

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมเสถียรภาพของหุ่นยนต์เดินสองขา

Stability Control of Bipedal Robot



เลขหมู่.....
เลขทะเบียน..... **62111**
วัน,เดือน,ปี **3 1 ก.ค. 2549**

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์เชิงกล
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมเสถียรภาพของหุ่นยนต์เดินสองขา

Stability Control of Bipedal Robot

โดย

นาย คุณธรรม เสงี่ยมเนตร

รหัสนี้ 44010054

นาย ชวลิต วงศ์ตั้งถิ่นฐาน

รหัสนี้ 44010094

อาจารย์ที่ปรึกษา

อ. คอน อิศรากร

อาจารย์ที่ปรึกษาร่วม

ผ.ศ. ถาวร เบญจนาสุทธี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์เชิงกล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2547

ภาควิชาระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมเสถียรภาพของหุ่นยนต์เดินสองขา

Stability Control of Bipedal Robot

ผู้จัดทำ

1. นาย คุณธรรม เสงี่ยมเนตร
2. นาย ชวติต วงศ์ตั้งถิ่นฐาน

..... *ดร. อิศรากร*

(อ.คอน อิศรากร)

อาจารย์ที่ปรึกษา

..... *ดร. เภมฉัตร*

(อ.ถาวร เภมฉัตร)

อาจารย์ที่ปรึกษาร่วม

(อ.ถาวร เภมฉัตร)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมเสถียรภาพของหุ่นยนต์เดินสองขา

คุณธรรม เส็งี่ยมเนตร

ชวลิต วงศ์ตั้งถิ่นฐาน

อาจารย์ที่ปรึกษา

อ. ดอน อิศรากร

อาจารย์ที่ปรึกษาร่วม

ผ.ศ. ถาวร เบญจนราสุทธี

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้กล่าวถึงการศึกษากระบวนการเคลื่อนไหวในท่าทางการเดินของมนุษย์ เพื่อเชื่อมโยงไปสู่การควบคุมเสถียรภาพการเดินของหุ่นยนต์เดินสองขา ให้มีการเดินอย่างสมดุล คล้ายมนุษย์ โดยที่หุ่นยนต์เดินสองขาถูกพัฒนาขึ้นมา มีจำนวนข้อต่อทั้งหมด 12 ข้อต่อคล้ายกับขา มนุษย์ ขาหุ่นยนต์ถูกขับเคลื่อนด้วยเซอร์โวมอเตอร์ ที่ได้รับสัญญาณควบคุมแบบสัญญาณพัลส์ (Pulse Width Modulate Signal, PWM) จากอุปกรณ์ควบคุมอิเล็กทรอนิกส์ (Micro Controller PIC) ที่ ถูกป้อนโปรแกรมการทำงาน และติดต่อข้อมูลผ่านเครื่องคอมพิวเตอร์ (Interface) นอกจากนี้ สัญญาณจากเซนเซอร์วัดความเอียง (Tilt sensor) ที่ติดตั้งที่ตัวหุ่น จะส่งแสดงค่ามุมการโน้มเอียง ในขณะที่หุ่นยนต์กำลังเดินได้เป็นอย่างดี ทำให้การควบคุมหุ่นยนต์ให้เดินอย่างมีประสิทธิภาพสามารถทำได้

Abstract

This paper presents the study of human locomotion in order to control the small-sized bipedal robot that will be performed a stable smooth walking. The developed bipedal robot has 12 DOF like human legs and is driven by using servomotors in each joint. The pulse width modulate signals are sent to control motors by using the custom-built controller (Micro Controller PIC) which is programmed and interfaced from the personal computer. Furthermore, the tilt angle from tilt sensor installed within the robot will effectively display the robot tilt while walking to user. Also, the controlling bipedal robot can be made easily.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

ชื่อเรื่อง	หน้า
บทคัดย่อ	I
สารบัญ	II
สารบัญรูปภาพ	IV
สารบัญตาราง	VII
บทที่ 1 บทนำ	1
บทที่ 2 รูปแบบการก้าวเดิน	3
2.1 ส่วนประกอบหลักที่ใช้ในการเดินของมนุษย์	3
2.1.1 ข้อต่อสะโพก	3
2.1.2 ข้อต่อเข่า	3
2.1.3 ข้อเท้าและเท้า	3
2.2 รูปแบบการเดินของมนุษย์	4
2.3 ทรวดทรงของการย่างก้าว	6
2.4 ปัจจัยสำคัญของการก้าวเดิน	8
2.5 การทรงตัวและการเดินของหุ่นยนต์	12
2.6 การทรงตัว	15
2.7 การก้าวเดิน	15
2.8 รูปแบบการเดินของหุ่นยนต์	16
บทที่ 3 การออกแบบโครงสร้างและส่วนควบคุม	21
3.1 ระบบการทำงานของหุ่นยนต์ Maimo1	21
3.2 ระบบการควบคุมตำแหน่ง	22
3.3 การออกแบบเชิงกล	25
3.4 รายละเอียดเซอร์โวมอเตอร์ที่เลือกใช้	28
3.5 ส่วนวงจรควบคุม	31
3.5.1 การติดต่อกับเครื่องคอมพิวเตอร์	31
3.5.2 ไมโครคอนโทรลเลอร์ PIC	36
3.5.3 รายละเอียดของเซนเซอร์วัดความเอียง	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเรื่อง	หน้า
3.5.4 การคำนวณค่ามุมเอียงจากเซนเซอร์วัดมุมเอียง	42
3.6 การออกแบบโปรแกรมควบคุมหุ่นยนต์ Maimo1	45
3.6.1 โปรแกรมติดต่อกับผู้ใช้	46
3.6.2 รายละเอียดการทำงานของโปรแกรม	52
บทที่ 4 ผลการทดลอง	70
4.1 ผลการทดลองการเดินของหุ่นยนต์	70
4.2 ผลการวัดค่ามุมเอียงจากเซนเซอร์วัดมุมเอียง	80
บทที่ 5 สรุปผลการดำเนินงาน	85
5.1 สรุปผลการดำเนินงาน	85
5.2 ปัญหาและอุปสรรค	85
5.3 แนวทางการพัฒนาต่อ	87
กิตติกรรมประกาศ	88
เอกสารอ้างอิง	89
ภาคผนวก	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

ชื่อรูป	หน้า
รูปที่ 1.1 แสดงไดอะแกรมการทำงานของหุ่นยนต์ Maimo1	1
รูปที่ 2.1 แสดงข้อต่อแบบลูกบอลกลมในเบ้าของข้อต่อสะโพก	3
รูปที่ 2.2 แสดงข้อต่อเข้าแบบบานพับ	4
รูปที่ 2.3 แสดงข้อเท้าด้านหน้าและด้านข้าง	4
รูปที่ 2.4 แสดงลักษณะการก้าวเดินปกติ ช่วงเวลา Stance 62% ช่วงเวลา Swing 38% และช่วงเวลาที่เท้าทั้งสองข้างสัมผัสพื้น (Double Limb Support) 25 %	5
รูปที่ 2.5 แสดงการแบ่งหมวดหมู่ของการก้าวเดินปกติ	6
รูปที่ 2.6 แสดงภาระโหลด (น้ำหนัก) ที่เกิดขึ้นแต่ละช่วงเวลาในระหว่างการตั้งเท้า (Stance)	7
รูปที่ 2.7 แสดงคาบเวลาการก้าวเดินปกติ	8
รูปที่ 2.8 แสดงท่าทางการก้าวเดินปกติ	8
รูปที่ 2.9 แสดงการเคลื่อนที่ของจุด CG ที่ลำตัว และการเคลื่อนที่ของข้อเท้า	9
รูปที่ 2.10 แสดงการหมุนของกระดูกเชิงกรานขณะก้าวขาขวา (Pelvic rotation) และการเอียงของกระดูกเชิงกราน (Pelvic tilt)	9
รูปที่ 2.11 แสดงการเคลื่อนมาด้านขวาของกระดูกเชิงกรานระหว่างการตั้งเท้า (Stance)	11
รูปที่ 2.12 แสดงภาพถ่ายจำลองในการเคลื่อนไหว	12
รูปที่ 2.13 วัฏจักรการเดินของมนุษย์	12
รูปที่ 2.14 แสดงมุมการเคลื่อนที่ของแต่ละข้อต่อ	13
รูปที่ 2.15 แสดงพื้นที่ภายใต้การเดินของหุ่นยนต์	13
รูปที่ 2.16 ลักษณะการเดินของหุ่นยนต์ที่ถูกออกแบบไว้	16
รูปที่ 2.17 แสดงตำแหน่งเซอร์โวมอเตอร์ที่ข้อต่อ เมื่อมองจากด้านหน้า	17
รูปที่ 2.18 แสดงตัวอย่างการก้าวเดินลำดับที่ 1 ถึง 7	17
รูปที่ 3.1 แสดง ไดอะแกรมการทำงานของหุ่นยนต์ Maimo1 โดยรวม	21
รูปที่ 3.2 แสดงผลตอบสนองของระบบเซอร์โวต่อการเปลี่ยนแปลงอินพุตแบบสแต็ป	23
รูปที่ 3.3 แสดงผลตอบสนองของระบบเซอร์โวต่อการเปลี่ยนแปลงโหลด	23
รูปที่ 3.4 แสดงหุ่นยนต์ Maimo1	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อรูป	หน้า
รูปที่ 3.5 แสดงรูปวาดส่วนข้อต่อสะโพก (3 DOF) และข้อต่อเข่า (1 DOF)	26
รูปที่ 3.6 ไดอะแกรมความยาวค่าน้ำส่วนขาของหุ่นยนต์ Maimo1	26
รูปที่ 3.7 แสดงข้อต่อส่วนสะโพก (3 DOF)	27
รูปที่ 3.8 แสดงข้อต่อเข่า (1 DOF)	27
รูปที่ 3.9 แสดงข้อต่อข้อเท้า (2 DOF)	27
รูปที่ 3.10 แสดงรายละเอียดภายในเซอร์โวมอเตอร์	28
รูปที่ 3.11 แสดงสัญญาณคาบเวลา ช่วง Duty cycle	28
รูปที่ 3.12 แสดงการหมุนของมอเตอร์ที่คาบเวลาต่างๆ	29
รูปที่ 3.13 แสดงบล็อกไดอะแกรมภายในวงจรจับเซอร์โวมอเตอร์	29
รูปที่ 3.14 แสดงบล็อกไดอะแกรมของวงจรภายในเซอร์โวมอเตอร์	30
รูปที่ 3.15 แสดงผลของฮีสเตอร์รีซิส	30
รูปที่ 3.16 แสดงการสื่อสารอนุกรมซิงโครนัสอย่างง่าย	32
รูปที่ 3.17 แสดงการสื่อสารอนุกรมอะซิงโครนัสอย่างง่าย	33
รูปที่ 3.18 แสดงการจัดวางตำแหน่งขาคอนเน็คเตอร์แบบ DB-9	34
รูปที่ 3.19 การเชื่อมต่อตัวรับตัวส่งแบบทั่วไป	34
รูปที่ 3.20 แสดงโครงสร้างภายในและตำแหน่งขาของ Max 232	36
รูปที่ 3.21 แสดงรายละเอียดของตำแหน่งขาไมโครคอนโทรลเลอร์ PIC	37
รูปที่ 3.22 เซนเซอร์วัดความเอียง	39
รูปที่ 3.23 บล็อกไดอะแกรมภายในเซนเซอร์วัดความเอียง	40
รูปที่ 3.24 แสดงเอาต์พุตของเซนเซอร์คาบเวลาช่วง T1 “ON”	41
รูปที่ 3.25 วงจรเชื่อมต่อเซนเซอร์วัดมุมเอียง โดยไมโครคอนโทรลเลอร์ PIC	41
รูปที่ 3.26 แสดงค่ามุม X ที่เปลี่ยนแปลง (แกน Y) เทียบกับคาบเวลาช่วง T1 “On” (%Duty cycle, แกน X) เนื่องจากการหมุนเอียงเซนเซอร์รอบแกน Y	42
รูปที่ 3.27 แสดงค่ามุม Y ที่เปลี่ยนแปลง (แกน Y) เทียบกับคาบเวลาช่วง T1 “On” (%Duty cycle, แกน Y) เนื่องจากการหมุนเอียงเซนเซอร์รอบแกน X	44
รูปที่ 3.28 แสดงบล็อกไดอะแกรมของหุ่นยนต์ Maimo1	45
รูปที่ 3.29 แสดงการลำดับการทำงานของโปรแกรมติดต่อกับผู้ใช้แบบอัตโนมัติ (Automatic)	46
รูปที่ 3.30 แสดงลำดับการทำงานของโปรแกรมติดต่อกับผู้ใช้แบบบังคับโดยผู้ใช้ (Manual)	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อรูป	หน้า
รูปที่ 3.31 แสดงหน้าต่างโปรแกรมควบคุมหุ่นยนต์ Maimo1	48
รูปที่ 3.32 แสดงส่วนประกอบส่วนที่ 1 ของโปรแกรมติดต่อผู้ใช้	49
รูปที่ 3.33 แสดงส่วนประกอบที่ 2 ของโปรแกรมติดต่อผู้ใช้	49
รูปที่ 3.34 เมื่อผู้ใช้กดคำสั่งเซท	50
รูปที่ 3.35 แสดงเมื่อผู้ใช้กดปุ่มตั้งเพียงจังหวัดเดียว	50
รูปที่ 3.36 แสดงให้เห็นค่าที่ส่งออกไปในทุกจังหวัด	51
รูปที่ 3.37 แสดงการรับค่ากลับมาของเซนเซอร์	51
รูปที่ 3.38 แสดงการป้อนค่าระยะเวลาที่ห่างกันของแต่ละจังหวัด	51
รูปที่ 3.39 แสดงการเคลื่อนที่และช่วงเวลาที่เปลี่ยนไปของ t	51
รูปที่ 3.40 เป็นปุ่มที่ใช้ส่งค่าออกไปทั้งหมด	52
รูปที่ 3.41 แสดงการตรวจสอบและการเก็บข้อมูลของไมโครคอนโทรลเลอร์	58
รูปที่ 3.42 แสดงลำดับการทำงานของไมโครคอนโทรลเลอร์หลัก	64
รูปที่ 3.43 แสดงลำดับการทำงานของไมโครคอนโทรลเลอร์รอง	65
รูปที่ 4.1 แสดงหุ่นยนต์ Maimo1	70
รูปที่ 4.2 แสดงการเดินทางในมุมด้านซ้ายของหุ่นยนต์ Maimo1	71
รูปที่ 4.3 กราฟความสัมพันธ์การเดินทางระหว่างค่าเปอร์เซ็นต์ความเร็วเคลื่อนที่ และจังหวัดการเดินทางในซ้าย	78
รูปที่ 4.4 กราฟความสัมพันธ์การเดินทางระหว่างค่าเปอร์เซ็นต์ความเร็วเคลื่อนที่ และจังหวัดการเดินทางในขวา	78
รูปที่ 4.5 ความสัมพันธ์ของค่ามุมเซนเซอร์แนวแกน X และ Y เทียบกับ จังหวัดการเดินทางทั้ง 16 จังหวัด	79
รูปที่ 4.6 กราฟค่ามุมของเซนเซอร์แนวแกน X และ Y เทียบกับ เวลาที่ใช้เดินที่ใช้เดินทั้ง 16 จังหวัด	79
รูปที่ 4.7 ผลการวัดค่าที่แกน X ที่ +90 องศา	80
รูปที่ 4.8 ผลการวัดค่าที่แกน X ที่ -90 องศา	81
รูปที่ 4.9 ผลการวัดค่าที่แกน Y ที่ +90 องศา	81
รูปที่ 4.10 ผลการวัดค่าที่แกน Y ที่ -90 องศา	81
รูปที่ 4.11 แสดงกราฟค่ามุม X เทียบกับค่าการเปลี่ยนแปลงคาบเวลาของ X	82
รูปที่ 4.12 แสดงกราฟค่ามุม Y เทียบกับค่าการเปลี่ยนแปลงคาบเวลาของ Y	83

สารบัญตาราง

ชื่อตาราง	หน้า
ตารางที่ 1 แสดงรายละเอียดตำแหน่งขาสัญญาณที่ใช้ภายในวงจรควบคุม PIC 16F877/A	38
ตารางที่ 2 แสดงรายละเอียดตำแหน่งขาสัญญาณเซนเซอร์วัดมุมเอียง	40
ตารางที่ 3 แสดงค่ามุมแนวแกน X เทียบกับค่าการเปลี่ยนแปลงของ X	82
ตารางที่ 4 แสดงค่ามุมแนวแกน Y เทียบกับค่าการเปลี่ยนแปลงของ Y	83



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

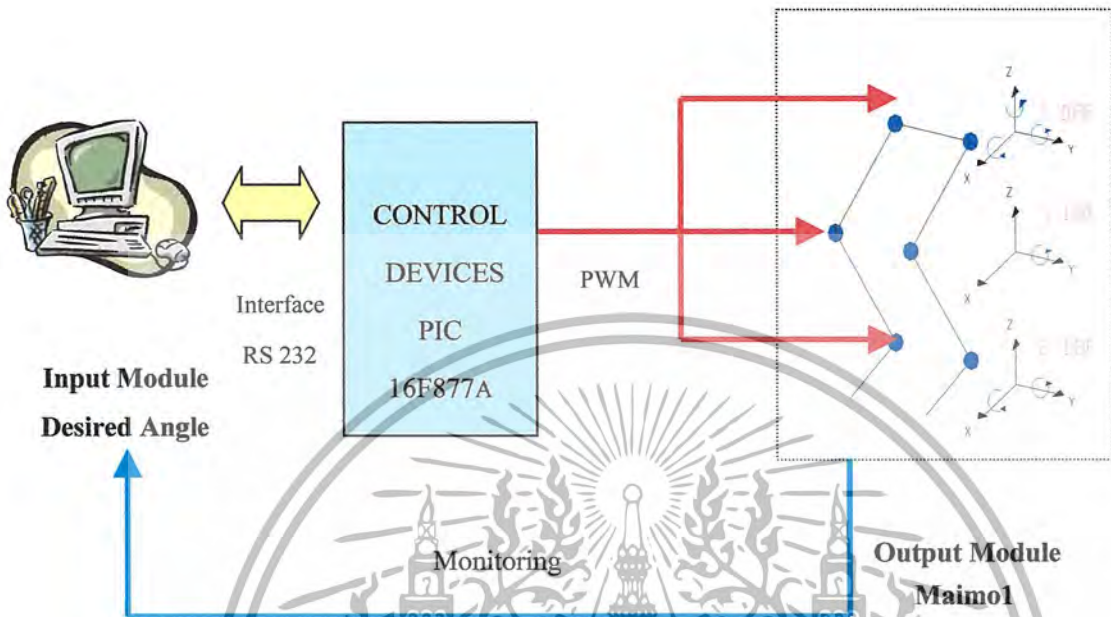
บทที่ 1

บทนำ

เทคโนโลยีการออกแบบและการพัฒนาหุ่นยนต์ระบบอัตโนมัติได้มีการพัฒนามานานแล้ว ซึ่งล้วนแต่มีจุดมุ่งหมายในการใช้งานแทนมนุษย์ เริ่มแรกหุ่นยนต์เคลื่อนที่ด้วยล้อนั้นมีความสามารถในการเคลื่อนที่ได้ดีแต่ไม่ทุกสภาวะเช่น ไม่สามารถเดินขึ้นบันได และล้อจับเคลื่อนที่ได้ตามเส้นทางที่ถูกสร้างขึ้นเท่านั้น และถ้าเปรียบเทียบกับเคลื่อนที่ของมนุษย์ซึ่งสามารถเคลื่อนที่ได้อย่างอิสระ ดังนั้นหุ่นยนต์สองขา (Bipedal robot) จึงได้ถูกสร้างและพัฒนาขึ้นมาให้มีความสามารถในการเคลื่อนที่คล้ายมนุษย์ และเพื่อที่จะใช้งานหุ่นยนต์ทำงานในสภาวะแวดล้อมที่ไม่เอื้ออำนวย เช่น พื้นผิวที่ทุรกันดาร หรือสภาวะแวดล้อมที่เป็นอันตราย เช่น ภายในโรงงานปฏิกรณ์นิวเคลียร์ หรือสภาวะแวดล้อมที่เป็นพิษ ที่มนุษย์ไม่ต้องการเสี่ยงอันตรายในการทำงาน นั้นเป็นเหตุผลหลักๆให้หุ่นยนต์เดินสองขาได้ถูกพัฒนาขึ้นมา

แนวทางการศึกษาและพัฒนาหุ่นยนต์เดินสองขาได้แบ่งเป็นสองหัวข้อคือการออกแบบเชิงกลของหุ่นยนต์ (Designed Robot) และการออกแบบโปรแกรมการควบคุม (Programming Design) หุ่นยนต์ที่ได้รับการออกแบบและสร้างขึ้นมาชื่อหุ่นยนต์ Maimo1 จะมีการเคลื่อนที่ของข้อต่อเลียนแบบสรีระวิทยาของมนุษย์ นั่นคือ ขาทั้งสองข้าง สามารถเคลื่อนที่ได้อิสระเหมือนขามนุษย์ ซึ่งมี จำนวนข้อต่อทั้งหมดอยู่ 12 องศาอิสระ (12 Degree Of Freedom (DOF)) โดยแต่ละข้างจะประกอบด้วย สะโพก (3 DOF) เข่า (1 DOF) และข้อเท้ามี (2 DOF) แสดงไดอะแกรมได้ดังรูปที่ 1.1

ข้อต่อของหุ่นยนต์ Maimo1 ที่ถูกจับเคลื่อนโดยเซอร์โวมอเตอร์ที่มีขายตามท้องตลาดถูกประกอบเป็นข้อต่อในแต่ละข้อต่อร่วมกับทฤษฎีการควบคุมตำแหน่ง (Position control) การที่จะออกแบบและควบคุมหุ่นยนต์เดินสองขาให้เดินคล้ายกับมนุษย์นั้น การศึกษาการเดินของมนุษย์เป็นสิ่งสำคัญในการเริ่มดำเนินงานอันดับแรก (Gait cycle) จากนั้นจึงเชื่อมโยงลักษณะท่าทางการเดินของมนุษย์ไปยังหุ่นยนต์ โดยการควบคุมเชิงตำแหน่ง ผ่านโปรแกรมที่ถูกพัฒนาขึ้นเพื่อใช้ส่งสัญญาณควบคุมแบบพัลส์ (Pulse width modulate, PWM) จากเครื่องคอมพิวเตอร์ส่วนบุคคล ผ่านไมโครคอนโทรลเลอร์ (PIC) เพื่อควบคุมหุ่นยนต์เดินสองขาให้เดินได้อย่างมีประสิทธิภาพ สามารถอธิบายได้ดัง รูปที่ 1.1



รูปที่ 1.1 แสดง โดอะแกรมการทำงานของหุ่นยนต์ Maim01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

รูปแบบการก้าวเดิน

ในการสร้างหุ่นยนต์เดินสองขาให้เคลื่อนที่ในรูปแบบการเดินของมนุษย์ (Gait pattern) สิ่งที่เป็นอันดับแรกคือการศึกษารูปแบบการเคลื่อนที่ของมนุษย์ (Human locomotion) และในส่วนของสร้างแบบจำลอง (หุ่นยนต์สองขา) จำเป็นที่จะต้องศึกษารายละเอียดที่เกี่ยวข้องไม่ว่าจะเป็น ส่วนประกอบที่ทำให้เกิดการเดิน และลักษณะการทำงานแต่ละส่วน แล้วจึงนำมาหาความสัมพันธ์ทั้งหมดเพื่อเชื่อมโยงลักษณะท่าทางการเดิน ไปยังหุ่นยนต์สองขา

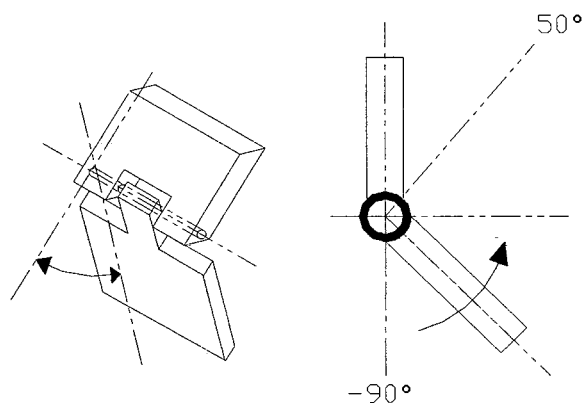
2.1 ส่วนประกอบหลักที่ใช้ในการเดินของมนุษย์

2.1.1 ข้อสะโพก (Hip Joint) เป็นข้อต่อแบบลูกบอลสวมเข้าไปในเบ้า (Ball & Socket) สามารถหมุนได้อย่างอิสระทั้ง 3 แนวแกน (Roll (X), Pitch (Y), Yaw (Z)) หรือ 3 Degree of freedom (DOF) ทำให้ข้อสะโพกสามารถหมุนได้รอบๆ จุดหมุน โดยทั่วไปสามารถงอได้ตั้งแต่ 0 ถึง 125 องศา ดังรูปที่ 2.1



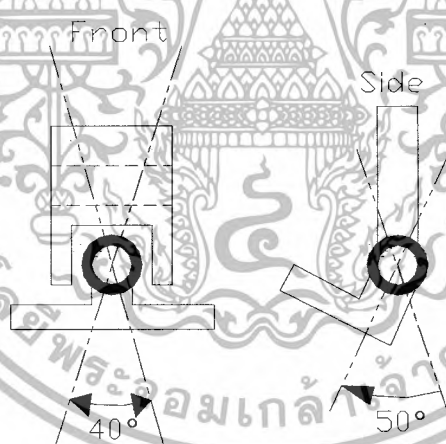
รูปที่ 2.1 แสดงข้อต่อแบบลูกบอลสวมในเบ้าของข้อต่อสะโพก

2.1.2 ข้อต่อเข่า (Knee Joint) เป็นข้อต่อชนิดแบบบานพับ (Hinge type) มีการเคลื่อนไหวแบบอิสระเพียงแกนเดียวเท่านั้น (1 DOF) ทำให้ขาสามารถงอ (Flex) และเหยียดได้ (Extend) โดยทั่วไปจะงอได้สูงสุดประมาณ 140 องศา ดังรูปที่ 2.2



รูปที่ 2.2 แสดงข้อต่อเข้าแบบบานพับ

2.1.3 ข้อเท้าและเท้า (Ankle & foot) สามารถหมุนได้อิสระเพียง 2 แกน (2 DOF) ถูกออกแบบมาให้เท้าเคลื่อนที่ในขณะที่ก้าวเดินได้ และเท้ามีไว้เป็นฐานรองรับน้ำหนักของตัวหุ่นยนต์ในขณะที่ยืนหรือก้าวเดิน โดยปกติข้อเท้าสามารถงอได้ประมาณ 0 ถึง 50 องศา แต่อีกแนวแกนที่ข้อเท้าบิดไปได้ประมาณ 40 องศาเท่านั้น ดังรูปที่ 2.3



รูปที่ 2.3 แสดงข้อเท้าด้านหน้าและด้านข้าง

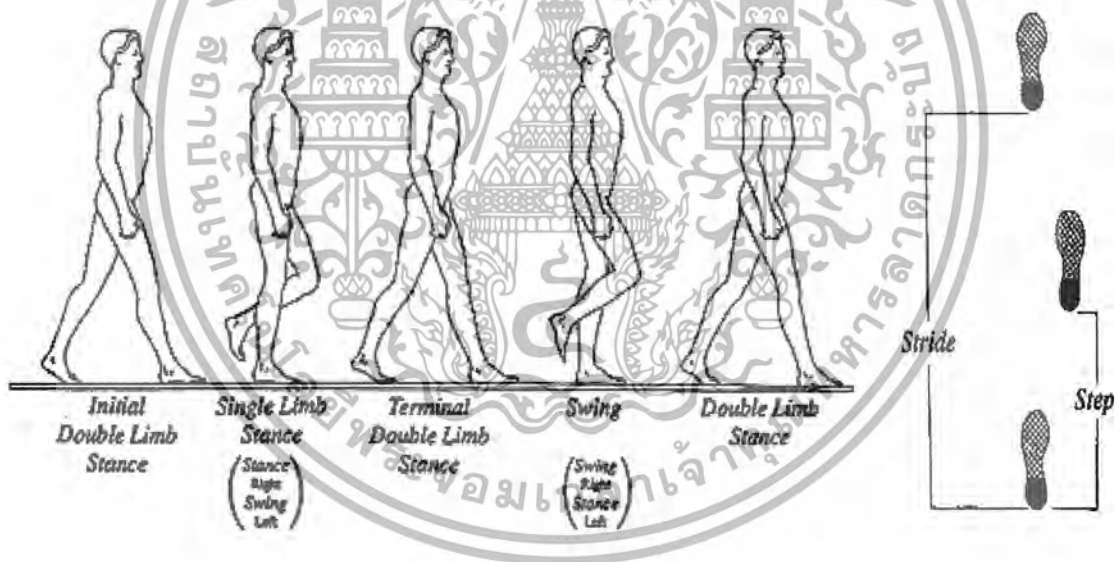
2.2 รูปแบบการเดินของมนุษย์ (Division of Gait Cycle)

รูปแบบการก้าวเดินของมนุษย์ถูกจำกัดโดยรูปร่างขา ตำแหน่งของกล้ามเนื้อและระบบประสาท โดยระบบประสาทจะสั่งการควบคุมการยืดหดของกล้ามเนื้อที่ข้อต่อ ทำให้เกิดการเคลื่อนไหวขึ้น จุดมุ่งหมายหลักคือ การใช้พลังงานของร่างกายที่ประหยัดที่สุดในการขนถ่ายน้ำหนักตัวไปข้างหน้าอย่างสมดุล ซึ่งจะต้องอาศัยการทำงานร่วมกันของระบบต่างๆ ในการเคลื่อนที่ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหนึ่งรอบของการก้าวเท้า (Stride) จะเกี่ยวพันกับคาบเวลาระหว่างสองเหตุการณ์ คือ เวลาที่เท้าข้างใดข้างหนึ่งสัมผัสพื้น (Stance) และเวลาที่เท้าอีกข้างหนึ่งลอยจากพื้น (Swing) สามารถอธิบายความสัมพันธ์ได้ เช่น สมมติว่าเท้าขวาและเท้าซ้ายเริ่มต้นสัมผัสพื้น จากนั้นก้าวเท้าซ้ายออกไปข้างหน้า ช่วงเวลาที่เท้าซ้ายลอยอยู่ในอากาศจะเรียกว่า Swing เพราะจะเกิดการโน้มตัวเอียงมาด้านขวา และขณะที่เท้าขวาสัมผัสพื้นอยู่ซึ่งน้ำหนักทั้งหมดจะถ่ายลงไปที่เท้าขวา จะเรียกช่วงเวลานี้ว่า Stance จนกระทั่งเท้าซ้ายสัมผัสพื้นและเท้าขวาเริ่มที่จะก้าวออกไปข้างหน้า ช่วงเวลา Stance และ Swing จะใช้เวลาประมาณ 62 % และ 38% ของเวลาทั้งหมด

ในการก้าวเดินปกติดังรูปที่ 2.4 ช่วงเวลาแรกก่อนที่เท้าซ้ายจะยกลอยจากพื้น (Initial Double Limb Support) จะใช้เวลาประมาณ 12 % ของเวลาที่ใช้ในการก้าวทั้งหมด และช่วงเวลาหลังจากเท้าซ้ายก้าวมาสัมผัสพื้น (Terminal Limb Support) จะใช้เวลาประมาณ 12 % เท่ากัน ดังนั้นในการก้าว 1 รอบ (Stride) ช่วงเวลาที่เท้าสองข้างสัมผัสพื้นจะใช้เวลารวมกันประมาณ 25 % ของเวลาทั้งหมด (Single Support เป็นช่วงเวลาที่เท้าข้างใดข้างหนึ่งสัมผัสพื้นหรือช่วง Swing นั่นเอง)



รูปที่ 2.4 แสดงลักษณะการก้าวเดินปกติ ช่วงเวลา Stance 62% ช่วงเวลา Swing 38% และช่วงเวลาที่เท้าทั้งสองข้างสัมผัสพื้น (Double Limb Support) 25 %

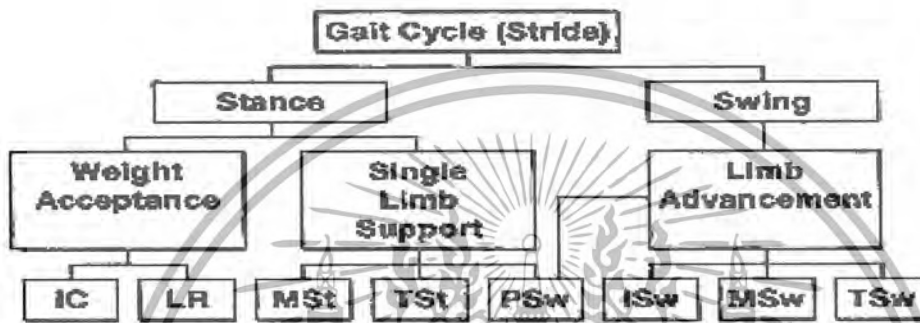
(อ้างอิงจาก http://www.oandp.com/jpo/library/1997_01_010.asp)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หมวดหมู่ของการย่างก้าว (Function Tasks of Gait)

การก้าวเดินปกติสามารถแบ่งได้เป็น 3 เทอม คือ การรับน้ำหนักด้วยขาสองข้าง (Weight acceptance) การรับน้ำหนักด้วยขาข้างเดียว (Single limb support) และการก้าวไปข้างหน้า (Limb advancement) สามารถแสดงได้ดังรูปที่ 2.5

Periods, Tasks, & Phases



รูปที่ 2.5 แสดงการแบ่งหมวดหมู่ของการก้าวเดินปกติ

(อ้างอิงจาก http://www.oandp.com/jpollibrary/1997_01_010.asp)

จากรูปที่ 2.5 แสดงความสัมพันธ์ของคาบเวลาของการก้าวเดิน จะพบว่าขั้นตอนของ Preswing มีส่วนทั้งในเฟสของ Stance และ Swing

เริ่มจากการรับน้ำหนักด้วยขาทั้งสองข้าง (Weight acceptance) จะประกอบด้วยส่วนย่อยคือการเริ่มต้นสัมผัส (IC , Initial contact) และการตอบสนองต่อน้ำหนัก (LR , Loading respond) เมื่อพิจารณารูปที่ 2.4 แรกเริ่มจะพบว่าขาทั้งสองข้างเริ่มจะสัมผัสพื้นพร้อมกัน ณ เวลานั้นจะเกิดการขนถ่ายน้ำหนักลงเท้าทั้งสองข้าง ขาทั้งสองข้างนั้นจะต้องรักษาเสถียรภาพจากสัมผัสพื้นของเท้าขวา ซึ่งช่วยแรงกระแทกที่เกิดขึ้นแล้วจะเข้าสู่ขั้นตอนต่อไปคือเท้าขวาสัมผัสพื้นรองรับน้ำหนักขาข้างเดียว เท้าซ้ายก้าวลอยในอากาศ จะเรียกว่า การรับน้ำหนักด้วยขาข้างเดียว (Single limb support) ซึ่งประกอบไปด้วย ช่วงครึ่งแรกของการรับน้ำหนักด้วยขาข้างเดียว (Mst , Mid stance) และช่วงท้ายของการรับน้ำหนักด้วยขาข้างเดียว (Tst , Terminal stance) จากรูป 2.4 จะพบว่าเท้าขวารองรับน้ำหนักและรักษาเสถียรภาพของร่างกายไว้ลำตัวโน้มเอียงมาทางด้านขวา ในขณะที่เท้าซ้ายก้าวพาตัวเคลื่อนไปข้างหน้า น้ำหนักจะเริ่มถ่ายไปยังขาซ้ายเมื่อใกล้สัมผัสพื้นมากขึ้น (Psw , Preswing) ขั้นตอนสุดท้ายคือ การก้าวไปข้างหน้า (Limb advancement) จะประกอบไปด้วย การถ่ายน้ำหนักจากการที่เท้าซ้ายสัมผัสพื้น (Psw , Preswing) จากนั้นจะเป็นช่วงที่ตัวเริ่มเอียงมาด้านซ้าย (ISw , Initial swing) สิ้นเท้าขวาจะลอยจากพื้นจนพาขาขวาเคลื่อนไปข้างหน้าจนเข้าขาขวาได้เต็มที่ แล้วจะเข้าสู่ช่วงครึ่งแรกของการโน้มเอียง (MSw , Mid swing) จากนั้นจะเข้าสู่ช่วงสุดท้าย (Terminal stance) เท้าขวาก้าวพาตัวเคลื่อนไป

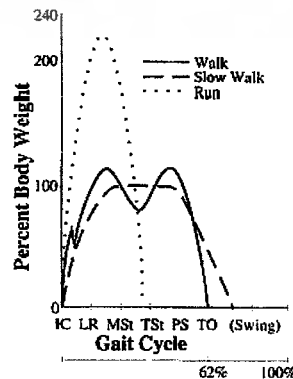
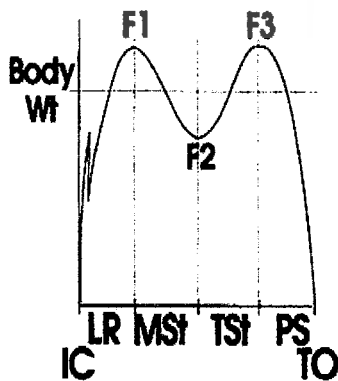
เอกส... โยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านหน้าจนสัมผัสพื้น(ขาขวาเหยียดออกเต็มที่จนสัมผัสพื้น) นำหนักถ่ายจากซ้ายมาขวาทำให้ลำตัวโน้มเอียงมาตรงกลาง จากจุดนี้จะเข้าสู่สภาวะเริ่มต้นใหม่ การก้าวเดินจะเป็นวัฏจักรแบบนี้เรื่อยไป (Gait cycle) ดังแสดงในรูปที่ 2.4 รูปที่ 2.7 และรูปที่ 2.8

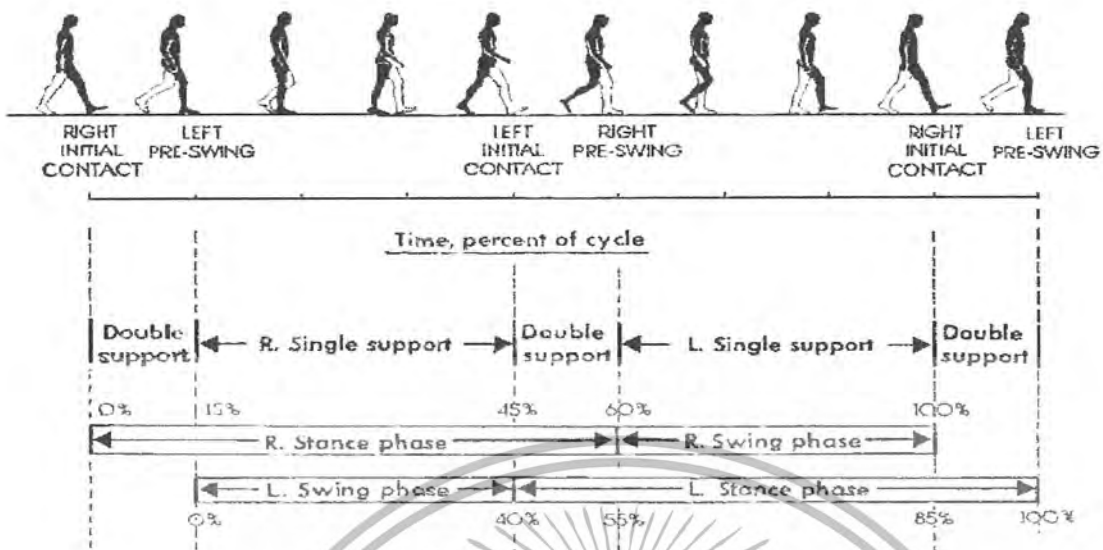
ดังนั้นวัฏจักรการก้าวเดิน (Gait Cycle) แบ่งเป็นเทอมของการตั้งเท้า (Stance) และเทอมของการแกว่งเท้า (Swing) ทั้งหมดนั้นสามารถสรุปได้ดังนี้

