สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาหุ่นยนต์อุตสาหกรรม 5 แกน
THE SCORBOT - ER III

T 0 8 3 7 2 0

ธีระพงษ์ ดำริห์ชอบ
ณัฐพล ศรีบุญเรือง

เลขหมู่.....
เลขทะเบียน.....83720
วัน,เดือน,ปี.....15 ก.ย. 2551
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาหุ่นยนต์อุตสาหกรรม 5 แกน
THE SCORBOT - ER III

โดย

ธีระพงษ์ ดำริห์ชอบ

ณัฐพล ศรีบุญเรือง

อาจารย์ที่ปรึกษา

ผศ.ดร. อุนนัต พิณโสภณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2551

ภาควิชา วิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การพัฒนาหุ่นยนต์อุสาหกรรม 5 แกน**

THE SCORBOT-ER III

ผู้จัดทำ

1.ธีระพงษ์ ดำริห์ชอบ รหัสประจำตัว 47015363

2. ณัฐพล ศรีบุญเรือง รหัสประจำตัว 47015393

_____อาจารย์ที่ปรึกษา

(ผศ.ดร.อุนนัต พิณโสภณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาหุ่นยนต์อุตสาหกรรม 5 แกน

(The SCORBOT – ER III)

นายธีระพงษ์ ดำริห์ชอบ 47015363
นายณัฐพล ศรีบุญเรือง 47015393
ผศ.ดร.อุนนัต พิณโสภณ อาจารย์ที่ปรึกษา

บทคัดย่อ

โครงการนี้เป็นการศึกษาถึงการควบคุมความเร็วของแขนกลรุ่น SCORBOT – ER III โดยมีวัตถุประสงค์เพื่อสร้างตัวควบคุมมือถือและตัวควบคุมอัตโนมัติ ให้แขนกลเคลื่อนที่ตามเงื่อนไขตามที่กำหนด การทำงานของแขนกลเริ่มจากไมโครคอนโทรลเลอร์ในตระกูล AVR ตัวที่หนึ่งรับค่าสัญญาณพัลส์ จาก Encoder และส่งค่าที่นับได้ต่อไปยังเครื่องคอมพิวเตอร์ หลังจากนั้นเครื่องคอมพิวเตอร์จะทำการเฉลี่ยความเร็ว โดยใช้วิธีเปรียบเทียบสัดส่วนของจำนวนค่าสัญญาณพัลส์นับได้จากมอเตอร์แต่ละตัว ให้เป็นค่าเปอร์เซ็นต์ PWM แล้วส่งสัญญาณให้กับไมโครคอนโทรลเลอร์ตัวที่สอง เมื่อได้รับค่า PWM แล้วไมโครคอนโทรลเลอร์ตัวที่สองจะทำงานตามคำสั่งโดยทำการสร้างสัญญาณ PWM ให้เป็นไปตามที่กำหนดจากเครื่องคอมพิวเตอร์ ในการคิดคำนวณของเครื่องคอมพิวเตอร์เพื่อให้แขนกลเคลื่อนที่นั้นจะคิดเป็นช่วงๆ หรือแบบจุดต่อจุด จากการทดลองพบว่ายิ่งแบ่งช่วงของการเคลื่อนที่มากยิ่งขึ้นจะทำให้การเคลื่อนที่ยิ่งราบเรียบยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCORBOT – ER III

Mr.Teerapong Dumrishob 47015363

Mr.Nattapon Sriboonruang 47015393

Dr.Unnut Pinsopon Advisor

Abstract

This project is the study of velocity control of SCORBOT – ER III robot arms. The purpose is to build the manual and automatic controllers of the robot. AVR microcontroller is used to receive the pulse signal from the encoder of each joint and transmit the signal to the computer. The control program is the computer will calculate the velocity command for each joint by converting the pulse signal to PWM control signal. The PWM control signal is then sent to the another AVR microcontroller for controlling the robot. From the experiments, the robot perform better tracking performances, if it is trained with larger numbers of training locations.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จะไม่สามารถเสร็จลุล่วงไปได้ด้วยดี ถ้าหากปราศจากคำแนะนำ และความร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน คือ ผศ.ดร.อุนนัต พิณโสภณ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เสร็จลงได้ด้วยดี ที่ให้ความเอาใจใส่ และช่วยเหลือมาโดยตลอด ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ท้ายที่สุด คณะผู้จัดทำขอขอบพระคุณ คณาจารย์และบุคลากร ของภาคเครื่องกล ที่ให้การปรึกษาและความช่วยเหลือมาโดยตลอด ขอขอบพระคุณมา ณ ที่นี้

นายธีระพงษ์ ดำริห์ชอบ

นายณัฐพล ศรีบุญเรือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูปภาพ	VII
บทที่ 1 บทนำ	
1.1 ประวัติความเป็นมาของหุ่นยนต์	1
1.2 วัตถุประสงค์	2
1.3 ขอบข่ายของงาน	3
1.4 แนวทางการดำเนินงาน	3
บทที่ 2 รายละเอียดอุปกรณ์ที่ใช้ในแขนกล	
2.1 รายละเอียดทั่วไปของแขนกล	4
2.2 คุณสมบัติของ SCORBOT-ER III	5
2.3 พื้นที่ใช้ในการทำงานของ SCORBOT–ER III	7
2.4 อุปกรณ์และทฤษฎีที่ใช้ใน SCORBOT- ER III	8
บทที่ 3 การออกแบบระบบควบคุมใน SCORBOT ER III	
3.1 ตัวควบคุมแบบมือถือ	12
3.2 ตัวควบคุมแบบอัตโนมัติ	13
3.3 พื้นฐานระบบควบคุมหุ่นยนต์	18
3.4 การออกแบบโปรแกรมการสั่งงาน	19
3.5 การออกแบบโปรแกรมคำนวนความเร็วที่ใช้ในแขนกล	20
3.6 PWM (Pulse Width Modulation)	23
บทที่ 4 รูปแบบและผลการทดลอง	
4.1 การทดสอบควบคุมการเคลื่นที่แบบสามมิติอย่างง่าย	26
4.2 การทดสอบควบคุมการเคลื่อนที่ของแขนกลในแนวระนาบเป็นเส้นตรง	29
4.3 การทดสอบควบคุมการเคลื่อนที่ของแขนกลในแนวระนาบเป็นเส้นโค้ง	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 วิเคราะห์และสรุปผลการทดลอง	
5.1 สรุปผลการทดลอง	36
5.2 แนวทางในการแก้ไข	36
บรรณานุกรม	37
ภาคผนวก ก ส่วนประกอบของหุ่นยนต์ SCORBOT- ER III	38
ภาคผนวก ข วิชวลเบสิกซอร์สโค้ด (Visual Basic Source Code)	41
ภาคผนวก ค ซีซอร์สโค้ด(C Source Code)	49
ภาคผนวก ง Data Sheet L298N	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 อธิบายโครงสร้างของแขนกล 5
ตารางที่ 2.2 อธิบายการเคลื่อนที่ของข้อต่อ 6
ตารางที่ 2.3 คุณสมบัติของมอเตอร์ 8
ตารางที่ 4.1 แสดงค่าผิดพลาดที่ได้จากการคำนวน 26
ตารางที่ 4.2 แสดงค่าXและYที่ได้จากการเทียบกับเส้นที่กำหนด 31
ตารางที่ 4.3 แสดงค่า X และ Y ที่ได้จากการวัดที่แขนกลเขียน 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 SCORBOT – ER III ที่ใช้ในการทดลอง 4
รูปที่ 2.2 ข้อต่อ และ Links ของแขนกลแบบ SCORBOT – ER III 6
รูปที่ 2.3 SCORBOT – ER III ด้านบน 7
รูปที่ 2.4 SCORBOT – ER III ด้านข้าง 8
รูปที่ 2.5 แสดงมอเตอร์ที่ใช้ในตัวหุ่น 9
รูปที่ 2.6 ตัวตรวจจับสัญญาณ 10
รูปที่ 2.7 limit sensor 10
รูปที่ 2.8 วงจรของ limit sensor 11
รูปที่ 3.1 ชุดบังคับหุ่นยนต์แบบมือถือ 12
รูปที่ 3.2 ไอซี L298N 13
รูปที่ 3.3 การจัดการขาของไอซี L298N 13
รูปที่ 3.4 การต่อวงจรใช้งานของ L298N 14
รูปที่ 3.5 แผงวงจรไมโครคอนโทรลเลอร์ (Microcontroller) 16
รูปที่ 3.6 ตัวอย่างการควบคุมแบบป้อนกลับ 18
รูปที่ 3.7 แผนผังลำดับการทำงานของโปรแกรมภาษาซีในไมโครคอนโทรลเลอร์ 19
รูปที่ 3.8 หน้าตาโปรแกรมวิชวลเบสิกที่ใช้กับหุ่นยนต์ 21
รูปที่ 3.9 แผนผังลำดับการทำงานของโปรแกรม Visual Basic (VB) 22
รูปที่ 3.10 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆกัน 25
รูปที่ 4.1 กราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งกับค่าผิดพลาดในแกน X, Y, Z 28
รูปที่ 4.2 แขนกลเขียนเส้นตรงโดยที่กำหนดจุด 2 จุด 29
รูปที่ 4.3 แขนกลเขียนเส้นตรงโดยที่กำหนดจุด 3 จุด 29
รูปที่ 4.4 แขนกลเขียนเส้นตรงโดยที่กำหนดจุด 5 จุด 30
รูปที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่างX และ Y เป็นเส้นตรงที่แขนกลเขียนขึ้น 31
รูปที่ 4.6 แขนกลเขียนเส้นพาราโบลาโดยที่กำหนดจุด 2 จุด 32
รูปที่ 4.7 แขนกลเขียนเส้นพาราโบลาโดยที่กำหนดจุด 3 จุด 33
รูปที่ 4.8 แขนกลเขียนเส้นพาราโบลาโดยที่กำหนดจุด 5 จุด 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างX และ Y เป็นเส้นพาราโบลาที่แขนกลเขียนขึ้น 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ประวัติความเป็นมาของหุ่นยนต์

หุ่นยนต์เป็นสิ่งประดิษฐ์ที่สร้างขึ้นมาสำหรับการช่วยการทำงานของมนุษย์ หุ่นยนต์ที่สร้างขึ้นในระยะแรก (ประมาณคริศตวรรษที่ 18) มีความสามารถในการทำงานไม่มากโดยได้ประยุกต์เครื่องจักรกลให้สามารถทำงานตามคำสั่งตามแถบกระดาษที่ป้อนเข้าไป ซึ่งถือว่ามีการทำงานแบบอัตโนมัติตามโปรแกรมได้ สามารถเปลี่ยนโปรแกรมโดยการเปลี่ยนแถบกระดาษได้ เครื่องจักรกลแบบดังกล่าวใช้ในอุตสาหกรรมทอผ้าเสียเป็นส่วนใหญ่ ในช่วงคริศตวรรษที่ 1940 เทคโนโลยีด้านอิเล็กทรอนิกส์และคอมพิวเตอร์เริ่มเข้ามามีบทบาทในวงการระบบควบคุมทางอุตสาหกรรม หุ่นยนต์ตัวแรกจึง ได้ถูกสร้างขึ้นโดยทีมวิจัยของสถาบันวิจัยสแตนฟอร์ดในช่วงปี ค.ศ.1960 และมีชื่อว่า "SHAKEY" โดยมีความสามารถในการจัดกล่องเป็นชั้นๆ ใช้ตัวตรวจจับเป็นกล้องโทรทัศน์ หลังจากนั้นในช่วงกลางคริศตวรรษที่ 1970 บริษัทเจนเนอร์รัลมอเตอร์ได้ร่วมมือกับสถาบันเทคโนโลยีแห่งแมตซาจูเซส (MIT) ได้พัฒนาหุ่นยนต์สำหรับประกอบชิ้นส่วนทางอุตสาหกรรมที่เรียกว่าพูม่า (PUMA : Programmable Universal Manipulator for Assembly)

กำเนิดของยุคหุ่นยนต์

ในที่นี้จะกล่าวถึงหุ่นยนต์อุตสาหกรรม การทำงานของหุ่นยนต์อุตสาหกรรมจะเลียนแบบร่างกายของมนุษย์โดยจะเลียนแบบเฉพาะส่วนของร่างกายที่จะนำไปใช้ประโยชน์ ในอุตสาหกรรมเท่านั้น นั่นคือช่วงแขนของมนุษย์ ดังนั้น บางคนอาจจะได้ยินคำว่าแขนกล ซึ่งก็หมายถึงหุ่นยนต์อุตสาหกรรม
ปัจจุบันและในอนาคตหุ่นยนต์อุตสาหกรรมจะเข้ามามีบทบาทในอุตสาหกรรมมากขึ้น โดยจะทำงานแทนมนุษย์ในงานต่าง ๆ เหล่านี้งานที่อันตราย เช่น งานยกเหล็กเข้าเตาหลอม งานที่เกี่ยวข้องกับสารเคมีงานซ้ำซากน่าเบื่อ เช่น งานยกสินค้าจากสายงานการผลิต งานประกอบ งานบรรจุผลิตภัณฑ์งานที่ต้องการคุณภาพมาตรฐานเดียวกัน เช่น งานเชื่อม งานตัด งานที่ต้องใช้ทักษะความชำนาญสูง เช่น งานเชื่อมแนว เชื่อมเลเซอร์งานที่ต้องใช้ความละเอียดประณีต เช่น งานประกอบชิ้นส่วนอิเล็กทรอนิกส์ งานตรวจสอบ (Inspection) ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเปรียบเทียบกับแรงงานคนมีข้อดี ข้อเสีย ดังนี้

ข้อดี

1. มีการทำงานที่แม่นยำ ผิดพลาดน้อย
2. ประสิทธิภาพการทำงานสูงและคงที่
3. สามารถปฏิบัติงานแทนแรงงานคน ในงานที่อันตรายได้
4. สามารถทำงานแทนแรงงานคนได้หลายคน ทำให้ต้นทุนในการผลิตลดลง
5. สามารถทำงานที่ซ้ำๆ และยากได้ดี
6. ลดความเสี่ยงของการเกิดอุบัติเหตุในโรงงาน เนื่องมาจากความประมาทของคนงาน
7. สามารถทำงานได้ตลอดเวลาโดยไม่ต้องหยุดพัก
8. หุ่นยนต์ไม่ต้องการสภาพแวดล้อมเช่นเดียวกับคน เช่น แสง การระบายอากาศ การป้องกันเสียงดัง
9. หุ่นยนต์ไม่ต้องการคำชม ไม่ต้องมีการประกัน ไม่ต้องมีวันหยุด
10. หุ่นยนต์สามารถพัฒนาให้ทำงานได้ในหลายๆด้าน แต่คนพัฒนาได้เพียงความว่องไว

ข้อเสีย

1. การนำหุ่นยนต์มาทำงานแทนแรงงานคนทำให้เกิดปัญหาทางด้านเศรษฐกิจ และสังคม เช่น ทำให้ คนขาดรายได้จนเกิดความไม่พอใจ
2. หุ่นยนต์ไม่สามารถทำงานได้ถ้าไม่มีผู้ควบคุม หรืออุปกรณ์ควบคุมเครื่องมือเสีย หรือพลังงานไม่พอ
3. หุ่นยนต์ถึงแม้จะเหนือกว่าคน แต่ก็มีข้อจำกัดคือ
    - ไม่ฉลาด
    - ไม่มีความคิดเป็นของตนเอง
    - Sensors
4. หุ่นยนต์มีราคาแพง

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานของแขนกล และสามารถนำความรู้ที่ได้ไปประยุกต์ใช้งาน
2. สามารถควบคุมความเร็วมอเตอร์ของแขนกลให้เหมาะสม
3. สามารถกำหนดทางเดินให้แขนกลเคลื่อนที่ ในลักษณะเบื้องต้น เช่น เส้นตรง, เส้นโค้ง เป็นต้นให้กับแขนกลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบข่ายของงาน

1. ออกแบบชิ้นส่วนอิเล็กทรอนิกส์ให้สามารถใช้กับแขนกลรุ่น SCORBOT – ER III ได้
2. สร้างโปรแกรมที่ใช้ควบคุมการทำงานของแขนกลเพื่อให้แขนกลเคลื่อนที่ตามเงื่อนไขอย่างง่าย เช่น เส้นตรง เส้นโค้ง ได้
3. ปรับปรุงและแก้ไขโปรแกรมให้มีประสิทธิภาพดียิ่งขึ้น
4. ทดสอบการทำงานของโปรแกรมในลักษณะของการใช้งานจริง

1.4 แนวทางการดำเนินงาน

1. ควรเพิ่มไมโครคอนโทรลเลอร์ให้มีจำนวนมากขึ้นเพื่อให้มอเตอร์เคลื่อนที่ได้สองทิศทาง
2. ควรหาไมโครคอนโทรลเลอร์ที่มีราคาถูกลงและมีความทนทาน
3. ดังนั้นต้องทราบค่ามุมและความเร็วทุกข้อของหุ่นยนต์ เพื่อนำค่าที่ทราบนั้นไปกำหนดความเร็ว
4. ส่วนของโปรแกรมที่ใช้จดจำค่า ควรที่จะจดจำค่าความเร็วในการหมุนของมอเตอร์ด้วย จะทำให้มอเตอร์แต่ละตัวมีความเร็วคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2

แขนกล SCORBOT- ER III

แขนกล SCORBOT-ER III ส่วนมากใช้ในการศึกษาในด้านการคอนโทรลในระบบต่างๆ ทำให้ได้รู้ถึงการทำงานของแขนกลในรูปแบบอื่นๆ ที่ใช้ในโรงงานอุตสหกรรม

2.1 รายละเอียดทั่วไปของแขนกล

SCORBOT–ER III คือหุ่นยนต์ที่มีข้อต่อตั้งฉากกับแนวราบซึ่งมีข้องอ 5 ส่วนติดกับที่ยึดจับวัตถุ ซึ่งมีความเป็นอิสระอยู่ 6 ระดับ การออกแบบจะออกแบบมาโดยให้มีตำแหน่งสิ้นสุดของที่ยึดจับและประยุกต์ให้มีพื้นที่ในการทำงานเพิ่มมากขึ้นแขนกลแบบ SCORBOT–ER III ซึ่งสามารถควบคุมการทำงานแบบใช้คนบังคับปุ่มกดคล้ายกับการบังคับคันโยก

