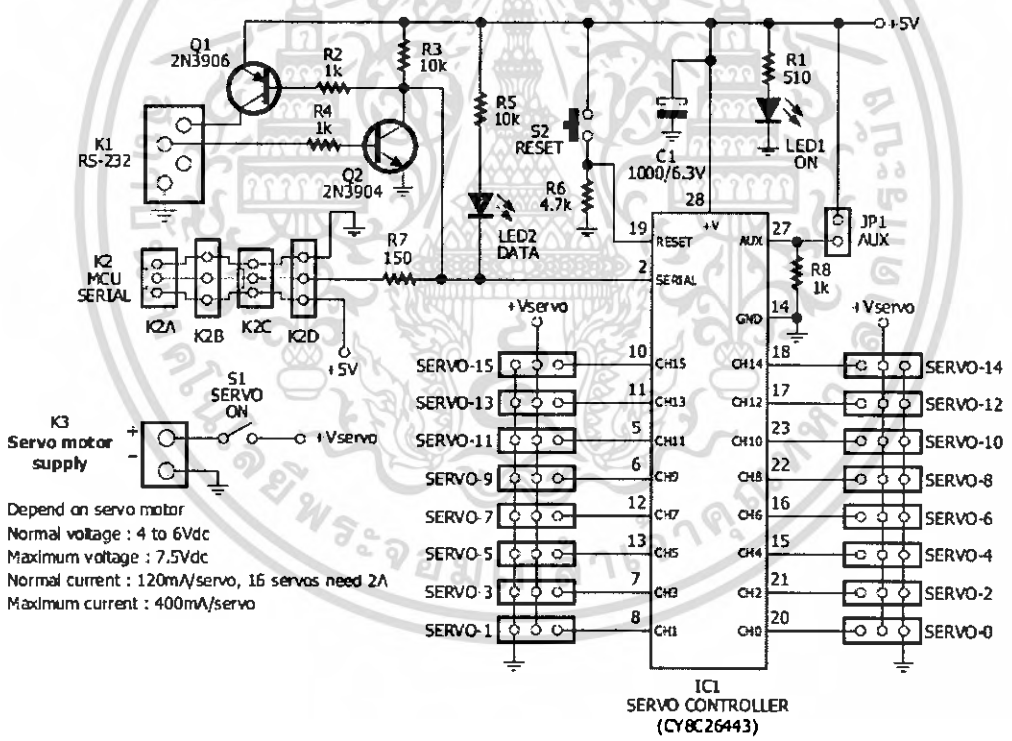
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การเชื่อมต่อและวิธีการใช้งานอุปกรณ์ต่างๆ

5.1 ส่วนประกอบหลักของหุ่นยนต์ห้าขา

5.1.1 ตัวหุ่นยนต์

1. เซอร์โวมอเตอร์รุ่น Futaba S3003 จำนวน 5 ตัวประกอบอยู่ที่ข้อต่อส่วนสุดท้ายของแต่ละขา
2. เซอร์โวมอเตอร์รุ่น S03T/STD/JR จำนวน 10 ตัวประกอบอยู่ที่ข้อต่อที่ 1 และ 2 ของแต่ละขา
3. บอร์ดควบคุมเซอร์โวมอเตอร์รุ่น ZX-SERVO16 จากบริษัท inex โดยบอร์ดนี้จะขับเซอร์โวทั้ง 15 ตัวจากหุ่นโดยตรงพร้อมๆกัน



รูปที่ 5.1 แสดงวงจรของบอร์ดควบคุมเซอร์โวมอเตอร์รุ่น ZX-SERVO16 [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 แหล่งจ่ายพลังงาน

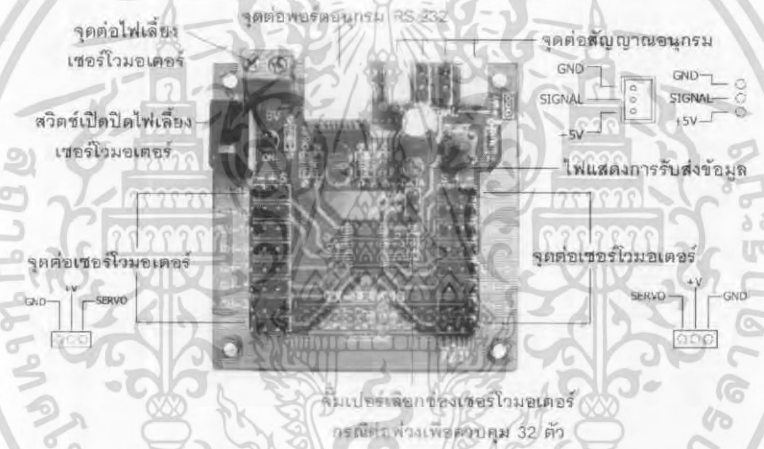
การใช้แหล่งจ่ายควรแยกแหล่งจ่ายไฟของบอร์ด ZX-SERVO16 ออกจากแหล่งจ่ายของเซอร์โวมอเตอร์ ไม่แนะนำให้ใช้แหล่งพลังงานเดียวกันเพราะจะทำได้เกิดการดึงไฟของเซอร์โวมอเตอร์ได้ โดยใช้พาวเวอร์ซัพพลายจ่ายไฟ 12 โวล์ทไปเข้าวงจรเร็กกูเลเตอร์เพื่อแปลงเป็น 5 โวล์ท สำหรับเลี้ยงบอร์ด ZX-SERVO16 และใช้แบ็ตเตอรี 6 โวล์ทขับเซอร์โวมอเตอร์

5.1.3 คอมพิวเตอร์และโปรแกรม

คอมพิวเตอร์ที่ใช้ต้องมีพอร์ตอนุกรม เพื่อติดต่อสื่อสารกับบอร์ด ZX-SERVO16โดยโปรแกรมที่ใช้ควบคุมหุ่นยนต์คือ ภาษา C และ คอมพิวเตอร์ที่ใช้ต้องมีระบบปฏิบัติการวินโดวส์

5.1.4 การเชื่อมต่ออุปกรณ์และการสั่งการ

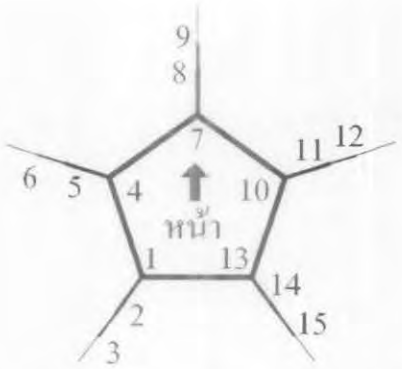
1. ปิดสวิตช์ไฟเลี้ยงเซอร์โวมอเตอร์ และ ของพาวเวอร์ซัพพลายเพื่อป้องกันไฟเกินเข้าบอร์ดขณะต่อสายไฟ



รูปที่ 5.2 บอร์ด ZX-SERVO16 [3]

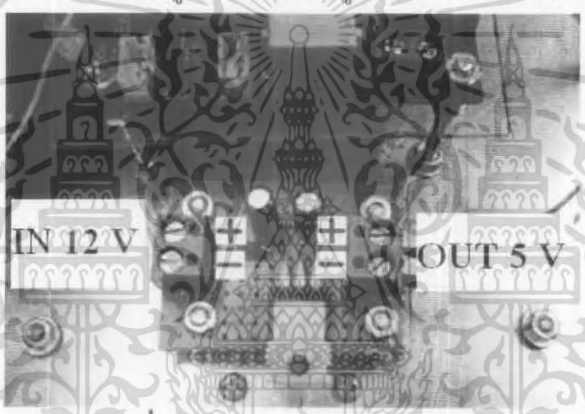
2. ต่อสายไฟเลี้ยงเซอร์โวมอเตอร์จากแบ็ตเตอรี 6 โวล์ท เข้าจุดต่อไฟเลี้ยงเซอร์โวมอเตอร์ "ห้ามต่อกลับขั้วเด็ดขาด"
3. ต่อสายไฟของเซอร์โวมอเตอร์เข้าบอร์ต ZX-SERVO16 ตามลำดับดังรูปที่ 3 จากพอร์ต 1 – 15 โดยเว้นพอร์ต 0 เอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



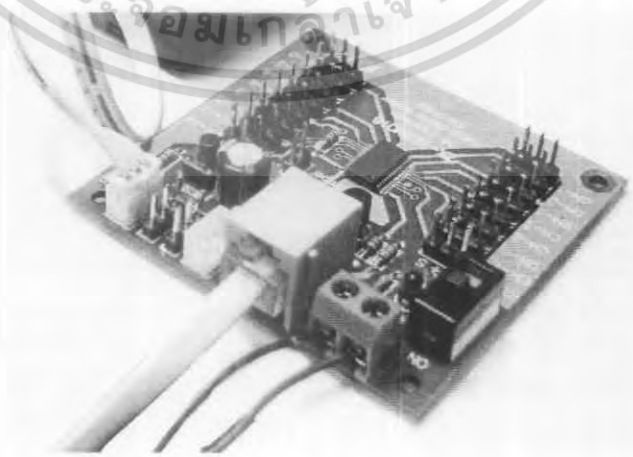
รูปที่ 5.3 แสดงลำดับของเซอร์โวมอเตอร์ที่ต้องต่อเข้าบอร์ด

3. ต่อสายไฟเลี้ยง 12 โวลท์ จากพาวเวอร์ซัพพลายเข้าเร็กกูเลเตอร์โดยฝั่งซ้ายมือจะเป็นทางเข้าของไฟ 12 โวล์ทจากพาวเวอร์ซัพพลาย ซึ่งการต่อให้ต่อขั้วบวกด้านบนและขั้วลบด้านล่าง โดยฝั่งขวามือจะเป็นไฟ5โวล์ทออกจากเร็กกูเลเตอร์ไปเข้าสู่บอร์ดZX-SERVO16เพื่อเป็นไฟเลี้ยงวงจร



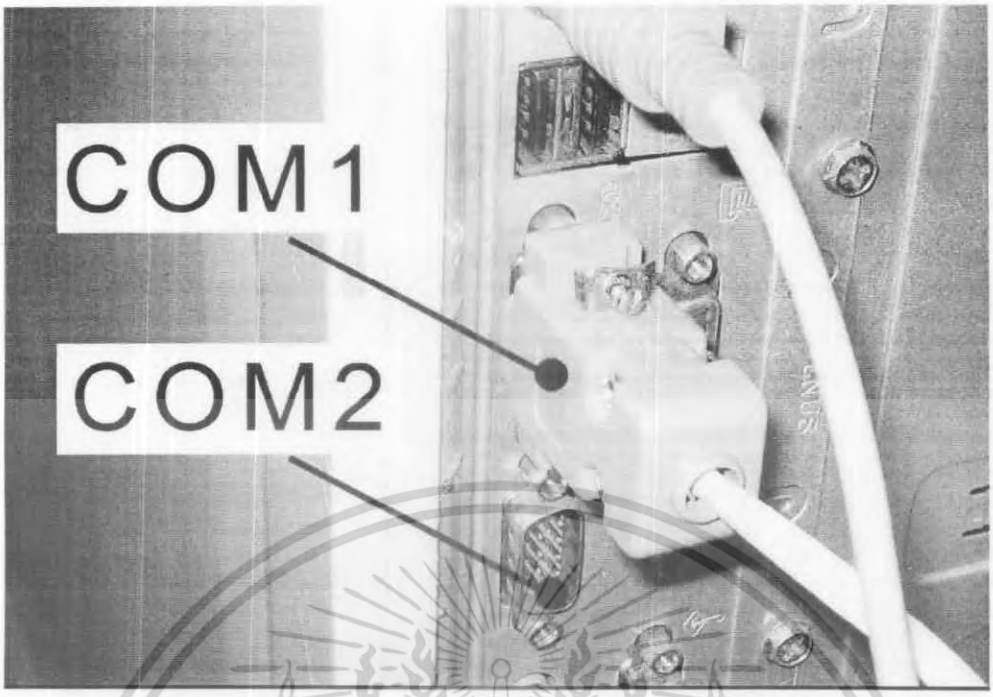
รูปที่ 5.4 แสดงบอร์ดเร็กกูเลเตอร์

4. ต่อสายอนุกรมที่ "จุดต่อพอร์ตอนุกรม RS-232" ของบอร์ด Servo16 เชื่อมต่อไปยังพอร์ตอนุกรมของคอมพิวเตอร์ ดังรูปที่ 5.5 และ รูปที่ 5.6



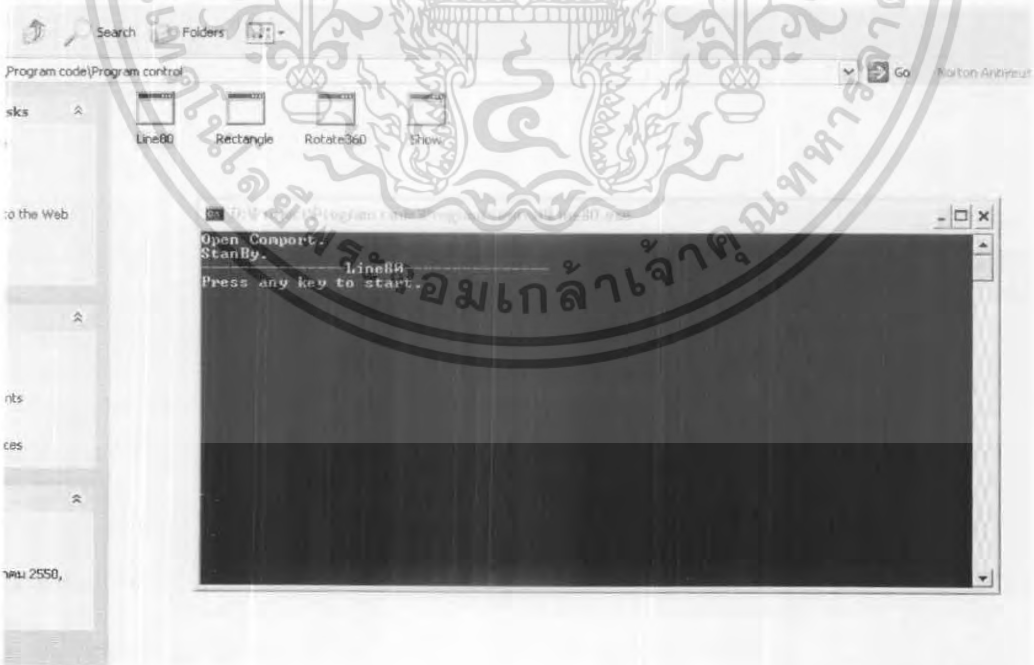
รูปที่ 5.5 แสดงการเชื่อมต่อบนบอร์ด ZX-SERVO16 [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 การเชื่อมต่อไปยังพอร์ตอนุกรม ของคอมพิวเตอร์

6. เปิดสวิตช์ไฟเลี้ยงเซอร์โวมอเตอร์ แล้วกดปุ่มรีเซ็ต เพื่อให้บอร์ดพร้อมรับค่าจากคอมพิวเตอร์
7. เปิดโปรแกรมที่ต้องการ เช่น โปรแกรม Line80 จะปรากฏหน้าต่างสีดำดังรูปที่ 5.6 และมีข้อความว่า Press any key to start.