1. **Initial Contact** เป็นช่วงเวลาในขณะที่ปลายเท้ากำลังจะสัมผัสพื้น เพื่อเตรียมตัวรับ โหลดหรือนำหนักตัวทั้งหมดและเท้าอีกข้างเริ่มจะออกจากพื้น ช่วงเวลานี้เป็นจุดเริ่มต้นของการตั้งเท้า (Stance)
2. **Loading Response** เป็นช่วงเวลาที่น้ำหนักทั้งหมดถูกถ่ายเทมาที่เท้าที่กำลังเหยียบพื้น ในขณะที่เท้าอีกข้างหนึ่งกำลังยกจากพื้น
3. **Mid Stance** เป็นช่วงเวลาที่เท้าที่ไม่ได้สัมผัสพื้นงอเข้าเต็มที่ ยกสูงจากพื้นมาอยู่ในแนวตั้งตรงกับเท้าที่เหยียบพื้น
4. **Terminal Stance** เป็นช่วงเวลาที่ส้นเท้าข้างที่เหยียบพื้นอยู่เริ่มยกสูงขึ้น และในขณะที่เดียวกันปลายเท้าด้านที่ลอยจากพื้นกำลังจะสัมผัสพื้น
5. **Preswing** เป็นช่วงที่เท้าด้านที่เหยียบพื้นก่อน ได้ยกลอยจากพื้น และเท้าด้านอีกด้านหนึ่งได้เหยียบพื้นเต็มระนาบก่อเกิดโมเมนต์ขึ้น ข้อเท้าที่สัมผัสพื้นเริ่มจะหมุน
6. **Initial swing** เป็นช่วงที่ขาข้างที่กำลังลอยอยู่งอเข้าได้มากที่สุด
7. **Midswing** เป็นช่วงที่ขาที่กำลังมาข้างหนึ่งงอเข้าได้มากที่สุดหรือปลายเท้าทั้งสองอยู่ในแนวตั้ง
8. **Terminal swing** เป็นช่วงสุดท้ายที่เท้าที่กำลังเข้ามาสัมผัสพื้นหรือเข้าไม่งอ ขึ้นตอนต่อไปก็จะซ้ำลำดับเดิม

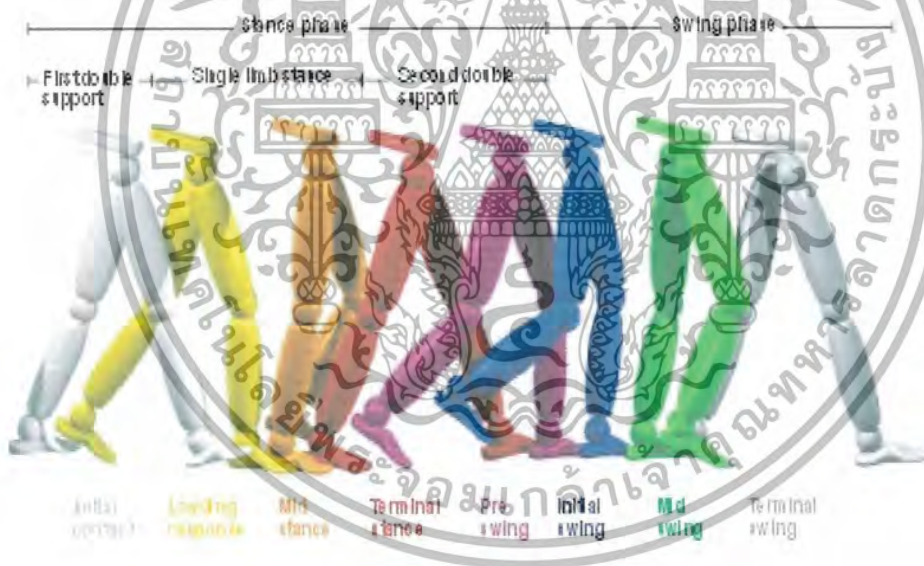
ในแต่ละช่วงการก้าวเดินปกตินั้นจะเกิดการถ่ายเทน้ำหนักไปที่ขาแต่ละข้าง สลับกันไปมา น้ำหนักที่เกิดขึ้นนั้นสามารถแสดงได้ดังกราฟรูปที่ 2.6



เอกสารนี้รูปที่ 2.6 แสดงการระโหลด (น้ำหนัก) ที่เกิดขึ้นแต่ละช่วงเวลาในระหว่างการตั้งเท้า (Stance) ระโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงคาบเวลาการก้าวเดินปกติ



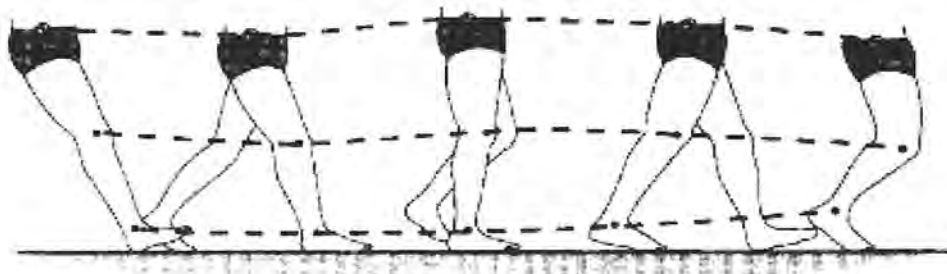
รูปที่ 2.8 แสดงท่าทางการก้าวเดินปกติ

(รูปที่ 2.6 ถึง รูปที่ 2.8 อ้างอิงจาก http://www.oandp.com/jpo/library/1997_01_010.asp)

2.4 ปัจจัยสำคัญของการก้าวเดิน (Determinants of Gait)

นิยามการเดิน หมายถึง การเคลื่อนที่ของจุดศูนย์กลางมวล (CG) ไปในทิศทางที่ต้องการ โดยจุด CG จะต้องกระเพื่อมขึ้นลงได้น้อยที่สุด ซึ่งเกิดจากข้อเท้าและเข่าเคลื่อนที่อย่างสัมพันธ์กันและจะทำให้เกิดการเดินที่ราบเรียบ ถ้าอยู่ในเงื่อนไขนี้แล้ว พลังงานจะถูกใช้ในการก้าวเดินน้อยที่สุด ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงการเคลื่อนที่ของจุด CG ที่ลำตัว และการเคลื่อนที่ของข้อเท้า

(อ้างอิงจาก http://www.oandp.com/jpo/library/1997_01_010.asp)

ปัจจัยที่มีผลกระทบที่ทำให้เสียพลังงานเพิ่มมี 6 ปัจจัยได้แก่

1. การหมุนของกระดูกเชิงกราน (Pelvic Rotation) ในการก้าวขาไปข้างหน้าจะทำให้กระดูกเชิงกรานหมุนรอบขาซ้ายที่สัมผัสพื้น โดยทั่วไปประมาณ 4 องศาเป็นผลให้จุด CG เคลื่อนลงอย่างรวดเร็ว หลังจากนั้นเท้าขวาก้าวไปสัมผัสพื้น ขาข้างซ้ายก็ก้าวเข้ามาเป็นผลให้กระดูกเชิงกรานหมุนรอบขาขวากลับคืนอีก 4 องศาเป็นผลให้ลำตัวตรง (จุดCGเคลื่อนขึ้นมาอยู่ในแนวระดับ) จากนั้นขาซ้ายก็ก้าวผ่านไปจนสัมผัสพื้นเป็นผลให้กระดูกเชิงกรานหมุนรอบขาขวากลับอีก 4 องศา (จุดCGเคลื่อนลงต่ำกว่าในแนวระดับ) รวมแล้วการก้าวขาซ้ายครบรอบ (Stride) กระดูกเชิงกรานหมุนรอบขาขวา 8 องศา แสดงได้รูปที่ 2.10



รูปที่ 2.10 แสดงการหมุนของกระดูกเชิงกรานขณะก้าวขาขวา (Pelvic rotation)

และการเอียงของกระดูกเชิงกราน (Pelvic tilt)

(อ้างอิงจาก http://www.oandp.com/jpo/library/1997_01_010.asp)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาด้านนี้ เมื่อนุญเตเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเอียงของกระดูกเชิงกราน (Pelvic Tilt) ที่ช่วงครึ่งแรกของการตั้งเท้า (Mid stance)

จุด CG อยู่ตำแหน่งสูงสุด ถ้าเท้าขวาเป็นเท้าที่รับน้ำหนัก ในขณะที่เดียวกับขาซ้ายลอยอยู่ในอากาศ กระดูกเชิงกรานจะเอียงลงมาด้านซ้ายประมาณ 5 องศา ทำให้จุด CG อยู่ต่ำกว่าเดิมเล็กน้อย หรือ กรณีสองเท้ารับน้ำหนักเท่ากัน ดังรูป 2.10 ทางขวา กระดูกเชิงกรานด้านซ้ายเอียงลงเล็กน้อย ปัจจัยของการหมุน หรือเอียงของกระดูกเชิงกรานมีผลต่อการกระเพื่อมขึ้นลงของจุด CG ซึ่งจะเกี่ยวข้องกับใช้พลังงานของร่างกายในการย่างก้าวด้วย

3. การงอเข่าระหว่างช่วงครึ่งแรกของการตั้งเท้า (Knee Flexion During Mid stance) ช่วงแรกที่เท้าขวาสัมผัสพื้น (เข้าขวาเหยียดตรง) และเท้าซ้ายเริ่มออกจากพื้น (Initial contact) จนกระทั่งเท้าซ้ายลอยจากพื้นก่อนมาอยู่ในช่วงครึ่งแรกของการตั้งเท้า (Mid stance) ช่วงนี้เองเท้าขวาจะงอเล็กน้อย (จุด CG เคลื่อนลง) จนกระทั่งเข้าขวาเหยียดตรงในช่วง Mid stance นี้ (จุด CG เคลื่อนขึ้น)

4. การเคลื่อนที่ของเท้าและข้อเท้า (Foot & Ankle Motion) เท้าและข้อเท้าเป็นส่วนสำคัญที่ทำให้เกิดการย่างก้าวที่ราบเรียบ (Smooth) ในช่วง Initial contact เริ่มแรก ดังรูปที่ 2.9 เท้าขวาเริ่มสัมผัสพื้น จะทำให้ข้อเท้าเริ่มเคลื่อนลง จนเท้าขวาแนบพื้นสนิท และเริ่มเคลื่อนขึ้นอีกครั้งเมื่อส้นเท้าขวาเคลื่อนขึ้น (Terminal stance & Preswing) จากรูปช่วง Stance นี้เองจุด CG จะเกิดการเคลื่อนที่อย่างราบเรียบ (Smooth)

5. การเคลื่อนที่ของเข่า (Knee Motion) การเคลื่อนที่ของเข่าจะเกี่ยวข้องกับการเคลื่อนที่ของเท้าและข้อเท้า ที่ Initial contact ดังรูปที่ 2.9 ก่อนเท้าขวาจะสัมผัสพื้นสนิท เข่าขวานั้น ได้เหยียดเต็มที่ จากนั้นข้อเท้าขวาจะเคลื่อนต่ำลงเป็นผลให้เข่าขวางอเล็กน้อย ข้อเท้าขวาจะอยู่ในระดับนี้ และเข่าขวาจะเหยียดตรงเต็มที่เมื่ออยู่ในช่วง Mid stance จนกระทั่งช่วง Terminal stance ผ่านไปเข่าขวาจะเริ่มงออีกครั้งพร้อมกับข้อเท้าขวาเคลื่อนขึ้น (Heel comes off) โดยทั่วไปแล้วเมื่อข้อเท้าถูกกดลง เข่าจะยืดออกไป แต่เมื่อข้อเท้าเคลื่อนขึ้น เข่าจะงอ การเคลื่อนที่ของเข่าจะเกี่ยวข้องกับเท้าและข้อเท้า แนวเส้นจุด CG จะเคลื่อนที่จะราบเรียบ และเกิดการใช้พลังงานในการเคลื่อนที่ได้น้อยที่สุด

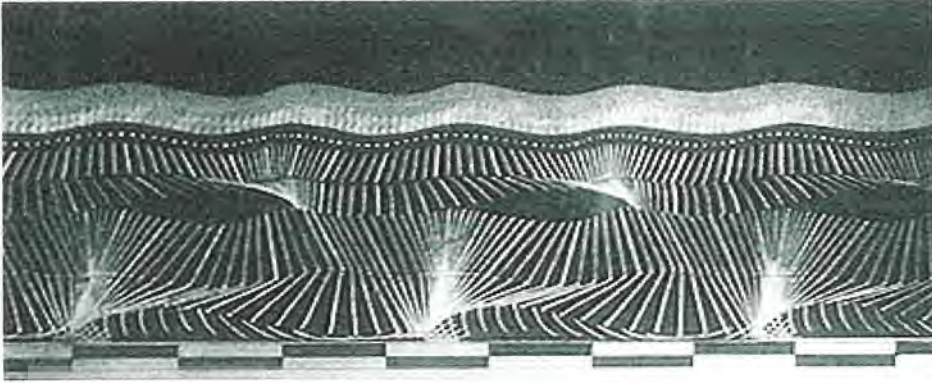
6. การขจัดข้างของกระดูกเชิงกราน (Lateral Pelvic Displacement) ในขณะที่อยู่ระหว่างการตั้งเท้า (Stance) กระดูกเชิงกรานจะเลื่อนไปด้านขาที่ยืนตั้งเท้า ระหว่างนั้นสะโพกอีกด้านหนึ่งก็จะเคลื่อนต่ำลง จุด CG ก็จะเลื่อนไปอยู่เหนือด้านขาที่สัมผัสพื้น ซึ่งการก้าวสั้นๆ แบบขาทั้งสองข้างชิดกันจะเป็นการลดระดับจุด CG ไม่ให้กระเพื่อมขึ้นสูงมากได้ (กระดูกเชิงกรานเคลื่อนที่เล็กน้อยมาทางด้านขาที่ยืนเหยียดพื้น) แต่จะเป็นการเดินที่ไม่มีความสะดวก แต่การก้าวแบบขาทั้งสองข้างห่างกันกว่าเดิม จะเพิ่มความสะดวกในการเดินได้มากกว่า แต่จุด CG จะเคลื่อนขึ้นลงมากกว่า และร่างกายอาจจะใช้พลังงานเพิ่มขึ้น การขจัดข้างของกระดูกเชิงกรานแสดงได้ดังรูปที่ 2.11



รูปที่ 2.11 แสดงการเคลื่อนมาด้านขวาของกระดูกเชิงกรานระหว่างการตั้งเท้า (Stance)
(อ้างอิงจาก http://www.oandp.com/jpollibrary/1997_01_010.asp)

ปัจจัยทั้งหมดทั้ง 6 ส่วนนี้จะเกิดที่ข้อต่อต่างๆทั้งหมดของกระดูกช่วงล่างของร่างกาย ถ้ามองง่าย ๆ แล้ว ก็จะเป็นเพียงการเหยียดขา (Extension) และการงอขา (Flexion) เท่านั้น แต่การเหยียดขาและการงอขาเกิดจากการทำงานของกล้ามเนื้อหลายส่วนที่สัมพันธ์กันมากเป็นผลให้เกิดการเดินขึ้น

สรุปได้ว่าการก้าวเดินปกติของมนุษย์ อาศัยส่วนประกอบที่ซับซ้อนของร่างกายและระบบพลังงานอัตโนมัติ การก้าวเดินอย่างสมดุลเป็นการต้านน้ำหนักจากแรงโน้มถ่วงเพื่อที่จะเคลื่อนที่พาร่างกายไปด้านหน้าและร่างกายใช้พลังงานน้อยที่สุด เสมือนว่าการเดินปกตินั้นทำให้มนุษย์รู้สึกไม่เหน็ดเหนื่อย เท่ากับการเดินที่ผิดแปลกไปจากการเดินปกติ ดังนั้นการเชื่อมโยงลักษณะการเดินของมนุษย์มายังหุ่นยนต์สองขาให้เดินก้าวเดินปกติ (จุด CG เคลื่อนขึ้นลงเล็กน้อย) จะช่วยประหยัดพลังงานของแหล่งจ่าย (Supply) ได้ดีมาก ภาพถ่ายจำลองการเคลื่อนไหวแสดงได้ดังรูปที่ 2.12 และการก้าวเดินปกติแสดงได้ดังรูปที่ 2.13



รูปที่ 2.12 แสดงภาพถ่ายจำลองในการเคลื่อนไหวก้าว



รูปที่ 2.13 วัฏจักรการเดินของมนุษย์

(รูปที่ 2.12 และรูปที่ 2.13 อ้างอิงจาก http://www.oandp.com/jpo/library/1997_01_010.asp)

การควบคุมหุ่นยนต์เดินสองขาสามารถอ้างอิงได้จากรูปแบบการก้าวเดินเดินของมนุษย์ โดยอาศัยหลักการ รักษาตำแหน่งของจุดศูนย์กลางมวล (CG หรือ COM) ไม่ให้หลุดออกนอกขอบเขตของฝ่าเท้า รายละเอียดที่จะช่วยเพิ่มข้อมูลในการก้าวเดินอย่างมีประสิทธิภาพของหุ่นยนต์ มีดังต่อไปนี้

2.5 การทรงตัวและการเดินของหุ่นยนต์ (Balancing and Walking of Bipedal Robot)

1. มุมการเคลื่อนที่อิสระ (Orientation Angles) โดยทั่วไปแล้วมุมที่เกิดจากการก้าวขาไป

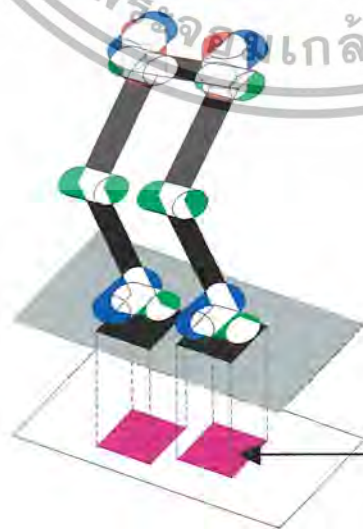
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้างหน้าหรือด้านหลัง (หมุนในแกน Y) จะเรียกว่ามุม Pitch ซึ่งจะทำให้จุดศูนย์กลางมวล (CG) เคลื่อนไปด้านหน้าหรือหลัง และมุมที่เกิดจากการกางขา (หมุนในแกน X) จะเรียกว่ามุม Roll ซึ่งจะ ทำให้จุดศูนย์กลางมวล (CG) เอียงมาด้านข้าง และสุดท้าย มุมที่เกิดจากการเบะเท้า (หมุนในแกน Z) จะเรียกว่า Yaw ซึ่งจะไม่มีผลในการรักษาเสถียรภาพในการเดินตรง แต่จะใช้ในการเลี้ยวมากกว่า ดัง รูปที่ 2.14



รูปที่ 2.14 แสดงมุมการเคลื่อนที่ของแต่ละข้อต่อ

2. พื้นที่รองรับในส่วนเท้า (Support Area) จากรูปแสดงพื้นที่ที่รองรับฝ่าเท้าหรือฐานรองรับเท้าทั้งสองข้างของหุ่นยนต์ การเดินอย่างมีเสถียรภาพนั้น จุด CG จะต้องตกอยู่ภายในฐานรองรับ หรือตกอยู่ระหว่างฐานรองรับเท้าทั้งสอง ดังรูปที่ 2.15



จุด CG จะต้องตกอยู่ภายใต้พื้นที่รองรับส่วนนี้ การเดินจึงจะมีเสถียรภาพ หรือไม่ล้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.15 แสดงพื้นที่ภายใต้การเดินของหุ่นยนต์

3. จุดศูนย์กลางมวล (Center of Mass, COM) หรือจุดศูนย์กลางแรงโน้มถ่วง มีขึ้นเนื่องจากแรงโน้มถ่วงที่กระทำต่อตัวหุ่นยนต์ แรงโน้มถ่วงที่กระทำต่อมวลของหุ่นทั้งหมดจะเท่ากับผลรวมของแรงโน้มถ่วงที่กระทำต่อมวลในแต่ละส่วนของตัวหุ่น โดยที่จุดศูนย์กลางมวลจะเป็นจุดที่แรงดึงดูดทั้งหมด (Total gravity) กระทำต่อตัวหุ่นที่จุดนี้ สามารถแสดงความสัมพันธ์ในรูปสมการคณิตศาสตร์ได้ดังนี้

$$\vec{r}_{COM} = \frac{\sum_i \vec{r}_i m_i}{\sum_i m_i} ; \quad i = 1, 2, \dots, n$$

โดยที่ m_i คือ มวลในแต่ละส่วน (Link) ของหุ่นยนต์

\vec{r}_i คือ เวกเตอร์การกระจัดระหว่างจุดศูนย์กลางมวลในแต่ละส่วนกับจุดศูนย์กลางมวลทั้งหมด

4. จุดศูนย์กลางแรงกระทำ (Centre of Force, COF) เป็นจุดที่แรงกระทำทั้งหมดกระทำที่แต่ละฝ่าเท้า แรงที่เกิดขึ้นบนฝ่าเท้านั้นจะเป็นแรงลัพธ์ เนื่องจากน้ำหนักทั้งหมดของหุ่น สามารถแสดงความสัมพันธ์ในรูปสมการคณิตศาสตร์ได้คล้ายกับจุดศูนย์กลางมวล แต่จุดศูนย์กลางรวมแรงนี้เกิดในแนวระนาบระหว่างฝ่าเท้าและพื้นผิวที่หุ่นกำลังยืนอยู่

$$\vec{r}_{COF} = \frac{\sum_i \vec{r}_i F_i}{\sum_i F_i} ; \quad i = 1, 2, \dots, n$$

โดยที่ F_i คือ แรงกระทำในแต่ละส่วนของฝ่าเท้า

\vec{r}_i คือ เวกเตอร์การกระจัดระหว่างจุดที่แรงกระทำแต่ละส่วนที่ฝ่าเท้ากับจุดศูนย์กลางรวมแรงกระทำทั้งหมดในระนาบฝ่าเท้า

5. จุดที่เกิดจากแรงโน้มถ่วงจากจุดศูนย์กลางมวลตัดผ่าน (Normal Projection of the Center of Mass, NPCM) เป็นจุดบนพื้นที่เกิดจากเส้นแรงโน้มถ่วงจากจุดศูนย์กลางมวลลากลงมาตั้งฉากกับพื้น (Vertically Projected Point of the Center of Gravity, VPCG) สามารถแสดงความสัมพันธ์ในรูปสมการคณิตศาสตร์ได้ดังนี้

$$V_{PCG} = \frac{1}{M} \sum_{i=1}^n m_i V_{PCGi} ; \quad i = 1, 2, \dots, n$$

โดยที่ M คือ มวลหุ่นยนต์ทั้งหมด

m_i คือ มวลในแต่ละส่วน (Link) ของหุ่นยนต์

V_{PCGi} คือ เส้นตรงที่ลากจุดศูนย์กลางมวลในแต่ละส่วนไปตั้งฉากกับพื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. จุดโมเมนต์รวมศูนย์ (Zero Moment Point) เป็นจุดบนพื้นที่เท้าสัมผัสอยู่ ผลรวมของโมเมนต์ที่เกิดขึ้นในขณะนั้นจะเท่ากับศูนย์ หรือในขณะที่หุ่นกำลังยืนอยู่ สามารถเรียกว่า NPCM ได้เหมือนกัน การที่ผลรวมของโมเมนต์เท่ากับศูนย์นั้น หมายความว่า จะไม่มีการหมุนเอียงเนื่องจากแรงบิดในแนวแกนใดๆ จะทำให้การทรงตัวของหุ่นในท่าทางนั้นมีเสถียรภาพหรือไม่ล้มนั่นเอง

2.6 การทรงตัว (Balancing)

ในการก้าวเดินของหุ่นยนต์เดินสองขาโดยทั่วไปนั้น จะถูกอ้างอิงมาจากการก้าวเดินของมนุษย์ ซึ่งกล่าวได้ว่าก่อนการก้าวเดินนั้น จะต้องยืนขึ้นเสียก่อน นั่นคือ การรักษาสถิตนั่นเอง (Balancing)

การทรงตัวที่เกิดขึ้นของหุ่นยนต์นั้น จะถูกควบคุมโดยแรงจากแต่ละส่วนของหุ่นยนต์ ซึ่งจะชดเชยซึ่งกันและกันเพื่อรักษาสถิต ในทางปฏิบัติ เช่น เซอร์วัดแรงกระทำ (Force sensor, Strain Gauge) หรือ เซอร์วัดความเอียง (Tilt Sensor) จะเป็นเครื่องมือในการวัดค่าเพื่อสร้างสัญญาณป้อนกลับ (Feedback) กลับไปยังตัวควบคุมเพื่อใช้ในการรักษาสถิตของหุ่นยนต์

2.7 การก้าวเดิน (Walking)

เมื่อรักษาสถิตได้แล้ว การก้าวเดินของหุ่นยนต์จะเป็นสิ่งตามมา การเดินสามารถแบ่งได้ 3 ระดับที่แตกต่างกัน คือ การก้าวเดินสถิต (Static Walking) การก้าวเดินพลวัต (Dynamic Walking) และการวิ่ง (Running)

1. การก้าวเดินเชิงสถิต (Static Walking) トラバドที่จุดบนพื้นที่เกิดจากเส้นแรงน้ำหนักจากจุดศูนย์กลางมวลที่ลากลงมาตั้งฉากกับพื้น (NPCM) ยังอยู่ในพื้นที่รองรับได้ฝ่าเท้า (Support Area) หุ่นยนต์ยังสามารถทรงตัวอยู่ได้ การก้าวเดินในลักษณะนี้เป็นการก้าวเดินเชิงสถิต ท่าทางการก้าวเดินคล้ายระบบทางกลศาสตร์ ซึ่งการก้าวเดินปกตินี้ลำตัว (CG หรือ COM) จะมีการเคลื่อนไหวไปด้านข้างแบบสลับซ้ายขวา ถ้าจุด NPCM ยังอยู่ภายใต้พื้นที่รองรับ ผลกระทบจากสภาวะพลวัต (Dynamic walking) ที่ทำให้การก้าวเดินไม่สมดุล (เสียการทรงตัว) จะลดลงได้ เช่นการก้าวเดินในท่าที่เท้าทั้งสองสัมผัสพื้น (Double Support Phase) จนลำตัวเริ่มโน้มตัวเอียงไปด้านข้าง (Swing Phase) ไปจนกระทั่งเท้าที่จะก้าว เริ่มแยกจากพื้น ความเร็วจะลดลงช่วงนี้ ดังนั้นอาจเปรียบได้กับการเดินเชิงสถิตนั่นเอง

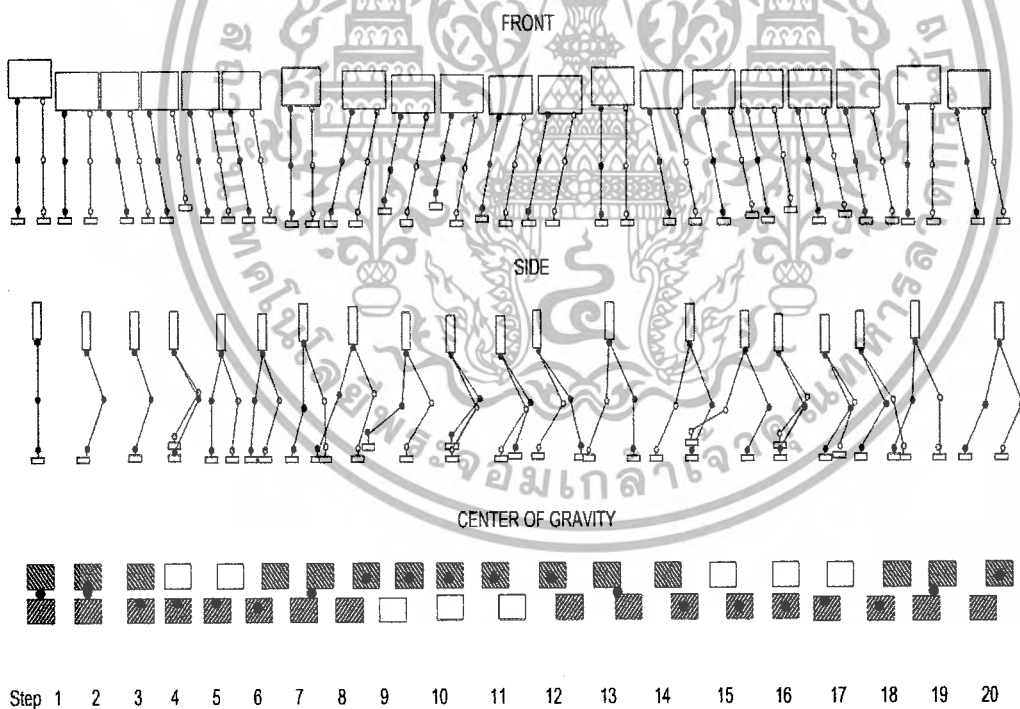
2. การก้าวเดินเชิงพลวัต (Dynamic Walking) ระเบียบวิธีการสำคัญที่ใช้ในการควบคุมการเดินแบบพลวัตก็คือ หลักของจุดรวมโมเมนต์ศูนย์ (Zero Moment Point, ZMP) จุด NPCM นั้นจะหลุดออกนอกฐานยืน เช่น การก้าวขาไปข้างหน้า น้ำหนักเริ่มถ่ายไปเท้าข้างที่ก้าว ถ้าหยุดการเคลื่อนไหวไว้ในลักษณะนี้ จะพบว่าเป็นท่าทางที่ไม่สมดุลในตัวเอง (Unstable condition) สรุปคือ การเดิน

แบบพลวัตนี้ จุดรวมโมเมนต์ศูนย์กลาง จะต้องอยู่ภายในขอบเขตพื้นที่รองรับ มิฉะนั้นการก้าวเดินจะไม่สมดุล ส่งผลให้หุ่นยนต์ล้มได้

3. การวิ่ง (Running) เป็นการก้าวเดินแบบพลวัตที่มีความเร็วสูง มีข้อแตกต่างกันตรงที่จะเกิดการรองรับน้ำหนักทั้งหมดด้วยขาข้างเดียว (Single Support Phase) สลับกันไปมา (Double Support Phase) จะเกิดใช้เวลาน้อย จึงไม่นำมาพิจารณา) และจะเกิดการ โน้มเอียงตัวไปด้านหน้าด้วยขาข้างที่ก้าวไปด้านหน้า

2.8 รูปแบบการเดินของหุ่นยนต์

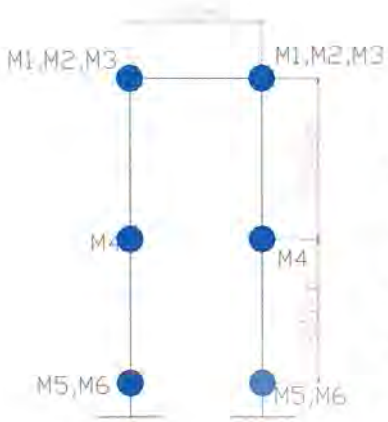
การเดินของหุ่นยนต์สองขาถูกออกแบบให้การเดินแต่ละก้าวมีลักษณะดังรูปที่ โดยจะถูกออกแบบให้จุดศูนย์กลางมวลตกอยู่ภายใต้พื้นที่ขอบเขตของฝ่าเท้า แต่ละก้าวการเดิน การก้าวเดินจะก้าวขาแต่ละข้างสลับกันไปจนกว่าจะหยุดก้าวเดินมาอยู่ในลักษณะยืนตรงสามารถอธิบายได้ดังรูปที่ 2.16 และ โครงสร้างข้อต่อหุ่นยนต์ที่ใช้เคลื่อนที่แสดงได้ดังรูปที่ 2.17



รูปที่ 2.16 ลักษณะการเดินของหุ่นยนต์ที่ถูกออกแบบไว้

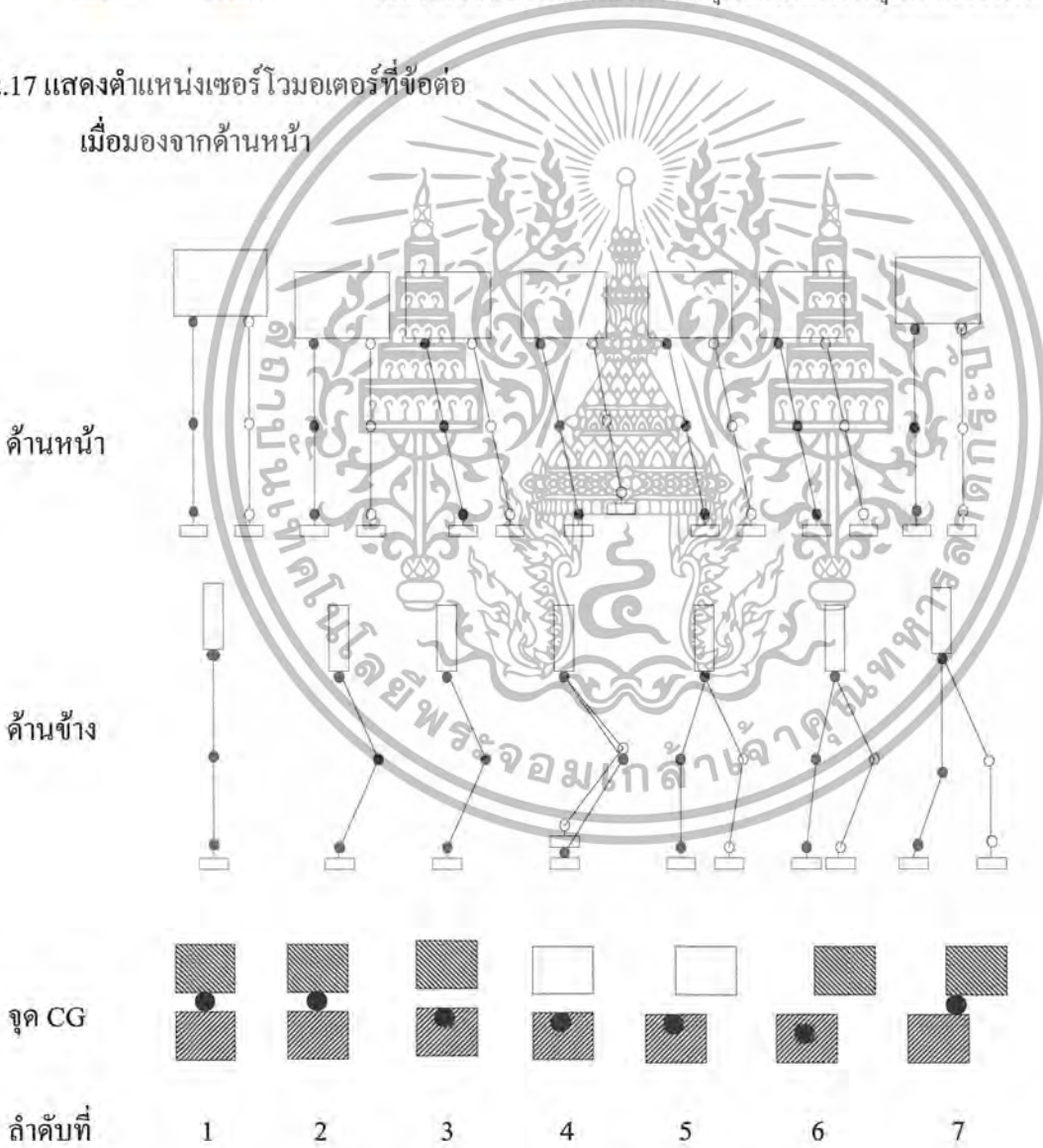
โครงสร้างของข้อต่อที่ใช้ในการขับเคลื่อนหุ่นยนต์แสดงได้ดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- M1 คือ เซอร์โวมอเตอร์ที่ใช้หมุนในทิศการเบะเท่า (Yaw)
- M2 คือ เซอร์โวมอเตอร์ที่ใช้หมุนในทิศการหมุนเอียงข้อเท้า (Roll)
- M3 คือ เซอร์โวมอเตอร์ที่ใช้หมุนในทิศการก้ำเท่า (Pitch)
- M4 คือ เซอร์โวมอเตอร์ที่ใช้หมุนในทิศการงอของเข่า
- M5 คือ เซอร์โวมอเตอร์ที่ใช้หมุนในการงอของข้อเท้า
- M6 คือ เซอร์โวมอเตอร์ที่ใช้หมุนในทิศการหมุนเอียงข้อเท้า

รูปที่ 2.17 แสดงตำแหน่งเซอร์โวมอเตอร์ที่ข้อต่อ
เมื่อมองจากด้านหน้า

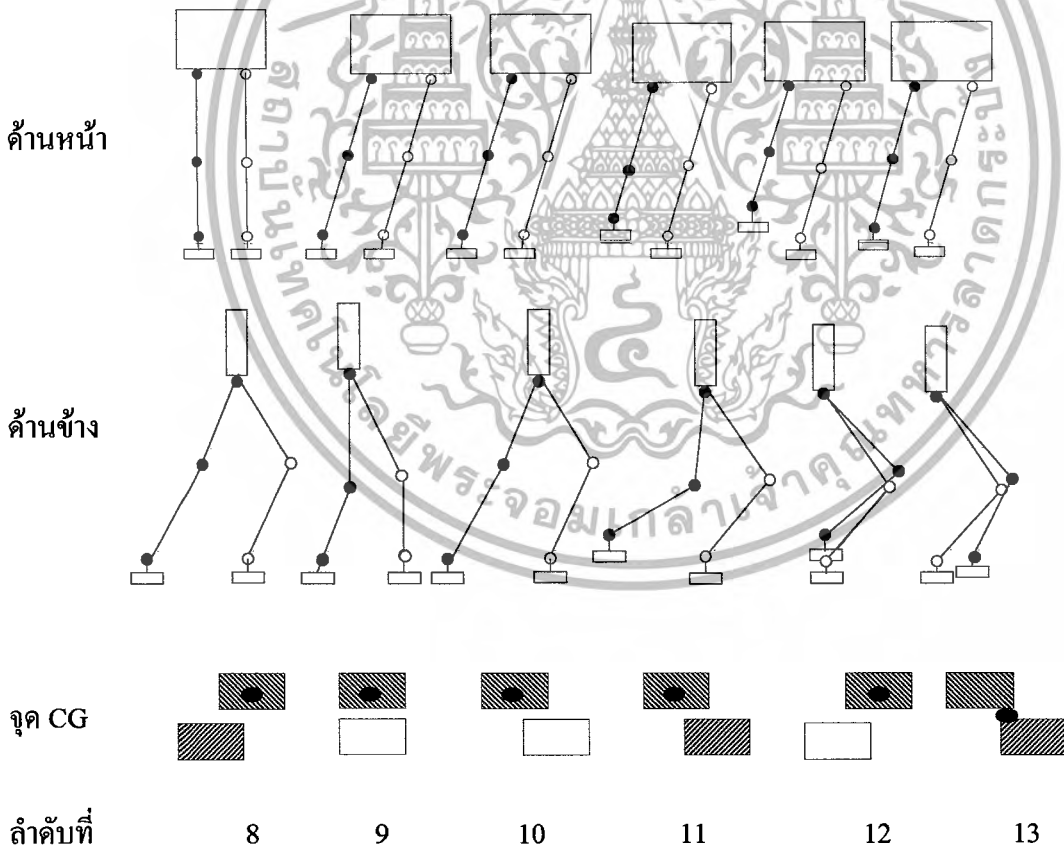


รูปที่ 2.18 แสดงตัวอย่างการก้าวเดินลำดับที่ 1 ถึง 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ละลำดับในรูปที่ 2.18 สามารถอธิบายได้ดังนี้