รูปที่ 2.1 SCORBOT – ER III ที่ใช้ในการทดลอง

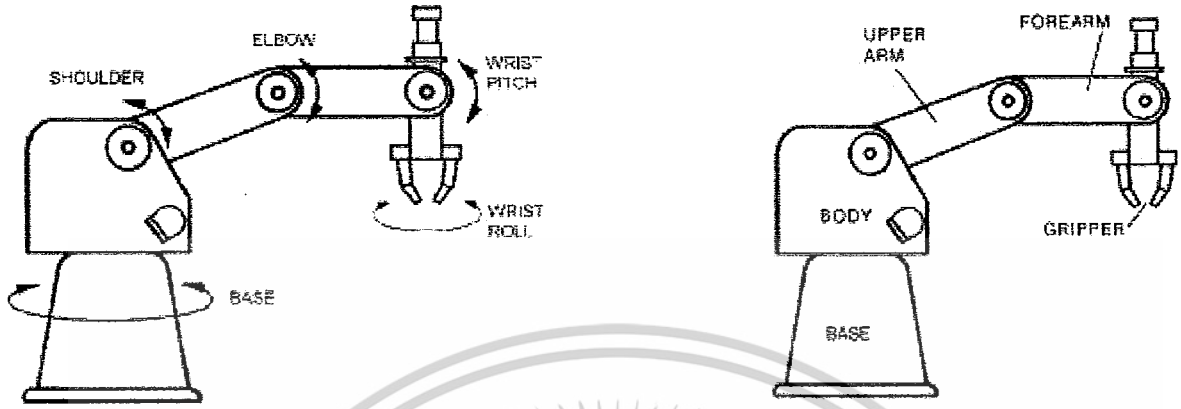
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คุณสมบัติของ SCORBOT-ER III

คุณสมบัติของ SCORBOT-ER III	
โครงสร้างทางกล	ข้อมูล
จำนวนแกน	5 แกน + ที่ยึดจับ
การเคลื่อนไหวของแกน แกน1 : ฐานหมุน แกน2 : ไหล่หมุน แกน3 : ข้อศอกหมุน แกน4 : ข้อมือตั้งขึ้น แกน5 : มือจับหมุน	 310° +130° / -35° ±130° ±130° ไม่จำกัด (ทางกล) / ±570° (ทางไฟฟ้า)
รัศมีของตัวขับสูงสุด	610 มม. (24.4")
จุดสิ้นสุดของมือจับ	ตัวตรวจจับสัญญาจะรับค่าที่วัดDC ข้อตัวจับ การวัดด้าน /แรงข้อตัวจับ โดยมีเซนเซอร์และตัวอ่านค่าที่มือจับ
ระยะมือจับกว้างสุด	75 มม. (3") ไม่มีแผ่นยาง 65 มม. (2.6") มีแผ่นยาง
HARD HOME	Micro switch จะถูกกำหนดตำแหน่งบนแต่ละแกนไว้
การประเมินผล	เครื่องตรวจจับสัญญาณบนแต่ละแกน
ระบบส่งกำลัง	เฟือง, สายพาน, เกลียวนำ
น้ำหนักสูงสุดที่สามารถจับยึดได้	1 กก. (2.2 lb), รวมที่ยึดจับ
ความสามารถเคลื่อนที่กลับไปมา	±0.5 มม. (0.02") ที่TCP
น้ำหนัก	11.5 กก. (25 lb)
ความเร็วสูงสุด	600 มม./วินาที (23.6"/วินาที)
อุณหภูมิขณะทำงาน	2°- 40° C (36°- 104° F)

ตารางที่ 2.1 อธิบายโครงสร้างของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ข้อต่อ และ Links ของแขนกลแบบ SCORBOT – ER III

จากรูปที่ 2.2 อธิบายการเคลื่อนที่ของข้อต่อว่าแต่ละข้อต่อทำงานดังต่อไปนี้

แกน	ชื่อข้อต่อ	การเคลื่อนไหว	มอเตอร์
1	ฐาน	หมุนรอบตัว	1
2	ช่วงไหล่	ยกสูงและต่ำกว่าแขนบน	2
3	ข้อศอก	ยกสูงและต่ำกว่าแขนล่าง	3
4	ข้อมือ	ยกสูงและต่ำกว่าที่ยึดจับ	4+5
5	มือจับ	หมุนที่ยึดจับ	4+5

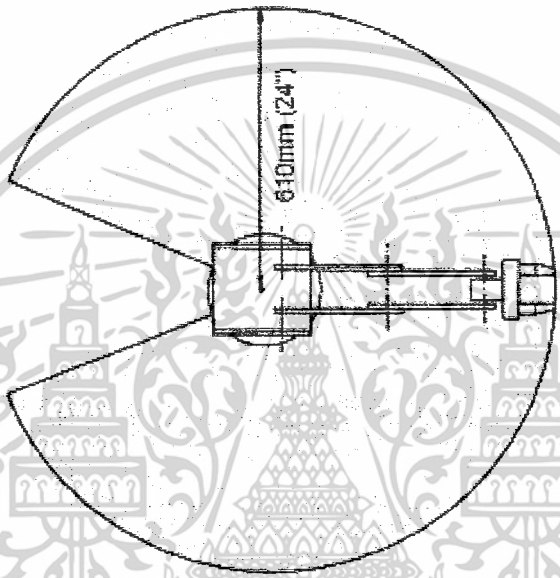
ตารางที่ 2.2 อธิบายการเคลื่อนที่ของข้อต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 พื้นที่ใช้ในการทำงานของ SCORBOT – ER III

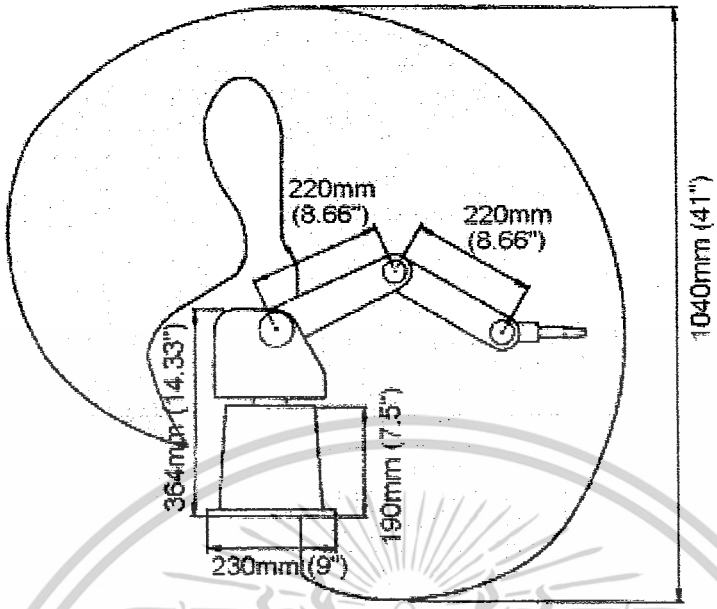
ความยาวของที่เชื่อมข้อต่อและองศาของการหมุนข้อต่อที่กำหนดไว้ พื้นที่ในการทำงานของหุ่นยนต์ดังแสดงไว้ในรูปที่ 2.3 แสดงขนาดและระยะของช่วงแขน SCORBOT – ER III

ฐานของหุ่นยนต์โดยปกติจะถูกกำหนดพื้นที่การทำงานไว้คงที่หรืออาจจะเคลื่อนไหวได้จะทำให้ขอบเขตการทำงานกว้างออกไป



รูปที่ 2.3 SCORBOT – ER III ด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 SCORBOT – ER III ด้านข้าง

2.4 อุปกรณ์และทฤษฎีที่ใช้ใน SCORBOT- ER III

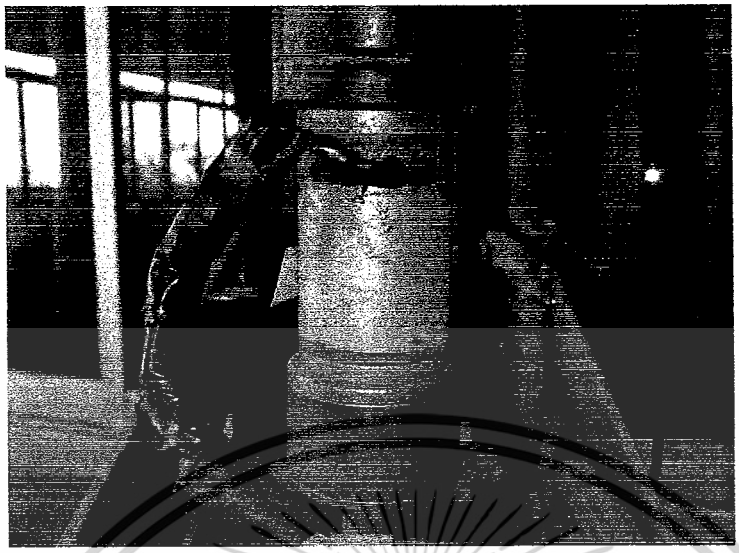
2.4.1 มอเตอร์ไฟฟ้า

มอเตอร์ไฟฟ้ากระแสตรงและเครื่องกำเนิดไฟฟ้ากระแสตรงนั้น มีส่วนประกอบสำคัญ ๆ ที่เหมือนกัน ซึ่งถ้าดูจากภายนอกแล้วจะมีลักษณะเหมือนกันทุกประการส่วนที่แตกต่างกันก็คือลักษณะการใช้งานเท่านั้น เครื่องกำเนิดไฟฟ้ากระแสตรง ได้รับพลังงานกลจากภายนอกมาขับอาร์เมเจอร์ให้หมุนตัดกับสนามแม่เหล็กที่เกิดจากขดลวดฟิลด์คอยล์ในเครื่องกำเนิดนั้น ส่วนมอเตอร์ไฟฟ้ากระแสตรงได้รับพลังงานจากภายนอกมาทำให้เกิดสนามแม่เหล็กขึ้นในมอเตอร์นั้นไปขับขดลวดตัวนำในอาร์เมเจอร์หมุน ดังนั้นเพลาของมอเตอร์ก็จะหมุน ซึ่งจะได้พลังงานกลออกไปใช้งาน

แรงดันไฟฟ้า	มอเตอร์ 12 V DC
ประสิทธิภาพมอเตอร์	15 oz. ในจุดทอร์คสูงสุด 70 W กำลังทอร์คสูงสุด
อัตราส่วนเฟือง	มอเตอร์ 1,2,3 127.1 : 1 มอเตอร์ 4.5 65.5 : 1 มอเตอร์ 6 19.5 : 1

ตารางที่ 2.3 คุณสมบัติของมอเตอร์

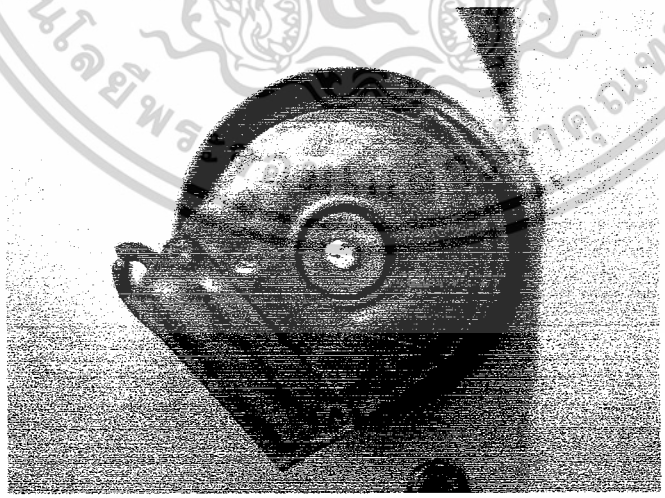
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



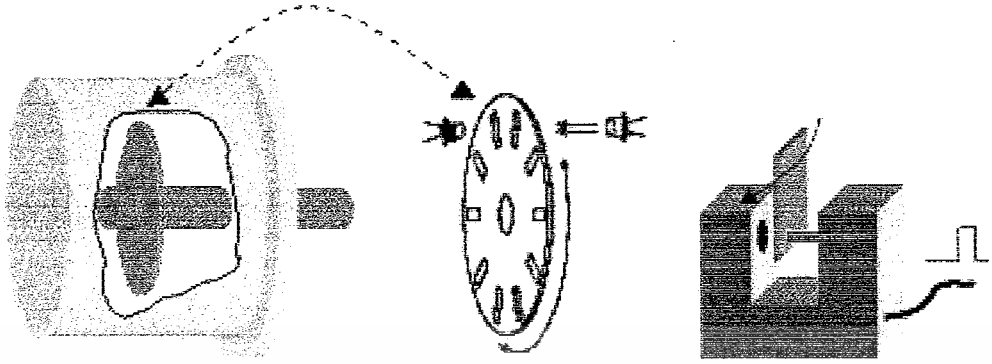
รูปที่ 2.5 แสดงมอเตอร์ที่ใช้ในตัวหุ่น

2.4.2 ตัวตรวจจับสัญญาณ

สถานที่และการเคลื่อนที่ของแต่ละแกนที่ได้จากการวัดโดยการตรวจจับสัญญาณทางไฟฟ้าของเพลาที่ทำให้แกนหมุนเมื่อแกนของหุ่นยนต์เคลื่อนที่ตัวตรวจจับสัญญาณจะทำให้เกิดสัญญาณทางไฟฟ้าสูงและต่ำสลับกันไป หมายเลขของสัญญาณจะเท่ากับจำนวนการเคลื่อนที่ของแกน ความต่อเนื่องของสัญญาณจะแสดงให้รู้ถึงทิศทางการเคลื่อนที่ Controller จะอ่านสัญญาณเหล่านี้และจะกำหนดขอบเขตและทิศทางการเคลื่อนที่ของแกน

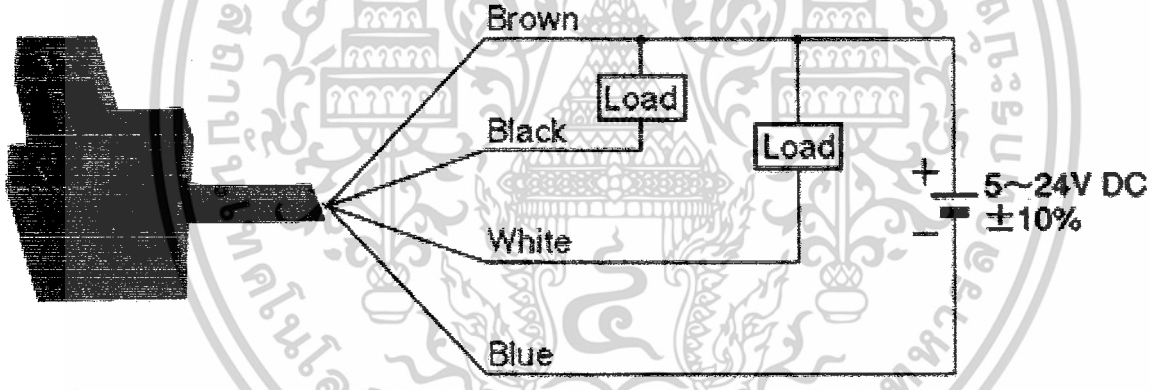


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



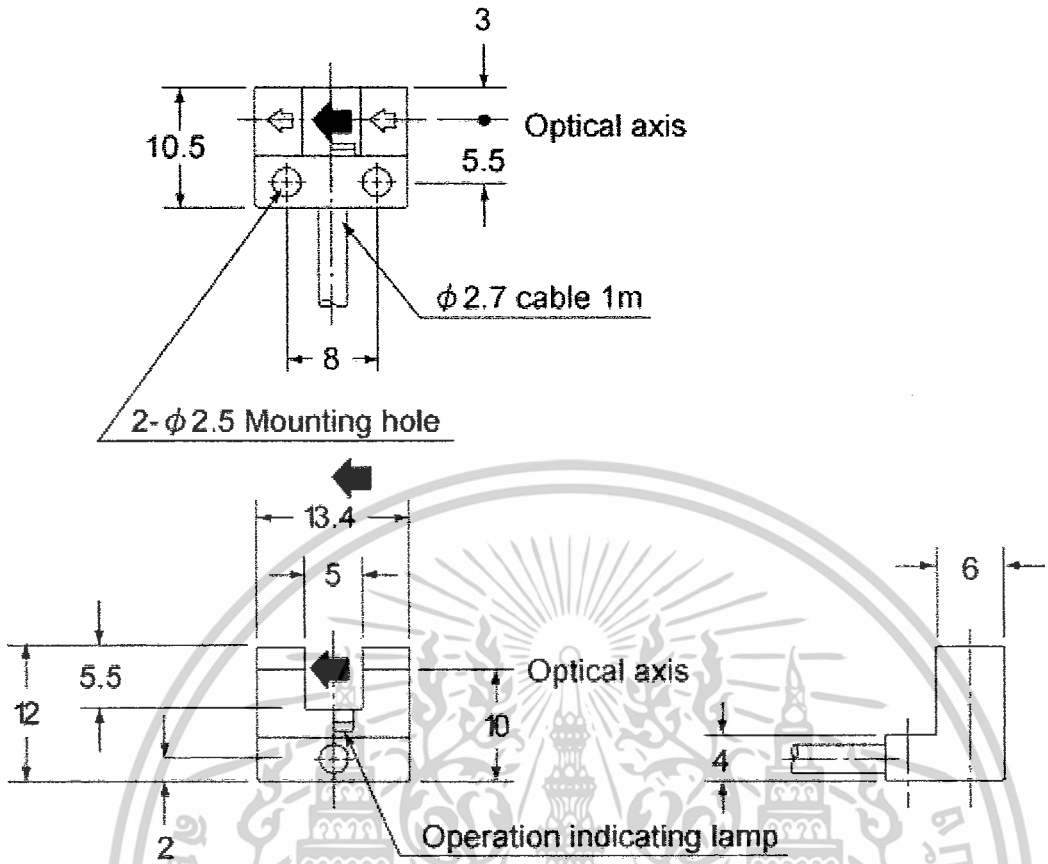
รูปที่ 2.6 ตัวตรวจจับสัญญาณ

โครงสร้างของ limit sensor มีวงจรดังรูปที่ 2.8



รูปที่ 2.7 limit sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 วงจรของ limit sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