รูปที่ 5.7 หน้าต่างโปรแกรม Line80

8. กดปุ่มใดๆก็ได้บนคีย์บอร์ด โปรแกรมจะทำงานทันที

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.8 หน้าต่างการรันโปรแกรม

9. การแก้ไขสิ่งผิดปกติ

- หากหุ่นยนต์นิ่งไม่ขยับเลยให้ตรวจสอบว่าพอร์ตอนุกรมในคอมพิวเตอร์ถูกโปรแกรมอื่นใช้อยู่หรือไม่หากพบให้ปิดโปรแกรมนั้นก่อนแล้วเริ่มโปรแกรมใหม่ หากพอร์ตอนุกรมยังไม่ถูกใช้ให้ตรวจสอบที่ขั้วไฟของทั้งแหล่งจ่ายและที่ตัวหุ่นว่ามีไฟเข้าหรือไม่
- หากหุ่นยนต์มีการบิดขาไปด้านใดด้านหนึ่งอยู่ตลอดให้รีบปิดสวิชต์แหล่งจ่ายทันทีเพื่อป้องกันการเสียหายที่มอเตอร์ แล้วทำการถอดขั้วสายไฟแล้วต่อใหม่
- หากหุ่นยนต์เดินตามที่สั่งแต่ก้าวขาไม่สุดหรือขาไม่มีแรงให้ตรวจสอบว่าแบ็ตเตอร์รีหมดหรือไม่

10. ข้อควรคำนึง

- ควรหยอดน้ำมันหล่อลื่นที่ข้อต่อทองเหลืองของหุ่นอยู่เสมอ
- ไม่ควรให้หุ่นต้องรับภาระหนักเป็นเวลานานเพราะจะทำให้เซอร์โวมอเตอร์เกิดความร้อนและพังในที่สุด
- ควรตรวจสอบการต่อสายไฟว่าถูกต้องหรือไม่ก่อนเปิดสวิตช์ เพราะการต่อสายไฟพลาดนั้นจะทำให้วงจรพังทันที
- "ไม่ควรเปลี่ยนสายไฟที่ใช้เชื่อมต่อหุ่นยนต์"เพราะขนาดของสายไฟมี "ผลโดยตรง" ต่อการสูญเสียพลังงานระหว่างหุ่นยนต์กับแหล่งจ่าย หากมีความจำเป็นต้องเปลี่ยนควรใช้สายไฟที่มีขนาดเท่ากับ หรือใหญ่กว่าสายไฟที่ใช้อยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

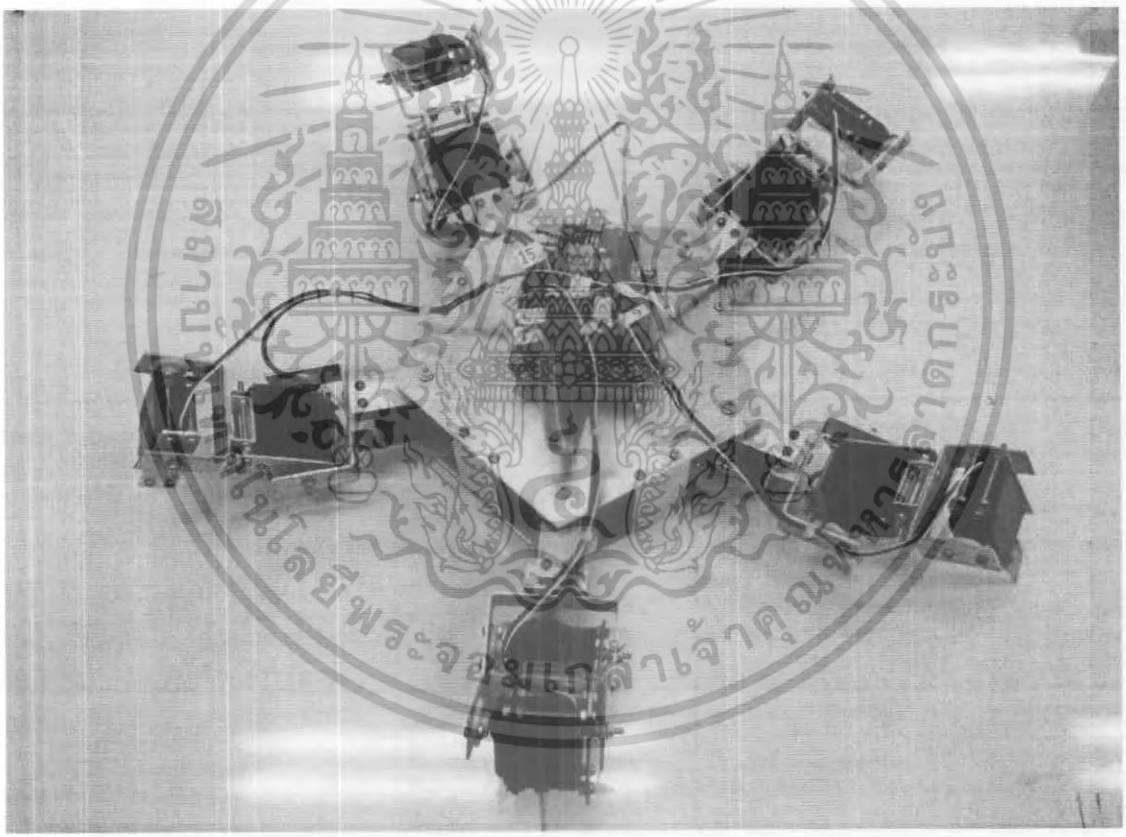
บทที่ 6

การทดลองและผลการทดลอง

หลังจากทำการสร้างหุ่น และเขียนโปรแกรมเสร็จเรียบร้อยแล้ว ขั้นตอนต่อมาก็คือการทดสอบประสทธิภาพของหุ่นยนต์ เพื่อนำข้อมูลที่ได้มาวิเคราะห์หาประสทธิภาพของหุ่นยนต์ โดยเราได้ทำการเตรียมอุปกรณ์เพื่อใช้ในการทดลองทั้ง 3 การทดลองดั้งต่อไปนี้

6.1 อุปกรณ์การทดลอง

1. ตัวหุ่นยนต์ห้าขาต้นแบบขนาดเล็ก ใช้ในการทดสอบประสิทธิภาพ



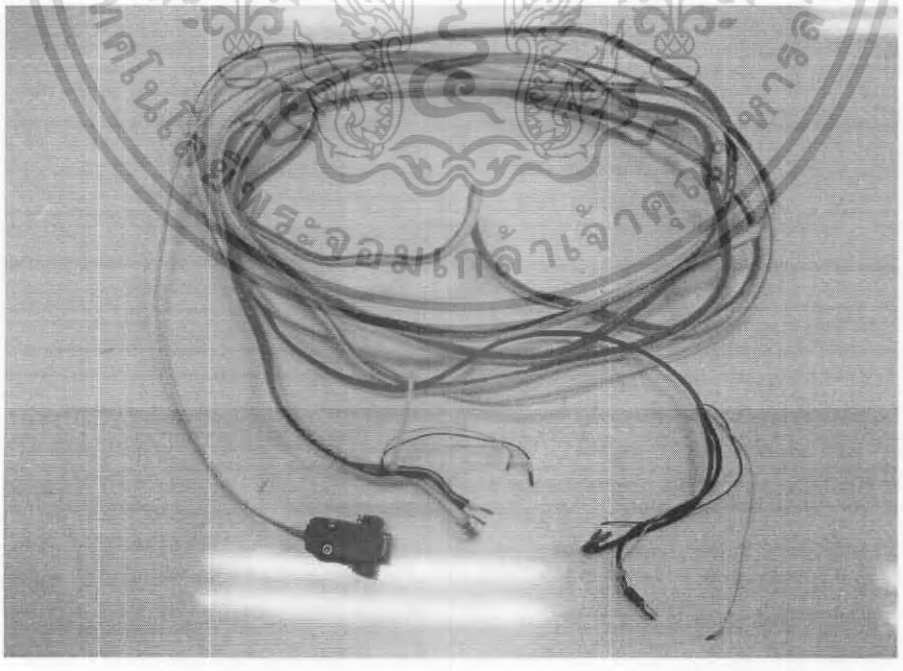
รูปที่ 6.1 แสดงหุ่นยนต์ต้นแบบห้าขาขนาดเล็กที่ใช้ทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คอมพิวเตอร์ เป็นส่วนคำนวณ และสั่งการให้หุ่นเดินตามที่เราต้องการ

รูปที่ 6.2 แสดงคอมพิวเตอร์ที่ใช้ควบคุมหุ่น

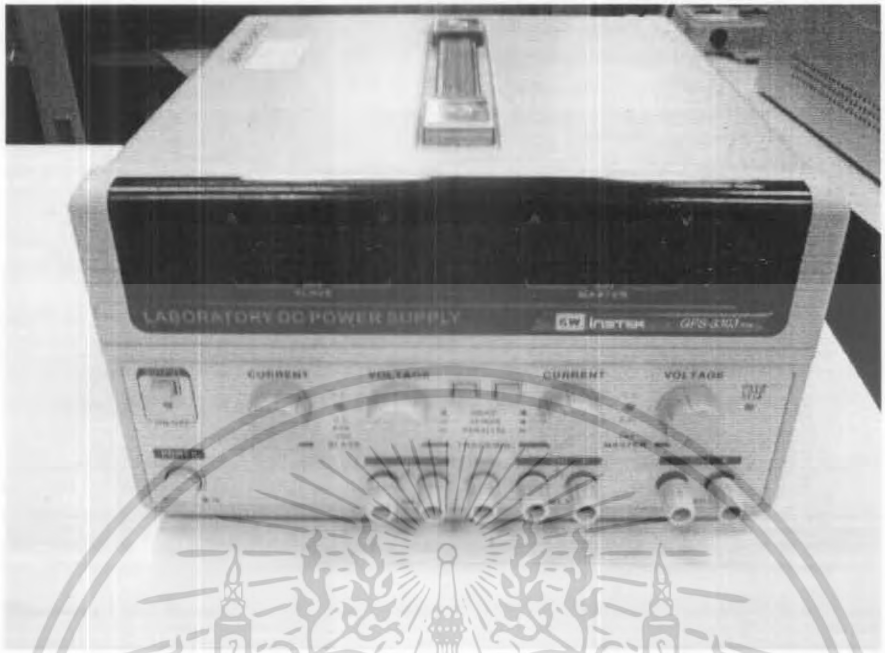
3 .สายต่อพ่วง เป็นสายสัญญาณที่ส่งคำสั่งไปยังตัวหุ่น และสายไปที่จ่ายไฟไปเลี้ยงเซอร์โวมอเตอร์ และบอร์ด ZX-SERVO 16



รูปที่ 6.3 แสดงสายต่อพ่วงที่ใช้ควบคุมหุ่น และจ่ายไฟเลี้ยง

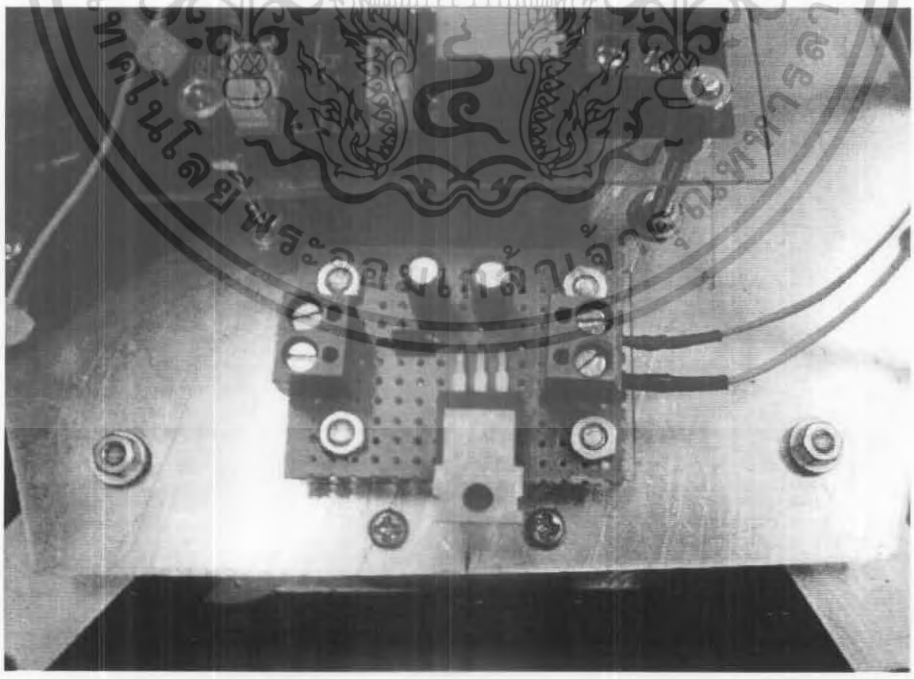
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. DC Power Supply เป็นตัวจ่ายไฟ 12V ไปยังบอร์ด Recgulater



รูปที่ 6.4 แสดงพาวเวอร์ ซัพพลาย

5. บอร์ด Recgulater เป็นตัวแปลงไฟ 12V ไปเป็น 5V ไปเลี้ยงวงจร



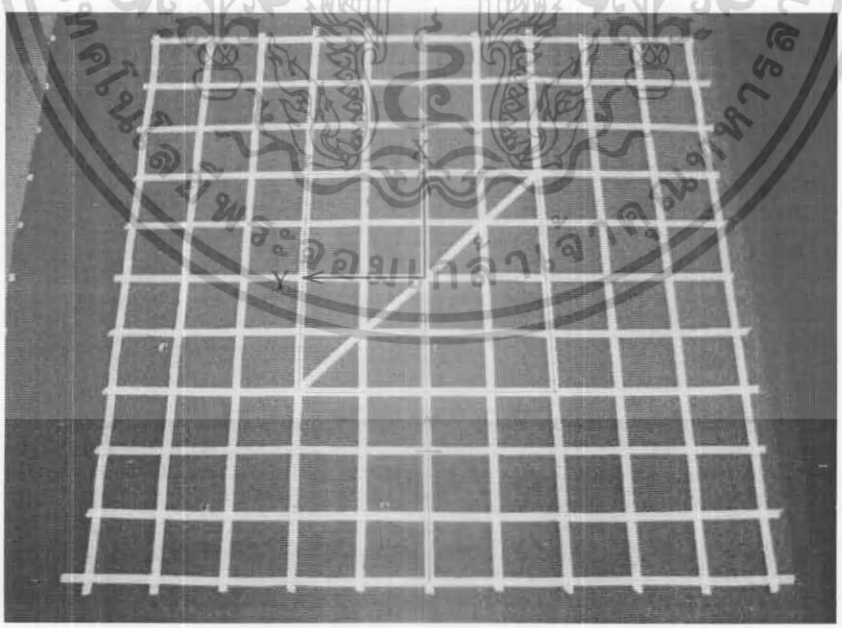
รูปที่ 6.5 แสดงเรกกูเลเตอร์แปลงไฟจาก 12 โวล์ท เป็น 5 โวล์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. แบทเตอร์รี่ เป็นตัวจ่ายไป 6V ไปเลี้ยงเซอร์โวมอเตอร์ หรืออาจใช้จาก Power Supply

