

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมแขนกล 5 แกน ด้วยคอมพิวเตอร์



เลขหมู่.....
เลขทะเบียน.....**83058**
วัน,เดือน,ปี...**31.08.2551**

b. 118 6089x
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**CONTROL SYSTEM FOR 5-AXIS ROBOT ARM
VIA COMPUTER**



MISS GRONGGAN GIRDPRACH

MR.NARA HANJAVANICH

MR.PATTANAPONG SINGHAPAT

MR.PARADON EAWIJETKUN

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INDUSTRIAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2007**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

หัวข้อปริญญาโท

ระบบควบคุมแขนกล 5 แกนด้วยคอมพิวเตอร์

CONTRL SYSTEM FOR 5-AXIS ROBOT ARM VIA COMPUTER

นักศึกษา

นางสาวกรรณิกา บุญเกิดปราษฎ์	รหัสประจำตัว	47010017
นายณรา หาญจวนิช	รหัสประจำตัว	47010362
นายพัฒนพงศ์ สิงหาปัด	รหัสประจำตัว	47010505
นายภราดร เอื้อวิจิตรกุล	รหัสประจำตัว	47010545

หลักสูตร

วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหกรรม

อาจารย์ผู้ควบคุมปริญญาโท

(ผศ.ดร. สรรพสิทธิ์ ถิ่นนรินทร์)

(ดร.อนิรุท ไชยจรรณิข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบควบคุมแขนกล 5 แกนด้วยคอมพิวเตอร์
นักศึกษา	นางสาวกรรณิกาญจน์ เกิดปราชญ์ นายนรา หาญจวนิช นายพัฒนพงศ์ สิงหาปัด นายภราดร เอื้อวิจิตรกุล
หลักสูตร	วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา	2550
อาจารย์ผู้ควบคุมปริญญานิพนธ์	ผศ.ดร. สรรพสิทธิ์ ลิ้มบรรดินันท์ ดร.อนิรุท ไชยจรรวนิช

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้จัดทำขึ้นโดยมีวัตถุประสงค์ เพื่อออกแบบวงจร และโปรแกรมควบคุมการทำงานของแขนกล 5 แกน โดยอาศัยหลักการทางวิชา ออโตเมชัน(Automation) และวิชาโรโบติก(Robotic) เพื่อให้สามารถเคลื่อนที่ได้ตามความต้องการ และเป็นระบบอัตโนมัติ โดยสั่งงานด้วยคอมพิวเตอร์ ผ่านทางพอร์ตอนุกรม (RS-232) ไปยังไมโครคอนโทรลเลอร์ PIC16F877A เพื่อควบคุมการทำงานของมอเตอร์ และการรับค่าแรงดันไฟฟ้าจากโพเทนทิโอมิเตอร์ ส่งกลับไปยังไมโครคอนโทรลเลอร์ PIC16F877A และคอมพิวเตอร์ เพื่อประมวลผลและรับคำสั่งการทำงานใหม่อีกครั้งหนึ่ง โดยใช้โปรแกรมภาษาซี ควบคุมการทำงานในไมโครคอนโทรลเลอร์ PIC16F877A และโปรแกรม Delphi สร้างส่วนอินเตอร์เฟซ (Interface) ติดต่อและควบคุมการทำงานของแขนกลผ่านทางคอมพิวเตอร์ ผลที่ได้รับจากการศึกษาครั้งนี้เป็นแนวทางในการพัฒนาศักยภาพการทำงานของ แขนกล 5 แกนด้วยคอมพิวเตอร์ ในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title Control System for 5 – Axis Robot Arm Via Computer
Student Miss Gronggan Girdprach
Mr. Nara Hanjawanich
Mr. Pattanapong Singhapat
Mr. Paradon aurwjitkun
Degree Bachelor of Engineering in Industrial Engineering
King Mongkut's Institute of Technology Ladkrabang
Academic Year 2007
Thesis Advisor Assistant Professor Dr. Sunpasit Limnarat
Dr. Amirut Chijaruwanich

ABSTRACT

The objective of this study is to design circuit and program for controlling 5 axis robot by using *automation and robotic* concepts. This robot is controlled from serial port (RS-232) to microcontroller PIC 16F877A by using C programming. Delphi program is also used to control interface part for operating robot arm through computer. The result of this study can be used as a concept to improve the potential of using 5 – axis robot in the future.

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง ระบบควบคุมแขนกล 5 แกน ด้วยคอมพิวเตอร์ สามารถสำเร็จลุล่วงไปได้ด้วยดี คณะผู้จัดทำขอกราบขอบพระคุณท่านอาจารย์ประจำภาควิชาวิศวกรรมอุตสาหกรรมทุกท่าน โดยเฉพาะอย่างยิ่ง ศศ.ดร. สรรพสิทธิ์ ลิ้มวรรณรัตน์ ที่กรุณาให้คำแนะนำ คำปรึกษา และกำลังใจที่ดียิ่งด้วยดีเสมอมา ดร.อนิรุท ไชยจารุวิช ที่กรุณาคอยให้คำแนะนำ และให้ความช่วยเหลือในด้านต่าง ๆ ดร.สกนธ์ คล่องบุญจิต ขอบพระคุณสำหรับคำชี้แนะที่ดีในการทำงาน

ขอบคุณเพื่อนทุกคนสำหรับความช่วยเหลือจนทำให้ปริญญานิพนธ์สำเร็จลุล่วง และคอยเป็นกำลังใจที่ดีตลอดมา



นางสาวกรรณกาญจน์ เกิดปราชญ์

นายนรา หาญจวิช

นายพัฒนพงศ์ สิงหาปัด

นายภราดร เอื้อวิจิตรกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ	
1.1 ความสำคัญของโครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	1
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	1
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 โพลเทนิโอมิเตอร์.....	2
2.2 การทำงานของพอร์ตอนุกรม.....	3
2.2.1 รูปแบบของข้อมูลอนุกรม.....	3
2.2.2 การเชื่อมต่อแบบมาตรฐาน RS-232.....	4
2.3 การควบคุมความเร็วมอเตอร์.....	7
2.3.1 การควบคุมแบบ PWM.....	7
2.3.2 การทำงานของสัญญาณ PWM.....	7
2.3.3 ข้อดีของการใช้สัญญาณ PWM ในการควบคุมความเร็วมอเตอร์.....	7
2.4 ไมโครคอนโทรลเลอร์.....	8
2.4.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์.....	8
2.4.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์.....	11
2.5 มอเตอร์ไฟฟ้ากระแสตรง.....	16
2.5.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง.....	16
2.5.2 หลักการของมอเตอร์กระแสไฟฟ้าตรง.....	18
2.6 การจัมมอเตอร์.....	18
2.6.1 การควบคุมทิศทางการหมุนของ DC Motor หลักการทำงานของวงจร H-Bridge Switching....	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.2 การควบคุมทิศทางการหมุนของ DC Motor สร้างวงจร H-Bridge Switching จาก Relay.....	20
2.6.3 การควบคุมทิศทางการหมุนของ DC Motor สร้างวงจร H-Bridge Switching จาก Transistor..	22
2.6.4 การนำ L298 มาใช้เป็น IC ควบคุมมอเตอร์.....	25
บทที่ 3 การวางแผนและการออกแบบ	
3.1 การวางแผนการดำเนินงาน.....	26
3.2 แผนการดำเนินงาน.....	26
3.2.1 การดำเนินงานด้านฮาร์ดแวร์.....	26
3.2.2 การดำเนินงานด้านวงจรอิเล็กทรอนิกส์.....	27
3.2.3 การดำเนินงานด้านซอฟต์แวร์.....	29
บทที่ 4 ผลการดำเนินงาน	
4.1 ผลการดำเนินงานด้านฮาร์ดแวร์.....	31
4.2 ผลการออกแบบ และสร้างวงจรควบคุมการทำงาน.....	33
4.3 ผลการออกแบบ และสร้างโปรแกรมควบคุมการทำงาน.....	35
4.4 ผลการทดลอง.....	38
บทที่ 5 สรุปและวิเคราะห์การดำเนินงาน	
5.1 สรุปผลการทดลอง.....	40
5.2 ข้อจำกัดและปัญหาที่พบของแขนกลอัตโนมัติ.....	40
5.3 ข้อเสนอแนะและแนวทางการพัฒนา.....	40
หนังสืออ้างอิง.....	41
ภาคผนวก ก โปรแกรมในส่วนภาษาซี.....	ผก-1
ภาคผนวก ข โปรแกรมในส่วนอินเทอร์เฟส.....	ผข-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 หน้าที่อื่นๆของพอร์ด A.....	12
ตารางที่ 2.2 หน้าที่อื่นๆของพอร์ด B.....	12
ตารางที่ 2.3 หน้าที่อื่นๆของพอร์ด C.....	13
ตารางที่ 2.4 หน้าที่อื่นๆของพอร์ด D.....	15
ตารางที่ 2.5 หน้าที่อื่นๆของพอร์ด E.....	15
ตารางที่ 4.1 ผลการทดลอง.....	38



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 ตัวต้านทานปรับค่าได้.....	2
รูปที่ 2.2 ตัวอย่างการต่อวงจรให้กับโพเทนทิโอมิเตอร์.....	2
รูปที่ 2.3 แผนภาพสัญญาณเวลาของข้อมูลแบบอนุกรมจำนวน 8 บิต พร้อมด้วยบิตเริ่มต้นบิตพาริตี และ บิตสุดท้าย.....	4
รูปที่ 2.4 เบอร์MAX232ซึ่งเป็นวงจรเชื่อมต่อแบบRS-232Cโดยการใช้ไฟเลี้ยง +5 โวลท์ เพียงชุดเดียว.....	4
รูปที่ 2.5 วงจรภายใน Max232.....	5
รูปที่ 2.6 การเชื่อมต่อขาสัญญาณ.....	5
รูปที่ 2.7 รายละเอียดสัญลักษณ์ และการทำงานของขาต่างๆ ของ Serial port.....	6
รูปที่ 2.8 สัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างกัน.....	7
รูปที่ 2.9 โครงสร้างภายใน PIC16F877A.....	10
รูปที่ 2.10 ขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F877A.....	11
รูปที่ 2.11 การจัดพื้นที่หน่วยความจำใน PIC16F877A.....	15
รูปที่ 2.12 ขดลวดพันอยู่รอบขั้วแม่เหล็ก.....	16
รูปที่ 2.13 ลักษณะของขั้วแม่เหล็ก.....	17
รูปที่ 2.14 ตัวหมุน (Rotor).....	17
รูปที่ 2.15 สภาวะเริ่มต้น.....	19
รูปที่ 2.16 สภาวะเมื่อ On S1 กับ S3 พร้อมกัน.....	19
รูปที่ 2.17 สภาวะเมื่อ On S2 กับ S4 พร้อมกัน.....	20
รูปที่ 2.18 สภาวะเมื่อได้รับแรงดันตลคร่อม ขา A และ B.....	20
รูปที่ 2.19 วงจรในสภาวะเริ่มต้น.....	21
รูปที่ 2.20 วงจรในสภาวะที่ RY1 ทำงาน.....	21
รูปที่ 2.21 วงจรในสภาวะที่ RY2 ทำงาน.....	22
รูปที่ 2.22 วงจรสวิตช์แบบง่ายใช้ทรานซิสเตอร์ทำหน้าที่เป็นสวิตช์ควบคุมมอเตอร์.....	23
รูปที่ 2.23 วงจรในสภาวะเริ่มต้น.....	23
รูปที่ 2.24 สภาวะของวงจรขณะ Q1 และ Q3 ทำงาน.....	24
รูปที่ 2.25 สภาวะของวงจรขณะ Q2 และ Q4 ทำงาน.....	25
รูปที่ 3.1 ตำแหน่งของโพเทนทิโอมิเตอร์บนแกนกล 5 แกน.....	27
รูปที่ 3.2 โพเทนทิโอมิเตอร์ที่ใช้ในวงจร.....	28
รูปที่ 3.3 รีเลย์ที่ใช้ในการขับเคลื่อนมอเตอร์.....	28
รูปที่ 3.4 ไมโครคอนโทรลเลอร์ตระกูล PIC 16F877A.....	28
รูปที่ 3.5 ตัวเชื่อมต่อพอร์ตอนุกรม (Serial Port RS232).....	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.6 ผังการทำงานของไมโครคอนโทรลเลอร์.....	30
รูปที่ 4.1 การติดตั้งโพเทนทิโอมิเตอร์ที่แกนที่ 1.....	31
รูปที่ 4.2 การติดตั้งโพเทนทิโอมิเตอร์ที่แกนที่ 2.....	32
รูปที่ 4.3 การติดตั้งโพเทนทิโอมิเตอร์ที่แกนที่ 3.....	32
รูปที่ 4.4 การติดตั้งโพเทนทิโอมิเตอร์ที่แกนที่ 4 และ 5.....	32
รูปที่ 4.5 แขนกล 5 แกนที่เสร็จสมบูรณ์แล้ว.....	33
รูปที่ 4.6 วงจรที่ใช้ควบคุมแขนกล 5 แกน.....	33
รูปที่ 4.7 วงจรส่วนของไมโครคอนโทรลเลอร์.....	34
รูปที่ 4.8 วงจรส่วนของรีเลย์ควบคุมมอเตอร์.....	34
รูปที่ 4.9 ส่วนเชื่อมต่อวงจรเข้ากับคอมพิวเตอร์.....	34
รูปที่ 4.10 หน้าต่างโปรแกรมควบคุมการเคลื่อนที่ของแขนกล 5 แกน.....	35
รูปที่ 4.11 ขั้นตอนการทำงานของโปรแกรมขั้นตอนที่ 1 และ 2.....	36
รูปที่ 4.12 ขั้นตอนการทำงานของโปรแกรมขั้นตอนที่ 3 และ 4.....	36
รูปที่ 4.13 ขั้นตอนการทำงานของโปรแกรมขั้นตอนที่ 5 และ 6.....	37
รูปที่ 4.14 ปุ่มหยุดการทำงานอัตโนมัติ.....	37
รูปที่ 4.15 กราฟเส้นความสัมพันธ์ระหว่างครั้งที่ทำการทดลอง และ ระยะห่าง ระหว่างจุดศูนย์กลางของชิ้นงานที่คลาดเคลื่อน.....	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของโครงการ

ในปัจจุบันอุตสาหกรรมต่าง ๆ มีการนำระบบหุ่นยนต์อัตโนมัติมาใช้กันมากขึ้น เพื่อใช้แทนแรงงานซึ่งอาจเกิดความล้าได้เมื่อทำงานเป็นเวลานาน และอาจเกิดความผิดพลาดได้ รวมทั้งนำหุ่นยนต์ไปใช้ในการทำงานที่มีความเสี่ยงต่อการเกิดอันตราย อีกทั้งยังสามารถช่วยให้ประหยัดเวลา ค่าใช้จ่าย และช่วยเพิ่มผลผลิต โครงการนี้จึงเกิดขึ้นเพื่อเป็นต้นแบบในการศึกษาระบบหุ่นยนต์ อัตโนมัติดังกล่าว และนำไปใช้และพัฒนาต่อไปในอนาคต

1.2 วัตถุประสงค์ของโครงการ

1. ศึกษาหลักการการทำงานทำงานเบื้องต้นและความคุมการเคลื่อนที่ของแขนกล 5 แกน
2. พัฒนาแขนกล 5 แกน พร้อมโปรแกรมควบคุมการเคลื่อนที่

1.3 ขอบเขตของโครงการ

1. ศึกษาการใช้โปรแกรม Delphi ในการสร้างส่วนอินเตอร์เฟซ (Interface) เพื่อสั่งงานแขนกล 5 แกน ให้เคลื่อนที่ไปยังตำแหน่งที่ต้องการ ได้อย่างถูกต้อง
2. ออกแบบแผงวงจรที่ทำให้แขนกล 5 แกน กับคอมพิวเตอร์เชื่อมต่อกัน ได้
3. ศึกษาการใช้โปรแกรมภาษาซี ในการควบคุมแขนกล 5 แกน ด้วยไมโครคอนโทรลเลอร์ PIC16F877A
4. ทำการสั่งงานแขนกลด้วยคอมพิวเตอร์ผ่านพอร์ตอนุกรม (Serial Port RS 232)
5. ควบคุมความแม่นยำในการเคลื่อนที่ของแขนกลด้วยโพเทนทิโอมิเตอร์ (Potentiometer)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โปรแกรมที่สามารถรองรับการทำงานแขนกล 5 แกน
2. สามารถนำความรู้ด้านเขียน โปรแกรมไปใช้ในการควบคุมไมโครคอนโทรลเลอร์ในงานอื่นๆ ได้
3. สามารถนำแขนกล 5 แกน ไปใช้ประยุกต์ในงานอื่นๆ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และเฝ้าระวังถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

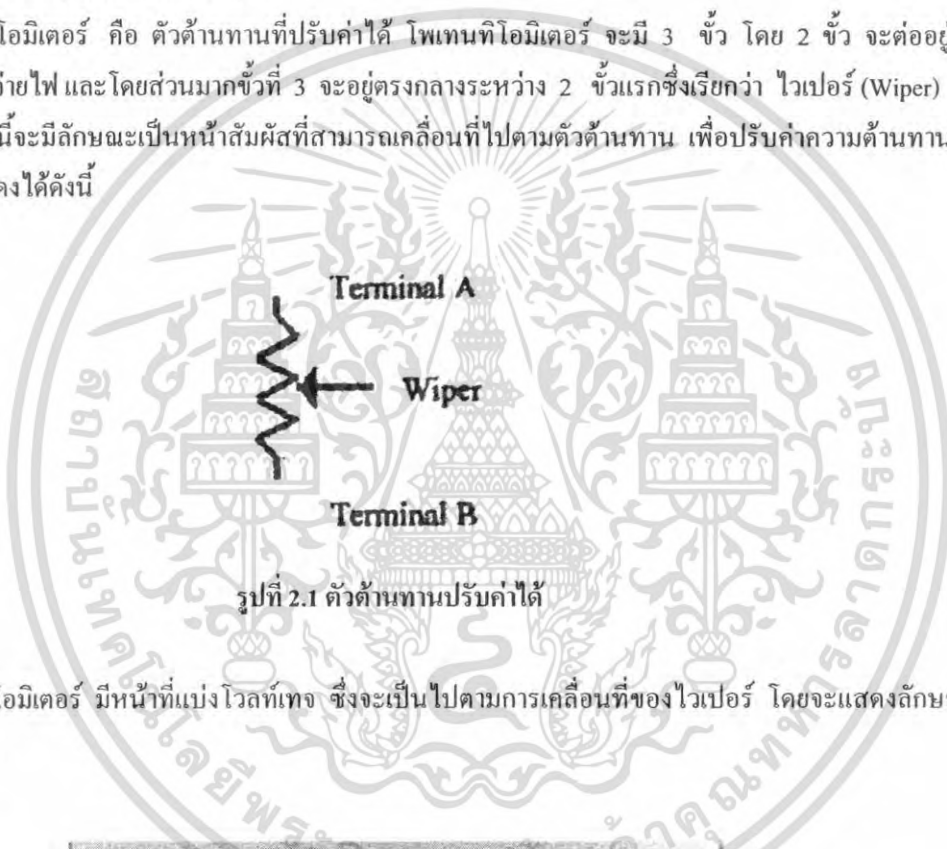
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

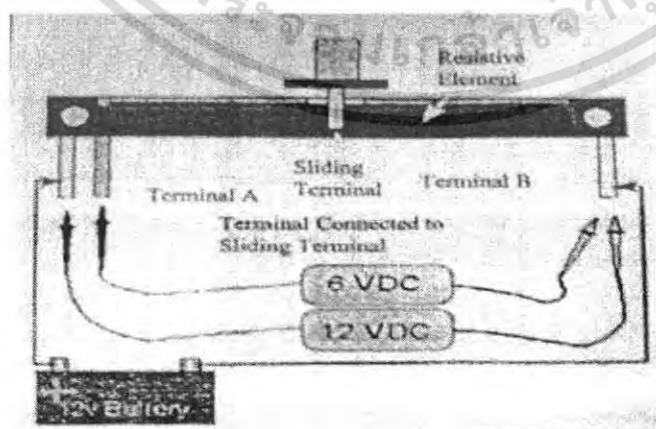
2.1 โปเทนติโอมิเตอร์ (Potentiometer)

โปเทนติโอมิเตอร์ คือ ตัวต้านทานที่ปรับค่าได้ โปเทนติโอมิเตอร์ จะมี 3 ขั้ว โดย 2 ขั้ว จะต่ออยู่กับขั้วไฟฟ้าจากแหล่งจ่ายไฟ และโดยส่วนมากขั้วที่ 3 จะอยู่ตรงกลางระหว่าง 2 ขั้วแรกซึ่งเรียกว่า ไวเปอร์ (Wiper)

ไวเปอร์ นี้จะมีลักษณะเป็นหน้าสัมผัสที่สามารถเคลื่อนที่ไปตามตัวต้านทาน เพื่อปรับค่าความต้านทานให้เพิ่มขึ้นหรือลดลง แสดงได้ดังนี้



โปเทนติโอมิเตอร์ มีหน้าที่แบ่งโวลต์เทจ ซึ่งจะเป็นไปตามการเคลื่อนที่ของไวเปอร์ โดยจะแสดงลักษณะการต่อวงจรตามรูป



รูปที่ 2.2 ตัวอย่างการต่อวงจรให้กับโปเทนติโอมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.2 จะเห็นว่ามี 2 ขั้วจะต่อหัวท้ายจะต่ออยู่กับขั้ว A และ B ซึ่งเป็น แบคเตอร์รี่ 12 โวลต์ ส่วนตัวกลางจะเป็นตัวอ้างอิงจากการปรับค่าความต้านทาน ซึ่งถ้าปรับให้ ไวเปอร์ เคลื่อนที่ไปด้านซ้ายบวก โวลท์เทจที่ได้จะสูงขึ้น และถ้าปรับให้ ไวเปอร์เคลื่อนที่ได้ด้านซ้ายลบ โวลท์เทจที่ได้จะลดลง

