ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

เครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ

AUTOMATIC CUTTING MACHINE WITH IMAGE INPUT

นายประทีป กันทา รหัส 41360355
นางสาวหทัยรัตน์ ธรรมาธิวัฒน์ รหัส 41360538

i 5094124 e.2
ปร
ป277ค.
2544

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2544

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 30 ม.ค. 2544
เลขทะเบียน วศ. 4400613
เลขเรียกหนังสือ TJ 1165 ป277ค
มหาวิทยาลัยนเรศวร

ใบรับรองโครงงานวิจัย

หัวข้อโครงงาน	เครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ		
	AUTOMATIC CUTTING MACHINE WITH IMAGE INPUT		
ผู้ดำเนินโครงงาน	นายประทีป	กันทา	รหัส 41360355
	นางสาวหทัยรัตน์	ธรรมาธิวัฒน์	รหัส 41360538
อาจารย์ที่ปรึกษา	อาจารย์ยงยุทธ	ชนบดีเฉลิมรุ่ง	
อาจารย์ที่ปรึกษาร่วม	อาจารย์สิทธิโชค	เชาวกุล	
สาขา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2544		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิจัย

.....ประธานกรรมการ
(อาจารย์สิทธิโชค เชาวกุล)

.....กรรมการ
(อาจารย์ประทีป ตรีรณโอภาส)

.....กรรมการ
(อาจารย์วัชร์วีร์ พืชพันธ์)

.....กรรมการ
(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงงาน	**เครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ**		
	AUTOMATIC CUTTING MACHINE WITH IMAGE INPUT		
ผู้ดำเนินโครงงาน	**นายประทีป**	**กันทา**	**รหัส 41360355**
	นางสาวหทัยรัตน์	**ธรรมาธิวัฒน์**	**รหัส 41360538**
อาจารย์ที่ปรึกษา	**อาจารย์ยงยุทธ**	**ชนบดีเฉลิมรุ่ง**	
อาจารย์ที่ปรึกษาร่วม	**อาจารย์สิทธิโชค**	**เชาวกุล**	
สาขา	**วิศวกรรมคอมพิวเตอร์**		
ภาควิชา	**วิศวกรรมไฟฟ้าและคอมพิวเตอร์**		
ปีการศึกษา	2544		

.....

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ เพื่อที่จะนำระบบคอมพิวเตอร์มาควบคุมการทำงานของเครื่องตัดโลหะด้วยแก๊ส ซึ่งจะใช้ไมโครคอนโทรเลอร์ (Microcontroller) ควบคุมการทำงานของสเต็ปเปอร์มอเตอร์ (Stepper Motor) ในการบังคับทิศทางการเคลื่อนที่ของเครื่องตัดโลหะให้เป็นไปตามภาพต้นแบบที่ถูกกำหนดโดยผู้ใช้งาน

ผลที่ได้จากการทำโครงงานนี้ คือ ได้เครื่องตัดโลหะอัตโนมัติด้วยแก๊สที่สามารถตัดโลหะได้ตามภาพต้นแบบ

Project Title	Automatic Cutting Machine with Image Input		
Name	Mr. Prateap	Gunta	ID. 41360355
	Miss Hatairat	Thamathivath	ID. 41360538
Project Advisor	Mr. Yongyut	Chonbodeechacernroong	
Co- Project Advisor	Mr. Sitthichok	Choawakul	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2001		

.....

ABSTRACT

This project is to study and develope about Automatic Cutting Machine with Image Input by using a computer system to control the gas – cutting machine. We use microcontroller to control stepper motor to direct the cutting machine to move according to the original picture fixed by user.

Finally, from this project we will have Automatic Cutting Machine by gas to cut metal according to the original picture.

กิตติกรรมประกาศ

การจัดทำโครงงานวิศวกรรมศาสตร์ครั้งนี้สำเร็จลุล่วงด้วยดี เนื่องจากได้รับคำแนะนำและความช่วยเหลือจากอาจารย์ยงยุทธ ชนบดีเฉลิมรุ่ง และอาจารย์สิทธิโชค เชาวกุล ที่ได้ให้คำปรึกษาชี้แนะแนวทางที่เป็นประโยชน์ในการทำโครงงานครั้งนี้และขอขอบใจเพื่อน ๆ สำหรับความช่วยเหลือในด้านต่าง ๆ