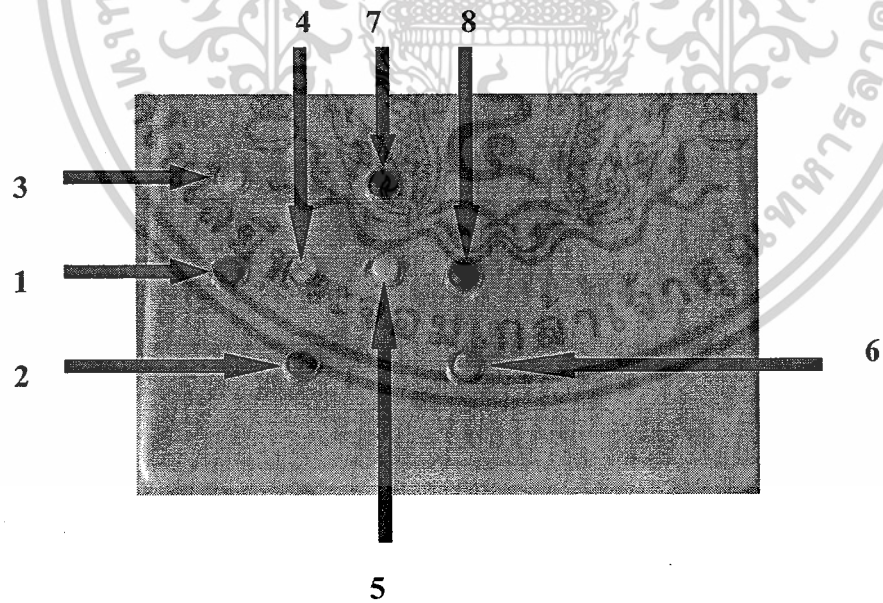
การออกแบบระบบควบคุมใน SCORBOT - ER III

ในที่นี้เราได้ทำการสร้างตัวควบคุมแบบมือถือและได้สร้างระบบควบคุมแบบอัตโนมัติโดยเราใช้ชุดควบคุมแบบ H-Bridge เป็นตัวขับมอเตอร์ ไมโครคอนโทรลเลอร์ที่เราใช้กับแผงวงจรควบคุมแบบอัตโนมัติเป็นไมโครคอนโทรลเลอร์ AVR เป็นตัวรับส่งค่าของแขนกล

3.1 ตัวควบคุมแบบมือถือ

ใช้ในการบังคับตัวหุ่นยนต์ให้เคลื่อนไปยังจุดหมายที่ต้องการ ในงานอุตสาหกรรมนั้นจะใช้ในการบังคับเพื่อที่ให้คอมพิวเตอร์เป็นตัวจดจำ

ปุ่มกดที่ 1 จะควบคุมมอเตอร์ตัวที่ 1 ให้หมุนขวา
ปุ่มกดที่ 2 จะควบคุมมอเตอร์ตัวที่ 1 ให้หมุนซ้าย
ปุ่มกดที่ 3 จะควบคุมมอเตอร์ตัวที่ 2 ให้ยกขึ้น
ปุ่มกดที่ 4 จะควบคุมมอเตอร์ตัวที่ 2 ให้ยกลง
ปุ่มกดที่ 5 จะควบคุมมอเตอร์ตัวที่ 3 ให้ยกขึ้น
ปุ่มกดที่ 6 จะควบคุมมอเตอร์ตัวที่ 3 ให้ยกลง
ปุ่มกดที่ 7 จะควบคุมมอเตอร์ตัวที่ 4 ให้หมุนขวา
ปุ่มกดที่ 8 จะควบคุมมอเตอร์ตัวที่ 4 ให้หมุนซ้าย



รูปที่ 3.1 ชุดบังคับหุ่นยนต์แบบมือถือ

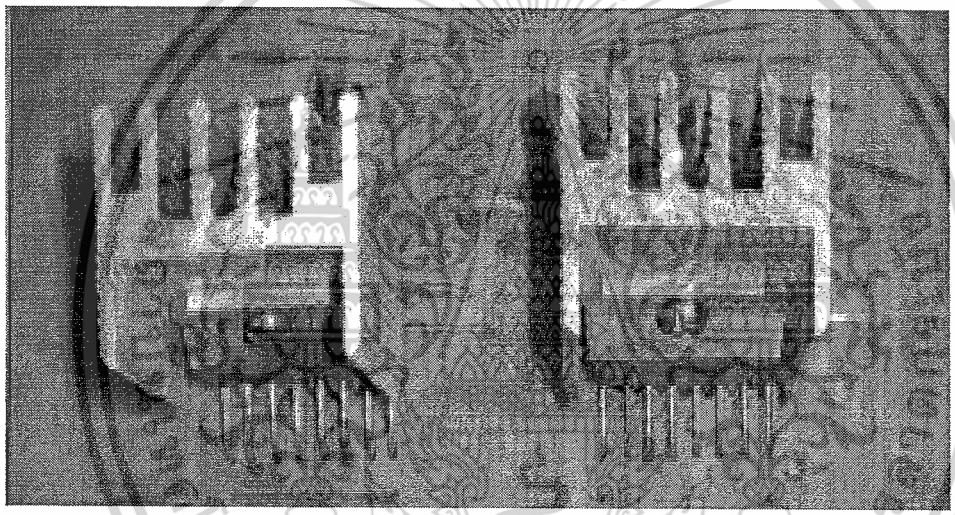
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ตัวควบคุมแบบอัตโนมัติ

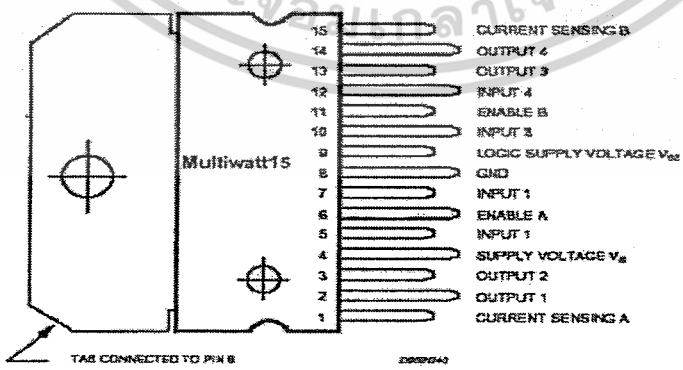
ตัวควบคุมแบบอัตโนมัติได้ทำการสร้างขึ้นนั้น อุปกรณ์ที่ใช้ในตัวควบคุมแบบอัตโนมัติได้แก่ ชุดควบคุมแบบ H- Bridge และไมโครคอนโทรลเลอร์

3.2.1 ชุดควบคุมแบบ H-Bridge

การเปลี่ยนทิศทางการหมุนของดีซีมอเตอร์แบบแม่เหล็กถาวรทำได้โดยการสลับขั้วแหล่งจ่ายไฟฟ้าที่ต่อเข้ากับตัวมอเตอร์และในทางปฏิบัติจะใช้วงจรอิเล็กทรอนิกส์ที่เรียกว่า H Bridge เป็นตัวจัดการการทำงาน ซึ่งในปัจจุบันนี้ได้มีการผลิต IC ขับมอเตอร์แบบ H Bride ขึ้นมามากมาย ในส่วนของโครงงานนี้ได้ศึกษา DUAL FULL-BRIDGE DRIVER L298N ซึ่งมีตัวถังและวงจรภายในดังรูป



รูปที่ 3.2 ไอซี L298N

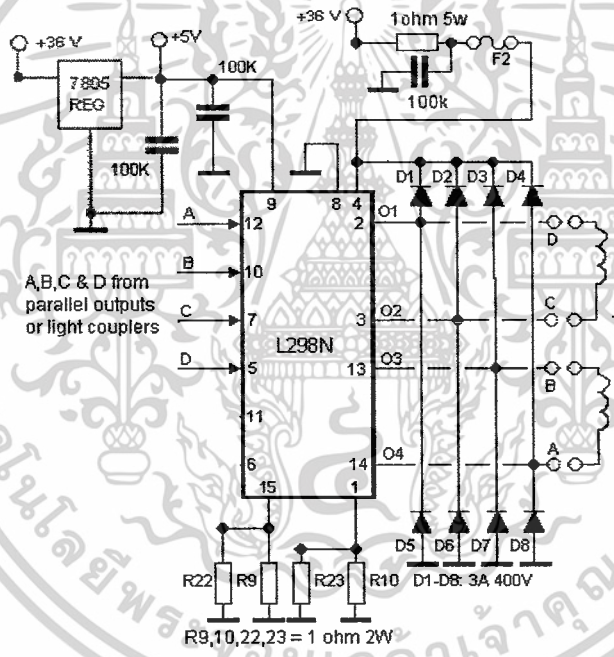


รูปที่ 3.3 การจัดการขาของไอซี L298N

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรภายในจะเห็นว่า L298N สามารถขับโหลดได้ 2 ช่องและสามารถรับสัญญาณ ควบคุมแบบ TTL (Transistor Transistor Logic) เพื่อที่จะควบคุมทิศทางการไหลของกระแส นอกจากนี้ยังมีขา Enable เป็นตัวตัดสินว่าจะให้โหลดที่ต่ออยู่ทำงานหรือไม่โดยไม่สนใจสัญญาณควบคุมขา Emitter ของ Transistor ทั้งสองข้างของแต่ละ bridge จะต่อกับตัวต้านทานภายนอกเพื่อที่จะใช้ในการกำหนดค่ากระแสไฟฟ้าที่ไหลได้ โดยหากเกินกว่าที่วงจรและโหลดที่ต่ออยู่สามารถที่จะรับได้ก็อาจจะมีวงจรเพื่อที่ทำการ Disable การทำงานของโหลดได้

จะเห็นว่าวงจรภายในไม่มีการต่อไดโอดเพื่อให้กระแสไฟฟ้าไหลได้เมื่อโหลดที่ใช้เป็นตัวเหนี่ยวนำ (มอเตอร์ รีเลย์) เนื่องจากสำหรับโหลดที่เป็นตัวเหนี่ยวนำค่ากระแสไฟฟ้าจะไม่สามารถเปลี่ยนเป็นศูนย์ได้ในเวลาทันทีทันใด (คล้ายกับกรณีที่ตัวเก็บประจุไม่สามารถเปลี่ยนค่าความต่างศักย์ตกคร่อมได้อย่างทันทีทันใด) จึงต้องมีการต่อไดโอดเพื่อให้กระแสไฟฟ้าไหลในกรณีที่เราเปิดสวิตซ์แต่มอเตอร์ยังคงหมุนอยู่โดยที่การต่อไดโอดภายนอกแสดงดังรูปที่ 3.4



รูปที่ 3.4 การต่อวงจรใช้งานของ L298N

สำหรับโครงการนี้ได้ทำการดัดแปลงวงจร H-Bridge โดยต้องการให้สามารถควบคุมทิศทางได้โดยการใช้สายควบคุมเพียงเส้นเดียว และใช้สัญญาณ Pulse Width Modulation ต่อเข้ากับขา Enable เพื่อให้ได้ระดับความเร็วตามต้องการ

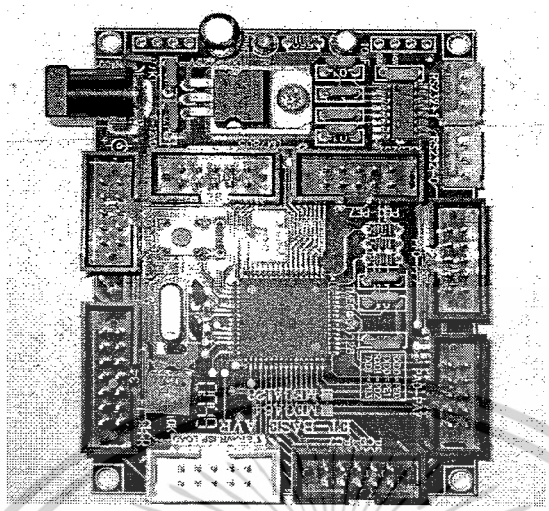
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองได้สร้างสัญญาณ Pulse Width Modulation โดยใช้ไมโครคอนโทรเลอร์ Atmega8 รับข้อมูลขนาด 8 บิตจากพอร์ตอนุกรมโดยใช้โปรแกรม Visual Basic 6 ในการเขียนโปรแกรมส่งข้อมูลจากคอมพิวเตอร์และใช้ UART ภายใต้ตัวไมโครคอนโทรลเลอร์ Atmega8 รับข้อมูลการสื่อสารแบบ Asynchronous Serial Communication แล้วนำข้อมูลมาสร้างสัญญาณโดยหากข้อมูลเป็นศูนย์ก็จะได้สัญญาณที่มีค่า Duty Cycle เป็น 0% (นั่นคือค่าเฉลี่ยความต่างศักย์เป็น 0 Volt) และหากข้อมูลเป็น 255 ก็จะได้สัญญาณที่มีค่า Duty Cycle เป็น 100% (นั่นคือค่าเฉลี่ยความต่างศักย์เป็น V supply Volt) และนำสัญญาณที่ได้ไปเข้าขา Enable ของ L298N ส่วนการเปลี่ยน ทิศทางก็ใช้การใส่ Logic 1 ให้กับขาที่เราต้องการให้กระแสไฟฟ้าไหลซึ่งจะทำให้มอเตอร์หมุนในทิศทางที่ต้องการ

3.2.2 ไมโครคอนโทรลเลอร์

เนื่องจากหุ่นยนต์เป็นสิ่งประดิษฐ์ทางวิศวกรรมที่สามารถทำงานได้แบบอัตโนมัติ ดังนั้นส่วนที่ขาดไม่ได้ของหุ่นยนต์ทุกชนิดคือหน่วยประมวลผลสำหรับสิ่งมีชีวิตการควบคุมการทำงานหลักจะถูกควบคุมโดยสมองของมนุษย์ทำหน้าที่ประมวลผลข้อมูลที่ได้รับการ โสตประสาทสัมผัสต่างๆ เช่นตา หู จมูกปาก เป็นต้น เมื่อสมองประมวลผลข้อมูลเสร็จก็สั่งให้อวัยวะเช่น แขนหรือขา ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่รวมเอาหน่วยประมวลผล หน่วยคำนวนทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาท์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียน โปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ(Microprocessor) ซึ่งเป็นอุปกรณ์อิเล็กทรอนิกส์ที่ประดิษฐ์มาจากสารกึ่งตัวนำ โดยภายใน ไมโครคอนโทรลเลอร์นี้ประกอบด้วยส่วนประกอบต่างๆจำนวนมากเช่น หน่วยสำหรับคำนวนทางคณิตศาสตร์ หน่วยประมวลผลสำหรับการตัดสินใจทางตรรกศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แผงวงจรไมโครคอนโทรลเลอร์ (Microcontroller)

ในที่นี้เราได้ใช้ไมโครคอนโทรลเลอร์ตระกูล AVR ของบริษัท ATMEL เบอร์ ATMEGA128 แบบ TQFP 64 PIN โดยสามารถต่อสัญญาณนาฬิกาได้ 16 MHz มีหน่วยความจำแบบ FLASH 128 kbyte หน่วยความจำโปรแกรม 4 kbyte และ RAM 4 kbyte

1. โครงสร้างของบอร์ด

หมายเลข 1 คือ MCU เบอร์ ATmega64 หรือ ATmega128 ซึ่งเป็น MCU ตระกูล AVR จาก ATMEL

หมายเลข 2 คือ Switch RESET ใช้สำหรับ Reset การทำงานของ MCU

หมายเลข 3 คือ Crystal ค่า 16 MHz

หมายเลข 4 คือ ตัวต้านทานสำหรับปรับค่าความสว่างให้ LCD

หมายเลข 5 คือ PORTC มีขนาด 8 Bit คือ PC0-PC7

หมายเลข 6 คือ พอร์ต ISP LOAD ใช้สำหรับดาวน์โหลด Hex File ให้กับ MCU

หมายเลข 7 คือ พอร์ต ET-CLCD สำหรับเชื่อมต่อกับ LCD ชนิด Character Type ซึ่งใช้การเชื่อมต่อแบบ 4 Bit

หมายเลข 8 คือ PORTD มีขนาด 8 Bit คือ PD0-PD7

หมายเลข 9 คือ PORTB มีขนาด 8 Bit คือ PB0-PB7

หมายเลข 10 คือ PORTE มีขนาด 8 Bit คือ PE0-PE7

หมายเลข 11 คือ PORTF มีขนาด 8 Bit คือ PF0-PF7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