2.2 การทำงานของพอร์ตอนุกรม (Serial Port)

รับส่งข้อมูลแบบอนุกรม ซึ่งมีข้อดีคือ ใช้สายสัญญาณน้อยและส่งได้เป็นระยะทางไกล สำหรับโครงการนี้ใช้การติดต่อกับ ไมโครคอนโทรลเลอร์ รับส่งข้อมูลแบบอนุกรมกับคอมพิวเตอร์ เพื่อทำการเปลี่ยนระดับสัญญาณไฟฟ้าได้มาตรฐานในการรับส่งข้อมูลผ่านทางพอร์ตอนุกรม (Serial Port) ของเครื่องไมโครคอมพิวเตอร์ ซึ่งสัญญาณข่าวสารที่ได้ได้อย่างถูกประมวลโดยโปรแกรม เพื่อแสดงผลออกทางหน้าจอคอมพิวเตอร์โดยปกติเครื่องไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นอนุกรมชื่อว่า RS-232 อยู่ในตัวมันเองอยู่แล้ว ซึ่งพอร์ต RS-232 นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่า RS-232 ก็เนื่องมาจาก สมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ ของอเมริกา หรือ EIA (RS-232 : Recommended Standard Number 232, EIA: Electronic Industry Association) ได้กำหนดมาตรฐานของอุปกรณ์สื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS-232 ความจริงแล้วมาตรฐานของรับส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่มาตรฐานนี้เป็นที่นิยมกันมากที่สุดสำหรับเครื่องคอมพิวเตอร์

2.2.1 รูปแบบของข้อมูลอนุกรม

วิธีการที่จะทำให้ข้อมูลสื่อสารอนุกรมมีความถูกต้องมากยิ่งขึ้น จะใช้การเพิ่มเติมบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง ได้แก่

2.2.1.1 บิตเริ่มต้น (Start Bit)

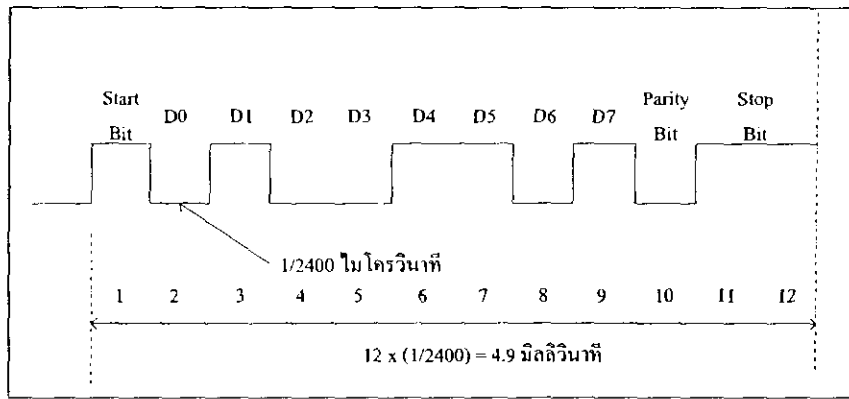
มีหน้าที่สำหรับการบ่งบอกให้วงจรทางด้านฮาร์ดแวร์ รับทราบถึง ตำแหน่งจุดเริ่มต้นของบิตข้อมูลกลุ่มใหม่ เพื่อที่จะทำการปรับจังหวะของสัญญาณการรับข้อมูลให้ตรงกัน ดังนั้นบิตเริ่มต้นนี้จึงจะถูกเพิ่มเข้าไปก่อนระดับลอจิกของสภาวะของสายสื่อสาร ขณะเมื่อไม่มีการส่งข้อมูล

2.2.1.2 บิตแสดงภาวะความเป็นเลขคู่หรือเลขคี่ (Parity Bit)

มีหน้าที่เพื่อตรวจสอบความถูกต้องของข้อมูล โดยทั่วไปมักเรียกว่า บิตพาริตีและจะนำไปแทรกต่อท้ายบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตข้อมูลที่เป็น 1 ซึ่งจะเป็นได้สองลักษณะ คือ พาริตีคู่ (Even Parity) หรือ พาริตีคี่ (Odd Parity)

2.2.1.3 บิตสุดท้าย (Stop Bit)

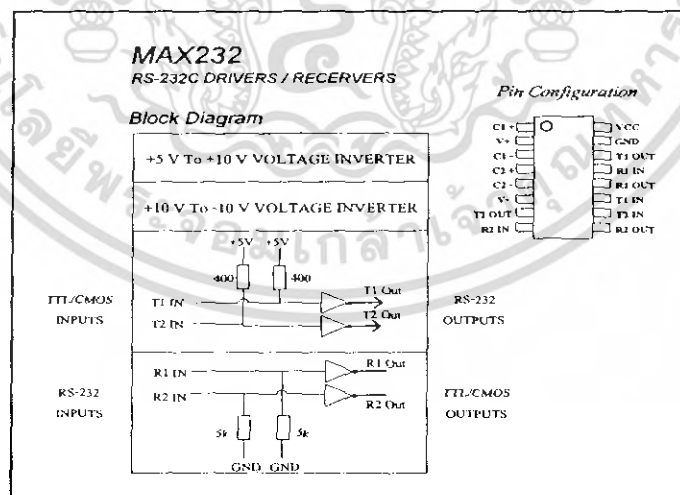
เป็นบิตที่เพิ่มเติมขึ้นเพื่อบ่งชี้ขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตสุดท้ายนี้อาจจะมีจำนวนมากกว่าหนึ่งบิตได้ คือ 1.5 บิต และ 2 บิต ดังนั้นกรณีของการส่งข้อมูล 8 บิตพร้อมบิตที่เพิ่มเติมเข้าไปโดยสมบูรณ์ คือ บิตเริ่มต้น บิตพาริตี และ บิตสุดท้าย รวมทั้งสิ้น 12 บิต ตามแผนภาพสัญญาณเวลาในรูปที่ 2 หากข้อมูลถูกส่งออกไปด้วยอัตราเร็ว 2400 บิตต่อวินาที เวลาโดยรวมในการส่งข้อมูลหนึ่งไบต์จะมีค่าเป็น 12×416 ไมโครวินาที หรือ 4.99 มิลลิวินาที



รูปที่ 2.3 แผนภาพสัญญาณเวลาของข้อมูลแบบอนุกรมจำนวน 8 บิตพร้อมด้วยบิตเริ่มต้นบิตพาริตี และ บิตสุดท้าย

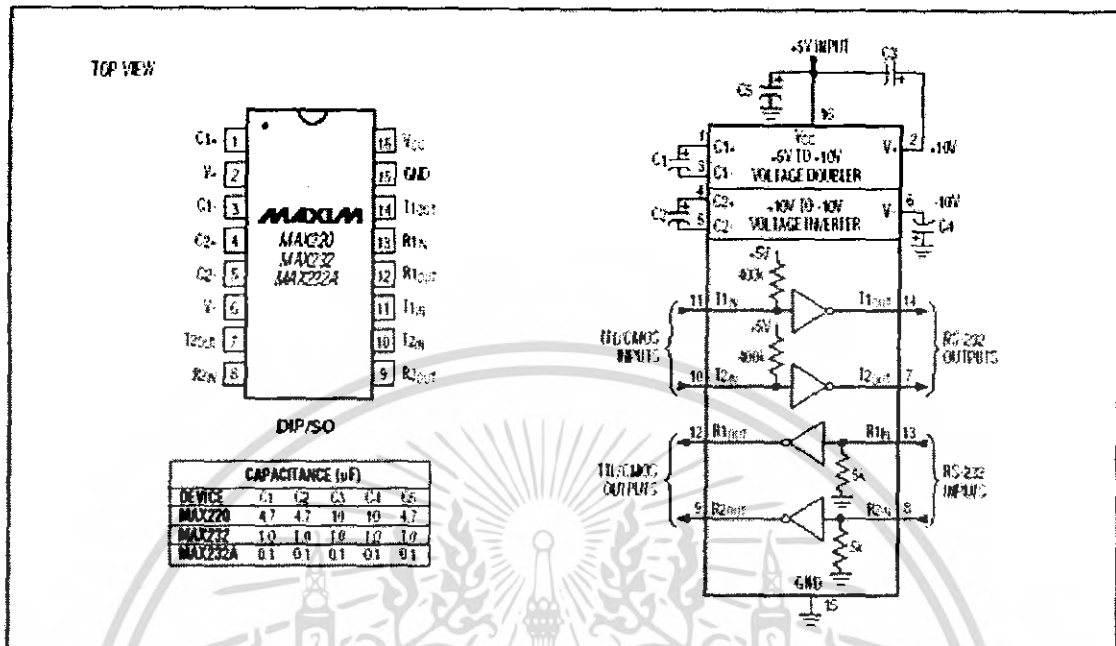
2.2.2 การเชื่อมต่อแบบมาตรฐาน RS-232

ในการเชื่อมต่อแบบอนุกรมเข้ากับอุปกรณ์คอมพิวเตอร์ต่างๆ เช่น คอมพิวเตอร์ เทลเลกซ์ หรือ โทรมิมพ์ เป็นต้น มักจะกำหนดให้การเชื่อมต่อตามมาตรฐาน RS-232 ทั้งนี้เพื่อให้มีการใช้งานเส้นสัญญาณหรือรูปแบบของตัวเชื่อมต่อที่สอดคล้องกัน จะได้ลดปัญหาการเข้ากันไม่ได้ระหว่างสัญญาณของอุปกรณ์ที่มาเชื่อมต่อกันทั้งสองด้านให้น้อยลง เนื่องจากระดับโวลเตจที่ใช้และการแทนความหมายของระดับลอจิกตามมาตรฐานนี้แตกต่างไปจากที่ใช้กันกันในระบบดิจิทัลทั่วไป โดยระดับสัญญาณของ RS-232 เป็นแบบไบโพลาร์ (Bipolar) ระดับโวลเตจทางด้านลบ ช่วง -3 โวลต์ ถึง -20 โวลต์ แทนค่าลอจิก 1 และโวลเตจทางด้านบวกช่วง +3 โวลต์ ถึง +20 โวลต์ แทนค่าลอจิก 0 ดังนั้นจะเห็นได้ว่า มีความจำเป็นต้องเพิ่มเดมูอุปกรณ์ หรือวงจรพิเศษเข้าไป เพื่อเปลี่ยนระดับโวลเตจจากระบบ 0 โวลต์ ถึง +5 โวลต์ จากขาสัญญาณของ 8051 เป็นระดับโวลเตจที่สูงกว่าค่า +3.0 โวลต์ หรือต่ำกว่า -3.0 โวลต์



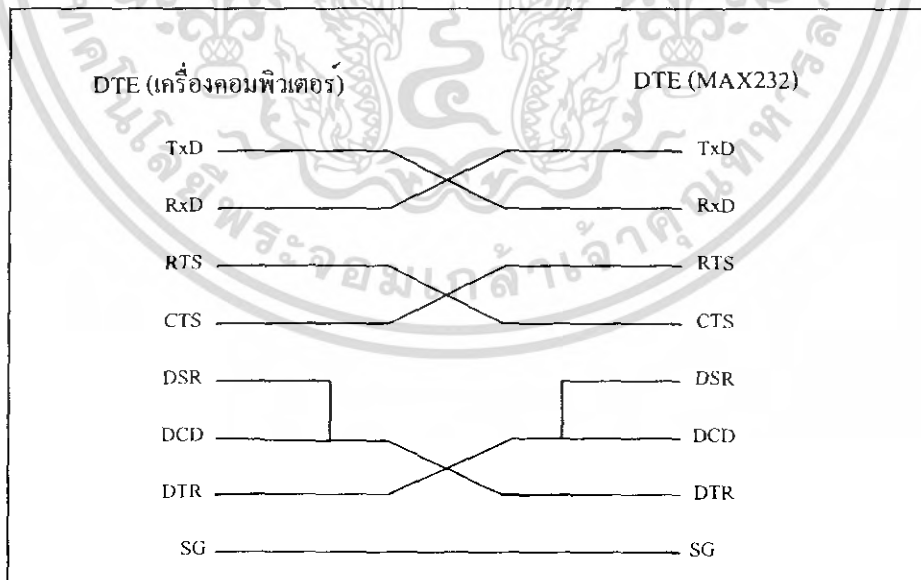
รูปที่ 2.4 ไอซีเบอร์ MAX232 ซึ่งเป็นวงจรเชื่อมต่อแบบ RS-232C โดยการใช้ไฟเลี้ยง +5 โวลต์ เพียงชุดเดียว

2.2.2.1 การต่อสัญญาณ Serial Port



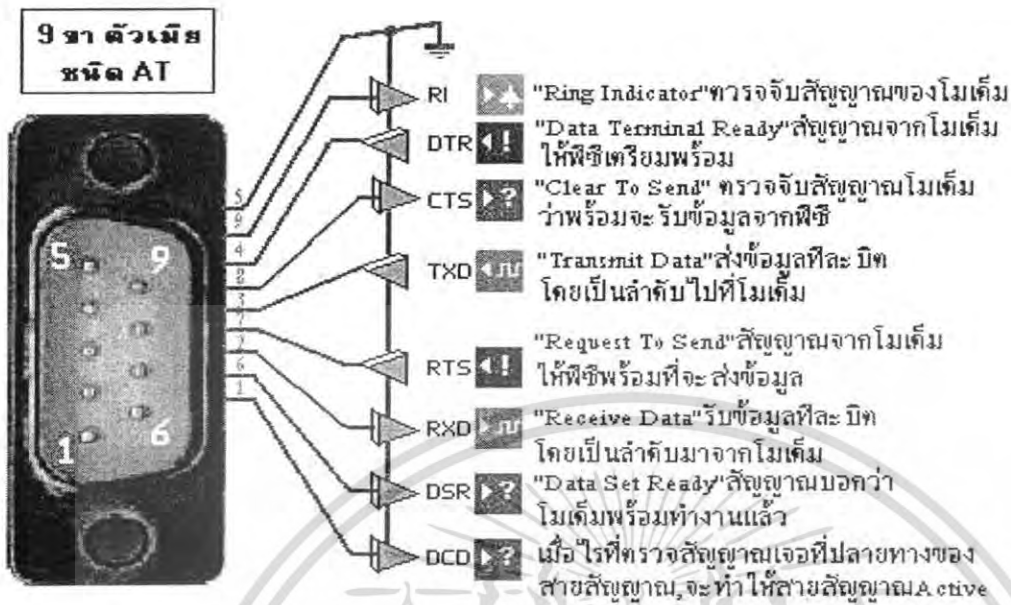
รูปที่ 2.5 วงจรภายใน Max232

การต่อสายอินเทอร์เฟซ RS232 ระหว่างเครื่องคอมพิวเตอร์ กับ MAX232



รูปที่ 2.6 การเชื่อมต่อขาสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และร้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 รายละเอียดสัญลักษณ์ และการทำงานของขาต่างๆ ของ Serial port

2.2.2.2 คำอธิบายการทำงานของแต่ละขาที่ใช้งานของ DB-9

-Transmit Data (TD) เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์พีซี) ไปยังโมเด็มหรือต่อกับไมโครคอมพิวเตอร์ตัวอื่น เมื่อไม่มีสัญญาณส่งออกสถานะภาพของลอคจิกที่ขาที่มีค่าเท่ากับ "1" หรือเทียบเท่า Stop Bit

-Received Data (RD) เป็นทางสัญญาณเข้าไปยัง DTE หรือไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณ รับเข้ามา ขานี้จะมีสถานะภาพลอคจิกเป็น "1"

-Request to Send (RTS) ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์เป็นการเรียกร้องที่จะส่งสัญญาณมาทาง TD สัญญาณนี้จะใช้คู่กับ CTS ที่อุปกรณ์รับ หากได้รับ สัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมก็จะส่งสัญญาณออกไปที่สาย CTS

-Clear to Send (CTS) เมื่อสายสัญญาณนี้อยู่ในสภาวะออฟ (Negative Voltage หรือลอคจิก "1") หมายความว่า อุปกรณ์รับกำลังบอกว่า พร้อมจะรับข้อมูลแล้ว

-Data Set Ready (DSR) เมื่อสายสัญญาณนี้อยู่ในสภาวะออน (ลอคจิก "0") จะเป็นการบอกไมโครคอมพิวเตอร์ว่า พร้อมที่จะส่งได้แล้ว

-Signal Ground (SG) ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆสายสัญญาณจะมีแรงดันเป็น "0" เมื่อเทียบกับสายสัญญาณอื่นๆ

-Data Terminal Ready (DTR) คอมพิวเตอร์เปิดสัญญาณนี้ให้ออน (ลอคจิก "0") เมื่อพร้อมที่จะติดต่อรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การควบคุมความเร็วมอเตอร์

การควบคุมความเร็วมอเตอร์ทำได้โดย

2.3.1 การควบคุมแบบ PWM (Pulse Width Modulation)

Pulse width modulation (PWM) คือ เทคนิคสำหรับควบคุมวงจรทางด้านฮาร์ดแวร์ โดยใช้สัญญาณเอาต์พุตแบบดิจิทัลของไมโครโปรเซสเซอร์ควบคุม

2.3.2 การทำงานของสัญญาณ PWM

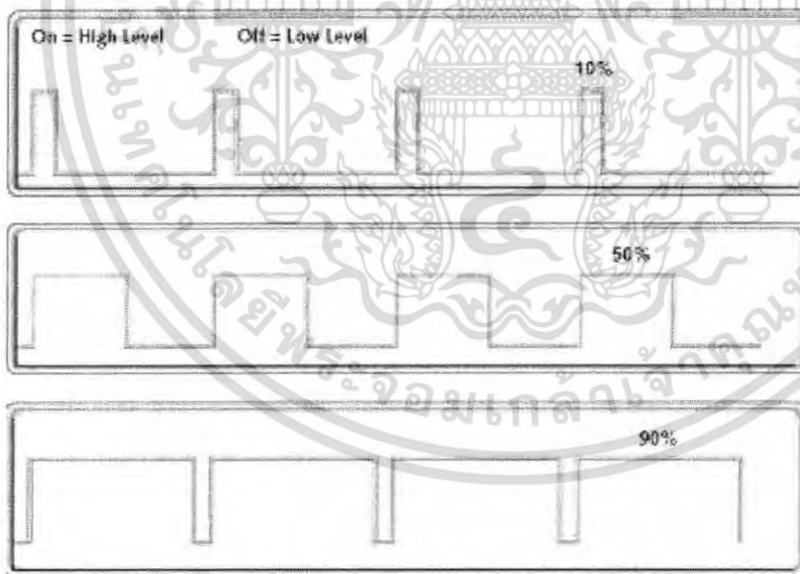
รูปที่ 2.8 แสดงสัญญาณ PWM ที่แตกต่างกัน 3 สัญญาณ

โดย 4a แสดงสัญญาณ PWM ที่ 10% duty cycle คือ สัญญาณในการออนจะเป็น 10% ของคาบสัญญาณ และจะออฟเป็น 90% ของคาบสัญญาณ

โดย 4b แสดงสัญญาณ PWM ที่ 50% duty cycle คือ สัญญาณในการออนจะเป็น 50% ของคาบสัญญาณ และจะออฟเป็น 50% ของคาบสัญญาณ

โดย 4c แสดงสัญญาณ PWM ที่ 90% duty cycle คือ สัญญาณในการออนจะเป็น 90% ของคาบสัญญาณ และจะออฟเป็น 10% ของคาบสัญญาณ

เช่น ถ้า Power Supply มี 9V และ duty cycle เป็น 10% จะได้เอาต์พุต 0.9V



รูปที่ 2.8 สัญญาณ PWM ซึ่งแสดงค่า duty cycles ที่ต่างๆกัน

2.3.3 ข้อดีของการใช้สัญญาณ PWM ในการควบคุมความเร็วมอเตอร์

ข้อดีของการใช้สัญญาณ PWM ในการควบคุมความเร็วมอเตอร์มีดังนี้

1. ในการอินเตอร์เฟสไมโครคอนโทรลเลอร์ กับ PWM สามารถทำได้ง่าย และ ใช้สัญญาณเอาต์พุตเพียงแค่

สัญญาณเดียวในการควบคุมความเร็ว
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. PWM มีประสิทธิภาพ คือ แหล่งจ่ายไฟ จะจ่ายกำลัง ได้เต็มที่ทั้งเปิด และ ปิด (FULL ON and FULL OFF)
3. PWM ทำให้ได้ค่า ทอร์ค และ ความเร็วสูงสุดของมอเตอร์ เพราะแหล่งจ่ายไฟจ่ายกำลัง ได้เต็มที่ทั้งเปิด และ ปิด (FULL ON and FULL OFF)

2.4 ไมโครคอนโทรลเลอร์ (PIC16F877A)

ไมโครคอนโทรลเลอร์ในตระกูล PIC16F877A เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่มีอุปกรณ์สนับสนุนภายในหลายอย่าง ได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์สำหรับรับส่งข้อมูล แบบอนุกรม และเป็นชิปรุ่นแรก ที่หน่วยความจำโปรแกรมเป็นแบบแฟลช และมีหน่วยความจำข้อมูลแบบ อีอีพรอม (EEPROM) ทำให้สามารถพัฒนาโปรแกรมได้ง่าย และสนับสนุนการทำงานแบบ อินเซอร์ทิต ดีบั๊กเกอร์ (In Circuit Debugger) ทำให้ไม่ต้องซื้ออีอีพรอมอิมูเลเตอร์ (EEPROM Emulator) มีคำสั่งภาษาแอสเซมบลี 35 คำสั่ง และมีวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (A/D) ขนาด 10 บิตอยู่ภายในด้วย เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายในนี้เอง ทำให้การใช้งานง่ายและมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้เรายังสามารถนำอุปกรณ์อื่นๆ มาต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ได้อีกด้วย

2.4.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ (PIC16F877A)

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ PIC16F877A แสดงในรูปที่ 2.9 ซึ่งมีคุณสมบัติหลักๆ ดังนี้