นายประทีป กันทา

นางสาวหทัยรัตน์ ธรรมาธิวัฒน์

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก
บทคัดย่อภาษาอังกฤษ ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ช

บทที่ 1 บทนำ
1.1 หลักการและเหตุผลของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบข่ายของโครงงาน 2
1.4 กิจกรรมการดำเนินงาน 2
1.5 ผลที่คาดว่าจะได้รับ 3
1.6 งบประมาณ 4

บทที่ 2 ทฤษฎีของเครื่องมือที่ใช้ทำเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ
2.1 สเต็ปเปอร์มอเตอร์ 5
2.2 มาตรฐาน RS-232 10
2.3 ET-EM PLUS 12
2.4 CP-SB31 14

บทที่ 3 การออกแบบเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ
3.1 โครงสร้างภาพรวมการทำงานของเครื่องตัดโลหะ 15
3.2 ศึกษาและออกแบบโครงสร้าง 15
3.3 ส่วนของโปรแกรม 16
3.4 การเลือกสเต็ปเปอร์มอเตอร์ 17
3.5 ส่วนของโปรแกรม 19
3.6 การใช้งานเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ 23

สารบัญ (ต่อ)

บทที่ 4 ผลการทดลองและการวิเคราะห์

4.1 การทดลองบอร์ดไดเวอร์กับโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์ 27

4.2 การทดลองในส่วนของโปรแกรมวาดภาพ 28

4.3 การทดลองการรับส่งข้อมูลระหว่าง Visual Basic กับ MCS-51 29

4.4 การทดลองฟังก์ชั่นต่าง ๆ ระหว่างโครงสร้างกับ MCS-51 29

บทที่ 5 บทสรุป

5.1 สรุปและวิจารณ์ผลการทดลอง 30

5.2 ปัญหาและวิธีการแก้ปัญหา 31

5.3 ข้อเสนอแนะ 31

เอกสารอ้างอิง 32

ภาคผนวก 33

ประวัติผู้เขียน 46

สารบัญตาราง

ตารางที่ หน้า

1.1 กิจกรรมการดำเนินงานด้านการศึกษาและค้นคว้าข้อมูล 2

1.2 กิจกรรมการดำเนินงานด้านปฏิบัติงานโครงงาน 3

2.1 ขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบเวฟ 8

2.2 ขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบ 2 เวฟ 9

2.3 ขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบครึ่งสเต็ป 10

2.4 รายละเอียดฟังก์ชันการทำงานของคอนเน็กเตอร์ของพอร์ต RS-232 แบบ 9 ขา 11

มหาวิทยาลัยนเรศวร

สารบัญรูป

รูปที่		หน้า
2.1	โครงสร้างภายในสเต็ปเปอร์มอเตอร์	5
2.2	การควบคุมการทำงานของสเต็ปเปอร์มอเตอร์	6
2.3	การจัดการขาคอนเน็กเตอร์ของพอร์ต RS-232 แบบ 9 ขา	12
3.1	โครงสร้างภาพรวมของเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ	15
3.2	ภาพ Isometric View ของส่วนโครงสร้าง	16
3.3	วงจรควบคุมสเต็ปเปอร์มอเตอร์	18
3.4	หน้าจอของโปรแกรม	19
3.5	โฟลว์ชาร์ตโปรแกรมหลักของโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์	20
3.6	โฟลว์ชาร์ตโปรแกรมย่อยหมุนทวนเข็มนาฬิกาแบบเวฟ	21
3.7	โฟลว์ชาร์ตโปรแกรมย่อยหมุนตามเข็มนาฬิกาแบบเวฟ	22
3.8	หน้าจอของโปรแกรมก่อนที่จะมีการทำงาน	23
3.9	หน้าจอของโปรแกรม GCODE	24
3.10	หน้าจอที่มีการเรียกใช้โปรแกรม GCODE	25
3.11	หน้าจอที่ได้มีการ Send_Data แล้ว	26