หมายเลข 12 คือ PORTA มีขนาด 8 Bit คือ PA0-PA7

หมายเลข 13 และ 14 คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป

หมายเลข 15 คือ LED Power ใช้สำหรับแสดงสถานะของแหล่งจ่ายไฟ +5VDC

หมายเลข 16 คือ ขั้วต่อแหล่งจ่ายไฟสำหรับเลี้ยงวงจรของบอร์ด

2. ขั้วต่อสัญญาณต่าง ๆ

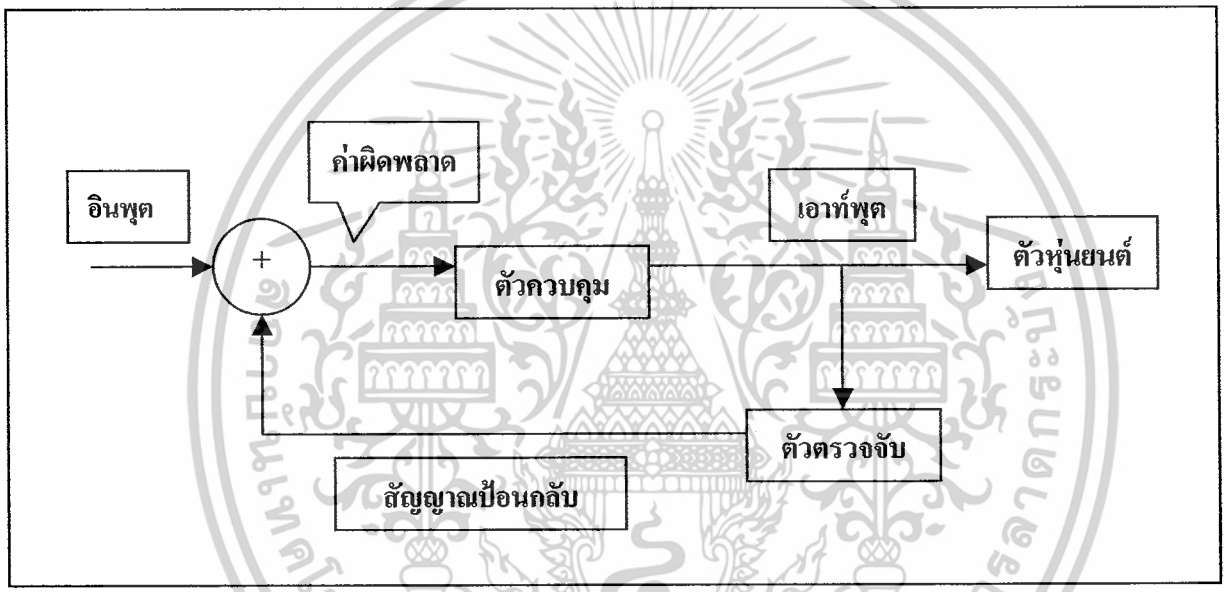
สำหรับขั้วต่อสัญญาณของพอร์ต I/O จาก MCU นั้นจะถูกออกแบบและจัดเตรียมไว้ผ่านทางขั้วต่อแบบ IDC-Header ขนาด 10 Pin (2X5) จำนวน 6 ชุด คือ PA, PB, PC, PD, PE, PF ตามลำดับ โดยที่ขั้วต่อสัญญาณแต่ละชุด จะประกอบไปด้วยสัญญาณของ I/O ที่เชื่อมต่อมาจากขาสัญญาณของ MCU โดยตรงทั้งหมด โดยจุดเชื่อมต่อกับสัญญาณภายนอกบอร์ดมีดังนี้

- ขั้วต่อแหล่งจ่ายไฟสำหรับเลี้ยงวงจรของบอร์ด
- ขั้วต่อ PORTA มีขนาด 8 Bit คือ PA0-PA7
- ขั้วต่อ PORTB มีขนาด 8 Bit คือ PB0-PB7
- ขั้วต่อ PORTC มีขนาด 8 Bit คือ PC0-PC7
- ขั้วต่อ PORTD มีขนาด 8 Bit คือ PD0-PD7
- ขั้วต่อ PORTE มีขนาด 8 Bit คือ PE0-PE7
- ขั้วต่อ PORTF มีขนาด 8 Bit คือ PF0-PF7
- ขั้วต่อ ET-CLCD สำหรับเชื่อมต่อกับ LCD ชนิด Character Type
- ขั้วต่อ RS232 จำนวน 2 ช่อง โดยเชื่อมต่อกับสัญญาณ PE0 (RXD0) และ PE1 (TXD0) จำนวน 1 ช่อง ส่วนที่เหลืออีก 1 ช่อง จะต่อกับสัญญาณ PD2 (RXD1) และ PD3 (TXD1) เพื่อให้ผู้ใช้สามารถต่อทดลองการติดต่อสื่อสาร RS232
- ขั้วต่อ ISP LOAD ใช้สำหรับดาวน์โหลด Hex File ให้กับ MCU

83720
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 พื้นฐานระบบควบคุมหุ่นยนต์

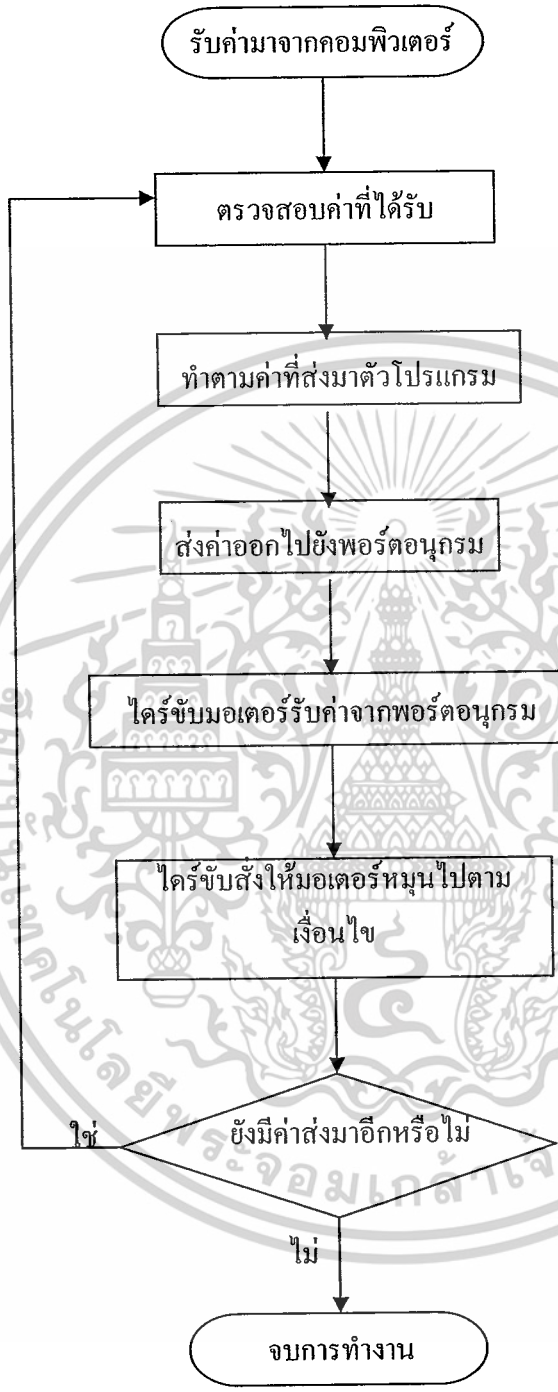
การทำงานของหุ่นยนต์ใช้ระบบควบคุมแบบป้อนกลับ (Feed-back Control System) ซึ่งมีไดอะแกรมดังรูปที่ 1.14 กล่าวคือ เมื่อผู้ใช้ป้อนค่าอินพุต (อาจจะเป็นตำแหน่งที่หุ่นยนต์จะเคลื่อนที่หรือเป็นเส้นทางการเคลื่อนที่ของหุ่นยนต์) ค่าที่ป้อนจะถูกเก็บไว้สำหรับคำนวนทิศทางการเคลื่อนที่ของหุ่นยนต์ ตัวควบคุมจะสั่งให้หุ่นยนต์เคลื่อนที่ไปหายังตำแหน่งเป้าหมาย โดยมีตัวตรวจจับเป็นอุปกรณ์ที่ใช้สำหรับอ่านค่าจากเอาท์พุตเพื่อนำค่าดังกล่าวมาเปรียบเทียบกับค่าอินพุต ถ้าค่าที่ได้ไม่ตรงกันจะมีค่าผิดพลาดเกิดขึ้น ซึ่งค่าผิดพลาดที่ได้จะถูกนำไปใช้ในการคำนวนด้วยตัวควบคุมเพื่อสั่งให้หุ่นยนต์แก้ไขและเคลื่อนที่ไปสู่เป้าหมายให้ได้



รูปที่ 3.6 ตัวอย่างการควบคุมแบบป้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบโปรแกรมการสั่งงาน



รูปที่ 3.7 แผนผังลำดับการทำงานของโปรแกรมภาษาซีในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การออกแบบโปรแกรมคำนวนความเร็วที่ใช้ในแขนกล

โปรแกรม Visual Basic (VB) เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่ นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า คอนโทรล(Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event - Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด ผู้ใช้กดปุ่มเมาส์ เป็นต้น

เครื่องมือ หรือ คอนโทรล ต่าง ๆ ที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form Textbox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจ็กต์) นั่นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน Visual Basic จะเป็นออบเจ็กต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แก้ไขคุณสมบัติของออบเจ็กต์นั้นได้โดยตรง ในทุกๆ ออบเจ็กต์จะมีคุณสมบัติ (properties) และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจ็กต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจ็กต์

ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Basic การเขียนโค้ดจะถูกแบ่งออกเป็นส่วนๆ เรียกว่า โพรซีเดอร์ (procedure) แต่ละโพรซีเดอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไปแล้ว ทำให้คอนโทรลหรือออบเจ็กต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็นการเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Basic ไม่สามารถทำได้

3.5.1 สาเหตุที่เลือกใช้โปรแกรมวิชวลเบสิกในการเขียนโปรแกรม

1. เป็นภาษาระดับกลาง มีลักษณะคล้ายภาษาอังกฤษ จึงทำให้ง่ายต่อการทำความเข้าใจและพัฒนาโปรแกรม

2. มีคำสั่งที่ใช้ในการติดต่อสื่อสารกับพอร์ตอนุกรมซึ่งสามารถนำไปควบคุมเครื่องได้

3. มีความสามารถในงานประเภทกราฟฟิคอยู่ในระดับที่น่าพอใจ ทำให้สามารถพัฒนาโปรแกรมได้อย่างสะดวก

4. วิชวลเบสิกใช้พัฒนาโปรแกรมที่ใช้สื่อสารกับผู้ใช้เป็นหลัก ดังนั้นจึงสามารถพัฒนาหน้าตา(Interface) ของโปรแกรมออกมาได้สวยงามน่าใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

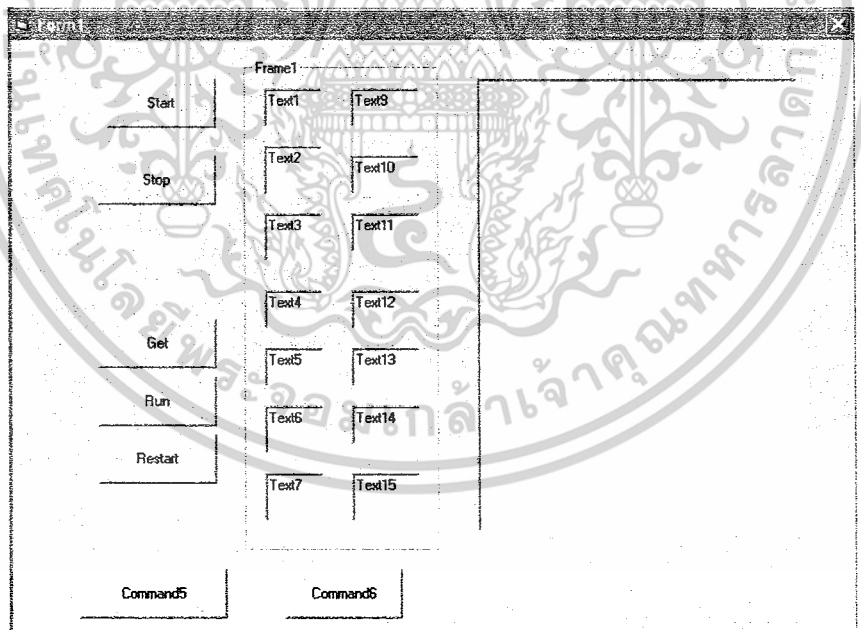
3.5.2 โปรแกรมคำนวนความเร็วในการเคลื่อนที่

โปรแกรมจึงได้ถูกพัฒนาออกให้ผู้ใช้สามารถใช้งานได้อย่างง่ายดาย โดยโปรแกรมจะแบ่งขั้นตอนต่างๆ ออกเป็น 2 ส่วน คือส่วนการรับค่าจากมอเตอร์และ ส่วนที่ส่งค่าจากคอมพิวเตอร์ออกไปเป็นตัวอักษร หลังจากนั้นก็ส่งออกไปที่พอร์ตอนุกรมได้เลยโดยมีการส่งออกมาที่ละตัวอักษรในระยะเวลาที่สามารถกำหนดได้ส่วนทางด้านไมโครคอนโทรลเลอร์ก็จะรับค่าที่ได้มาแปรเป็นลักษณะการทำงานที่เราต้องการเพื่อให้ตัวหุ่นเคลื่อนที่

ในการเคลื่อนที่ของแขนกลเองยังคงเป็นการเคลื่อนที่แบบแยกส่วนแต่มอเตอร์ยังหมุนด้วยความเร็วที่ไม่แน่นอนทั้งนี้เราสามารถพัฒนาต่อไปได้

3.5.3 หลักการทำงานของโปรแกรม

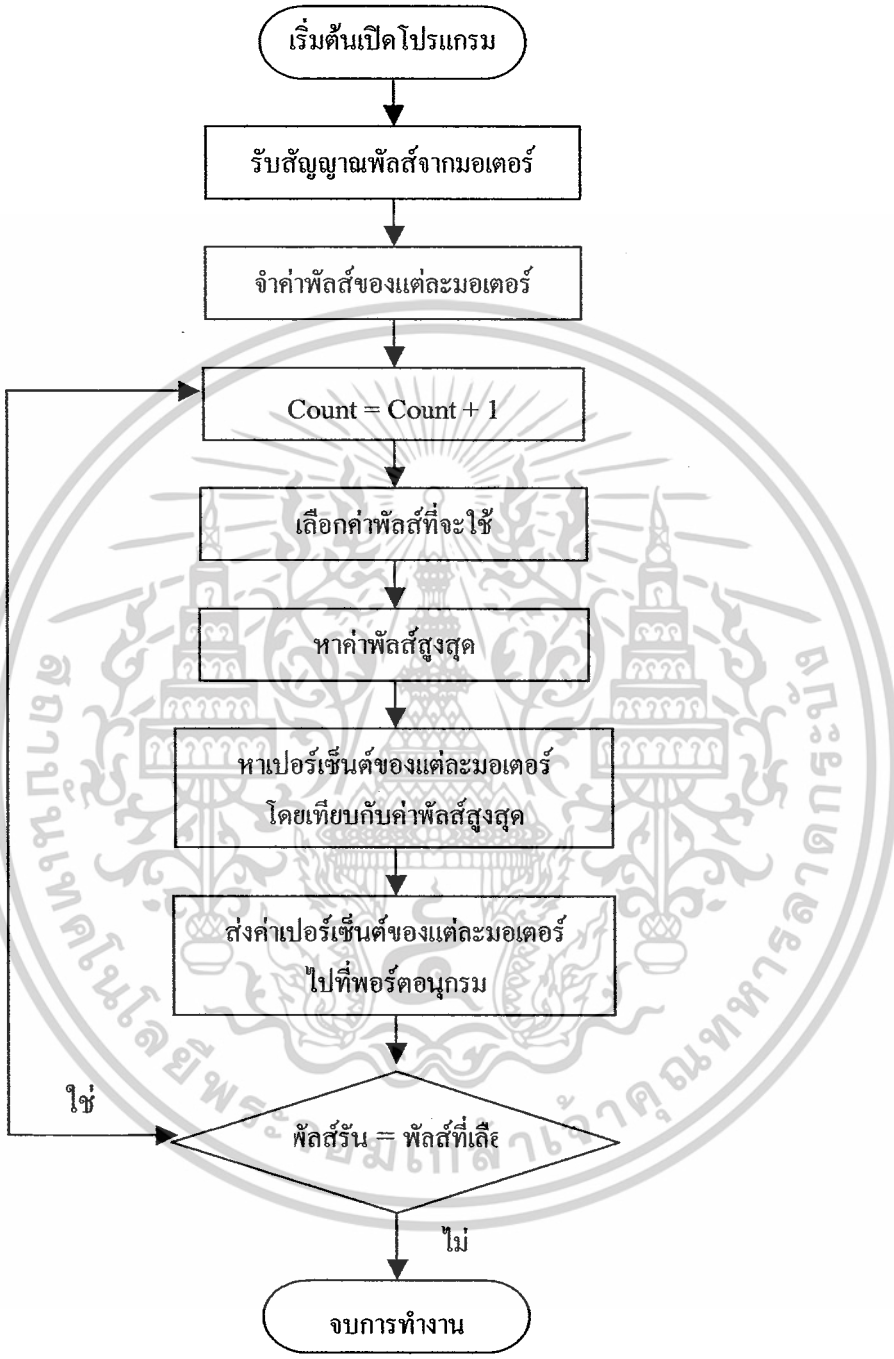
หลังจากที่เราได้ทำการเปิดโปรแกรมจะพบกับหน้าต่างของรูปภาพโปรแกรม ซึ่งเราจะเริ่มต้นตั้งตัวหุ่นให้อยู่ที่จุดศูนย์จากนั้นทำการบังคับตัวหุ่นไปยังเป้าหมายและตัวโปรแกรมจะทำการรับค่าไปที่คอมพิวเตอร์ทำการจำค่า โปรแกรมจะทำการคำนวนหาค่าเปอร์เซ็นต์และส่งค่าออกไปเป็นตัวอักษรจากนั้นโปรแกรมจะนับค่าให้เท่ากับค่าที่จำไว้ในขณะที่มอเตอร์ทำการหมุนไปด้วยและโปรแกรมจะไปทำค่าที่จำค่าอื่นจนหมดค่าที่ทำการจำไว้ การส่งค่าไปที่ไมโครคอนโทลเลอร์จะส่งค่าไปเป็นเปอร์เซ็นต์และไมโครคอนโทลเลอร์ทำการเปลี่ยนเปอร์เซ็นต์เป็นความเร็วส่งไปที่มอเตอร์ของตัวหุ่นยนต์และตัวหุ่นยนต์จะเคลื่อนที่ไปตามที่เราต้องการ



รูปที่ 3.8 หน้าตาโปรแกรมวิชวลเบสิกที่ใช้กับหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 แผนผังลำดับการทำงานของโปรแกรม Visual Basic (VB)