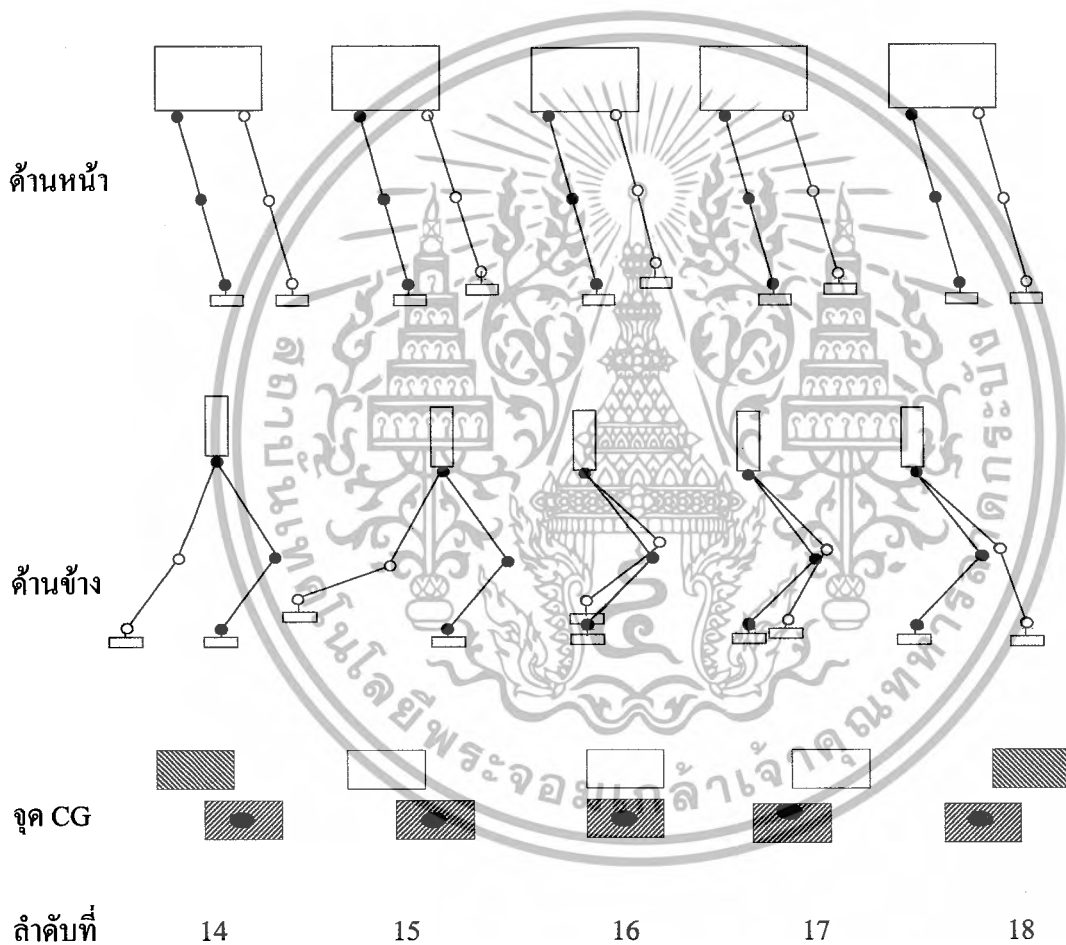
- Step 1:** หุ่นยนต์ ยืนลักษณะตรง Servo motor ทุกตัวถูกตั้งที่ 0 องศา
- Step 2:** หุ่นยนต์ตั้งขาทั้งสองข้าง (เตรียมพร้อมที่จะก้าว) จุด CG อยู่ตรงกลางตัวของหุ่นยนต์
- Step 3:** หุ่นยนต์ เอียงตัวไปด้านขวาในลักษณะที่ขาทั้งสองข้าง จุด CG ตกอยู่ที่เท้าขวา
- Step 4:** ในขณะที่น้ำหนักตกลงที่เท้าขวา หุ่นยนต์ยกขาซ้ายขึ้นมา จุด CG ตกอยู่ที่เท้าขวาข้างเดียว
- Step 5:** ในขณะที่หุ่นยนต์หุ่นยนต์เอียงขวาอยู่ นั้นก็ก้าวขาซ้ายไปข้างหน้าด้วย M4 (เท้าซ้ายยังไม่แตะพื้น)
- Step 6:** ในขณะที่หุ่นยนต์หุ่นยนต์เอียงขวาอยู่ เท้าซ้ายเริ่มแตะพื้น
- Step 7:** หุ่นยนต์โน้มตัวมาขึ้นลักษณะตรง ด้วยการหมุนของ M4, M5 ในขาซ้ายโดยที่จุด CG อยู่ระหว่าง เท้าทั้งสอง



- Step 8:** หุ่นยนต์โน้มตัวมาทางด้านซ้าย นำหนักถ่ายมาทางเท้าซ้าย

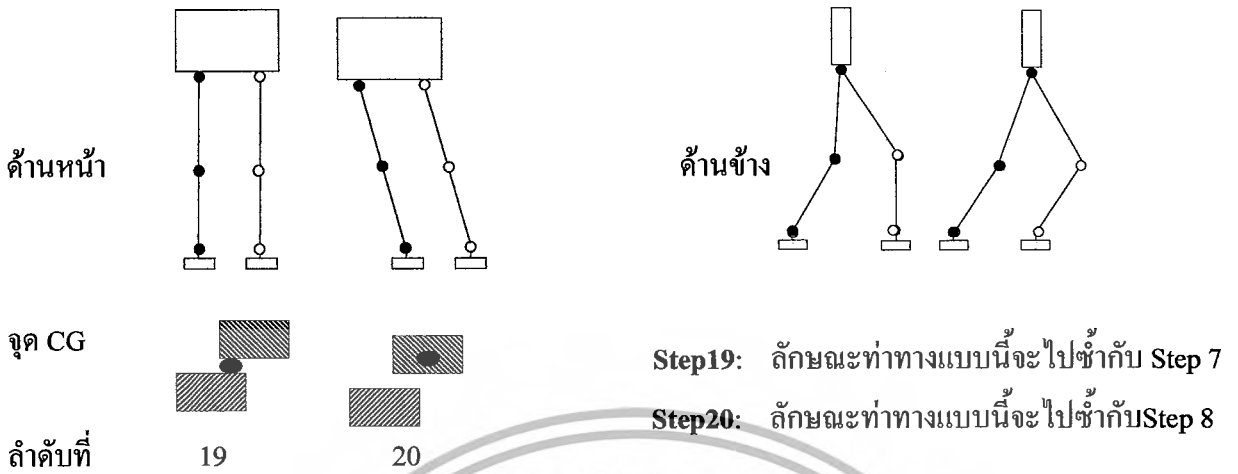
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Step 9:** หุ่นยนต์ยืนด้วยขาซ้ายขาเดียว และขาขวากำลังงอ เข้าขึ้นก่อนที่จะก้าวไปข้างหน้า+
- Step10:** หุ่นยนต์ยืนด้วยขาซ้ายเพียงขาเดียว และก้าวขาขวา ไปข้างหน้าด้วย M3
- Step11:** ในขณะที่หุ่นยนต์ยืนด้วยขาซ้ายเพียงขาเดียว M4 ขา ขวาลงเตะขาไปข้างหน้า ในขณะที่เท้าขวายังไม่แตะพื้น
- Step12:** น้ำหนักของหุ่นยนต์ตกลงที่เท้าซ้าย และเท้าขวาเริ่ม แตะพื้น
- Step13:** หุ่นยนต์โน้มตัวมายืนลักษณะตรง เท้าสองข้าง สัมผัสพื้น (คล้ายกับ Step 7 แต่ขาคนละข้าง) จุด CG อยู่ระหว่างเท้าทั้งสองข้าง



- Step15:** หุ่นยนต์ยืนด้วยขาขวาขาเดียว และขาซ้ายกำลังงอเข้าขึ้น ก่อนที่จะก้าวไปข้างหน้า (คล้ายกับ Step 9 แต่ขาคนละข้าง)
- Step16:** คล้ายกับ Step 10 แต่ขาคนละข้าง หรือ ทำทางลักษณะนี้จะวนไปซ้ำกับ Step 4
- Step17:** ลักษณะท่าทางแบบนี้จะ ไปซ้ำกับ Step 5
- Step18:** ลักษณะท่าทางแบบนี้จะ ไปซ้ำกับ Step 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ทั้งหมดนี้เป็นการก้าวเดินของหุ่นยนต์ซึ่งจะก้าวขาสลับข้างกันไปมาเหมือนการก้าวเดินของมนุษย์ จุดศูนย์กลางมวลจะเคลื่อนที่ไปสู่อีกข้างที่รองรับน้ำหนักเสมอ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

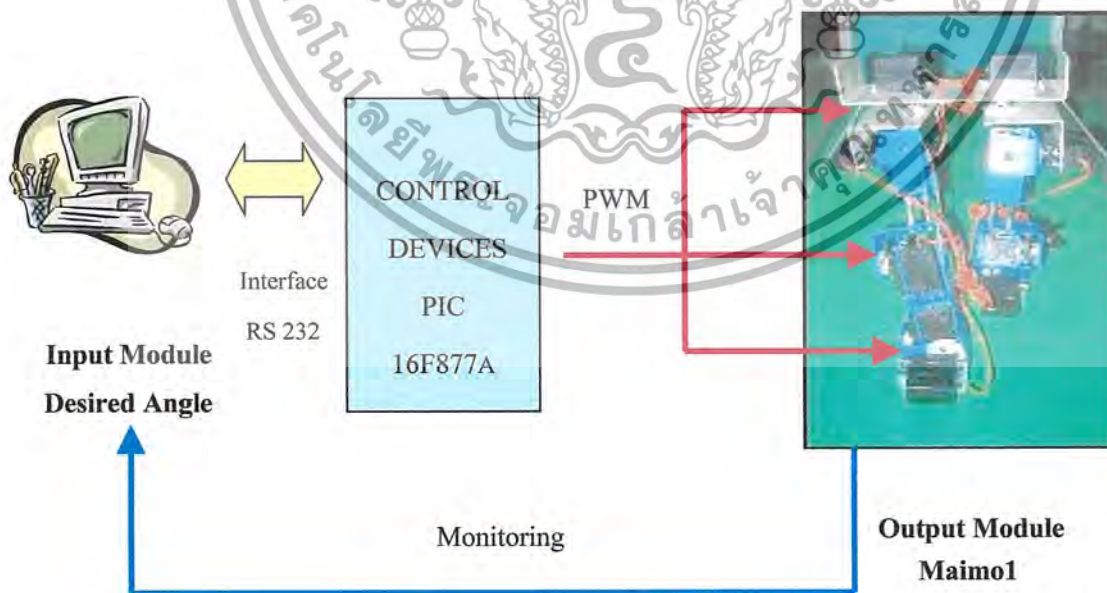
บทที่ 3

การออกแบบโครงสร้างและส่วนวงจรควบคุม

รูปแบบการก้าวเดินของหุ่นยนต์เดินสองขาถูกเชื่อมโยงมาจากรูปแบบการก้าวเดินของมนุษย์ หุ่นยนต์เดินสองขาที่ถูกออกแบบถูกควบคุมด้วยระบบการควบคุมเชิงตำแหน่ง ซึ่งผู้ควบคุมสามารถสังเกตตำแหน่งข้อต่อและควบคุมข้อต่อให้เคลื่อนไหวตามความต้องการได้ รายละเอียดการออกแบบโครงสร้างหุ่นยนต์และส่วนวงจรควบคุมถูกออกแบบตามระบบการทำงาน ซึ่งมีรายละเอียดดังนี้

3.1 ระบบการทำงานของหุ่นยนต์ Maimo1

การออกแบบหุ่นยนต์ Maimo1 ประกอบไปด้วยส่วน โครงสร้างและส่วนวงจรควบคุม ส่วนโครงสร้างถูกออกแบบเฉพาะส่วนท่อนขาทั้งสองข้างโดยอ้างอิงข้อต่อที่ใช้เคลื่อนไหวจากขามมนุษย์ แต่ละข้อต่อขับเคลื่อนโดยเซอร์โวมอเตอร์สำเร็จรูปที่มีขายตามท้องตลาด มีจุดประสงค์เพื่อที่จะศึกษาการก้าวเดินของหุ่นยนต์เทียบจากการก้าวเดินของมนุษย์ โดยการควบคุมเชิงตำแหน่งการก้าวเดินจากการส่งค่ามุมที่ต้องการผ่านเครื่องคอมพิวเตอร์ให้แต่ละข้อต่อเคลื่อนไหวให้เป็นไปตามรูปแบบการก้าวเดินของมนุษย์ สามารถอธิบายได้ดังรูป 3.1



รูปที่ 3.1 แสดงไต่อะแกรมการทำงานของหุ่นยนต์ Maimo1 โดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมหุ่นยนต์ Maimo1จะใช้หลักการควบคุมเชิงตำแหน่ง (Position control) เพราะต้องการให้มอเตอร์หมุนตำแหน่งมุมของขาแต่ละข้างให้เป็นมุมตำแหน่งที่หุ่นยนต์ก้าวเดินได้อย่างสมดุล โดยการสั่งงานให้มอเตอร์หมุนไปในตำแหน่งที่กำหนดและยึดตำแหน่งนั้นไว้จนกว่าจะได้รับคำสั่งควบคุมมาใหม่ ระบบการควบคุมเชิงตำแหน่งสามารถอธิบายได้ดังนี้

3.2 ระบบการควบคุมตำแหน่ง (Position Control)

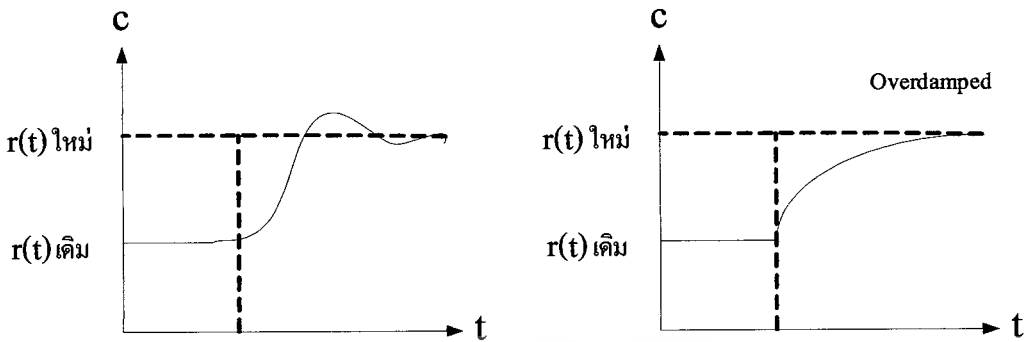
ระบบการควบคุมตำแหน่ง ตำแหน่งเชิงมุมของมอเตอร์ ที่ต่ออยู่กับโหลด จะต้องสอดคล้องกับคำสั่งอินพุต เมื่อสัญญาณควบคุมครั้งที่ มอเตอร์จะล็อคตำแหน่งค้างไว้ เมื่อสัญญาณควบคุมแปรค่าไปอย่างต่อเนื่อง ตำแหน่งของมอเตอร์ก็จะเคลื่อนตามการควบคุม ตรงเท่าที่การแปรค่าไปของสัญญาณควบคุมอยู่ภายใต้แบนด์วิดท์ (Band Width) ของลูป

ตัวเซ็นเซอร์ตำแหน่งอาจจะเป็นพวกที่ให้สัญญาณเอาต์พุตได้ต่อเนื่อง เช่นตัวต้านทานปรับค่าได้ ซึ่งเรียกว่าเป็นแบบอะนาล็อกคอนโทรล (Analog Control) เพราะได้สัญญาณป้อนกลับอย่างต่อเนื่อง และมีความละเอียดในการควบคุมตำแหน่งได้ดีเยี่ยม อย่างไรก็ตามสัญญาณจะถูกจำกัดอยู่ภายใต้ช่วงตำแหน่งช่วงหนึ่งเท่านั้น และการลิมิตนี้จะเป็นตัวจำกัดการเคลื่อนที่

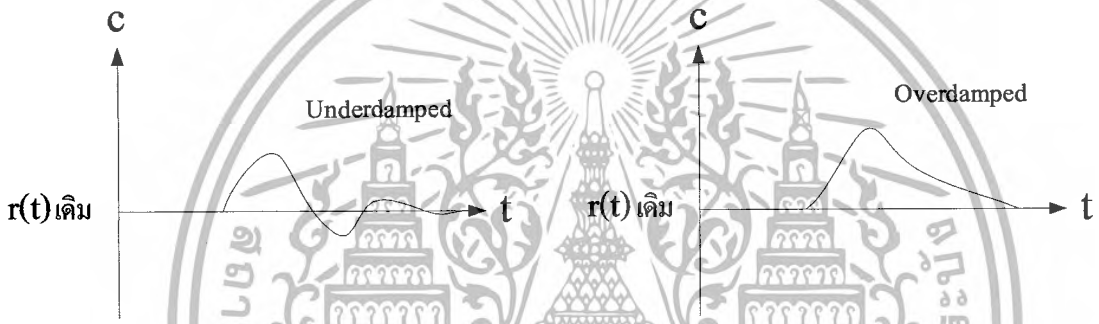
มีหลักความจริงที่ว่า การควบคุมตำแหน่งสุดท้าย (Output) ให้ตรงกับสัญญาณอินพุตนั้นแทบจะเป็นไปไม่ได้เพราะการควบคุมจะต้องมีสัญญาณความคลาดเคลื่อน (Error) ไว้ป้อนกลับเพื่อชดเชยระยะขั้วที่ขาดหรือเกินไป ดังนั้นจุดมุ่งหมายหลักก็คือ การทำให้ระบบมีความเที่ยงตรงที่สถานะคงที่ (Steady state accuracy) ให้เป็นที่น่าพอใจ

ถ้าอัตราขยายที่ต้องการนั้นทำให้ระบบ ไม่มีเสถียรภาพ หรือมีการแกว่ง (Under damping) เมื่อนั้นระบบจะต้องได้รับการชดเชย โดยวิธีทางที่จะให้ผลตอบสนองเที่ยงตรงมากขึ้น การวิเคราะห์ระบบเซอร์โวโดยทั่วไป จะต้องพิจารณาคุณสมบัติการทำงานของระบบอยู่ 2 ประการ คือ

1. คุณสมบัติเชิงพลวัต (Dynamic property) หมายถึง ผลตอบสนองของเอาต์พุตต่อการเปลี่ยนแปลงสัญญาณอินพุต หรือผลตอบสนองต่อการเปลี่ยนแปลง โหลด (Under damped, Critical damped, Over damped) สามารถแสดงได้ดังรูปที่ 3.2 และรูปที่ 3.3



รูปที่ 3.2 แสดงผลตอบสนองของระบบเซอร์โวต่อการเปลี่ยนแปลงอินพุทแบบสแต็ป



รูปที่ 3.3 แสดงผลตอบสนองของระบบเซอร์โวต่อการเปลี่ยนแปลงโพลด์

องค์ประกอบสำคัญที่เป็นกำหนดคุณสมบัติไดนามิกได้แก่

อัตราขยายลูป (Loop gain)

ค่าสูงเกิดผลตอบสนองลดลงแบบแกว่ง (Under damped)

ค่าปานกลางเกิดผลตอบสนองลดลงวิกฤต (Critical damped)

ค่าต่ำเป็นผลให้ผลตอบสนองลดลงอย่างช้าๆ (Over damped)

อัตราขยายของลูปป้อนกลับความเร็ว

ค่าสูงเกิดผลตอบสนองลดลงอย่างช้าๆ (Over damped)

ค่าปานกลางเกิดผลตอบสนองลดลงวิกฤต (Critical damped)

ค่าต่ำเป็นผลให้ผลตอบสนองลดลงแบบแกว่ง (Under damped)

ความเสียดทาน

ค่าสูงเกิดผลตอบสนองลดลงอย่างช้าๆ (Over damped)

ค่าปานกลางเกิดผลตอบสนองลดลงวิกฤต (Critical damped)

ค่าต่ำเป็นผลให้ผลตอบสนองลดลงแบบแกว่ง (Under damped)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการป้อนกลับแรงบิด

จะทำให้ผลตอบสนองเป็นอิสระต่อโพลทอรัค

ระบบควบคุมตำแหน่งหมุน จะประกอบไปด้วยลูปควบคุมความเร็วรอบ และลูปควบคุมแรงบิดรวมกัน โดยที่ถ้าแยกพิจารณา ระบบการควบคุมเป็นกรณีๆ จะสรุปผลได้ดังนี้

1. เมื่อระบบมีลูปควบคุมตำแหน่งเพียงลูปเดียว
 - เสถียรภาพของระบบยัง ไม่ดีพอ
 - ความเร็วของระบบในการตอบสนองยัง ไม่ดีนัก
2. เมื่อระบบมีลูปควบคุมตำแหน่งและลูปควบคุมความเร็วรอบ
 - ความเร็วในการตอบสนองยังคงเดิม
 - เสถียรภาพของระบบดีขึ้น
3. เมื่อระบบมีลูปควบคุมตำแหน่งและลูปควบคุมแรงบิด
 - ความเร็วในการตอบสนองดีขึ้น
 - เสถียรภาพของระบบ ไม่ดีขึ้น
4. เมื่อระบบมีลูปควบคุมตำแหน่ง ลูปควบคุมความเร็วรอบ และลูปควบคุมแรงบิด
 - ความเร็วในการตอบสนองดีขึ้น
 - เสถียรภาพของระบบดีขึ้นอย่างมาก

2. **คุณสมบัติเชิงสถิต (Static property)** หมายถึง ผลตอบสนองของระบบเมื่อเข้าสู่สภาวะคงที่ (Steady state) จะแสดงผลอยู่ในค่าความคลาดเคลื่อน ระหว่างสัญญาณอินพุตและเอาต์พุต เมื่อเวลาเข้าใกล้ค่าอนันต์หมายถึงความเที่ยงตรงของระบบ

องค์ประกอบสำคัญที่เป็นกำหนดคุณสมบัติเชิงสถิต ได้แก่

อัตราขยายลูป (Loop gain)

ค่าสูง ค่าความคลาดเคลื่อนมีค่าน้อย ระบบเที่ยงตรงดี

ค่าปานกลาง ค่าความคลาดเคลื่อนมีค่าปานกลาง ระบบเที่ยงตรง พอใช้ได้

ค่าต่ำ ค่าความคลาดเคลื่อนมีค่าสูง ระบบไม่เที่ยงตรง

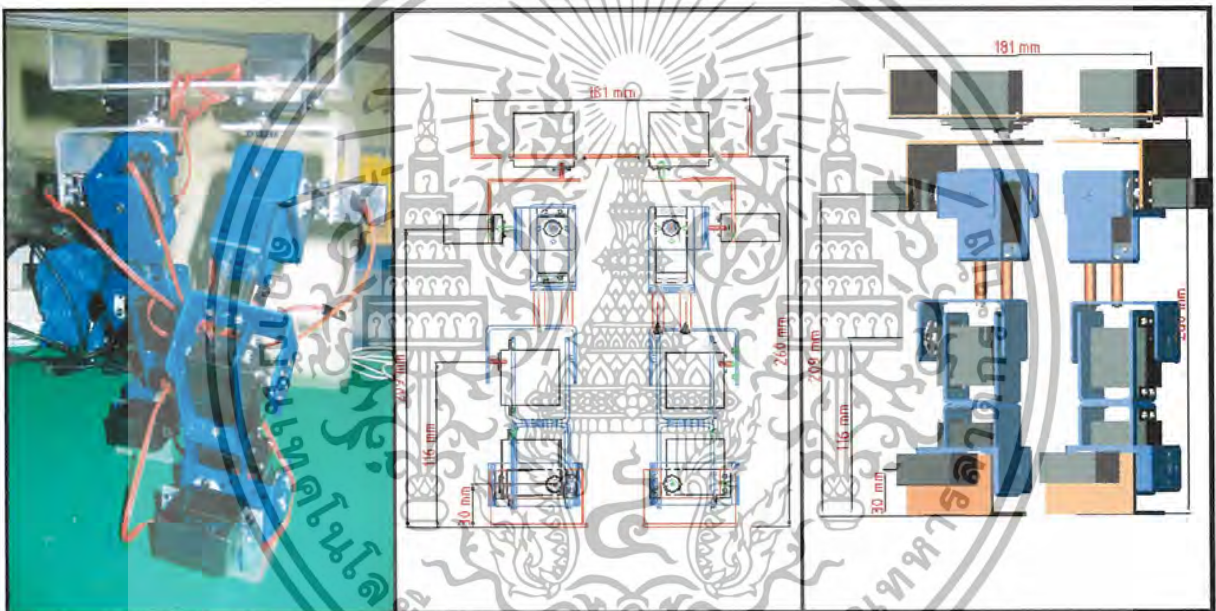
สรุปได้ว่าระบบการควบคุมเชิงตำแหน่งนั้นควรพิจารณาถึงลูปควบคุมเชิงความเร็วและลูปควบคุมเชิงทอรัคด้วยเพื่อผลตอบสนองที่รวดเร็วและมีเสถียรภาพและอัตราขยายลูปจะต้องมีค่าที่เหมาะสมเพียงพอที่จะทำให้ระบบมีค่าความคลาดเคลื่อนระหว่างตัวแปรอินพุตและตัวแปรเอาต์พุตมีค่าน้อยที่สุด หรือระบบมีความเที่ยงตรงนั่นเอง (หมุนได้ตามตำแหน่งที่ต้องการ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบเชิงกล

ในการออกแบบโครงสร้างหุ่นยนต์สองขาให้มีโครงสร้างที่แข็งแรง ยืดหยุ่น ทนทานต่อการใช้งาน นั้นจำเป็นที่จะต้องอาศัยความรู้ทางกลศาสตร์ โดยเบื้องต้นนั้น หุ่นยนต์สองขาจะต้องสามารถขยับเขยื้อนท่อนขาในแต่ละข้อต่อ (Joint) ซึ่งสามารถเอาชนะแรงเฉื่อยที่เกิดขึ้นในการก้าวเดินได้

หุ่นยนต์ Maimo1 แต่ละข้อต่อนั้นถูกออกแบบด้วยเซอร์โวมอเตอร์ (Servo motor) ซึ่งเป็นมอเตอร์ไฟฟ้ากระแสตรงมีคุณสมบัติการควบคุมตำแหน่งที่แม่นยำ แรงบิดสูง น้ำหนักเบา และหาได้ง่ายตามท้องตลาด มาควบคุมให้แต่ละท่อนขามุนตามตำแหน่งในการก้าวเดิน เซอร์โวมอเตอร์แต่ละตัวถูกยึดด้วยโครงอะลูมิเนียมที่มีคุณสมบัติแข็งแรง ทนทาน น้ำหนักเบา และไม่เกิดสนิม ประกอบกันเป็นตัวหุ่นดังรูปที่ 3.4

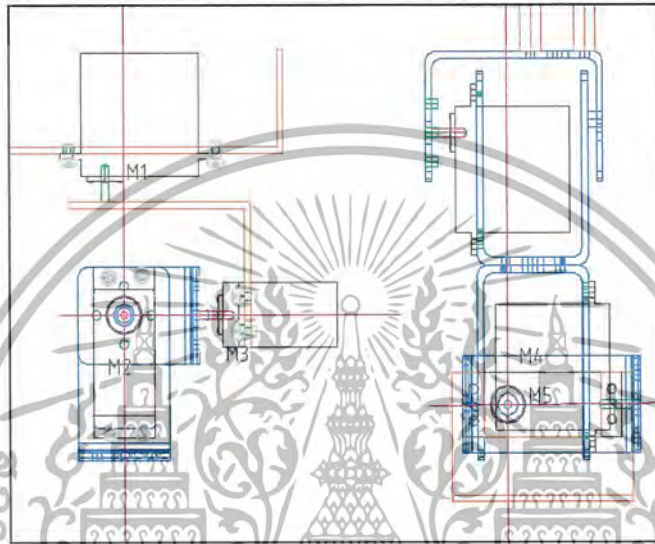


รูปที่ 3.4 แสดงหุ่นยนต์ Maimo1

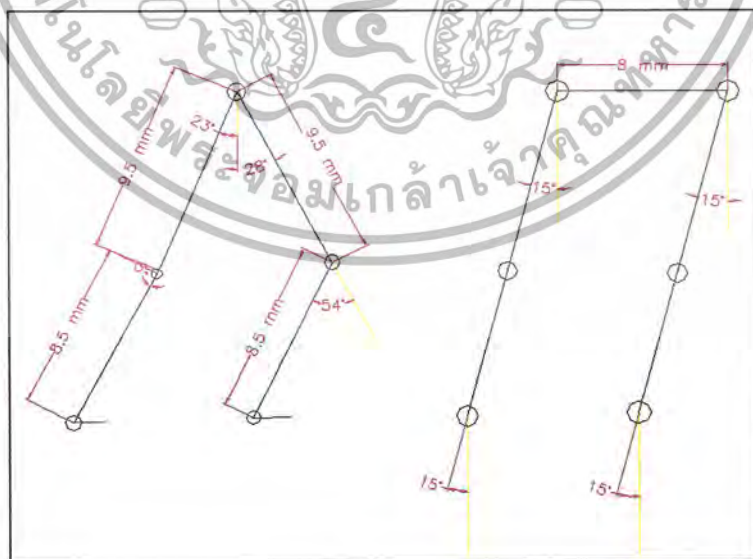
เซอร์โวมอเตอร์ 12 ตัวถูกมาประกอบเป็นข้อต่อทั้งหมด ส่วนท่อนสะโพกแต่ละข้างเซอร์โวมอเตอร์ถูกยึดด้วยโครงอะลูมิเนียม ท่อนสะโพกนี้ขยับได้ในอิสระ 3 แนวแกน (Pitch, Roll, Yaw) เนื่องจากท่อนสะโพกจะต้องใช้แรงบิดในการยกขาก้าวเดินและท่อนข้อเท้าจะต้องใช้แรงบิดที่สูงในการรองรับน้ำหนักทั้งหมดของตัวหุ่นดังนั้นอัตราทดเฟืองจะต้องสูงพอ ดังนั้นวัสดุที่ใช้ทำล้อเฟืองเฟืองภายในเซอร์โวมอเตอร์ที่ท่อนสะโพกและข้อเท้านี้จะเป็น ทองเหลือง เนื่องจากคุณสมบัติทนทานต่อการสึกหรอ ให้แรงบิดคงที่ ไม่เกิดเหตุการณ์เฟืองครูด จากแรงบิดสูงที่เปลี่ยนแปลงอย่างรวดเร็ว ส่วนเซอร์โวมอเตอร์ตำแหน่งอื่น จะใช้พลาสติกเป็นตัวล้อเฟืองเพราะไม่ต้องการแรงบิดมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดหุ่นยนต์ Maimo1 มีความสูงโดยรวมตั้งแต่เท้า จนถึงสะโพกเท่ากับ 26 Cm. และความกว้างของหุ่นเท่ากับ 18 Cm. น้ำหนักทั้งหมดเท่ากับ 2.5 กิโลกรัมประกอบไปด้วยค่าองศาอิสระ (Degree of Freedom, DOF) ทั้งหมดเท่ากับ 12 DOF โดยแบ่งเป็นข้างละ 6 DOF ดังนี้ 3 DOF ณ ตำแหน่งสะโพก 1 DOF ณ ตำแหน่งหัวเข่า และ 2 DOF ณ ตำแหน่งข้อเท้า แสดงได้ดังรูปที่ 3.5 และรูปที่ 3.6



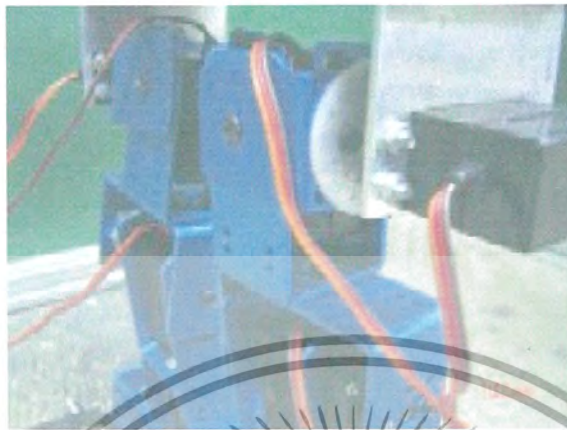
รูปที่ 3.5 แสดงรูปวาดส่วนข้อต่อสะโพก (3 DOF) และข้อต่อเข่า (1 DOF)



รูปที่ 3.6 โดอะแกรมความยาวด้านหน้าส่วนขาของหุ่นยนต์ Maimo1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปข้อต่อของหุ่นยนต์ Maimo1 มีลักษณะตามรูปที่ 3.7 รูปที่ 3.8 และรูปที่ 3.9



รูปที่ 3.7 แสดงข้อต่อส่วนสะโพก (3 DOF)



รูปที่ 3.8 แสดงข้อต่อเข่า (1 DOF)



รูปที่ 3.9 แสดงข้อต่อข้อเท้า (2 DOF)

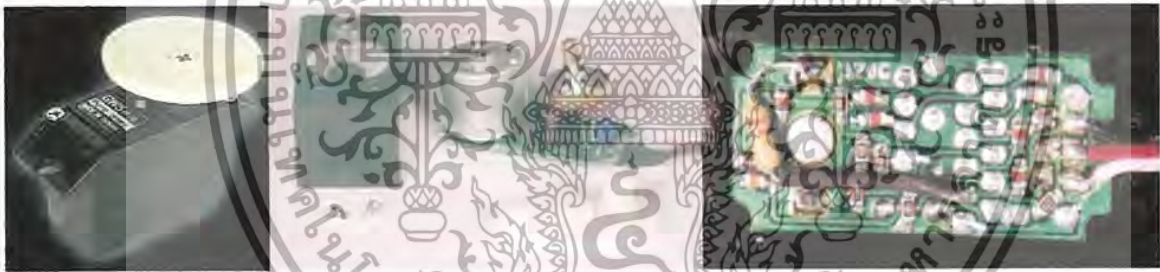
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 รายละเอียดเซอร์โวมอเตอร์ที่เลือกใช้

ลักษณะโครงสร้างทั่วไปเซอร์โวมอเตอร์ ภายในประกอบด้วยชุดเกียร์ (Gear box) และส่วนควบคุมต่างๆดังรูป 3.10 โดยมีสายต่อใช้งานเซอร์โวมอเตอร์เพียง 3 เส้นเท่านั้น คือ สายไฟเลี้ยง (Vcc) สายกราวด์ (GND) และสายสัญญาณควบคุม (Control line) ซึ่งสามารถควบคุมให้มอเตอร์หมุนซ้าย หรือขวาได้จากการสั่งงานของสายสัญญาณควบคุม (Control line) เพียงเส้นเดียว

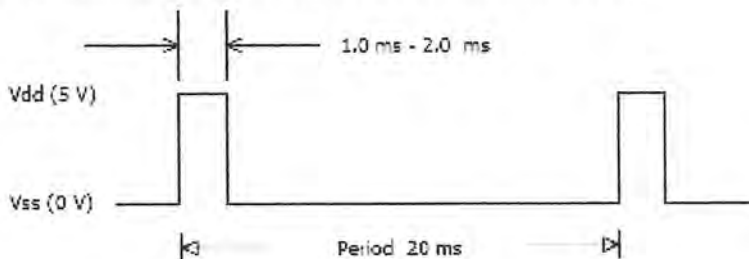
เหตุผลที่เซอร์โวมอเตอร์ถูกเลือกใช้ เป็นเพราะความต้องการออกแบบหุ่นยนต์ให้มีขนาดเล็ก และน้ำหนักเบา ซึ่งเซอร์โวมอเตอร์มีคุณสมบัตินี้ นอกจากนี้เซอร์โวมอเตอร์ยังให้แรงบิดสูง กินพลังงานต่ำและที่สำคัญ สามารถควบคุมทิศทางการหมุนด้วยแรงดันลอจิก ที่เป็น TTL ได้โดยตรงโดยไม่ต้องมีวงจรถับ (Driver circuit) เพราะวงจรถับถูกรวมไว้ภายในตัวแล้ว สัญญาณที่ใช้ควบคุมเป็นสัญญาณ PWM แบบ TTL level และแรงดันไฟเลี้ยงจะอยู่ประมาณ 4 – 6 โวลต์

โครงสร้างภายในเซอร์โวมอเตอร์ ประกอบไปด้วย ตัวต้านทานปรับค่าได้ (Potentiometer) ที่ถูกยึดไว้กับแกนเพลามอเตอร์ เนื่องจาก ภายในตัวต้านทานปรับค่าได้ เป็นแบบหมุนได้ครึ่งรอบ 180° จึงเป็นผลให้การหมุนของเซอร์โวมอเตอร์หมุนได้เพียง 180° เท่านั้น แต่ถ้าต้องการให้เซอร์โวมอเตอร์หมุนได้มากกว่านี้ก็สามารถปรับแต่งตัวเฟืองได้



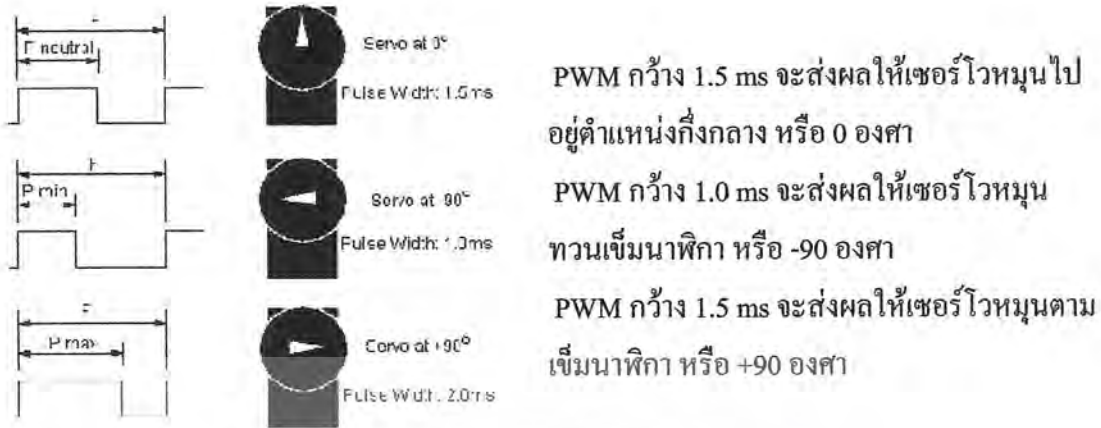
รูปที่ 3.10 แสดงรายละเอียดภายในเซอร์โวมอเตอร์ (อ้างอิงจาก www.ett.co.th)