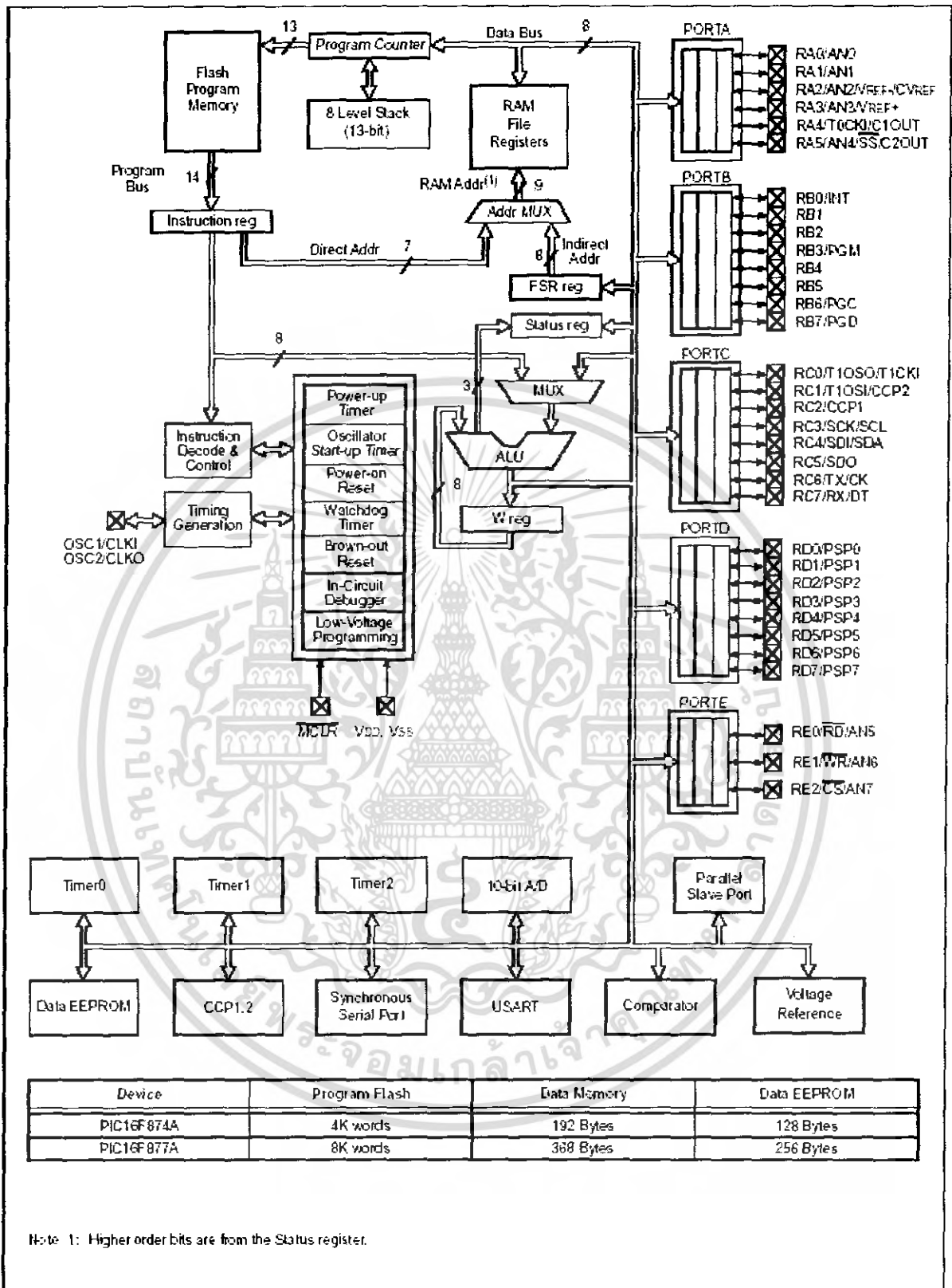
1. มีคำสั่งที่เป็นภาษาแอสเซมบลี 35 คำสั่ง
2. ใน 1 คำสั่งใช้เวลาทำงาน 1 ถึง 2 ไชเคิล
3. ทำงานได้สูงสุดที่สัญญาณนาฬิกาตั้งแต่ไปตรงถึง 20 MHz
4. ทำงานแบบ Pipe-line สามารถทำงาน 2 อย่างในเวลาเดียวกันได้
5. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8 KWord (1 word = 14 บิต)
6. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8 KWord (1 word = 14 บิต)
7. มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 368 ไบท์
8. มีหน่วยความจำข้อมูลแบบ EEPROM ขนาด 256 ไบท์
9. ตอบสนองการอินเทอร์รัปต์ได้ทั้งหมด 14 แหล่ง
10. มี stack ให้ใช้ได้สูงสุด 8 ระดับ
11. มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up และ Watchdog Timer
12. มีระบบ Code Protection กันการคัดลอก
13. มีโหมดประหยัดพลังงาน (Sleep Mode)
14. สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือวงจร RC ก็ได้
15. สามารถโปรแกรมด้วยไฟ +5 VDC ได้
16. ใช้การ โปรแกรมแบบ In-Circuit Serial Programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ 8 อังอาจอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. ทำงานที่ไฟเลี้ยง 2 VDC ถึง 5.5 VDC
18. Current Sink และ Current Source อยู่ที่ 25 mA
19. มี Timer/Counter 3 ตัว คือ Timer 0 ขนาด 8 บิต, Timer 1 ขนาด 16 บิต และ Timer 2 ขนาด 8 บิต
20. มีโมดูล Capture/Compare/PWM (Pulse Width Modulation) 2 ชุด
21. มีวงจรแปลงสัญญาณแอนะล็อกเป็นดิจิทัล (A/D Converter) ขนาด 10 บิต
22. มีระบบ USART สำหรับเชื่อมต่อการสื่อสารแบบ RS232
23. มีระบบตรวจระดับไฟเลี้ยง (Brown-out Reset)
24. มี I/O พอร์ตทั้งหมด 5 พอร์ต แต่ละพอร์ตมีจำนวนบิตไม่เท่ากัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และ 9 ให้องค์งอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

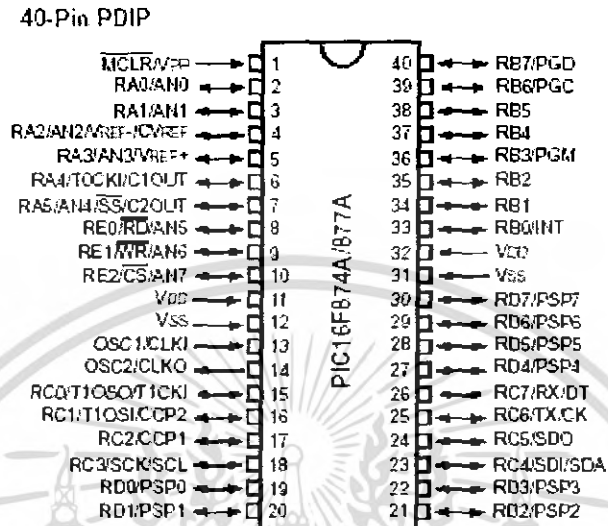


รูปที่ 2.9 โครงสร้างภายใน PIC16F877A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ 10. อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ (PIC16F877A)

ไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877A เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา แสดงดังภาพที่ 2.10



รูปที่ 2.10 ขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F877A

2.4.2.1 หน้าที่และการใช้งานของสัญญาณต่างๆ

1. MCLR/Vpp : Master Clear (Reset) Input/Programming Voltage Input ทำหน้าที่เป็นขาสัญญาณรีเซ็ต (Reset) เมื่อขานี้ได้รับลอจิก 0 ไมโครคอนโทรลเลอร์จะถูกรีเซ็ต และทำหน้าที่เป็นขาสัญญาณรับแรงดันขณะทำการบันทึกโปรแกรมลงหน่วยความจำของไมโครคอนโทรลเลอร์
2. VDD : Positive Supply (+2.00 V ถึง +5.5 V) ทำหน้าที่เป็นขาไฟเลี้ยงของไมโครคอนโทรลเลอร์
3. VSS : Ground ทำหน้าที่เป็นขากาวาน์
4. OSC1/CLKIN : Oscillator Crystal Input/External Clock Source Input ทำหน้าที่เป็นขาสัญญาณอินพุตสำหรับต่อคริสตอล ในกรณีที่อยู่ในโหมดการใช้สัญญาณนาฬิกาจากภายนอก
5. OSC2/CLKOUT : Oscillator Crystal Output/External Clock Source Output ทำหน้าที่เป็นขาสัญญาณเอาต์พุตสำหรับต่อคริสตอล ในกรณีที่อยู่ในโหมดการใช้สัญญาณนาฬิกาจากภายนอก
6. RA0 - RA5 : พอร์ต A มีจำนวน 6 ขา เป็นพอร์ตแบบสองทิศทาง (Bi-directional I/O Port) คือเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และนำออกอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 หน้าที่อื่นๆของพอร์ต A

RA0/AN0 RA0 AN0	2	3	19	19	I/O I	TTL	PORTA is a bidirectional I/O port. Digital I/O. Analog input 0.
RA1/AN1 RA1 AN1	3	4	20	20	I/O I	TTL	
RA2/AN2/VREF-/CVREF RA2 AN2 VREF- CVREF	4	5	21	21	I/O I I O	TTL	
RA3/AN3/VREF+ RA3 AN3 VREF+	5	6	22	22	I/O I I	TTL	
RA4/T0CKI/C1OUT RA4	6	7	23	23	I/O I O	ST	
T0CKI C1OUT					I O		
RA5/AN4/SS/C2OUT RA5 AN4 SS C2OUT	7	8	24	24	I/O I I O	TTL	

7. RB0 - RB7 : พอร์ต B มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้บางขายังทำหน้าที่รับสัญญาณอินพุตจากการอินเทอร์รัปต์ (Interrupt) จากภายนอกด้วย แสดงดังตารางที่ 2.2

ตารางที่ 2.2 หน้าที่อื่นๆของพอร์ต B

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT RB0 INT	33	36	8	9	I/O I	TTL-ST ¹	PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. Digital I/O. External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM RB3 PGM	36	39	11	12	I/O I	TTL	Digital I/O. Low-voltage ICSP programming enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC RB6 PGC	39	43	16	16	I/O I	TTL-ST ²	Digital I/O. In-circuit debugger and ICSP programming clock.
RB7/PGD RB7 PGD	40	44	17	17	I/O I/O	TTL-ST ²	Digital I/O. In-circuit debugger and ICSP programming data.

Legend: I = input O = output I/O = input/output P = power
 -- = Not used TTL = TTL input ST = Schmitt Trigger input

8. RC0 - RC7 : พอร์ต C มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.3

ตารางที่ 2.3 หน้าที่อื่นๆของพอร์ต C

Pin Name	PDP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O-P Type	Buffer Type	Description
RC0:T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	34	I/O O I	ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1 external clock input.
RC1:T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	35	I/O I I/O	ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	36	I/O I/O	ST	Digital I/O. Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	18	20	37	37	I/O I/O I/O	ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	42	I/O I I/O	ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	24	26	43	43	I/O O	ST	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	25	27	44	44	I/O O I/O	ST	Digital I/O. USART asynchronous transmit. USART1 synchronous clock.
RC7/RX/DT RC7 RX DT	26	29	1	1	I/O I I/O	ST	Digital I/O. USART asynchronous receive. USART synchronous data.

9. RD0 - RD7 : พอร์ต D มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.4

ตารางที่ 2.4 หน้าที่อื่นๆของพอร์ต D

Pin Name	PDIP Pin#	PLCC Pin#	TOFP Pin#	QFN Pin#	I/O Type	Buffer Type	Description
RD0:PSP0 RD0 PSP0	19	21	38	38	I/O I/O	ST:TTL ⁽³⁾	PORTD is a bidirectional I/O port or Parallel Slave Port when interfacing to a microprocessor bus. Digital I/O. Parallel Slave Port data.
RD1:PSP1 RD1 PSP1	20	22	39	39	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD2:PSP2 RD2 PSP2	21	23	40	40	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD3:PSP3 RD3 PSP3	22	24	41	41	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD4:PSP4 RD4 PSP4	27	30	2	2	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD5:PSP5 RD5 PSP5	28	31	3	3	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD6:PSP6 RD6 PSP6	29	32	4	4	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.
RD7:PSP7 RD7 PSP7	30	33	5	5	I/O I/O	ST:TTL ⁽³⁾	Digital I/O. Parallel Slave Port data.

10. RE0 – RE2 : พอร์ต E มีจำนวน 3 ขา เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.5

ตารางที่ 2.5 หน้าที่อื่นๆของพอร์ต E

RE0:RD/AN5 RE0 RD AN5	8	9	25	25	I/O I I	ST:TTL ⁽³⁾	PORTE is a bidirectional I/O port. Digital I/O. Read control for Parallel Slave Port. Analog input 5.
RE1:WR/AN6 RE1 WR AN6	9	10	26	26	I/O I I	ST:TTL ⁽³⁾	Digital I/O. Write control for Parallel Slave Port. Analog input 6.
RE2:CS/AN7 RE2 CS AN7	10	11	27	27	I/O I I	ST:TTL ⁽³⁾	Digital I/O. Chip select control for Parallel Slave Port. Analog input 7.

2.4.2.2 การจัดการหน่วยความจำของ PIC16F877A

ไมโครคอนโทรลเลอร์ PIC16F877A มีหน่วยความจำโปรแกรมแบบแฟลชซึ่งใช้สำหรับเก็บโปรแกรมควบคุมการทำงานของ PIC มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 368 ไบต์ หน่วยความจำข้อมูลแบบอีอีพรอม (Data Memory EEPROM) ขนาด 256 ไบต์ และได้มีการจัดการหน่วยความจำของ PIC ออกเป็นแบงก์ต่างๆ ได้ทั้งหมด 4 แบงก์ คือ Bank 0 – Bank 3 และแต่ละแบงก์ก็มีขนาด 128 ไบต์ ดังรูปที่ 2.11

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 180h
TMR0 01h	OPTION REG 81h	TMR0 101h	OPTION REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEDARH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPAD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h		General Purpose Register 16 Bytes 117h	General Purpose Register 16 Bytes 197h
RCSTA 18h	TXSTA 98h		
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch	CMCON 9Ch		
CCP2CON 1Dh	CVRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes 7Fh	General Purpose Register 80 Bytes ACh	General Purpose Register 80 Bytes 120h	General Purpose Register 80 Bytes 1A0h
	accesses 70h-7Fh EFh FFh	accesses 70h-7Fh 16Fh 170h	accesses 70h-7Fh 1EFh 1F0h
Bank 0	Bank 1	Bank 2	Bank 3

Unimplemented data memory locations, read as '0'.
 Not a physical register.

Note 1: These registers are not implemented on the PIC16F876A.
Note 2: These registers are reserved; maintain these registers clear.

รูปที่ 2.11 การจัดพื้นที่หน่วยความจำใน PIC16F877A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบงก์ 0 และแบงก์ 1 จะแบ่งออกเป็น 2 ส่วนเหมือนกันคือ ส่วนที่ 1 ขนาด 32 ไบต์ เป็นพื้นที่ของรีจิสเตอร์ไฟลด์ที่เรียกว่า SFR (Special Function Register) ซึ่งใช้สำหรับกำหนดเงื่อนไขการทำงานและบันทึกสถานการณ์ทำงานของไมโครคอนโทรลเลอร์ ส่วนที่ 2 ขนาด 96 ไบต์เป็นพื้นที่หน่วยความจำสำหรับใช้งานทั่วไป ซึ่งใช้สำหรับเก็บผลลัพธ์และเงื่อนไขต่างๆ ของโปรแกรม

แบงก์ 2 และแบงก์ 3 จะแบ่งออกเป็นสองส่วน แต่มีขนาดแตกต่างกับแบงก์ 0 และแบงก์ 1 คือ ส่วนที่ 1 ขนาด 16 ไบต์ เป็นพื้นที่ของรีจิสเตอร์ไฟลด์ ส่วนที่ 2 ขนาด 112 ไบต์ เป็นพื้นที่หน่วยความจำสำหรับใช้งานทั่วไป

2.5 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรง เป็นต้นกำเนิดขับเคลื่อนที่สำคัญอย่างหนึ่งในโรงงานอุตสาหกรรมเพราะมีคุณสมบัติที่ดีเด่นในด้านการปรับความเร็วได้ ตั้งแต่ความเร็วต่ำสุด จนถึงสูงสุด นิยมใช้กันมากในโรงงานอุตสาหกรรม เช่น โรงงานทอผ้า โรงงานเส้นใยโพลีเอสเตอร์ โรงงานถลุงโลหะหรือให้ เป็นต้นกำเนิดในการขับเคลื่อนรถไฟ เป็นต้นในการศึกษาเกี่ยวกับมอเตอร์ไฟฟ้ากระแสตรง จึงควรรู้จักอุปกรณ์ต่าง ๆ ของมอเตอร์ไฟฟ้ากระแสตรง และเข้าใจถึงหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบต่างๆ

2.5.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

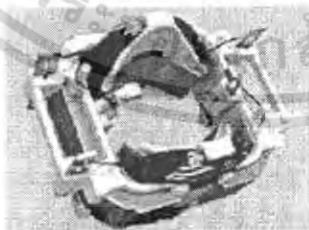
มอเตอร์ไฟฟ้ากระแสตรงที่ส่วนประกอบที่สำคัญ 2 ส่วนดังนี้

2.5.1.1 ส่วนที่อยู่กับที่หรือที่เรียกว่าสเตเตอร์ (Stator)

ส่วนต่าง ๆ ของสเตเตอร์ (Stator) ประกอบด้วย

1. เฟรมหรือ โยค (Frame or Yoke) เป็น โครงภายนอกทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กจากขั้วเหนือไปขั้วใต้ให้ครบวงจร และยึดส่วนประกอบอื่นๆ ให้แข็งแรงทำด้วยเหล็กหล่อหรือเหล็กแผ่นหนาหมุนเป็นรูปทรงกระบอก

2. ขั้วแม่เหล็ก (Pole) ประกอบด้วย แกนขั้วแม่เหล็ก และ ขดลวด



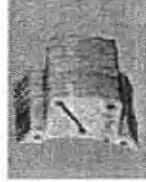
รูปที่ 2.12 ขดลวดพันอยู่รอบขั้วแม่เหล็ก

แกนขั้ว (Pole Core) ทำด้วยแผ่นเหล็กบางๆ กั้นด้วยฉนวนประกบกันเป็นแท่งยึดติดกับเฟรม ส่วนปลายที่ทำเป็นรูปโค้งนั้นเพื่อโค้งรับรูปกลมของตัวโรเตอร์เรียกว่าขั้วแม่เหล็ก (Pole Shoes) มีวัตถุประสงค์ให้ขั้วแม่เหล็กและโรเตอร์ใกล้ชิดกันมากที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุด เพื่อให้เกิดช่องอากาศน้อยที่สุดจะมีผลให้เส้นแรงแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านโทสมุดกลาง พระจอมเกล้าลาดกระบัง

จากขั้วแม่เหล็กจากขั้วแม่เหล็กผ่าน ไปยังโรเตอร์มากที่สุดแล้วทำให้เกิดแรงบิดหรือกำลังบิดของโรเตอร์มากเป็นการทำให้มอเตอร์มีกำลังหมุน (Torque)



รูปที่ 2.13 ลักษณะของขั้วแม่เหล็ก

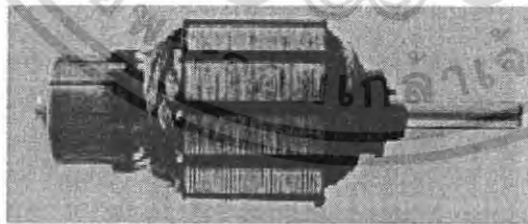
ขดลวดสนามแม่เหล็ก (Field Coil) จะพันอยู่รอบๆ แกนขั้วแม่เหล็กขดลวดนี้ ทำหน้าที่รับกระแสจากภายนอก เพื่อสร้างเส้นแรงแม่เหล็กให้เกิดขึ้น และเส้นแรงแม่เหล็กนี้ จะเกิดการหักล้างและเสริมกันกับสนามแม่เหล็กของอาร์มาเจอร์ทำให้เกิดแรงบิดขึ้น

2.5.1.2 ตัวหมุน (Rotor)

ตัวหมุนหรือเรียกว่าโรเตอร์ตัวหมุนนี้ทำให้เกิดกำลังงานมีแกนวางอยู่ในตลับลูกปืน (Ball Bearing) ซึ่งประกอบอยู่ในแผ่นปิดหัวท้าย (End Plate) ของมอเตอร์

ตัวโรเตอร์ประกอบด้วย 4 ส่วนด้วยกัน คือ

1. แกนเพลลา (Shaft)
2. แกนเหล็กอาร์มาเจอร์ (Armature Core)
3. คอมมิวเตเตอร์ (Commutator)
4. ขอลวดอาร์มาเจอร์ (Armature Winding)



รูปที่ 2.14 ตัวหมุน (Rotor)

แกนเพลลา (Shaft) เป็นตัวสำหรับยึดคอมมิวเตเตอร์ และยึดแกนเหล็กอาร์มาเจอร์ (Armature Core) ประกอบเป็นตัวโรเตอร์แกนเพลลานี้จะวางอยู่บนเบร้ง เพื่อขังลับให้หมุนอยู่ในแนวนิ่งไม่มีการสั่นสะเทือนได้