รูปที่ 3.9 แผนผังลำดับการทำงานของโปรแกรม Visual Basic (VB)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 PWM (Pulse Width Modulation)

3.6.1 การสร้างสัญญาณพัลส์โดยใช้โมดูล PWM

การใช้งานฮาร์ดแวร์โมดูลที่มีหน้าที่เฉพาะเช่น PWM นี้ จะช่วยประหยัดเวลาการทำงานของหน่วยประมวลผลมากกว่าวิธีการวนรอ (polling) เพราะทำให้หน่วยประมวลผลว่างและพร้อมที่จะทำงานอื่นได้ทันที PIC แต่ละเบอร์ก็จะมีโมดูลที่ทำหน้าที่เฉพาะแตกต่างกันไป ซึ่งเป็นสิ่งหนึ่งที่เราต้องใช้ในการพิจารณาเลือกเบอร์ PIC ให้เหมาะกับงานของเรา

ในการใช้งานโมดูล PWM ให้เปิดดู datasheet ในหัวข้อ 8.0 Capture/Compare/PWM (CCP) Module และในหัวข้อ 7.0 Timer2 Module เพื่อทำความเข้าใจวิธีการใช้งานโมดูล PWM ซึ่งสามารถสรุปได้ดังนี้

1. ตั้งค่า PWM period โดยกำหนดค่าที่รีจีสเตอร์ PR2 ซึ่งหาได้โดยคำนวนจากสูตร

PWM period = [(PR2)+1] x 4 x Tosc x (TMR2 prescale value)

ตัวอย่างการคำนวน Period

สมมติว่าถ้าเราต้องการให้ PWM period = 0.5 ms

Tosc = 1/f = 0.05 us (เนื่องจากใช้ crystal ความถี่ f = 20 MHz)

TMR2 prescale value = 16 (เลือกใช้ค่า 1, 4 หรือ 16 ที่จะทำให้ได้ PR2 ไม่เกิน 256)

แทนค่าจะได้

PR2 = PWM period /[4 x Tosc x (TMR2 prescale value)] – 1

PR2 = 0.5 ms/[4 x 0.05 us x 16] – 1

PR2 = 155.25 ≈ 155

2. ตั้งค่า PWM duty cycle โดยกำหนดค่าที่รีจีสเตอร์ CCPR1L และ CCP1CON<5:4> (8 + 2 = 10 บิต) ค่า CCPR1L และ CCP1CON<5:4> หาได้โดยคำนวนจากสูตร

PWM duty cycle = (CCPR1L:CCP1CON<5:4>) x Tosc x (TMR2 prescale value)

ตัวอย่างการคำนวน Duty Cycle

สมมติว่าถ้าเราต้องการให้ PWM duty cycle = 125 us

Tosc = 1/f = 0.05 us (เนื่องจากใช้ crystal ความถี่ f = 20 MHz)

TMR2 prescale value = 16 (ใช้ค่าเท่ากับค่าที่ใช้ในการคำนวน PWM period)

แทนค่าจะได้

CCPR1L:CCP1CON<5:4> = (PWM duty cycle) / [Tosc x (TMR2 prescale value)]

CCPR1L:CCP1CON<5:4> = 125 us / (0.05 us x 16)

CCPR1L:CCP1CON<5:4> = 156.25 ≈ 156 (เลขฐานสิบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CCPR1L:CCP1CON<5:4> = 0010011100 (เลขฐานสอง)

ดังนั้นจะได้ว่า

CCPR1L = 00100111 = 0x27 (8 บิตทางซ้าย)

CCP1CON<5:4> = 00 (2 บิตทางขวา)

3. กำหนดค่ารีจีสเตอร์ที่จำเป็นอื่นๆ

กำหนดให้ขา RC2/CCP1 เป็นขาเอาท์พุตของสัญญาณ โดยกำหนดให้ TRISC2 = 0

กำหนดให้ขา CCP1 ให้เป็นโหมด PWM โดยกำหนดที่ CCP1CON (CCP1M3:CCP1M0 = 11xx โดยที่ x คือค่า 0 หรือ1 ก็ได้)

กำหนดให้ TMR2 prescale value (ดู datasheet ในหัวข้อที่ 7.0 Timer2 Module หน้า 61) ให้เท่ากับค่าที่คำนวนไว้ (prescale value = 16) โดยกำหนดที่รีจีสเตอร์ T2CON (T2CKPS1:T2CKPS0 = 1x)

4. สั่งให้ Timer2 เริ่มทำงาน โดยกำหนดให้ TMR2ON = 1

3.6.2 หลักการทำงาน PWM

หลักการทำงานของหุ่นยนต์ เริ่มจากรับค่ามาจากมอเตอร์ และทำการเก็บค่าไว้ นำค่าที่เก็บไว้ มาหาค่าเปอร์เซ็นต์ ทำการส่งข้อมูลที่จะควบคุมหุ่นยนต์ผ่านพอร์ตอนุกรม เพื่อให้ไมโครคอนโทรลเลอร์ AVR MEGA 128 นำข้อมูลที่ได้รับเปลี่ยนเป็นสัญญาณ PWM ไปควบคุมชุดขับมอเตอร์ (drive motor) แต่ละข้อต่อ

ขณะนี้ได้คิดค้นวิธีการควบคุมทิศทางการเคลื่อนที่ของหุ่นยนต์เป็นแบบโปรแกรม PWM มาใช้ควบคุมแหล่งจ่ายไฟโดยใช้ไมโครคอนโทรลเลอร์ และนำไปขับมอเตอร์ชุดขับเคลื่อน

3.6.3 การควบคุมแบบ PWM (Pulse Width Modulation)

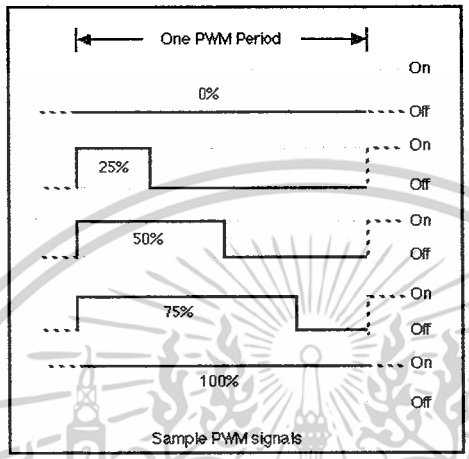
คือ การควบคุมแรงดันให้กับโหลดเพื่อควบคุมความเร็ว โดยกำหนด Duty Cycle ของการ On และ Off เป็นเปอร์เซ็นต์

3.6.4 การทำงานของสัญญาณ PWM

- เส้นที่ 2 แสดงสัญญาณ PWM ที่ 25% duty cycle คือ สัญญาณในการออนจะเป็น 25% ของคาบสัญญาณ และ จะออฟเป็น 25% ของคาบสัญญาณ
- เส้นที่ 3 แสดงสัญญาณ PWM ที่ 50% duty cycle คือ สัญญาณในการออนจะเป็น 50% ของคาบสัญญาณ และ จะออฟเป็น 50% ของคาบสัญญาณ
- เส้นที่ 4 แสดงสัญญาณ PWM ที่ 75% duty cycle คือ สัญญาณในการออนจะเป็น 75% ของคาบสัญญาณ และ จะออฟเป็น 75% ของคาบสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เส้นที่ 5 แสดงสัญญาณ PWM ที่ 100% duty cycle คือ สัญญาณในการออนจะเป็น 100% ของคาบสัญญาณ และ จะออฟเป็น 0% ของคาบสัญญาณ



รูปที่ 3.10 แสดงสัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆกัน

3.6.5 เหตุผลที่ใช้ PWM ในการควบคุมความเร็วมอเตอร์

- PWMง่ายในการอินเตอร์เฟสกับไมโครคอนโทรลเลอร์ และใช้เพียงแค่เอาท์พุตสัญญาณเดียวในการควบคุมความเร็ว
- ไมโครคอนโทรลเลอร์ที่ใช้มีฟังก์ชั่นในการทำงานแบบ PWM โดยไมโครคอนโทรลเลอร์จะเปลี่ยนสัญญาณอินพุตที่รับเข้ามาให้เป็นสัญญาณเอาท์พุตแบบ PWM
- เมื่อมีการควบคุมกำลังของมอเตอร์จะทำให้แขนกลเคลื่อนที่ในทิศทางต่างๆได้อย่างราบเรียบและแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ทำการทดสอบหาค่าประสิทธิภาพของตัวหุ่นยนต์โดยดูการเคลื่อนที่ของหุ่นยนต์ไปยังเป้าหมายการทดลองนี้จะเป็นการตรวจเช็คว่าการทำงานของตัวหุ่นยนต์ยังมีข้อบกพร่องที่จะต้องแก้ไขบ้างหรือไม่ และจะทำการแก้ไขอย่างไรจึงจะทำให้การทำงานของตัวหุ่นยนต์ได้มีประสิทธิภาพมากที่สุด

4.1 การทดสอบควบคุมการเคลื่อนที่แบบสามมิติอย่างง่าย

เราจะทำการทดสอบโดยการบังคับด้วยมือให้หุ่นยนต์เคลื่อนที่จากจุดเริ่มต้นไปยังจุดที่กำหนดแล้วทำการเก็บค่าที่วัดได้ของแต่ละมอเตอร์ไว้ นำหุ่นยนต์ไปยังจุดเริ่มต้นและให้ทำงานอัตโนมัติตามเงื่อนไขที่เรากำหนดเป็นจำนวน 10 ครั้งและจะได้ค่าระยะที่วัดได้จากการเคลื่อนที่โดยอัตโนมัติตามที่เราต้องการเทียบกับค่าที่บังคับด้วยตัวควบคุมแบบมือถือ โดยที่เราให้มือจับของแขนกลเป็นจุดอ้างอิง

จำนวนครั้ง	E_x(%)	E_y(%)	E_z(%)
1	5.194	0.413	2.898
2	1.887	0.826	3.458
3	2.532	2.092	0.299
4	3.846	4.273	0.298
5	2.194	1.667	0.593
6	1.219	2.521	3.179
7	0.613	1.214	2.046
8	1.887	0.408	0.887
9	1.818	1.612	2.046
10	0.621	2.008	2.616

ตารางที่ 4.1 แสดงค่าผิดพลาดที่ได้จากการคำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 การคำนวนหาค่าผิดพลาดของหุ่นยนต์

ค่าผิดพลาดแกน X

$$E_X = ((X_{เก่า} - X_{ครั้งที่\ 1})/X_{ครั้งที่\ 1})*100 \quad (1)$$

$$E_X = ((16.2 - 15.4)/15.4)*100$$

$$E_X = 5.194\%$$

ค่าผิดพลาดแกน Y

$$E_y = ((Y_{เก่า} - Y_{ครั้งที่\ 1})/Y_{ครั้งที่\ 1})*100 \quad (2)$$

$$E_y = ((24.4 - 24.3)/24.3)*100$$

$$E_y = 0.413\%$$

ค่าผิดพลาดแกน Z

$$E_Z = ((Z_{เก่า} - Z_{ครั้งที่\ 1})/Z_{ครั้งที่\ 1})*100 \quad (3)$$

$$E_Z = ((33.5 - 34.5)/34.5)*100$$

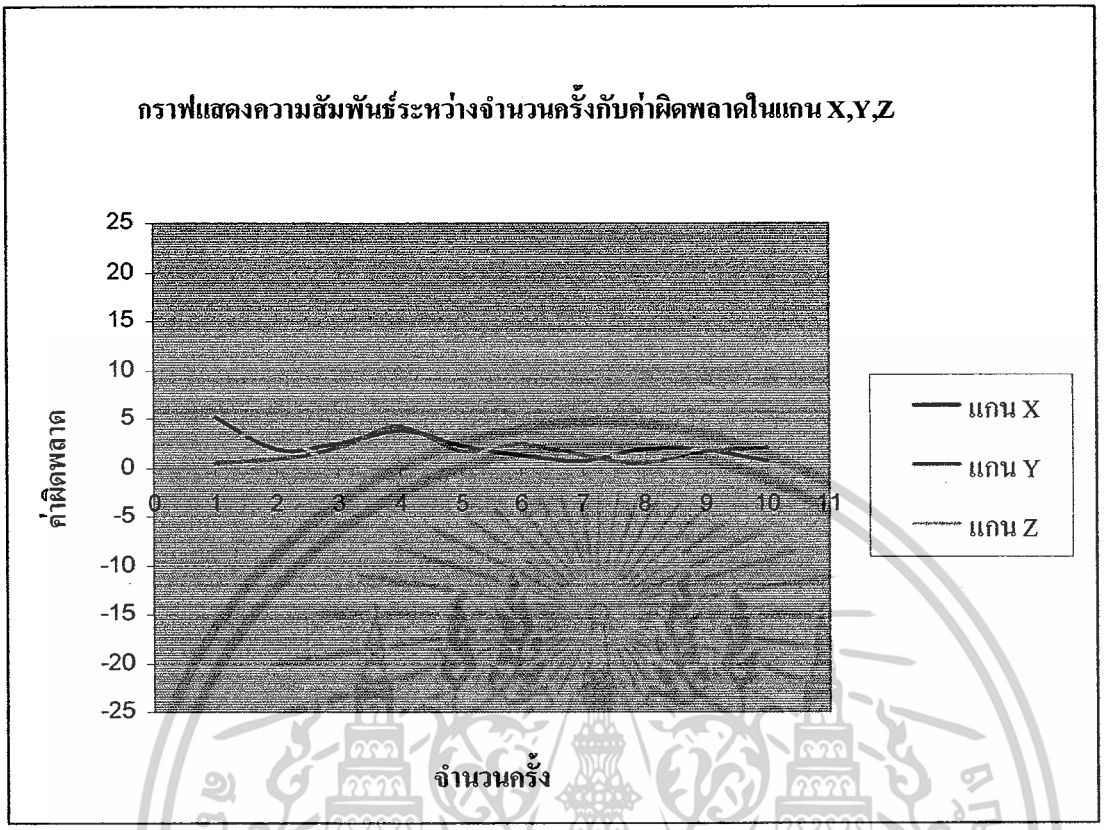
$$E_Z = 2.898\%$$

จากสมการที่ (1) ค่า $X_{เก่า}$ คือค่าระยะที่วัดได้จากการบังคับด้วยตัวควบคุมแบบมือถือในแนวแกน X ค่า X ครั้งที่ 1 คือค่าระยะที่วัดได้จากการทำงานอัตโนมัติโดยควบคุมด้วยคอมพิวเตอร์ในแนวแกน X ส่วนค่า X ครั้งที่ 2, 3, 4, ... ,10 คำนวนเช่นเดียวกับกับค่า X ครั้งที่ 1

จากสมการที่ (2) ค่า $Y_{เก่า}$ คือค่าระยะที่วัดได้จากการบังคับด้วยตัวควบคุมแบบมือถือในแนวแกน Y ค่า Y ครั้งที่ 1 คือค่าระยะที่วัดได้จากการทำงานอัตโนมัติโดยควบคุมด้วยคอมพิวเตอร์ในแนวแกน Y ส่วนค่า Y ครั้งที่ 2, 3, 4, ... ,10 คำนวนเช่นเดียวกับกับค่า Y ครั้งที่ 1

จากสมการที่ (3) ค่า $Z_{เก่า}$ คือค่าระยะที่วัดได้จากการบังคับด้วยตัวควบคุมแบบมือถือในแนวแกน Z ค่า Z ครั้งที่ 1 คือค่าระยะที่วัดได้จากการทำงานอัตโนมัติโดยควบคุมด้วยคอมพิวเตอร์ในแนวแกน Z ส่วนค่า Z ครั้งที่ 2, 3, 4, ... ,10 คำนวนเช่นเดียวกับกับค่า Z ครั้งที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



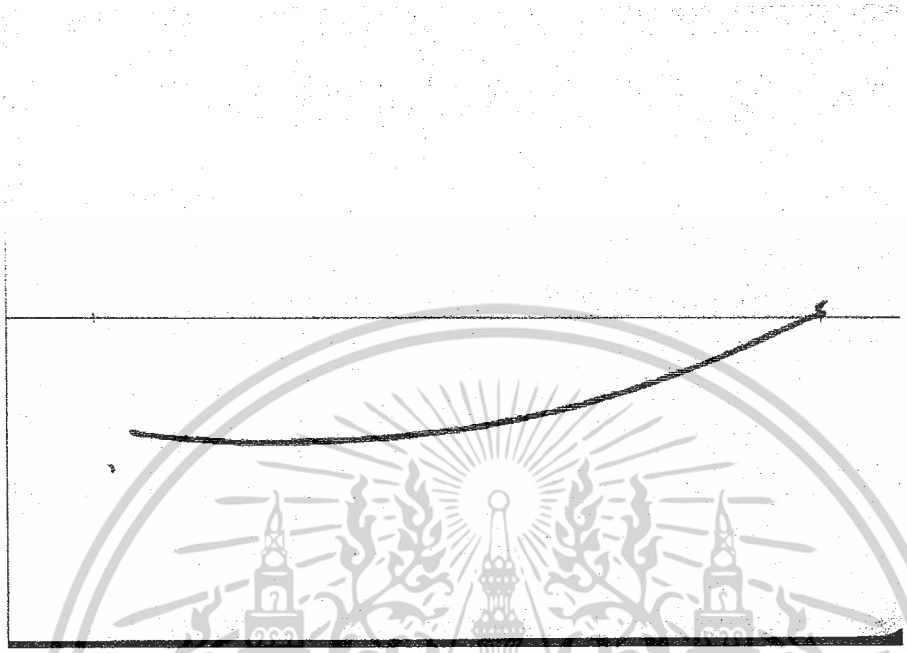
รูปที่ 4.1 กราฟแสดงความสัมพันธ์ระหว่างจำนวนครั้งกับค่าผิดพลาดในแกน X, Y, Z