ส่วนการควบคุมให้มอเตอร์หมุนไปในทิศทางอื่นๆ ก็สามารถป้อนสัญญาณพัลส์ที่ระดับความกว้างต่างๆ โดยอ้างอิงจากจุด 3 จุดที่กล่าวมานี้เช่น ถ้าต้องการให้มอเตอร์หมุนไป $+45^\circ$ เราต้องป้อนสัญญาณพัลส์ที่กว้าง 1.75 ms เป็นต้น และสัญญาณพัลส์จะต้องจ่ายให้กับมอเตอร์ทุกๆ 20 ms (ขึ้นอยู่กับค่าทางบริษัทกำหนดให้) เพื่อรักษาค่าตำแหน่งของมอเตอร์นั้นไว้



รูปที่ 3.11 แสดงสัญญาณคาบเวลา ช่วง Duty cycle “ON” 1.0 - 2.0 ms (อ้างอิงจาก www.ett.co.th)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

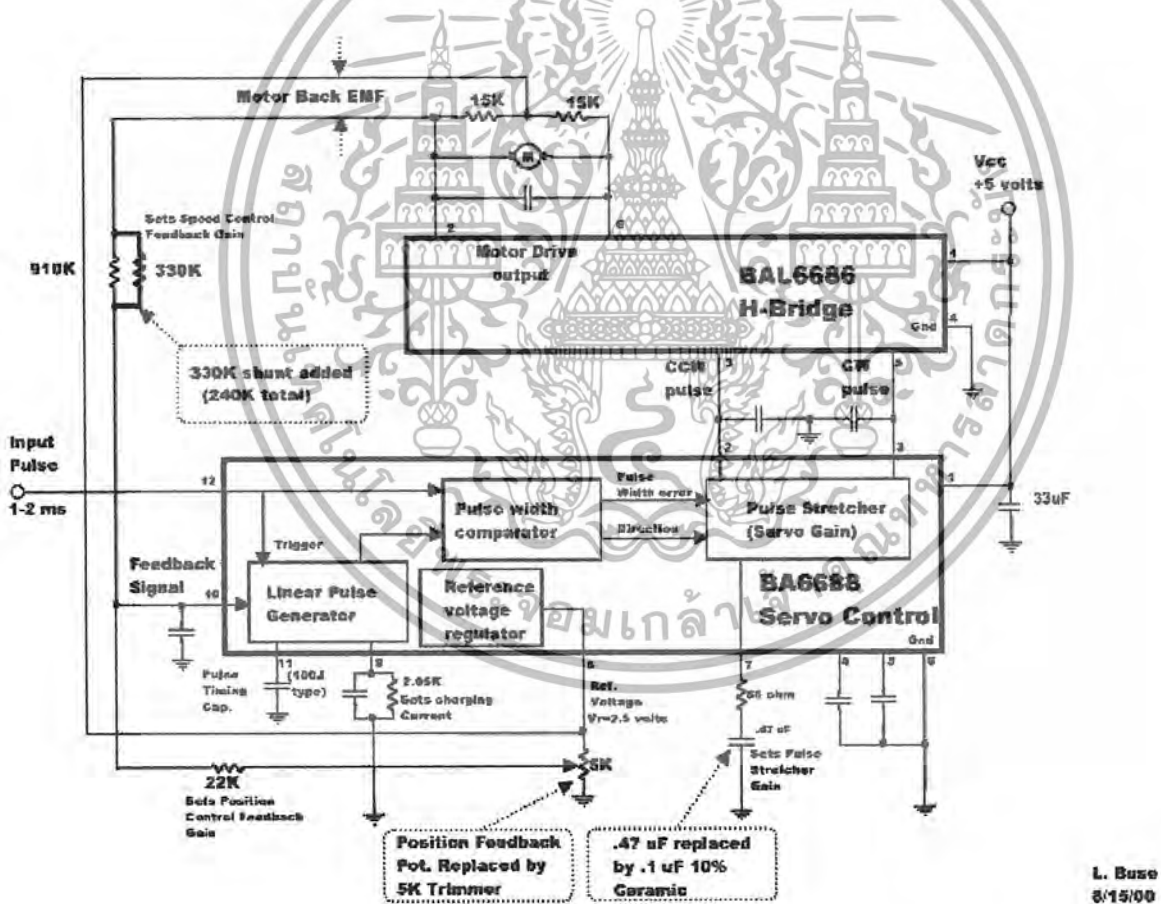


PWM กว้าง 1.5 ms จะส่งผลให้เซอร์โวหมุนไปอยู่ตำแหน่งกึ่งกลาง หรือ 0 องศา

PWM กว้าง 1.0 ms จะส่งผลให้เซอร์โวหมุนทวนเข็มนาฬิกา หรือ -90 องศา

PWM กว้าง 2.0 ms จะส่งผลให้เซอร์โวหมุนตามเข็มนาฬิกา หรือ +90 องศา

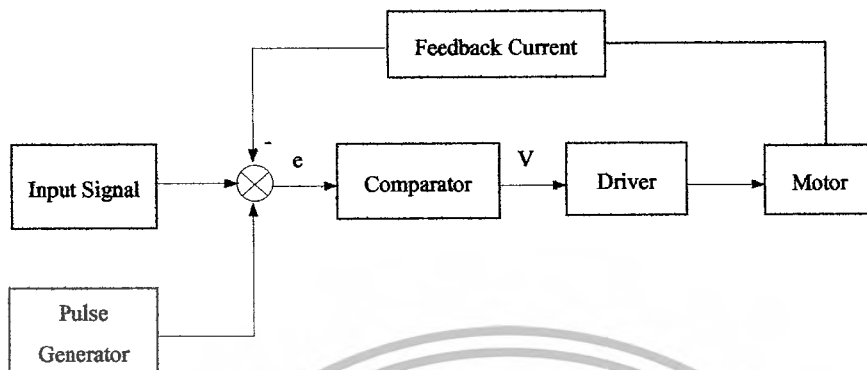
รูปที่ 3.12 แสดงการหมุนของมอเตอร์ที่คาบเวลาต่างๆ (อ้างอิงจาก www.ett.co.th)



รูปที่ 3.13 แสดงบล็อกไดอะแกรม ภายในวงจรขับเซอร์โวมอเตอร์ (อ้างอิงจาก www.ett.co.th)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

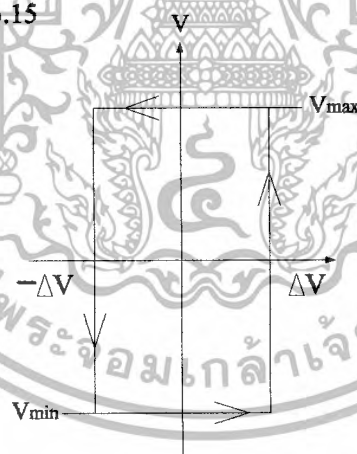
จากรูปที่ 3.13 สามารถเขียนบล็อกไดอะแกรม ของวงจรขับ PWM Amplifier ใหม่ได้ดังนี้ แสดงได้ดังรูปที่ 3.14



รูปที่ 3.14 แสดงบล็อกไดอะแกรมของวงจรภายในเซอร์โวมอเตอร์

แต่ละส่วนของบล็อกไดอะแกรมสามารถอธิบายได้ดังนี้

1. วงจรเปรียบเทียบแรงดัน (Comparator) เป็นวงจรขยายที่มีอัตราขยายสูง และมีการป้อนกลับแบบบวก ซึ่งมีผลในลักษณะฮิสเทอรีซิส (Hysteresis) ดังนั้น เอาท์พุทแรงดันจะมีทั้งด้านบน (V_{max}) และลบ (V_{min}) ดังรูปที่ 3.15



รูปที่ 3.15 แสดงผลของฮิสเทอรีซิส

2. วงจรขับ (Driver) เป็นวงจรขยายที่ให้แรงดันเอาท์พุทเท่ากับแหล่งจ่าย (V_{dc}) วงจรขับนี้จะมีค่าเป็นบวกหรือลบนั้นขึ้นอยู่กับค่าแรงดันที่วงจรเปรียบเทียบแรงดัน และจะมีอัตราขยายคงที่
3. การป้อนกลับของกระแส (Feedback Signal) วงจรป้อนกลับแบบ H-Bridge กระแสจะไหลออกจากปลายขา H ทั้งสองขา ความแตกต่างกันของกระแสจะถูกป้อนกลับ
4. ตัวสร้างสัญญาณพัลส์ (Frequency Generator) จะมีไว้เพื่อกำหนดความถี่จากสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้อนกลับ เพื่อ ไปเปรียบเทียบกับสัญญาณอินพุทที่วงจรเปรียบเทียบแรงดัน

จากรายละเอียดข้างต้นของเซอร์โวมอเตอร์ สามารถสรุปได้ว่าสัญญาณเอาต์พุท (ตำแหน่งการหมุนสุดท้าย) จะเป็นไปตามค่าสัญญาณอินพุทที่กำหนด ค่าความคลาดเคลื่อน (Error) ที่ขาดเหลืออยู่ นั้นจะต้องถูกชดเชยให้เหลือน้อยที่สุด แต่ในทางความเป็นจริงแล้ว เซอร์โวมอเตอร์ที่เลือกใช้ไม่สามารถที่จะทำให้สัญญาณความคลาดเคลื่อนเท่ากับศูนย์ได้ กำลังงานได้สูญเสียในรูปแบบต่างๆ เช่น การสูญเสียทางไฟฟ้าเช่นความร้อนที่เกิดขึ้นที่ขดลวดอาเมเจอร์ และ การสูญเสียเชิงกลเช่น ความเสียดทานของล้อเฟือง เป็นต้น อย่างไรก็ตามก็ถือได้ว่า ค่าสัญญาณความคลาดเคลื่อน (Error) นั้นมีค่าน้อย ถ้าเลือกใช้เซอร์โวมอเตอร์จากผู้ผลิตที่มีคุณภาพ

3.5 ส่วนวงจรควบคุม

หุ่นยนต์ Maimo1 จะขับเคลื่อนเซอร์โวมอเตอร์แต่ละตัว ให้เป็นไปตามรูปแบบการก้าวเดินของมนุษย์ โดยสัญญาณควบคุมที่ใช้คือสัญญาณความกว้างพัลส์ (Pulse Width Modulation, PWM) เซอร์โวมอเตอร์จะหมุนตามความถี่ที่เปลี่ยนแปลงไป หรือค่าความถี่ไซเคิลที่เปลี่ยนแปลง สัญญาณความกว้างพัลส์ (PWM) ถูกสร้างโดยไมโครคอนโทรลเลอร์ PIC ตามค่าข้อมูลที่ถูกส่งผ่านคอมพิวเตอร์โดยโปรแกรมวิซวลเบสิก (Visual Basic) ผู้ควบคุมสามารถส่งคำถามที่ต้องการผ่านหน้าจอคอมพิวเตอร์ให้เซอร์โวมอเตอร์หมุนได้ตามต้องการ จากนั้นค่ามุมจากเซนเซอร์วัดความเอียง (Tilt Sensor) จะถูกส่งผ่านมายังหน้าจอคอมพิวเตอร์เพื่อบ่งบอกถึงค่าความเอียงของตัวหุ่นยนต์ รายละเอียดของส่วนวงจรควบคุมมีดังนี้

3.5.1 การติดต่อกับเครื่องคอมพิวเตอร์

โปรแกรม Visual Basic ซึ่งเป็นโปรแกรมที่สามารถติดต่อกับผู้ใช้ได้ค่อนข้างง่าย และรวดเร็ว การเชื่อมต่อนั้นกระทำผ่านพอร์ตสื่อสารอนุกรม (Serial Port) ซึ่งเป็นพอร์ตติดต่อกับคอมพิวเตอร์ที่มีใช้อยู่ทั่วไป รายละเอียดมีดังต่อไปนี้

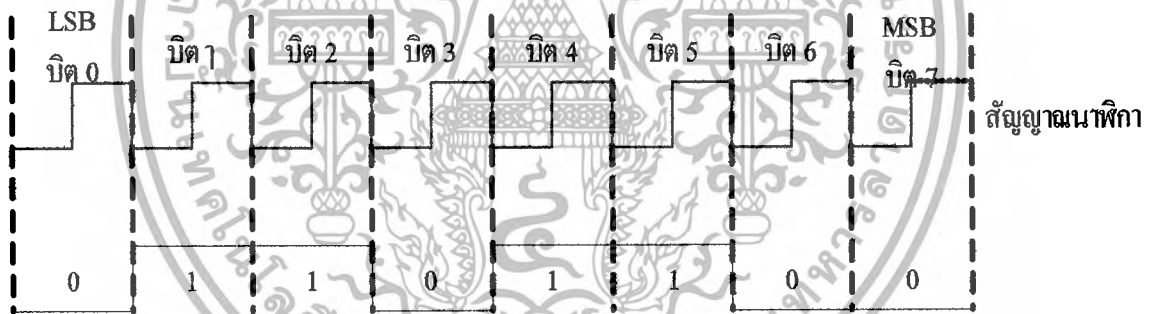
การสื่อสารแบบอนุกรม (Serial Interface) เป็นการรับส่งข้อมูลระหว่างคอมพิวเตอร์และอุปกรณ์ต่อพ่วงอื่น ครั้งละหลายบิต โดยตัวรับของคอมพิวเตอร์จะต้องรอข้อมูลมาครบทุกบิตก่อนจึงทำการประมวลผล ดังนั้นจึงส่งผลให้การสื่อสารแบบอนุกรมอาจมีความเร็วช้ากว่าการติดต่อสื่อสารแบบขนาน แต่ในด้านสายสัญญาณรับส่งข้อมูล การสื่อสารแบบอนุกรมจะใช้จำนวนสายน้อยกว่ามาก โดยใช้อย่างน้อยสุดเพียง 3 เส้นเท่านั้น (สายรับสัญญาณ (RxD) สายส่งสัญญาณ (TxD) และสายกราวด์ (GND)) และสามารถใส่ สายสัญญาณได้ยาวกว่า ปัญหาในเรื่องระยะทางของการสื่อสารแบบอนุกรมจึงมีน้อยกว่าการสื่อสารแบบขนานอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการสื่อสารอนุกรม

1. ระยะทางของสายอนุกรมมีความยาวได้มากกว่าสายแบบขนานมาก ทั้งนี้เพราะสัญญาณของพอร์ตอนุกรม ซึ่งส่วนใหญ่ใช้มาตรฐานแบบ RS-232 จะมีค่าแรงดันไฟฟ้าประมาณ -3 Volt ถึง -15 Volt สำหรับ Logic 1 (Mark) และ +3 Volt ถึง +15 Volt สำหรับ Logic 0 (Space) ส่วน +3 Volt ถึง +3 Volt ไม่ใช้งาน (Undefined)
2. สายอนุกรมใช้สายสัญญาณน้อยกว่าแบบขนานถ้าต่อในลักษณะ Null Modem จะใช้เพียง 3 เส้นเท่านั้น ส่วนแบบขนานใช้ 9 ถึง 25 เส้น
3. ปัจจุบันนี้ไมโครคอนโทรลเลอร์ มักจะมีการผนวกพอร์ตสื่อสารอนุกรมไว้ด้วย เพราะใช้จำนวนน้อยกว่าแบบขนาน และได้ปรับปรุงถึง RS-232E ที่กำหนดค่าอัตราการรับส่งข้อมูล (Baud rate) เกิน 20K Baud

การสื่อสารแบบอนุกรมแบ่งออกเป็น 2 แบบ คือ การสื่อสารแบบซิงโครนัส และการสื่อสารแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาที่รวมอยู่กับสายอยู่กับการรับส่งข้อมูลด้วย สายสัญญาณอย่างน้อยที่ใช้ประกอบไปด้วย สายสัญญาณนาฬิกา (Clock line) สายข้อมูล (Data line) และสายกราวด์ (GND) ดังรูปที่ 3.16



รูปที่ 3.16 แสดงการสื่อสารอนุกรมซิงโครนัสอย่างง่าย

การสื่อสารข้อมูลแบบอะซิงโครนัส

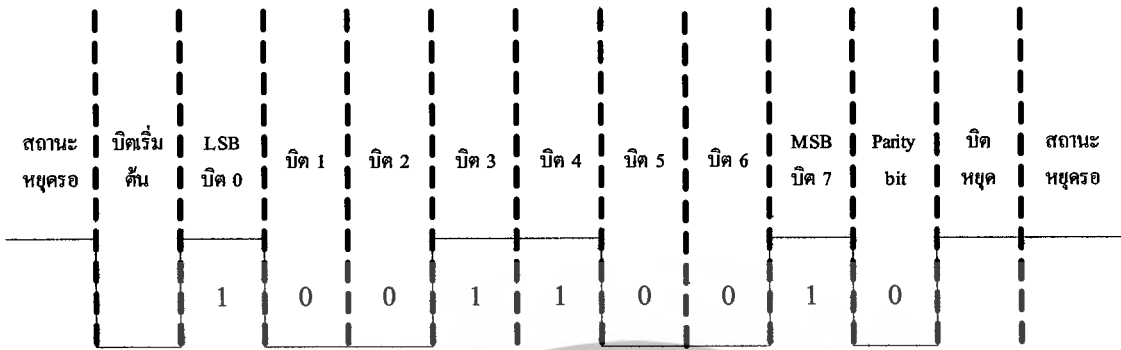
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับส่งข้อมูลไปในสายโดยไม่จำเป็นจะต้องสายสัญญาณนาฬิกาที่รวมเหมือนการรับส่งแบบซิงโครนัสแต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน โดยเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดการรับส่งข้อมูลว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baudrate) มีหน่วยเป็นบิตต่อวินาที (bit per second: bps)

รูปแบบข้อมูลที่ใช้ในการรับส่งข้อมูลแบบอะซิงโครนัสประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิตและบิตข้อมูลอนุกรมจะมีค่าขนาด 5 ถึง 8 บิต
2. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. บิตปิดท้าย (Stop) จะมีขนาด 1,1.5 หรือ 2 บิต



รูป 3.17 แสดงการสื่อสารอนุกรมอะซิงโครนัสอย่างง่าย

รูปที่ 3.17 แสดงการสื่อสารอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลมาหา Data จะมีสถานะลอจิก 1 ซึ่งเรียกสถานะนี้ว่าสถานะหยุด การเริ่มต้นส่งข้อมูลจะเริ่มต้นจากให้หา Data มีสถานะลอจิก 0 ในช่วงเวลานี้เรียกว่า “บิตเริ่มต้น” จากนั้นข้อมูลจะถูกส่งออกไป โดยเริ่มต้นจากบิตที่น้อยสำคัญต่ำสุด (LSB) ก่อนซึ่งข้อมูลอาจจะเป็น 5, 6, 7 หรือ 8 บิตก็ได้จากนั้นตามด้วยบิตพริตี้ (Parity bit) ซึ่งใช้ในการตรวจสอบข้อผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายคือ บิตปิดท้าย ซึ่งจะให้หา Data มีสถานะลอจิกเป็น 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1, 1.5 หรือ 2 บิตเพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้การออกแบบมาสำหรับรับส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับส่งข้อมูล (Baud rate) ตามมาตรฐาน พอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่ามากขึ้นตามเทคโนโลยีคอมพิวเตอร์ ซึ่งถ้าไม่ผ่านโมเด็มอาจจะกำหนดค่าบอดเรตได้ถึง 155200 บิตต่อวินาที

การตรวจสอบพริตี้ สามารถกำหนดได้เป็นแบบคี่ (Odd) หรือแบบคู่ (Even) หรือไม่มีการตรวจสอบพริตี้ก็ได้ การตรวจสอบพริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก 1 ภายในข้อมูลที่ส่งไปหนึ่งไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยการรวมบิตพริตี้เข้าไปด้วย เช่น ข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง ข้อมูลในไบต์นี้มีลอจิก 1 จำนวน 4 ตัวซึ่งเป็นจำนวนคู่ ดังนั้นถ้ากำหนดบิตพริตี้เป็นคู่ ค่าในบิตพริตี้จะต้องมีลอจิกเป็น 0 แต่ถ้าพริตี้เป็นคี่ ค่าพริตี้บิตจะมีค่าเป็น 1 เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพริตี้รวมกันเป็นเลขคี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

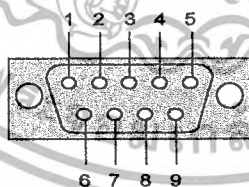
มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ถูกออกแบบมาเพื่อใช้ในการรับส่งข้อมูลอะซิงโครนัสสองทิศทางระหว่างอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment, DTE) กับวงจรรับข้อมูลปลายทาง (Data Circuit Terminating, DCE) โดยที่อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ถูกส่งมาจากอุปกรณ์ DTE เท่านั้น

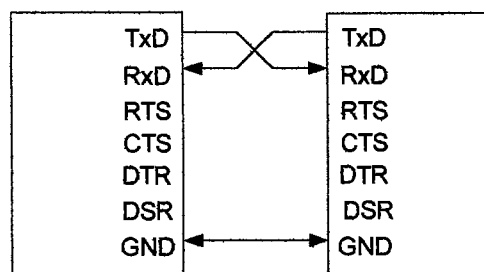
ข้อแตกต่างระหว่างอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เราเห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE (พอร์ตอนุกรมของคอมพิวเตอร์) จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของอุปกรณ์ DCE (อุปกรณ์ปลายทางเช่น โมเด็ม) จะเป็นแบบตัวเมีย สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกเชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ด้วยความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

การเชื่อมต่อตัวคอนเน็กเตอร์สำหรับพอร์ต RS-232

มาตรฐานการเชื่อมต่อแบบ RS-232 ที่ใช้ในโรงงานนี้จะใช้คอนเน็กเตอร์แบบ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-9 จะมีขาต่อใช้งานเพียง 9 เส้น ส่วนพอร์ตอื่นๆถูกใช้งานไม่มากนักจึงถูกยกเลิกไป การจัดรูปแบบขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9 แสดงได้ดังรูปที่ 3.18 และการเชื่อมต่อในโรงงานนี้กำหนดให้ตัวรับคือ ไมโครคอนโทรลเลอร์ PIC ส่วนตัวส่งคือคอมพิวเตอร์ การเชื่อมต่อแสดงได้ดังรูปที่ 3.19



รูปที่ 3.18 แสดงการจัดวางตำแหน่งขาของคอนเน็กเตอร์แบบ DB-9



รูปที่ 3.19 การเชื่อมต่อตัวรับตัวส่งแบบทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดและหน้าที่การทำงานของขาพอร์ตอนุกรมที่เลือกใช้ RS-232 มีดังนี้

1. **ขาที่ 2 (Receive Data, RxD)** ขานี้มีหน้าที่รับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูล เก็บไว้ในรีจิสเตอร์บัฟเฟอร์
2. **ขาที่ 3 (Transmitted Data, TxD)** ขานี้มีหน้าที่ส่งข้อมูลที่เก็บในบัฟเฟอร์สำหรับส่งข้อมูลออกไปยังตัวรับ
3. **ขาที่ 5 (Ground, GND)** ขากราวด์

การควบคุมสัญญาณ (Flow Control)

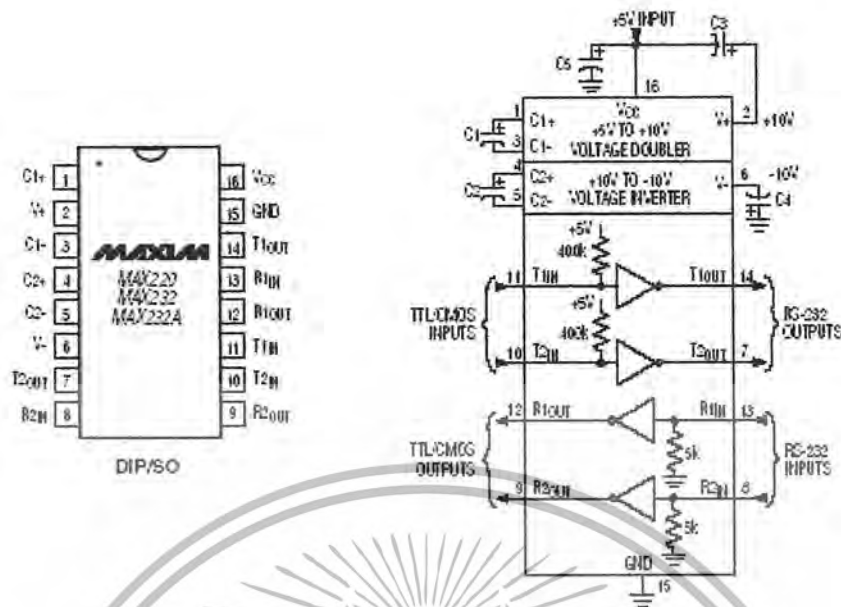
การติดต่อระหว่าง DTE และ DCE จะมีการควบคุมการไหลของข้อมูลเพื่อไม่ให้เกิด Overflow ขึ้นซึ่งมีสองรูปแบบคือ Hard Flow Control และ Software Flow Control

สำหรับ Software Flow Control มักจะเรียกว่า Xon/Xoff flow control ซึ่งใช้รหัส ASCII 17 เป็นสัญญาณ Xon และใช้รหัส ASCII 19 เป็นสัญญาณ Xoff จะมีหลักการการทำงาน คือ Modem มี Buffer ซึ่งรับข้อมูลจากคอมพิวเตอร์อยู่ ถ้าข้อมูลใกล้เต็มเมื่อไร สัญญาณ Xoff จะถูกส่งออกไปเพื่อให้คอมพิวเตอร์หยุดส่งข้อมูลชั่วคราว หลังจากนั้นถ้า Buffer ว่างเมื่อไร สัญญาณ Xon ก็จะถูกส่งให้แก่คอมพิวเตอร์ เพื่อให้คอมพิวเตอร์ส่งข้อมูลต่อ การควบคุมวิธีนี้ ประหยัดสายสัญญาณ เพราะข้อมูลถูกส่งผ่าน TxD และ RxD แต่อาจทำให้เกิดความล่าช้าในกรณีที่การสื่อสารมีความเร็วต่ำ เพราะแต่ละตัวอักษร (ASCII Code) มีขนาด 10 บิต

ตัวแปลงสัญญาณ RS-232

สัญญาณลอจิก 0 ของ RS-232 จะมีค่า +3 ถึง +25 Volt และสัญญาณลอจิก 1 ของ RS-232 มีค่า -3 ถึง -25 Volt ช่วง -3 ถึง +3 Volt จะไม่สามารถใช้งานได้ ระดับสัญญาณนี้ใช้ได้กับทุกสัญญาณที่ไม่ใช่เฉพาะสัญญาณรับส่งเท่านั้นแต่ยังรวมถึงสัญญาณควบคุมต่างๆเช่น DTR, RTS, CTS เป็นต้น

IC ที่ใช้ในตัวรับตัวส่งมักเป็นเบอร์ 1488 (RS-232 Driver) และ 1489 (RS-232 Receiver) โดยแต่ละตัวประกอบด้วย Inverter 4 ตัวและต้องการไฟเลี้ยง 2 ชุด คือ +7.5 ถึง +15 Volt และ -7.5 ถึง -15 Volt ซึ่งอาจจะมีปัญหาเกิดขึ้นได้ ถ้าเครื่องที่มีไฟเลี้ยง +5 Volt เพียงชุดเดียว แต่ก็ยังมี IC อีกตัวหนึ่งคือ IC ตระกูล Max232 ซึ่งมีวงจรเพิ่มแรงดันไฟฟ้าให้สูงขึ้นจาก +5 Volt ได้เป็น +10 Volt และ -10 Volt พร้อมทั้งมีขา TxD และ RxD สองชุดอยู่ในชุด IC เดียวกัน และรองรับ Baud rate ได้ถึง 120 Kbps จึงสะดวกมากกว่าเพราะใช้ IC เพียงตัวเดียว



รูปที่ 3.20 แสดง โครงสร้างภายในและตำแหน่งขาของ Max 232

(อ้างอิงจาก www.maxim-ic.com)

ส่วนการที่เราจะนำข้อมูลใช้ก็ต้องแปลงเป็นรูปแบบขานก่อน ซึ่งเป็นหน้าที่หลักของ Universal Asynchronous Receiver Transmitter (UART) ซึ่งจะทำหน้าที่แปลงข้อมูลที่อยู่ในรูปอนุกรมแบบอะซิงโครนัส เป็นข้อมูลขานเข้าสู่อุปกรณ์คอมพิวเตอร์ หรือแปลงข้อมูลที่อยู่ในรูปขานเป็นข้อมูลอนุกรมแบบอะซิงโครนัส ไปยังอุปกรณ์ปลายทาง นอกจากนั้นแล้ว UART ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น ค่าอัตราเร็วการรับส่งข้อมูล (Baud Rate) รูปแบบการส่งข้อมูล ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายถอดข้อมูล (เช่น พาริตีเฟรมข้อมูล โอเวอร์รัน เป็นต้น)

3.5.2 ไมโครคอนโทรลเลอร์ PIC

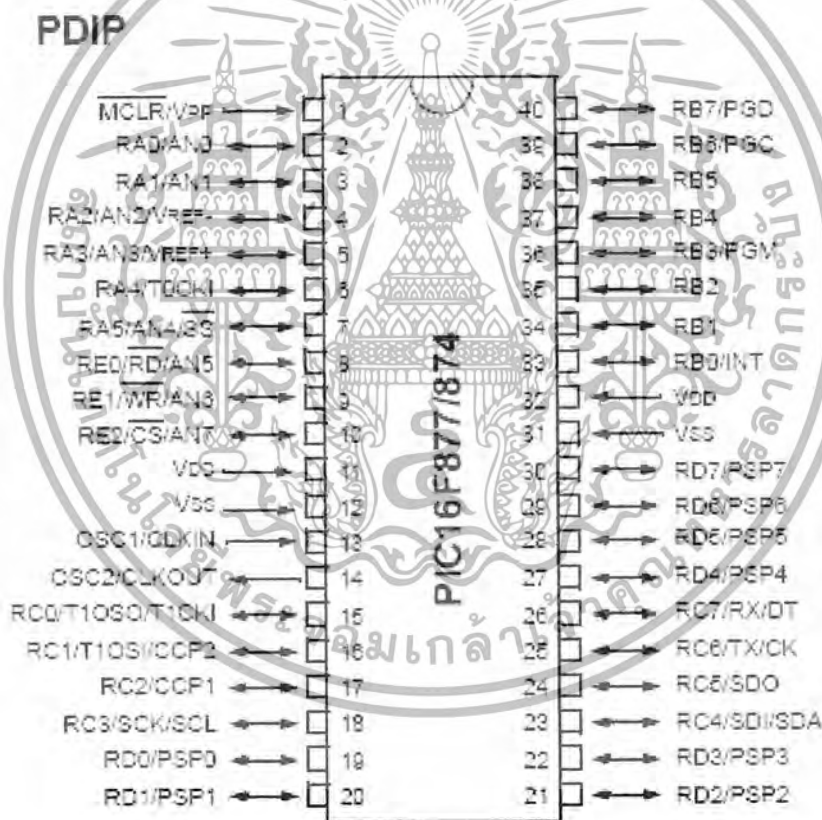
ไมโครคอนโทรลเลอร์ PIC 16F877A ถูกนำมาใช้ในการควบคุมเซอร์โวมอเตอร์ทั้งหมดและ PIC 16F877 ถูกนำมาใช้ในการส่งค่าจากเซนเซอร์วัดความเอียงซึ่งในหุ่น Maimo1ซึ่ง PIC ทั้งสองเบอร์มีตำแหน่งขาที่เหมือนกันดังรูปที่ 3.17 วงจรควบคุมของ PIC 16F877A นี้จะทำการสร้างสัญญาณ PWM ออกมาตามมาตรฐานที่ใช้ในการควบคุมเซอร์โวมอเตอร์ โดยการป้อนค่าของมุมของข้อต่อต่างๆผ่านเครื่องคอมพิวเตอร์ทาง RS-232 ในอัตรา 115200 bps หรือต่ำกว่า ซึ่งเพียงพอต่อการควบคุมเซอร์โว ความละเอียดของมุมที่เซอร์โวสามารถทำได้ประมาณ 3 องศา ซึ่งความละเอียดของพัลส์ที่ควบคุมได้ประมาณ 10 ไมโครวินาที ส่วนวงจรควบคุม PIC 16F877 จะทำการรับค่าการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาที่พู่ทจากเซนเซอร์วัดความเอียง แปลงผลให้อยู่ในรูปค่ามุมเอียงที่ต้องการ โดยส่งข้อมูลผ่าน RS-232 ให้ผู้ควบคุมหุ่นยนต์ทราบหน้าจอบคอมพิวเตอร์ว่าหุ่นยนต์เอียงตัวไปเท่าใดแล้ว

ข้อมูลตำแหน่งขาสำหรับกำหนดการทำงานของไมโครคอนโทรลเลอร์ PIC

PIC 16F87X มีรีจิสเตอร์พิเศษที่บรรจุข้อมูลสำหรับการกำหนดการทำงานทั้งหมด ชื่อ Configuration word โดยภายใน Configuration word จะบรรจุข้อมูลของการเลือกป้องกันการอ่านข้อมูล ความสามารถรีเซตอัตโนมัติเมื่อค่าไฟเลี้ยงลดต่ำกว่ากำหนด ควบคุมการทำงานของวอตซ์ด็อกไทเมอร์ หรือเลือกแหล่งกำเนิดสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ การกำหนดการทำงานถูกกำหนดผ่านโปรแกรมภาษาเบสิก โครงสร้างตำแหน่งขาแสดงได้ดังรูปที่ 3.21 ตารางที่ 1 แสดงรายละเอียดตำแหน่งขาสัญญาณที่ใช้ภายในวงจรควบคุม PIC 16F877/A



รูปที่ 3.21 แสดงรายละเอียดของตำแหน่งขาไมโครคอนโทรลเลอร์ PIC

(อ้างอิงจาก www.microchip.com)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 แสดงรายละเอียดตำแหน่งขาสัญญาณที่ใช้ภายในวงจรควบคุม PIC 16F877/A

ชื่อขา	ตำแหน่งขา	ชนิดของขา	ชนิดวงจรบัฟเฟอร์	รายละเอียดการทำงาน
MLCR/VPP	1	Input	ขมิตต์ทริกเกอร์	ขารับสัญญาณรีเซตหลักที่ลอจิก 0
VDD	11,32	Input	None	ไฟเลี้ยงตั้งแต่ +2V ถึง +5.5V
VSS	12,31	Input	None	ขาต่อกราวด์
OSC1	13	Input	ขมิตต์ทริกเกอร์	ขาต่อคริสตอลรับสัญญาณ
CLKIN			ซิมอส	นาฬิกา จากภายนอก
OSC2	14	Output	None	ขาต่อคริสตอล/ รับสัญญาณ
CLKOUT				นาฬิกา ความถี่ 1/4 ของ OSC1
ขาพอร์ต C				
RC1	16	Input/Output	ขมิตต์ทริกเกอร์	ขาพอร์ต RC1/ Input แคปเจอร์
CCP2				Output PWM
RC2	17	Input/Output	ขมิตต์ทริกเกอร์	ขาพอร์ต RC2/ Input แคปเจอร์
CCP1				Output PWM
RC6/TxD	25	Input/Output	ขมิตต์ทริกเกอร์	ขาพอร์ต RC6/ Output USART
RC7RxD	26	Input/Output	ขมิตต์ทริกเกอร์	ขาพอร์ต RC7/ Input USART
ขาพอร์ต B				
RB1 - RB7	33 - 40	Input/Output	TTL	ขาพอร์ต RB1 - RB7

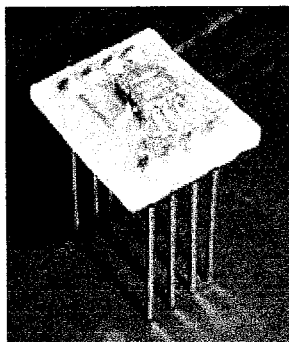
3.5.3 รายละเอียดของเซนเซอร์วัดความเอียง

ในการเลือกใช้เซนเซอร์ เซนเซอร์วัดความเอียง (Tilt sensor) ได้ถูกใช้ในการวัดระดับความเอียงของตัวหุ่นยนต์ ใช้ในการวัดมุมเอียง (Roll) และมุมเงยของตัวหุ่น (Pitch) โดยเซนเซอร์ที่เลือกใช้ผลิตโดยบริษัท MEMSIC เบอร์ MXD 2120ML ดังรูปที่ 3.22

เซนเซอร์วัดความเอียง มีหลักการทำงาน โดยอาศัยหลักการกระจายตัวของความร้อน ภายในฟองอากาศ (Air cavity) ที่อยู่ภายในตัวของเซนเซอร์และส่งสัญญาณที่วัดออกมาได้ในรูปแบบ PMW (Pulse Width Modulate) ซึ่งความกว้างของพัลส์จะแปรผกผันกับมุมเอียงที่วัดได้ อย่างไรก็ตามการติดตั้งที่กลางตัว

หุ่นยนต์เพื่อลดตำแหน่งจุด CG ว่าตกอยู่ ณ ตำแหน่งใดในแต่ละก้าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 เซนเซอร์วัดความเอียง

สามารถใช้วัดค่าความเร่งแบบ 2 แกน คือ แกน X และแกน Y สร้างบนมาตรฐาน CMOS ที่มีระบบการตรวจจับเซนเซอร์สมบูรณ์แบบบนพื้นฐาน SoC (System on Chip) หรือเป็นตัวอย่างหนึ่งของไอซีที่พัฒนาบนพื้นฐานของเทคโนโลยีที่ชื่อว่า MEMS สามารถวัดค่าแบบเต็มค่าสเกลในช่วง $-2g$ ถึง $+2g$ และมีความไว (Sensitivity) 12.5% ต่อ g สามารถวัดค่าได้ทั้งความสั่นสะเทือน (Dynamic หรือ Vibration) และวัดค่าความโน้มถ่วง (Static Acceleration หรือ Gravity) ลักษณะการออกแบบถูกออกแบบให้มีเซนเซอร์แบบความร้อนที่มีการแผ่กระจายความร้อนโดยมีการตรวจจับทั้งในแกน X และ แกน Y สามารถทนทานต่อการสั่นสะเทือนถึง $50,000 g$ และยังมีerkการกำจัดแรงที่มีลักษณะแบบปะทะที่มีผลต่อการวัดให้น้อยลง เพื่อผลการวัดค่าที่มีประสิทธิภาพ รวมถึงค่าอัตราการลดทอนต่างๆ ที่มีผลต่อสัญญาณให้น้อยลง MXD2020ML มีขาสัญญาณดิจิทัลเอาต์พุต 2 ขา คือ $D_{OUT X}$ และ $D_{OUT Y}$ สัญญาณเอาต์พุตนี้จะเปลี่ยนแปลงค่าเป็นสัดส่วนโดยตรงกับค่าความโน้มเอียงหรือค่าความเร่ง ซึ่งการแปลงค่าสัญญาณเอาต์พุตนี้สามารถแปลงค่าโดยตรงกับการเชื่อมต่อกับตัวไมโครคอนโทรลเลอร์ได้