รูปที่ 6.6 แสดงแบทเตอร์รี่ที่ใช้จ่ายไฟให้กับเซอร์โวมอเตอร์

7. พื้นที่ใช้ทดสอบ โดยใช้พื้นที่ขนาด 100 x 100 เซ็นติเมตร แล้วทำการตีตารางโดยแบ่งช่องละ 10 เซ็นติเมตร

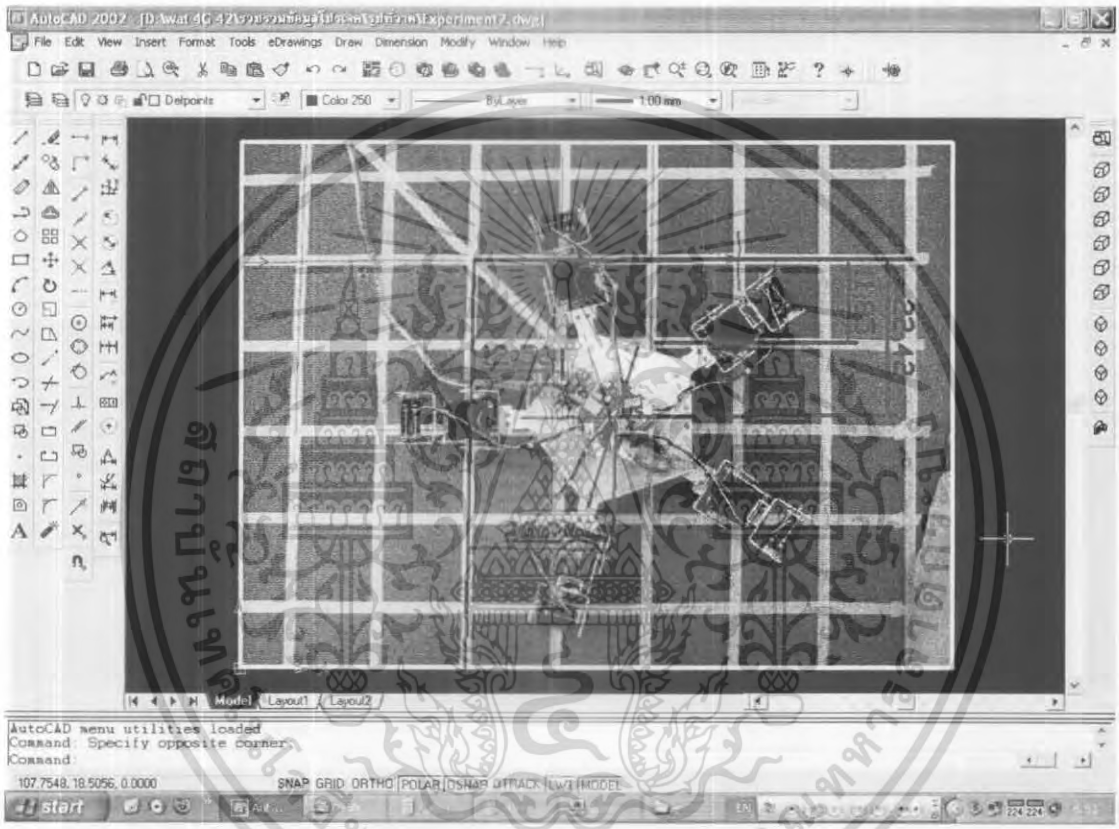


รูปที่ 6.7 แสดงสนามที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การอ่านค่าผลการทดลอง

เนื่องจากการหุ่นยนต์ที่ทดลองมีรูปร่างที่ลำบากต่อการเก็บค่าตำแหน่งของค่าจุดศูนย์กลางหุ่น และการอ่านค่าโดยตรงนั้นมีค่าผิดพลาดสูง จึงได้ทำการอ่านค่า โดยนำวีดีโอที่ถ่ายการทดลองในแนวตั้งฉาก(สูงจากสนามทดลอง 2.4 มตร) ไปทำการจับภาพ เพื่อดึงเอาภาพของการเคลื่อนที่แต่ละรอบการเดินออกมาวิเคราะห์ ด้วยโปรแกรม Auto CAD ดังรูปที่ 6.8 ซึ่งทำให้สะดวก รวดเร็ว และอ่านค่าได้ง่าย อีกทั้งยังแม่นยำกว่าการวัดด้วยตาเปล่า ซึ่งมีวิธีการหาค่าดังนี้



รูปที่ 6.8 แสดงการอ่านค่าผลการทดลอง

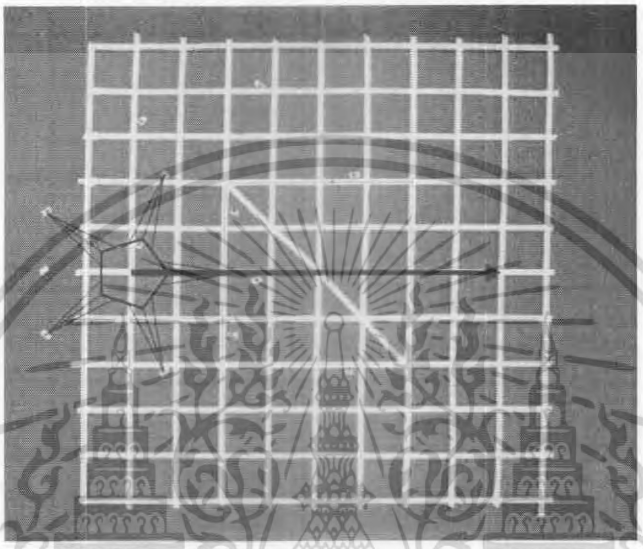
1. สร้างเส้นสมมุติของแกน X และแกน Y ขึ้นมา เพื่ออ้างอิงจุดเริ่มต้น
2. วัดระยะห่างของเส้นตารางในแนวแกน X และแนวแกน Y
3. หาจุดศูนย์กลางของตัวหุ่นด้วยการลากเส้น อย่างน้อย 2 เส้นจากมุมมาตั้งฉากกับด้านตรงข้ามมุม แล้วเราจะได้จุดศูนย์ลำตัวหุ่นยนต์
4. วัดความห่างของจุดหุ่นยนต์มวลกับแกนอ้างอิง X-Y ที่สร้างขึ้นมาในข้อ 1
5. นำค่าที่ได้จากข้อ 4 ไปหารกับระยะห่างของตารางที่ได้จากข้อ 2 แล้วนำไปคูณกับ 100 มิลลิเมตร ซึ่งเป็นค่าความยาวจริงของระยะห่างตาราง ก็จะได้ระยะห่างจากจุดศูนย์กลางกับแกนอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การทดลองเดินเป็นเส้นตรง

6.3.1 วิธีทำการทดลอง

การทดลองนี้จะให้หุ่นเดินจากจุด (-400, 0) ไปยัง (400, 0) ระยะทาง 800 มิลลิเมตร ดังรูปที่ 6.9 แล้วทำการถ่ายวีดีโอในแนวตั้งฉากสูงจากพื้นสนาม 240 เซ็นติเมตร จำนวน 5 รอบ จากนั้นนำเอาวีดีโอการเดินของแต่ละรอบมาทำการจับภาพ 11 ครั้งต่อรอบ แล้วนำมาทำการวิเคราะห์ด้วยโปรแกรม AutoCAD เพื่อทำการหาค่า X-Y ของตัวหุ่น ในแต่ละก้าว



รูปที่ 6.9 แสดงเส้นทางการเดินเป็นเส้นตรงของหุ่นยนต์

6.3.2 ผลการทดลองการเดินเป็นเส้นตรง

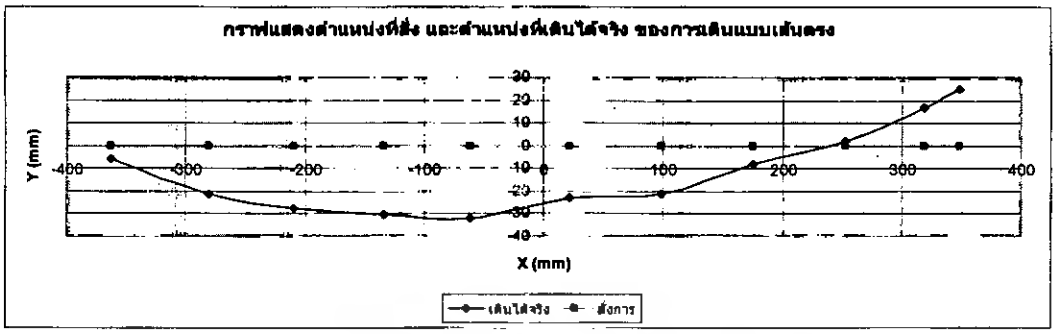
เมื่อหาค่า X-Y จากการจับภาพจากวีดีโอแล้วนำไปบันทึกผลลงในตารางดังรูปที่ 6.1 ได้ค่าดังนี้

ครั้งที่1		ครั้งที่2		ครั้งที่3		ครั้งที่4		ครั้งที่5		ค่าเฉลี่ย	
X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
-369	-15	-364	-2	-351	-13	-367	0	-361	0	-362.4	-6
-281	-20	-287	-20	-279	-28	-279	-16	-276	-21	-280.4	-21
-215	-26	-217	-26	-214	-33	-200	-28	-198	-24	-208.8	-27.4
-136	-29	-145	-28	-129	-33	-126	-28	-133	-34	-133.8	-30.4
-63	-38	-78	-27	-62	-36	-48	-25	-56	-35	-61.4	-32.2
5	-25	12	-24	27	-27	34	-20	29	-19	21.4	-23
93	-20	80	-24	110	-21	107	-18	101	-24	98.2	-21.4
171	-10	164	0	179	-9	189	-10	175	-12	175.6	-8.2
250	-8	231	14	255	0	260	0	272	5	253.6	2.2
320	11	302	30	326	20	328	8	317	16	318.6	17
336	19	332	38	356	22	361	19	357	25	348.4	24.6

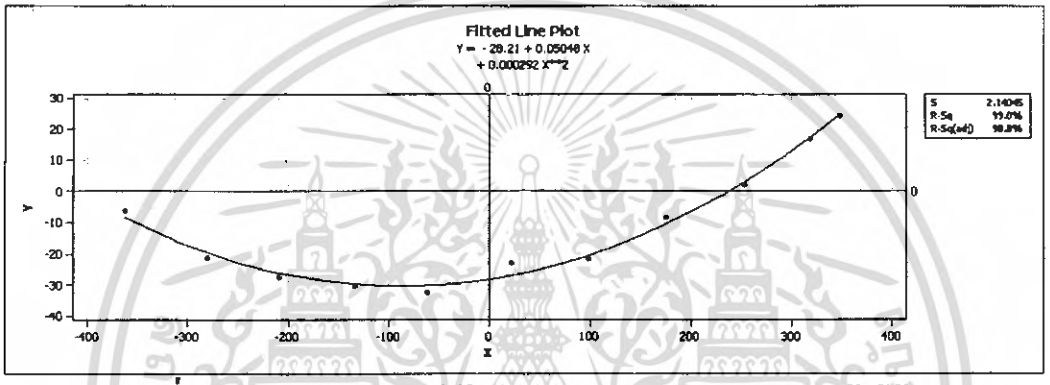
ตารางที่ 6.1 แสดงค่า X-Y จากการทดลองให้หุ่นยนต์เดินเป็นเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นนำค่าค่าเฉลี่ยของ X-Y จากตารางที่ 6.1 มาสร้างกราฟเพื่อดูลักษณะการเปลี่ยนตำแหน่งของหุ่นยนต์ จะได้กราฟดังรูปที่ 6.10



รูปที่ 6.10 แสดงผลการเดินของหุ่น



รูปที่ 6.11 แสดงผลจากการใช้โปรแกรม MINITAB หาสมการเส้นโค้ง

6.3.3 สรุปผลการทดลองการเดินเป็นเส้นตรง

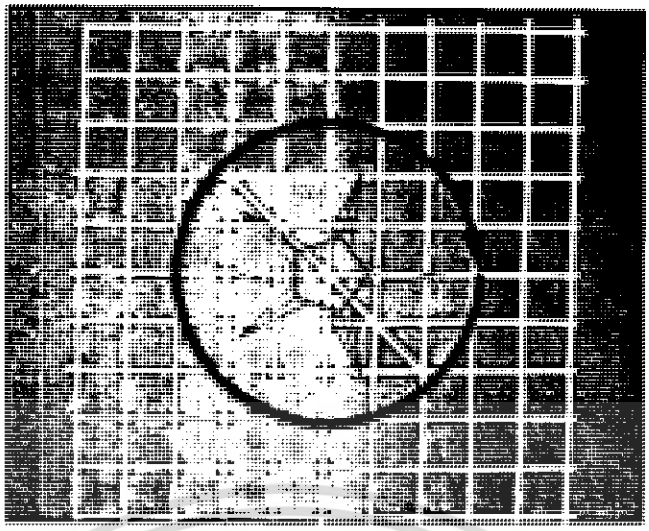
การเดินเป็นเส้นตรง จากรูป 6.10 และ รูปที่ 6.11 จะเห็นได้ว่าหุ่นเราจะมีการเดินเป็นเส้นโค้งโดยมีสมการเป็น $Y= -28.21 + 0.05X + 0.00029X^2$ ซึ่งโดยรวมแล้วหุ่นยนต์ของเรายังเดินได้ไม่ตรงเท่าที่ควร ยังคงมีการโค้งไปมา แต่หุ่นยนต์ก็ยังสามารถเคลื่อนตัวไปข้างหน้าได้อย่างต่อเนื่อง

6.4 การทดลองการเลี้ยว

6.4.1 วิธีทำการทดลอง

การทดลองนี้จะให้หุ่นยนต์เดินหมุนตัวอยู่กับที่ตามเข็มนาฬิกา 360 ตรงตำแหน่ง (0,0) ดังรูปที่ 6.10 แล้วทำการถ่ายวีดีโอในแนวตั้งฉากสูงจากพื้นสนาม 240 เซ็นติเมตร จำนวน 5 รอบ จากนั้นนำเอาวีดีโอการเดินของแต่ละรอบมาทำการจับภาพ 24 ครั้งต่อรอบ แล้วนำมาทำการวิเคราะห์ด้วยโปรแกรม Auto CAD เพื่อทำการหาค่า X-Y ของตัวหุ่น ในแต่ละก้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.12 แสดงเส้นทางการลักษณะการหมุนตัว