บทที่ 1
บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันพบว่าคอมพิวเตอร์ได้เข้ามามีบทบาทต่อชีวิตประจำวันของมนุษย์อย่างมาก เพราะคอมพิวเตอร์สามารถช่วยอำนวยความสะดวกในการทำงานของมนุษย์ในหลาย ๆ ด้าน โดยเฉพาะอย่างยิ่งงานทางด้านอุตสาหกรรมต่าง ๆ ซึ่งจะมีการใช้เครื่องจักรกลหลายชนิดมาช่วยในการทำงาน เครื่องจักรกลบางชนิดสามารถนำระบบคอมพิวเตอร์และไมโครคอนโทรลเลอร์ (Microcontroller) เข้ามาช่วยควบคุมการทำงานให้เป็นไปอย่างอัตโนมัติ ซึ่งจะช่วยให้การทำงานเป็นไปได้อย่างสะดวกและรวดเร็วยิ่งขึ้น การทำงานของเครื่องจักรกลบางชนิดอาจก่อให้เกิดอันตรายกับมนุษย์ได้ ดังนั้นการนำระบบคอมพิวเตอร์และไมโครค่อนโทรลเลอร์มาช่วยควบคุมการทำงานของเครื่องจักรกลแทนการทำงานโดยมนุษย์ สามารถช่วยลดอันตรายการเกิดอุบัติเหตุในการทำงานได้อีกประการหนึ่ง

โครงงานนี้มีจุดมุ่งหมายที่จะนำระบบคอมพิวเตอร์มาควบคุมการทำงานของเครื่องตัดโลหะด้วยแก๊ส ซึ่งจะใช้ไมโครคอนโทรลเลอร์ควบคุมการทำงานของสเต็ปเปอร์มอเตอร์ (Stepper Motor) ในการบังคับ ทิศทางการเคลื่อนที่ของเครื่องตัดโลหะให้เป็นไปตามภาพต้นแบบที่ถูกกำหนดโดยผู้ใช้งาน

1.2 วัตถุประสงค์

1. เพื่อให้เข้าใจถึงหลักการและวิธีการนำ AutoCAD มาประยุกต์ใช้กับโปรแกรม GCODE มาช่วยในการเขียนโปรแกรมเพื่อสั่งให้ ไมโครคอนโทรลเลอร์ ไปควบคุมหัวเป่าแก๊ส
2. เพื่อให้มีความรู้ความเข้าใจในหลักการทำงานของไมโครคอนโทรลเลอร์ที่จะใช้ควบคุมการทำงานของเครื่องตัดโลหะด้วยแก๊ส โดยใช้การสื่อสารข้อมูลอนุกรม (Serial port) ซึ่งจะต้องใช้หลักการของการอินเทอร์เฟสเข้ามาร่วมด้วย
3. เพื่อให้เข้าใจหลักการการทำงานของสเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์ 6 สาย 4 เฟส ในการควบคุมทิศทางและการเคลื่อนที่ของเครื่องตัดโลหะด้วยแก๊ส
4. เพื่อให้มีความรู้และความเข้าใจในหลักการของการอินเทอร์เฟส ให้มีการทำงานสื่อสารสัมพันธ์กันไปตลอด
5. เพื่อพัฒนาระบบอุตสาหกรรมโดยการให้ระบบคอมพิวเตอร์เข้ามามีบทบาทในการควบคุมอุปกรณ์ที่ทำหน้าที่ผลิตสินค้าหรือผลิตชิ้นงานในโรงงานอุตสาหกรรม

1.3 ขอบข่ายการทำงาน

โครงงานเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบนี้ จะมีการนำภาพต้นแบบที่เป็นภาพลายเส้นรูปทรงเรขาคณิตมาทำการวิเคราะห์ภาพด้วยคอมพิวเตอร์แล้วนำมาอินเทอร์เฟสกับไมโครคอนโทรลเลอร์ ซึ่งไมโครคอนโทรลเลอร์จะทำการควบคุมการทำงานของสเต็ปเปอร์มอเตอร์ให้เครื่องตัดโลหะสามารถตัดให้ได้ชิ้นงานที่มีรูปร่างลักษณะคล้ายกับภาพต้นแบบที่นำมาใช้งาน

1.4 กิจกรรมการดำเนินงาน

1. ศึกษาและค้นคว้าข้อมูลเพื่อใช้ในการทำโครงงาน

ตารางที่ 1.1 กิจกรรมการดำเนินงานด้านการศึกษาและค้นคว้าข้อมูล

กิจกรรม	เดือน – ปี				
	ต.ค. 43	พ.ย. 43	ธ.ค. 43	ม.ค. 44	ก.พ. 44
ศึกษาโปรแกรม AutoCAD	←	—	—	→	
ศึกษาโปรแกรม Gcode	←	—	—	→	
ศึกษาไมโคร-คอนโทรลเลอร์		←	—	—	→
ศึกษาการทำงานของสเต็ปเปอร์มอเตอร์			←	—	→
การส่งสัญญาณ			←	—	→
ศึกษาภาษา C		←	—	—	→
ศึกษาโปรแกรม Visual Basic				←	→