แกนเหล็กอาร์มาเจอร์ (Armature Core) ทำด้วยแผ่นเหล็กบางอาบฉนวน (Laminated Sheet Steel) เป็นที่สำหรับพันขดลวดอาร์มาเจอร์ซึ่งสร้างแรงบิด (Torque)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอมมิวเตเตอร์ (Commutator) ทำด้วยทองแดงออกแบบเป็นซี่แต่ละซี่มีฉนวนไมก้า (Mica) คั่นระหว่างซี่ของคอมมิวเตเตอร์ ส่วนหัวซี่ของคอมมิวเตเตอร์ จะมีร่องสำหรับใส่ปลายสาย ของขดลวดอาร์มาเจอร์ ตัวคอมมิวเตเตอร์นี้อัดแน่นติดกับแกนเพลลา เป็นรูปกลมทรงกระบอก ก มีหน้าที่สัมผัสกับแปรงถ่าน (Carbon Brushes) เพื่อรับกระแสจากสายป้อนเข้าไปยัง ขดลวดอาร์มาเจอร์เพื่อสร้างเส้นแรงแม่เหล็กอีกส่วนหนึ่งให้เกิดการหักล้างและเสริมกันกับเส้นแรงแม่เหล็กอีกส่วน ซึ่งเกิดจากขดลวดขั้วแม่เหล็ก ดังกล่าวมาแล้วเรียกว่า ปฏิกริยามอเตอร์ (Motor action)

ขดลวดอาร์มาเจอร์ (Armature Winding) เป็นขดลวดพันอยู่ในร่องสลอท (Slot) ของแกนอาร์มาเจอร์ ขนาดของลวดจะเล็กหรือใหญ่ละจำนวนรอบจะมากหรือน้อยนั้นขึ้นอยู่กับวิธีการออกแบบของตัวโรเตอร์ชนิดนั้นๆ เพื่อที่จะให้เหมาะสมกับงานต่างๆ ที่ต้องการ

2.5.2 หลักการของมอเตอร์กระแสไฟฟ้าตรง (Motor Action)

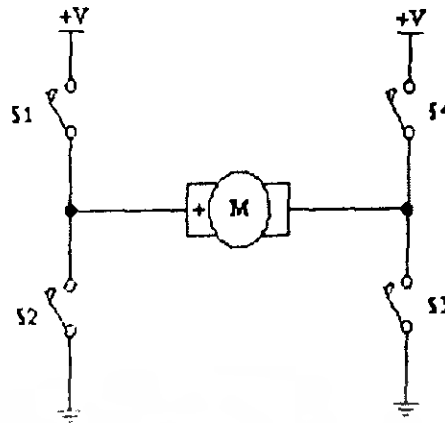
หลักการของมอเตอร์ไฟฟ้ากระแสตรง (Motor Action) เมื่อเป็นแรงดันกระแสไฟฟ้าตรงเข้าไปในมอเตอร์ ส่วนหนึ่งจะแปรงถ่านผ่านคอมมิวเตเตอร์เข้าไปในขดลวดอาร์มาเจอร์สร้างสนามแม่เหล็กขึ้น และกระแสไฟฟ้าอีกส่วนหนึ่งจะไหลเข้าไปในขดลวดสนามแม่เหล็ก (Field coil) สร้างขั้วเหนือ-ใต้ขึ้นจะเกิดสนามแม่เหล็ก 2 สนาม ในขณะเดียวกัน ความคุณสมบัติของเส้นแรงแม่เหล็ก จะไม่ตัดกัน ทิศทางตรงข้ามจะหักล้างกัน และทิศทางเดียวจะเสริมแรงกัน ทำให้เกิดแรงบิดในตัวอาร์มาเจอร์ ซึ่งวางแกนเพลลาและแกนเพลลานี้ สวมอยู่กับคัลลูปป็นของมอเตอร์ ทำให้อาร์มาเจอร์นี้หมุนได้ ขณะที่ตัวอาร์มาเจอร์ทำหน้าที่หมุนได้นี้เรียกว่า โรเตอร์ (Rotor) ซึ่งหมายความว่าตัวหมุน การที่อำนาจเส้นแรงแม่เหล็กทั้งสองมีปฏิกริยาต่อกัน ทำให้ขดลวดอาร์มาเจอร์ หรือโรเตอร์หมุนไปนั้น เป็นไปตามกฎซ้ายของเฟลมมิ่ง (Fleming's left hand rule)

2.6 การขับมอเตอร์

มอเตอร์กระแสตรงจะมามีการทำงานเพียง 3 State เท่านั้น คือ หมุนตามเข็มนาฬิกา หมุนทวนเข็มนาฬิกา และหยุดนิ่งเท่านั้น ซึ่งอัตราความเร็วในการหมุนและแรงบิดจะขึ้นอยู่กับกระแสและแรงดันที่จ่ายให้มันถ้าหากแรงบิดของมอเตอร์ไม่เพียงพอต่อการทำงานเราก็สามารถที่จะทำการทศรอบของการหมุนลงก็จะได้แรงบิดเพิ่มมากขึ้น หรือ เราสามารถควบคุมความเร็วของมอเตอร์ได้โดยการจ่ายพลังงานในลักษณะ pulse width modulation (PWM)

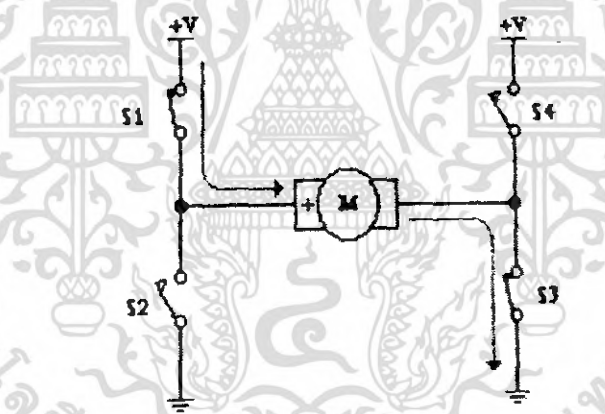
2.6.1 การควบคุมทิศทางการหมุนของ DC Motor หลักการทำงานของวงจร H-Bridge Switching

เริ่มจากหลักการของวงจรมัน จะประกอบไปด้วย สวิตซ์ 4 ตัว นั่นก็คือ S1 S2 S3 และ S4 นั่นเองซึ่งในรูปตัวอย่าง จะใช้ DC-Motor เป็น Load ของวงจรมันเอง ในสภาวะเริ่มต้นสวิตซ์ทุกตัว Off อยู่ ก็จะไม่มียะไรเกิดขึ้นทั้งสิ้น เพราะไม่มีกระแสไฟฟ้าไหลเข้าสู่มอเตอร์ ดังรูปที่ 2.13



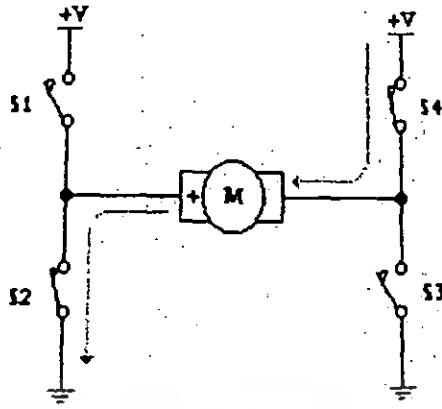
รูปที่ 2.15 สภาวะเริ่มต้น

และเมื่อเราทำการ On สวิตช์ S1 และ S3 พร้อมกันดังรูปที่ 2.14 จะเป็นการเชื่อมวงจร ทำให้มีกระแสไฟฟ้า ไหลผ่านมอเตอร์ จากขั้วบวกของมอเตอร์ ไปยังขั้วลบของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้ในทิศทาง Forward (จะหมุนแบบตามเข็มนาฬิกา หรือทวนเข็มนาฬิกานั้นขึ้นอยู่กับ ลักษณะของการพันขดลวดภายในมอเตอร์)



รูปที่ 2.16 สภาวะเมื่อ On S1 กับ S3 พร้อมกัน

และในทางกลับกัน ถ้าหากเราทำการ On สวิตช์ S2 และ S4 พร้อมกันดังรูปที่ 2.15 ก็จะเป็นการเชื่อมวงจร และทำให้เกิดกระแสไฟฟ้า ไหลผ่านมอเตอร์จากขั้วของมอเตอร์ไปยังขั้วบวกของมอเตอร์ จึงทำให้มอเตอร์สามารถหมุนได้และเป็นการหมุนในทิศทาง Reverse (กลับทิศทางกับกรณีแรก)



รูปที่ 2.17 สถานะเมื่อ On S2 กับ S4 พร้อมกัน

สรุปได้ว่าวงจรนี้อาศัยสวิตช์ 4 ตัว เพื่อบังคับทิศทางการไหลของกระแสไฟฟ้า ที่ไหลผ่านมอเตอร์ เพื่อควบคุมให้มอเตอร์หมุนตามทิศทางที่เราต้องการ โดยการผลัดกัน On และ off สวิตช์พร้อมกัน 2 ตัวนั้น

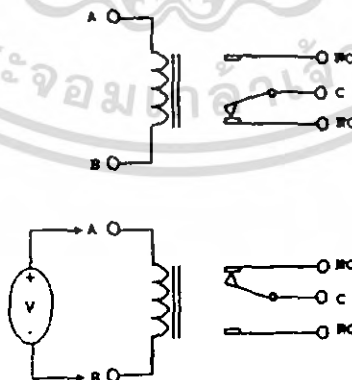
2.6.2 การควบคุมทิศทางการหมุนของ DC Motor สร้างวงจร H-Bridge Switching จาก Relay

ภายใน โครงสร้างของรีเลย์จะประกอบไปด้วยขดลวด (Coil) 1 ชุด และ หน้าสัมผัส (Contactor) ซึ่งในหน้าสัมผัส 1 ชุดจะประกอบไปด้วย

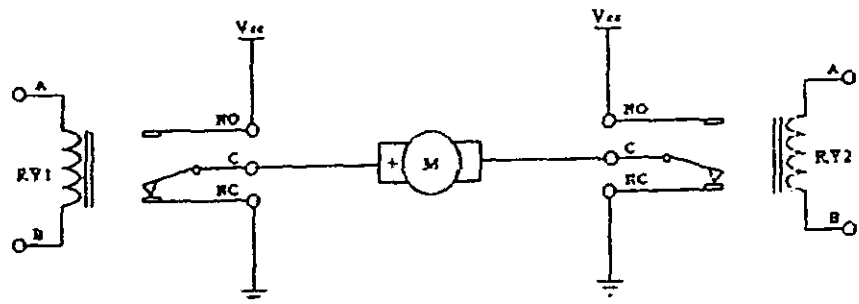
- หน้าสัมผัสแบบปกติปิด (Normally close หรือ NC.) ซึ่งในสถานะปกติ ขานี้จะต่อกับขาร่วม (Common)
- หน้าสัมผัสแบบปกติเปิด (Normally open หรือ NO.) ขานี้จะต่อเข้ากับขาร่วม (Common) เมื่อขดลวดมีแรงดันตกคร่อม หรือกระแสไหลผ่านในปริมาณที่เพียงพอ

ในรีเลย์ 1 ตัว อาจมีหน้าสัมผัสมากกว่า 1 ชุด เช่น 2 ชุด, 4 ชุด เป็นต้นแล้วแต่ผู้ผลิต

เมื่อขดลวดได้รับแรงดันตกคร่อม ขา A และ B จะทำให้อุปกรณ์กระแสไหลผ่านขดลวด ซึ่งจะทำให้เกิดอำนาจสนามแม่เหล็กดึงดูดให้หน้าสัมผัส NO และ C ติดกันดังรูป



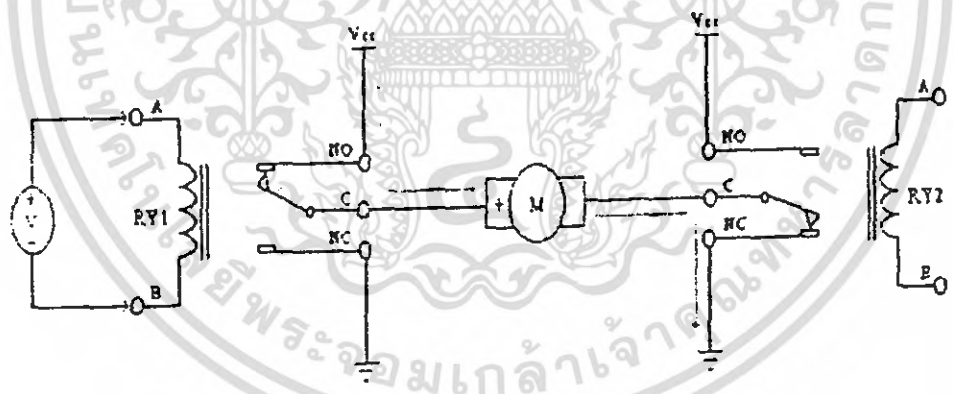
รูปที่ 2.18 สถานะเมื่อได้รับแรงดันตกคร่อม ขา A และ B



รูปที่ 2.19 วงจรในสถานะเริ่มต้น

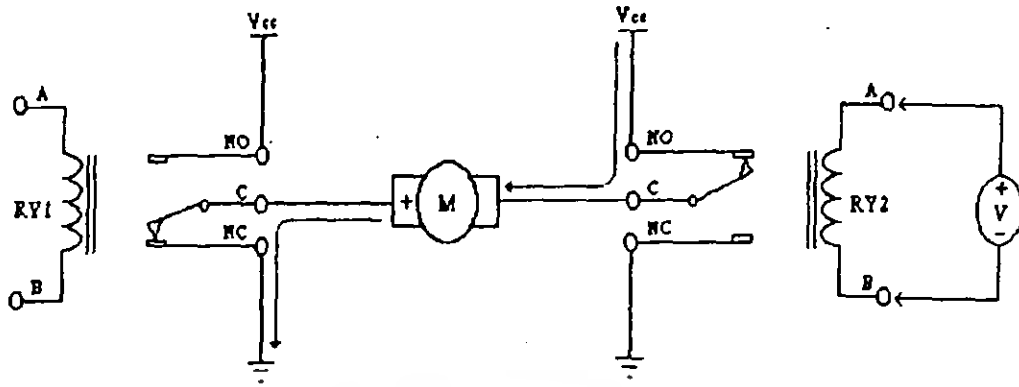
วงจรที่เราจะทดลองกันในคราวนี้ จะประกอบด้วย รีเลย์ 2 ตัว คือ RY1 และ RY2 ซึ่ง Load ก็คือ DC-motor ซึ่งต่ออยู่กับขั้วร่วม (C.) ของ RY1 และ RY2 โดยขั้วบวกของมอเตอร์ต่ออยู่ที่ขา C. ของ RY1 และขั้วลบของมอเตอร์ต่ออยู่ที่ขา C. ของ RY2 โดยที่ขา NO. ของ RY1 และ RY2 จะต่ออยู่กับขั้วบวกของแหล่งจ่ายไฟ ที่จะจ่ายให้มอเตอร์ (V_{cc}) และขา NC. ของ RY1 และ RY2 จะต่อลงกราวด์

กรณีที่ RY1 ทำงาน เมื่อ RY1 ทำงาน (มีกระแสไหลผ่านขดลวดในปริมาณที่เพียงพอ) จะทำให้เกิดอำนาจสนามแม่เหล็กไฟฟ้า ดึงดูดให้ขา NO และขา C. ของ RY1 ติดกัน ส่งผลให้มีกระแสจากแหล่งจ่าย (V_{cc}) ผ่านเข้าสู่ขั้วบวกของมอเตอร์ ผ่านไปยังขา C. ของ RY2 ซึ่งต่ออยู่ที่ NC และลงกราวด์ ทำให้มีกระแสไหลผ่านมอเตอร์ในทิศทางบวก และครบวงจรจึงทำให้มอเตอร์สามารถหมุนในทิศทาง Forward ได้ดังรูป



รูปที่ 2.20 วงจรในสถานะที่ RY1 ทำงาน

กรณีที่ RY2 ทำงาน เมื่อ RY2 ทำงาน (มีกระแสไหลผ่านขดลวดในปริมาณที่เพียงพอ) จะทำให้เกิดอำนาจสนามแม่เหล็กไฟฟ้าดึงดูดให้ขา NO. และ C. ของ RY2 ติดกัน ส่งผลให้มีกระแสไหลจากแหล่งจ่าย (V_{cc}) ผ่านเข้าสู่ขั้วลบของมอเตอร์ ผ่านไปยังขา C. ของ RY1 ซึ่งต่ออยู่ที่ NC และลงกราวด์ ทำให้มีกระแสไหลผ่านมอเตอร์ในทิศทางลบ และครบวงจรจึงทำให้มอเตอร์สามารถหมุนในทิศทาง Reverse ได้ดังรูป



รูปที่ 2.21 วงจรในสถานะที่ RY2 ทำงาน

ถึงแม้ว่ารีเลย์จะเป็นอุปกรณ์ที่นำมาประยุกต์สร้างเป็นวงจร H-Bridge ได้ง่าย ไม่ยุ่งยากก็ตาม แต่รีเลย์นั้นเป็นอุปกรณ์แม่เหล็กที่มีส่วนเคลื่อนไหวกว้างไกล เพราะฉะนั้นก็ย่อมจะมีข้อจำกัดทางกลอยู่บ้าง เช่น

ผลกระทบของการตอบสนอง รีเลย์เป็นอุปกรณ์ที่มีความเร็วในการทำงานต่ำ เช่นรีเลย์ชนิดแรงดันต่ำ (กระตุ้นขดลวดไม่เกิน 24 V.) จะใช้เวลาในการทำงานประมาณ 10 – 50 ms และรีเลย์ขนาดใหญ่ที่ใช้ควบคุมมอเตอร์ในโรงงานอุตสาหกรรมนั้น อาจใช้เวลาในการทำงานมากกว่า 100 ms เสียทีเดียว

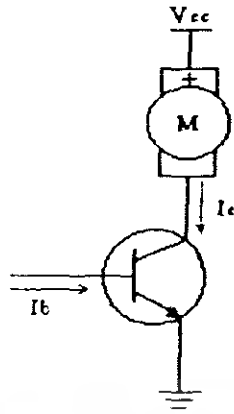
ผลกระทบจากอำนาจแม่เหล็ก รีเลย์เป็นอุปกรณ์แม่เหล็ก ดังนั้นจึงไม่สามารถหนีปัญหานี้ได้ จึงไม่ใช่เรื่องแปลกที่หลายคนอาจเคยมีปัญหารีเลย์ไปรบกวนการทำงานของวงจรไมโครฯ การแก้ไขอาจมีหลายวิธี เช่น

- แยกกราวด์ คือ การแยกกราวด์ของแหล่งจ่ายไฟสำหรับวงจรไมโครฯ และแหล่งจ่ายไฟกระตุ้นรีเลย์ออกจากกัน โดยใช้อุปกรณ์ Opto Coupler
- แยกบอร์ด คือ การแยกการทำงานในส่วนของวงจรรีเลย์ ออกไปจากบอร์ดไมโครฯ แล้วทำการซึลด์ให้ดี
- ลดขนาดแรงดันกระตุ้น คือ การเปลี่ยนตัวรีเลย์ เช่นเปลี่ยนจากรีเลย์ขนาด 24 V มาเป็นขนาด 12 V

2.6.3 การควบคุมทิศทางหมุนของ DC Motor สร้างวงจร H-Bridge Switching จาก Transistor

ทรานซิสเตอร์เป็นอุปกรณ์สารกึ่งตัวนำ (Semiconductor device) ที่เราสามารถนำคุณสมบัติของการ Cutoff และการ Saturation มาประยุกต์ใช้งานเป็นสวิทช์ได้ และที่สำคัญมันเป็นสวิทช์อิเล็กทรอนิกส์ ที่เราสามารถควบคุมการปิด/เปิดได้

จากรูป 2.20 เป็นวงจรสวิตช์แบบง่ายๆ โดยการนำทรานซิสเตอร์มาเป็นสวิทช์ควบคุมมอเตอร์ หลักการคิดง่ายๆ ก็คือ เมื่อเราป้อนกระแส I_B ด้วยปริมาณที่มากพอก็จะทำให้ทรานซิสเตอร์ทำงาน (On) จะทำให้กระแส I_C ไหล แปลว่ามีกระแสไหลผ่านมอเตอร์ได้ กระแส I_B มากเพียงพอที่จะทำให้ทรานซิสเตอร์อยู่ในสถานะอื่นตัวได้

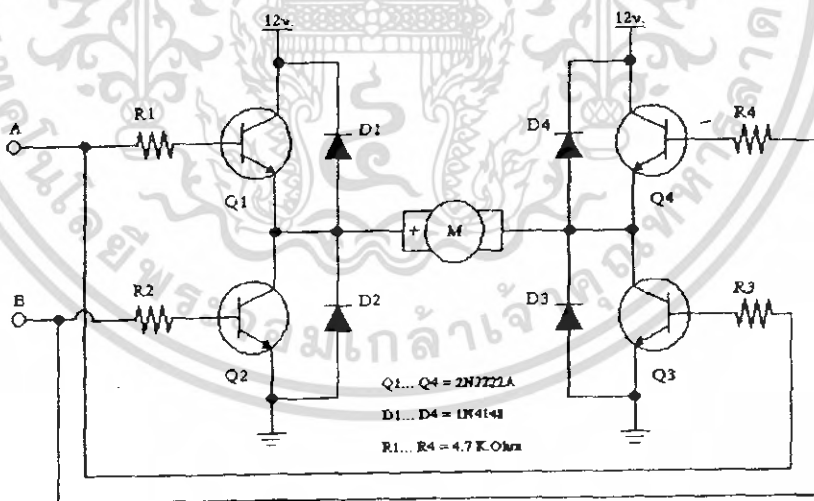


รูปที่ 2.22 วงจรสวิตช์แบบง่ายใช้ทรานซิสเตอร์ทำหน้าที่เป็นสวิตช์ควบคุมมอเตอร์

ในสถานะอิ่มตัว (Saturation mode) นี้ ทรานซิสเตอร์จะทำงานเหมือนกับสวิตช์ปิดวงจร ค่าความต้านทานระหว่างขา C และขา E จะมีค่าต่ำใกล้เคียงศูนย์ กระแส I_c ที่ไหลจะมีค่าเข้าใกล้ I_c (max)