จากกราฟ แกน X จะเห็นได้ว่าค่าผิดพลาดสูงสุดมีค่าเท่ากับ 5.194 และค่าต่ำสุดมีค่าเท่ากับ 0.613
จากกราฟ แกน Y จะเห็นได้ว่าค่าผิดพลาดสูงสุดมีค่าเท่ากับ 4.273 และค่าต่ำสุดมีค่าเท่ากับ 0.408
จากกราฟ แกน Z จะเห็นได้ว่าค่าผิดพลาดสูงสุดมีค่าเท่ากับ 3.458 และค่าต่ำสุดมีค่าเท่ากับ 0.298

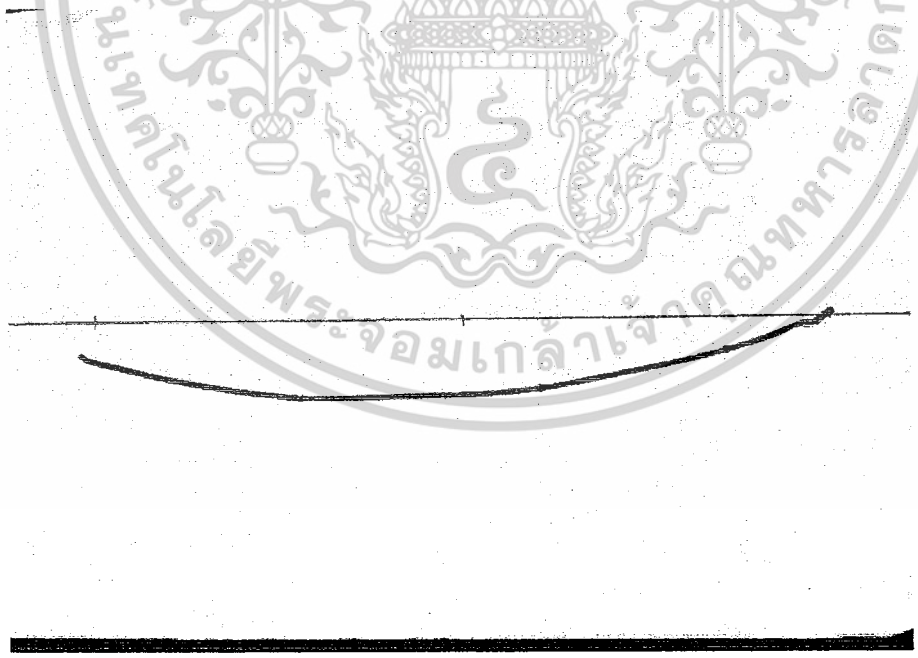
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบควบคุมการเคลื่อนที่ของแขนกลในแนวระนาบเป็นเส้นตรง

การทดสอบให้แขนกลเขียนเป็นเส้นตรงโดยกำหนดจุดเป็น 2 , 3และ 5 จุด

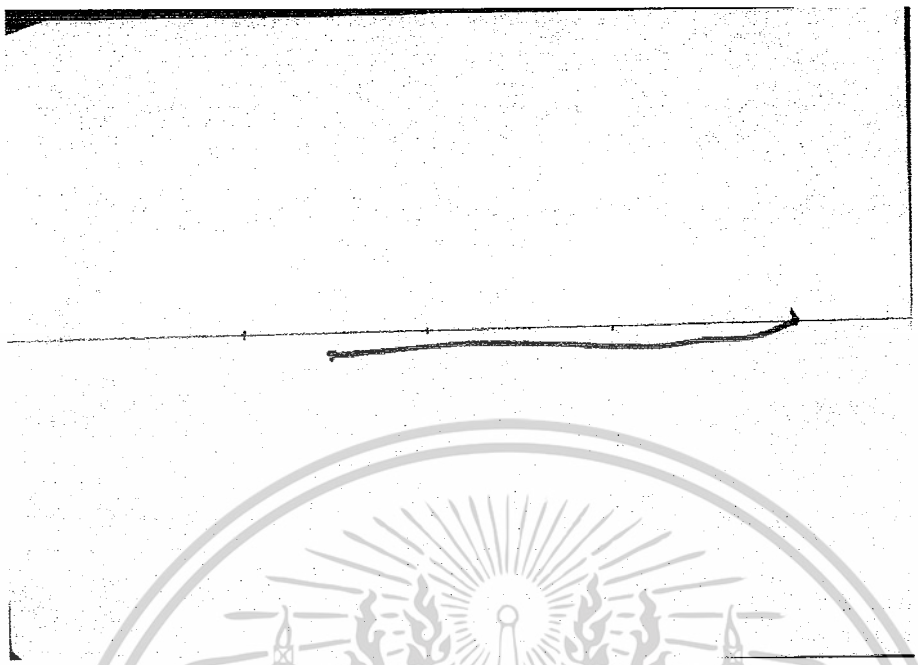


รูปที่4.2 แขนกลเขียนเส้นตรงโดยที่กำหนดจุด 2 จุด ซึ่งเส้นที่ได้ยังโค้งอยู่มาก



รูปที่4.3 แขนกลเขียนเส้นตรงโดยที่กำหนดจุด 3 จุด ซึ่งเส้นที่ได้โค้งน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.4 แขนกลเขียนเส้นตรงโดยที่กำหนดจุด 5 จุด ซึ่งเส้นที่ได้โค้งน้อยมาก

จากรูปที่ 4.2, 4.3, 4.4 เราได้ทำการทดสอบให้เขียนเป็นเส้นตรง จะเห็นว่ารูปที่ 4.2 จะมีค่าผิดพลาดมาก เนื่องจากมีความละเอียดของช่วงที่ทำการเฉลี่ยค่าน้อย ส่วนรูปที่ 4.3 เราได้เพิ่มช่วงที่ทำการเฉลี่ยค่าเป็น 3 จุด จะเห็นได้ว่าค่าผิดพลาดลดลงเส้นเริ่มตรงมากขึ้น รูปที่ 4.4 มีผิดพลาดน้อยที่สุดเส้นตรงตรงกว่ารูปที่ 4.2 และ 4.3 ดังนั้นจะเห็นได้ว่าแขนกลทำงานได้ตามที่เรากำหนดเงื่อนไข แต่จะมีค่าผิดพลาดอยู่บางส่วน เนื่องจากความฝืดของหัวปากกาที่ใช้ในการเขียนเส้นตัวโปรมแกรมที่ใช้ในการควบคุม

การทดสอบให้แขนกลเขียนเป็นเส้นตรงโดยที่บังคับแขนกลให้ตัวแขนกลเก็บค่าไปตามจุดที่เราได้กำหนดไว้ สั่งให้แขนกลทำงานอัตโนมัติไปตามจุดที่เรากำหนด โดยที่เราได้นำเส้นตรงที่กำหนด 5จุดมาทำการทดสอบ นำค่าที่วัดได้จากเส้นที่เขียนโดยอัตโนมัติ ไปหาสมการเส้นตรงและพ็อตกราฟจากสมการเส้นตรง

$$y = ax + b$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกน X(cm)	แกน Y(cm)
0	0.1
2	-0.2
4	-0.3
6	-0.4
8	-0.3
10	-0.2
12	-0.1
14	-0.2
16	-0.3
18	0.1

ตารางที่4.2 แสดงค่าXและYที่ได้จากการเทียบกับเส้นที่กำหนด

นำค่าที่วัดได้ไปจากเส้นที่เขียนเส้นตรงโดยอัตโนมัติ ไปพ็อตกราฟ จะได้กราฟดังต่อไปนี้

รูปที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่างX และ Y เป็นเส้นตรงที่แขนกลเขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟเราจะได้สมการเส้นตรงดังนี้

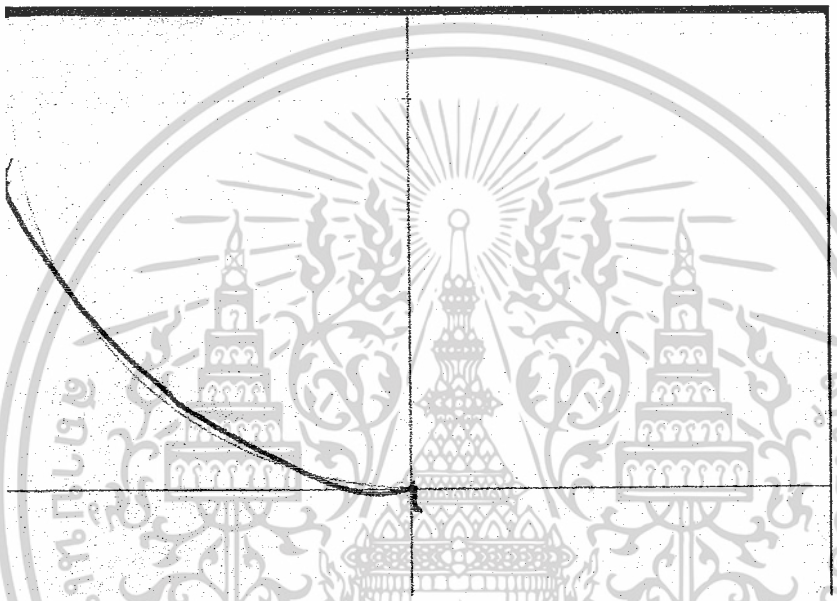
$$y = 0.0024242x + (-0.20182)$$

a = 0.0024242

b = -0.20182

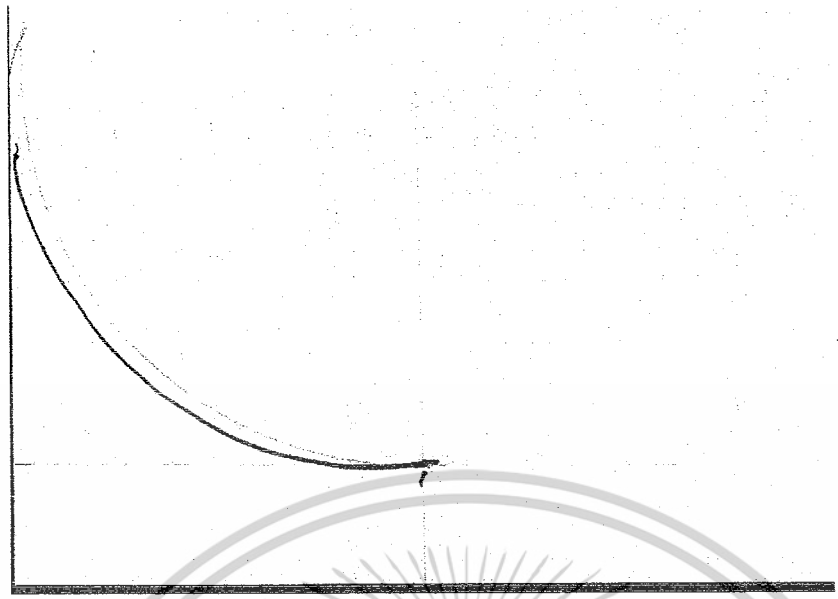
4.3 การทดสอบควบคุมการเคลื่อนที่ของแขนกลในแนวระนาบเป็นเส้นโค้ง

การทดสอบให้แขนกลเขียนเป็นเส้นพาราโบลาโดยกำหนดจุดเป็น 2 , 3และ 5 จุด

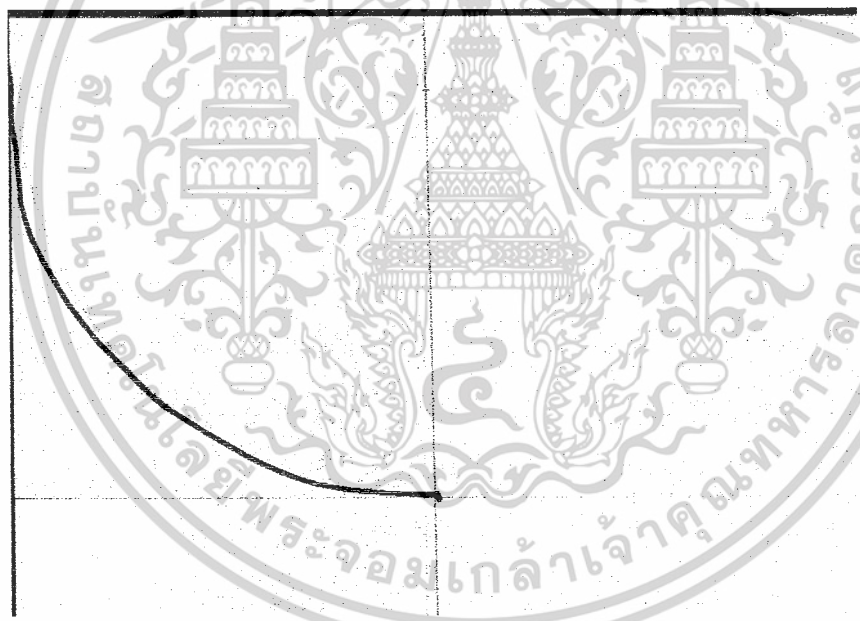


รูปที่4.6 แขนกลเขียนเส้นพาราโบลาโดยที่กำหนดจุด 2 จุด ซึ่งเส้นที่ได้ยังไม่ใกล้เคียงกับเส้นที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่4.7 แขนกลเขียนเส้นพาราโบลาโดยที่กำหนดจุด 3 จุด ซึ่งเส้นที่ได้เริ่มใกล้เคียงกับเส้นที่กำหนด



รูปที่4.8 แขนกลเขียนเส้นพาราโบลาโดยที่กำหนดจุด 5 จุด ซึ่งเส้นที่ได้ใกล้เคียงกับเส้นที่กำหนดมากขึ้น

จากรูปที่ 4.6, 4.7, 4.8 เราได้ทำการทดสอบให้เขียนเป็นเส้นพาราโบลาโดยกำหนดจุดเป็น 2 , 3 และ5 จุด จากรูปที่ 1 จะมีค่าผิดพลาดมากเนื่องจากมีความละเอียดของช่วงที่ทำการเฉลี่ยค่าอยู่แค่ 2 จุด ส่วนรูปที่ 4.7 เรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้เพิ่มช่วงที่ทำการเฉลี่ยค่าเป็น 3 จุดจะเห็นได้ว่าค่าผิดพลาดลดลงเส้นเริ่มโค้งมากขึ้น รูปที่ 4.8 มีผิดพลาดน้อยที่สุดเส้นที่เขียนมีความโค้งเท่ากับเส้นที่กำหนดมากกว่ารูปที่ 4.6 และ 4.7

การทดสอบโดยที่เราได้กำหนดจุดไว้แล้วและทำการกำหนดรูปพาราโบลาโดยที่กำหนดจุด 5 จุด แล้วให้แขนกลเขียนตามจุดหรือรูปที่เราได้กำหนดไว้ ในการทดสอบนี้เราให้แขนกลเขียนเป็นครึ่งพาราโบลาและค่าสมการพาราโบลาที่เรากำหนดคือ

$$y = ax^2 + bx + c$$

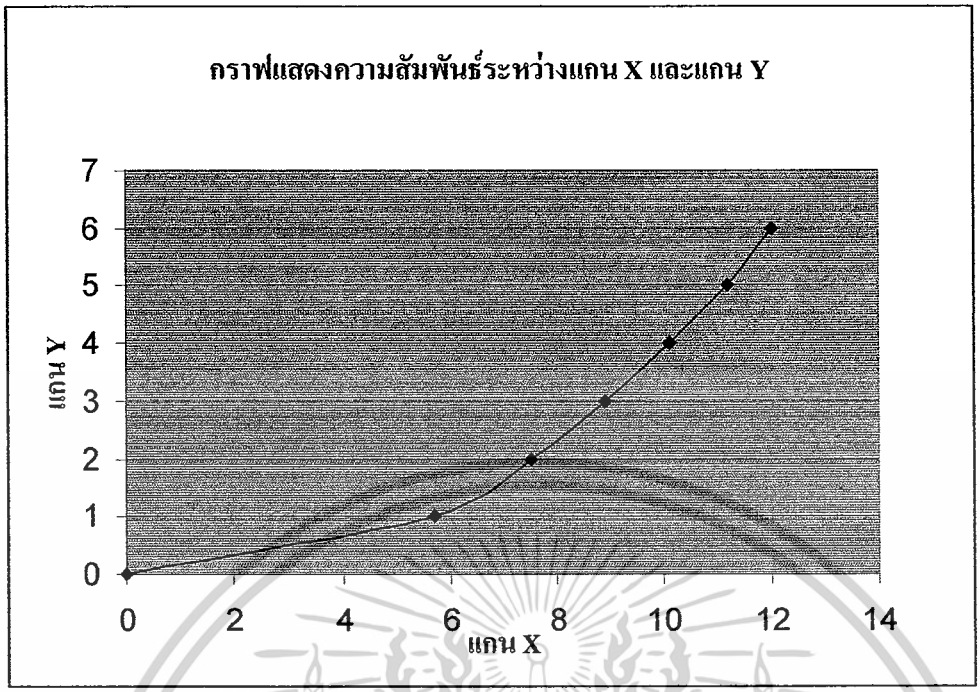
$$y = 0.058x^2 + (-0.217)x + 0.649$$

ค่าของครึ่งพาราโบลาที่แขนกลเขียนตามที่เรากำหนดอยู่ในตารางที่ 4.3 จากนั้นเรานำค่าที่วัดได้ไปพ็อตกราฟพาราโบลาดังรูปที่ 4.9

แกน X(cm)	แกน Y(cm)
0	0
5.7	1
7.5	2
8.9	3
10.1	4
11.2	5
12	6

ตารางที่4.3 แสดงค่า X และ Y ที่ได้จากการวัดที่แขนกลเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 กราฟแสดงความสัมพันธ์ระหว่างX และ Y เป็นเส้นพาราโบลาที่แขนกลเขียนขึ้น