2. แผนปฏิบัติงานโครงงาน

ตารางที่ 1.2 กิจกรรมการดำเนินงานด้านปฏิบัติงานโครงงาน

กิจกรรม	เดือน – ปี 2544						
	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
ออกแบบโครงสร้างเครื่องตัดโลหะ	←	→					
สร้างโครงสร้าง		←	→				
ทำการทดลองใช้งานเครื่องตัดโลหะ			←	→			
เก็บข้อมูลที่ได้จากการทดลอง				←	→		
วิเคราะห์ข้อมูลจากการทดลอง					←	→	
สรุปผลของการทดลองเครื่องตัดโลหะ						←	→
จัดทำรูปเล่ม							← →

1.5 ผลที่คาดว่าจะได้รับ

1. มีความเข้าใจถึงหลักการและวิธีการวิเคราะห์ภาพของคอมพิวเตอร์
2. มีความรู้ความเข้าใจในหลักการการทำงานของไมโครคอนโทรลเลอร์
3. มีความเข้าใจในหลักการทำงานและการควบคุมของสเต็ปเปอร์มอเตอร์
4. มีความรู้ความเข้าใจในหลักการของการอินเทอร์เฟส
5. มีความรู้ความเข้าใจในการทำงานร่วมกันของส่วนต่าง ๆ ของโครงงานนี้
6. มีความสามารถนำคอมพิวเตอร์มาช่วยงานอุตสาหกรรม

1.6 งบประมาณ 2,000 บาท เป็นค่าวัสดุและอุปกรณ์ โดยขออนุมัติถัวเฉลี่ยทุกรายการ

บอร์ด ET – SMCC	1,600	บาท
บอร์ด CP – SB31	800	บาท

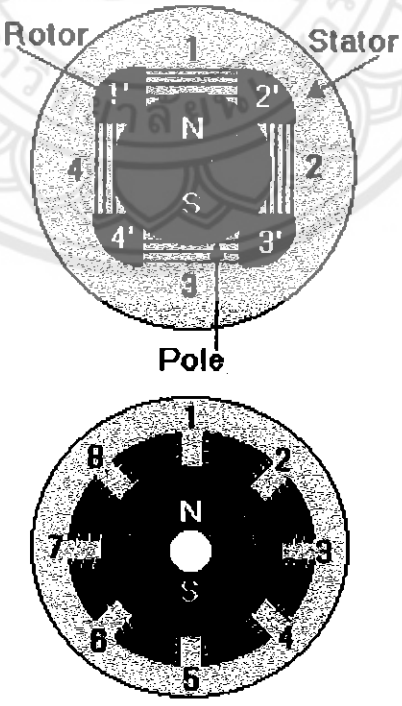
บทที่ 2
ทฤษฎีของเครื่องมือที่ใช้ทำเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ

2.1 สเต็ปเปอร์มอเตอร์[1]

2.1.1 การทำงานของสเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์ เป็นมอเตอร์ที่มีลักษณะเมื่อมีการป้อนไฟฟ้าให้กับมอเตอร์ทำให้หมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งต่างจากมอเตอร์ทั่วไปที่จะหมุนทันทีและตลอดเวลาเมื่อป้อนแรงดันไฟฟ้า ข้อดีของสเต็ปเปอร์มอเตอร์ คือ สามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลข (องศาหรือระยะทาง) ได้อย่างละเอียดโดยใช้คอมพิวเตอร์หรือไมโครคอนโทรลเลอร์เป็นเครื่องกำหนดและจัดเก็บตัวเลข