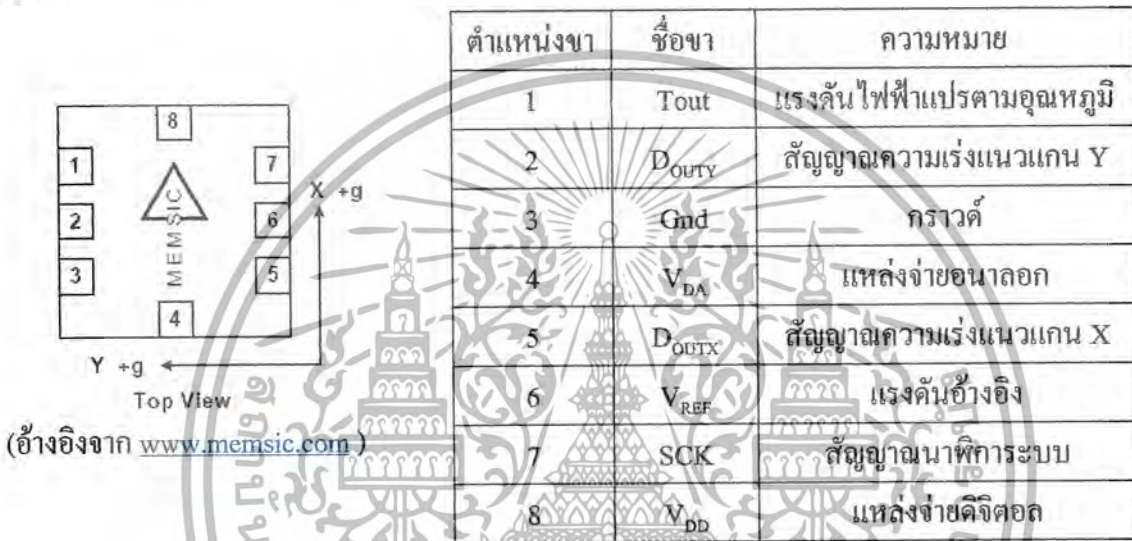
หลักการทำงานอยู่บนพื้นฐานการถ่ายโอนความร้อนแบบกระจายความร้อนอย่างเป็นธรรมชาติโดยมีตัวกลางในการสร้างความแตกต่างของแรงโน้มถ่วงที่จะทำการวัด คือ แก๊ส

การทำงานของ MXD2020ML จะมีแหล่งกำเนิดความร้อนที่อยู่กึ่งกลางช่องว่างภายในซิลิกอนชิปเหมือนกับช่องว่างระหว่างสารประกอบอลูมิเนียมและโพลิซิลิกอนเทอร์โมไฟล์ (อยู่ในกลุ่มพวกเทอร์โมคัปเปิ้ล) ที่มีตำแหน่งอยู่ระหว่างกลางของแกนสองแกนที่ตัดกัน ลักษณะการทำงานและวัดผลนั้นหากว่าความเร่งหรือแรงโน้มถ่วงไม่มีการเปลี่ยนแปลง อุณหภูมิ ณ จุดเปรียบเทียบที่เป็นลักษณะทางกายภาพของวัสดุที่ใช้ตรวจสอบจะสมมาตรกับแหล่งกำเนิดความร้อน ซึ่งนั่นก็หมายความว่าอุณหภูมิจะเท่ากันทั้งสี่ส่วนของตัวเทอร์โมไฟล์ก็เท่ากับว่าแรงดันเอาต์พุตไม่มีการเปลี่ยนแปลงแต่เมื่อใดที่มีแรงกระทำจากภายนอกทำให้เกิดแรงโน้มถ่วงที่เปลี่ยนแปลงไปหรือในกรณีเกิด

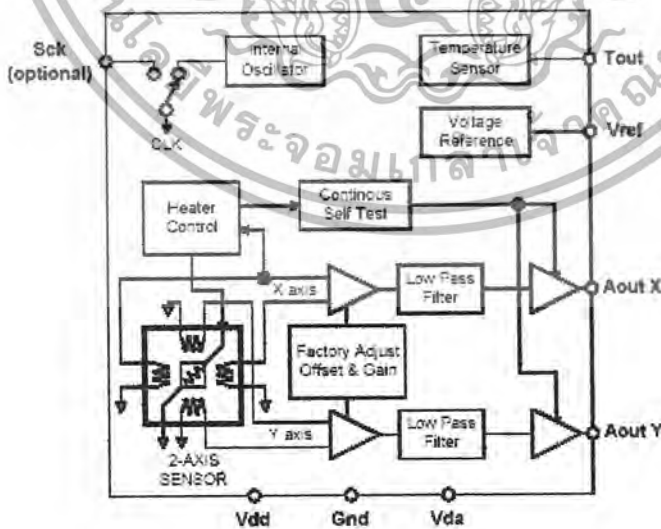
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเอียงหรือการเคลื่อนไหวก็ทำให้โครงสร้างของอุณหภูมิกายในตัวตรวจสอบเปลี่ยนแปลงไปในลักษณะการกระจายความร้อนแบบอิสระ ซึ่งมีผลทำให้โครงสร้างทางกายภาพเสียดสมดุลหรือไม่สมมาตร จึงเป็นผลให้เกิดการเปลี่ยนแปลงค่าแรงดันเอาต์พุตตามระดับค่าที่กำหนดจากโรงงาน โดยค่าการเปลี่ยนแปลงแรงดันที่เกิดจากอุณหภูมิต่างที่เปลี่ยนแปลงภายในตัวตรวจสอบนี้ จะเป็นสัดส่วนกับแรงโน้มถ่วงที่มากระทำว่าเกิดขึ้นตรงส่วนไหนระหว่างแกน X และแกน Y

ตำแหน่งขาของเซนเซอร์วัดมุมเอียงแสดงได้ดังตารางที่ 2 และบล็อกไดอะแกรมแสดงดังรูปที่ 3.23



ตารางที่ 2 แสดงรายละเอียดตำแหน่งขาสัญญาณ

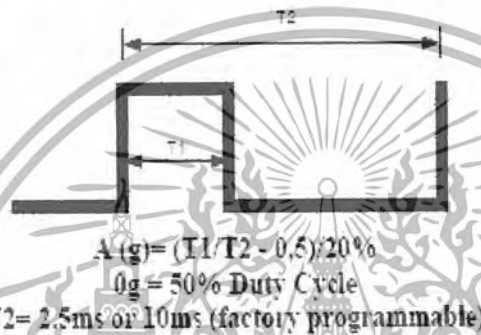


รูปที่ 3.23 บล็อกไดอะแกรมภายในเซนเซอร์วัดความเอียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ (อ้างอิงจาก www.memsic.com) นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติค่าของสัญญาณพัลส์ที่เกิดขึ้นจะมีอยู่ทั้งสองด้านคือทั้งทางด้านแกน X และแกน Y ซึ่งเป็นผลจากวงจรพัลส์วีดิทัศน์มอดูเลเตอร์วีดีโอที่เกิดขึ้นในสภาวะเมื่อแรงโน้มถ่วงปกติหรือค่าจีมีคงที่ เอาท์พุทที่ได้จะถูกเซตให้มีค่าสัญญาณพัลส์ให้มีสัดส่วนระหว่าง T1/T2 เท่ากับ 50% แต่เมื่อใดที่มีแรงมากระทำให้เกิดการเปลี่ยนแปลง ค่า Duty Cycle ก็จะเปลี่ยนแปลงไปตามสเกลค่าเท่ากับ 20% ต่อ ค่าจีหนึ่งค่า ซึ่งค่าที่ได้นี้เป็นค่าที่ได้รับการชดเชยทั้ง Offset error และ Sensitivity error แล้ว ค่าที่ว่านี้จะมีค่าช่วงเวลาเท่ากับ 10 ms หรือเท่ากับ 100 Hz และ 2.5 ms หรือเท่ากับ 400 Hz

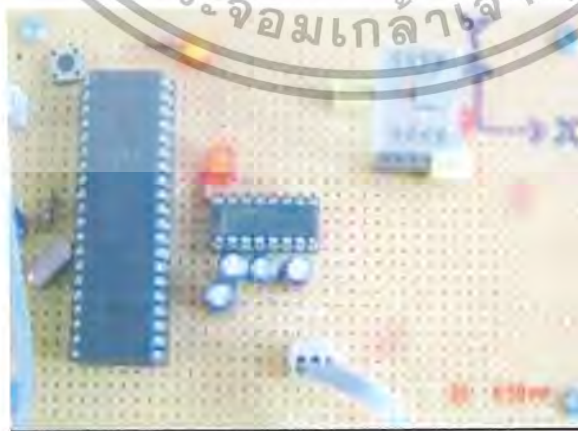
เอาท์พุทคาบเวลาของเซนเซอร์วัดมุมเอียง แสดงได้ดังอัตราส่วนร้อยละของคาบเวลา T1 ต่อคาบเวลา T2 ดังรูปที่ 3.24



รูปที่ 3.24 แสดงเอาท์พุทคาบเวลาช่วง T1 "ON"

(อ้างอิงจาก www.memsic.com)

เซนเซอร์วัดมุมเอียงจะส่งสัญญาณเอาท์พุทออกมาในรูปวีดีโอซีเคิล ที่มีคาบเวลาเปลี่ยนแปลงตามความเอียงในแนวแกน X และ Y คาบสัญญาณนี้ถูกส่งเข้าสู่ไมโครคอนโทรลเลอร์ PIC เพื่อแปลงค่าให้อยู่ในรูปตัวแปรที่ส่งค่าผ่านการอินเตอร์เฟส (RS-232) ออกมาในรูปค่ามุมเป็นองศาได้ โดยกระบวนการทางโปรแกรม วงจรเชื่อมต้อมีลักษณะดังรูปที่ 3.25



รูปที่ 3.25 วงจรเชื่อมต่อเซนเซอร์วัดมุมเอียงโดยไมโครคอนโทรลเลอร์ PIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 การคำนวณค่ามุมเอียงจากเซนเซอร์

การคำนวณค่ามุมเอียงจากเซนเซอร์วัดความเอียง สามารถใช้การประมาณเชิงเส้นได้ (Linear Approximation) โดยการเขียนกราฟค่ามุมที่เปลี่ยนแปลงไปเทียบกับคาบเวลาช่วง “On” หรือ (Duty cycle) ซึ่งใช้การบันทึกเอาที่ทุกคาบเวลา (Duty cycle) ที่เซนเซอร์เอียงแต่ละมุมไปโดยเครื่องมือวัด (Oscilloscope) การคำนวณค่ามุมแบบเชิงเส้นนั้นจะใช้การหาค่าความชันของสมการเส้นตรงระหว่างจุดที่ทราบค่า เช่นจุดที่เซนเซอร์วางอยู่ในแนวระนาบ และจุดที่เซนเซอร์เอียง ซึ่งค่าความชันที่ได้นี้จะถูกใช้ประมาณค่าในทุกๆมุมที่เหลือด้วย วิธีการนี้เรียกว่าการประมาณค่าแบบเชิงเส้น

การประมาณค่ามุมเอียงในแนวแกน X

การประมาณค่ามุมเอียงในแนวแกน X เอาที่ทุกคาบเวลาช่วง “On” ในแนวแกน X ถูกบันทึกค่าจากเครื่องมือวัด (Oscilloscope) เช่น มุม X เอียง $+90$ องศาบันทึกคาบเวลาได้ 6.9 ms มุม X เอียง -90 องศาบันทึกคาบเวลาได้ 2.9 ms หรือ มุม X อยู่ในแนวระดับบันทึกคาบเวลาได้ 5.0 ms จากนี้ก็นำค่าที่ได้มาเขียนกราฟดังรูปที่ 3.26 การประมาณเชิงเส้นจะเลือกใช้เพียงสองจุดก็สามารถหาค่าความชันได้แล้ว



รูปที่ 3.26 แสดงค่ามุม X ที่เปลี่ยนแปลง (แกน Y) เทียบกับคาบเวลาช่วง T1 “On” (%Duty cycle, แกน X) เนื่องจากการหมุนเอียงเซนเซอร์รอบแกน Y

ค่าความชันที่ได้จากกราฟในรูปที่ 3.26 คำนวณได้จากสมการเชิงเส้น

$$y - y_1 = m_x(x - x_1)$$

เลือกใช้จุด (2.9, -90) และ (6.9, +90) จะได้

$$-90 - 90 = m_x(2.9 - 6.9)$$

$$\text{จะได้ } m_x = 4.5$$

ดังนั้นค่ามุม X ที่จุดอื่นๆนั้นสามารถใช้ค่า $m_x = 4.5$ หาสมการเชิงเส้นได้ดังนี้

เลือกจุด (50, 0) จะได้

$$y - 0 = 4.5(x - 50)$$

$$y = 4.5x - 225$$

หรือ

$$\theta_x = 4.5T_x - 225$$

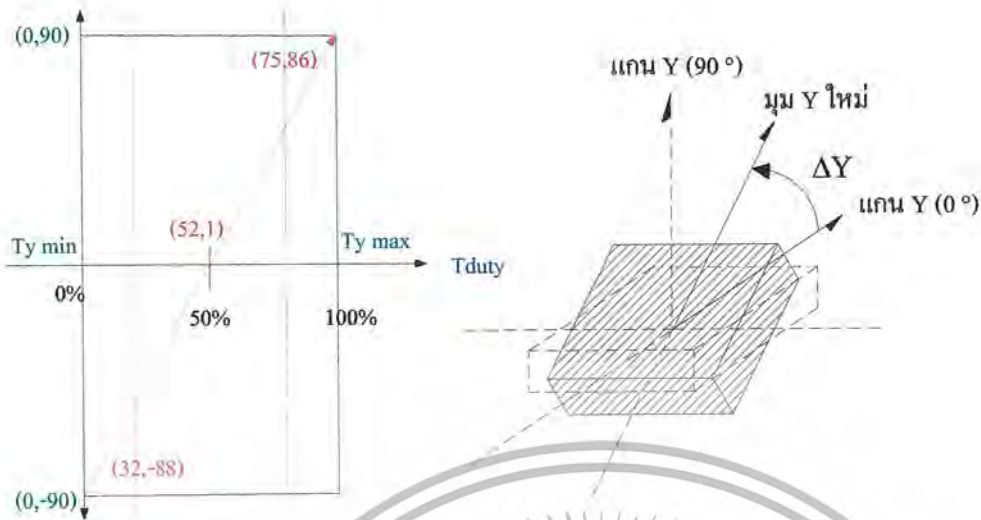
$$\therefore T_x = \frac{1}{4.5}(\theta_x + 225)$$

คาบเวลาช่วง T1 “Om” ที่มุมอื่นๆก็หาได้โดยแทนค่ามุม X ที่ต้องการ

การประมาณค่ามุมเอียงในแนวแกน Y

การประมาณค่ามุมเอียงในแนวแกน Y เอาที่พูดคาบเวลาในแนวแกน Y ถูกบันทึกค่าจาก เครื่องมือวัด (Oscilloscope) เช่น มุม Y เอียง +86 องศาบันทึกคาบเวลาได้ 7.5 ms มุม X เอียง -88 องศาบันทึกคาบเวลาได้ 3.2 ms หรือ มุม Y อยู่ในแนวระดับบันทึกคาบเวลาได้ 5.2 ms จากนั้นก็นำค่าที่ได้มาเขียนกราฟดังรูปที่ 3.27 การประมาณเชิงเส้นจะเลือกใช้เพียงสองจุดก็สามารถหาค่าความชันได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.27 แสดงค่ามุม Y ที่เปลี่ยนแปลง (แกน Y) เทียบกับคาบเวลาช่วง T_1 "On" (%Duty cycle, แกน Y) เนื่องจากการหมุนเชิงเส้นเซอร์รอบแกน X

ค่าความชันที่ได้จากกราฟในรูปที่ 3.27 คำนวณได้จากสมการเชิงเส้น

$$y - y_1 = m_y(x - x_1)$$

เลือกใช้จุด (3.2, -88) และ (7.5, +86) จะได้

$$-88 - 86 = m_y(3.2 - 7.5)$$

$$\text{จะได้ } m_y = 4.0465$$

ดังนั้นค่ามุม X ที่จุดอื่น ๆ นั้นสามารถใช้ค่า $m_y = 4.0465$ หาสมการเชิงเส้นได้ดังนี้

$$\text{เลือกจุด (52, 1) จะได้ } y - 1 = 4.0465(x - 52)$$

$$\text{หรือ } \theta_y = 4.0465 T_y - 209.418$$

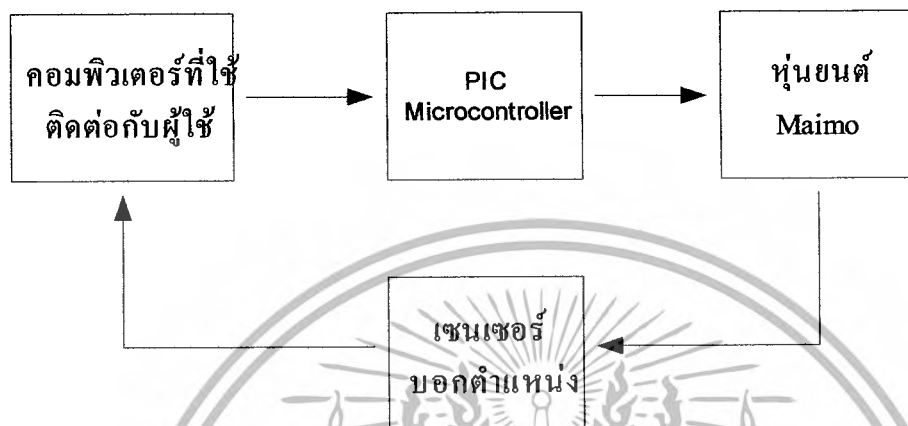
$$\therefore T_y = \frac{1}{4.0465} (\theta_y + 209.418)$$

คาบเวลาช่วง T_1 "On" ที่มุมอื่น ๆ ก็หาได้โดยแทนค่ามุม Y ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบโปรแกรมควบคุมหุ่นยนต์ Maimo1

ระบบทั้งหมดของการควบคุมหุ่นยนต์ Maimo1 ถูกออกแบบให้มีระบบในการควบคุมการเคลื่อนไหวโดยรวมดังรูปที่ 3.28



รูปที่ 3.28 แสดงบล็อกไดอะแกรมของหุ่นยนต์ Maimo1

เมื่อผู้ใช้ทราบค่าที่ต้องการระบุตำแหน่งองศาในการเคลื่อนไหวของมอเตอร์แต่ละตัวบนหุ่นยนต์ Maimo1 หรือแม้แต่เมื่อผู้ใช้ต้องการทดลองหาค่าองศาที่เหมาะสมของมอเตอร์แต่ละตัวแล้ว คอมพิวเตอร์ซึ่งติดต่อกับผู้ใช้ทางพอร์ตอนุกรมจะส่งข้อมูลนั้นไปยัง PIC ไมโครคอนโทรลเลอร์ โดยที่ทาง PIC ไมโครคอนโทรลเลอร์ จะทำการแยกข้อมูลแล้วส่งข้อมูลที่จำเป็นสำหรับมอเตอร์แต่ละตัวแล้วส่งค่าออกไปเพื่อขับให้มอเตอร์ซึ่งทำให้หุ่นยนต์เริ่มมีการเคลื่อนไหว

เมื่อหุ่นยนต์มีการเคลื่อนไหวแล้วนั้น เซนเซอร์ที่ถูกติดไว้บนตัวหุ่นยนต์ Maimo1 ซึ่งสามารถบอกค่าที่เปลี่ยนแปลงไปขององศาของหุ่นยนต์ได้ในแกน X และ แกน Y ส่งค่าที่เปลี่ยนแปลงนั้นกลับมายังพอร์ตอนุกรมเข้าคอมพิวเตอร์ที่ติดต่อกับผู้ใช้ ทำให้ผู้ใช้ทราบค่าที่เปลี่ยนแปลงขององศาของหุ่นยนต์ในสองแกน

หุ่นยนต์ Maimo1 สามารถเคลื่อนไหวได้ทั้งแบบอัตโนมัติ (Automatic) และสามารถควบคุมได้โดยผู้ใช้ (Manual) ซึ่งมีหลักการโดยสังเขปดังนี้

1. การควบคุมหุ่นยนต์โดยอัตโนมัติ (Automatic) นั้น จะค่อนข้างซับซ้อนเพราะค่าทุกค่าที่ส่งออกไปจะต้องกำหนดออกไปในครั้งเดียว ผู้ใช้จะไม่มีส่วนเกี่ยวข้องจนกว่าหุ่นยนต์จะเคลื่อนไหวจนถึงจังหวะที่กำหนด โดยก่อนหน้าที่จะสั่งให้หุ่นยนต์เดินโดยอัตโนมัติแล้วนั้น ผู้ใช้จะต้องทำการจำลองการเดินผ่านโปรแกรมที่ผู้ใช้ติดต่อกับผู้ใช้และ PIC ไมโครคอนโทรลเลอร์เสียก่อน

(Programming Interface) เมื่อแน่ใจในทุกจังหวะการเดินของหุ่นยนต์แล้วจึงทำการสั่งให้หุ่นยนต์โดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การควบคุมหุ่นยนต์โดยผู้ใช้ (Manual) สามารถทำได้โดยผู้ใช้จำเป็นต้องสังเกตการเคลื่อนไหวทุกจังหวะของหุ่นยนต์ แล้วสั่งให้หุ่นยนต์เคลื่อนที่ในทุกจังหวะ

3.6.1 โปรแกรมที่ใช้ติดต่อกับผู้ใช้ (Programming Interface)

การควบคุมโดยอัตโนมัติ แสดงลำดับการทำงานได้ดังรูปที่ 3.29



รูปที่ 3.29 แสดงการลำดับการทำงานของโปรแกรมติดต่อกับผู้ใช้แบบอัตโนมัติ

(Automatic Programming Control)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมโดยผู้ใช้ลำดับการทำงานได้ดังรูปที่ 3.30

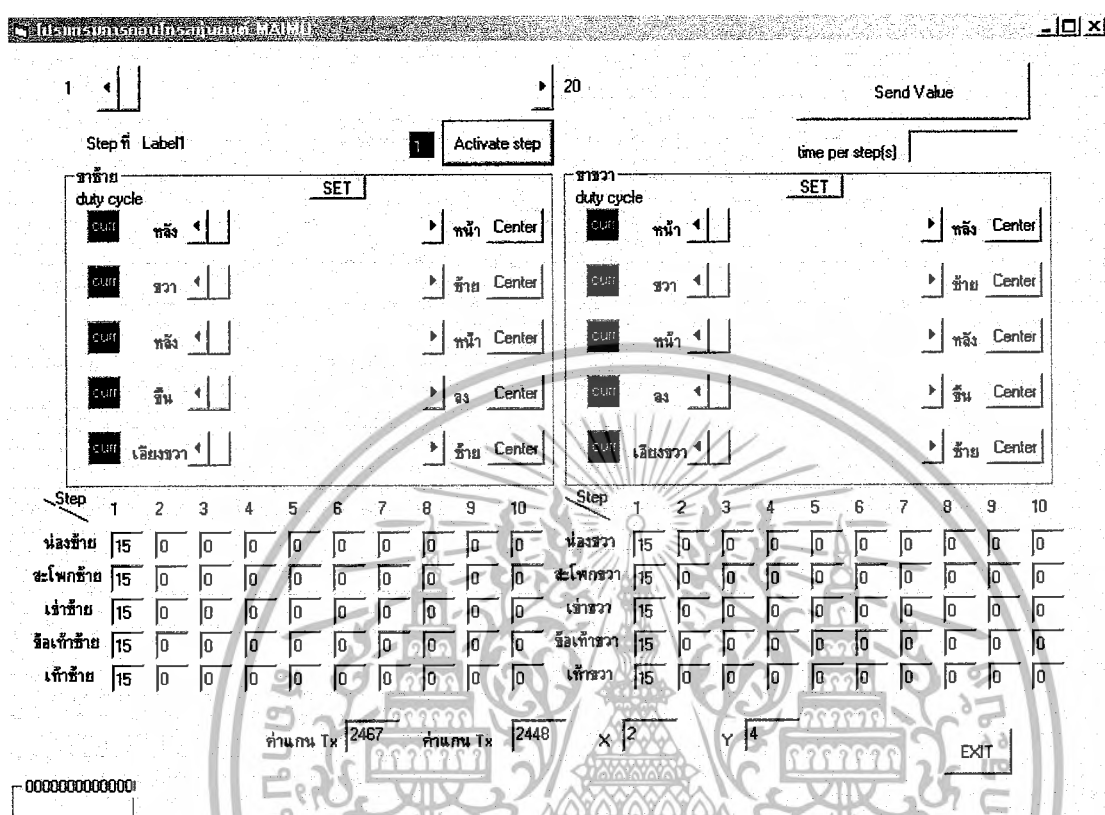


รูปที่ 3.30 แสดงลำดับการทำงานของโปรแกรมติดต่อกับผู้ใช้แบบบังคับ โดยผู้ใช้ (Manual)

จากรูปที่ 3.29 และรูปที่ 3.30 จะเห็นได้ว่าการควบคุมแบบอัตโนมัติและการควบคุมโดยผู้ใช้ (Manual) จะมีความยุ่งยากซับซ้อนที่แตกต่างกัน ซึ่งในรายละเอียดของโปรแกรมมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบและหน้าที่ต่างๆ ของโปรแกรมแสดงหน้าต่างควบคุมได้ดังรูปที่ 3.31



รูปที่ 3.31 แสดงหน้าต่าง โปรแกรมควบคุมหุ่นยนต์ Maimo1

ส่วนประกอบที่สำคัญของโปรแกรมแบ่งออกเป็น 2 ส่วน คือ

1. ส่วนของการรับข้อมูลจากผู้ใช้ เป็นส่วนที่ใช้สำหรับรับข้อมูล (Input) จากผู้ใช้ เช่น การเลือกจังหวะการเดินของหุ่นยนต์ องศาของเซอร์โวมอเตอร์ ระยะเวลาของแต่ละจังหวะการเดินที่ต่างกัน
2. ส่วนของการแสดงผล เป็นส่วนที่บอกค่าต่าง ๆ ที่ผู้ใช้จำเป็นต้องทราบ เช่น ค่าองศาของมอเตอร์ทุกตัวในแต่ละจังหวะของการทำงานของหุ่นยนต์ ค่ามุมของเซนเซอร์ที่เปลี่ยนแปลงไปของหุ่นยนต์ในแกน X และแกน Y

หน้าที่ของส่วนประกอบของ โปรแกรมติดต่อกับผู้ใช้

ส่วนที่ 1 เป็นส่วนที่ใช้เลือกจังหวะการเดินของหุ่นยนต์ดังรูปที่ 3.32

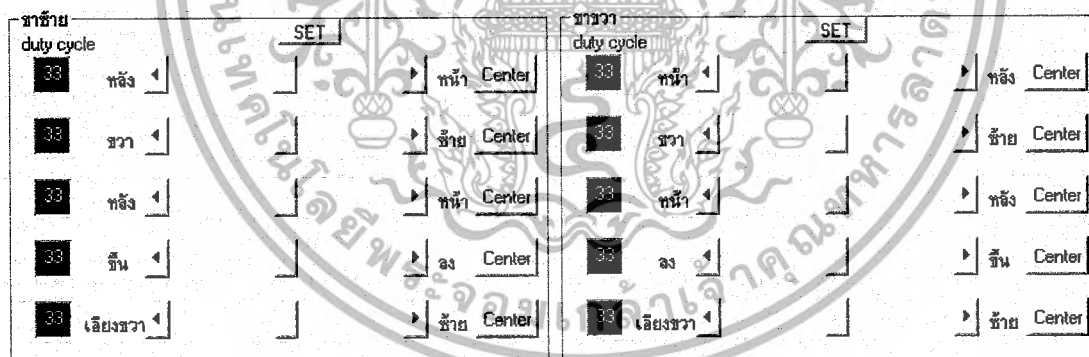


รูปที่ 3.32 แสดงส่วนประกอบส่วนที่ 1 ของ โปรแกรมติดต่อกับผู้ใช้

ส่วนนี้ใช้สำหรับเลือกจังหวะการเดินของหุ่นยนต์ โดยควรจะเลือกจาก Step ที่ 1 ไปจนถึง Step ที่ 20 นั้นหมายความว่า การเคลื่อนที่ของหุ่นยนต์นั้นมีทั้งหมด 20 จังหวะนั่นเอง โดยเมื่อผู้ใช้ได้ กำหนดค่าของมอเตอร์ทุกตัวแล้ว ให้ผู้ใช้คลิก **Activate step** แล้วโปรแกรมจะทำการจดจำข้อมูล ของจังหวะที่ 1 เอาไว้ จากนั้นผู้ใช้สามารถทำแบบเดียวกันนี้ไปเรื่อยๆ จนถึง Step ที่ 20 โปรแกรมจะ จำข้อมูลไว้ทั้งหมดพร้อมที่จะส่งค่าออกไปแบบ Automatic ซึ่งจะอธิบายการทำงานของโปรแกรมอีก ครั้งในส่วนของ โปรแกรม

ส่วนที่ 2 เป็นส่วนที่ใช้ปรับค่าองศาของมอเตอร์รวมทั้งบอก Duty Cycle ในขณะนั้นดังรูปที่

3.33



รูปที่ 3.33 แสดงส่วนประกอบที่ 2 ของ โปรแกรมติดต่อกับผู้ใช้

ส่วนนี้จะมีอยู่ด้วยกัน 4 ปุ่มปฏิบัติการคือ

1. เมื่อผู้ใช้ขยับสวิตช์รอลล์บาร์ค่าที่เปลี่ยนแปลงไปคือค่ามุมองศาที่เปลี่ยนแปลงไปของเซอร์ โวมอเตอร์บนหุ่นยนต์ซึ่งเป็นส่วนหนึ่งที่ขาดไม่ได้ เมื่อผู้ใช้ขยับสวิตช์รอลล์บาร์จะมีผลทำให้มอเตอร์หมุนด้วย

2. ในช่อง Duty Cycle จะเป็นช่องที่บอกกับผู้ใช้ว่าในขณะที่ ค่า Duty cycle ที่ส่งออกไปยัง เซอร์โวเป็นสัญญาณพัลส์วิดซ์มอดูเลชันว่ามีค่าเท่าไร โดยมีค่าอยู่ระหว่าง 15 ถึง 51 (อธิบายในส่วนของ โปรแกรมสร้างพัลส์วิดซ์มอดูเลชัน)
3. ปุ่มเซทจะปุ่มที่เป็นตัวป้อนค่าที่พร้อมจะเดินของหุ่นยนต์โดยแบ่งเป็นขาซ้ายและขาขวา เมื่อกดปุ่มค่าจะเป็นดังรูปที่ 3.34

ขาซ้าย duty cycle		SET	ขาขวา duty cycle	
43	หลัง		หน้า	Center
39	ขวา		ซ้าย	Center
42	หลัง		หน้า	Center
15	ขึ้น		ลง	Center
38	เฉียงขวา		ซ้าย	Center

รูปที่ 3.34 เมื่อผู้ใช้กดคำสั่งเซท

4. ปุ่มเซเตอร์เป็นปุ่มที่ทำให้ค่าของแต่ละมอเตอร์เปลี่ยนเป็นค่ากลางดังรูปที่ 3.34

ส่วนที่ 3 ดังรูปที่ 3.35 เป็นส่วนที่แสดงค่าในทุกจังหวะที่คำสั่งจะส่งออกไปให้กับไมโคร

คอนโทรลเลอร์

Step	1	2	3	4	5	6	7	8	9	10	Step	1	2	3	4	5	6	7	8	9	10
หลังซ้าย	15	0	0	0	0	0	0	0	0	0	หลังขวา	15	0	0	0	0	0	0	0	0	0
สะโพกซ้าย	15	0	0	0	0	0	0	0	0	0	สะโพกขวา	15	0	0	0	0	0	0	0	0	0
เท้าซ้าย	15	0	0	0	0	0	0	0	0	0	เท้าขวา	15	0	0	0	0	0	0	0	0	0
ข้อเท้าซ้าย	15	0	0	0	0	0	0	0	0	0	ข้อเท้าขวา	15	0	0	0	0	0	0	0	0	0
เท้าซ้าย	15	0	0	0	0	0	0	0	0	0	เท้าขวา	15	0	0	0	0	0	0	0	0	0

รูปที่ 3.35 แสดงเมื่อผู้ใช้กดปุ่มสั่งเพียงจังหวะเดียว

จากรูปเมื่อผู้ใช้สั่ง **Activate step** ที่จังหวะที่ 1 จะทำให้โปรแกรมจำค่าทั้งหมดนั้นไว้ในหน่วยความจำอะเรย์ซึ่งในที่นี้กำหนดให้ค่าจังหวะแรกของทุกมอเตอร์เป็น 15 ซึ่งในรูปที่จะแสดงให้เห็นว่าเมื่อผู้ใช้กดปุ่มคำสั่ง **Activate step** ไปทั้งหมด 10 จังหวะจะทำให้มีผลดังรูปที่ 3.36 ซึ่งค่าทั้งหมดจะถูกเก็บไว้ในหน่วยความจำอะเรย์ (ในขณะที่เดียวกันต้องเปลี่ยนค่าที่สกรอลบาร์ในส่วนที่ 1 ด้วย)

Step	1	2	3	4	5	6	7	8	9	10	Step	1	2	3	4	5	6	7	8	9	10
ห้องซ้าย	43	43	43	43	43	43	43	43	43	33	ห้องขวา	30	30	30	30	30	30	30	30	30	33
อะโปกซ้าย	39	39	39	39	39	39	39	39	39	33	อะโปกขวา	37	37	37	37	37	37	37	37	37	33
เช่าซ้าย	42	42	42	42	42	42	42	42	42	33	เช่าขวา	31	31	31	31	31	31	31	31	31	33
ถือเท้าซ้าย	15	15	15	15	15	15	15	15	15	33	ถือเท้าขวา	51	51	51	51	51	51	51	51	51	33
เท้าซ้าย	38	38	38	38	38	38	38	38	38	33	เท้าขวา	36	36	36	36	36	36	36	36	36	33

รูปที่ 3.36 แสดงให้เห็นค่าที่ส่งออกไปในทุกจังหวะ

ส่วนที่ 4 ดังรูปที่ 3.37 เป็นส่วนที่แสดงค่าของเซนเซอร์ที่รับกลับมา

$$\text{ค่าแกน Tx} \begin{matrix} | \\ \hline 2469 \\ \hline \end{matrix} \quad \text{ค่าแกน Tx} \begin{matrix} | \\ \hline 2449 \\ \hline \end{matrix} \quad \times \begin{matrix} | \\ \hline 2 \\ \hline \end{matrix} \quad \text{Y} \begin{matrix} | \\ \hline 4 \\ \hline \end{matrix}$$

รูปที่ 3.37 แสดงการรับค่ากลับมาของเซนเซอร์

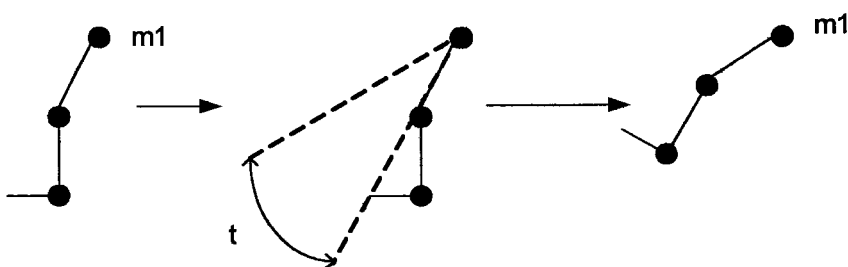
ค่ามุม X มุม Y ที่เปลี่ยนแปลงไป
 ค่าแกน Tx
 ค่าแกน Ty

ส่วนที่ 5 ดังรูปที่ 3.38 เป็นส่วนที่รับค่าระยะเวลาที่ห่างกันของแต่ละจังหวะ

time per step(s)

รูปที่ 3.38 แสดงการป้อนค่าระยะเวลาที่ห่างกันของแต่ละจังหวะ

ค่าที่ป้อนมีหน่วยเป็นวินาที ซึ่งสามารถอธิบายได้ดังรูปที่ 3.39

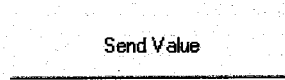


รูปที่ 3.39 แสดงการเคลื่อนที่และช่วงเวลาที่ใช้ของ t

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีเพื่ออุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สังเกต m1 เมื่อก่อนเปลี่ยนแปลงและหลังเปลี่ยนแปลง ซึ่ง t เป็นส่วนที่สำคัญส่วนหนึ่ง หากค่า t ไม่พอเหมาะอาจทำให้หุ่นยนต์ไม่ทรงตัวและล้มได้

ส่วนที่ 6 ดังรูป 3.40 ส่วนที่เป็นคำสั่งส่งค่าทั้งหมดที่อยู่ในอะเรย์ออกไปทางพอร์ตอนุกรมทั้งหมด



รูปที่ 3.40 เป็นปุ่มที่ใช้ส่งค่าออกไปทั้งหมด

โดยค่าที่ส่งออกไปนั้นจะเรียงตั้งแต่ มอเตอร์ 1 จังหวะ 1, มอเตอร์ 2 จังหวะ 1, มอเตอร์ 3 จังหวะ 1,, มอเตอร์ 10 จังหวะ 10 ซึ่งวิธีการทำงานของโปรแกรมจะอธิบายละเอียดในบทของรายละเอียดและหลักการการทำงานของโปรแกรม

3.6.2 รายละเอียดและการทำงานของโปรแกรม

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
Private a (0 To 20, 1 To 10) As Integer
Private set 1(0To 10) As Integer
Private b (2,2) As Integer
Private time As Integer
Private i As Integer
Private k As Integer
Private j As Integer
Private DATA (5) As Integer
Private Datax, sp, spl As String
Private Qx As Double
```

```
Private Sub ActivateCommand_Click ()
    For i = 1 To 10
        a(Hscroll1.Value, i) = Hscroll (i).Value
    Next i
    Text14.Text = Hscroll1.Value
    For k = 1 To 10
        Text1 (k).Text = a(k, 1)
        Text2 (k).Text = a(k, 2)
        Text3 (k).Text = a(k, 3)
    
```

‘ Activate step and keep the data in Array

‘ Step change

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text4(k).Text = a(k, 4)
Text5 (k).Text = a(k, 5)
Text6 (k).Text = a(k, 6)
Text7 (k).Text = a(k, 7)
Text8 (k).Text = a(k, 8)
Text9 (k).Text = a(k, 9)
Text10 (k).Text = a(k, 10)

```

```

Next k
End Sub

```

```

Private Sub Command1_Click ()

```

```

Dim v As Integer

```

```

For i = 1 To 20

```

```

For j = 1 To 10

```

```

a(0, j) = 15

```

```

If i = 1 Then

```

```

MSComm1.Output = Chr$ (255)

```

```

MSComm1.Output = Chr$ (j)

```

```

MSComm1.Output = Chr$ (a (i, j) )

```

```

Elseif a (i, j) < a (i - 1, j) Then

```

```

Do

```

```

MSComm1.Output = Chr$ (255)

```

```

MSComm1.Output = Chr$ (j)

```

```

MSComm1.Output = Chr$ (a (i - 1, j))

```

```

a (i - 1, j) = a (i - 1, j) - 1

```

```

Sleep 45

```

```

Loop Until a (i, j) = a (i - 1, j)

```

```

Elseif a (i, j) > a (i - 1, j) Then

```

```

Do

```

```

MSComm1.Output = Chr$ (255)

```

```

MSComm1.Output = Chr$ (j)

```

```

MSComm1.Output = Chr$ (a (i - 1, j))

```

```

Sleep 45

```

```

a(i - 1, j) = a(i - 1, j) + 1

```

```

Loop Until a (i, j) = a(i - 1, j)

```

```

Else

```

```

End If

```

```

Else

```

```

End If

```

```

Debug.Print i; j; a (i, j)