6.4.2 ผลการทดลองการเลี้ยว

เมื่อหาค่า X-Y และมุมที่หุ่นยนต์หมุนได้จริง จากการจับภาพจากวีดีโอแล้วนำไปบันทึกผลลงในตารางดังรูปที่ 6.1 ได้ค่าดังนี้

ภาพที่	รอบที่ 1			รอบที่ 2		
	X	Y	มุมที่หมุน	X	Y	มุมที่หมุน
1	0.113208	-0.18099	-14	0.061526	-0.2	-14
2	0.041838	-0.28678	-28	0.041017	-0.31818	-29
3	-0.05578	-0.36694	-43	-0.0082	-0.43388	-44
4	-0.07793	-0.4686	-57	-0.07383	-0.44215	-58
5	0.179655	-0.54298	-72	-0.15423	-0.52479	-72
6	-0.30107	-0.54628	-88	-0.27482	-0.50826	-88
7	-0.39377	-0.56529	-102	-0.32732	-0.54215	-103
8	-0.47908	-0.50248	-117	-0.41099	-0.52231	-117
9	-0.59393	-0.46942	-132	-0.50123	-0.46612	-131
10	-0.64807	-0.38926	-147	-0.54963	-0.40413	-147
11	-0.66776	-0.29174	-161	-0.55619	-0.3124	-161
12	-0.73175	-0.23388	-176	-0.60788	-0.24959	-175
13	-0.67432	-0.1843	-190	-0.5767	-0.17355	-191
14	-0.65546	-0.1438	-205	-0.50205	-0.12645	-204
15	-0.57998	-0.07273	-220	-0.45283	-0.07273	-218
16	0.466776	-0.01653	-235	-0.39295	-0.02562	-231
17	-0.35603	0.024793	-249	-0.26907	0.002479	-248
18	-0.26333	0.047107	-263	-0.14356	0.001653	-262
19	-0.17637	0.028099	-279	-0.0484	0.008264	-278
20	-0.05004	-0.01818	-295	0.064807	-0.01653	-292
21	0.037736	-0.1	-308	0.14274	-0.04793	-306
22	0.084495	-0.18347	-324	0.21329	-0.15207	-322
23	0.156686	-0.2562	-337	0.272354	-0.2281	-337
24	0.228876	-0.36777	-352	0.360131	-0.36116	-353

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

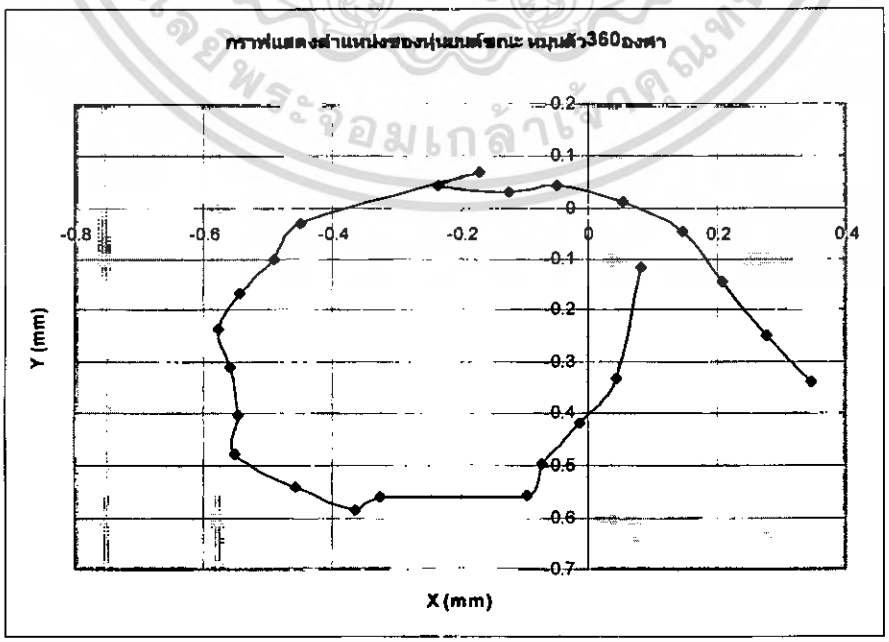
ภาพที่	รอบที่ 3			รอบที่ 4		
	X	Y	มุมที่หมุน	X	Y	มุมที่หมุน
1	0.07137	-0.18099	-12	0.077933	-0.20909	-14
2	0.034454	-0.29339	-28	0.053322	-0.34463	-28
3	0.004922	-0.43636	-43	-0.04922	-0.38926	-44
4	-0.04758	-0.5438	-56	-0.137	-0.4719	-56
5	-0.14028	-0.58512	-72	-0.21821	-0.56529	-71
6	-0.24692	-0.59835	87	-0.55209	-0.57273	-87
7	-0.34619	-0.61984	-101	-0.39705	-0.66777	-101
8	-0.44463	-0.58678	-115	-0.50205	-0.62149	-118
9	-0.52748	-0.51488	-130	-0.57424	-0.53554	-132
10	-0.57178	-0.45289	-145	-0.32732	-0.45537	-148
11	-0.57506	-0.35124	-159	-0.34208	-0.38678	-161
12	-0.58819	-0.26116	-173	-0.30763	-0.29256	-176
13	-0.5726	-0.1686	-189	-0.2781	-0.23636	-191
14	-0.55127	-0.11653	-203	-0.16653	-0.14793	-205
15	-0.46103	-0.0281	-219	-0.22724	-0.05455	-220
16	-0.38802	0.068595	-234	-0.10172	0.02562	-234
17	-0.28712	0.093388	-247	0.028712	-0.01736	-248
18	-0.15751	0.071074	-262	0.130435	-0.00826	-264
19	0.067268	0.060331	-278	0.005742	0.068595	-279
20	0.044299	0.004132	-293	0.134537	0.068595	-295
21	0.141099	-0.05868	-307	0.228876	0.041322	-310
22	0.200164	-0.1595	-321	0.302707	-0.06033	-325
23	0.306809	-0.23141	-335	0.347006	-0.22066	-339
24	0.378999	-0.31157	-351	0.388843	-0.35041	-355

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่	รอบที่ 5			ค่าเฉลี่ย		
	X	Y	มุมที่หมุน	X	Y	มุมที่หมุน
1	0.087777	0.190909	-13	0.082363	-0.11603	-13.4
2	0.04676	-0.41901	-28	0.043478	-0.3324	-28.2
3	0.040197	-0.46612	-42	-0.01362	-0.41851	-43.2
4	-0.03199	-0.55124	-56	-0.07367	-0.49554	-56.6
5	-0.15012	-0.56116	-71	-0.09664	-0.55587	-71.6
6	-0.24856	-0.56612	-86	-0.32469	-0.55835	-52.4
7	-0.34783	-0.5281	-48	-0.36243	-0.58463	-91
8	-0.44381	-0.46446	-116	-0.45611	-0.5395	-116.6
9	-0.55866	-0.4	-131	-0.55111	-0.47719	-131.2
10	-0.63413	-0.31488	-143	-0.54619	-0.40331	-146
11	-0.65546	-0.21653	-159	-0.55931	-0.31174	-160.2
12	-0.64151	-0.14463	-174	-0.57539	-0.23636	-174.8
13	-0.61854	-0.07273	-189	-0.54405	-0.16711	-190
14	-0.57096	0.027273	-203	-0.48925	-0.10149	-204
15	-0.51272	0.07438	-217	-0.44676	-0.03074	-218.8
16	-0.43806	0.290083	-231	-0.1708	0.06843	-233
17	-0.28794	0.112397	-243	-0.23429	0.04314	-247
18	-0.19196	0.050413	-259	-0.12518	0.032397	-262
19	-0.09598	0.058678	-277	-0.04955	0.044793	-278.2
20	0.068089	0.02562	-293	0.052338	0.012727	-293.6
21	0.171452	-0.06033	-305	0.144381	-0.04512	-307.2
22	0.2379	-0.16942	-321	0.207711	-0.14496	-322.6
23	0.295324	-0.30661	-336	0.275636	-0.2486	-336.8
24	0.368335	-0.30661	-351	0.345037	-0.3395	-352.4

ตารางที่ 6.2 แสดงผลการทดลองของการเลี้ยว

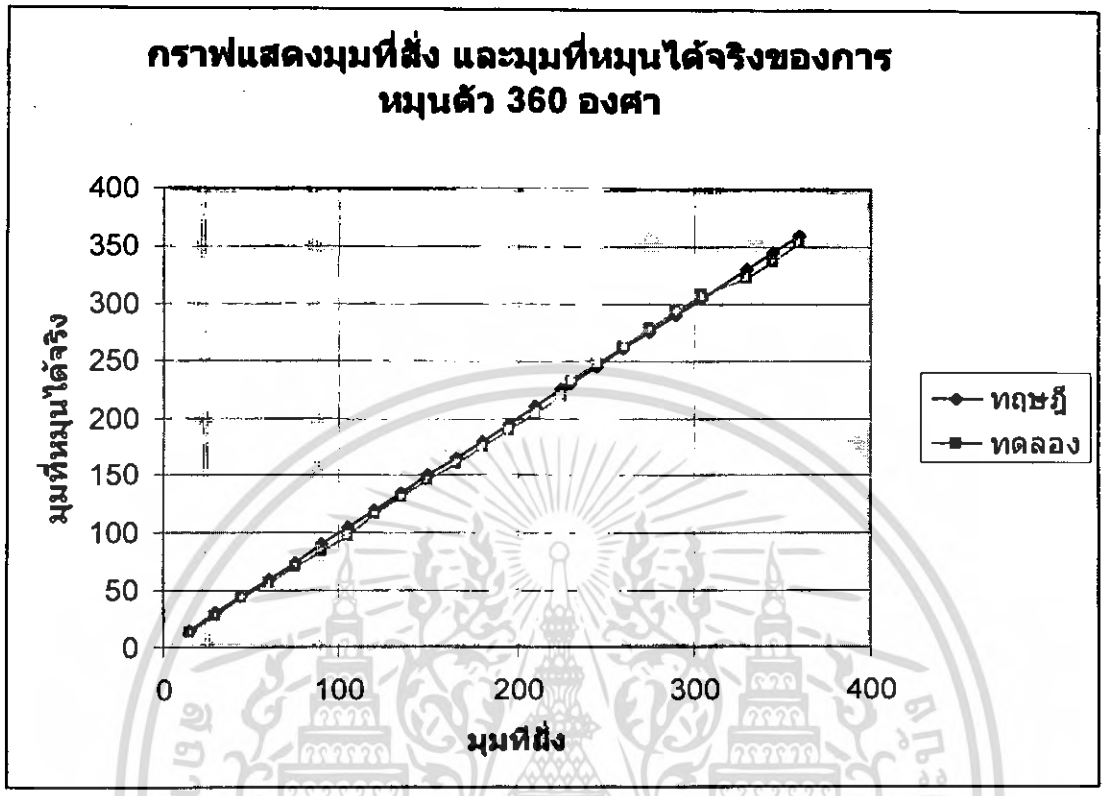
จากนั้นนำค่าค่าเฉลี่ยของ X-Y จากตารางที่ 6.2 มาสร้างกราฟเพื่อดูลักษณะการเปลี่ยนตำแหน่งของหุ่นยนต์ จะได้กราฟดังรูปที่ 6.12



รูปที่ 6.13 แสดงจุดศูนย์กลางของตัวหุ่นจากการหมุนรอบตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และนำค่าเฉลี่ยของมุมที่ได้จากตาราง 6.2 มาทำการวิเคราะห์ด้วยกราฟเพื่อเปรียบเทียบมุมที่หุ่นยนต์หมุนได้จริงกับมุมทางทฤษฎี ดังรูปที่ 6.14



รูปที่ 6.14 แสดงองศาที่หมุนได้เทียบกับค่าทางทฤษฎี

6.4.3 สรุปผลการทดลอง

การหมุนอยู่กับที่ตามเข็มนาฬิกา 360 องศา เราสามารถบังคับให้หุ่นหมุ่นตัวเองได้แต่เรายังไม่สามารถบังคับให้หุ่นหมุนอยู่กับที่ได้ หุ่นยังมีการเคลื่อนตัวออกจากจุดศูนย์กลางการหมุนที่ตั้งไว้ (0,0) ซึ่งมีความผิดพลาดสูงถึงเกือบ 6 เซ็นติเมตร แต่สำหรับมุมที่หุ่นสามารถหมุนได้จริงนั้นเมื่อเทียบกับองศาที่สั่งไปแล้วพบว่ามีความผิดพลาดน้อยมากดังแสดงในรูปที่ 6.14

6.5 การทดลองการเดินเป็นสี่เหลี่ยมจัตุรัส

6.5.1 วิธีทำการทดลอง

ในการทดลองนี้เราจะให้หุ่นยนต์เดินเป็นรูปสี่เหลี่ยมจัตุรัส

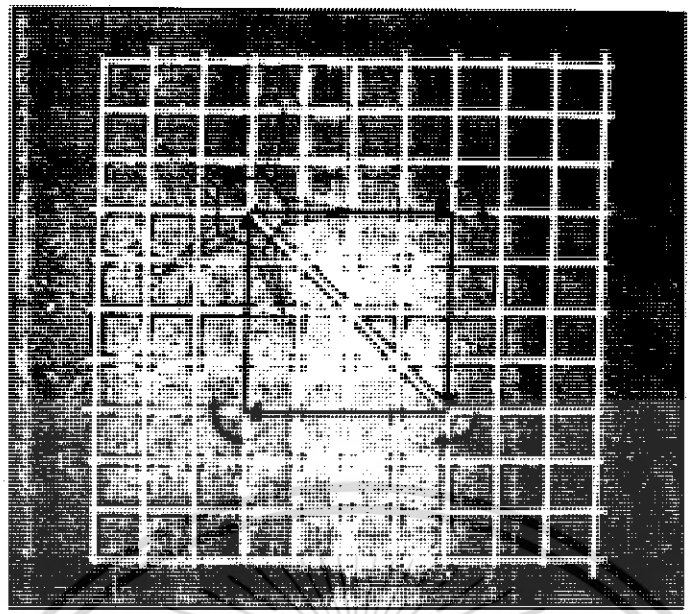
โดยเริ่มที่ (-200, 200) แล้วเดินไปยัง (200, 200) หมุนรอบตัวเองตามเข็มนาฬิกา 90 องศา

เดินไปยัง (200, -200) หมุนรอบตัวตัวเองตามเข็มนาฬิกา 90 องศา

เดินไปยัง (-200, -200) หมุนรอบตัวเองตัวตามเข็มนาฬิกา 90 องศา

เดินไปสิ้นสุดที่ (-200, 200) ดังรูปที่ 6.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.15 แสดงรูปแบบการเดินเป็น สี่เหลี่ยมของหุ่นยนต์