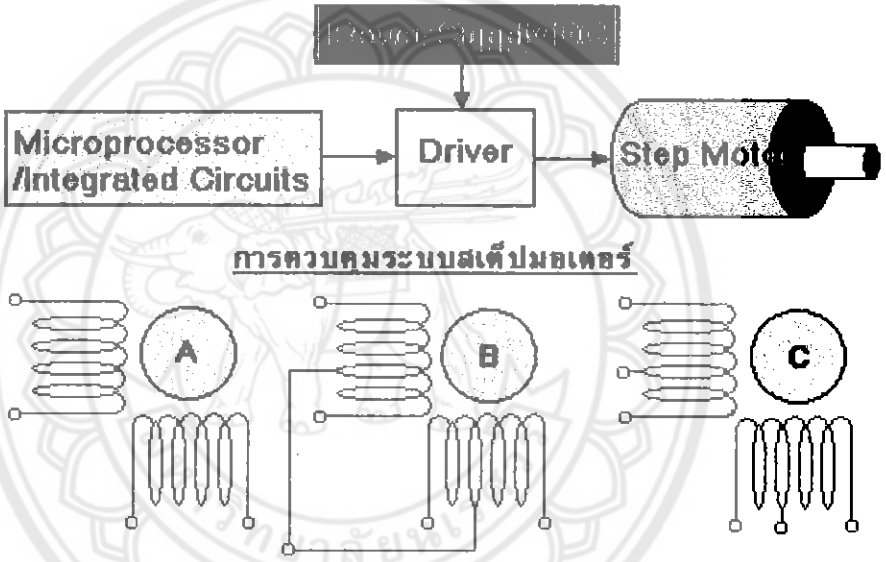
โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ทำมาจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมาประกอบกันเป็นชั้นๆ โดยที่แต่ละซี่นั้นจะมีคอยล์ (ขดลวด) พันสวมอยู่ เมื่อมีการป้อนกระแสผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic) ดังรูป 2.1 การแสดงถึงองค์ประกอบของสเต็ปเปอร์มอเตอร์



รูปที่ 2.1 โครงสร้างภายในสเต็ปเปอร์มอเตอร์

ลักษณะการนำไปใช้งาน สเต็ปเปอร์มอเตอร์ใช้งานลักษณะระบบเปิด (Open Loop System) คือ สเต็ปเปอร์มอเตอร์สามารถทำงานได้โดยไม่ต้องมีการป้อนค่าพารามิเตอร์กลับมา (Feed back) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งที่แน่นอนนั้น จะต้องทำการป้อนกลับไปยังระบบและตัวบอกตำแหน่งว่าถูกต้องหรือผิดผลาดให้รับทราบ

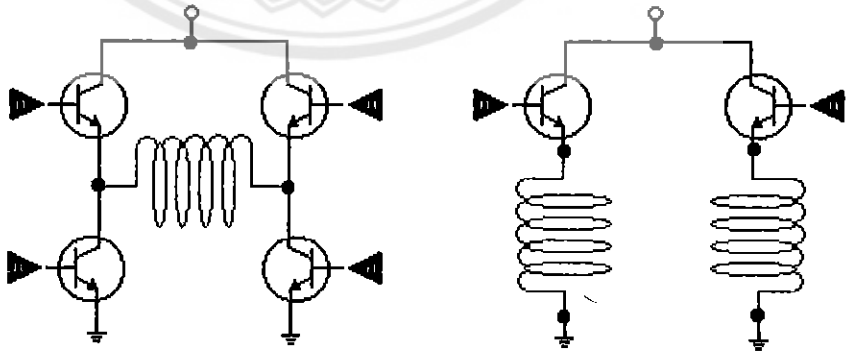
วิธีที่ใช้กับสเต็ปเปอร์มอเตอร์ คือ เรานำลิมิทสวิตช์ติดตามตำแหน่งที่จะตรวจจับ (limit switch) เมื่อสเต็ปเปอร์มอเตอร์เริ่มหมุนแล้วหมุนไปจนถึงตำแหน่งของสวิตช์ตรวจจับสัญญาณ สวิตช์ทำงานก็จะป้อนกลับไปสู่ระบบ ซึ่งก็จะทำให้รู้การทำงานของสเต็ปเปอร์มอเตอร์ตลอดจนตัววงจรไมโครคอนโทรลเลอร์เองจะมีจุดอ้างอิง (reference point) ไว้ให้เริ่มต้นการทำงานและอ้างอิงตำแหน่งได้อย่างถูกต้อง