```

```

Next j

```

```

v = Val(Text13.Text) * 1000

```

```

Sleep v

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next i
End Sub
```

```
Private Sub rightSet_Click()
    HScroll(1).Value = 30
    HScroll(2).Value = 37
    HScroll(3).Value = 31
    HScroll(4).Value = 51
    HScroll(5).Value = 36
    For m = 1 To 5
        MSComm1.Output = Chr$(255)
        MSComm1.Output = Chr$(m)
        MSComm1.Output = Chr$(HScroll(m).Value)
    Next m
End Sub
```

```
Private Sub leftset_Click()
    HScroll(6).Value = 43
    HScroll(7).Value = 39
    HScroll(8).Value = 42
    HScroll(9).Value = 15
    HScroll(10).Value = 38
    For n = 6 To 10
        MSComm1.Output = Chr$(255)
        MSComm1.Output = Chr$(n)
        MSComm1.Output = Chr$(HScroll(n).Value)
    Next n
End Sub
```

```
Private Sub CommandCenter_Click(Index As Integer)
    HScroll(Index).Value = 33
End Sub
```

```
Private Sub CommandExit_Click()
    Timer1.Enabled = False
    MSComm1.PortOpen = False
End Sub
```

```
Public Sub Form_Load ()
    sp = " QX="
    sp1 = " QY="
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MSComm1.Settings = "2400, N, 8, 1"      ' Baud rate 2400,No parity,none stop bit
MSComm1.CommPort = 1                   ' Choose Comm port
MSComm1.PortOpen = True                 ' Open Comm port
Timer1.Interval = 75
Timer1.Enabled = True
Datax = App.Path & "\robotx.txt"
End Sub

```

```

Private Sub HScroll_Change(motor As Integer) ' Send position to motor
degree = Str(HScroll(motor).Value)
MSComm1.Output = Chr$(255)                ' Start Bit
MSComm1.Output = Chr$(motor)
MSComm1.Output = Chr(degree)
Text(motor).Text = Hscroll (motor).Value ' Text change tell the value
Debug.Print degree; motor;                 ' Tell on debug
End Sub

```

```

Private Sub HScroll_Scroll (motor As Integer)
HScroll_Change (motor)
End Sub

```

```

Private Sub Hscroll1_Change () ' Label show number of STEP
Label1 = Hscroll1.Value
End Sub

```

```

Sub Text13_Change ()
Text13.SetFocus
End Sub

```

```

Private Sub Timer1_Timer ()
Dim a As String
On Error Resume Next
MSComm1.DTREnable = False
MSComm1.DTREnable = True
MSComm1.InputLen = 1

```

```

Start:
For n = 1 To 4
Do
DoEvents
Loop Until MSComm1.InBufferCount > 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA (n) = Asc(MSCComm1.Input)
Debug.Print DATA (n)
If DATA (1) < 254 Then GoTo start
Next n
If DATA (4) <> 10 Then GoTo start
If DATA (1) =255 Then
    Ty = 256 * DATA (2) + DATA(3)
    Qy = (-1/11.194)*(Ty-2492.5)
    Text16.Text = Ty
    Text18.Text = Round (Qy)
End If
If DATA (1) =254 Then
    Tx = 256*DATA (2) + DATA (3)
    Qx = (-1/11.194)*(Tx-2492.5)
    Text15.Text = Tx
    Text17.Text = Round (Qx)
End If
Open Datax For Append As #1
Print #1, sp ; Text17.Text ; sp1; Text18.Text
Close #1 'Close the file
End Sub

```

การประกาศค่าตัวแปร

การประกาศค่าตัวแปรว่าจะใช้ฟังก์ชัน Sleep เพื่อใช้ในการหน่วงเวลา

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)
```

การประกาศค่าตัวแปรอะเรย์เพื่อใช้เป็นหน่วยความจำของ โปรแกรม

```
Private a(0 To 20, 1 To 10) As Integer
```

‘ หน่วยความจำอะเรย์ Matrix 20*10

```
Private set1(0 To 10) As Integer
```

‘ หน่วยความจำอะเรย์ Matrix 1*10

```
Private b(2, 2) As Integer
```

‘ หน่วยความจำอะเรย์ Matrix 2*2

```
Private DATA(5) As Integer
```

‘ หน่วยความจำอะเรย์ Matrix 1*5

การประกาศค่าตัวแปรที่ใช้ในโปรแกรม

```
Private time,i,j,k As Integer
```

‘ เป็นหน่วยความจำที่เก็บค่าตั้งแต่ 0 ถึง 255

```
Private Datax, sp, sp1 As String
```

‘ เป็นอะเรย์ที่ใช้เก็บเฉพาะตัวอักษร

```
Private Qx As Double
```

การตั้งค่าเมื่อโปรแกรมเริ่มทำงาน

```
Public Sub Form_Load()
```

‘ โปรแกรมเริ่มทำงาน

```
sp = " QX="
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sp1 = "QY="
MSComm1.Settings = "2400,N,8,1"
MSComm1.CommPort = 1
MSComm1.PortOpen = True
Timer1.Interval = 75
Timer1.Enabled = True
Datax = App.Path & "\robotx.txt"
End Sub

```

‘ กำหนดให้อัตราบอร์ດเรทเป็น 2400, 8 บิท
 ‘ เลือกพอร์ตอนุกรม 1
 ‘ ทำการเปิดพอร์ตอนุกรม 1
 ‘ ช่วงเวลาทำงานอยู่ที่ 75 มิลลิวินาที
 ‘ กำหนดให้ timer 1 ทำงาน
 ‘ กำหนดตัว Datax เชื่อมโยงไปยัง File robotx.txt เพื่อใช้เก็บมุมมองเซนเซอร์ในแต่ละการวนลูป

เมื่อโปรแกรมเริ่มทำงานค่าจะถูกตั้งค่าโดยโปรแกรม กำหนดให้ค่าความเร็วในการสื่อสารระหว่าง คอมพิวเตอร์และไมโครคอนโทรลเลอร์เท่ากับ 2400 ซึ่งไม่มี 8 บิท กำหนดให้เปิดพอร์ตอนุกรม 1 อีกทั้งยังกำหนด Timer ให้ทำงานที่ช่วงเวลากำหนดเท่ากับ 75 มิลลิวินาที

คำสั่งปรับเปลี่ยนองศาของมอเตอร์โดย Scroll Bar

```

Private Sub HScroll_Change(motor As Integer)
degree = Str(HScroll(motor).Value)
MSComm1.Output = Chr$(255)
MSComm1.Output = Chr$(motor)
MSComm1.Output = Chr(degree)
Text(motor).Text = HScroll(motor).Value
Debug.Print degree; motor;
End Sub

```

‘ กำหนดให้เป็น Scroll bar แบบอะเรย์คอนโทรล
 ‘ กำหนดตัวแปร Degree เก็บข้อมูลแบบสตริง
 ‘ บิทเริ่มต้น
 ‘ อันดับของมอเตอร์
 ‘ ค่าของมุม(องศา)
 ‘ บอกค่า Duty Cycle แก่ผู้ใช้

```

Private Sub HScroll_Scroll(motor As Integer)

```

‘ เป็นส่วนของโปรแกรมที่ทำให้เกิด Scroll bar ได้ละเอียดมากขึ้น

```

HScroll_Change(motor)
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้งานคลิกเมื่อ Scroll bar ซึ่งเป็นหน่วยควบคุมแบบอะเรย์ (Control array) การทำงานโดยใช้หน่วยควบคุมแบบอะเรย์จะทำให้ลดความซับซ้อนของโปรแกรมลงเช่น เมื่อกำหนด Private Sub HScroll_Change (Motor As Integer) นั้นหมายถึง ผู้ใช้สามารถควบคุมการเปลี่ยนแปลงไปของ Scroll bar ได้ตั้งแต่ 0 ถึง 255 คำสั่งดังกล่าวอย่าง Hscroll(1), Hscroll(34) เป็นต้น

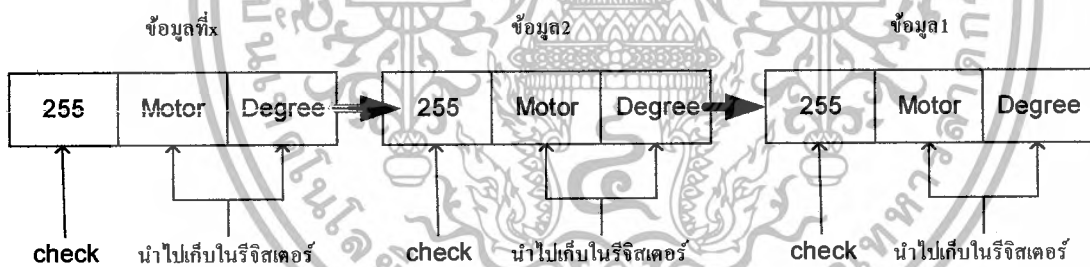
จากโปรแกรมควบคุมเมื่อผู้ใช้งานคลิกที่ปุ่ม Scroll Bar และทำให้ค่าบน Scroll bar เปลี่ยนแปลง จะทำให้โปรแกรมส่งค่า 3 ค่าผ่านทางพอร์ตอนุกรมด้วยคำสั่ง MSComm1.Output ส่งค่าออกไปเป็นข้อมูลแบบสตริงซึ่งเป็นเฉพาะตัวอักษรเท่านั้น แต่ละค่ามีความหมายดังนี้คือ

255 : เป็นค่าที่เปรียบเสมือนบิตเริ่มต้นที่จะสั่งให้ริจิสเตอร์ในไมโครคอนโทรลเลอร์จดจำค่าในบิตต่อไป

Motor : เป็นค่าที่ส่งออกไปเพื่อให้ทราบว่าค่าองศา(degree) นั้นเป็นของมอเตอร์ที่เท่าไร ซึ่งค่านี้ได้มาจากคอนโทรลอะเรย์

Degree : เป็นค่าของมุมของแต่ละมอเตอร์

รูปที่ 3.41 แสดงการส่งข้อมูลออกทางพอร์ตอนุกรมอย่างง่าย (จะอธิบายเพิ่มเติมอีกครั้งในส่วนของโปรแกรมในไมโครคอนโทรลเลอร์) คือ



รูปที่ 3.41 แสดงการตรวจสอบและการเก็บข้อมูลของไมโครคอนโทรลเลอร์

คำสั่ง Set และคำสั่ง Center

เนื่องจากคำสั่ง RightSet และ LeftSet มีการทำงานคล้ายคลึงกันดังนั้นในที่นี้จะกล่าวถึงเพียง RightSet เท่านั้น

Private Sub rightSet_Click()

HScroll(1).Value = 30

HScroll(2).Value = 37

HScroll(3).Value = 31

กำหนดให้ Scroll bar ที่เป็นหน่วยควบคุมแบบอะเรย์มีค่า 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HScroll(4).Value = 51
HScroll(5).Value = 36
For m = 1 To 5          ‘ วนลูปเพื่อส่งค่าออกไปให้ครบ 5 ค่า
    MSComm1.Output = Chr$ (255)
    MSComm1.Output = Chr$ (m)
    MSComm1.Output = Chr$ (Hscroll (m).Value)
Next m
End Sub

```

เมื่อผู้ใช้โปรแกรมกดปุ่มคำสั่ง Set โปรแกรมจะส่งค่าออกทางพอร์ตอนุกรมทันที 5 ค่าโดยส่งเรียงลำดับจากค่าที่ 1 จนถึงค่าที่ 5 ด้วยการวนลูป For...Next ส่งผลให้หุ่นยนต์ยืนในท่าที่พร้อมจะเดิน โดยจำเป็นต้องกดที่ขาซ้ายและขวา

ส่วนคำสั่ง Center ที่อยู่ด้านล่างนี้นั้นใช้หลักการของหน่วยควบคุมแบบอะเรย์เช่นกัน ทำให้ผู้ใช้สามารถสั่งให้แต่ละมอเตอร์ไปอยู่ในตำแหน่งกลางได้ตามต้องการ

```

Private Sub CommandCenter_Click(Index As Integer)
HScroll(Index).Value = 33
End Sub

```

คำสั่งเก็บค่ามุมของมอเตอร์ในแต่ละจังหวะ

```

Private Sub ActivateCommand_Click()

```

ของ

‘ คำสั่งในการสั่งโปรแกรมให้เก็บค่ามุมของมอเตอร์ในแต่ละจังหวะ

```

For i = 1 To 10
a(Hscroll1.Value, i) = HScroll(i).Value

```

‘ ทำการเก็บข้อมูลเข้าอะเรย์ matrix 20*10

```

Next i

```

```

Text14.Text = Hscroll1.Value

```

‘ แสดงค่าออกบนตารางแสดงค่าส่วนที่ 3

```

For k = 1 To 10

```

```

Text1 (k).Text = a(k, 1)

```

```

Text2 (k).Text = a(k, 2)

```

```

Text3 (k).Text = a(k, 3)

```

```

Text4(k).Text = a(k, 4)

```

```

Text5 (k).Text = a(k, 5)

```

```

Text6 (k).Text = a(k, 6)

```

```

Text7 (k).Text = a(k, 7)

```

```

Text8 (k).Text = a(k, 8)

```

```

Text9 (k).Text = a(k, 9)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text10 (k).Text = a(k, 10)
Next k

End Sub

เมื่อผู้ใช้งานปุ่มคำสั่ง Activate จะเป็นการเก็บค่าข้อมูลของมุมของมอเตอร์ในแต่ละจังหวะของการเดิน ซึ่งอันดับของจังหวะของการเดินผู้ใช้จะเป็นผู้กำหนดเอง นำข้อมูลเก็บเข้าไปในอะเรย์เมตริกซ์ 20*10 ซึ่งจะนำข้อมูลเหล่านี้ออกมาใช้ในภายหลัง เพื่อการเดินแบบอัตโนมัติ และเมื่อนำข้อมูลเก็บในอะเรย์แล้วข้อมูลจะนำมาแสดงในส่วนที่ 3 ทำให้ผู้ใช้ทราบค่าของมอเตอร์ในแต่ละจังหวะของการเดินของหุ่นยนต์

คำสั่งการส่งค่าทั้งหมดออกไปควบคุมหุ่นยนต์

Private Sub Command1_Click() คำสั่ง Send value เพื่อส่งค่าออกไปทั้ง

หมด

Private Sub Command1_Click ()

Dim v As Integer

For i = 1 To 20

For j = 1 To 10

a(0, j) = 15

If i = 1 Then

MSComm1.Output = Chr\$ (255)

MSComm1.Output = Chr\$ (j)

MSComm1.Output = Chr\$ (a (i, j))

ElseIf a (i, j) < a (i - 1, j) Then

Do

MSComm1.Output = Chr\$ (255)

MSComm1.Output = Chr\$ (j)

MSComm1.Output = Chr\$ (a (i - 1, j))

a (i - 1, j) = a (i - 1, j) - 1

Sleep 45

Loop Until a (i, j) = a (i - 1, j)

ElseIf a (i, j) > a (i - 1, j) Then

Do

MSComm1.Output = Chr\$ (255)

MSComm1.Output = Chr\$ (j)

MSComm1.Output = Chr\$ (a (i - 1, j))

Sleep 45

a(i - 1, j) = a(i - 1, j) + 1

Loop Until a (i, j) = a(i - 1, j)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
  End If
Else
  End If
Debug.Print i; j; a (i, j)
Next j

```

```
v = Val(Text13.Text) * 1000
```

‘ ค่าระยะเวลาห่างของแต่ละจังหวะการเดิน

```
Sleep v
```

‘ ฟังก์ชันการหน่วงเวลาของโปรแกรม

```

Next i
End Sub

```

เมื่อผู้ใช้คลิกคำสั่ง Send Value ค่าทั้งหมดที่อยู่ในอะเรย์จะถูกส่งออกไปทางพอร์ตอนุกรม โดยที่ค่าเริ่มต้นจะกำหนดให้เป็นค่าที่ผู้ใช้กำหนด (เลื่อน scroll bar) แต่เนื่องจากเมื่อเซอร์โวมอเตอร์มีการเปลี่ยนตำแหน่งในจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง ดังนั้นจึงแก้ปัญหาการกระชากของมอเตอร์โดยสั่งให้มอเตอร์ค่อยๆ เปลี่ยนองศาของมอเตอร์ทีละเล็กน้อยโดยคำสั่ง

```

Do
  MSComm1.Output = Chr$ (255)
  MSComm1.Output = Chr$ (j)
  MSComm1.Output = Chr$ (a (i - 1, j))
  a (i - 1, j) = a (i - 1, j) - 1
  Sleep 45
Loop Until a (i, j) = a (i - 1, j)
ElseIf a (i, j) > a (i - 1, j) Then

```

```

Do
  MSComm1.Output = Chr$ (255)
  MSComm1.Output = Chr$ (j)
  MSComm1.Output = Chr$ (a (i - 1, j))
  Sleep 45
  a(i - 1, j) = a(i - 1, j) + 1
Loop Until a (i, j) = a(i - 1, j)

```

สามารถอธิบายได้ดังตัวอย่าง

กำหนดให้ค่าเริ่มต้นเป็น 30 เมื่อค่า Duty Cycle เปลี่ยนจาก 30 เป็น 45 โปรแกรมจะส่งค่าตั้ง แต่ 30 ไปจนถึง 45 คือ 30.31.32...45 เนื่องจาก $45 > 30$ เมื่อค่า Duty Cycle เปลี่ยนจาก 30 เป็น 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมจะส่งค่าจาก 30 ลงไป 15 คือ 30.29.28...15 เนื่องจาก 15<30 รูปแบบนี้จะทำให้มอเตอร์หมุนรอบเรียบมากขึ้น

ส่วนฟังก์ชัน Sleep ใช้สำหรับหน่วงเวลาในแต่ละจังหวะดังรูปที่... ซึ่งผู้ใช้สามารถกำหนดค่าหน่วงเวลานี้ได้จากสูตร

$$\text{ค่าหน่วงเวลา} = \text{ค่าที่ป้อน} * 1000 \text{ (มิลลิวินาที)}$$

โปรแกรมในส่วนของ Tilt Sensor

เริ่มจากตัวเซนเซอร์ให้สัญญาณเอาต์พุตพัลส์ทั้งแกน X และ แกน Y ออกมาจากนั้นสัญญาณเอาต์พุตพัลส์จะถูกนับคาบเวลาของการเอียงมุม (Duty cycle) โดยไมโครคอนโทรลเลอร์ PIC16F877 จากนั้นจะแปลงค่าให้อยู่ในรูปรหัสแอสกีและถูกส่งมาโดยสายสัญญาณ RxD ของพอร์ตอนุกรม RS-232 เข้าแสดงผลที่จอคอมพิวเตอร์โดยโปรแกรมวิซวลเบสิก (Visual Basic) ซึ่งจะทำการรับค่ารหัสแอสกีและแปลงค่าให้อยู่ในรูปค่ามุมเอียงทั้งแนวแกน X และแนวแกน Y ส่วนของโปรแกรมวิซวลเบสิกสามารถอธิบายได้ดังนี้ (ส่วนโปรแกรมนับคาบเวลาของการเอียงมุมที่ไมโครคอนโทรลเลอร์จะอธิบายในหัวข้อการติดต่อไมโครคอนโทรลเลอร์)

```
Private Sub Timer1_Timer() ' กำหนดการใช้งาน Timer1
Dim a As String ' ถ้าเกิดความผิดพลาดขึ้นจะข้ามไปทำคำสั่งถัดไป
On Error Resume Next
MSComm1.DTREnable = False
MSComm1.DTREnable = True ' กำหนดการบ่งบอกว่าการส่งข้อมูลเกิดขึ้นที่ขา
DTR นี้ โดยการเปลี่ยนจาก False เป็น True
MSComm1.InputLen = 1 ' กำหนดความยาวของข้อมูลที่จะส่ง
```

Start:

For n = 1 To 4

Do

DoEvents

Loop Until MSComm1.InBufferCount > 0

DATA (n) = Asc(MSComm1.Input)

Debug.Print DATA (n)

If DATA (1) < 254 Then GoTo start

Next n

If DATA (4) <> 10 Then GoTo start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในส่วนนี้เริ่มจากค่ามุมในรูปรหัสแอสกี (MScComm1.Input) ถูกส่งมาจากไมโครคอนโทรลเลอร์ PIC16F877 ทางพอร์ต MScComm1 (ข้อมูลเข้ามาที่บัฟเฟอร์แล้ว แสดงได้โดย MScComm1.InBufferCount > 0) จากนั้นค่ารหัสแอสกีจะถูกแปลงเป็นค่าสตริงโดยคำสั่ง DATA (n) = Asc(MScComm1.Input) ซึ่งเก็บในตัวแปร DATA (n) จากการวนลูปเมื่อค่า DATA (n) น้อยกว่า 254 และค่า DATA (4) ไม่เท่ากับ 10 จะกลับไปวนลูปใหม่ (ค่ามุมรหัสแอสกีจาก PIC16F877 จะถูกส่งมาเรื่อยๆ ค่าที่ได้จะเข้าลูปก็ต่อเมื่อตรงตามเงื่อนไขโปรแกรมข้างล่างนี้)

```

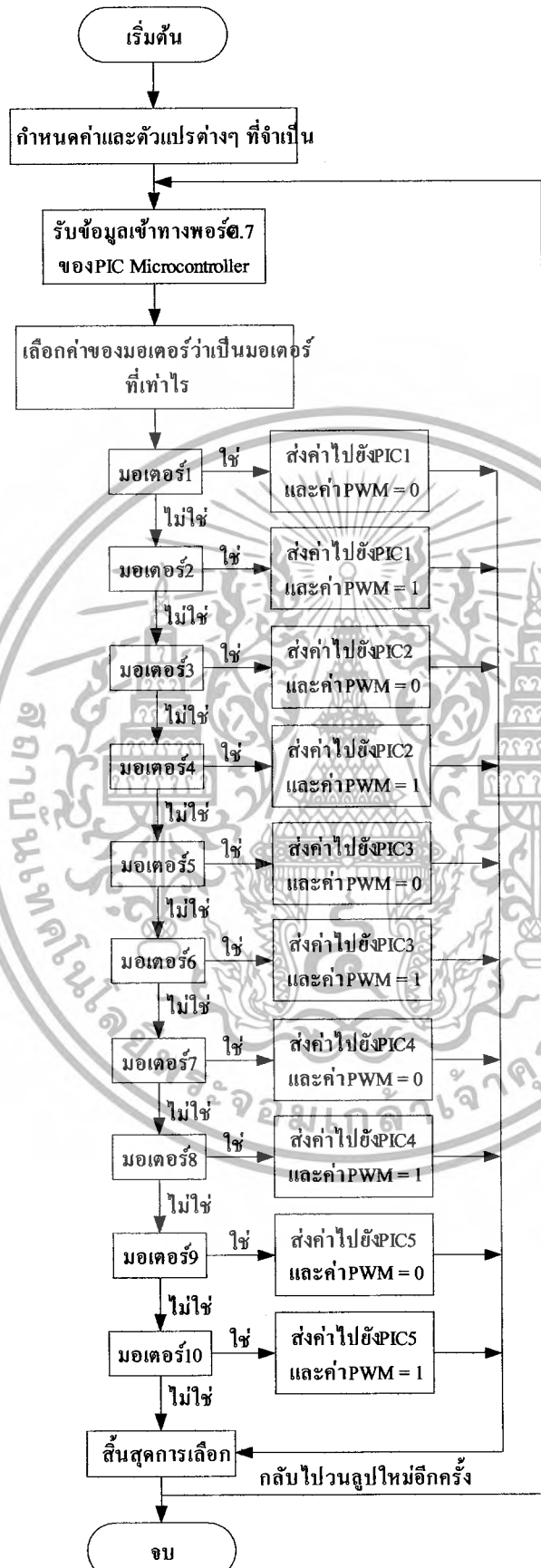
If DATA (1) =255 Then
  Ty = 256 * DATA (2) + DATA(3)
  Qy = (-1/11.194)*(Ty-2492.5)
  Text16.Text = Ty
  Text18.Text = Round (Qy)
End If
If DATA (1) =254 Then
  Tx = 256*DATA (2) + DATA (3)
  Qx = (-1/11.194)*(Tx-2492.5)
  Text15.Text = Tx
  Text17.Text = Round (Qx)
End If
Open Datax For Append As #1
Print #1, sp ; Text17.Text ; sp1; Text18.Text
Close #1 'Close the file
End Sub

```

ถ้าค่า DATA (1) = 255 หรือ DATA (1) = 254 ซึ่งตรงตามเงื่อนไข ค่ามุม Tx หรือค่ามุม Ty จะถูกคำนวณ โดยสมการเชิงเส้นตรง (ซึ่งอธิบายใน ส่วนการคำนวณมุมเอียงจากเซนเซอร์) จากนั้นค่าทั้งหมดจะถูกบันทึกการเปลี่ยนแปลงทุกๆครั้งที่ผู้ควบคุมสั่งรัน โปรแกรมโดยคำสั่ง

Open Datax For Append As #1	‘	สั่งเปิดการบันทึกค่าเปลี่ยนแปลง
Print #1, sp ; Text17.Text ; sp1; Text18.Text	‘	เริ่มบันทึกค่า
Close #1 'Close the file	‘	สั่งปิดการบันทึกค่าเปลี่ยนแปลง

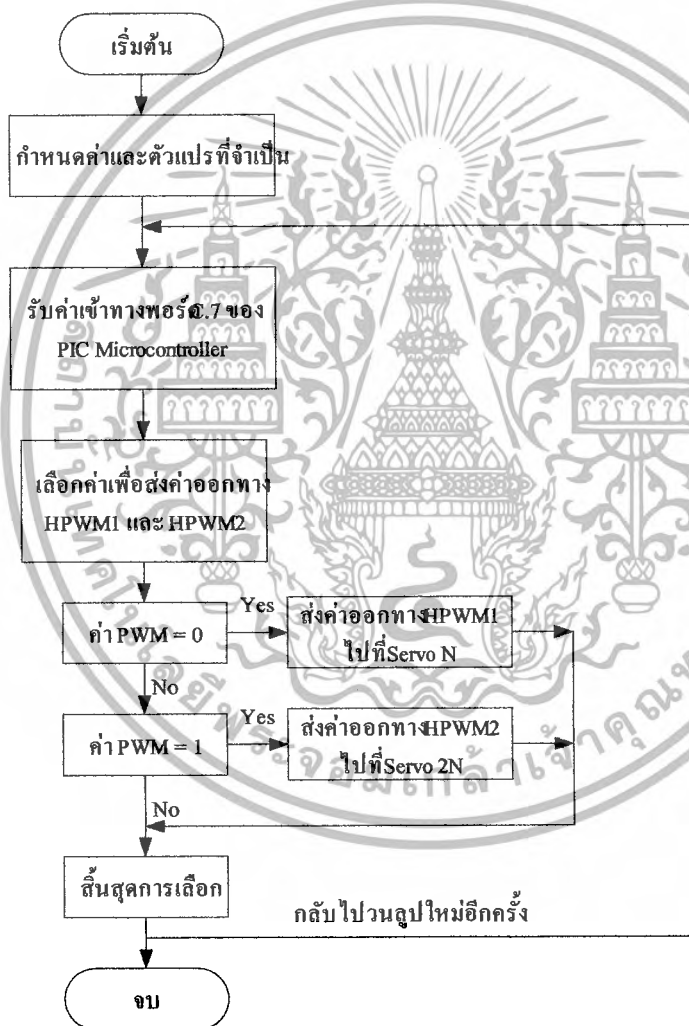
การติดต่อกับไมโครคอนโทรลเลอร์ แสดงลำดับการทำงานได้ดังรูปที่ 3.42



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนสิทธิ์ในเนื้อหา ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของไมโครคอนโทรลเลอร์จะแบ่งการทำงานเป็น 2 ส่วนหลัก คือ

1. ไมโครคอนโทรลเลอร์หลัก ทำหน้าที่รับข้อมูลทั้งหมดจากคอมพิวเตอร์ผ่านทางโปรแกรมติดต่อผู้ใช้ แล้วส่งข้อมูลไปยังไมโครคอนโทรลเลอร์รอง ซึ่งในระบบนี้ใช้ไมโครคอนโทรลเลอร์เพียงตัวเดียว
2. ไมโครคอนโทรลเลอร์รอง ทำหน้าที่รับข้อมูลที่ได้อุปกรณ์แล้วนำไปสร้างสัญญาณพัลส์วิดท์มอดูเลชัน เพื่อควบคุมเซอร์โวมอเตอร์ ซึ่งระบบนี้ใช้ไมโครคอนโทรลเลอร์ที่เป็นตัวควบคุมมอเตอร์ทั้งหมด 5 ตัว มีลำดับการทำงานดังรูปที่ 3.43



รูปที่ 3.43 แสดงลำดับการทำงานของไมโครคอนโทรลเลอร์รอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดโปรแกรมในไมโครคอนโทรลเลอร์

ภาษาที่ใช้ในการควบคุมนี้ใช้ภาษาเบสิกซึ่งมีข้อดีคือสามารถเข้าใจง่ายและมีประสิทธิภาพสูง โดยโปรแกรมนี้นี้ถูกออกแบบออกมาเป็นสองส่วนเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์หลักและไมโครคอนโทรลเลอร์รองดังนี้

โปรแกรมในส่วนของไมโครคอนโทรลเลอร์หลัก

```

Include "modedefs.bas"
Define osc 4
Define HPWM2_TIMER 1
Define HPWM1_TIMER 2
Trisc.7 = 1
portd.1 = 1
trisd = $00
trisB = $00
degree var byte
degree1 var byte
a var byte
check var byte
motor var byte
pwmvalue0 var byte
pwmvalue1 var byte
pwmvalue0 = 0
pwmvalue1 = 1
loop:serin2 portc.7, 396, [wait(255), motor, degree]
a = degree
select case motor
case 1
serout2 portd.2, 16780, ["A", pwmvalue0, a]

```

‘ใช้ข้อสวิตเลเตอร์ความถี่ 4 เมกกะเฮิรตซ์
‘ กำหนดให้ timer 1 สร้างพัลส์วัดซ์มอดูเลชั่นแบบต่อเนื่องออกทาง พอร์ต RC2
‘ กำหนดให้ timer 2 สร้างพัลส์วัดซ์มอดูเลชั่นแบบต่อเนื่องออกทาง พอร์ต RC1
‘ ให้ พอร์ต c.7 เป็นพอร์ตอินพุท
‘ ให้ พอร์ต d.1 เป็นพอร์ตตรวจสอบโดย LED
‘ ให้ พอร์ต D เป็นพอร์ตเอาต์พุท
‘ ให้ พอร์ต B เป็นพอร์ตเอาต์พุท
‘ กำหนดค่าตัวแปร degree มีขนาด 1 ไบท์
‘ กำหนดค่าของตัวแปร pwmvalue0 = 0
‘ กำหนดค่าของตัวแปร pwmvalue1 = 1
‘ รับค่าจากคอมพิวเตอร์เข้าทางพอร์ต C.7
‘ เลือกมอเตอร์ด้วยคำสั่ง Case
‘ ส่งค่าของมุมและ ค่า pwmvalue0 ไปที่ PIC1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 2
  serout2 portd.2,16780,["A",pwmvalue1,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue1 ไปที่
                                             PIC1
case 3
  serout2 portd.3,16780 ,["A",pwmvalue1,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue0 ไปที่
                                             PIC2
case 4
  serout2 portd.3,16780 ,["A",pwmvalue1,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue1 ไปที่
                                             PIC2
case 5
  serout2 portb.0,16780 ,["A",pwmvalue0,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue0 ไปที่
                                             PIC3
case 6
  serout2 portb.0,16780,["A",pwmvalue1,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue1 ไปที่
                                             PIC3
case 7
  serout2 portb.1,16780 ,["A",pwmvalue0,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue0 ไปที่
                                             PIC4
case 8
  serout2 portb.1,16780 ,["A",pwmvalue1,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue1 ไปที่
                                             PIC4
case 9
  serout2 portb.2,16780 ,["A",pwmvalue0,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue0 ไปที่
                                             PIC5
case 10
  serout2 portb.2,16780 ,["A",pwmvalue1,a]  ‘ ส่งค่าของมุมและ ค่าpwmvalue1 ไปที่
                                             PIC5
end select
goto loop
end

```

เมื่อไมโครคอนโทรลเลอร์หลักรับค่ามุม (Motor) และค่าอันดับ (Degree) ของมอเตอร์แล้ว โปรแกรมจะรับค่าเข้าสู่เงื่อนไขในการเลือกไมโครคอนโทรลเลอร์รองที่กำลังจะถูกส่งออกไป เช่น ถ้าค่าที่รับเข้ามา motor = 2 และ degree = 30 ไมโครคอนโทรลเลอร์หลักจะส่งค่า 30 และค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pwmvalue1 ออกไปยังไมโครคอนโทรลเลอร์รอตวี่ที่ 1 (มีทั้งหมด 5 ตัว) เป็นต้น เพราะเหตุนี้ไมโครคอนโทรลเลอร์หลักจะเปรียบเสมือนหน่วยประมวลผลหลักของระบบนั่นเอง

โปรแกรมในส่วนของไมโครคอนโทรลเลอร์รอต

```

Include "modedefs.bas"
Define Osc 4
DEFINE HPWM2_TIMER 1
DEFINE HPWM1_TIMER 2
trisc.7 = 1
trisc.1 = 0
trisc.2 = 0
trisd.2 = 1
trisd.1 = 0
trisd.0 = 0
degree var byte
degree1 var byte
check var byte
motor var byte
pwmvalue var byte
portd.1 = 1
loop:serin2 portd.2,16780,[wait("A"),pwmvalue,degree]
  select case pwmvalue
    case 0
      Hpwm 1,(degree*255)/100,245
    case 1
      Hpwm 2,(degree*255)/100,245
  end select
goto loop
end

```

‘ ใช้ฮอสซิลเลเตอร์ความถี่ 4 เมกะเฮิรตซ์

‘ กำหนดให้ timer 1 สร้างพัลส์วี่ดซ์มอดูเลชั่นแบบต่อเนื่องออกทาง พอร์ต RC2

‘ กำหนดให้ timer 1 สร้างพัลส์วี่ดซ์มอดูเลชั่นแบบต่อเนื่องออกทาง พอร์ต RC2

‘ ให้ พอร์ต c.7 เป็นพอร์ตอินพุท

‘ ให้ พอร์ต c.1 เป็นพอร์ตเอาต์พุท

‘ ให้ พอร์ต c.2 เป็นพอร์ตเอาต์พุท

‘ ให้ พอร์ต d.2 เป็นพอร์ตอินพุท

‘ ให้ พอร์ต d.1 เป็นพอร์ตเอาต์พุท

‘ ให้ พอร์ต d.0 เป็นพอร์ตเอาต์พุท

‘ กำหนดค่าตัวแปร degree มีขนาด 1 ไบท์

‘ ตรวจสอบการทำงาน โดย LED

‘ ส่งค่าพัลส์วี่ดซ์มอดูเลชั่นออกทางพอร์ต RC2

‘ ส่งค่าพัลส์วี่ดซ์มอดูเลชั่นออกทางพอร์ต RC1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อไมโครคอนโทรลเลอร์รองรับค่ามุมและค่า Pwmvalue เข้ามาทางพอร์ต D.2 แล้ว โปรแกรมจะตรวจสอบว่าค่า Pwmvalue = 0 หรือ Pwmvalue = 1 แล้วส่งค่าออกไปทางพอร์ต C.1 หรือ C.2 ซึ่งจากคำสั่ง

```
Hpwm 1, (degree*255)/100,245
```

จะเห็นได้ว่าโปรแกรมจะต้องนำค่ามุมไปคูณกับ 255 แล้วหารด้วย 100 เพื่อเป็นเปอร์เซ็นต์ของความถี่ 245 ซึ่งค่าที่ได้ออกมาจะทำให้เกิดสัญญาณมอดูเลชันอยู่ในช่วงระหว่าง 1 มิลลิวินาทีถึง 2 มิลลิวินาที ที่เป็นค่าที่ทำให้เซอร์โวมอเตอร์ทำงาน ด้วยเหตุนี้เองโปรแกรมนี้อาจต้องถูกออกแบบให้ใช้ออสซิลเลเตอร์ที่สามารถสร้างความถี่ได้ต่ำที่สุดเท่ากับ 244 เมกะเฮิร์ตซ์

โปรแกรมเซนเซอร์ในส่วนไมโครคอนโทรลเลอร์

```
Include "modedefs.bas"
```

```
Define Osc 20
```

```
trisc.6=0
```

```
trisb.4=1
```

```
trisb.5=1
```

```
pin var portb.5
```

```
PIN1 VAR PORTB.4
```

```
Pulsin pin, 0, TX
```

```
Pulsin PIN1, 0, TY
```

‘ใช้งานออสซิลเลเตอร์ความถี่ 20 เมกะเฮิร์ตซ์

‘ กำหนดให้ขา C6 เป็นขาเอาต์พุต

‘ กำหนดให้ขา B4 เป็นขาอินพุต

‘ กำหนดให้ขา B5 เป็นขาอินพุต

‘ ประกาศตัวแปร pin แทนพอร์ต B5

‘ ประกาศตัวแปร PIN แทนพอร์ต B4

‘ ใช้งานคำสั่งนับสัญญาณพัลส์ช่วงลบ

‘ ใช้งานคำสั่งนับสัญญาณพัลส์ช่วงลบ

การทำงานเริ่มจากที่เซนเซอร์เอียงมุมในแนวแกน X หรือ Y สัญญาณเอาต์พุต (Duty cycle) ของแนวแกน X หรือ Y จะถูกส่งมาสู่ไมโครคอนโทรลเลอร์ทางพอร์ต B4 และ B5 ซึ่งเป็นพอร์ตอินพุตหรือเอาต์พุต (รับสัญญาณแบบ TTL) โดยใช้คำสั่งนับสัญญาณพัลส์ช่วงลบ (Pulsin)

```
SEROUT2 PORTC.6, 396, [255, SmX.Highbyte, SmX.Lowbyte, 10]
```

```
SEROUT2 PORTC.6, 396, [254, SmY.Highbyte, SmY.Lowbyte, 10]
```

เป็นคำสั่งที่ใช้ส่งข้อมูลแบบอนุกรมโดยจะทำการส่งข้อมูลที่ละตัวโดยจะส่ง 255 ตามด้วยข้อมูลไบต์สูงและข้อมูลไบต์ต่ำของตัวแปร SmX หรือ SmY และปิดท้ายด้วย 10 ข้อมูลจำพวกนี้อยู่ในรูปแบบของรหัสแอสกีและถูกส่งผ่าน RS-232 เพื่อที่จะรับการต่อข้อมูลด้วยโปรแกรมวิซวลเบสิก ดังที่กล่าวไปในหัวข้อข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลองการเดินของหุ่นยนต์

1. การทดลองโดยควบคุมการเดินของหุ่นยนต์ Maimo1 ที่ละจังหวะ โดยมีผู้ควบคุม (Manual Control) ในการทดลองครั้งแรกหุ่นยนต์ Maimo1 ประกอบด้วยเซอร์โวมอเตอร์ทั้ง 12 ตัว แต่การเดินของหุ่นยนต์นั้นไม่สามารถควบคุมให้ตรงตัวได้ (หุ่นยนต์ล้ม) เนื่องจากพบปัญหาการออกแบบเชิงกลในเรื่องน้ำหนักและความสูงของตัวหุ่นยนต์ ดังนั้นเซอร์โวมอเตอร์ (M1, M2 อ่างอิงตำแหน่งในบทที่ 2 หน้า 17) ที่ใช้ในการหมุนเบะเท้าซึ่งไม่เกี่ยวข้องกับการเดิน ไปข้างหน้าของหุ่นยนต์จึงถูกตัดออก ดังรูปที่ 4.1



รูปที่ 4.1 แสดงหุ่นยนต์ Maimo1 ภายหลังจากการลดความสูง และเซอร์โวมอเตอร์ M1 ทั้งด้านซ้ายและขวา