6.5.2 ผลการทดลองการเดินเป็นสี่เหลี่ยมจัตุรัส

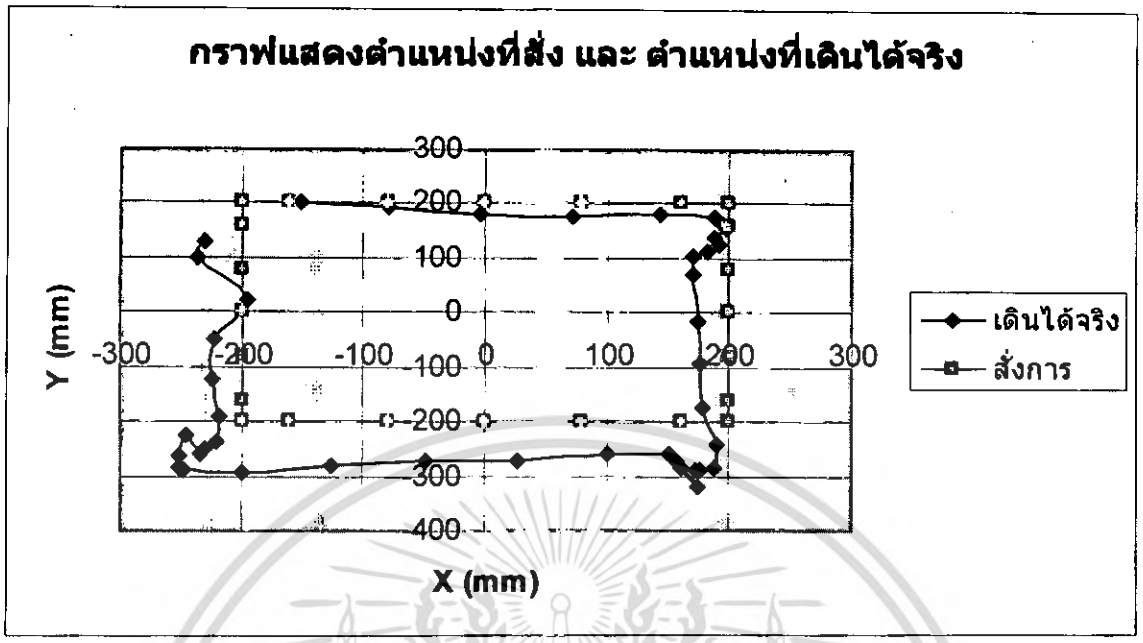
เมื่อเราหาค่า X-Y ที่ได้จากการทดลอง ดังตารางที่ 6.3

ภาพที่	X	Y	ภาพที่	X	Y
1	-151	202	22	174	-321
2	-79	194	23	156	-271
3	-4	181	24	150	-258
4	71	175	25	101	-260
5	143	180	26	27	-270
6	188	177	27	-48	-274
7	193	162	28	-126	-280
8	197	153	29	-199	-292
9	188	139	30	-248	-289
10	191	124	31	-251	-284
11	182	114	32	-252	-264
12	171	103	33	-246	-226
13	170	68	34	-235	-260
14	175	-16	35	-231	-251
15	177	-92	36	-220	-236
16	178	-172	37	-219	-192
17	190	-243	38	-225	-121
18	188	-285	39	-223	-48
19	176	-291	40	-195	21
20	172	-291	41	-236	100
21	160	-285	42	-230	130

ตารางที่ 6.3 แสดงค่าการทดลองการเดินเป็นสี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาเราก็จะทำการวิเคราะห์ด้วยกราฟ โดยการพลอตค่า X-Y ลงในกราฟดังรูปที่ 6.15



รูปที่ 6.16 แสดงการเปรียบเทียบตำแหน่งสั่งการ และ ตำแหน่งที่เดินได้จริง ของหุ่นยนต์

6.5.3 สรุปผลกรทดลองการเดินเป็นสี่เหลี่ยมจัตุรัส

หุ่นยนต์มีความยาวในการก้าวค่อนข้างคงที่จึงทำให้ขนาดของสี่เหลี่ยมที่เดินได้ใกล้เคียงกับขนาดของที่สั่งไป ความผิดพลาดส่วนใหญ่เกิดมาจากการเดินไม่ตรงของหุ่นยนต์ และ ความผิดพลาดจากการเลี้ยวที่ทำให้จุดศูนย์กลางของตัวหุ่นเปลี่ยนไป

6.6 วิจารณ์ผลการทดลอง

จากการทดลองเราพบว่าเราสามารถบังคับให้หุ่นยนต์สามารถเดินใกล้เคียงทิศทางที่เราต้องการได้ แต่การเดินยังคงมีข้อผิดพลาดอีกมาก ดังที่ได้แสดงไปแล้ว อันเนื่องมาจาก

1. การเดินมีการเลื่อนไถล เนื่องจากหุ่นเดินไปเหยียบเส้นเทปกาวบนสนาม
2. สนามที่ใช้ทดลองไม่เรียบเพียงพอทำให้ตำแหน่งการจรดปลายขาเปลี่ยนไป
3. ความผิดพลาดจากการประกอบหุ่นยนต์เนื่องจากการประกอบทั้งหมดทำด้วยมือทำให้มีข้อผิดพลาดหลายส่วนเป็นผลให้มิติของหุ่นยนต์ไม่เป็นไปตามที่คำนวนไว้
4. ลักษณะท่าทางการก้าวขาของหุ่นยนต์เองเนื่องจากหุ่นโยกตัวก่อนแล้วค่อยก้าวขา ในจังหวะแรกคือขา1 และ 4 จะก้าวก่อน จังหวะถัดมาคือ ขา 2 และ 5 เนื่องจากจังหวะขา 1 และ 4 ก้าวเสร็จแล้วนั้นจะทำให้ปลายขา 1 อยู่ห่างลำตัวหุ่นมากที่สุดเป็นผลให้เกิดความผิดพลาดจากการที่เซอร์โวมอเตอร์ไม่สามารถรักษาตำแหน่งตนเองไว้ได้ดีพอเนื่องจากต้องรับภาระมาก

การลดข้อผิดพลาดเราควรใช้การใช้สีดีเส้นแทนการใช้เทปกาวติดและควรตึงสนามให้ตึงเพื่อให้เรียบ ตรวจสอบและแก้ไขมิติของหุ่นยนต์ ให้ตรงตามแบบที่ได้ออกไว้ และ ควรเลี่ยนท่าทางการเดินของหุ่นยนต์เพื่อไม่ให้เซอร์โวมอเตรชอร์แต่ละตัวรับภาระหนักเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปและวิจารณ์

สรุปผลการดำเนินงานการพัฒนาและควบคุมต้นแบบหุ่นยนต์ 5 ขา ได้ศึกษาและสร้างหุ่นยนต์ห้าขาต้นแบบขึ้นมา ได้ทำการพัฒนารูปแบบการเดินจากรุ่นพี่ และเขียนโปรแกรมภาษาซีเพื่อควบคุมการเดินหุ่นยนต์ได้สำเร็จ โดยได้บรรลุวัตถุประสงค์หลักที่ตั้งไว้ทั้งหมดซึ่งได้แก่ศึกษาถึงหลักการทำงานเบื้องต้นของหุ่นยนต์ การทำงานของไมโครคอนโทรลเลอร์ การใช้เซอร์โวมอเตอร์เป็นต้นกำลัง การใช้ภาษาซีในการควบคุมหุ่นยนต์การประยุกต์ความรู้ด้านวิศวกรรมเครื่องกลมาใช้การสร้างต้นแบบของหุ่นยนต์ขึ้นมา พร้อมเขียนโปรแกรมควบคุม และยังได้ทำการทดลองเพิ่มเติมเพื่อวัดการเดินของหุ่นยนต์ที่เดินได้จริงด้วย

7.1 ปัญหาที่เกิดขึ้นและแนวทางแก้ไข

ปัญหาหลักของการทำงานก็คือ การใช้งานบอร์ด ZX-SERVO16 เนื่องจากเป็นบอร์ตที่ไม่เป็นที่นิยมแพร่หลายทำให้ต้องศึกษาทดลองเองแทบทั้งหมดอีกทั้งเป็นบอร์ดที่มีราคาถูกจึงไม่มีวงจรป้องกันการลัดวงจร เป็นผลให้บอร์ดเกิดความเสียหายขณะทดลองใช้ ต้องเสียเวลาและเงินเพิ่ม และยังมีปัญหาเรื่องของสายสัญญาณ ที่เล็กเกินไป จึงแก้ไขโดยการเพิ่มขนาดของสายสัญญาณ และเพิ่มวงจรเร็กกูเลเตอร์เข้าไปเพื่อขจัดปัญหาการสูญเสียพลังงานขณะส่งผ่านสายไฟ และปัญหาที่สำคัญที่สุดก็คือข้อจำกัดด้านงบประมาณ เพราะงบที่ได้น้อยทำให้เรามีเซอร์โวมอเตอร์ให้เลือกได้น้อย และ ได้เซอร์โวมอเดอร์ที่มีความแม่นยำต่ำส่งผลให้หุ่นยนต์ที่ได้มีการเดินที่ผิดพลาดสูง

7.2 แนวทางในการพัฒนาต่อ

เนื่องจากงบประมาณที่จำกัดทำให้สามารถสร้างหุ่นยนต์ห้าขาที่มีการทำงานแบบส่งค่าออกไปทางเดียว (Open loop control) และ เซอร์โวมอเตอร์ที่ใช้ก็มีคุณภาพต่ำทำให้ยากต่อการรักษาตำแหน่งที่ต้องการไว้ การพัฒนาขั้นต่อไปมีแนวทางดังนี้

1. มีอุปกรณ์ตรวจจับสิ่งกีดขวางที่ทำหน้าที่เปรียบเสมือนดวงตาของหุ่นยนต์ ตรวจจับและส่งค่ากลับมาเป็นระยะทางว่าตัวหุ่นอยู่ห่างจากสิ่งกีดขวางเป็นระยะทางเท่าใด หรือ สิ่งกีดขวางนั้นๆ มีความสูงเท่าใด เพื่อไปวิเคราะห์ว่าหุ่นยนต์จะสามารถข้ามสิ่งกีดขวางนั้นได้หรือไม่ หรือหากข้ามไม่ได้จะสามารถหลีกเลี่ยงไปด้านใดได้บ้าง

2. ควรใช้เซอร์โวมอเตอร์ที่มีสมรรถนะสูงกว่าเดิมทั้งแรงบิด และ ความละเอียดเมื่อทำงานร่วมกับอุปกรณ์ตรวจจับสิ่งกีดขวางจะสามารถรักษาระยะห่างของตัวหุ่นจากสิ่งกีดขวางหรือกำแพงได้อย่างแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. พัฒนาการติดต่อสื่อสารจากใช้สายเป็นแบบไร้สาย และติดตั้งแหล่งพลังงานภายในตัวหุ่นเพื่อนำไปใช้ปฏิบัติงานในภาคสนามได้จริงต่อไป

4. ควรมีอุปกรณ์รับรู้ความดันที่ปลายขาของหุ่นยนต์เพื่อปรับเปลี่ยนการกระจายน้ำหนักของหุ่นยนต์ เพื่อไม่ให้เซอร์โวมอเตอร์แต่ละตัวนั้นรับภาระมากเกินไป และยังเป็นการยืดอายุการใช้งานของเซอร์โวมอเตอร์อีกด้วย

5. สามารถพัฒนาเป็นหุ่นยนต์กู้ภัยโดยการติดกล้องไปที่ตัวหุ่นเพื่อการรับรู้ ซึ่งการที่หุ่นใช้ขาในการเดินจะทำให้ได้เปรียบในการเข้าไปในพื้นที่ ที่มีความขรุขระมากกว่าหุ่นยนต์ที่ใช้ล้อในการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] บัญชา โสภณ,ประชาธิบ ราศรีวิสุทธิ์,วิชญ์พันธ์ กันนิ่ม "การออกแบบและวิเคราะห์หุ่นยนต์ห้าขา", ภาควิชาวิศวกรรมเครื่องกล สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2548.

[2] วัชรินทร์ เคารพ "คู่มือการใช้งาน SERVO MOTER",บริษัท อีทีที จำกัด, 2546.

[3] "คู่มือการใช้งาน บอร์ดควบคุมเซอร์โวมอเตอร์ 16 ช่อง",บริษัท อินโนเวดีฟ เอ็กเพอริเมนต์ จำกัด.

[4] www.inex.co.th

[5] www.etteam.com

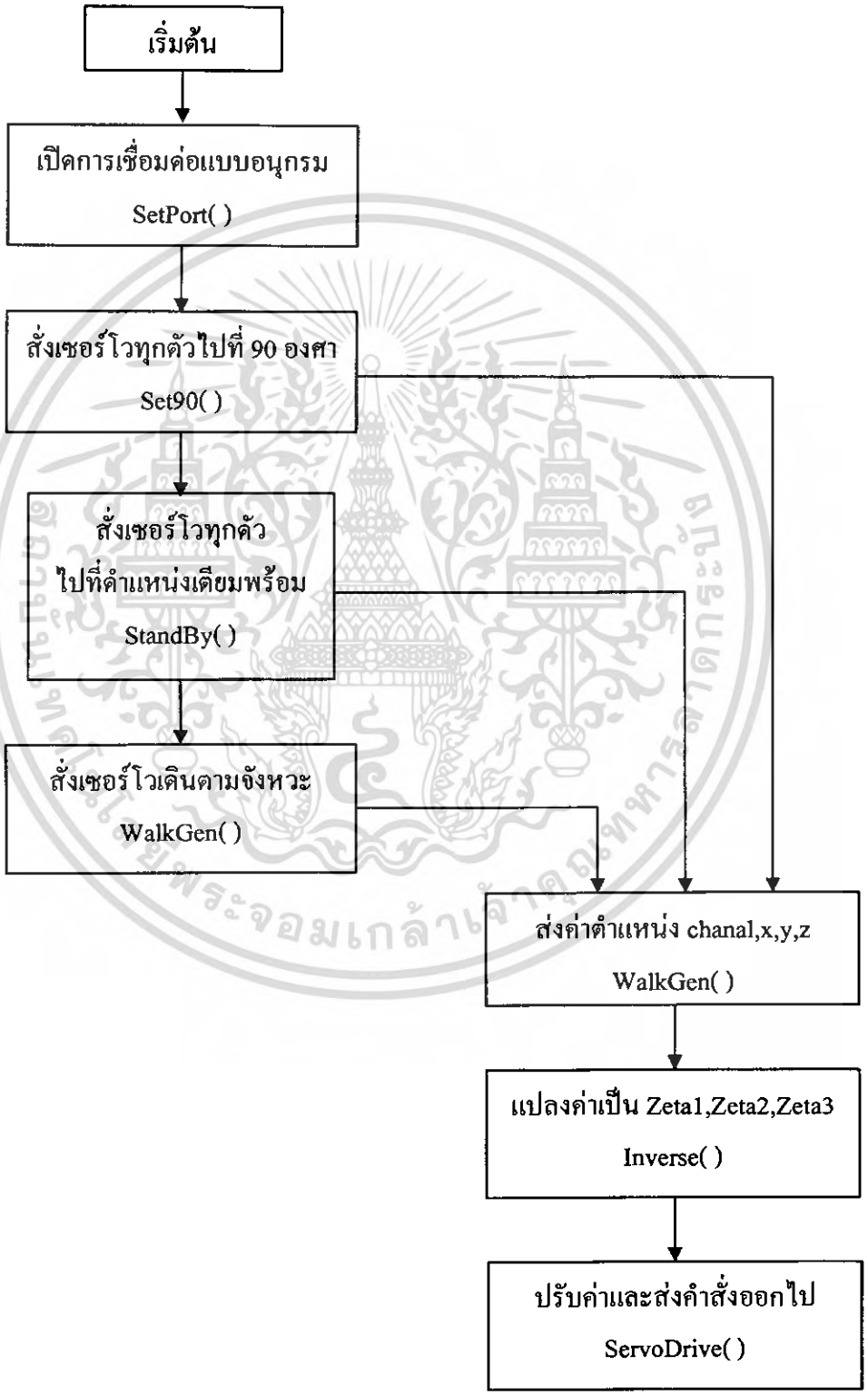
[6] สถาพร ลักษณะเจริญ "วิศวกรรมหุ่นยนต์",สสท.,2545.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