จากกราฟเราจะได้สมการพาราโบลาที่แขนกลเขียนดังนี้

$$y = 0.051x^2 + (- 0.118)x + 0.0045$$

a = 0.051

b = - 0.118

c = 0.0045

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิเคราะห์และสรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากวัตถุประสงค์ของโครงงานต้องการให้หุ่นยนต์ทำงานตามเงื่อนไขที่กำหนดโดยการบังคับด้วยมือและให้โปรแกรมที่ใช้จดจำค่า Pulse ของมอเตอร์และใช้โปรแกรม Visual Basic (VB) ส่งค่าที่จดจำไว้ไปยังไมโครคอนโทรลเลอร์เพื่อสั่งให้ไดร์ฟขับทำงาน หุ่นยนต์จึงสามารถเคลื่อนที่โดยอัตโนมัติ การควบคุมหุ่นยนต์โดยใช้โปรแกรม Pulse Width Modulation (PWM) ทำให้การเคลื่อนที่ของหุ่นยนต์มีความเร็วและระยะทางไม่คงที่ แผงไดร์ฟขับถูกสร้างขึ้นใหม่เพื่อให้เหมาะสมกับการใช้งานกับหุ่นยนต์มากที่สุด โดยส่วนของไมโครคอนโทรลเลอร์ใช้แหล่งพลังงานหนึ่งและส่วนของไดร์ฟขับใช้อีกแหล่งพลังงานหนึ่ง ไดร์ฟขับต้องรับภาระโหลดของหุ่นยนต์มากจึงทำให้เกิดความร้อน หุ่นยนต์สามารถเคลื่อนที่ได้อย่างอัตโนมัติแต่ในบางครั้งหุ่นอาจจะมีการเคลื่อนที่เบี่ยงเบนออกจากเป้าหมายไปบ้าง ไมโครคอนโทรลเลอร์มีจำนวนน้อยเกินไปทำให้มอเตอร์เคลื่อนที่ได้ทิศทางเดียวขาดไมโครคอนโทรลเลอร์อีกหนึ่งตัวเพราะไมโครคอนโทรลเลอร์มีราคาสูง

5.2 แนวทางในการแก้ไข

เนื่องจากการควบคุมหุ่นยนต์ใช้โปรแกรม Pulse Width Modulation (PWM) จึงทำให้การเคลื่อนที่ของหุ่นยนต์มีความเร็วไม่คงที่ จึงต้องปรับความเร็วที่เปอร์เซ็นต์ PWM ทุกครั้งที่ทิศทางเปลี่ยนแปลงไป ดังนั้นต้องทราบค่ามุมและความเร็วทุกข้อของหุ่นยนต์ เพื่อนำค่าที่ทราบนั้นไปกำหนดความเร็ว โดยส่วนของโปรแกรมที่ใช้จดจำค่า ควรที่จะจดจำค่าความเร็วในการหมุนของมอเตอร์ด้วย จะทำให้ความเร็วของแต่ละมอเตอร์มีค่าคงที่

แผงวงจรไดร์ฟขับตัวไดร์ฟขับต้องรับภาระโหลดของหุ่นยนต์มากจึงทำให้ไดร์ฟขับเกิดการเสียหายบ่อยครั้ง ควรหาชุดไดร์ฟขับที่เหมาะสมกับการใช้กับหุ่นยนต์ ควรเพิ่มไมโครคอนโทรลเลอร์ให้มีจำนวนมากขึ้นเพื่อให้มอเตอร์เคลื่อนที่ได้สองทิศทางและควรหาไมโครคอนโทรลเลอร์ที่มีราคาถูกลงและมีความทนทาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] โชติพันธุ์ หล่อเลิศสุนทรและฐิตะพันธุ์ หล่อเลิศสุนทร ,“สอนเขียน Visual Basic 6.0 ให้เป็นProject”, Soft Express&Publishing , 2543

[2] ประจิน พลังสันติกุล ,“การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ AVR ด้วยภาษา C กับ WinAVR (C Compiler)”,แอพซอฟต์เทค , 2549

[3] อภิชาติ ภู่พลับ ,“เริ่มต้นเขียนโปรแกรมติดต่อกันและควบคุมฮาร์ดแวร์ด้วย Visual Basic 6”, Infopress Develoop Book, 2546

[4] http://www.hearnecber.com/industrial/JB32454.html-9k

[5] http://www.sti-sl.es

[6] http://www.robocluster.dk/luna/lunaz/luna_projecter/scorbot/

[7] http://www.berufskollegdinslaken.de/projekte/scorbot/Default.htm

[8] http://www.techniker-forum.de/technik-53/roboter-8713.html

[9] http://www.strategicalliance.niu.edu/PMT.html

[10] http://www.eng.mut.ac.th/Mechatronics/Article_detail.asp%

[11] http://www.thaiwbi.com

[12] http://www.spu.ac.th/~ktm/chapter11.html

[13] http://www-cdr.stanford.edu/NextLink/Expert.html

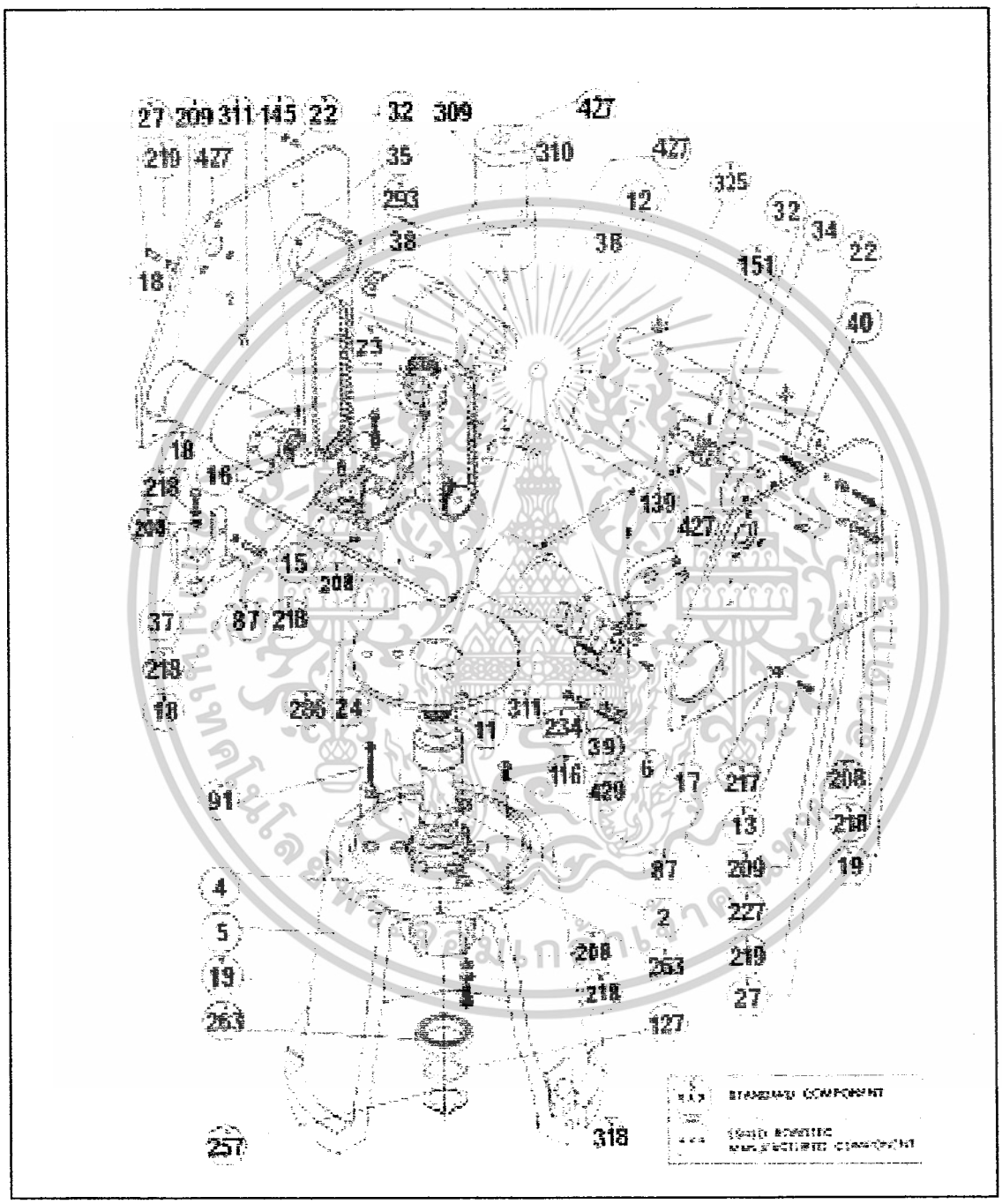
[14] http://www.9engineer.com/Industrial%20Robot.htm

[15] http://www.en.kku.ac.th

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

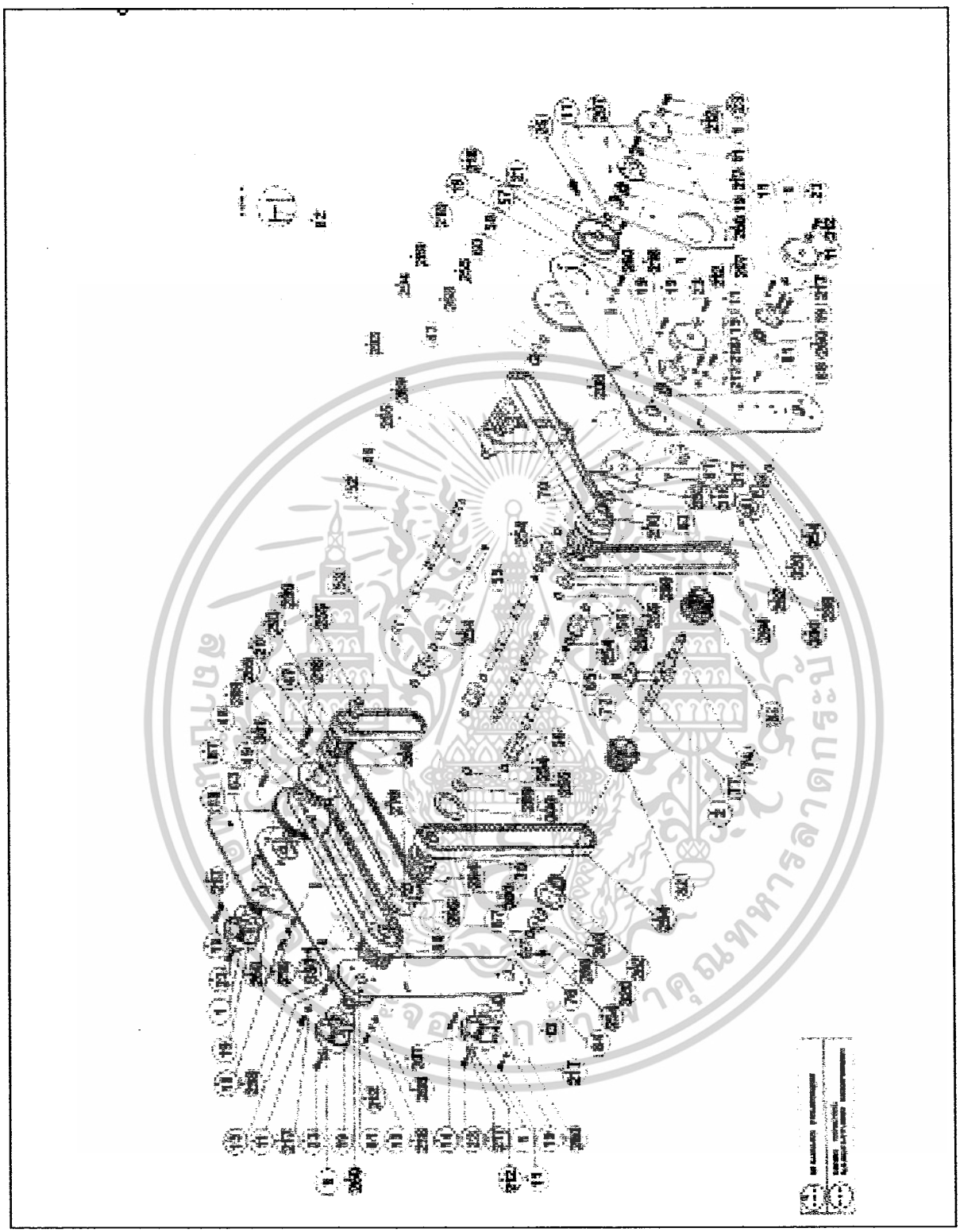
ภาคผนวก ก

ส่วนประกอบของหุ่นยนต์ SCORBOT- ER III



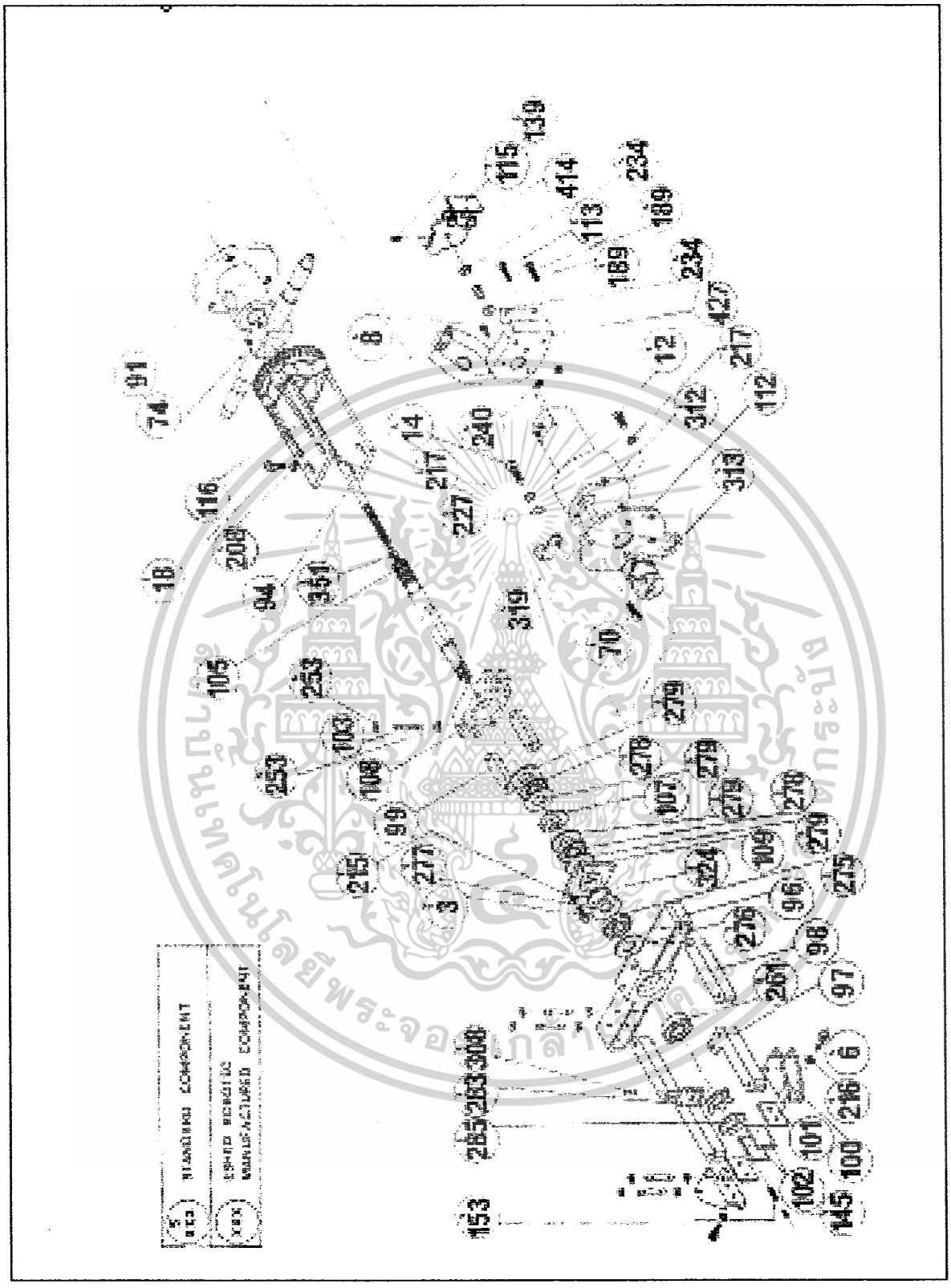
รูปที่ ก-1 ส่วนประกอบของแขนกล SCORBOT ER-III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-2 ส่วนประกอบของแขนกล SCORBOT ER-III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-3 ส่วนประกอบของแขนกล SCORBOT ER-III

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

วิชวลเบสิกซอร์สโค้ด (Visual Basic Source Code)