ในสถานะคัทออฟ (Cutoff mode) นี้ จะเกิดขึ้นเมื่อเราหยุดจ่ายกระแส I_b ($I_b = 0$) ทรานซิสเตอร์จะทำงานเหมือนกับสวิตช์เปิดวงจร ค่าความต้านทานระหว่างขา C และขา E จะมีค่าเป็นอนันต์ กระแส I_c จะมีค่าเข้าใกล้ศูนย์

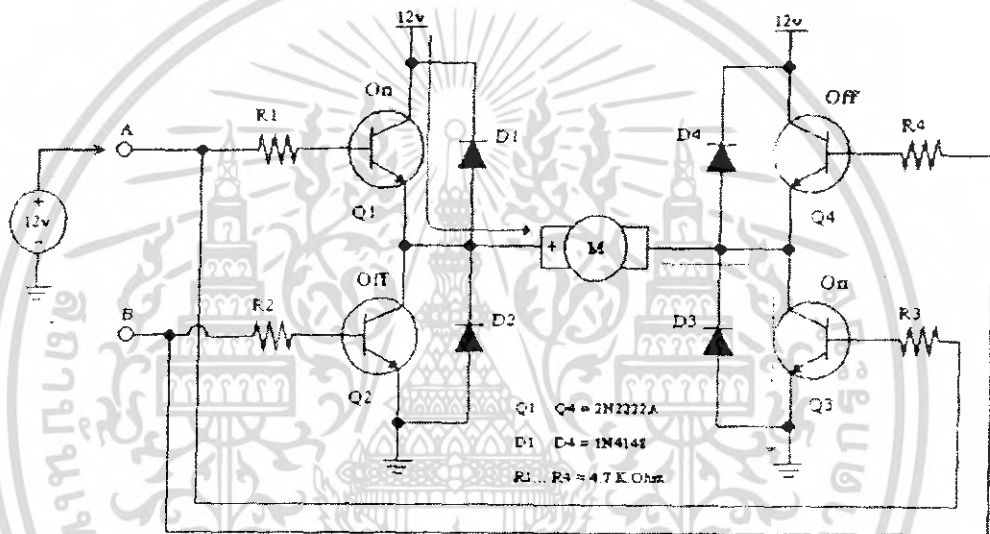
ข้อดีของการนำทรานซิสเตอร์มาประยุกต์ใช้งานเป็นสวิตช์นั้น น่าจะเป็นส่วนของการควบคุมที่สามารถตอบสนองจังหวะของการเปิด/ปิด สวิตช์ได้นับล้านครั้งต่อวินาที และที่สำคัญคือไม่ทำให้เกิดปัญหาการบวมจากสนามแม่เหล็กอย่างเร็วเลย



รูปที่ 2.23 วงจรในสถานะเริ่มต้น

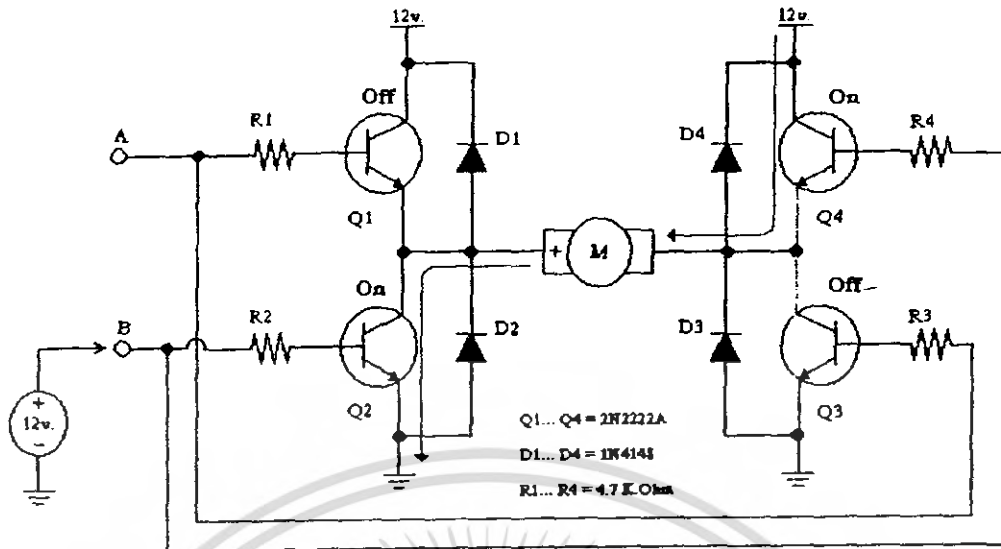
วงจรที่เราจะทดลองกันคราวนี้ จะประกอบด้วยทรานซิสเตอร์ Q1 Q2 Q3 และ Q4 ทุกตัวใช้เบอร์ 2N2222A ส่วน R1 R2 R3 และ R4 ใช้ค่าความต้านทาน 4.7k ทำหน้าที่จำกัดกระแส I_b ส่วนไดโอด D1 D2 D3 และ D4 ใช้เบอร์ 1N4148 ทำหน้าที่ป้องกันกระแสไหลย้อนกลับจากมอเตอร์ในขณะที่ทรานซิสเตอร์หยุดทำงาน แหล่งจ่ายของวงจรจะใช้เป็นแบตเตอรี่รีชขนาด 12 V หรือจาก Power Supply ก็ได้ ส่วนมอเตอร์นั้นให้ใช้เป็นมอเตอร์ตัวเล็กเล็กๆ เนื่องจากจำกัดกระแสของวงจรเอาไว้ที่ 500 mA

กรณีที่ Q1 และ Q3 ทำงาน เมื่อมีการจ่ายแรงดัน 12 โวลต์ เข้าที่จุด A ทำให้มีกระแสไหลผ่าน R1 เข้าสู่ base ของ Q1 และมีกระแสไหลผ่าน R3 เข้าสู่ base ของ Q3 ทำให้ Q1 และ Q3 ทำงาน เปรียบเสมือนสวิตช์ปิดวงจร ส่งผลให้กระแสไหลจากแหล่งจ่าย (12 V) ผ่านขา Collector และ Emitter ของ Q1 ผ่านเข้าสู่ขั้วบวกของมอเตอร์ ผ่านไปยังขา Collector และ Emitter ของ Q3 ทำให้มีกระแสไหลผ่านมอเตอร์ในทิศทางบวกและครบวงจร จึงทำให้มอเตอร์สามารถหมุนในทิศทาง Forward ได้



รูปที่ 2.24 สภาวะของวงจรขณะที่ Q1 และ Q3 ทำงาน

กรณีที่ Q2 และ Q4 ทำงาน เมื่อมีการจ่ายแรงดัน 12 V เข้าที่จุด B ทำให้มีกระแสไหลผ่าน R2 เข้าสู่ base ของ Q2 และมีกระแสไหลผ่าน R4 เข้าสู่ base ของ Q4 ทำให้ Q2 และ Q4 ทำงาน เปรียบเสมือนสวิตช์ปิดวงจร ส่งผลให้กระแสไหลจากแหล่งจ่าย (12 V) ผ่านขา Collector และ Emitter ของ Q4 ผ่านเข้าสู่ขั้วลบของมอเตอร์ ผ่านไปยังขา Collector และ Emitter ของ Q2 ทำให้มีกระแสไหลผ่านมอเตอร์ในทิศทางลบและครบวงจร จึงทำให้มอเตอร์สามารถหมุนในทิศทาง Reward ได้



รูปที่ 2.25 สถานะของวงจรขณะ Q2 และ Q4 ทำงาน

2.6.3.1 ข้อควรระวัง

อย่าทำให้ทรานซิสเตอร์ทั้ง 4 ตัว ทำงานพร้อมกันอย่างเด็ดขาด เพราะจะทำให้เกิดการลัดวงจร ถึงแม้ว่าการสร้างวงจร H-Bridge switching จากทรานซิสเตอร์นั้นจะไม่มีปัญหา การรบกวนจากอำนาจสนามแม่เหล็ก และยังสามารถตอบสนองการทำงานได้เร็วมาก (เร็วกว่ารีเลย์แชนเท้า) แต่ก็ยังมีอุปสรรคหลายตัว ทำให้วงจรมีขนาดใหญ่และวุ่นวาย

2.6.4 การนำ L298 มาใช้เป็น IC ควบคุมมอเตอร์

การนำ L298 ไปใช้งานนั้นไม่มีอะไรยุ่งยาก ไม่จำเป็นต้องคอยอุปกรณ์มากมาย เหมือน ตอนใช้ทรานซิสเตอร์ เพียงมันเองก็สามารถทำงานได้ L298 สามารถให้กระแสได้สูงสุดถึง 4 A ควบคุมมอเตอร์ได้ 2 ตัว โดยที่ใช้กับแรงดันไฟฟ้าสูงสุดได้ 46 V ทนความร้อนได้ดี จะใช้มาตรฐาน TTL เป็นลอจิกอินพุต แต่ต้องใช้อุปกรณ์ที่ป้องกัน back-EMF ในที่นี้เราใช้ IC เบอร์ 74LS245 เป็นบัฟเฟอร์ เพื่อป้องกัน back-EMF และ ใช้ drive อุปกรณ์เช่น รีเลย์ solenoids และ DC motor

บทที่ 3

การวางแผนและการออกแบบ

3.1 การวางแผนการดำเนินงาน

แผนการดำเนินงานประกอบด้วย 10 ขั้นตอน ได้แก่

1. วางแผนการทำงาน
2. ศึกษาและวิเคราะห์โครงสร้างของแขนกล
3. ทำการปรับปรุงแก้ไขในจุดที่บกพร่องของแขนกล 5 แกน
4. ศึกษาหาข้อมูลเกี่ยวกับวงจรควบคุม
5. ศึกษาภาษาซีเพื่อควบคุม Microcontroller
6. ศึกษาเกี่ยวกับ โปรแกรม Delphi
7. วางแผนสร้างวงจรควบคุม
8. ทำการเขียน โปรแกรม
9. ทำการทดสอบการทำงานและสรุปผล
10. จัดทำปฏิญานินพนธ์

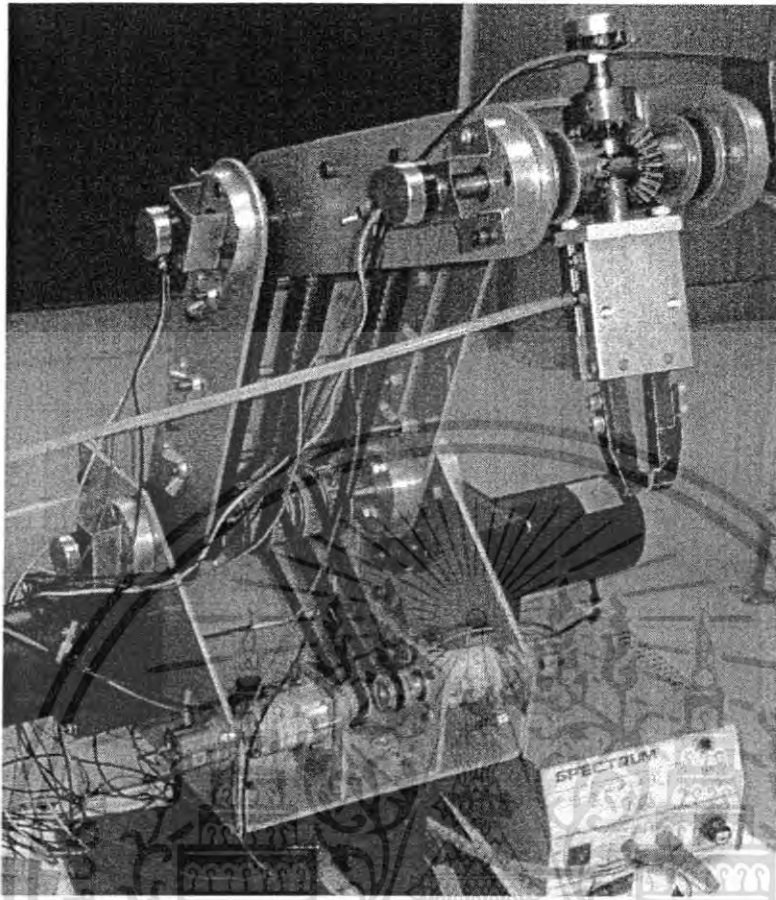
3.2 แผนการดำเนินงาน

ในส่วนของการออกแบบและพัฒนาการควบคุมการทำงานของแขนกล 5 แกน โดยใช้คอมพิวเตอร์ ได้ทำการวางแผนการทำงานออกเป็น 3 ส่วน คือ ส่วนของฮาร์ดแวร์ ส่วนของวงจรอิเล็กทรอนิกส์ และส่วนของ โปรแกรมที่ใช้ติดต่อและควบคุมการทำงาน เพื่อให้ง่ายต่อการทำความเข้าใจ รายละเอียดในการออกแบบแต่ละส่วน มีดังนี้

3.2.1 การดำเนินงานด้านฮาร์ดแวร์

ขั้นตอนการดำเนินงานด้านฮาร์ดแวร์ประกอบด้วย

1. ศึกษาการเคลื่อนที่ของแขนกลเดิม
2. ทำการตรวจสอบหาจุดบกพร่องของแขนกล และทำการประเมินสภาพของแขนกลเดิม
3. ดำเนินการจัดซื้อวัสดุและอุปกรณ์ที่ต้องใช้ในการซ่อมแซมและต่อเติม
4. ดำเนินการแก้ไขในส่วนที่เสียหาย
5. ทำการติดตั้ง โฟเทนทีโอเมเตอร์ตามข้อต่อแกนการเคลื่อนที่ และนำวงจรควบคุมมาติดตั้งกับแขนกล

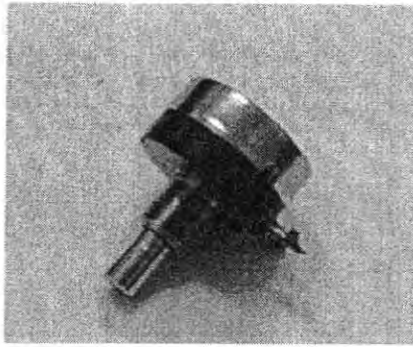


รูปที่ 3.1 ตำแหน่งของโพเทนทีโอมิเตอร์บนแขนกล 5 แกน

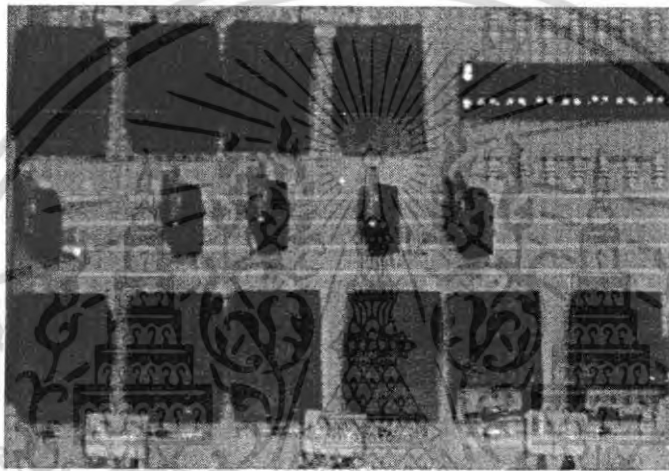
3.2.2 การดำเนินงานด้านวงจรอิเล็กทรอนิกส์

ส่วนของการออกแบบวงจรสำหรับรับ-ส่งข้อมูลระหว่าง ฮาร์ดแวร์ กับ ซอฟต์แวร์ วงจรที่นำมาใช้งานจะเลือกตามลักษณะการทำงานที่ต้องการ โดยสามารถใช้งานกับเซ็นเซอร์ตัวบอกตำแหน่งของแขนกลแต่ละแกน ซึ่งจะอ่านค่าผ่านการทำงานของ PIC 16F877A ซึ่งเป็นไมโครคอนโทรลเลอร์ โดยจะทำการแปลงค่าจากอนาล็อก ที่อ่านค่ามาจากตัวเซ็นเซอร์ให้เป็นค่าดิจิทัล เพื่อให้คอมพิวเตอร์จะสามารถรับค่าได้และจะแสดงผลผ่านทางพอร์ตอนุกรม (Serial Port) และส่วนของการประมวลผลนั้นจะทำการประมวลผลจากไมโครคอนโทรลเลอร์ตระกูล PIC 16F877A ซึ่งจะทำหน้าที่ขับเคลื่อนมอเตอร์ และทำการตรวจสอบตำแหน่งของแขนกล

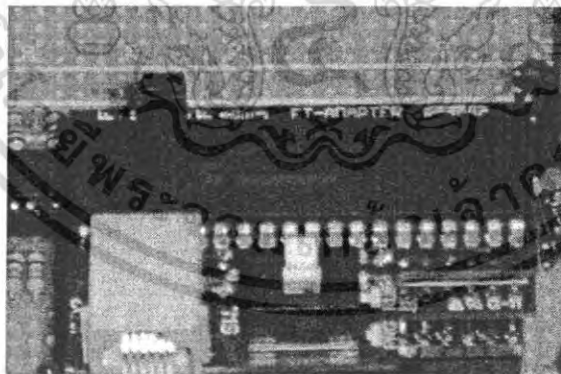
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โปเทนติโอมิเตอร์ที่ใช้ในวงจร



รูปที่ 3.3 รีเลย์ที่ใช้ในการขับเคลื่อนมอเตอร์



รูปที่ 3.4 ไมโครคอนโทรลเลอร์ตระกูล PIC 16F877A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ตัวเชื่อมต่อพอร์ตอนุกรม (Serial Port RS232)

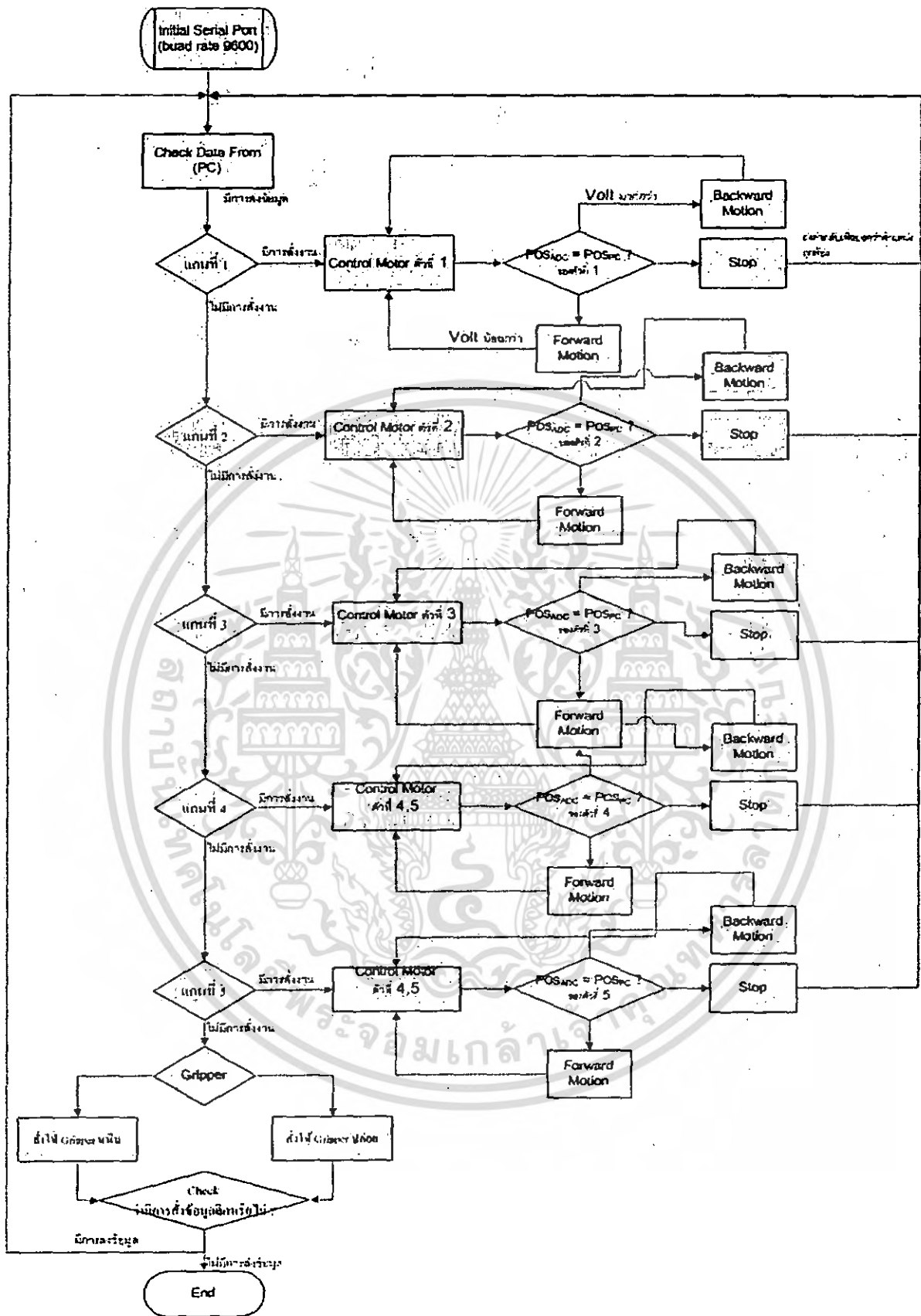
3.2.3 การดำเนินงานด้านซอฟต์แวร์

ได้ทำการศึกษาการเขียนโปรแกรมภาษาซี และ Delphi เพื่อควบคุมแขนกลให้สามารถรับ-ส่งข้อมูล โดยในส่วนของภาษานั้นจะถูกเขียนลงบน Microcontroller ซึ่งจะเป็นตัวประมวลผลให้แขนกลนั้นเคลื่อนที่ รวมถึงการบันทึกตำแหน่งและการตรวจสอบตำแหน่ง ซึ่งจะทำงานก็ต่อเมื่อรับค่ามาจากคอมพิวเตอร์ ในส่วนของ Delphi จะทำหน้าที่เป็นโปรแกรมอินเตอร์เฟซ ระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ โดยแบ่งการทำงานออกเป็น