2. ในการทดลองครั้งต่อมา หุ่นยนต์ Maimo1 ซึ่งถูกลดความสูง น้ำหนักจากการตัดเซอร์โวมอเตอร์ M1, M2 ออกไป (เหลือเซอร์โวมอเตอร์ทั้งหมด 10 ตัว) และเพิ่มแรงเสียดทานที่ผิวเท้าโดยการใส่กระดาษทรายเพื่อป้องกันการลื่นไถล แสดงได้ดังรูปที่ 4.1 ผลการทดลองข้างต้น การควบคุมการเดินสามารถทำได้ โดยหุ่นยนต์จะเดินไปที่ละจังหวะตามรูปแบบการเดินของมนุษย์ แต่การเดินยังไม่สามารถเดินเหมือนมนุษย์ได้เพราะ มีปัญหาทางการควบคุมที่เกิดจากปัญหาเชิงกล คือ ข้อต่อของเซอร์โวมอเตอร์ในท่อนสะโพกที่ใช้การก้าวไปด้านหน้าและหลัง (M3 ทั้งซ้ายและขวา) มีลักษณะโยกไปมาไม่ยึดแน่น ซึ่งรบกวนการควบคุมการเดินของหุ่นยนต์ (แต่ก็ยังสามารถควบคุมการเดินให้ดำเนินไปได้)

3. ในการควบคุมการเดินของหุ่นยนต์ Maimo1 พบว่าไม่สามารถสั่งงานให้เซอร์โวมอเตอร์ทำงานพร้อมกันทั้งหมดได้ เนื่องจาก โปรแกรมอินเตอร์เฟสมีลักษณะการส่งข้อมูลแบบอนุกรม เซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

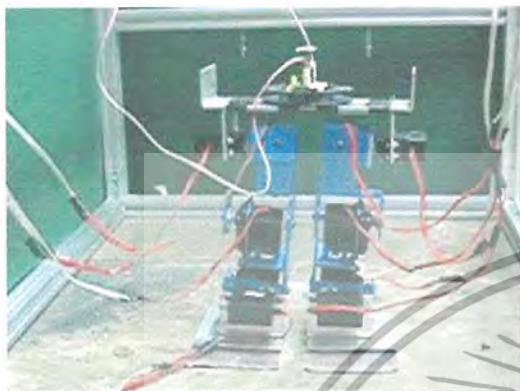
มอเตอร์จึงทำงานทีละตัว เป็นผลให้การเดินของหุ่นยนต์ไม่เป็นไปตามรูปแบบการเดินของมนุษย์เท่าที่ควร ผลการทดลองแสดงการเดินในมุมมองด้านซ้ายของหุ่นยนต์ Maimo1 ดังรูปที่ 4.2



รูปที่ 4.2 แสดงการเดินในมุมมองด้านซ้ายของหุ่นยนต์ Maimo1

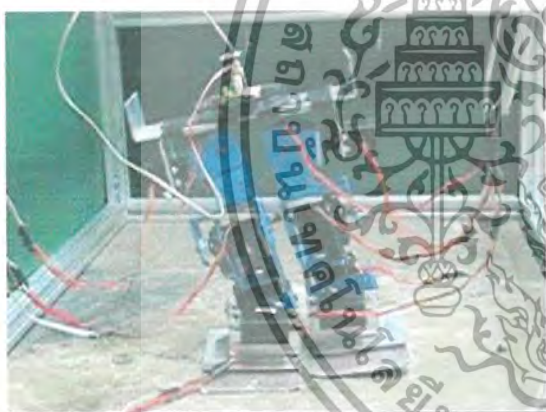
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะการเดินของหุ่นยนต์ Maimo1 ทั้ง 16 จังหวะแสดงผลการก้าวขาเดินไปข้างหน้า 2 ก้าวดังนี้
 จังหวะที่ 1 กำหนดค่าทุกค่าอยู่ในช่วงท่าที่พร้อมจะเดิน (ยืนตรง)



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	43	34	6	-21
2	39	37		
3	42	31		
4	15	51		
5	36	37		

จังหวะที่ 2 เอียงข้อเท้าขวาเพื่อถ่ายน้ำหนักลงสู่เท้าขวาเตรียมพร้อมที่จะยกขาซ้าย



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	43	34	17	-11
2	39	37		
3	42	31		
4	15	51		
5	29	26		

จังหวะที่ 3 กดสะโพกขวาดำลงเพื่อรับน้ำหนักมาอยู่ขาขวาทั้งหมด ยกขาซ้าย พร้อมทั้งกางขา
 ซ้ายออกพร้อมที่จะก้าวออกไป



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	43	34	45	-4
2	42	21		
3	42	31		
4	15	51		
5	29	26		

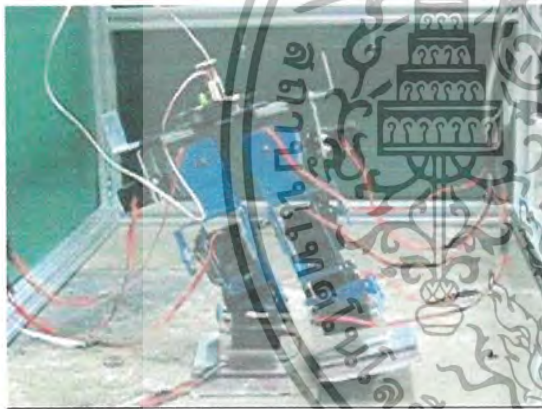
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะที่ 4 ก้าวขาซ้ายไปข้างหน้า โดยน้ำหนักอยู่ที่ขาขวา



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	34	49	28
2	42	21		
3	42	31		
4	15	51		
5	29	26		

จังหวะที่ 5 กดสะโพกซ้ายต่ำลงโดยการยกสะโพกขวา เพื่อให้เท้าซ้ายแตะพื้น



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	34	30	24
2	42	31		
3	42	31		
4	15	51		
5	37	26		

จังหวะที่ 6 เอียงข้อเท้าขวาถ่ายน้ำหนักลงเท้าซ้าย เท้าซ้ายจึงแนบสนิทพื้น



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	34	-18	25
2	42	31		
3	42	31		
4	15	51		
5	41	39		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะที่ 7 เอียงข้อเท้าซ้าย ถ่าน้ำหนักไปเท้าซ้าย เตรียมพร้อมที่จะยกเท้าขวา



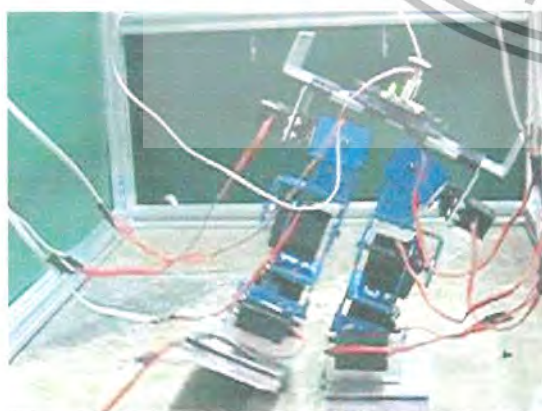
มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	34	-41	10
2	42	31		
3	42	31		
4	15	51		
5	46	39		

จังหวะที่ 8 กดสะโพกซ้ายต่ำลง ยกขาขึ้น พร้อมทั้งกางขาขวาออกพร้อมที่จะก้าวไปข้างหน้า



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	34	-39	8
2	51	31		
3	42	31		
4	15	51		
5	46	39		

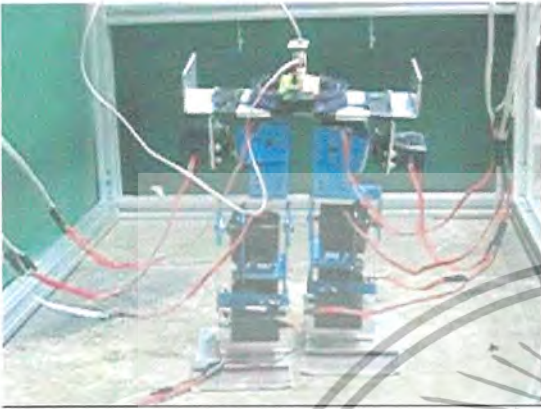
จังหวะที่ 9 ก้าวขาขวามาข้างหน้า น้ำหนักอยู่ที่เท้าซ้าย



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	41	30	-37	6
2	51	31		
3	42	31		
4	15	51		
5	46	39		

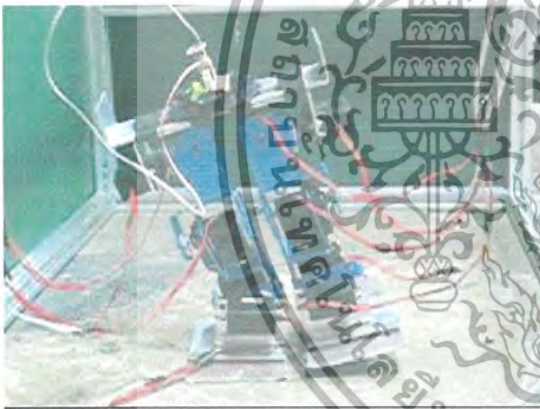
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะที่ 10 ยกสะโพกซ้ายขึ้นและถ่ายน้ำหนักมาที่ฝ่าเท้าขวาที่ละน้อย



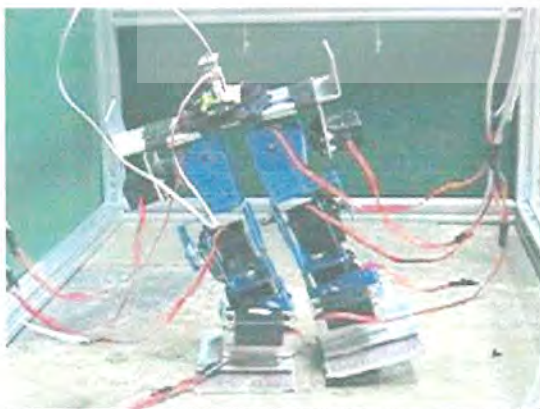
มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	41	30	-22	25
2	41	31		
3	42	31		
4	15	51		
5	46	39		

จังหวะที่ 11 ถ่ายน้ำหนักจากขาซ้ายไปที่ขาขวาทั้งหมด ด้วยข้อเท้าซ้าย



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	41	30	8	18
2	41	31		
3	42	31		
4	15	51		
5	32	39		

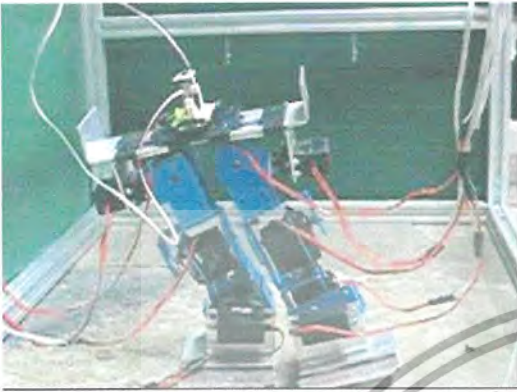
จังหวะที่ 12 เอียงข้อเท้าขวามากขึ้นเพื่อเตรียมยกขาซ้าย น้ำหนักอยู่ที่ฝ่าเท้าขวา



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	41	30	28	11
2	41	31		
3	42	31		
4	15	51		
5	32	27		

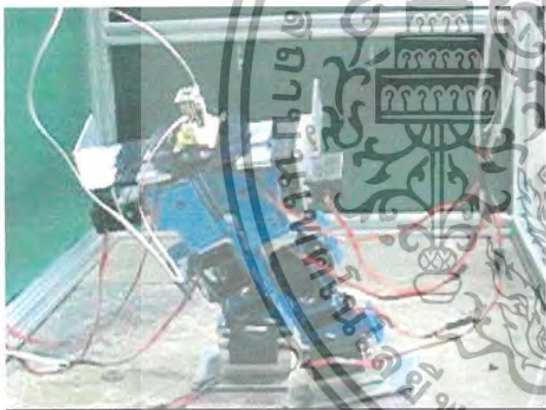
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะที่ 13 กดสะโพกขวาต่ำลงถ่ายน้ำหนักเพื่อยกขาซ้าย



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	41	30	47	14
2	41	22		
3	42	31		
4	15	51		
5	32	27		

จังหวะที่ 14 ก้าวขาซ้ายออกมาเพื่อยืนตรง



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	30	47	10
2	41	22		
3	42	31		
4	15	51		
5	32	27		

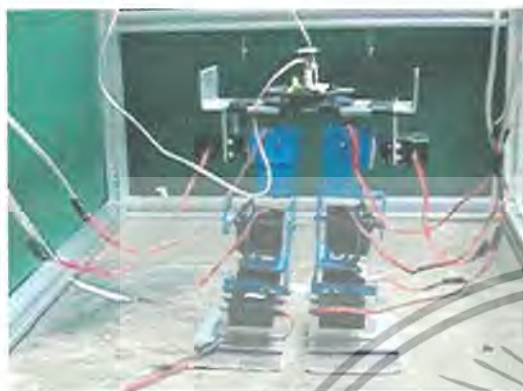
จังหวะที่ 15 ถ่ายน้ำหนักจากสะโพกขวาเพื่อยืนตรง



มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	42	30	24	22
2	41	31		
3	42	31		
4	15	51		
5	32	27		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะที่ 16 ยืนตรง (กลับมาทำยืนตรงเหมือนตำแหน่งเริ่มต้นใหม่อีกครั้ง)

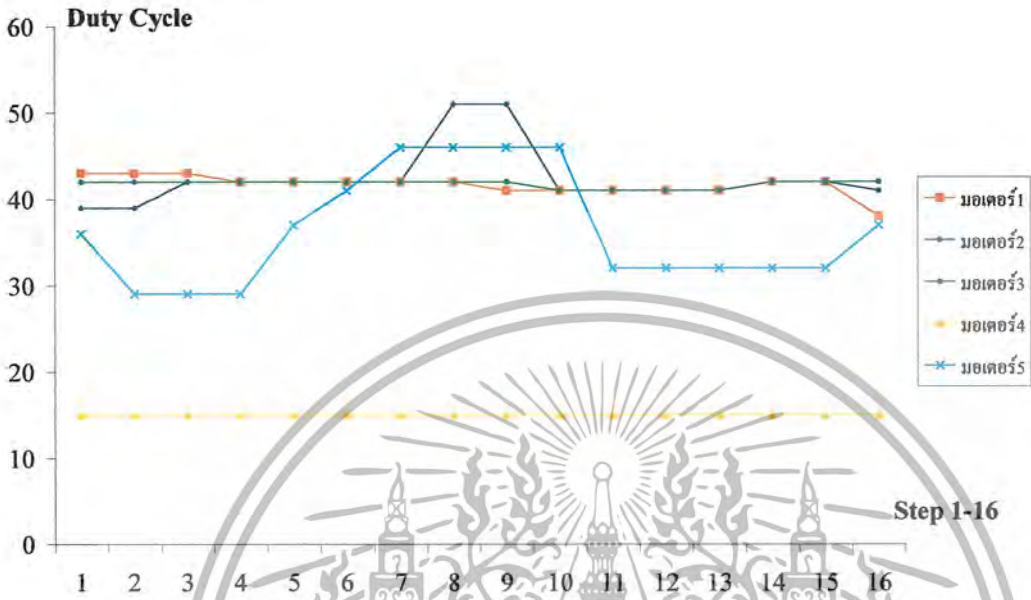


มอเตอร์	Duty Cycle		มุม	
	ขาซ้าย	ขาขวา	X	Y
1	38	33	0	-2
2	41	38		
3	42	25		
4	15	51		
5	37	38		

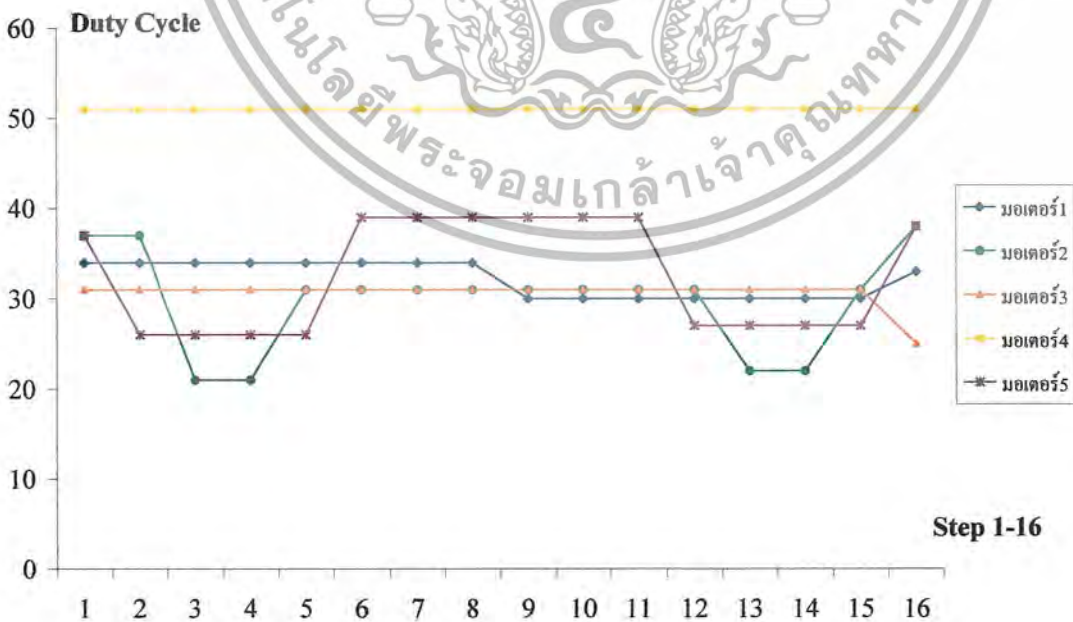
ทั้ง 16 จังหวะเป็นการก้าวเดินในหนึ่งรอบวัฏจักร (Gait cycle) ของหุ่นยนต์ Maimo1 ที่ถูกควบคุมโดยผู้ใช้ที่ละจังหวะ (Step Manual Control) เซนเซอร์วัดมุมเอียงจะถูกติดตั้งอยู่บนฐานของลำตัวหุ่นยนต์ ส่งค่ามุมเอียงในแนวแกน X และ Y แสดงผลออกทางหน้าจอคอมพิวเตอร์

ค่าในแต่ละช่องของตารางเป็นค่าเปอร์เซ็นต์ของดิวตี้ไซเคิล หรือค่า PWM ของเซอร์โวมอเตอร์ในแต่ละตัว โดยที่ค่าความถี่ที่ใช้คือ 245 Hz หรือ คาบเวลาทั้งหมด 4.08 ms (เช่น ในจังหวะแรก หุ่นยนต์ยืนตรง ดิวตี้ไซเคิลขาซ้าย และขาขวาคือ 43% และ 34% ของคาบเวลาทั้งหมด หมายความว่า คาบเวลาที่ใช้ในขาซ้ายและขาขวาคือ 1.755 ms และ 1.3872 ms)

ความสัมพันธ์ระหว่างค่าเปอร์เซ็นต์ดิวตี้ไซเคิลของขาซ้ายและขาขวา เทียบกับจังหวะการเดินทีละจังหวะ ถูกแสดงดังรูปที่ 4.3 และรูปที่ 4.4 และความสัมพันธ์ของค่ามุมเซนเซอร์แนวแกน X และ Y เทียบกับจังหวะการเดินและเทียบกับเวลาที่ใช้เดินทั้ง 16 จังหวะ ถูกแสดงได้ดังรูปที่ 4.5 และรูปที่ 4.6

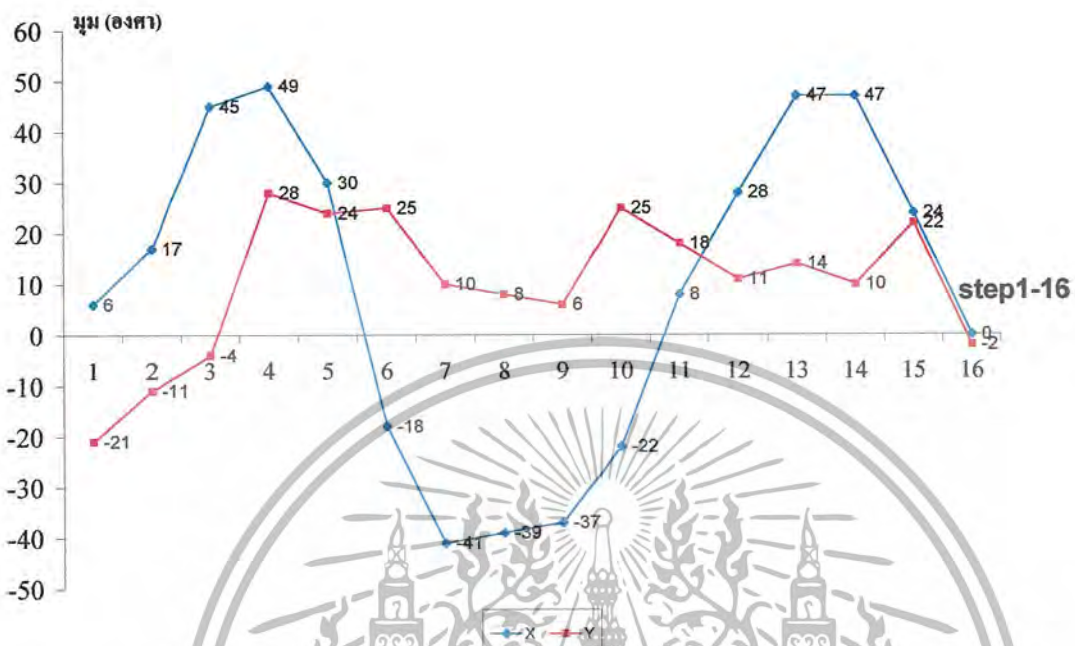


รูปที่ 4.3 กราฟความสัมพันธ์การเดินระหว่างค่าเปอร์เซ็นต์ดีวีดีไซเคิลและจังหวะการเดินในขาซ้าย

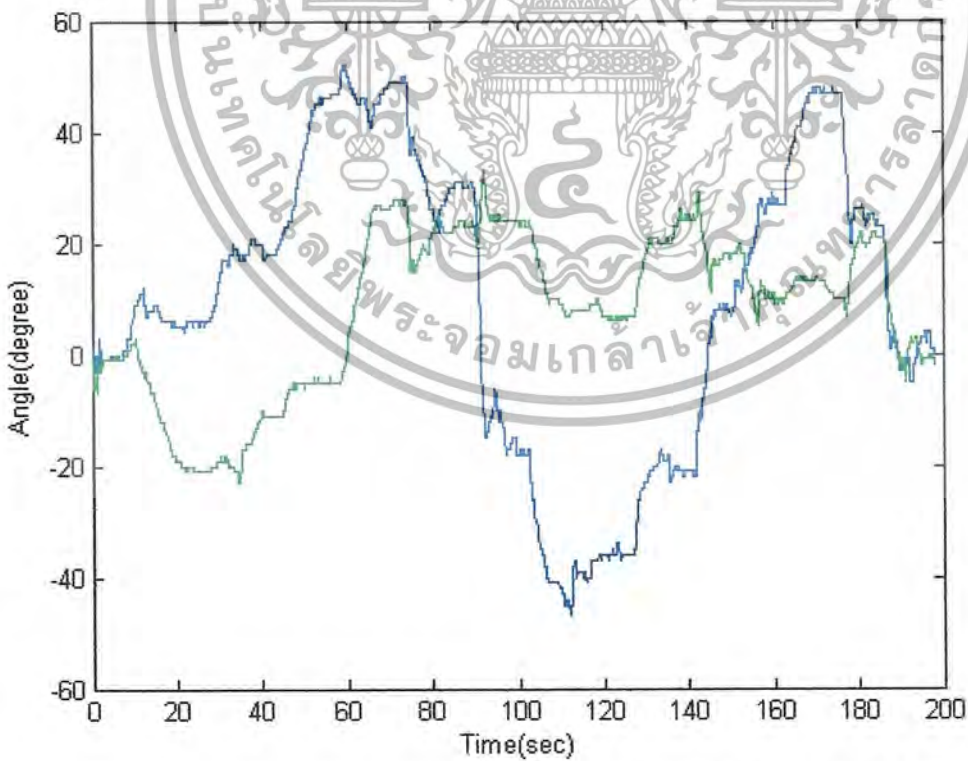


รูปที่ 4.4 กราฟความสัมพันธ์การเดินระหว่างค่าเปอร์เซ็นต์ดีวีดีไซเคิลและจังหวะการเดินในขาขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ความสัมพันธ์ของค่ามุมเซนเซอร์แนวแกน X และ Y เทียบกับจังหวัดการเดินทาง 16 จังหวัด



รูปที่ 4.6 กราฟค่ามุมของเซนเซอร์แนวแกน X และ Y เทียบกับเวลาที่ใช้เดินที่ใช้เดินทั้ง 16 จังหวัด

(เส้นสีน้ำเงินคือ ค่ามุม X เส้นสีเขียวคือค่ามุม Y)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ในการทดลองต่อมาเป็นการทดลองให้หุ่นยนต์ Maimo1 เดินแบบอัตโนมัติ (Automatic Programming Control) เริ่มจากจังหวะแรก ต้องการให้หุ่นยนต์ยืนตรง ก็ปรับค่าควิตี้ไซเคิลของเซอร์โวมอเตอร์ทุกตัวให้อยู่ในลักษณะท่าทางหุ่นยนต์ยืนตรงเตรียมพร้อมที่จะเดินแล้วอินพุตค่าควิตี้ไซเคิลทั้งหมดโดยการกดปุ่ม Activate บนหน้าต่างควบคุมของโปรแกรมวิซวลเบสิก จังหวะที่ 2 และจังหวะต่อไปจนถึงจังหวะสุดท้ายก็กระทำเหมือนกัน พอได้จังหวะท่าทางการเดินทั้งหมดแล้ว ก็กดปุ่ม Send Value ค่าข้อมูลทั้งหมดในแต่ละท่าทางจะถูกส่งไปขับมอเตอร์ทุกตัวหมุนเป็นจังหวะการก้าวเดินตั้งแต่จังหวะแรกไปจนถึงจังหวะสุดท้ายที่ 16 (การอินพุตจังหวะสามารถทำได้ถึง 20 จังหวะซึ่งแล้วแต่จังหวะท่าทางที่ต้องการ ถ้าต้องการให้หุ่นยนต์ก้าวเดินได้ละเอียดมากขึ้น ก็เพิ่มจังหวะให้มากกว่าเดิม)

อย่างไรก็ตามจังหวะการเดินของหุ่นยนต์นั้นไม่เป็นไปตามรูปแบบท่าทางที่ปรับไว้เท่าใดนัก เนื่องจากพบปัญหาที่งานหมุนเซอร์โวมอเตอร์ในส่วนสะโพก M3 ทั้งซ้ายและขวา (ดังที่กล่าวแล้วในข้อ 2) ที่มีลักษณะหลวม ไม่แน่นเนื่องจากถูกดึงด้วยน้ำหนักของมอเตอร์ด้านล่างที่เหลื่อ การก้าวเดินจึงผิดแปลกไปจากเดิม และในส่วนการหมุนเอียงข้อเท้าของมอเตอร์ M6 ทั้งด้านซ้ายและขวา ซึ่งต้องรับน้ำหนักด้านบนของตัวหุ่นยนต์ทั้งหมด มีลักษณะหลวมเช่นเดียวกัน แต่โปรแกรมก็ยังควบคุมให้เดินได้แต่ไม่คึก

4.2 ผลการวัดค่ามุมเอียงจากเซนเซอร์

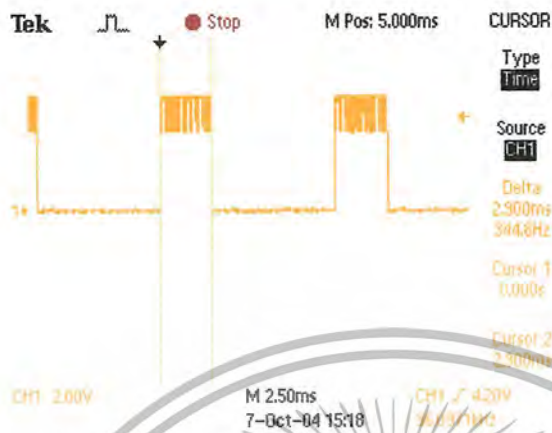
การเก็บค่าผลทดลองการเปลี่ยนแปลงของมุมทางด้านแกน X และแกน Y ของเซนเซอร์วัดมุมเอียง (Tilt Sensor) เทียบกับเวลา โดยการติดการเปลี่ยนแปลงของมุมทางด้านแกน X และแกน Y เป็นเปอร์เซ็นต์และนำค่าที่ได้มาเขียนกราฟเปรียบเทียบการเปลี่ยนแปลง ซึ่งได้ผลการทดลองดังรูปที่

4.7

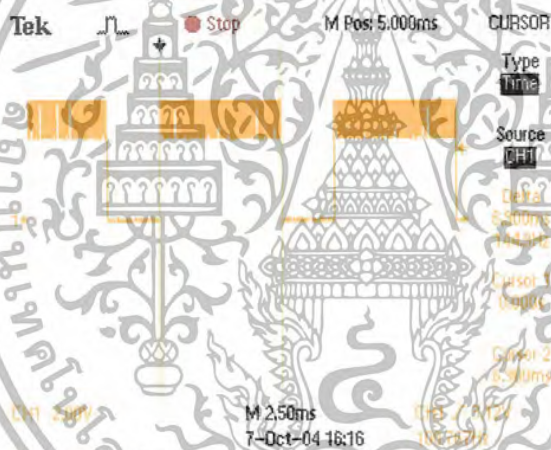


รูปที่ 4.7 ผลการวัดค่าที่แกน X ที่ +90 องศา ซึ่งได้การเปลี่ยนแปลงเวลาเท่ากับ 6.8 ms

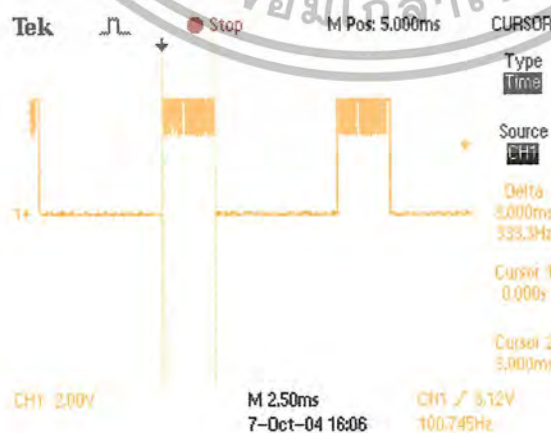
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 ผลการวัดค่าที่แกน X เท่ากับ -90 องศา ซึ่งได้การเปลี่ยนแปลงเวลาเท่ากับ 2.9 ms



รูปที่ 4.9 ผลการวัดค่าที่แกน Y เท่ากับ +90 องศา ซึ่งได้การเปลี่ยนแปลงเวลาเท่ากับ 6.9 ms



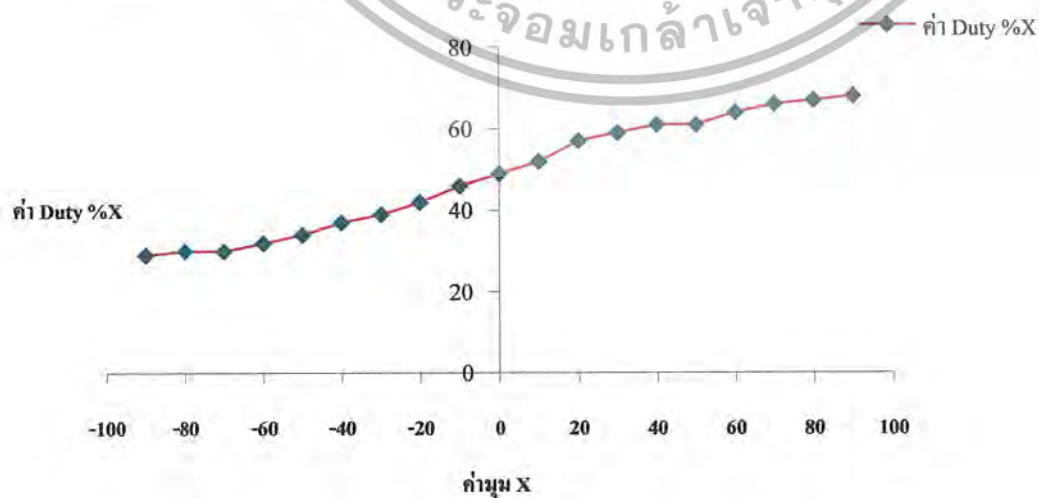
รูปที่ 4.10 ผลการวัดค่าที่แกน Y เท่ากับ -90 องศา ซึ่งได้การเปลี่ยนแปลงเวลาเท่ากับ 3.0 ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทำนองเดียวกัน ค่ามุมที่เหลื่อระหว่าง $(-90, +90)$ ก็สามารถวัดออกมาได้โดยใช้เครื่องมือวัด (Oscilloscope) แล้วนำค่าทั้งหมดมาเขียนกราฟเปรียบเทียบระหว่างค่ามุมและค่าความถี่ไซเคิลในแต่ละมุมเอียง ใ้คั้งนี้

ตารางที่ 3 แสดงค่ามุมแนวแกน X เทียบกับค่าการเปลี่ยนแปลงของ X

มุม X	TIME/DIV (ms) จากโรงงาน	1 คาบเวลา (ms)	การเปลี่ยนแปลงของ X Duty cycle (ms)	% X
90	2.5	10	6.9	69
80	2.5	10	6.8	68
70	2.5	10	6.7	67
60	2.5	10	6.6	66
50	2.5	10	6.4	64
40	2.5	10	6.1	61
30	2.5	10	5.9	59
20	2.5	10	5.7	57
10	2.5	10	5.2	52
0	2.5	10	5.0	50
-10	2.5	10	4.6	46
-20	2.5	10	4.2	42
-30	2.5	10	3.9	39
-40	2.5	10	3.7	37
-50	2.5	10	3.4	34
-60	2.5	10	3.2	32
-70	2.5	10	3.0	30
-80	2.5	10	3.0	30
-90	2.5	10	2.9	29

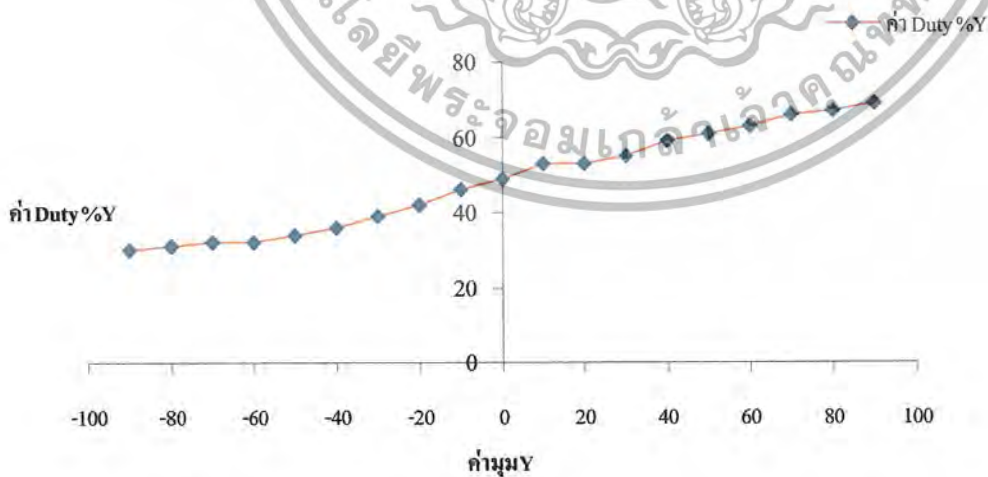


รูปที่ 4.11 แสดงกราฟค่ามุม X เทียบกับค่าการเปลี่ยนแปลงคาบเวลาของ X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4 แสดงค่ามุมแนวแกน Y เทียบกับค่าการเปลี่ยนแปลงของ Y

มุม Y	TIME/DIV (ms)จากโรงแจ้งงาน	1 คาบเวลา (ms)	การเปลี่ยนแปลงของ Y Duty cycle (ms)	%Y
90	2.5	10	6.9	69
80	2.5	10	6.7	67
70	2.5	10	6.6	66
60	2.5	10	6.3	63
50	2.5	10	6.1	61
40	2.5	10	5.9	59
30	2.5	10	5.5	55
20	2.5	10	5.3	53
10	2.5	10	5.3	53
0	2.5	10	4.9	49
-10	2.5	10	4.6	46
-20	2.5	10	4.2	42
-30	2.5	10	3.9	39
-40	2.5	10	3.6	36
-50	2.5	10	3.4	34
-60	2.5	10	3.2	32
-70	2.5	10	3.2	32
-80	2.5	10	3.1	31
-90	2.5	10	3.0	30



รูปที่ 4.12 แสดงกราฟค่ามุม Y เทียบกับค่าการเปลี่ยนแปลงคาบเวลาของ Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลข้อมูลจากการวัดค่ามุมเทียบกับค่าดิฟฟ์ไซเคิล โดยเครื่องมือวัด (Oscilloscope)ข้างต้น พบว่า ในแนวแกน X ค่ามุมเอียงของแกน X เอียงเท่ากับ 90 องศาที่ค่าดิฟฟ์ไซเคิล 6.9 ms และค่ามุมเอียงของแกน X เอียงเท่ากับ -90 องศาที่ค่าดิฟฟ์ไซเคิล 6.9 ms ช่วงนี้เองที่พบว่าการแปรค่ามุมเทียบกับค่าดิฟฟ์ไซเคิลมีลักษณะเชิงเส้น ซึ่งเกี่ยวเนื่องกับหัวข้อ 3.5.4 การคำนวณค่ามุมเอียงจากเซนเซอร์ ดังนั้นจึงสามารถใช้การประมาณแบบเชิงเส้นได้ สมการความสัมพันธ์ที่ได้คือ คือ $T_x = (\theta_x + 225)/4.5$ และในแนวแกน Y ก็เช่นเดียวกัน ช่วงค่ามุมเอียงของแกน Y เอียงเท่ากับ 86 องศาที่ค่าดิฟฟ์ไซเคิล 7.5 ms และค่ามุมเอียงของแกน Y เอียงเท่ากับ -88 องศาที่ค่าดิฟฟ์ไซเคิล 3.2 ms จะเป็นแบบเชิงเส้น สมการความสัมพันธ์ที่ได้คือ $T_y = (\theta_y + 209.418)/4.0465$ ซึ่งในการพัฒนาต่อสามารถใช้ค่ามุมทั้ง θ_x และ θ_y เป็นสัญญาณป้อนกลับเพื่อรักษาตำแหน่งของหุ่นยนต์ในการก้าวเดินได้