การควบคุมระบบสเต็ปมอเตอร์

A) แบบไบโพล่าร์ B)แบบยูนิโพลาร์ 5 สาย C)แบบยูนิโพลาร์ชนิด 6 สาย

การพันขดลวดบนสเตเตอร์ของสเต็ปมอเตอร์



A) แบบไบโพลาร์ B) แบบยูนิโพลาร์

คือ ต่อเข้ากับแหล่งจ่ายไฟหรือจากพอร์ตพีซีเพื่อทริกทรานซิสเตอร์ให้ทำงาน

วงจรการจ่ายไฟให้กับสเต็ปมอเตอร์

รูปที่ 2.2 การควบคุมการทำงานของสเต็ปเปอร์มอเตอร์

โดยแนวทางของสเต็ปเปอร์มอเตอร์เป็นอุปกรณ์จำพวกเชิงกลทางไฟฟ้าโดยมีกรุ๊ปของไบนารีโวลท์เตทเป็นอินพุตและการเคลื่อนที่แบบเชิงมุมเป็นเอาท์พุต หรือว่าหมุนทีละสเต็ปซึ่งอยู่ระหว่าง 0.1-30 องศา อยู่ที่โครงสร้างของสเต็ปเปอร์มอเตอร์โดยตามสัญญาณพัลซ์ที่จ่ายให้กับขดสเตเตอร์ทำให้เกิดแรงผลักแก่โรเตอร์หมุนไป สเต็ปเปอร์มอเตอร์มีขดลวดหลายชุด เรียกว่า เฟส (Phase) ดังนั้นสัญญาณที่ต่อเนื่องเป็นซีเควน (Sequence) ลักษณะของไบนารี่ (Binary) ซึ่งจะต้องไปผ่านวงจรไดร์เวอร์ (driver) ก็จะทำให้โรเตอร์หมุนไปอย่างต่อเนื่อง ดังรูปที่ 2.2 การควบคุมการทำงานของสเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์โดยทั่วไปมีจำนวนของขั้วแม่เหล็กหรือจำนวนสเต็ปต่อรอบเป็นจำนวนมาก ปกติอยู่ที่ประมาณ 100—400 สเต็ปต่อรอบ การมีจำนวนสเต็ปมาก ๆ นี้ไม่ได้เพิ่มที่จำนวนขั้วแม่เหล็กไฟฟ้าที่สเตเตอร์ แต่ทำได้โดยเพิ่มจำนวนซี่ขั้วแม่เหล็กที่สเตเตอร์ จำนวนสเต็ปต่อรอบทั้งหมดจะได้จากการคูณจำนวนขั้วแม่เหล็กบนสเตเตอร์และจำนวนซี่ที่โรเตอร์ ดังเช่น ถ้ามีขั้วแม่เหล็ก 3 ขั้วบนสเตเตอร์และ 8 ซี่ขั้วแม่เหล็กบนโรเตอร์ สเต็ปเปอร์มอเตอร์ตัวนี้จะทำงานที่ 24 สเต็ปต่อรอบ หรือหมุนเป็นมุม 15 องศาต่อสเต็ป

การใช้วงจรดิจิตอลคอนโทรลเลอร์ กำหนดการจ่ายกำลังไฟฟ้าเข้าสู่ขดลวดบนสเตเตอร์แบบซีเควนเชียลทำให้สามารถควบคุมการเคลื่อนที่ทุกสเต็ปได้ แต่การควบคุมด้วยดิจิตอลไม่จำเป็นต้องมีการป้อนกลับ การเคลื่อนที่ทุกสเต็ปได้จากการคำนวณจำนวนรอบหรือมุมในการหมุนที่ต้องการ แล้วจึงส่งข้อมูลที่ได้ไปควบคุมการหมุนของสเต็ปเปอร์มอเตอร์ พิกัดในการทำงาน อาทิ ความเร็ว มุมในการเคลื่อนที่ ตำแหน่งของเพลาถูกกำหนดจากข้อมูลที่ส่งมาควบคุม

2.1.2 ประเภทของสเต็ปเปอร์มอเตอร์

การพันขดลวดบนสเตเตอร์ของสเต็ปเปอร์มอเตอร์จะมีการพันด้วยกัน 2 วิธี คือ แบบไบโพลาร์ (Bipolar)กับแบบยูนิโพลาร์ (Unipolar)

2.1.2.1 แบบไบโพลาร์

จะมีการพันขดลวดหนึ่งขด(จะกี่รอบก็แล้วแต่สเป็กใช้งาน)ในแต่ละขั้วแม่เหล็กของสแตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้นที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้าซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้เพียงการกลับทิศทางของการไหลในกระแสไฟฟ้าโดยมาจากการควบคุมของวงจรสวิทชิ่งให้กลับขั้วไฟฟ้า