1. การศึกษาการเขียนแผนภาพการทำงาน (Flow Chart) ของโปรแกรม
2. การออกแบบหน้าต่างของโปรแกรม
3. การสร้างรูปแบบของโปรแกรม
4. การตรวจสอบการทำงานของโปรแกรม
5. การทดลองโปรแกรมกับแขนกล 5 แกน

โดยในส่วนของไมโครคอนโทรลเลอร์มีผังการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 ผังการทำงานของไมโครคอนโทรลเลอร์

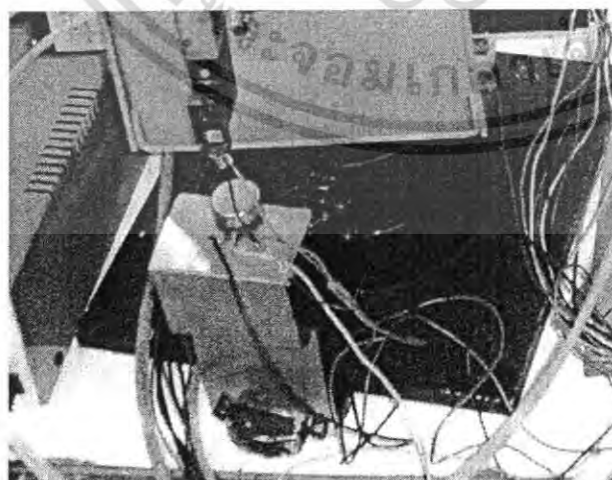
บทที่ 4

ผลการดำเนินงาน

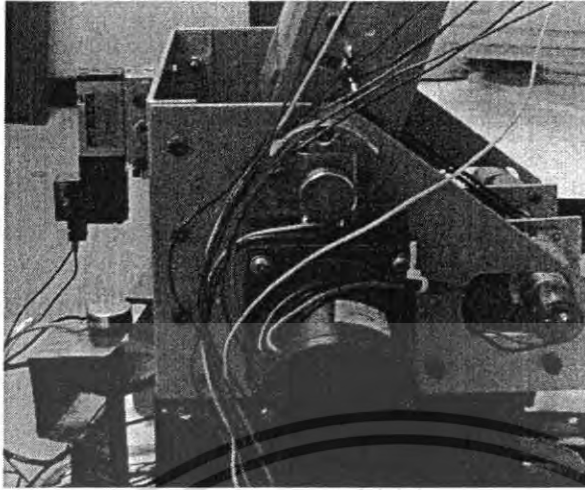
การศึกษาของโครงการแบ่งได้ 3 ด้าน คือ ด้านฮาร์ดแวร์ ซึ่งเกี่ยวข้องกับการศึกษาหลักการการทำงานของแขนกล 5 แขน โดยพิจารณาถึงความสามารถในการใช้งานของแขนกล 5 แขน รวมทั้งซ่อมแซมส่วนที่เสียหายของแขนกล 5 แขน ส่วนต่อมาในด้านของวงจรควบคุมการทำงาน การออกแบบวงจรสำหรับรับ-ส่งข้อมูลระหว่าง ฮาร์ดแวร์ และ ซอฟต์แวร์ วงจรที่นำมาใช้งานจะเลือกตามลักษณะการทำงานที่ต้องการ โดยเลือกใช้ไมโครคอนโทรลเลอร์ PIC 16F877A และใช้รีเลย์ (Relay) ในการควบคุมการเคลื่อนที่ของมอเตอร์ ซึ่งไมโครคอนโทรลเลอร์จะทำการรับค่าจากโพเทนทิโอมิเตอร์ (Potentiometer) ทั้ง 5 ตัว แล้วทำการแปลงค่าที่อ่านได้จาก อนาล็อก (Analog) เป็นดิจิทัล (Digital) เพื่อที่คอมพิวเตอร์จะสามารถรับค่าได้ และจะแสดงผลบนหน้าจอกอมพิวเตอร์ โดยผ่านพอร์ตอนุกรม (Serial Port) ส่วนสุดท้าย คือ ส่วนของโปรแกรมควบคุมการทำงานของแขนกล 5 แขน ทำการศึกษาการเขียนโปรแกรมด้วย Delphi เพื่อเขียนโปรแกรมควบคุมแขนกล 5 แขน ให้สามารถรับ-ส่งข้อมูลการควบคุมการเคลื่อนที่ระหว่างแขนกล 5 แขนกับคอมพิวเตอร์

4.1 ผลการดำเนินงานด้านฮาร์ดแวร์

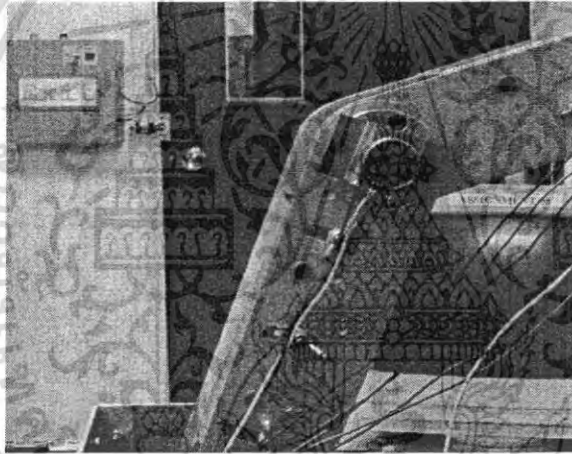
จากการตรวจสอบ ได้มีการทำการซ่อมแซมส่วนที่เสียหายให้แขนกล 5 แขนอยู่ในสภาพที่สมบูรณ์ สามารถทำงานได้เต็มประสิทธิภาพ และได้มีการทำการติดตั้งโพเทนทิโอมิเตอร์ รวมทั้งทำการออกแบบ และติดตั้งส่วนที่ทำการยึดจับโพเทนทิโอมิเตอร์ทั้ง 5 แขนดังแสดงดังรูป



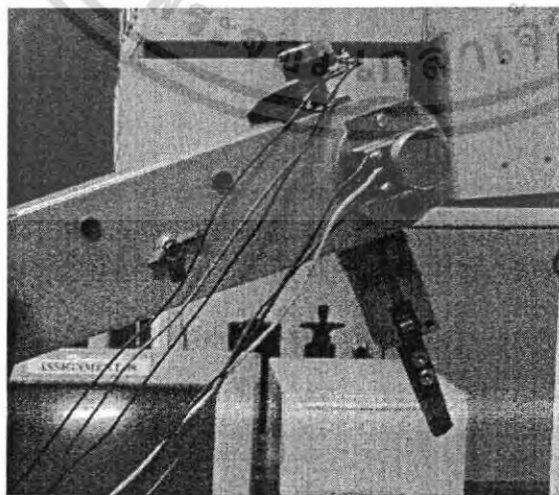
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การติดตั้งโพเทนทีโอมิเตอร์ที่แกนที่ 2

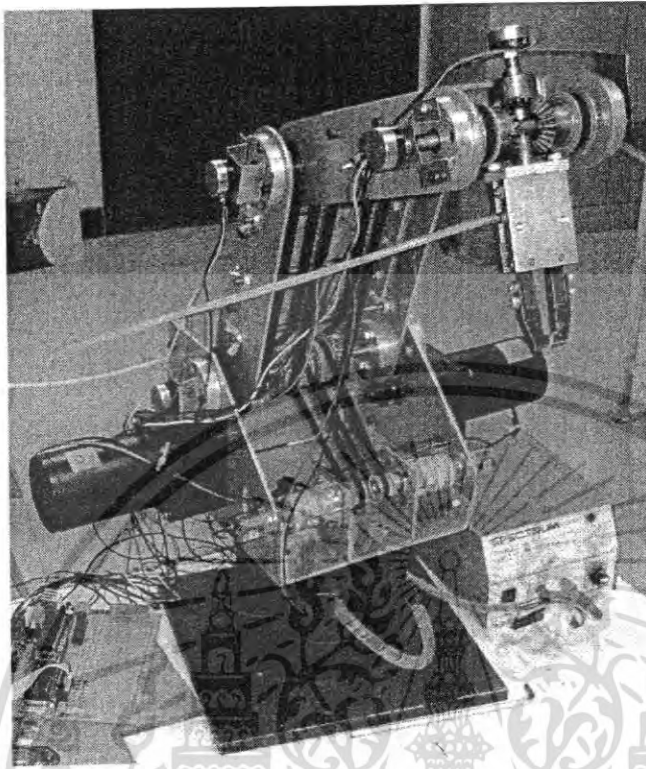


รูปที่ 4.3 การติดตั้งโพเทนทีโอมิเตอร์ที่แกนที่ 3



รูปที่ 4.4 การติดตั้งโพเทนทีโอมิเตอร์ที่แกนที่ 4 และ 5

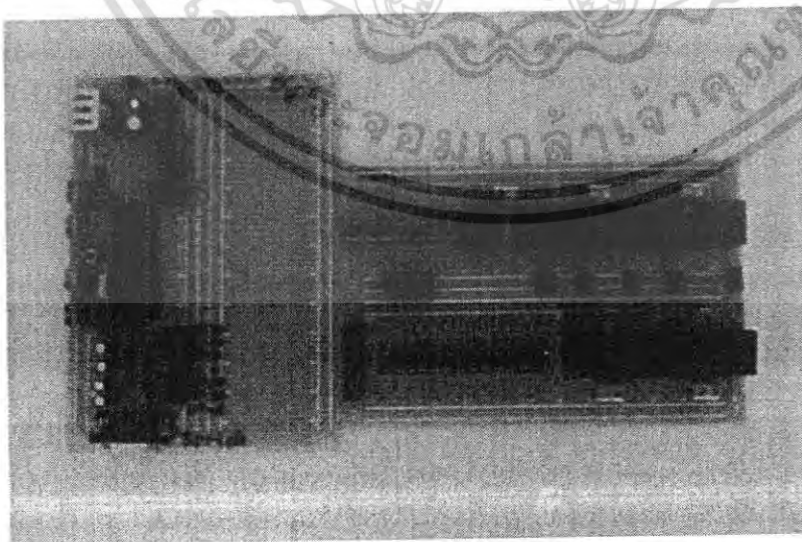
เอกสารนี้เป็นเอกสารที่ลงนามไว้สำหรับใช้เพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แชนกัล 5 แคนที่เสร็จสมบูรณ์แล้ว

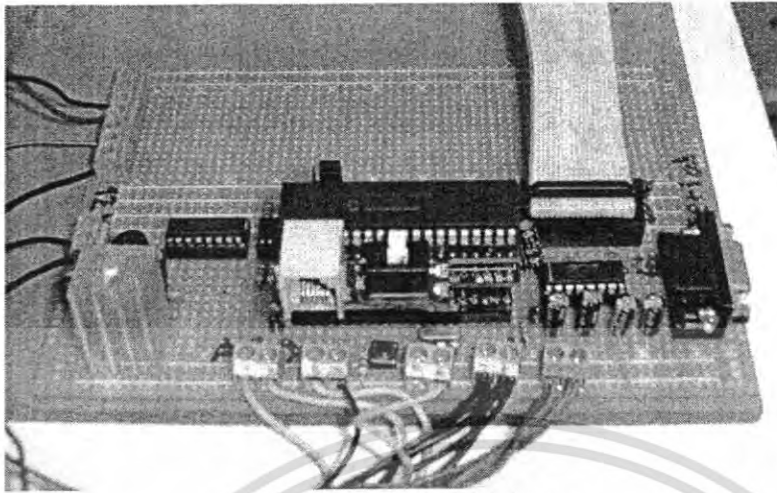
4.2 ผลการออกแบบ และสร้างวงจรควบคุมการทำงาน

จากขั้นตอนการออกแบบ และสร้างวงจรควบคุมการทำงานแสดงได้ดังนี้

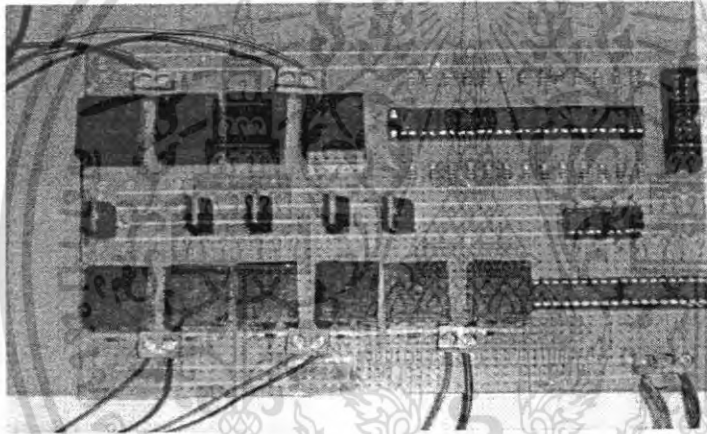


รูปที่ 4.6 วงจรที่ใช้ควบคุมแชนกัล 5 แคน

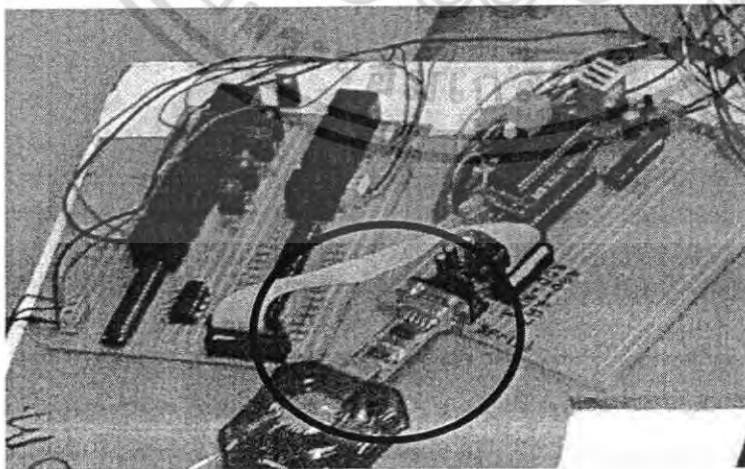
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 วงจรส่วนของไมโครคอนโทรลเลอร์



รูปที่ 4.8 วงจรส่วนของรีเลย์ควบคุมมอเตอร์

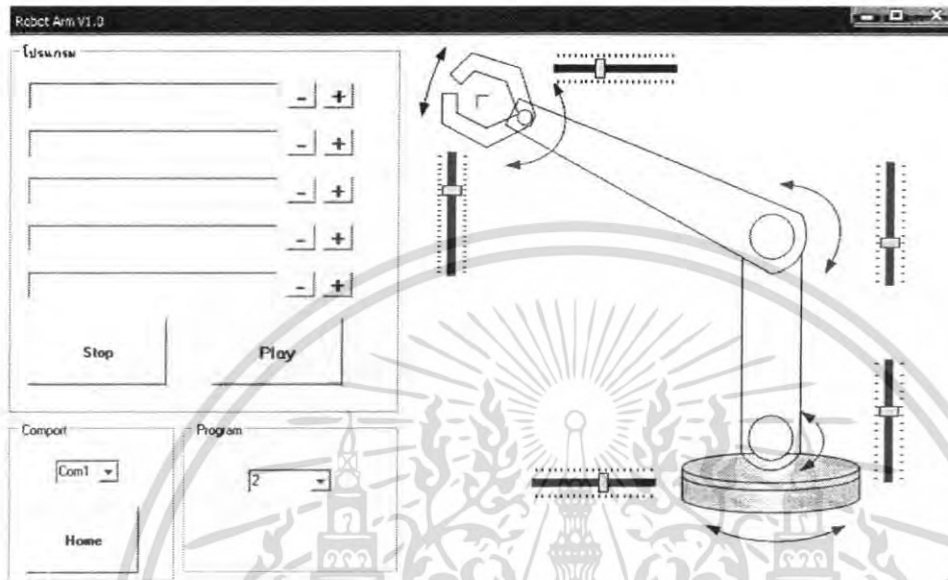


รูปที่ 4.9 ส่วนเชื่อมต่อวงจรเข้ากับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการออกแบบ และสร้างโปรแกรมควบคุมการทำงาน

ผลการออกแบบและสร้าง โปรแกรมที่ใช้ในการควบคุมการเคลื่อนที่ของแขนกล 5 แกน ดังรูป



รูปที่ 4.10 หน้าต่างโปรแกรมควบคุมการเคลื่อนที่ของแขนกล 5 แกน

4.4.1 ขั้นตอนการทำงานของโปรแกรม

โปรแกรมมีขั้นตอนการทำงาน ดังนี้

1. เลือกคอมพอร์ต (Port)

เลือกพอร์ต (Port) ที่เชื่อมต่อระหว่างคอมพิวเตอร์กับแขนกล 5 แกน ให้ถูกต้อง

2. ตั้งค่าเริ่มต้น

ก่อนทำการตั้งให้แขนกล 5 แกนเคลื่อนที่ ต้องทำการตั้งค่าให้แขนกล 5 แกนอยู่ในตำแหน่งเริ่มต้น (Home)

3. ใช้รูปแบบการบังคับด้วยตัวเองโดยตรงก่อน

4. เก็บค่าเข้าหน่วยความจำ

เมื่อได้รูปแบบการทำงานของแขนกล 5 แกนแล้ว ให้ทำการเก็บค่าโดยกดปุ่มเครื่องหมายบวกเพื่อทำการเก็บค่าที่ต้องการไว้ จนครบทุกตำแหน่ง โดยจะเก็บได้สูงสุด 5 รูปแบบการเคลื่อนที่

5. เลือกจำนวนโปรแกรมให้ตรงกับที่ทำการบันทึกไว้

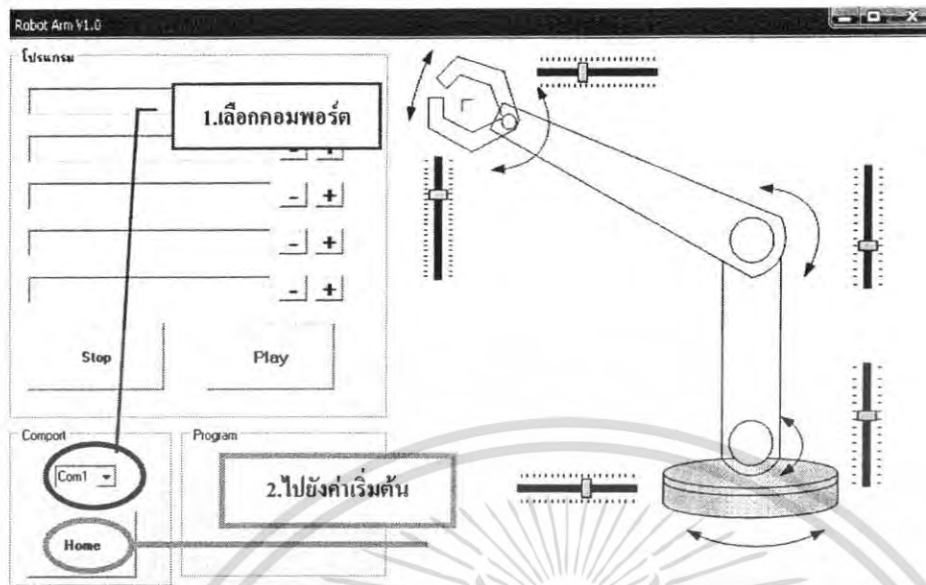
6. เริ่มการทำงานของแขนกล 5 แกน

หลังจากเลือกจำนวนโปรแกรมเรียบร้อยแล้ว กดปุ่มเล่น (Play) เพื่อให้โครงหุ่นยนต์เคลื่อนที่ตามที่บันทึกไว้โดยอัตโนมัติ

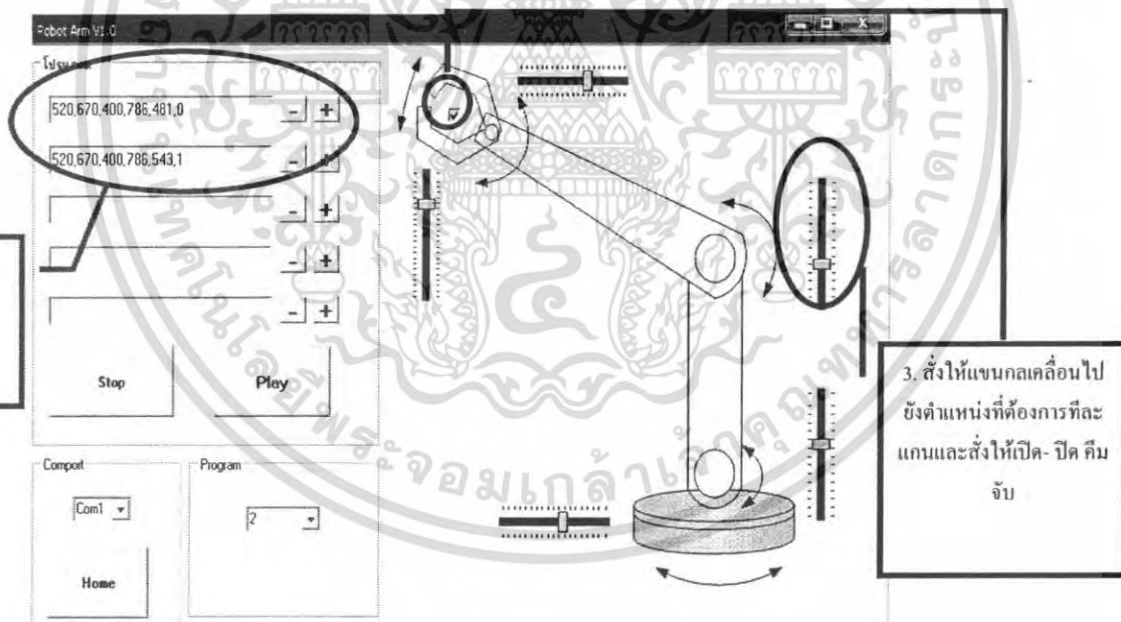
7. หากมีเหตุขัดข้องเกิดขึ้น สามารถกดหยุดอัตโนมัติได้โดยกดปุ่มหยุด (Stop) โครงหุ่นยนต์จะหยุด