การทำงานของโปรแกรม

Flow Chart



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังก์ชันต่างๆ ในโปรแกรม

OpenComPort("COM1")

ในส่วนหัวของโปรแกรมนั้นเป็นรูปแบบสำเร็จ ของการเปิดพอร์ตอนุกรมซึ่งสามารถเรียกใช้ได้ทันทีด้วยการเรียกฟังก์ชัน SetPort() ซึ่งโปรแกรมจะจัดการเชื่อมต่อพอร์ตอนุกรมให้เองโดยอัตโนมัติ ในการติดต่อสื่อสารกับบอร์ด ZX-SERVO16 เราจะสามารถเลือกใช้ อัตราการส่งข้อมูลหรือ Baud rate ได้ 2 ระดับคือ ที่ 2400 บิทต่อวินาที กับที่ 38400 บิทต่อวินาที โดยสามารถเปลี่ยนค่าได้ที่ตัวแปร BaudRate หากต้องการใช้ระดับ 2400 ให้ใส่ค่า BaudRate=CBR_2400; หากต้องการใช้ระดับ 38400 ให้ใส่ค่า BaudRate=CBR_38400;

PutSerial(char dat)

เป็นฟังก์ชันส่งค่าออกพอร์ตซีเรียลโดยจะสามารถส่งได้ทีละตัวเท่านั้น

int GetSerial()

เป็นฟังก์ชันรับค่าจากพอร์ตซีเรียลโดยจะสามารถรับได้ทีละตัวเท่านั้น

SetPort()

เป็นชุดคำสั่งในการตั้งค่า BaudRate เพื่อให้สะดวกต่อการเลือกใช้

ServoDrive(int CH,float DG)

เป็นฟังก์ชันสำหรับรับการขับเซอร์โวมอเตอร์ไปยังตำแหน่งที่ต้องการ โดยการระบุหมายเลขช่องของเซอร์โว และ ตำแหน่งเป็นองศามาให้ ฟังก์ชันจะทำการคำนวณและปรับค่า (Offset) ตามที่ตั้งเอาไว้และจะส่งค่าที่ได้คำนวณแล้วออกไปยังบอร์ด ZX-SERVO16 โดยตรง

Inverse(int Leg,float X,float Y,float Z)

เป็นฟังก์ชันคำนวณหลักของโปรแกรมนี้ จะทำการรับค่า หมายเลขขา และตำแหน่ง X,Y,Z ที่ต้องการให้ปลายขาไปอยู่ซึ่งจะเอาค่า X,Y,Z ที่ได้ไปคำนวณจลนศาสตร์ผกผันแบบกลใกสองก้านโยง เพื่อให้ได้ค่าของมุมทั้ง 3 ที่เซอร์โวมอเตอร์ต้องทำ แล้วส่งค่ามุมทั้ง 3 ไปยังฟังก์ชัน ServoDrive(int CH,float DG) เพื่อส่งค่าออกไป

StandBy(void)

เป็นฟังก์ชันสำหรับสั่งให้หุ่นยนต์มาอยู่ในสถานะพร้อมทำงาน หรือ ยืนอยู่ปกตินั่นเอง

WalkGen(float Distance,int Lead)

เป็นฟังก์ชันสำหรับสั่งให้หุ่นเดินโดย จะทำการกำหนดจุดปลายของแต่ละขาในแต่ละจังหวะการเดินแล้วส่งค่านั้นๆ ในรูปแบบ X,Y,Z ไปให้กับฟังก์ชัน Inverse(int Leg,float X, float Y,float Z) เพื่อคำณวนเป็นมุมทั้ง 3 โดยฟังก์ชันนี้ต้องกำหนดหมายเลขขาที่จะใช้เป็นขานำ (Lead) เพื่อกำหนดทิศทางการเดินด้วย ซึ่งขาปกติที่ใช้นำคือขาหมายเลข 3

Rotate(float Alpha)

เป็นฟังก์ชันสำหรับหมุนตัวเพื่อเปลี่ยนทิศทางการเดินของหุ่นโดยจะต้องกำหนดองศาในการหมุนมา แล้วฟังก์ชันจะทำการคำนวณหาจำนวนครั้งที่ต้องหมุนโดยจะทำการหมุนทีละ 15 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CallRotate()

เป็นฟังก์ชันติดต่อสื่อสารกับผู้ใช้งานเพื่อระบุค่าองศาที่ต้องการให้หุ่นหมุน

SentXY(float X,float Y)

เป็นฟังก์ชันที่รับค่าพิกัด X,Y ที่ต้องการให้หุ่นเดินไป โดยโปรแกรมจะคำนวณหามุมที่ต้องหมุนตัวและหมายเลขขาที่ต้องนำในการเดิน จำนวนก้าวที่ต้องก้าว แล้วส่งค่าที่คำนวณได้ไปยังฟังก์ชัน WalkGen(float Distance,int Lead) และ ฟังก์ชัน Rotate(float Alpha)

void Set90()

เป็นฟังก์ชันสั่งให้เซอร์โวมอเตอร์ทุกตัวหมุนไปอยู่ในตำแหน่ง 90 องศา เพื่อให้ผู้ใช้งานตรวจสอบความถูกต้องของค่าที่ปรับ(Offset)

main(int argc, char *argv[])

เป็นฟังก์ชันหลักในการสั่งการทั้งหมด โดยสามารถเลือกชุดคำสั่งที่มีอยู่แล้ว หรือ สร้างชุดคำสั่งเองก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมที่ใช้ควบคุมหุ่น

```
//-------------------------------------------------------------------------
//Inverse1.1 by The_Miller Power by imem.
//"Five legs robot" Mechanical Engineering Department
//King's Mongkut Institute of Technology Ladkrabang.
//This program use for control "Five legs robot" by generate Zeta1,Zeta2,Zeta3
//from the end of leg position in X,Y,Z and sent it out reaal time to
//"ZX-Servo16" board on the body of robot to drive 15 servo of all robot legs.
//Email:pisarn@gmail.com
//23/02/07 23:42 :]

//-------------------------------------------------------------------------

//
#include <windows.h>
#include <stdio.h>

//-------------------------------------------------------------------------
//Define Perameter.
#define  PI   3.141592

//-------------------------------------------------------------------------
//Global Parameter Declearation.

   int BaudRate;//For comunicated with ZX-Servo16 >> 2400 or 38400 bps
   float Gnd = -(40.00+28.55);
//-------------------------------------------------------------------------
//"Open com port"

//This part use for initial parameter to open "com port" of
//computer for comunicated with "ZX-Servo16" by serial "RS232".
```

เอกสารนี้มีลิขสิทธิ์ ผู้ใดจะนำไปใช้ประโยชน์ด้านการค้า ต้องได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//You can change BaudRate of RS232 in line --> dcb.BaudRate = BaudRate;
//"Caution!" the ZX-Servo16 can use BaudRate "2400 bps" and "38400bps"
//In the start ZX-Servo16 "Automatic" set BaudRate == 2400 you can change
//it to 38400 by use comment in function "void SetPort()"to sent command
//'!' 'S' 'C' 'B' 'R' 1 13
//If you want to change it back to 2400 sent command
//'!' 'S' 'C' 'B' 'R' 1 13
//or press reset buttom in ZX-Servo16 board.

DCB dcb;
HANDLE hCom;
DWORD dwError;
BOOL fSuccess;
COMMTIMEOUTS CommTimeouts;

BOOL CloseComPort() {
        if (hCom)
                CloseHandle(hCom);
        return 0;
}

BOOL OpenComPort(LPSTR ComPort) {

        if (hCom)
                CloseHandle(hCom);

        hCom = CreateFile(ComPort,
                GENERIC_READ | GENERIC_WRITE,
                0,    /* comm devices must be opened w/exclusive-access */
                NULL, /* no security attrs */
                OPEN_EXISTING, /* comm devices must use OPEN_EXISTING */
                0,    /* not overlapped I/O */
                NULL  /* hTemplate must be NULL for comm devices */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    );

        if (hCom == INVALID_HANDLE_VALUE) {
                dwError = GetLastError();
                MessageBox(GetActiveWindow(), "Cannot Open Selected I/O Port", "Error",
MB_ICONERROR | MB_OK);
                return 0;
        }

        fSuccess = GetCommState(hCom, &dcb);

        if (!fSuccess) {
                MessageBox(GetActiveWindow(), "Cannot Open Selected I/O Port", "Error",
MB_ICONERROR | MB_OK);
                return 0;
        }

        dcb.BaudRate = BaudRate;//BaudRate Parameter;
        dcb.ByteSize = 8;
        dcb.Parity = NOPARITY;
        dcb.StopBits = ONESTOPBIT;
        dcb.fRtsControl = RTS_CONTROL_DISABLE;
        dcb.fDtrControl = DTR_CONTROL_DISABLE;

        fSuccess = SetCommState(hCom, &dcb);
        if (!fSuccess) {
                MessageBox(GetActiveWindow(), "Cannot Open Selected I/O Port", "Error",
MB_ICONERROR | MB_OK);
                /* Handle the error. */
                return 0;
        }

        CommTimeouts.ReadIntervalTimeout = MAXDWORD;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CommTimeouts.ReadTotalTimeoutConstant = 0;
        CommTimeouts.ReadTotalTimeoutMultiplier = 0;
        CommTimeouts.WriteTotalTimeoutConstant = 1000;
        CommTimeouts.WriteTotalTimeoutMultiplier = 1000;

        fSuccess = SetCommTimeouts(hCom, &CommTimeouts);
        if (!fSuccess) {
                MessageBox(GetActiveWindow(), "Cannot Open Selected I/O Port", "Error",
MB_ICONERROR | MB_OK);
                /* Handle the error. */
                return 0;
        }

        return 1;

}
//---------------------------------------------------------------------------
//sent out serial

//This function use for sent out command in serial "RS232" past com port
//of computer "It can sent only One by One".

int PutSerial(char dat) {//PutSerial(Character or number to sent out);
   DWORD cc;
   char sdat[1];
   sdat[0]=dat;
   WriteFile(hCom, sdat, 1, &cc, NULL);
//   Sleep(1);
        return 0;
}

//---------------------------------------------------------------------------
//Recieve serial
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//This function use for recieve command to computer in serial "RS232"
//past comport "It can recieve only One by One".

int GetSerial() {
   DWORD cc;
   char sdat[1];
   ReadFile(hCom, sdat, 1, &cc, NULL);
      if (cc>0)
      return sdat[0];
            else
      return -1;
}

//---------------------------------------------------------------------
//Select BaudRate 2400 or 38400.

//This function use for initial BaudRate of comunication.
//ZX-Servo16 board use RaudRate=2400 in the start or after press "Reset"buttom.
//Change BaudRate by use command in the comment it will sent
//'!' 'S' 'C' 'B' 'R' 1 13 to change BaudRate to 38400 and chance parameter
//BaudRate=CBR_38400; and call function "OpenComPort("COM1")" to open
//com port which use BaudRate = 38400

//"Warning">> Use BaudRate = 38400 will make comunication faster but will
//make program "Unstable".

void SetPort(){
//BaudRate = 2400 use this.
//      BaudRate=CBR_2400;//BaudRate=2400.
//      OpenComPort("COM1");
//Use this instead to change BaudRate from 2400 ot 38400
//BaudRate = 38400 use this.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BaudRate=CBR_2400;//BaudRate=2400.
    OpenComPort("COM1");
    PutSerial('!');
    PutSerial('S');
    PutSerial('C');
    PutSerial('S');
    PutSerial('B');
    PutSerial('R');
    PutSerial(1);
    PutSerial(13);
    BaudRate=CBR_38400;//BaudRate=38400.
    OpenComPort("COM1");

}

//---------------------------------------------------------------------
//Change Degree to High & Low byte and sent out to Servo chanal.