Program

```
Private Sub Command1_Click()
If MSComm1.PortOpen = False Then
   MSComm1.PortOpen = True
End If
Count_Num_Motor = 0
Timer1.Enabled = True
End Sub

Private Sub Command2_Click()
MSComm1.PortOpen = False
Timer1.Enabled = False
End Sub

Private Sub Command3_Click()

Num_Count_x = Num_Count_x + 1

      Len_A(Num_Count_x) = Count_A - Len_A(Num_Count_x - 1)
      Len_B(Num_Count_x) = Count_B - Len_B(Num_Count_x - 1)
      Len_C(Num_Count_x) = Count_C - Len_C(Num_Count_x - 1)
      Len_D(Num_Count_x) = Count_D - Len_D(Num_Count_x - 1)
      Len_E(Num_Count_x) = Count_E - Len_E(Num_Count_x - 1)
      Len_F(Num_Count_x) = Count_F - -Len_F(Num_Count_x - 1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Text8.Text = Text8.Text & Len_A(Num_Count_x) & Len_B(Num_Count_x) &
Len_C(Num_Count_x) & Len_D(Num_Count_x) & Len_E(Num_Count_x) & Len_F(Num_Count_x)
        Command5.Caption = Num_Count_x


End Sub

Private Sub Command4_Click()
If MSComm3.PortOpen = False Then
   MSComm3.PortOpen = True
End If
If MSComm1.PortOpen = False Then
   MSComm1.PortOpen = True
End If
' Sent Code
Call Cal
End Sub
 Public Sub Cal()
 Dim Sent As String
Dim score(6) As Integer
Dim i As Integer, max As Integer

 Count_A = 0
 Count_B = 0
 Count_C = 0
 Count_D = 0
 Count_E = 0

score(1) = Len_A(Num_Count_y)
score(2) = Len_B(Num_Count_y)
score(3) = Len_C(Num_Count_y)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
score(4) = Len_D(Num_Count_y)
score(5) = Len_E(Num_Count_y)
score(6) = Len_F(Num_Count_y)

For i = 1 To 6
   If score(i) > max Then
      max = score(i)
   End If
Next i
Command5.Caption = max
If max = 0 Then
   max = 1
End If

Text9.Text = Len_A(Num_Count_y)
Text10.Text = Len_B(Num_Count_y)
Text11.Text = Len_C(Num_Count_y)
Text12.Text = Len_D(Num_Count_y)
Text13.Text = Len_E(Num_Count_y)
Text14.Text = Len_F(Num_Count_y)

Compair_Encoder_A = Len_A(Num_Count_y)
Compair_Encoder_B = Len_B(Num_Count_y)
Compair_Encoder_C = Len_C(Num_Count_y)
Compair_Encoder_D = Len_D(Num_Count_y)
Compair_Encoder_E = Len_E(Num_Count_y)

 If Compair_Encoder_B < 10 Then
    Compair_Encoder_B = 1000
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
M = 45 * (Len_A(Num_Count_y) / max) \ 1
N = 50 * (Len_B(Num_Count_y) / max) \ 1
O = 30 * (Len_C(Num_Count_y) / max) \ 1
P = 50 * (Len_D(Num_Count_y) / max) \ 1
Q = 50 * (Len_E(Num_Count_y) / max) \ 1
R = 50 * (Len_F(Num_Count_y) / max) \ 1
'M = 50
'N = 0
'O = 0
'P = 0
'Q = 50
'R = 0
'Command6.Caption = Chr(M)
MSComm3.Output = "A" & Chr(M) & Chr(N) & Chr(O) & Chr(P) & Chr(Q) & Chr(R)
Count_Num_Motor = 1
Timer1.Enabled = True
End Sub
'Text8.Text = Text8.Text & Chr(U) & Chr(P) & Chr(T) & Chr(G) & Chr(K) & Chr(F)
Public Sub Count_Number()
If MSComm3.PortOpen = False Then
   MSComm3.PortOpen = True
End If
If MSComm1.PortOpen = False Then
   MSComm1.PortOpen = True
End If


    For i = 1 To 3
       MSComm3.Output = "A" & Chr(0) & Chr(0) & Chr(0) & Chr(0) & Chr(0) & Chr(0)
       Timer1.Enabled = False
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next i
Command6.Caption = Num_Count_y
Num_Count_y = Num_Count_y + 1
If Num_Count_y - 1 < Num_Count_x Then
For i = 1 To 3
  Call Command4_Click
Next i
End If
End Sub


Private Sub Command7_Click()
Num_Count_y = 0
 Count_A = 0
 Count_B = 0
 Count_C = 0
 Count_D = 0
 Count_E = 0
Command4_Click
End Sub

Private Sub Form_Load()
Num_Count = 0
Num_Count_y = 1
End Sub


Private Sub Timer1_Timer()
Dim InString As String

'à¡ç°¤èÒà¾ÕÂ§µÑÇà´ÕÂÇ¹Ñ°¨Ò¡ ºÑ¿à¿ÍÃì·Ñé§ËÁ´·Õè¨ÁÕ
If MSComm3.PortOpen = False Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MSComm3.PortOpen = True
End If
If MSComm1.PortOpen = False Then
    MSComm1.PortOpen = True
End If
 MSComm1.InputLen = 1
 'ÃÍÃÑº¤èÒ·Õè
 If MSComm1.InBufferCount Then
 InString = MSComm1.Input
If InString = "a" Then
  Count_A = Count_A + 1
 Text1.Text = Count_A
End If
If InString = "b" Then
    Count_B = Count_B + 1
    Text2.Text = Count_B
End If
If InString = "c" Then
  Count_C = Count_C + 1
Text3.Text = Count_C
  End If
 If InString = "d" Then
    Count_D = Count_D + 1
Text4.Text = Count_D
End If
'If InString = "e" Then
 '  Count_E = Count_E + 1
  'Text5.Text = Count_E
'End If
'If InString = "f" Then
    'Count_F = Count_F + 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'Text6.Text = Count_F
'End If
'áÊ´§¤èÒ·ÕèàÃÒä´éÃÑºÁÒÍí¡ä»·Õè Text ¢Í§ªèÍ§·Ò§ÃÑº
'Rx.Text = Rx.Text + InString
 End If
If Count_Num_Motor Then
 If Count_A > Compair_Encoder_A Then
      M = 0
      Call Count_Number
 End If
 If Count_B > Compair_Encoder_B Then
      N = 0
      Call Count_Number
 End If
 If Count_C > Compair_Encoder_C Then
      O = 0
      Call Count_Number
 End If
 If Count_D > Compair_Encoder_D Then
      P = 0
      Call Count_Number
 End If
 If Count_E > Compair_Encoder_E Then
      Q = 0
       Call Count_Number
 End If
 If Count_F > 100 Then
      R = 0
      Call Count_Number
 End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
End Sub
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ซีซอร์สโค้ด(C Source Code)

……..AVR 1……..

```
This program was produced by the
CodeWizardAVR V1.24.7e Evaluation
Automatic Program Generator
© Copyright 1998-2005 Pavel Haiduc, HP InfoTech s.r.l.
http://www.hpinfotech.com
e-mail:office@hpinfotech.com

Project :
Version :
Date    : 28/5/2008
Author  : Freeware, for evaluation and non-commercial use only
Company :
Comments:


Chip type           : ATmega128
Program type        : Application
Clock frequency     : 16.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 1024
*****************************************************/

#include <mega128.h>
#include <stdio.h>
//unsigned int A,B,C,D,E,F;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void delay (void)
{
   unsigned int i,j;
   for(i=0;i<100;i++)
   for(j=0;j<100;j++);
}
// Declare your global variables here
// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
    printf("a");
  //delay();
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here
    printf("b");
  //delay();
}

// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here
   printf("c");
   //delay();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// External Interrupt 3 service routine
interrupt [EXT_INT3] void ext_int3_isr(void)
{
// Place your code here
    //printf("d");
   //delay();
}

// External Interrupt 4 service routine
interrupt [EXT_INT4] void ext_int4_isr(void)
{
// Place your code here
    //printf("e");
      //delay();
}

// External Interrupt 5 service routine
interrupt [EXT_INT5] void ext_int5_isr(void)
{
// Place your code here
   // printf("f");
      //delay();
}

// Standard Input/Output functions


void main(void)
{
// Declare your local variables here
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTE=0x00;
DDRE=0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTF=0x00;
DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer 3 Stopped
// Mode: Normal top=FFFFh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: On
// INT1 Mode: Rising Edge
// INT2: On
// INT2 Mode: Rising Edge
// INT3: On
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// INT3 Mode: Rising Edge
// INT4: On
// INT4 Mode: Rising Edge
// INT5: On
// INT5 Mode: Rising Edge
// INT6: Off
// INT7: Off
EICRA=0xFF;
EICRB=0x0F;
EIMSK=0x3F;
EIFR=0x3F;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 9600
UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
#asm("sei")

while (1)
    {
    // Place your code here

    };
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

.....AVR 2 .....

This program was produced by the

CodeWizardAVR V1.24.7e Evaluation

Automatic Program Generator

© Copyright 1998-2005 Pavel Haiduc, HP InfoTech s.r.l.

http://www.hpinfotech.com

e-mail:office@hpinfotech.com

Project :

Version :

Date : 3/8/2007

Author : Freeware, for evaluation and non-commercial use only

Company :

Comments:

Chip type : ATmega128

Program type : Application

Clock frequency : 16.000650 MHz

Memory model : Small

External SRAM size : 0

Data Stack size : 1024

*****/

```
#include <mega128.h>

// External Interrupt 0 service routine
interrupt [EXT_INT0] void ext_int0_isr(void)
{
// Place your code here
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

// External Interrupt 1 service routine
interrupt [EXT_INT1] void ext_int1_isr(void)
{
// Place your code here

}

// External Interrupt 2 service routine
interrupt [EXT_INT2] void ext_int2_isr(void)
{
// Place your code here

}

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Get a character from the USART1 Receiver
#pragma used+
char getchar1(void)
{
char status,data;
while (1)
    {
    while (((status=UCSR1A) & RX_COMPLETE)==0);
    data=UDR1;
    if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
      return data;
    };
}
#pragma used-

// Write a character to the USART1 Transmitter
#pragma used+
void putchar1(char c)
{
while ((UCSR1A & DATA_REGISTER_EMPTY)==0);
UDR1=c;
}
#pragma used-

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here
 unsigned char dat[8];
void main(void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=Out Func6=Out Func5=Out Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=0 State6=0 State5=0 State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0xE0;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=Out Func4=Out Func3=Out Func2=In Func1=In Func0=In
// State7=T State6=T State5=0 State4=0 State3=0 State2=T State1=T State0=T
PORTE=0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DDRE=0x38;

// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTF=0x00;
DDRF=0x00;

// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 15.626 kHz
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// OC1C output: Non-Inv.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0xA9;
TCCR1B=0x05;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
OCR1CH=0x00;
OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Clock value: 15.626 kHz
// Mode: Ph. correct PWM top=00FFh
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Non-Inv.
// OC3B output: Non-Inv.
// OC3C output: Non-Inv.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0xA9;
TCCR3B=0x05;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: On
// INT0 Mode: Rising Edge
// INT1: On
// INT1 Mode: Rising Edge
// INT2: On
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// INT2 Mode: Rising Edge
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x3F;
EICRB=0x00;
EIMSK=0x07;
EIFR=0x07;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;
ETIMSK=0x00;

// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 9600
UCSR0A=0x00;
UCSR0B=0x18;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x67;

// USART1 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART1 Receiver: On
// USART1 Transmitter: On
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// USART1 Mode: Asynchronous
// USART1 Baud rate: 9600
UCSR1A=0x00;
UCSR1B=0x18;
UCSR1C=0x06;
UBRR1H=0x00;
UBRR1L=0x67;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// Global enable interrupts
#asm("sei")

while (1)
    {
    // Place your code here
    unsigned char m;
     dat[0]=getchar();
    if (dat[0]=='A')
    {
     for(m=0;m<8;m++)
     {
          dat[m]=getchar();
     }
          OCR1AH=0x00;
          OCR1AL=dat[0];
          OCR1BH=0x00;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OCR1BL=dat[1];
        OCR1CH=0x00;
        OCR1CL=dat[2];
        OCR3AH=0x00;
        OCR3AL=dat[3];
        OCR3BH=0x00;
        OCR3BL=dat[4];
        OCR3CH=0x00;
        OCR3CL=dat[5];

    for(m=0;m<8;m++)
     {
       putchar(dat[m]);
     }
   }
   };
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

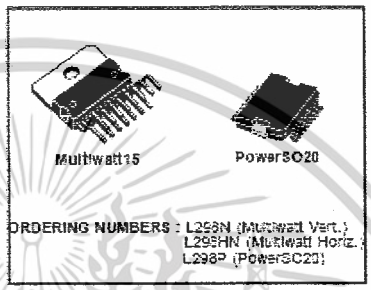
ภาคผนวก ง

Data Sheet L298N

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

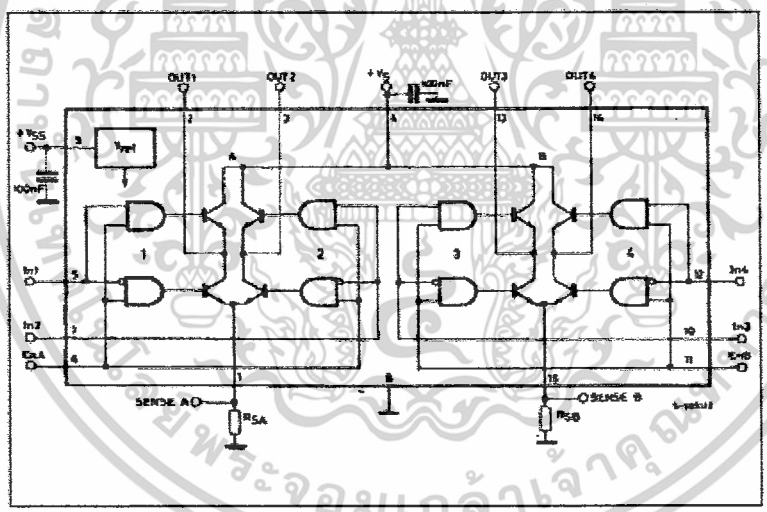


ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM



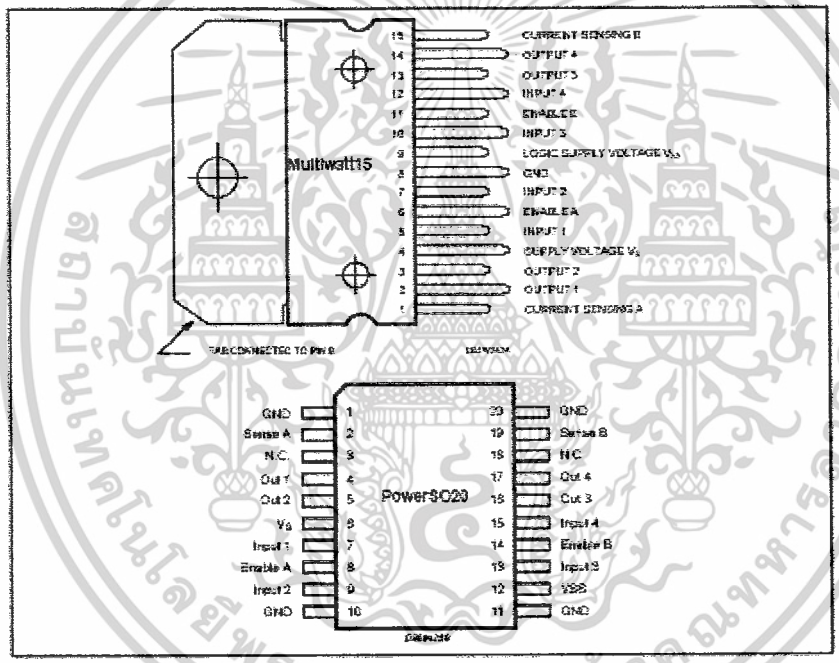
January 2000 1/13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100μs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\,j\text{-case}}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\,j\text{-amb}}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13,15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_J = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition		V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)			4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0	V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L	V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0	V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L	V_i = X			6	mA
V_{IL}	Input Low Voltage (pins 5, 7, 10, 12)			−0.3		1.5	V
V_{IH}	Input High Voltage (pins 5, 7, 10, 12)			2.3		V_{SS}	V
I_{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L				−10	µA
I_{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	V_i = H ≤ V_{SS} −0.6V			30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)			−0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)			2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L				−10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} −0.6V			30	100	µA
$V_{CE(sat)H}$	Source Saturation Voltage	I_L = 1A I_L = 2A		0.95	1.35 2	1.7 2.7	V V
$V_{CE(sat)L}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)		0.85	1.2 1.7	1.6 2.3	V V
$V_{CE(sat)}$	Total Drop	I_L = 1A (5) I_L = 2A (5)		1.60		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)			−1 (1)		2	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		µs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs

1) Sensing voltage can be −1 V for t ≤ 50 µsec; in steady state V_{sens} min ≥ −0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

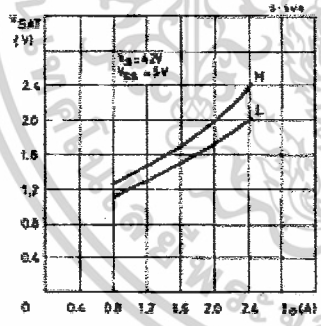
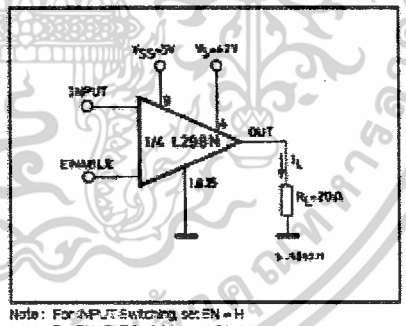


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

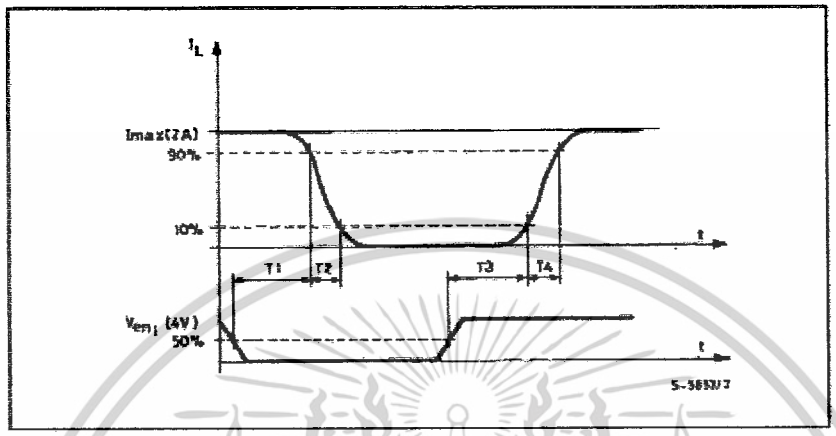
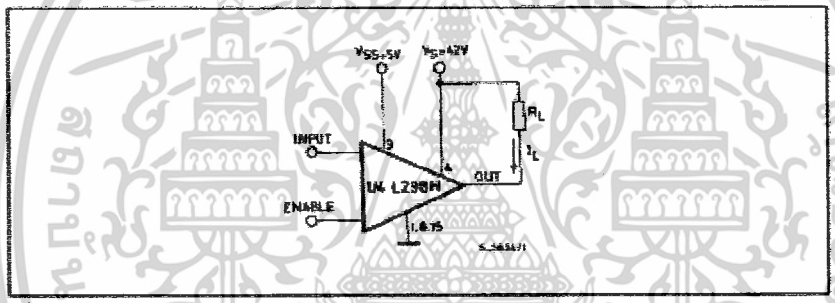


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching set EN = H
For ENABLE Switching, set IN = L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้