เคลื่อนที่ทันที ดังแสดงตามรูป

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

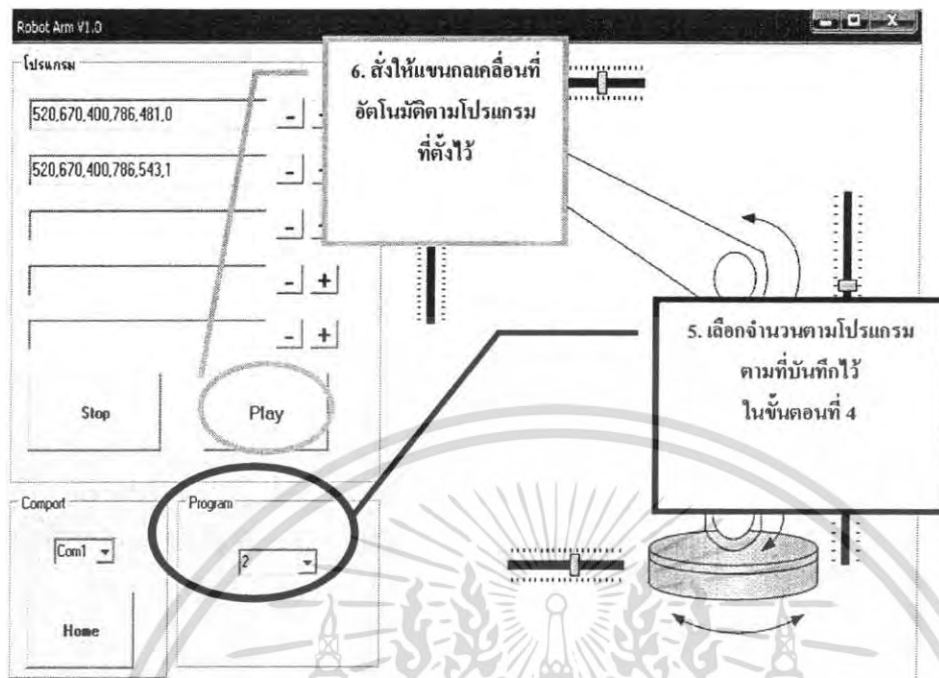


รูปที่ 4.11 ขั้นตอนการทำงานของโปรแกรมขั้นตอนที่ 1 และ 2

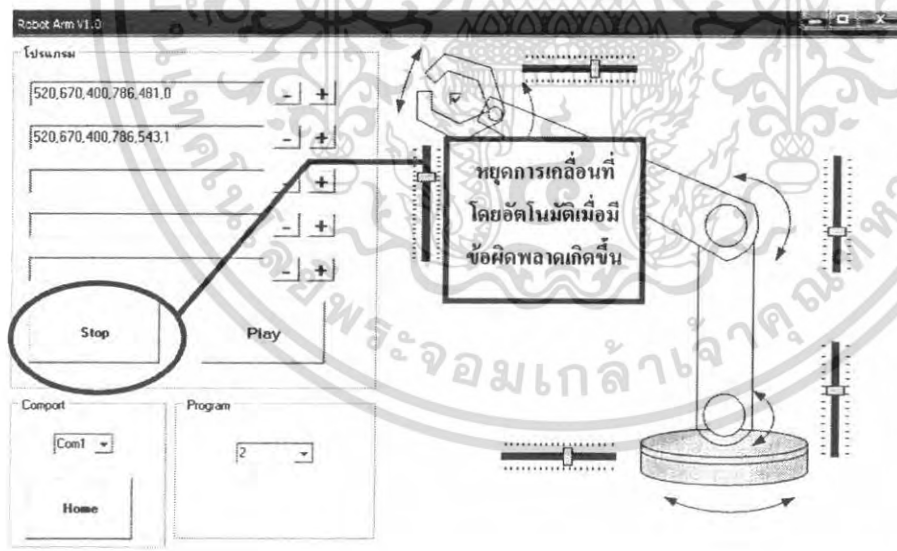


รูปที่ 4.12 ขั้นตอนการทำงานของโปรแกรมขั้นตอนที่ 3 และ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 ขั้นตอนการทำงานของโปรแกรมขั้นตอนที่ 5 และ 6



รูปที่ 4.14 ปุ่มหยุดการทำงานอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

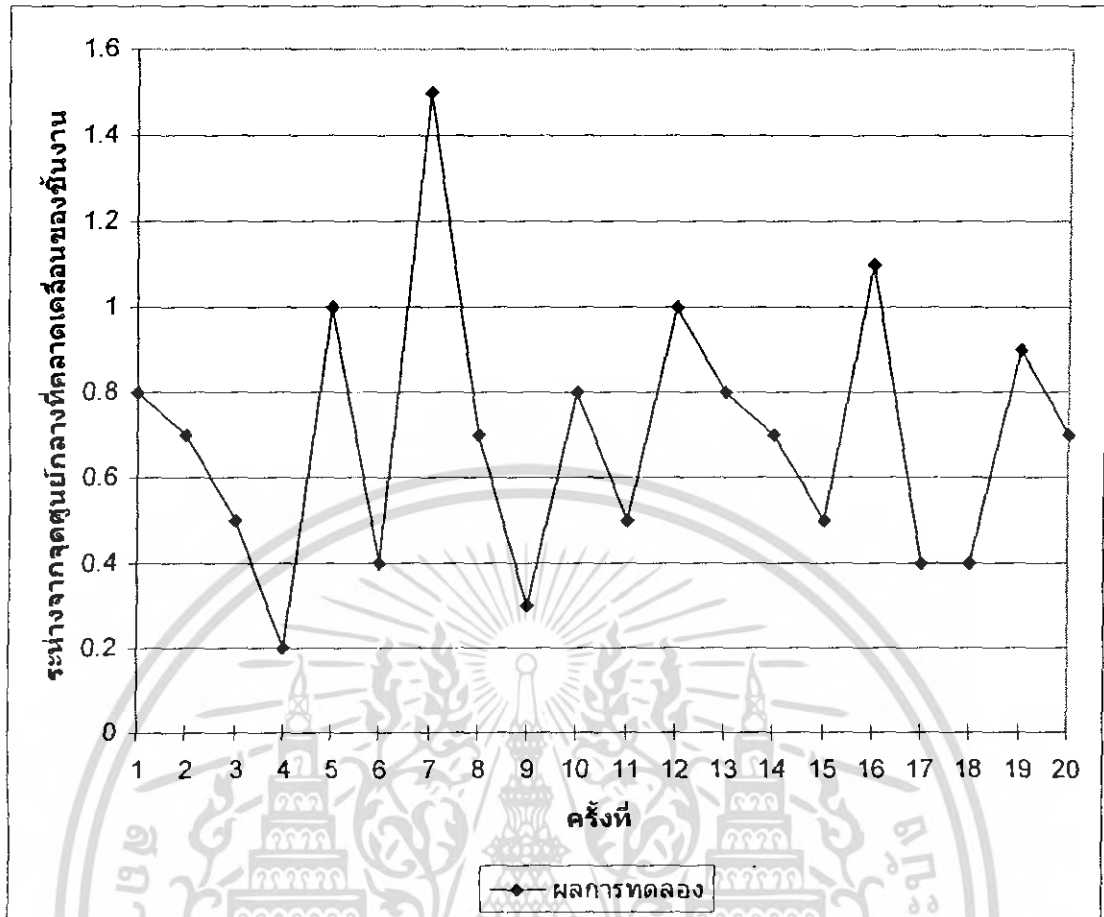
4.4 ผลการทดลอง

เพื่อทดสอบความแม่นยำในการเคลื่อนที่ของแขนกล 5 แกน ได้มีการออกแบบ และ ทำการทดลองโดยทดลองให้แขนกล 5 แกน ทำการเคลื่อนที่เพื่อไปหยิบชิ้นงานตัวอย่างจากตำแหน่งเริ่มต้นที่กำหนดไว้ ไปยังอีกตำแหน่งหนึ่งที่ต้องการ แล้วทำการวัดระยะทางที่คลาดเคลื่อน ไประหว่างจุดศูนย์กลางของชิ้นงาน แล้วนำค่าที่ได้ไปสร้างกราฟผลการทดลองแสดงได้ดังนี้

ตารางที่ 4.1 ผลการทดลอง

ครั้งที่ทำการทดลอง	ระยะห่างระหว่างจุดศูนย์กลางที่คลาดเคลื่อนไปของชิ้นงาน (cm)
1	0.8
2	0.7
3	0.5
4	0.2
5	1
6	0.4
7	1.5
8	0.7
9	0.3
10	0.8
11	0.5
12	1
13	0.8
14	0.7
15	0.5
16	1.1
17	0.4
18	0.4
19	0.9
20	0.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ 38 อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 กราฟเส้นความสัมพันธ์ระหว่างครั้งที่ทำการทดลอง และ ระยะห่างระหว่างจุดศูนย์กลางของชิ้นงานที่คลาดเคลื่อน

จากการทดลองทั้งหมด 20 ครั้ง จะมิตัวความคลาดเคลื่อนของระยะห่างระหว่างจุดศูนย์กลางของชิ้นงานเฉลี่ยเท่ากับ 0.695 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ 39 อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิเคราะห์การดำเนินงาน

ในสายงานอุตสาหกรรมที่ทันสมัยจำเป็นต้องมีการใช้งานแขนกลอัตโนมัติเข้ามาใช้ในสายการผลิตเพื่อความเร็ว และมีมาตรฐานที่ดีขึ้น ในการพัฒนาแขนกล 5 แกนจะทำให้ทราบถึงหลักการการทำงานของแขนกลอัตโนมัติ

5.1 สรุปผลการทดลอง

จากผลการทดลองที่ได้ จะเห็นได้ว่าในการเคลื่อนที่ของแขนกลอัตโนมัตินั้นจะมีความคลาดเคลื่อนประมาณ 0.695 มิลลิเมตร ซึ่งในงานอุตสาหกรรมนั้นถือว่าเป็นค่าความคลาดเคลื่อนที่มากพอสมควร ซึ่งค่าความคลาดเคลื่อนนี้เกิดจากค่าความคลาดเคลื่อนสะสมในการทำงานของแขนกลในแต่ละแกน และความไม่ละเอียดของโพเทนทิโอมิเตอร์ รวมถึงการบวกระยะมือของหัวช่องสัญญาณ ที่อ่านจาก โพเทนทิโอมิเตอร์ขณะสั่งงาน แต่ปัญหานี้พบครั้งนี้ก็สามารถเป็นแนวทางในการศึกษาเกี่ยวกับการควบคุมแขนกลอัตโนมัติที่มีแม่นยำในการเคลื่อนที่ที่สามารถใช้งานได้จริงในงานอุตสาหกรรม ซึ่งผู้ใดที่สนใจก็สามารถนำปัญหานี้ไปใช้ประกอบการศึกษาและพัฒนาให้แขนกลอัตโนมัติทำงานได้ดียิ่งขึ้นไปอีก

5.2 ข้อจำกัดและปัญหาที่พบของแขนกลอัตโนมัติ

1. พื้นที่การทำงานของแขนกลอัตโนมัติมีน้อย ทำให้การเคลื่อนที่ถูกจำกัด
2. โครงสร้างของแขนกลขณะเคลื่อนที่เกิดการขัดข้อง ทำให้แขนกลเกิดความเสียหาย และทำให้การสิ้นสะท้อนและเกิดเสียงขึ้น
3. ไมโครคอนโทรลเลอร์เมื่อมีการสั่งงานเป็นเวลานานจะเกิดการดำเนินงานที่ผิดพลาดขึ้น

5.3 ข้อเสนอแนะและแนวทางการพัฒนา

ปัญหานี้พบครั้งนี้สามารถสรุปข้อเสนอแนะและแนวทาง ในการพัฒนาโครงการนี้ได้ดังนี้

1. เพิ่มหน้าที่การทำงานของโปรแกรมในการสื่อสารกับผู้ใช้งาน มีระบบช่วยเหลือ (Help) เพื่อให้สะดวกแก่การศึกษาและทำงานด้วยตัวเอง
2. ควรปรับปรุงโครงสร้างของแขนกลอัตโนมัติให้มีความเสถียรภาพในการเคลื่อนที่มากกว่าเดิม
3. ทำให้โปรแกรมสามารถบันทึกตำแหน่งจากการแสดงผลไว้เป็นเอกสารอ้างอิงหรือเก็บเป็นข้อมูลได้ โดยออกแบบโปรแกรมให้สามารถจัดเก็บในรูปแบบข้อมูลได้
4. พัฒนาให้การเคลื่อนที่ของแขนกลอัตโนมัติมีความแม่นยำมากยิ่งขึ้น เช่น เปลี่ยนเซ็นเซอร์ให้มีความละเอียดมากยิ่งขึ้น
5. ปรับปรุงวิธีวัดความคลาดเคลื่อนในการทำงานของแขนกลอัตโนมัติในแต่ละแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา แล 40 ้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- ดอนสัน ปงผาล , ทิววัลย์ คำน่านอง , 2550 . ไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน . พิมพ์ครั้งที่ 1 . กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท.
- นัฐวินท์ ก้ามหวางส์ , ศักดิ์เดช ไตรสุวรรณ และอนุศักดิ์ ชูโลก , 2548 . โครงหุ่นยนต์ 5 แกน . ปรินิพนธ์ ภาควิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.
- เปรมศักดิ์ คุ่มดำไทย , รณน เจียรตระกูล และเมธัส หีบเงิน , 2546 . แบบจำลองแขนกลอัตโนมัติ . ปรินิพนธ์ ภาควิชาวิศวกรรมอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข
โปรแกรมในส่วนอินเตอร์เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <16F877A.h>
#define TXD PIN_C6
#define RXD PIN_C7
#define CLOCK_SP 11059200
#define ADC=10

#fuses HS
#fuses NOLVP, NOWDT
#fuses NOPROTECT
#use delay(clock=CLOCK_SP)
#use rs232(baud=9600, xmit=TXD,rcv=RXD)
#use fast_io(A)
#define FF 0
#define RR 1
#include <stdlib.h>
int8 pwm,Level,Motor,Dir;
int16 Pos[2];
int1 Flag;

void Motor1(int8 Dir)
{
    if(Dir=='0'){output_high(PIN_B0);
        output_low(PIN_B1);
    }else
    {output_high(PIN_B1);
        output_low(PIN_B0);
    }
}

void Motor2(int8 Dir)
{
    if(Dir=='0'){output_high(PIN_B2);
        output_low(PIN_B3);
    }else
    {output_high(PIN_B3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา หรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        output_low(PIN_B2);
    }
}
void Motor3(int8 Dir)
{
    if(Dir=='0'){output_high(PIN_B4);
        output_low(PIN_B5);
    }else
    {output_high(PIN_B5);
        output_low(PIN_B4);
    }
}
void Motor4(int8 Dir)
{
    if(Dir=='0'){output_high(PIN_B6);
        output_low(PIN_B7);
        output_high(PIN_D0);
        output_low(PIN_D1);
    }else
    {output_high(PIN_B7);
        output_low(PIN_B6);
        output_low(PIN_D0);
        output_high(PIN_D1);
    }
}
void Motor5(int8 Dir)
{
    if(Dir=='0'){output_high(PIN_B6);
        output_low(PIN_B7);
        output_low(PIN_D0);
        output_high(PIN_D1);
    }else
    { output_high(PIN_B7);
        output_low(PIN_B6);
        output_high(PIN_D0);
        output_low(PIN_D1);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา ผท.๕.2 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

void MotorOff(void)
{
    output_b(0xFF);
    output_high(PIN_D0);
    output_high(PIN_D1);
}

main()
{signed int16 Value,i,Temp,P;
char Ch;
int8 PFlag;
    enable_interrupts(GLOBAL);
    // enable_interrupts(INT_TIMER0);
    setup_timer_0(RTCC_INTERNAL | RTCC_DIV_8);
    set_timer0(61);
    setup_adc_ports(ALL_ANALOG);
    setup_adc(ADC_CLOCK_DIV_32);
    //set_uart_speed(19200);
    enable_interrupts(INT_RDA);
    i = 0;
    set_tris_b(0x00);
    output_b(0xFF);
    set_tris_d(0xF8);
    output_d(0xFF);
    Level = 0;
    Flag = 0;
    while(1)
    {
        if(Flag==1)
            { //printf("ADC : %ld = %ld \n\r", Value,Temp);
                P = (Pos[0]<<8)+Pos[1];

                if(Motor=='1')
                    { New:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา ผก.๓3 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set_adc_channel(0);
PFlag = 1;
while(PFlag==1)
{ enable_interrupts(INT_TIMER0);
  Value = Read_ADC();
  Temp = abs(P - Value);
  if(Temp < 10)
  {PFlag = 0;
  }else
  {if(P>Value){Motor1('0');}
    else{Motor1('1');}

    // printf("ADC : %ld = %ld \n\r", Value, Temp);
  }
} MotorOff();
//delay_ms(200);
// Value = Read_ADC();
//Temp = abs(P - Value);
//if(Temp > 20){goto New;}
disable_interrupts(INT_TIMER0);
putc('O');putc('K');

}else
// ----- Motor -----//
if(Motor=='2')
{
  New2:
  set_adc_channel(1);
  PFlag = 1;
  while(PFlag==1)
  { enable_interrupts(INT_TIMER0);
    Value = Read_ADC();
    Temp = abs(P - Value);
    if(Temp < 10)
    {PFlag = 0;
    }else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {if(P>Value){Motor2('0');}
          else{Motor2('1');}

// printf("ADC : %ld = %ld \n\r",Value,Temp);
    }
} MotorOff();
// delay_ms(200);
// Value = Read_ADC();
// Temp = abs(P - Value);
// if(Temp > 20){goto New2;}
disable_interrups(INT_TIMER0);
putc('O');putc('K');
}else
if(Motor=='3')
{ New3:
set_adc_channel(2);
PFlag = 1;
while(PFlag==1)
{ enable_interrups(INT_TIMER0);
Value = Read_ADC();
Temp = abs(P - Value);
if(Temp < 10)
{PFlag = 0;
}else
{if(P>Value){Motor3('0');}
else{Motor3('1');}

// printf("ADC : %ld = %ld \n\r",Value,Temp);
}
} MotorOff();
// delay_ms(200);
//Value = Read_ADC();
//Temp = abs(P - Value);
//if(Temp > 20){goto New3;}
disable_interrups(INT_TIMER0);
putc('O');putc('K');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา หรือต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}else
if(Motor=='4')
{New4:
set_adc_channel(3);
PFlag = 1;
while(PFlag==1)
{ enable_interrupts(INT_TIMER0);
Value = Read_ADC();
Temp = abs(P - Value);
if(Temp < 10)
{PFlag = 0;
}else
{if(P>Value){Motor4('0');}
else{Motor4('1');}
// printf("ADC : %ld = %ld \n\r",Valuc,Temp);
}
} MotorOff();
//delay_ms(200);
//Value = Read_ADC();
// Temp = abs(P - Value);
//if(Temp > 20){goto New4;}
disable_interrupts(INT_TIMER0);
putc('O');putc('K');
}else
if(Motor=='5')
{New5:
set_adc_channel(4);
PFlag = 1;
while(PFlag==1)
{ enable_interrupts(INT_TIMER0);
Value = Read_ADC();
Temp = abs(P - Value);
if(Temp < 10)
{PFlag = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา ๕๗-๖๕ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        output_high(PIN_C2);
        output_high(PIN_C3);
        output_high(PIN_C4);
    }

    pwm++;
}

#int_rda
void Serial232(void)
{int8 Ch;
  ch = getc();

  if(Level==0)
  { if(Ch == ':'){Level=1;}
    // putc('1');
  }else

  if(Level==1)
  {if((Ch>='0')&&(Ch<='6'))
    { Motor = Ch;
      Level = 2;
      // putc('2');
    }else Level = 0;
  }else

  if(Level==2)
  {// putc('3');
    if(Ch<4){ Pos[0] = Ch;
              Level = 3;
            }else Level = 0;

  }else

  if(Level==3)
  { Pos[1] = Ch;
    //putc('4');
    Level=4;
  }else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา ผศ.๑-๘ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(Level==4)
{if(Ch==13){Flag = 1;}
//putc('x');
Level=0;
}
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit Arm1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, VaClasses, VaComm, jpeg, ExtCtrls, VrControls, VrWheel,  
Buttons, VrTrackBar;
```