//This function recieve "chanal" and "Degree"  that you want then offet it
//from error of hardware assembly after that invert degree of chanal 1,4,7,10,13
//to inverse rotating direction of servo for compartable with mathematic model
//then change "Degree" from degree to Hight byte & Low byte of Futaba s3003
//and S03T/STD/JR  and finaly sent it out by function PutSerial()

float ServoDrive(int CH,float DG){//ServoDrive(Chanal want to sent,Degree of servo)
  float Block;//Futaba >> 1 Degree=4.71 Block   S03T >> 1 Degree= 4.51 Block.
  int hb,lb,SP=0;//High Byte,Low Byte

//  printf("\nDegree = %d-%.2f",CH,DG);//Input check

//Offet servo degree --> Use for offet error from hardware assembly.
//                If you change hardware you have to calibrate it first.
  switch(CH){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 0:
   DG+=0;//Confirm
   break;
case 1:
   DG+=7;//Confirm
   break;
case 2:
   DG-=3;//Confirm
   break;
case 3:
   DG+=2;//Confirm
   break;
case 4:
   DG+=18;//Confirm
   break;
case 5:
   DG-=10;//Confirm
   break;
case 6:
   DG-=3;//Confirm
   break;
case 7:
   DG+=10;//Confirm
   break;
case 8:
   DG+=-5;//Confirm
   break;
case 9:
   DG+=-10;//Confirm
   break;
case 10:
   DG+=20;//Confirm
   break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 11:
      DG+=-15;//Confirm
      break;
    case 12:
      DG+=-3;//Confirm
      break;
    case 13:
      DG+=+12;//Confirm
      break;
    case 14:
      DG+=-18;//Confirm
      break;
    case 15:
      DG+=0;//Confirm
      break;
    }
//Invert degree of Joint 2 & 3 --> Use for change rotating direction of servo
//                    moter because of hardware assembly direction.
//                    This function "Do not" change first joint
//                    of each legs(it mean ch1,ch4,ch7,ch10,ch13)

   if(!((CH==1)||(CH==4)||(CH==7)||(CH==10)||(CH==13))){
      DG=180-DG;

//   printf("Inverse %.2f",DG    );
   }



//   printf("\nch=%d,DG=%.2f",CH,DG);//Check input after offet & Invert
//   printf("\nIn ServoDrive");//Show status of program

//Change Degree to High byte & Low byte of Futaba S3003
//Block(Futaba S3003) = Degree x 4.9333
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//If 288<Block        High byte = 1, Low byte = Block+28
//If 288<=Block>484   High byte = 2, Low byte = Block-288
//If 484<=Block>740   High byte = 3, Low byte = Block-484
//If Block>740        High byte = 4, Low byte = Block-740

//"Caution!">> In range of Block>740 the servo moter has it limit to permit
//          be sure that your input Degree don't generate exeed value
//          (Each servo type has different limit)

  if((CH==3)||(CH==6)||(CH==9)||(CH==12)||(CH==15)){
    Block=DG*4.9333;//Servo Futaba S3003
    if(Block<228){
      hb=1;
      lb=(int)Block+28;
    }
    else if((Block>=228)&&(Block<484)){
      hb=2;
      lb=(int)Block-228;
    }
    else if((Block>=484)&&(Block<740)){
      hb=3;
      lb=(int)Block-484;
    }
    else{
      hb=4;
      lb=(int)Block-740;
    }
  }

//Change Degree to Hiht byte & Low byte of S03T/STD/JR
//Block(S03T/STD/JR) = Degree x 4.5444
//If 226<Block        High byte = 1, Low byte = Block+30
//If 226<=Block>482   High byte = 2, Low byte = Block-226
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//If 482<=Block>738     High byte = 3, Low byte = Block-482
//If Block>738          High byte = 4, Low byte = Block-738

//"Caution!">> In range of Block>740 the servo moter has it limit to permit
//          be sure that your input Degree don't generate exeed value
//          (Each servo type has different limit)

  else{
    Block=DG*4.5444;//Servo S03T/STD/JR
    if(Block<226){
      hb=1;
      lb=(int)Block+30;
    }
    else if((Block>=226)&&(Block<482)){
      hb=2;
      lb=(int)Block-226;
    }
    else if((Block>=482)&&(Block<738)){
      hb=3;
      lb=(int)Block-738;
    }
    else{
      hb=4;
      lb=(int)Block-738;
    }
  }

//sent DATA to ZX-Servo16 Board by function PutSerial() one by one
    PutSerial('!');
    PutSerial('S');
    PutSerial('C');
    PutSerial(CH);
    PutSerial(SP);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    PutSerial(lb);
    PutSerial(hb);
    PutSerial(13);
    Sleep(1);//Delay for Lad Data out << "Very important parameter!"
}

//---------------------------------------------------------------------------
//Change position X,Y,Z to Zeta1,Zeta2,Zeta3

//This function gemerate Zeta1,Zeta2,Zeta3 from X,Y,Z position.

void Inverse(int Leg,float X,float Y,float Z){
   float L1=68.9,L2=68.9,L3=80;
   float R1,R2,R3,Zeta1,Zeta2,Zeta3;
   float ZetaM,ZetaN,ZetaP,ZetaMP;
   float P,A,B,C,D,L3x,L3y;

//L1 = Length from Joint 1 to Joint 2
//L2 = Length from Joint 2 to Joint 3
//L3 = Length from Joint 3 to End

//R1 = Length from Joint 1 to Joint 2 in X axis
//R2 = Length from Joint 2 to Joint 3 in X axis
//R3 = Length from Joint 3 to End in X axis

//Zeta1 = Degree of Joint 1
//Zeta2 = Degree of Joint 2
//Zeta3 = Degree of Joint 3

//P = Vector from Joint 2 to End
//A = Buffer1
//B = Buffer2
//C = Buffer3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//ZetaN = Degree of L3 to X axis
//ZetaP = Degree of Vector P to X axis
//ZetaMP = ZetaP+Zeta2


   R1=L1;
   R3=sqrt(X*X+Y*Y);
//   printf("\nR3=%.2f",R3);          //Check R3
   Zeta1 = atan(Y/X);
//   printf("\nZeta1=%.2f",Zeta1);    //Check Zeta1
   ZetaP = atan(Z/(R3-R1));
//   printf("\nZetaP=%.2f",ZetaP);    //Check ZetaP
   P=(R3-R1)/cos(ZetaP);
//   printf("\nP=%.2f",P);            //Check P
   A=(-(L3*L3)+(P*P)+(L2*L2));
//   printf("\nA=%.2f",A);            //Check A
   B=(2*P*L2);
//   printf("\nB=%.2f",B);            //Check B
   C=A/B;
//   printf("\nC=%.2f",C);            //Check C
   ZetaMP=acos(C);
//ZetaMP=acos((-(L3*L3)+(P*P)+(L2*L2))/(2*P*L2));
//   printf("\nZetaMP=%.2f",ZetaMP);      //Check ZetaMP


   ZetaM=ZetaMP+ZetaP;
   R2=L1+L2*cos(ZetaM);
//   printf("\nR2 = %.2f",R2);
//   printf("\nsin ZetaP = %.2f",sin(ZetaP));
//   printf("\nsin ZetaM = %.2f",sin(ZetaM));
//   printf("\nL3 = %.2f",L3);
   L3x=R3-R2;
   L3y=P*sin(ZetaP)-L2*sin(ZetaM);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//   printf("\nL3x = %.2f \nL3y = %.2f",L3x,L3y);
//   printf("L3x/L3y = %.2f",L3x/L3y);

   if(L3x>0)ZetaN=atan(L3y/L3x);
   else ZetaN=atan(L3y/L3x)-3.14;

//   ZetaN=asin((P*sin(ZetaP)-(L2*sin(Zeta2)))/L3);
//ZetaN=(((Zeta3)*180/PI)+180-Zeta2);
//printf("ZetaN=%.2f \n",ZetaN);       //Check ZetaN

//Offet for Zero
//printf("\nZeta1 in Radian = %.2f",Zeta1);
//printf("\nZeta2 in Radian = %.2f",Zeta2);
//printf("\nZetaN in Radian = %.2f",ZetaN);

   Zeta1 = ((Zeta1)*180/PI)+90;
   Zeta2 = ((ZetaM)*180/PI)+90;
   Zeta3 = 270+(ZetaN*180/PI)-Zeta2;
//Zeta3 = (((Zeta3)*180/PI)+180-Zeta2);

//printf("\nZ1=%.2f",Zeta1);          //Check output Zeta1
   ServoDrive(3*Leg-2,Zeta1);
//printf("\nZ2=%.2f",Zeta2);          //Check output Zeta2
   ServoDrive(3*Leg-1,Zeta2);
//printf("\nZ3=%.2f",Zeta3);          //Check output Zeta3
   ServoDrive(3*Leg,Zeta3);
}

//-------------------------------------------------------------------------
//StandBy

//This function use for prepare legs' position for walking step.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void StandBy(void){
//Inverse(Leg,X,Y,Z);
      Inverse(1,150.46,0,Gnd);
      Inverse(2,150.46,0,Gnd);
      Inverse(3,150.46,0,Gnd);
      Inverse(4,150.46,0,Gnd);
      Inverse(5,150.46,0,Gnd);
}

//---------------------------------------------------------------------------
//WalkGen >> Walking Generate Function

//This function use for sent out "End of legs position" of each legs.
//It define the position of each legs in term of X,Y,Z then generate
//Zeta1,Zeta2,Zeta3 and sent out to ZX-Sevo16 by Inverse function.

void WalkGen(float Distance,int Lead){

  int A,B,C,D,E;
  int WalkStep=1,Round=1,j,time=200;
  float Buffer=Distance,Range=80,HRange=0.5*Range,StepHigh=100;
  float Sx,Sy,Delta1x,Delta1y,Delta2x,Delta2y;
  float A1x,A1y,A1z,B1x,B1y,B1z,C1x,C1y,C1z,D1x,D1y,D1z;
  float A2x,A2y,A2z,B2x,B2y,B2z,C2x,C2y,C2z,D2x,D2y,D2z;
  float A3x,A3y,A3z,B3x,B3y,B3z,C3x,C3y,C3z,D3x,D3y,D3z;
  float A4x,A4y,A4z,B4x,B4y,B4z,C4x,C4y,C4z,D4x,D4y,D4z;
  float A5x,A5y,A5z,B5x,B5y,B5z,C5x,C5y,C5z,D5x,D5y,D5z;

  switch(Lead){
    case 1:
      A=4;B=5;C=1;D=2;E=3;
      brcak;
    case 2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        A=5;B=1;C=2;D=3;E=4;
        break;
      case 3:
        A=1;B=2;C=3;D=4;E=5;
        break;
      case 4:
        A=2;B=3;C=4;D=5;E=1;
        break;
      case 5:
        A=3;B=4;C=5;D=1;E=2;
        break;
      default:
        A=1;B=2;C=3;D=4;E=5;
        printf("\nError wrong Lead leg it must 1,2,3,4 or 5.\nSet Lead=5");
    }



//   printf("\nln WalkGen");                    //Report status
//1Round = 11Step

    j=Distance/Range;
    if((Buffer-j*Range)!=0)j++;

    Sx=150.46;//StandBy in X axis
    Delta1x=HRange*cos(0.62);
    Delta1y=HRange*sin(0.62);
    Delta2x=HRange*cos(1.27);
    Delta2y=HRange*sin(1.27);

//  S1x=150.46;   1Sy=0;        1Sz=Gnd;
    A1x=Sx+Delta1x;   A1y=Delta1y;   A1z=Gnd;
    B1x=Sx+Delta1x;   B1y=Delta1y;   B1z=Gnd+StepHigh;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    C1x=Sx-Delta1x;   C1y=-Delta1y;   C1z=Gnd+StepHigh;
    D1x=Sx-Delta1x;   D1y=-Delta1y;   D1z=Gnd;

// printf("\n\nA1x= %.2f A1y = %.2f",A1x,A1y);
// printf("\nB1x= %.2f B1y = %.2f",B1x,B1y);
// printf("\nC1x= %.2f C1y = %.2f",C1x,C1y);

// 2Sx=150.46;   2Sy=0;       2Sz=Gnd;
    A2x=Sx-Delta2x;   A2y=Delta2y;    A2z=Gnd;
    B2x=Sx-Delta2x;   B2y=Delta2y;    B2z=Gnd+StepHigh;
    C2x=Sx+Delta2x;   C2y=-Delta2y;   C2z=Gnd+StepHigh;
    D2x=Sx+Delta2x;   D2y=-Delta2y;   D2z=Gnd;
// printf("\n\nA2x= %.2f A2y = %.2f",A2x,A2y);
// printf("\nB2x= %.2f B2y = %.2f",B2x,B2y);
// printf("\nC2x= %.2f C2y = %.2f",C2x,C2y);

// 3Sx=150.46;   3Sy=0;       3Sz=Gnd;
    A3x=Sx-HRange;    A3y=0;        A3z=Gnd;
    B3x=Sx-HRange;    B3y=0;        B3z=Gnd+StepHigh;
    C3x=Sx+HRange;    C3y=0;        C3z=Gnd+StepHigh;
    D3x=Sx+HRange;    D3y=0;        D3z=Gnd;

// printf("\n\nA3x= %.2f A3y = %.2f",A3x,A3y);
// printf("\nB3x= %.2f B3y = %.2f",B3x,B3y);
// printf("\nC3x= %.2f C3y = %.2f",C3x,C3y);

// 4Sx=150.46;   4Sy=0;       4Sz=Gnd;
    A4x=Sx-Delta2x;   A4y=-Delta2y;   A4z=Gnd;
    B4x=Sx-Delta2x;   B4y=-Delta2y;   B4z=Gnd+StepHigh;
    C4x=Sx+Delta2x;   C4y=Delta2y;    C4z=Gnd+StepHigh;
    D4x=Sx+Delta2x;   D4y=Delta2y;    D4z=Gnd;
// printf("\n\nA4x= %.2f A4y = %.2f",A4x,A4y);
// printf("\nB4x= %.2f B4y = %.2f",B4x,B4y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// printf("\nC4x= %.2f C4y = %.2f",C4x,C4y);

// 5Sx=150.46;   5Sy=0;       5Sz=Gnd;
   A5x=Sx+Delta1x;   A5y=-Delta1y;   A5z=Gnd;
   B5x=Sx+Delta1x;   B5y=-Delta1y;   B5z=Gnd+StepHigh;
   C5x=Sx-Delta1x;   C5y=Delta1y;    C5z=Gnd+StepHigh;
   D5x=Sx-Delta1x;   D5y=Delta1y;    D5z=Gnd;

// printf("\n\nA5x= %.2f A5y = %.2f",A5x,A5y);
// printf("\nB5x= %.2f B5y = %.2f",B5x,B5y);
// printf("\nC5x= %.2f C5y = %.2f",C5x,C5y);



      while(Round<=j){

//    getch();
//      printf("%d",WalkStep);          //Check WalkStep


      if((Round==j)&&(WalkStep==1)){
      Range=(Buffer-(Round-1)*Range);
//      printf("\nRange = %.2f",Range);

      HRange=0.5*Range;

   Delta1x=HRange*cos(0.62);
   Delta1y=HRange*sin(0.62);
   Delta2x=HRange*cos(1.27);
   Delta2y=HRange*sin(1.27);

// S1x=150.46;   1Sy=0;       1Sz=Gnd;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   A1x=Sx+Delta1x;   A1y=Delta1y;    A1z=Gnd;
   B1x=Sx+Delta1x;   B1y=Delta1y;    B1z=Gnd+StepHigh;
   C1x=Sx-Delta1x;   C1y=-Delta1y;   C1z=Gnd+StepHigh;
   D1x=Sx-Delta1x;   D1y=-Delta1y;   D1z=Gnd;

// printf("\n\nA1x= %.2f A1y = %.2f",A1x,A1y);
// printf("\nB1x= %.2f B1y = %.2f",B1x,B1y);
// printf("\nC1x= %.2f C1y = %.2f",C1x,C1y);

// 2Sx=150.46;   2Sy=0;      2Sz=Gnd;
   A2x=Sx-Delta2x;   A2y=Delta2y;    A2z=Gnd;
   B2x=Sx-Delta2x;   B2y=Delta2y;    B2z=Gnd+StepHigh;
   C2x=Sx+Delta2x;   C2y=-Delta2y;   C2z=Gnd+StepHigh;
   D2x=Sx+Delta2x;   D2y=-Delta2y;   D2z=Gnd;
// printf("\n\nA2x= %.2f A2y = %.2f",A2x,A2y);
// printf("\nB2x= %.2f B2y = %.2f",B2x,B2y);
// printf("\nC2x= %.2f C2y = %.2f",C2x,C2y);

// 3Sx=150.46;   3Sy=0;      3Sz=Gnd;
   A3x=Sx-HRange;    A3y=0;        A3z=Gnd;
   B3x=Sx-HRange;    B3y=0;        B3z=Gnd+StepHigh;
   C3x=Sx+HRange;    C3y=0;        C3z=Gnd+StepHigh;
   D3x=Sx+HRange;    D3y=0;        D3z=Gnd;