2.1.2.2 แบบยูนิโพลาร์

แบบยูนิโพลาร์นี้จะมีการพันขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ซึ่งจะทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามเช่นกัน การกลับทิศทางขั้วแม่เหล็กเปลี่ยนไปมาทำได้โดยใช้วงจรสวิทชิงให้สลับขั้วหนึ่งไปยังอีกขั้วหนึ่งแทนกัน โดยปกติขดลวดทั้งสองจะมีการเชื่อมต่อกันหรือมีจุดร่วมเพื่อลดจำนวนของสายไฟที่ต่อจากสเต็ปเปอร์มอเตอร์ วงจรจ่ายกำลังไฟฟ้าของมอเตอร์

แบบยูนิโพลาร์ทำได้ง่ายกว่าชนิดไบโพลาร์ เพราะมันต้องการเพียงสวิตช์ธรรมดาในการเปิดและปิดกำลังไฟฟ้าให้กับขดลวดบนสเตเตอร์ในทิศทางที่ต้องการให้หมุนได้ทันทีของสเต็ปเปอร์มอเตอร์

พื้นฐานการสวิทชิ่งดูรูปที่ 2.2 ชื่อวงจรการจ่ายไฟให้กับสเต็ปเปอร์มอเตอร์ การพันขดลวดทั้ง 2 แบบที่กล่าวมาข้างต้นแตกต่างกันที่ แบบยูนิโพล่าร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพลาร์และสายไฟที่ต่อมาจากตัวสเต็ปเปอร์มอเตอร์ซึ่งแบบไบโพลาร์จะมี 4 สาย ส่วนเป็นแบบยูนิโพลาร์จะมี 5 สาย หรือ 6 สาย

2.1.3 การสั่งงานควบคุมการหมุนของสเต็ปเปอร์มอเตอร์

การควบคุมและสั่งงานให้สเต็ปเปอร์มอเตอร์ทำงานไปทีละสเต็ปสามารถทำได้โดยการจ่ายกำลังไฟไปยังขดลวดในแต่ละขดบนสเตเตอร์ โดยการป้อนจะทำในลักษณะเป็นลำดับ หรือที่เรียกว่าซีเควนเวียล ในลูปที่ถูกต้อง ซึ่งจะแบ่งออกได้เป็น 3 รูปแบบ คือ แบบเวฟ (wave) แบบ 2 เฟส (2 phase) และแบบครึ่งสเต็ป (half step) ทั้ง 3 แบบนี้ก็จะมีข้อดีและข้อเสียต่างกันออกไป

2.1.3.1 แบบเวฟ (wave)

จะเป็นการกระตุ้นแบบที่ง่ายที่สุด ซึ่งจะทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งๆเรียงกันไป ตัวอย่างเช่น ขดที่ 1 , 2 , 3 , 4 , 1 , 2 , 3 , 4 เป็นลำดับอย่างนี้ หรือ ขด 1 , 4 , 3 , 2 , 1 , 4 , 3 , 2 เป็นลำดับกันไป ดังนั้นจึงมีขดลวดเพียงขดเดียวในเวลาหนึ่งที่ถูกกระตุ้นเท่านั้น ทั้งนี้ขึ้นอยู่กับทิศทางที่ต้องให้มอเตอร์หมุนไป วงจรที่นำมากระตุ้นนั้นจะมีราคาค่อนข้างจะถูกกว่าและง่ายกว่าดังในรูปที่2.2 ของวงจรการจ่ายไฟ ซึ่งสามารถเขียนขั้นตอนการทำงานเป็นตารางที่ 2.1 ออกมาได้ดังนี้

ตารางที่ 2.1 ขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบเวฟ

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON			
2		ON		
3			ON	
4				ON
5	ON			
6		ON		

2.1.3.2 แบบ 2 เฟส (2 Phase)

แบบนี้ก็จะคล้ายกับการกระตุ้นในแบบเวฟแต่จะต่างกันตรงที่ แบบ 2 เฟส จะทำการกระตุ้นโดยการจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขด ที่อยู่ใกล้กัน ในเวลาเดียวกันและจะเรียงลำดับกันไป ดังเช่นแบบเดียวกับแบบเวฟ ตัวอย่างการกระตุ้นขดลวดในลักษณะซีเควน ดังนี้ 12 , 23 , 34 , 41 , 12 , 23 , 34 , 41 เรียงลำดับกันไปเรื่อยๆ หรือจะเป็นแบบ14 , 43 , 32 , 21 , 14 , 43 , 32 , 21 เรียงกันไปเรื่อยๆเช่นกันถ้าจะมากล่าวถึงข้อดีข้อเสียของแบบ 2 เฟสแล้วมีดังนี้