บทที่ 5

สรุปผลการดำเนินงาน

5.1 สรุปผลการดำเนินงาน

จากการศึกษาและทดลองวิจัยในโครงการนี้ ถูกแบ่งออกเป็นสองส่วนคือ ส่วนการออกแบบเชิงกล และส่วนการออกแบบวงจรควบคุม ในส่วนของการออกแบบเชิงกล โครงสร้างหุ่นยนต์ถูกออกแบบข้อต่อโดยใช้เซอร์โวมอเตอร์ที่มีแรงบิดเพียงพอที่จะขับเคลื่อนตัวหุ่นยนต์ให้ก้าวเดินไปได้ และในส่วนการออกแบบวงจรควบคุม ภาษาเบสิก และไมโครคอนโทรลเลอร์จะถูกนำมาใช้ติดต่อกับผู้ควบคุมหุ่นยนต์ผ่านการอินเตอร์เฟซทางหน้าจอคอมพิวเตอร์ โดยผู้ควบคุมสามารถทราบจังหวะการก้าวเดินของหุ่นยนต์ โดยเซนเซอร์วัดมุมเอียงที่ติดตั้งบนตัวหุ่นยนต์จะส่งข้อมูลในแนวแกน X และ Y บ่งบอกมุมโน้มเอียงเวลาหุ่นยนต์ก้าวเดินทางหน้าจอคอมพิวเตอร์พร้อมกับการควบคุมหุ่นยนต์นั้นผู้ใช้จะควบคุมผ่านหน้าจอคอมพิวเตอร์เช่นเดียวกัน การควบคุมหุ่นยนต์นั้นแบ่งออกเป็นสองส่วนคือ การควบคุมแบบบังคับเองโดยผู้ควบคุม (Manual Control) ซึ่งผู้ควบคุมจะบังคับให้เซอร์โวมอเตอร์แต่ละตัวหมุนไปตามลักษณะของรูปแบบการเดินของมนุษย์ และการควบคุมการเดินจากการสั่งงานทางโปรแกรมแบบอัตโนมัติ (ไม่ต้องมีผู้ควบคุม, Automatic Programming Control) ซึ่งผู้ควบคุมจะอินพุตข้อมูลของทุกจังหวะท่าทางการเดินให้โปรแกรมทำการรัน โปรแกรมให้ทุกจังหวะการก้าวเดินนั้นสัมพันธ์กันไปแบบอัตโนมัติ

อย่างไรก็ตาม การควบคุมการเดินของหุ่นยนต์ทั้งสองแบบเป็นแบบลูปเปิด (Open Loop Control) ที่มีลักษณะการควบคุมการเดินโดยผู้ควบคุมบังคับให้หุ่นยนต์เดินตามจังหวะท่าทางที่มองด้วยสายตาเท่านั้น ซึ่งจะมีปัญหาเกิดขึ้นตามทฤษฎีการควบคุมแบบลูปเปิดที่จะมีเสถียรภาพการควบคุมไม่ดีเท่ากับการควบคุมแบบลูปปิด (Close Loop Control) อย่างไรก็ตามการพัฒนาได้ออนั้น ข้อมูลจากเซนเซอร์วัดมุมเอียงจำเป็นต่อการควบคุมแบบลูปปิด โดยมุมแต่ละค่าที่ถูกวัดออกมาทั้งแนวแกน X และ Y จะเป็นสัญญาณป้อนกลับให้หุ่นยนต์รักษาตำแหน่งที่ก้าวเดินไว้อย่างมีเสถียรภาพได้

5.2 ปัญหาและอุปสรรค

การทดลองวิจัยโครงการหุ่นยนต์เดินสองขา พบเจอปัญหาและอุปสรรคดังนี้

1. ในการอินเตอร์เฟซระหว่างผู้ใช้และคอมพิวเตอร์ตัวจ่ายสัญญาณนาฬิกาจากภายนอก (Crystal) ใช้อัตราการรับส่งข้อมูลหรือบิตเรตระหว่างไมโครคอนโทรลเลอร์ (PIC) มีค่าเท่ากับ 2400 บิตต่อวินาทีซึ่งไม่ได้ตั้งไว้เท่ากับค่ามาตรฐาน (9600 บิตต่อวินาที) จึงเป็นผลให้เกิดความล่าช้าในการรับส่งข้อมูลขึ้น จึงทำให้หุ่นยนต์เคลื่อนที่ได้ไม่ทันกับที่ผู้ควบคุมสั่งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เซอร์โวมอเตอร์ที่เลือกใช้เกิดปัญหาที่พบเจอดังนี้

- อุณหภูมิจะสูงขึ้น เมื่อใช้ไประยะหนึ่ง (ไม่เหมาะกับการใช้งานที่เวลานาน)
- เกิดปัญหา เฟืองสึกหรือได้ง่ายมาก ถ้าความถี่แปรค่าไปจากช่วงทำงาน
- เกิดปัญหาการหมุนได้มุมที่ไม่ตรงกับที่ตั้งค่าไว้ (Error)
- ค่าแรงบิดในส่วนข้อเท้า และเข่ามีค่า 7.2 Kg-Cm ซึ่งไม่เพียงพอที่จะถือค้ำน้ำหนักให้หยุดค้างได้ ณ เวลานานๆ (อาจจะทำให้เฟืองครูดได้) การเพิ่มค่าแรงบิดหมายถึงการเพิ่มขนาดและราคาของเซอร์โวมอเตอร์

3. ค่อนข้างเสียเวลาในการเลือกโปรแกรมคำสั่งเพื่อคุมเซอร์โวมอเตอร์ระหว่าง Pulsout และ คำสั่ง Hpw โดย Pulsout ใช้ควบคุมความถี่ ส่วน Hpw ใช้ควบคุม Duty cycle ให้แปรค่าตามการสั่งงาน

4. จากการทดลองครั้งแรกที่ต้องการให้หุ่นยนต์ Maimo1 (เซอร์โวมอเตอร์ทั้ง 12 ตัว) เคลื่อนที่ได้ตามรูปแบบการก้าวเดินของมนุษย์ (Gait Pattern) แต่ภายหลังจากการสั่งงานควบคุมทางโปรแกรม ทั้งแบบมีผู้ควบคุม (Manual Control) และแบบไม่มีผู้ควบคุม (Automatic Programming Control) นั้นพบว่า หุ่นยนต์สามารถเคลื่อนที่ได้ทุกส่วนแต่ไม่สามารถเดินได้ เนื่องจากเกิดปัญหาทางการออกแบบเชิงกล คือ ความสูงของหุ่นยนต์ไม่เหมาะสมกับความกว้างของฝ่าเท้า และมีปัจจัยเรื่องน้ำหนักมาเกี่ยวข้องด้วย น้ำหนักของเซอร์โวมอเตอร์ในส่วนของการแบะเท้า (มอเตอร์ M1 ทั้งซ้ายและขวา)หนักไปทางด้านหลังของหุ่นยนต์จึงเป็นผลให้จุดศูนย์กลางมวลถูกดึงไปทางด้านหลังเวลาก้าวเดิน เป็นผลให้การสั่งการควบคุมไม่เป็นไปตามที่ต้องการให้หุ่นยนต์เดินได้ (หุ่นยนต์ล้ม)

ดังนั้นการแก้ไขปัญหานี้จึงทำการตัดเซอร์โวมอเตอร์ในส่วนนี้ออกไป (มอเตอร์ M1 ทั้งซ้ายและขวา) และทำการลดความสูงของหุ่นยนต์ลง พร้อมกับการเพิ่มพื้นที่สัมผัสพื้นในส่วนของเท้าหุ่นยนต์ ผลที่ได้คือหุ่นยนต์มีเสถียรภาพในการเดินมากขึ้น (จุดศูนย์กลางมวลของหุ่นยนต์ต่ำลง)

5. การควบคุมการเดินจากการสั่งงานทางโปรแกรมโดยมีผู้ควบคุม (Manual Control) ให้ก้าวเดินตามรูปแบบการเดินของมนุษย์ (Gait Pattern) นั้นเป็นการควบคุมแบบลูปเปิด (Open Loop Control) จังหวะการก้าวแต่ละก้าวนั้นผู้ควบคุมประมาณทางสายตาเท่านั้น ซึ่งก่อให้เกิดจังหวะการเดินที่ผิดไปจากเดิม (ไม่สามารถควบคุมให้เดินจังหวะเดิมได้ในแต่ละครั้ง) รวมกับปัญหาทางเซอร์โวมอเตอร์ที่ไม่หยุดหมุนเมื่อได้ตำแหน่งตามต้องการแล้ว (Error) แต่การควบคุมให้ก้าวเดินยังสามารถทำได้ แต่ไม่ถนัดนัก

6. การควบคุมการเดินจากการสั่งงานทางโปรแกรมแบบอัตโนมัติ (ไม่ต้องมีผู้ควบคุม, Automatic Programming Control) เช่นเดียวกันกับการควบคุมโดยผู้ควบคุม การปรับค่าตัวชี้ไขเคล็ดของมอเตอร์ ให้เป็นท่าทางการก้าวเดินที่ละจังหวะ นั้นเป็นการควบคุมแบบลูปเปิด (Open Loop Control) ถ้าเกิดความผิดพลาดจากสายตาผู้ควบคุมในการปรับจังหวะท่าทางของหุ่นยนต์ให้ก้าวเดินผิดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวัด หุ่นยนต์นั้นจะเดินได้ไม่ตรงกับความต้องการหรืออาจจะล้ม เนื่องจากจังหวัดทางที่คิดไป (โปรแกรมทำงานไปแล้วไม่สามารถแก้ไขจังหวัดที่คิดไปได้ จะต้องทำการอินพุตจังหวัดใหม่ซึ่งจะเสียเวลามาก)

จากผลการทดลองที่ได้มา ปัญหาและอุปสรรคที่เกิดขึ้นส่วนใหญ่่นั้นมาจากปัญหาการออกแบบเชิงกล เนื่องจากการออกแบบหุ่นยนต์ Maimo1 ในทอมแรกนั้น ไม่ได้คำนึงถึงผลกระทบจากปัญหาเชิงกลเท่าที่ควรจึงส่งผลกระทบต่อการทำงานโดยรวม และค่อนข้างใช้เวลามากไปกับการพัฒนาโปรแกรมควบคุมเซอร์โวมอเตอร์ และเซนเซอร์วัดมุมเอียง อย่างไรก็ตามผลการทดลองที่ได้มานั้นไม่ตรงตามเป้าหมายที่ตั้งไว้เท่าใดนัก แต่ก็สร้างความพึงพอใจแก่ทีมงานของเราอย่างมาก

5.3 แนวทางการพัฒนาต่อ

การพัฒนาหุ่นยนต์ Maimo1 ให้สามารถก้าวเดินได้อย่างมีประสิทธิภาพ และคล้ายคลึงกับการก้าวเดินของมนุษย์นั้นควรที่จะแก้ไขข้อเสียต่างๆที่เกิดขึ้นในหุ่นยนต์ดังนี้

1. ในส่วนของโครงสร้างหุ่นยนต์ ความแข็งแรงของข้อต่อควรที่จะถูกปรับปรุงให้ยืดหยุ่นมากกว่าเดิม โดยการออกแบบโครงสร้างใหม่โดยคำนึงถึงเรื่องความสมดุลของเซอร์โวมอเตอร์ที่จัดวางบนตัวหุ่นยนต์ ซึ่งจะมีปัญหาเรื่องความสูงและน้ำหนักมาเกี่ยวข้อง
2. ปรับปรุงโปรแกรมคำสั่งที่จะควบคุมเซอร์โวมอเตอร์ทั้งหมด ควรจะสั่งงานให้เซอร์โวมอเตอร์หมุนพร้อมกันทั้งหมดในลักษณะก้าวเดินได้ ไม่ใช่สั่งงานให้เซอร์โวมอเตอร์หมุนทีละตัว เพราะถ้าเช่นนั้นการก้าวเดินของหุ่นยนต์จะไม่ใช้รูปแบบการก้าวเดินของมนุษย์
3. ในส่วนของเซอร์โวมอเตอร์ แรงบิดในส่วนสะโพกที่ใช้ในการก้าวเดิน ข้อเท้า และเข่า นั้นควรที่จะเพิ่มมากกว่าเดิม เพื่อสามารถทำให้หุ่นยนต์ค้างตำแหน่ง เช่น ในท่าทางการยืนขาเดียวได้อย่างสมดุลมากขึ้น แต่อาจทำให้เซอร์โวมอเตอร์ในส่วนนี้ใหญ่ขึ้น และราคาสูงกว่าเดิมมาก
4. ควรที่จะปรับปรุงหุ่นยนต์ให้สามารถก้าวเดินได้อิสระ เคลื่อนที่ซ้ายขวาได้ ไม่เฉพาะเดินตรงอย่างเดียวเท่านั้น โดยการเพิ่มเซอร์โวมอเตอร์ที่ใช้หมุนเบาะเท้าออกทางด้านข้าง (M1 ทั้งซ้ายและขวา) แต่เวลาหุ่นยนต์เดินตรง ควรที่จะล็อคตำแหน่งไม่ให้เซอร์โวมอเตอร์ส่วนนี้หมุนไปรบกวนท่าทางการเดินตรงของหุ่นยนต์ รวมไปถึงการก้าวเดินบนทางลาดชันได้อย่างมีประสิทธิภาพ
5. การควบคุมหุ่นยนต์ Maimo1 นี้ควรที่จะเป็นแบบลูปปิด (Close Loop Control) เพื่อให้หุ่นยนต์รักษาตำแหน่งการหมุนของเซอร์โวมอเตอร์ได้ด้วยตัวมันเองโดยการศึกษาทฤษฎีการควบคุม เช่น ทฤษฎีจุดรวมโมเมนต์ศูนย์ (Zero Moment Point, ZMP) เพื่อใช้ควบคุมหุ่นยนต์ให้ก้าวเดินเชิงพลวัต (Dynamic Walking) ได้ดีขึ้น โดยข้อมูลที่ใช้ได้แก่จุดศูนย์กลางมวลของหุ่นยนต์ (CG) ซึ่งคำนวณได้จากเรขาคณิตวิเคราะห์จากการโน้มเอียงของตัวหุ่นยนต์ (ผลมุมเอียงแนวแกน X และ Y จากเซนเซอร์วัดมุมเอียง) หรือคำนวณจากโครงสร้างเชิงกลตามหลักกลศาสตร์

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ทางคณะผู้จัดทำขอกราบขอบพระคุณรองศาสตราจารย์ ดร. จงกล งามวิวิทย์ สำหรับห้องโปรเจ็ค (Process Room) หากไม่มีห้องนี้ ทางคณะผู้จัดทำคงทำงานอย่างยากลำบากกว่าเดิมมาก และขอขอบคุณอาจารย์ที่ช่วยดักเตือน สอนแบบอย่างการทำงานที่ดีให้แก่ทางคณะผู้จัดทำ ทางคณะผู้จัดทำจะนำสิ่งที่อาจารย์ได้สอนไปปฏิบัติงานให้ดีที่สุด

ขอขอบพระคุณ อาจารย์ คอน อิศรากร อาจารย์ที่ปรึกษา ที่ช่วยติดตามความก้าวหน้าของงานอยู่เสมอ ปัญหาและอุปสรรคต่างๆ ที่เกิดขึ้นนั้นช่วยสอนให้ทราบว่า ถ้าพึงการทำงานด้วยตนเองเพียงอย่างเดียวจะไม่ก่อให้เกิดประสิทธิภาพในการทำงาน ไม่เกิดการงานเป็นทีม ซึ่งนำไปสู่ความขัดแย้งในการทำงานที่เพิ่มขึ้น ทางคณะผู้จัดทำขอรับรองว่า การทำงานให้บรรลุเป้าหมายในโครงการนี้อาจไม่เท่าที่ควรในสายตาของอาจารย์ แต่โครงการนี้ก็ช่วยให้ทางคณะผู้จัดทำได้ตระหนักถึงปัญหาในการทำงานที่เกิดขึ้น และจะนำไปปรับปรุงแก้ไขคุณภาพการทำงานให้ดีขึ้นมากกว่านี้

ขอขอบพระคุณ อาจารย์ ถาวร เบลญจนาสุทธิ อาจารย์ที่ปรึกษาร่วม ในส่วนของเรื่องการพัฒนาโปรแกรมควบคุม ทำให้คณะผู้จัดทำสามารถมองเห็นแนวทางการพัฒนาที่กระจ่างขึ้น

ขอขอบคุณภาควิชาวิศวกรรมระบบควบคุม สำหรับเอื้อเฟื้ออุปกรณ์และเครื่องมือทั้งหมด และพี่สไตร์ที่ช่วยเหลือในเรื่องการให้ยืมอุปกรณ์

ขอขอบคุณพี่ สองเมือง นันทขว้าง ในส่วนเรื่องการใช้ห้องโปรเจ็ค ซึ่งทางคณะผู้จัดทำทราบดีว่าคงจะรบกวนเวลาการทำงานของท่านอย่างมากมาย ทางคณะผู้จัดทำขอขอบคุณพี่มากครับ

ขอขอบคุณพี่ วราวุธ พรินทรากุล อย่างสุดซึ่งที่คอยให้คำปรึกษาแก้ไขปัญหา ตลอดจนความช่วยเหลือในทุกๆ สิ่งที่ได้รับมา ทางคณะผู้จัดทำรู้สึกซาบซึ้งใจมากครับ

สุดท้ายนี้ ทางคณะผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ของทางคณะผู้จัดทำที่ได้ให้ความสนับสนุนตลอดมา จนทำให้ทางคณะผู้จัดทำได้ศึกษาเล่าเรียนจนถึงทุกวันนี้ และขอขอบคุณเพื่อนๆ ที่คอยให้คำปรึกษาการแก้ไขปัญหาที่เกิดขึ้น

ทั้งหมดนี้ที่ทำให้ทางคณะผู้จัดทำมีกำลังใจในการทำงาน จนทำให้ปริญญาานิพนธ์ฉบับนี้เสร็จสมบูรณ์ ทางคณะผู้จัดทำขอกราบขอบพระคุณอย่างสุดซึ่ง

เอกสารอ้างอิง

- [1] ผ.ศ. โยชิน เปรมปราณีรัชต์, “ ระบบเซอร์โวและอิเล็กทรอนิกส์คอนโทรลมอเตอร์ ” , พิมพ์ครั้งที่ 1 , Japan International Cooperation Agency (JICA) , พ.ศ. 2533
- [2] คอน อิสรากร, ชาญวัฒน์ รัชตเสรีกุล, ธงชัย มณฑาทิพย์กุล, “ ขากล ” , ปรินญาณิพนธ์สำหรับปรินญาวิศวกรรมศาสตรบัณฑิต , สาขาอิเล็กทรอนิกส์ , สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , พ.ศ. 2541
- [3] Jochen Zimmermann, “ Balancing of Bipedal Robot Using Force Feedback ” , Diplomarbeit, Prof.Dipl.-Inf.H.J.Unkelbach : University of Applied Science Koblez : Department for Electronic and Information Engineering, Prof.Dr.rer.nat.habil.T.Braunl : University of Western Australia : School of Electrical, Electronic and Computer Engineering, 2003
- [4] วชิรินทร์ เคารพ, “ คู่มือการใช้งาน SERVO MOTOR ” , พิมพ์ครั้งที่ 1 , บริษัททีทีทีจำกัด , พ.ศ. 2546
- [5] Ed Ayyappa, MS, CPO, “ Normal Human Locomotion, Part 1: Basic Concepts and Terminology ” , Journal of Prosthetics and Orthotics 1997, Vol 9, Num 1, p.10.
URL: http://www.oandp.org/jpo/library/1997_01_010.asp
- [6] Edmond Ayyappa, CPO, “ Normal Human Locomotion, Part 2: Motion, Ground Reaction Force and Muscle Activity ” , Journal of Prosthetics and Orthotics 1997; Vol 9, Num 2, p 49, URL: http://www.oandp.org/jpo/library/1997_02_049.asp
- [7] อรรถพล บุญยะ โภคา, “ เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม ” , พิมพ์ครั้งที่ 2 , บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด
- [8] กฤษดา ใจเย็น, “ เรียนรู้และใช้งาน PICBASIC PRO คอมพิวเตอร์ ” , บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด
- [9] พัฒนพงศ์ พันธะติวงศ์, “ การเขียนโปรแกรมคอมพิวเตอร์ด้วย Microsoft Visual Basic V.6.0 ” , บริษัท ซีอีคยูเคชั่น จำกัด (มหาชน), พ.ศ. 2545
- [10] ฉัฐพล วงศ์สุนทรชัย, “ เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ PIC16F87X ” , บริษัท อิน โนเวตีฟ เอ็กเพอริเมนต์ จำกัด
- [11] “ Improved, Ultra Low Noise, Dual Axis Accelometer with Digital Outputs ” : MEMSIC INC.: www.memsic.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MICROCHIP

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

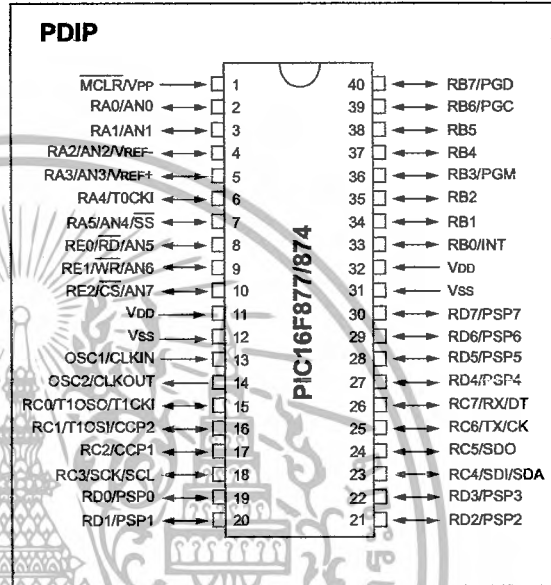
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)



Improved, Ultra Low Noise $\pm 1 g$ Dual Axis Accelerometer with Digital Outputs

MXD2020GL/HL
MXD2020ML/NL

FEATURES

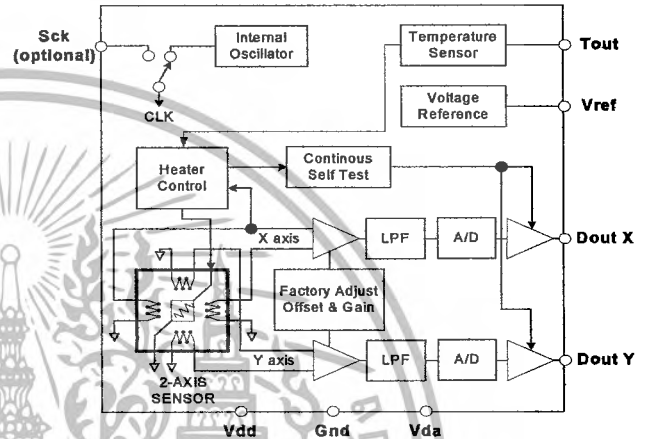
- Resolution better than 1 milli-g
- Dual axis accelerometer fabricated on a monolithic CMOS IC
- On chip mixed mode signal processing
- No moving parts
- 50,000 g shock survival rating
- 17 Hz bandwidth expandable to >160 Hz
- 3.0V to 5.25V single supply continuous operation
- Continuous self test
- Independent axis programmability (special order)
- Compensated for Sensitivity over temperature
- Ultra low initial Zero-g Offset

APPLICATIONS

- Automotive** – Vehicle Security/Vehicle stability control/
Headlight Angle Control/Tilt Sensing
- Security** – Gas Line/Elevator/Fatigue Sensing
- Information Appliances** – Computer Peripherals/PDA's/Mouse
Smart Pens/Cell Phones
- Gaming** – Joystick/RF Interface/Menu Selection/Tilt Sensing
- GPS** – electronic Compass tilt Correction
- Consumer** – LCD projectors, pedometers, blood pressure
Monitor, digital cameras

GENERAL DESCRIPTION

The MXD2020GL/HL/ML/NL is a low cost, dual axis accelerometer fabricated on a standard, submicron CMOS process. It is a complete sensing system with on-chip mixed mode signal processing. The MXD2020GL/HL/ML/NL measures acceleration with a full-scale range of $\pm 1 g$ and a sensitivity of 20%/g. It can measure both dynamic acceleration (e.g. vibration) and static acceleration (e.g. gravity). The MXD2020GL/HL/ML/NL design is based on heat convection and requires no solid proof mass. This eliminates stiction and particle problems associated with competitive devices and provides shock survival of 50,000 g, leading to significantly lower failure rate and lower loss due to handling during assembly.



**MXD2020GL/HL/ML/NL FUNCTIONAL BLOCK
DIAGRAM**

The MXD2020GL/HL/ML/NL provides two digital outputs that are set to 50% duty cycle at zero g acceleration. The outputs are digital with duty cycles (ratio of pulse width to period) that are proportional to acceleration. The duty cycle outputs can be directly interfaced to a micro-processor.

The typical noise floor is $0.2 \text{ mg}/\sqrt{\text{Hz}}$ allowing signals below 1 milli-g to be resolved at 1 Hz bandwidth. The MXD2020GL/HL/ML/NL is packaged in a hermetically sealed LCC surface mount package (5 mm x 5 mm x 2 mm height) and is operational over a -40°C to 105°C (M/NL) and 0°C to 70°C (G/HL) temperature range.

Information furnished by MEMSIC is believed to be accurate and reliable. However, no responsibility is assumed by MEMSIC for its use, nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of MEMSIC.

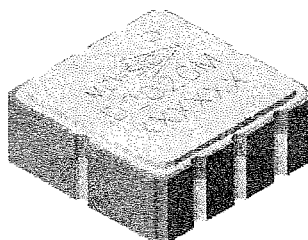
©MEMSIC, Inc.
800 Turnpike St., Suite 202, North Andover, MA 01845
Tel: 978.738.0900 Fax: 978.738.0196
www.memsic.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Supply Voltage (V_{DD}, V_{DA})-0.5 to +7.0V
 Storage Temperature-65°C to +150°C
 Acceleration50,000 g

*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; the functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.



Pin Description: LCC-8 Package

Pin	Name	Description
1	T _{OUT}	Temperature (Analog Voltage)
2	D _{OUTY}	Y-Axis Acceleration Digital Signal
3	Gnd	Ground
4	V _{DA}	Analog Supply Voltage
5	D _{OUTX}	X-Axis Acceleration Digital Signal
6	V _{ref}	2.5V Reference
7	Sck	Optional External Clock
8	V _{DD}	Digital Supply Voltage

THEORY OF OPERATION

The MEMSIC device is a complete dual-axis acceleration measurement system fabricated on a monolithic CMOS IC process. The device operation is based on heat transfer by natural convection and operates like other accelerometers having a proof mass. The stationary element, or 'proof mass', in the MEMSIC sensor is a gas.

A single heat source, centered in the silicon chip is suspended across a cavity. Equally spaced aluminum/polysilicon thermopiles (groups of thermocouples) are located equidistantly on all four sides of the heat source (dual axis). Under zero acceleration, a temperature gradient is symmetrical about the heat source, so that the temperature is the same at all four thermopiles, causing them to output the same voltage.

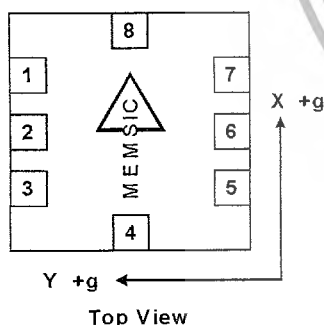
Acceleration in any direction will disturb the temperature profile, due to free convection heat transfer, causing it to be asymmetrical. The temperature, and hence voltage output of the four thermopiles will then be different. The differential voltage at the thermopile outputs is directly proportional to the acceleration. There are two identical acceleration signal paths on the accelerometer, one to measure acceleration in the x-axis and one to measure acceleration in the y-axis. Please visit the MEMSIC website at www.memsic.com for a picture/graphic description of the free convection heat transfer principle.

Ordering Guide

Model	Package Style	Digital Output	Temperature Range
MXD2020GL	LCC - 8	100 Hz	0 to 70°C
MXD2020HL	LCC - 8	400Hz	0 to 70°C
MXD2020ML	LCC - 8	100 Hz	-40 to 105°
MXD2020NL	LCC - 8	400 Hz	-40 to 105°

All parts are shipped in tape and reel packaging.

Caution: ESD (electrostatic discharge) sensitive device.



Note: The MEMSIC logo's arrow indicates the +X sensing direction of the device. The +Y sensing direction is rotated 90° away from the +X direction following the right-hand rule. Small circle indicates pin one(1).

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

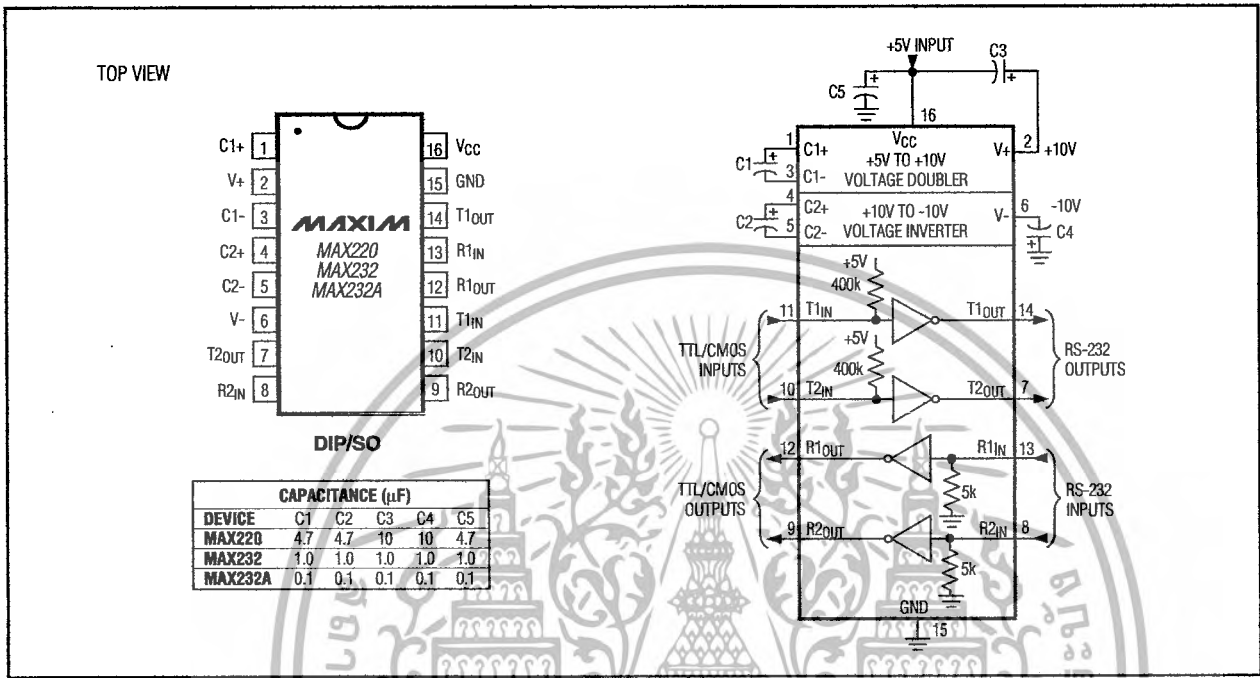


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

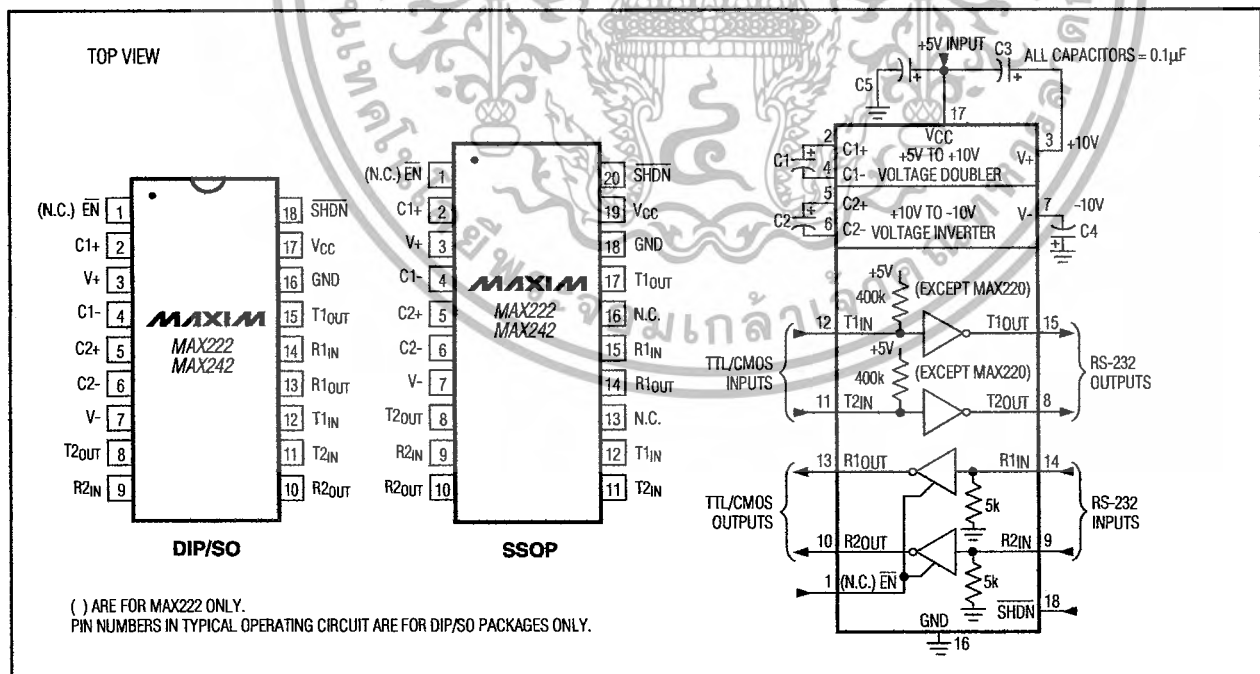


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

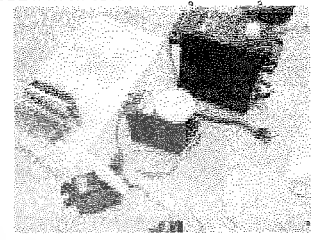
SERVO MOTOR

ในการทำเครื่องกล, ทำหุ่นยนต์, รถหุ่นยนต์, แขนกล ด้วย SERVO MOTOR จาก อิทีที

คุณ... สามารถทำหรือสร้างระบบเครื่องกลต่างๆ เช่น กล, หุ่นยนต์ ต่างๆ ได้เองโดยง่าย ด้วยระบบ SERVO MOTOR ไม่ต้องออกแบบในส่วนตัวเกี่ยวกับ ส่วนควบคุมต่างๆ ใช้สัญญาณเพียง 1 เส้น ในการควบคุม รวมทั้งในเรื่องของแรงบิด แรงยก โดยมีให้เลือกได้ตั้งแต่ 2.3 Kg-CM ถึง 15 Kg-CM ไม่ต้องกลัวเรื่องมีแรงบิดแรงยกไม่พออีกแล้ว SERVO MOTOR ... คือ มอเตอร์ไฟฟ้ากระแสตรง DC MOTOR ที่ถูกประกอบรวมด้วยชุดเกียร์ และส่วนควบคุมต่างๆ ไว้ในโมดูลเดียวกัน โดยจะมีสัญญาณใช้งาน 1 เส้น และอีก 2 เส้น เป็น VCC และ GND เท่านั้น ซึ่งสามารถควบคุมให้ตัว SERVO MOTOR หมุนซ้าย หรือ ขวาได้ +90 องศา - 90 องศา (180 องศา) โดยสามารถสั่งงานในการหมุนให้หมุนไปได้ตามองศาต่างๆ ที่ต้องการ ได้ด้วยตัวของ SERVO MOTOR เอง เช่น ต้องการหมุน 1 องศา หรือ 15 องศา ก็ได้ ไม่ต้องมีส่วนควบคุม หรือ SENSOR ใดๆ กลับมาตรวจสอบอีก ทำให้ง่าย และสะดวกในการใ้ภาหุ้เข้าไปประยุกต์ใช้งานต่างๆ ได้จริง...

- การควบคุมการทำงานของ SERVO MOTOR ทำได้โดยการป้อนสัญญาณความกว้างของพัลส์ให้กับตัว SERVO MOTOR ซึ่งจะได้ทิศทางการหมุนและตำแหน่งของการหมุน
- สามารถใช้งานกับไฟ DC ได้ 4 - 6 VOLT, หมุนได้ 180 องศา และสามารถปรับแต่งตัว SERVO MOTOR ให้สามารถหมุนได้รอบตัวก็ได้ เช่น ทำเป็นชุดล้อรถหุ่นยนต์
- ขั้วต่อจะเป็นแบบมาตรฐาน ขั้ว JR TYPE
- มีจำหน่ายด้วยกันอยู่ 5 แบบ 5 ขนาด ด้วยกัน เป็นของบริษัท GWS และอีก 1 แบบ 1 รุ่น ของบริษัท HITEC HS-232

- ◆ รุ่น MICRO/STD/JR จะเป็นรุ่นที่มีขนาดเล็กน้ำหนักเบา ราคา 790.-
- ◆ รุ่น S03N/STD/JR จะเป็นรุ่นมาตรฐานใช้งานทั่วไป ราคา 640.-
- ◆ รุ่น S03T/STD/JR จะเป็นรุ่นมาตรฐานแต่จะมีแรงบิดสูงกว่า S03N ราคา 640.-
- ◆ รุ่น S04/BBM/JR จะเป็นรุ่นที่ใหญ่ขึ้นมีแรงบิด 13 Kg-CM ราคา 990.-
- ◆ รุ่น S666/STD/JR จะเป็นรุ่นที่ใหญ่ที่สุดมีแรงบิด 15Kg-CM ราคา 1,490.-
- ◆ รุ่น HS-322 จะเป็นรุ่นมาตรฐานของ HITEC มีแรงบิด 3.5 Kg-CM ราคา 750.-



Model	STD	BBM	Size (LxWxH) mm/in	Weight		Speed (sec/60°)	4.8 V Torque		Speed (sec/60°)	6 V Torque	
				g	oz		Kg-cm	Oz-in		Kg-cm	Oz-in
Micro	X		28x14x29.8 1.1x0.55x1.17	18	0.63	0.16	1.8	25	0.13	2.30	32
S03N	X		39.5x20.0x35.6 1.56x0.79x1.40	41	1.44	0.23	2.40	47	0.18	4.00	56
S03T	X		39.5x20.0x39.6 1.56x0.79x1.56	46.0	1.62	0.33	7.20	100	0.27	8.00	111
S04		X	54.4x26.5x51.5 2.14x1.04x2.03	114	4.00	0.25	10.00	138.88	0.20	13.00	180.5
S666	X		63.0x32.0x61.6 2.48x1.26x2.43	142.4	5.02	0.28	13.00	181	0.22	15.00	208
HS-322	X		40x20x36.5 1.57x0.78x1.43	43	1.51	0.19	3	41.66	0.15	3.5	48.6

STD = Oiliness Bearingg BBM = Ball Bearings Metal Gear

วิธีการใช้งาน SERVO MOTOR พร้อมตัวอย่างโปรแกรม จากที่ทีมงาน อิทีที ราคา 40 บาท

4 ... คู่มืออธิบายการทำงานของตัว SERVO MOTOR การควบคุม, การทำให้ตัว SERVO MOTOR หมุนได้ 360 องศา พร้อมตัวอย่างการควบคุมโปรแกรมใช้งานกับ CPU บอร์ดต่างๆ ของทาง อิทีที เช่น

- การควบคุม SERVO MOTOR ด้วย IC LM555
- การควบคุม SERVO MOTOR ด้วย CPU PIC
- การควบคุม SERVO MOTOR ด้วย CPU AVR
- การควบคุม SERVO MOTOR ด้วย CPU MCS-51
- การควบคุม SERVO MOTOR ด้วย CPU BASIC STAMP
- การควบคุม SERVO MOTOR ด้วย CPU Z80



พร้อมทั้งคุณสมบัติต่างๆ ของ SERVO MOTOR แบบต่างๆ ที่จะนำเข้ามาให้ใช้งานอีกหลายรุ่น ด้วยคู่มือนี้คุณจะสามารถใช้ SERVO MOTOR ได้เป็นอย่างดี