// printf("\n\nA3x= %.2f A3y = %.2f",A3x,A3y);
// printf("\nB3x= %.2f B3y = %.2f",B3x,B3y);
// printf("\nC3x= %.2f C3y = %.2f",C3x,C3y);

// 4Sx=150.46;   4Sy=0;      4Sz=Gnd;
   A4x=Sx-Delta2x;   A4y=-Delta2y;   A4z=Gnd;
   B4x=Sx-Delta2x;   B4y=-Delta2y;   B4z=Gnd+StepHigh;
   C4x=Sx+Delta2x;   C4y=Delta2y;    C4z=Gnd+StepHigh;
   D4x=Sx+Delta2x;   D4y=Delta2y;    D4z=Gnd;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// printf("\n\nA4x= %.2f A4y = %.2f",A4x,A4y);
// printf("\nB4x= %.2f B4y = %.2f",B4x,B4y);
// printf("\nC4x= %.2f C4y = %.2f",C4x,C4y);

// 5Sx=150.46;   5Sy=0;       5Sz=Gnd;
   A5x=Sx+Delta1x;   A5y=-Delta1y;   A5z=Gnd;
   B5x=Sx+Delta1x;   B5y=-Delta1y;   B5z=Gnd+StepHigh;
   C5x=Sx-Delta1x;   C5y=Delta1y;    C5z=Gnd+StepHigh;
   D5x=Sx-Delta1x;   D5y=Delta1y;    D5z=Gnd;

// printf("\n\nA5x= %.2f A5y = %.2f",A5x,A5y);
// printf("\nB5x= %.2f B5y = %.2f",B5x,B5y);
// printf("\nC5x= %.2f C5y = %.2f",C5x,C5y);

    }


        //Check Round
      switch(WalkStep){

      case 1:                          //Step1
        printf("\nRound %d >> ",Round);
        Inverse(A,A1x,A1y,A1z);
        Inverse(B,A2x,A2y,A2z);
        Inverse(C,A3x,A3y,A3z);
        Inverse(D,A4x,A4y,A4z);
        Inverse(E,A5x,A5y,A5z);
        //printf("Case1");
        sleep(time);
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 2:
  Inverse(A,B1x,B1y,B1z);          //Step2
  Inverse(B,A2x,A2y,A2z);
  Inverse(C,A3x,A3y,A3z);
  Inverse(D,B4x,B4y,B4z);
  Inverse(E,A5x,A5y,A5z);
  //printf("Case2");
  sleep(time);
  break;

case 3:
  Inverse(A,C1x,C1y,C1z);          //Step3
  Inverse(B,A2x,A2y,A2z);
  Inverse(C,A3x,A3y,A3z);
  Inverse(D,C4x,C4y,C4z);
  Inverse(E,A5x,A5y,A5z);
  //printf("Case3");
  sleep(time);
  break;

case 4:
  Inverse(A,D1x,D1y,D1z);          //Step4
  Inverse(B,A2x,A2y,A2z);
  Inverse(C,A3x,A3y,A3z);
  Inverse(D,D4x,D4y,D4z);
  Inverse(E,A5x,A5y,A5z);
  //printf("Case3");
  sleep(time);
  break;

case 5:
  Inverse(A,D1x,D1y,D1z);          //Step5
  Inverse(B,B2x,B2y,B2z);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Inverse(C,A3x,A3y,A3z);
  Inverse(D,D4x,D4y,D4z);
  Inverse(E,B5x,B5y,B5z);
  //printf("Case3");
  sleep(time);
  break;

case 6:
  Inverse(A,D1x,D1y,D1z);          //Step6
  Inverse(B,C2x,C2y,C2z);
  Inverse(C,A3x,A3y,A3z);
  Inverse(D,D4x,D4y,D4z);
  Inverse(E,C5x,C5y,C5z);
  //printf("Case3");
  sleep(time);
  break;

case 7:
  Inverse(A,D1x,D1y,D1z);          //Step7
  Inverse(B,D2x,D2y,D2z);
  Inverse(C,A3x,A3y,A3z);
  Inverse(D,D4x,D4y,D4z);
  Inverse(E,D5x,D5y,D5z);
  //printf("Case3");
  sleep(time);
  break;

case 8:
  Inverse(A,D1x,D1y,D1z);          //Step8
  Inverse(B,D2x,D2y,D2z);
  Inverse(C,B3x,B3y,B3z);
  Inverse(D,D4x,D4y,D4z);
  Inverse(E,D5x,D5y,D5z);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    //printf("Case3");
    sleep(time);
    break;

  case 9:
    Inverse(A,D1x,D1y,D1z);          //Step9
    Inverse(B,D2x,D2y,D2z);
    Inverse(C,C3x,C3y,C3z);
    Inverse(D,D4x,D4y,D4z);
    Inverse(E,D5x,D5y,D5z);
    //printf("Case3");
    sleep(time);
    break;

  case 10:
    Inverse(A,D1x,D1y,D1z);          //Step10
    Inverse(B,D2x,D2y,D2z);
    Inverse(C,D3x,D3y,D3z);
    Inverse(D,D4x,D4y,D4z);
    Inverse(E,D5x,D5y,D5z);
    //printf("Case3");
    sleep(time);
    break;

  }//End switch
  Sleep(1);//Delay for servo move
  //printf("\n\n");
printf(" w%d",WalkStep);
WalkStep++;//WalkStep = WalkStep+1;
  if(WalkStep==11){//If WalkStep end Round = Round+1

    Round++;
    WalkStep=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }//End if

    }//End while
  StandBy();
}

void Rotate(float Alpha){
  int time=100,j,k=1,X,Y;
  float i=1,Buffer=Alpha,Range=15;
  float R,r3in,r3out,Gamma;
  float Beta;

  j=Alpha/Range;
  j=abs(j);
  printf("\nRotate = %.2f degree",Alpha);
  printf("\nRotate = %d time >>",j);
  if((abs(Buffer)-Range*j)!=0)j++;

  for(;i<=j;i++){
        printf(" r%d",k);
        k++;
    //Change from Degree to Radian
    if(Alpha>0)Alpha=Range;
    if(Alpha<0)Alpha=-Range;
    if((i==j)&&(Buffer>0))Alpha=Buffer-Range*(j-1);
    if((i==j)&&(Buffer<0))Alpha=Buffer+Range*(j-1);
//    printf("\nBuffer = %.2f",Buffer);          //Chaeck buffer
//    printf("\nAlpha = %.2f",Alpha);            //Check Alpha
    Alpha=(1.38*Alpha*PI)/180;                  //Offet Error
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
R=78.79;
r3in=150.46;
Beta=atan((R*sin(Alpha))/((R*(1-cos(Alpha)))+r3in));
//   Gamma=asin((2*(R+r3in)*sin(Alpha/2)*sin(0.5*PI-0.5*Alpha))/r3in);
Gamma=Alpha+Beta;
r3out=(r3in*sin(0.5*PI-0.5*Alpha))/(sin(0.5*PI-0.5*Alpha)-Beta);

X=r3out*cos(Gamma);
Y=r3out*sin(Gamma);

Inverse(1,150.46,0,Gnd+40);                    //Up
Inverse(4,150.46,0,Gnd+40);
sleep(time);

Inverse(1,X,r3out*sin(Gamma),Gnd+40);    //Move
Inverse(4,X,Y,Gnd+40);
sleep(time);

Inverse(1,X,Y,Gnd);      //Down
Inverse(4,X,Y,Gnd);
sleep(time);

Inverse(2,150.46,0,Gnd+40);                    //Up
Inverse(5,150.46,0,Gnd+40);
sleep(time);

Inverse(2,X,Y,Gnd+40);    //Move
Inverse(5,X,Y,Gnd+40);
sleep(time);

Inverse(2,X,Y,Gnd);      //Down
Inverse(5,X,Y,Gnd);
sleep(time);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Inverse(3,150.46,0,Gnd+40);                    //Up
    sleep(time);

    Inverse(3,X,Y,Gnd+40);    //Move
    sleep(time);

    Inverse(3,X,Y,Gnd);       //Down
    sleep(time);

    StandBy();
  }
}

void CallRotate(){
  float Alpha;
  printf("\nPlease key Rotate angle in Degree (float) : ");
  scanf("%.2f",&Alpha);
  Rotate(Alpha);
}

//---------------------------------------------------------------------
//Set90 >> Every servo go to 90 Degree

//This function take all servo go to 90 degree to approximate check the abnormal
//of each servo.
void SentXY(float X,float Y){

  int Lead=3;
  float Distance,Alpha;

  Distance=sqrt(X*X+Y*Y);
  printf("\nDistance = %.2f mm",Distance);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//When Devide by Zero
   if((X==0)&&(Y>0)){
      Alpha=90;
   }
   if((X==0)&&(Y<0)){
      Alpha=-90;
   }
   if(X!=0){
      Alpha=(atan(fabs(Y)/fabs(X)))*180/PI;
   }
   printf("Here Alpha = %.2f",Alpha);
//Define Degree
   if((X<=0)&&(Y>0)){
      Alpha=180-Alpha;
   }
   if((X<0)&&(Y<=0)){
      Alpha=-180+Alpha;
   }
   if((X>=0)&&(Y<0)){
      Alpha=Alpha;
   }
   if((X==0)&&(Y==0)){
      Alpha=0;
      Distance=0;
   }
   printf("\nDegree =  %.2f degree",Alpha);
//Define Leg to Lead
   if((Alpha>=-36)&&(Alpha<36)){
   Lead=3;
   }
   else if((Alpha>=36)&&(Alpha<108)){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Lead=2;
    Alpha=Alpha-72;
    }

    else if((Alpha>=108)&&(Alpha<180)){
    Lead=1;
    Alpha=Alpha-144;
    }

    else if((Alpha>=-180)&&(Alpha<-108)){
    Lead=5;
    Alpha=Alpha+144;
    }

    else if((Alpha>=-108)&&(Alpha<-36)){
    Lead=4;
    Alpha=Alpha+72;
    }
//Display parameter
    printf("\nX = %.2f,Y = %.2f",X,Y);
    printf("\nRotate = %.2f degree",Alpha);
    printf("\nLead = Leg%d",Lead);
//   getch();
    Rotate(Alpha);
    WalkGen(Distance,Lead);

}

void CallWalkGen(){
    float X,Xmin=-1000,Xmax=1000,Y,Ymin=-1000,Ymax=1000;

    printf("\nPlease insert position X in millimatre[-1000 to 1000]. X = ");
    scanf(" %.2f",&X);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while((X<Xmin)||(X>Xmax)){
    printf("\nWrong input X must between -1000 to 1000.");
    printf("\nInsert new input X = ");
    scanf(" %.2f",&X);
    }

    printf("\nPlease insert position Y in millimatre[-1000 to 1000]. Y = ");
    scanf(" %.2f",&Y);
    while((Y<Ymin)||(Y>Ymax)){
    printf("\n!Wrong input Y must between -1000 to 1000.");
    printf("\nInsert new input Y = ");
    scanf(" %.2f",&Y);
    }

    printf("\nX = %.2f  \nY = %.2f ",X,Y);
    SentXY(X,Y);

}



void Set90(){
    int X;
    for(X=0;X<=15;X++){
        ServoDrive(X,90);
        Sleep(100);
    }
    printf("\nSet90");
}

//---------------------------------------------------------------------------

//Main Function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//This is the main function which control step of program in order.
//Open com port --> Set 90Degree --> StanBy for walk --> Walk Gengerate --> End

int main(int argc, char *argv[])
{
   int Flag=1;
   char Quick='n';
   float x;

//Set BaudRate and Open comport
   printf("Open Comport.");
   SetPort();
   printf("\nStanBy.");
   StandBy();



/*
   printf("\n----------Line80----------");
   printf("\nPress any key to start.");
   getch();
   WalkGen(1000,2);
*/



/*
   printf("\n----------Rotate360----------");
   printf("\nPress any key to start.");
   getch();
   Rotate(-180);
   Rotate(-180);
/*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
   printf("\n----------Rectangle----------");
   printf("\nPress any key to start.");
   getch();
   WalkGen(400,3);
   Rotate(-90);
   WalkGen(400,3);
   Rotate(-90);
   WalkGen(400,3);
   Rotate(-90);
   WalkGen(400,3);
*/

   printf("\n----------Show!----------");
   printf("\nPress any key to start.");
   getch();

   Gnd = -(80.00+28.55);
   StandBy();
   sleep(500);

   Gnd = -(40.00+28.55);
   StandBy();
   sleep(500);

   Gnd = -(80.00+28.55);
   StandBy();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sleep(500);

Gnd = -(40.00+28.55);
StandBy();
sleep(500);

Gnd = -(80.00+28.55);
StandBy();
sleep(500);

Gnd = -(40.00+28.55);
StandBy();
sleep(500);

Gnd = -(80.00+28.55);
StandBy();
sleep(500);

Gnd = -(40.00+28.55);
StandBy();
sleep(500);

Gnd = -(80.00+28.55);
StandBy();
sleep(500);

Gnd = -(20.00+28.55);
WalkGen(80,3);
printf("\nLow.");

Gnd = -(60.00+28.55);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WalkGen(160,3);
printf("\nHigh.");

Gnd = -(30.00+28.55);
WalkGen(80,3);
printf("\nLow.");

printf("\nHand plese ^_^");

Inverse(2,150,0,120);
Inverse(4,150,0,120);
sleep(300);
Inverse(2,150,0,0);
Inverse(4,150,0,0);
sleep(300);
Inverse(2,150,0,120);
Inverse(4,150,0,120);
sleep(300);
Inverse(2,150,0,0);
Inverse(4,150,0,0);
sleep(300);
Inverse(2,150,0,120);
Inverse(4,150,0,120);
sleep(300);
Inverse(2,150,0,0);
Inverse(4,150,0,0);
sleep(300);
Inverse(2,150,0,120);
Inverse(4,150,0,120);
sleep(300);
Inverse(2,150,0,0);
Inverse(4,150,0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printf("\nBye bye.");
Inverse(2,150,30,120);
Inverse(4,150,-30,120);
sleep(300);
Inverse(2,150,-30,120);
Inverse(4,150,30,120);
sleep(300);
Inverse(2,150,30,120);
Inverse(4,150,-30,120);
sleep(300);
Inverse(2,150,-30,120);
Inverse(4,150,30,120);
sleep(300);
Inverse(2,150,30,120);
Inverse(4,150,-30,120);
sleep(300);
Inverse(2,150,-30,120);
Inverse(4,150,30,120);
sleep(300);
Inverse(2,150,30,120);
Inverse(4,150,-30,120);
sleep(300);
Inverse(2,150,-30,120);
Inverse(4,150,30,120);
sleep(300);
Inverse(2,150,30,120);
Inverse(4,150,-30,120);
sleep(300);
Inverse(2,150,-30,120);
Inverse(4,150,30,120);
sleep(300);
Inverse(2,150,30,120);
Inverse(4,150,-30,120);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    sleep(300);
    Inverse(2,150,-30,120);
    Inverse(4,150,30,120);
    sleep(300);

//Display status
    printf("\n-: Finish :-\n");
    system("PAUSE");
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้