```
type
```

```
TForm1 = class(TForm)
```

```
    VaComm1: TVaComm;
```

```
    Image1: TImage;
```

```
    Timer1: TTimer;
```

```
    Timer2: TTimer;
```

```
    Timer3: TTimer;
```

```
    Timer4: TTimer;
```

```
    Timer5: TTimer;
```

```
    Timer6: TTimer;
```

```
    GroupBox1: TGroupBox;
```

```
    Edit1: TEdit;
```

```
    SpeedButton2: TSpeedButton;
```

```
    Edit2: TEdit;
```

```
    Edit3: TEdit;
```

```
    Edit4: TEdit;
```

```
    SpeedButton3: TSpeedButton;
```

```
    SpeedButton4: TSpeedButton;
```

```
    SpeedButton5: TSpeedButton;
```

```
    SpeedButton1: TSpeedButton;
```

```
    SpeedButton7: TSpeedButton;
```

```
    SpeedButton8: TSpeedButton;
```

```
    SpeedButton9: TSpeedButton;
```

```
    SpeedButton10: TSpeedButton;
```

```
    SpeedButton11: TSpeedButton;
```

```
    CheckBox1: TCheckBox;
```

```
    VrTrackBar1: TVrTrackBar;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VrTrackBar2: TVrTrackBar;
VrTrackBar4: TVrTrackBar;
VrTrackBar3: TVrTrackBar;
VrTrackBar5: TVrTrackBar;
Timer7: TTimer;
Timer8: TTimer;
Timer9: TTimer;
Timer10: TTimer;
Edit5: TEdit;
SpeedButton6: TSpeedButton;
SpeedButton12: TSpeedButton;
Timer11: TTimer;
Timer12: TTimer;
Timer13: TTimer;
GroupBox2: TGroupBox;
ComboBox1: TComboBox;
SpeedButton13: TSpeedButton;
GroupBox3: TGroupBox;
CheckBox6: TCheckBox;
CheckBox2: TCheckBox;
CheckBox3: TCheckBox;
CheckBox4: TCheckBox;
CheckBox5: TCheckBox;
ComboBox2: TComboBox;
CheckBox7: TCheckBox;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);
procedure Button11Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา **พจนานุกรม** และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure VrWheel5Change(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure VrWheel3Change(Sender: TObject);
procedure VrWheel2Change(Sender: TObject);
procedure Timer2Timer(Sender: TObject);
procedure VrWheel1Change(Sender: TObject);
procedure Timer3Timer(Sender: TObject);
procedure VrWheel4Change(Sender: TObject);
procedure Timer4Timer(Sender: TObject);
procedure Timer5Timer(Sender: TObject);
procedure SpeedButton2Click(Sender: TObject);
procedure SpeedButton3Click(Sender: TObject);
procedure SpeedButton4Click(Sender: TObject);
procedure SpeedButton5Click(Sender: TObject);
procedure SpeedButton6Click(Sender: TObject);
procedure SpeedButton8Click(Sender: TObject);
procedure SpeedButton9Click(Sender: TObject);
procedure SpeedButton10Click(Sender: TObject);
procedure SpeedButton11Click(Sender: TObject);
procedure SpeedButton12Click(Sender: TObject);
procedure VrTrackBar5Change(Sender: TObject);
procedure VrTrackBar1Change(Sender: TObject);
procedure VrTrackBar2Change(Sender: TObject);
procedure VrTrackBar3Change(Sender: TObject);
procedure VrTrackBar4Change(Sender: TObject);
procedure VaComm1RxChar(Sender: TObject; Count: Integer);
procedure SpeedButton7Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Timer6Timer(Sender: TObject);
procedure Timer7Timer(Sender: TObject);
procedure Timer8Timer(Sender: TObject);
procedure Timer9Timer(Sender: TObject);
procedure Timer10Timer(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure Timer11Timer(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure Timer13Timer(Sender: TObject);
procedure Timer12Timer(Sender: TObject);
procedure ComboBox1Change(Sender: TObject);
procedure SpeedButton13Click(Sender: TObject);
procedure SpeedButton14Click(Sender: TObject);
procedure Edit2Change(Sender: TObject);

private
  { Private declarations }

public
  { Public declarations }

end;

var
  Form1: TForm1;
  Step,Run : Integer;
  Flag,Rec : Integer;
  Pos : Array[1..5,1..6] of Integer;
  Start : Integer;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
  VaComm1.WriteText(':11'+#13);

end;

procedure TForm1.Button2Click(Sender: TObject);
begin
  VaComm1.WriteText(':10'+#13);

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาข้อ ๔.๕ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TForm1.Button3Click(Sender: TObject);
begin
VaComm1.WriteText(':21'+#13);
end;

procedure TForm1.Button4Click(Sender: TObject);
begin
VaComm1.WriteText(':20'+#13);
end;

procedure TForm1.Button5Click(Sender: TObject);
begin
VaComm1.WriteText(':31'+#13);
end;

procedure TForm1.Button6Click(Sender: TObject);
begin
VaComm1.WriteText(':30'+#13);
end;

procedure TForm1.Button7Click(Sender: TObject);
begin
VaComm1.WriteText(':41'+#13);
end;

procedure TForm1.Button8Click(Sender: TObject);
begin
VaComm1.WriteText(':40'+#13);
end;

procedure TForm1.Button9Click(Sender: TObject);
begin
VaComm1.WriteText(':51'+#13);
end;

procedure TForm1.Button10Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา พ.ร.บ. ๕๖ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
VaComm1.WriteText(':50'+#13);
end;

procedure TForm1.Button11Click(Sender: TObject);
begin
VaComm1.WriteText(':61'+#13);
end;

procedure TForm1.VrWheel5Change(Sender: TObject);
begin
Timer5.Enabled := True;
end;

procedure TForm1.Timer1Timer(Sender: TObject);
Var Value : Integer;
begin
Run := 0;
Timer1.Enabled := False;
Value := (VrTrackBar1.MaxValue - VrTrackBar1.Position)+VrTrackBar1.MinValue;
//Value := VrWheel1.Position;
VaComm1.WriteText(':1'+chr(Value div 256)+chr(Value Mod 256)+#13);
end;

procedure TForm1.VrWheel3Change(Sender: TObject);
begin
Timer3.Enabled := True;
end;

procedure TForm1.VrWheel2Change(Sender: TObject);
begin
Timer2.Enabled := True;
end;

procedure TForm1.Timer2Timer(Sender: TObject);
Var Value : Integer;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
Run := 0;
Timer2.Enabled := False;
Value := (VrTrackBar2.MaxValue - VrTrackBar2.Position)+VrTrackBar2.MinValue;
//Value := VrWheel2.Position;
VaComm1.WriteText(':2'+chr(Value div 256)+chr(Value Mod 256)+#13);
end;

```

```

procedure TForm1.VrWheel1Change(Sender: TObject);

```

```

begin
Timer1.Enabled := True;
end;

```

```

procedure TForm1.Timer3Timer(Sender: TObject);

```

```

Var Value : Integer;
begin
Run := 0;
Timer3.Enabled := False;
Value := (VrTrackBar3.MaxValue - VrTrackBar3.Position)+VrTrackBar3.MinValue;
//Value := VrWheel3.Position;
VaComm1.WriteText(':3'+chr(Value div 256)+chr(Value Mod 256)+#13);
end;

```

```

procedure TForm1.VrWheel4Change(Sender: TObject);

```

```

begin
Timer4.Enabled := True;
end;

```

```

procedure TForm1.Timer4Timer(Sender: TObject);

```

```

Var Value : Integer;
begin
Run := 0;
Timer4.Enabled := False;
Value := (VrTrackBar4.MaxValue - VrTrackBar4.Position)+VrTrackBar4.MinValue;
//Value := VrWheel4.Position;
VaComm1.WriteText(':4'+chr(Value div 256)+chr(Value Mod 256)+#13);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end;

```
procedure TForm1.Timer5Timer(Sender: TObject);
```

```
Var Value : Integer;
```

```
begin
```

```
Run := 0;
```

```
Timer5.Enabled := False;
```

```
Value := VrTrackBar5.Position;
```

```
//Value := VrWheel5.Position;
```

```
VaComm1.WriteText(':5'+chr(Value div 256)+chr(Value Mod 256)+#13);
```

```
end;
```

```
procedure TForm1.SpeedButton2Click(Sender: TObject);
```

```
begin
```

```
Edit1.Text := "";
```

```
end;
```

```
procedure TForm1.SpeedButton3Click(Sender: TObject);
```

```
begin
```

```
Edit2.Text := "";
```

```
end;
```

```
procedure TForm1.SpeedButton4Click(Sender: TObject);
```

```
begin
```

```
Edit3.Text := "";
```

```
end;
```

```
procedure TForm1.SpeedButton5Click(Sender: TObject);
```

```
begin
```

```
Edit4.Text := "";
```

```
end;
```

```
procedure TForm1.SpeedButton6Click(Sender: TObject);
```

```
begin
```

```
Edit5.Text := "";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อ ผพ.-8 จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end;

procedure TForm1.SpeedButton8Click(Sender: TObject);

Var St : String;

begin

St := IntToStr(VrTrackBar1.Position)+'';

St := St +IntToStr(VrTrackBar2.Position)+'';

St := St +IntToStr(VrTrackBar3.Position)+'';

St := St +IntToStr(VrTrackBar4.Position)+'';

St := St +IntToStr(VrTrackBar5.Position)+'';

Pos[1,1] := (VrTrackBar1.MaxValue - VrTrackBar1.Position)+VrTrackBar1.MinValue;

Pos[1,2] := (VrTrackBar2.MaxValue - VrTrackBar2.Position)+VrTrackBar2.MinValue;

Pos[1,3] := (VrTrackBar3.MaxValue - VrTrackBar3.Position)+VrTrackBar3.MinValue;

Pos[1,4] := (VrTrackBar4.MaxValue - VrTrackBar4.Position)+VrTrackBar4.MinValue;

Pos[1,5] := VrTrackBar5.Position;

if(CheckBox1.Checked) Then Begin St := St + '1'; Pos[1,6] := 1; End

Else begin St := St + '0'; Pos[1,6] := 0; End;

Edit1.Text := St;

End;

procedure TForm1.SpeedButton9Click(Sender: TObject);

Var St : String;

begin

St := IntToStr(VrTrackBar1.Position)+'';

St := St +IntToStr(VrTrackBar2.Position)+'';

St := St +IntToStr(VrTrackBar3.Position)+'';

St := St +IntToStr(VrTrackBar4.Position)+'';

St := St +IntToStr(VrTrackBar5.Position)+'';

Pos[2,1] := (VrTrackBar1.MaxValue - VrTrackBar1.Position)+VrTrackBar1.MinValue;

Pos[2,2] := (VrTrackBar2.MaxValue - VrTrackBar2.Position)+VrTrackBar2.MinValue;

Pos[2,3] := (VrTrackBar3.MaxValue - VrTrackBar3.Position)+VrTrackBar3.MinValue;

Pos[2,4] := (VrTrackBar4.MaxValue - VrTrackBar4.Position)+VrTrackBar4.MinValue;

Pos[2,5] := VrTrackBar5.Position;

if(CheckBox1.Checked) Then Begin St := St + '1'; Pos[2,6] := 1; End

Else begin St := St + '0'; Pos[2,6] := 0; End;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา พ.บ. ๙๖ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Edit2.Text := St;
```

```
end;
```

```
procedure TForm1.SpeedButton10Click(Sender: TObject);
```

```
Var St : String;
```

```
begin
```

```
St := IntToStr(VrTrackBar1.Position)+',';
```

```
St := St +IntToStr(VrTrackBar2.Position)+',';
```

```
St := St +IntToStr(VrTrackBar3.Position)+',';
```

```
St := St +IntToStr(VrTrackBar4.Position)+',';
```

```
St := St +IntToStr(VrTrackBar5.Position)+',';
```

```
Pos[3,1] := (VrTrackBar1.MaxValue - VrTrackBar1.Position)+VrTrackBar1.MinValue;
```

```
Pos[3,2] := (VrTrackBar2.MaxValue - VrTrackBar2.Position)+VrTrackBar2.MinValue;
```

```
Pos[3,3] := (VrTrackBar3.MaxValue - VrTrackBar3.Position)+VrTrackBar3.MinValue;
```

```
Pos[3,4] := (VrTrackBar4.MaxValue - VrTrackBar4.Position)+VrTrackBar4.MinValue;
```

```
Pos[3,5] := VrTrackBar5.Position;
```

```
if(CheckBox1.Checked) Then Begin St := St + '1'; Pos[3,6] := 1; End
```

```
Else begin St := St + '0'; Pos[3,6] := 0; End;
```

```
Edit3.Text := St;
```

```
end;
```

```
procedure TForm1.SpeedButton11Click(Sender: TObject);
```

```
Var St : String;
```

```
begin
```

```
St := IntToStr(VrTrackBar1.Position)+',';
```

```
St := St +IntToStr(VrTrackBar2.Position)+',';
```

```
St := St +IntToStr(VrTrackBar3.Position)+',';
```

```
St := St +IntToStr(VrTrackBar4.Position)+',';
```

```
St := St +IntToStr(VrTrackBar5.Position)+',';
```

```
Pos[4,1] := (VrTrackBar1.MaxValue - VrTrackBar1.Position)+VrTrackBar1.MinValue;
```

```
Pos[4,2] := (VrTrackBar2.MaxValue - VrTrackBar2.Position)+VrTrackBar2.MinValue;
```

```
Pos[4,3] := (VrTrackBar3.MaxValue - VrTrackBar3.Position)+VrTrackBar3.MinValue;
```

```
Pos[4,4] := (VrTrackBar4.MaxValue - VrTrackBar4.Position)+VrTrackBar4.MinValue;
```

```
Pos[4,5] := VrTrackBar5.Position;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อ ผข.110 จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(CheckBox1.Checked) Then Begin St := St + '1'; Pos[4,6] := 1; End
```

```
Else begin St := St + '0'; Pos[4,6] := 0; End;
```

```
Edit4.Text := St;
```

```
end;
```

```
procedure TForm1.SpeedButton12Click(Sender: TObject);
```

```
Var St : String;
```

```
begin
```

```
St := IntToStr(VrTrackBar1.Position)+',';
```

```
St := St +IntToStr(VrTrackBar2.Position)+',';
```

```
St := St +IntToStr(VrTrackBar3.Position)+',';
```

```
St := St +IntToStr(VrTrackBar4.Position)+',';
```

```
St := St +IntToStr(VrTrackBar5.Position)+',';
```

```
Pos[5,1] := (VrTrackBar1.MaxValue - VrTrackBar1.Position)+VrTrackBar1.MinValue;
```

```
Pos[5,2] := (VrTrackBar2.MaxValue - VrTrackBar2.Position)+ VrTrackBar2.MinValue;
```

```
Pos[5,3] := (VrTrackBar3.MaxValue - VrTrackBar3.Position)+VrTrackBar3.MinValue;
```

```
Pos[5,4] := (VrTrackBar4.MaxValue - VrTrackBar4.Position)+VrTrackBar4.MinValue;
```

```
Pos[5,5] := VrTrackBar5.Position;
```

```
if(CheckBox1.Checked) Then Begin St := St + '1'; Pos[5,6] := 1; End
```

```
Else begin St := St + '0'; Pos[5,6] := 0; End;
```

```
Edit5.Text := St;
```

```
end;
```

```
procedure TForm1.VrTrackBar5Change(Sender: TObject);
```

```
begin
```

```
Timer5.Enabled := True;
```

```
end;
```

```
procedure TForm1.VrTrackBar1Change(Sender: TObject);
```

```
begin
```

```
Timer1.Enabled := True;
```

```
end;
```

```
procedure TForm1.VrTrackBar2Change(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อ ผท.11 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
Timer2.Enabled := True;
end;

procedure TForm1.VrTrackBar3Change(Sender: TObject);
begin
Timer3.Enabled := True;
end;

procedure TForm1.VrTrackBar4Change(Sender: TObject);
begin
Timer4.Enabled := True;
end;

procedure TForm1.VaComm1RxChar(Sender: TObject; Count: Integer);
Label ToStart;
Var St : String;
begin
if Count >= 2 Then
Begin
St := VaComm1.ReadText;
if (St = 'OK') And (Run = 1) Then
Begin
if Step = 1 Then Timer7.Enabled := True;
if Step = 2 Then Timer9.Enabled := True;
if Step = 3 Then Timer8.Enabled := True;
if Step = 4 Then Timer10.Enabled := True;
if Step = 5 Then Timer12.Enabled := True;
if Step = 6 Then
Begin
if(Start=0) then
Begin
if (Rec = ComboBox2.ItemIndex+1) Then Begin if ComboBox2.ItemIndex = 0 Then Edit1.Color := Clred;
if ComboBox2.ItemIndex = 1 Then Edit2.Color := Clred;
if ComboBox2.ItemIndex = 2 Then Edit3.Color := Clred;
if ComboBox2.ItemIndex = 3 Then Edit4.Color := Clred;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา พ.ร.บ. 12 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ComboBox2.ItemIndex = 4 Then Edit5.Color := Clred;
//if CheckBox7.Checked Then Rec := 0 else
Rec := 6;

```

```
End;
```

```

if (Rec=1) And CheckBox3.Checked Then Begin Edit1.Color := Clred; Rec := 2; End else
if (Rec=2) And CheckBox4.Checked Then Begin Edit2.Color := Clred; Rec := 3; End else
if (Rec=3) And CheckBox5.Checked Then Begin Edit3.Color := Clred; Rec := 4; End else
if (Rec=4) And CheckBox6.Checked Then Begin Edit4.Color := Clred; Rec := 5; End;

```

```

if (Rec=6) Then Timer6.Enabled := False
else Timer6.Enabled := True;

End else Start := 0;
End;
End;
End;
end;

procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
Edit1.Color := ClWindow;
Edit2.Color := ClWindow;
Edit3.Color := ClWindow;
Edit4.Color := ClWindow;
Edit5.Color := ClWindow;
Run := 1;
Timer11.Enabled := True;
end;

```

```

procedure TForm1.FormCreate(Sender: TObject);
begin
Run := 1;
Step := 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อ พ.บ. 13 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Flag := 1;
Rec := 1;
Pos[1,1] := (VrTrackBar1.MaxValue - 520)+VrTrackBar1.MinValue;
Pos[1,2] := (VrTrackBar2.MaxValue - 670)+VrTrackBar2.MinValue;
Pos[1,3] := (VrTrackBar3.MaxValue - 400)+VrTrackBar3.MinValue;
Pos[1,4] := (VrTrackBar4.MaxValue - 620)+VrTrackBar4.MinValue;
Pos[1,5] := 533;
Pos[1,6] := 1;
Start := 1;
//Timer11.Enabled := True;
end;

```

```

procedure TForm1.Timer6Timer(Sender: TObject);
Var Value : Integer;
begin
Step := 1;
//if (Not CheckBox2.Checked) And (Rec=1) Then Rec := 2;
//if (Not CheckBox3.Checked) And (Rec=2) Then Rec := 3;
//if (Not CheckBox4.Checked) And (Rec=2) Then Rec := 4;
//if (Not CheckBox5.Checked) And (Rec=2) Then Rec := 5;
//if (Not CheckBox6.Checked) And (Rec=2) Then Rec := 6;
Timer6.Enabled := False;
Value := Pos[Rcc,1];
VaComm1.WritcText(':'1+chr(Value div 256)+chr(Value Mod 256)+#13);
end;

```

```

procedure TForm1.Timer7Timer(Sender: TObject);
Var Value : Integer;
begin
Step := 2;
Timer7.Enabled := False;
Value := Pos[Rec,2];
VaComm1.WriteText(':'2+chr(Value div 256)+chr(Value Mod 256)+#13);
end;

```

```

procedure TForm1.Timer8Timer(Sender: TObject);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา **พ.บ. 14** ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Var Value : Integer;
begin
Step := 4;
Timer8.Enabled := False;
Value := Pos[Rec,3];
VaComm1.WriteText(':3'+chr(Value div 256)+chr(Value Mod 256)+#13);

end;

```

```

procedure TForm1.Timer9Timer(Sender: TObject);
Var Value : Integer;
begin
Step := 3;
Timer9.Enabled := False;
Value := Pos[Rec,4];
VaComm1.WriteText(':4'+chr(Value div 256)+chr(Value Mod 256)+#13);

end;

```

```

procedure TForm1.Timer10Timer(Sender: TObject);
Var Value : Integer;
begin
Step := 5;
Timer10.Enabled := False;
Value := Pos[Rec,5];
VaComm1.WriteText(':5'+chr(Value div 256)+chr(Value Mod 256)+#13);

end;

```

```

procedure TForm1.SpeedButton1Click(Sender: TObject);
Var i,j : Integer;
begin
for i := 1 to 5 Do
for j := 1 to 6 Do
Pos[i,j] := 0;
Edit1.Text := "";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อ **ผข.** และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Edit2.Text := "";
Edit3.Text := "";
Edit4.Text := "";
Edit5.Text := "";
end;

```

```

procedure TForm1.Timer11Timer(Sender: TObject);

```

```

begin
Timer11.Enabled := False;
Rec := 0;
if CheckBox2.Checked Then Rec := 1 else
if CheckBox3.Checked Then Rec := 2 else
if CheckBox4.Checked Then Rec := 3 else
if CheckBox5.Checked Then Rec := 4 else
if CheckBox6.Checked Then Rec := 5;
if Rec > 0 Then Timer6.Enabled := True;
if Start = 1 Then
Begin
Timer6.Enabled := True;
Rec := 1; // Start := 0;
End;
end;

```

```

procedure TForm1.CheckBox1Click(Sender: TObject);

```

```

begin
Timer13.Enabled := True;
end;

```

```

procedure TForm1.Timer13Timer(Sender: TObject);

```

```

Var St : String;
begin
Run := 0;
Timer13.Enabled := False;
St := '6'+#0;
if CheckBox1.Checked Then St := St+'0'+#13
else St := St+'1'+#13;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา พ.ช. 16 ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VaComm1.WriteText(St);
```

```
end;
```

```
procedure TForm1.Timer12Timer(Sender: TObject);
```

```
Var St : String;
```

```
begin
```

```
Step := 6;
```

```
Timer12.Enabled := False;
```

```
St := IntToStr(Pos[Rec,6]);
```

```
VaComm1.WriteText('6'+#0+St+#13);
```

```
end;
```

```
procedure TForm1.ComboBox1Change(Sender: TObject);
```

```
begin
```

```
VaComm1.PortNum := ComboBox1.ItemIndex + 1;
```

```
end;
```

```
procedure TForm1.SpeedButton13Click(Sender: TObject);
```

```
begin
```

```
if Not VaComm1.Active Then VaComm1.Open;
```

```
Timer11.Enabled := True;
```

```
end;
```

```
procedure TForm1.SpeedButton14Click(Sender: TObject);
```

```
begin
```

```
Run := 1;
```

```
Step := 0;
```

```
Flag := 1;
```

```
Rec := 1;
```

```
Pos[1,1] := (VrTrackBar1.MaxValue - 520)+VrTrackBar1.MinValue;
```

```
Pos[1,2] := (VrTrackBar2.MaxValue - 670)+VrTrackBar2.MinValue;
```

```
Pos[1,3] := (VrTrackBar3.MaxValue - 400)+VrTrackBar3.MinValue;
```

```
Pos[1,4] := (VrTrackBar4.MaxValue - 620)+VrTrackBar4.MinValue;
```

```
Pos[1,5] := 533;
```

```
Pos[1,6] := 1;
```

```
Start := 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Timer11.Enabled := True;
```

```
end;
```

```
procedure TForm1.Edit2Change(Sender: TObject);
```

```
begin
```

```
end;
```

```
end.
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา ๒๕๖๔ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้