ข้อดี การที่จะเพิ่มจำนวนขดลวดที่ถูกกระตุ้นจะทำให้แรงบิดได้มากกว่าแบบเวฟ ซึ่งโรเตอร์จะหมุนด้วยแรง ดึงแบบเต็ม ๆ แรงจากทั้ง 2 ขดลวดที่กระตุ้นพร้อมกัน

ข้อเสีย แบบ 2 เฟส จะกระตุ้นขดลวดนั้นต้องใช้กำลังไฟมากขึ้นเป็น 2 เท่าของแบบเวฟ สามารถเขียนลำดับการกระตุ้นของขดลวดแบบ 2 เฟส ได้ดังในตารางที่ 2.2

ตารางที่ 2.2 ขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบ 2 เฟส

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	ON		
2		ON	ON	
3			ON	ON
4	ON			ON
5	ON	ON		
6		ON	ON	

2.1.3.3 แบบครึ่งสเต็ป

แบบครึ่งสเต็ปนี้เป็นรูปแบบที่เกิดจากการผสมผสานของการกระตุ้นระหว่าง แบบเวฟ กับ แบบ 2 เฟส เพื่อให้จำนวนรอบของสเต็ปให้มากขึ้นเป็น 2 เท่า ซึ่งในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเรื่อยๆเป็นลำดับ ตัวอย่างต่อไปนี้ 1 , 12 , 2 , 23 , 3 , 34 , 4 , 41 , 1 , 12 , 2 , 23 , 3 , 34 , 4 , 41 , 1 เป็นลำดับอยู่อย่างนี้เรื่อยไป ถ้าจะกลับทิศทางการหมุนก็จะได้เป็นดังนี้ 1 , 41 , 4 , 43 , 3 , 32 , 2 , 21 , 1 , 41 , 4 , 43 , 3 , 32 , 2 , 21 , 1 เป็นลำดับกัน

ข้อดี การกระตุ้นแบบนี้จะให้แรงบิดที่เพิ่มมากขึ้นเนื่องจากช่วงสเต็ปที่มีระยะสั้นลง อีกประการหนึ่ง แต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน เป็นผลให้ค่าตำแหน่งความถูกต้องมากขึ้นไปด้วย

ข้อเสีย ก็คงเช่นเดียวกับแบบ 2 เฟส ที่ต้องจ่ายกำลังไฟเป็น 2 เท่าของแบบเวฟหรือจะใช้เท่ากับแบบ 2 เฟส

ดังนั้น สามารถนำลำดับการทำงานของแบบครึ่งเฟส ในรูปของตารางที่ 2.3

ตารางที่ 2.3 ขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบครึ่งสเต็ป

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON			
2	ON	ON		
3		ON		
4		ON	ON	
5			ON	
6			ON	ON
7				ON
8	ON			ON
9	ON			
10	ON	ON		

2.2 มาตรฐาน RS-232

การที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกัน สามารถติดต่อสื่อสารกันแบบอะซิงโครนัสได้นั้นจะต้องมีอุปกรณ์ที่เป็นขั้วต่อมาตรฐาน ซึ่งมีมากมายหลายชนิดด้วยกัน โดยมาตรฐานที่ใช้กันอย่างกว้างขวาง คือ มาตรฐาน RS-232 ซึ่งกำหนดลักษณะของสัญญาณทางไฟฟ้าที่ถูกใช้ในการเชื่อมต่อแบบอนุกรมโดยตรง มีเพียง 2 ลักษณะ คือ ไบนารี 0 หรือแรงดันไฟฟ้าบวก และไบนารี 1 หรือแรงดันไฟฟ้าลบ

การอินเตอร์เฟส RS-232 มีอยู่ 2 แบบด้วยกัน คือ โดยพื้นฐานแล้วหากเป็นส่วนที่ใช้ต่อเข้ากับอุปกรณ์ประมวลผลจะถูกเรียกว่า DTE (Data Terminal Equipment) และส่วนที่ติดต่อกับอุปกรณ์สื่อสานจะเรียกว่า DCE (Data Circuitterminating Equipment) โดยทั่วไปจะใช้ขั้วต่อ BD25 ตัวผู้สำหรับต่อกับ DTE ส่วน DCE ใช้ขั้วต่อ BD25 ตัวเมีย

การทำงานของ RS-232

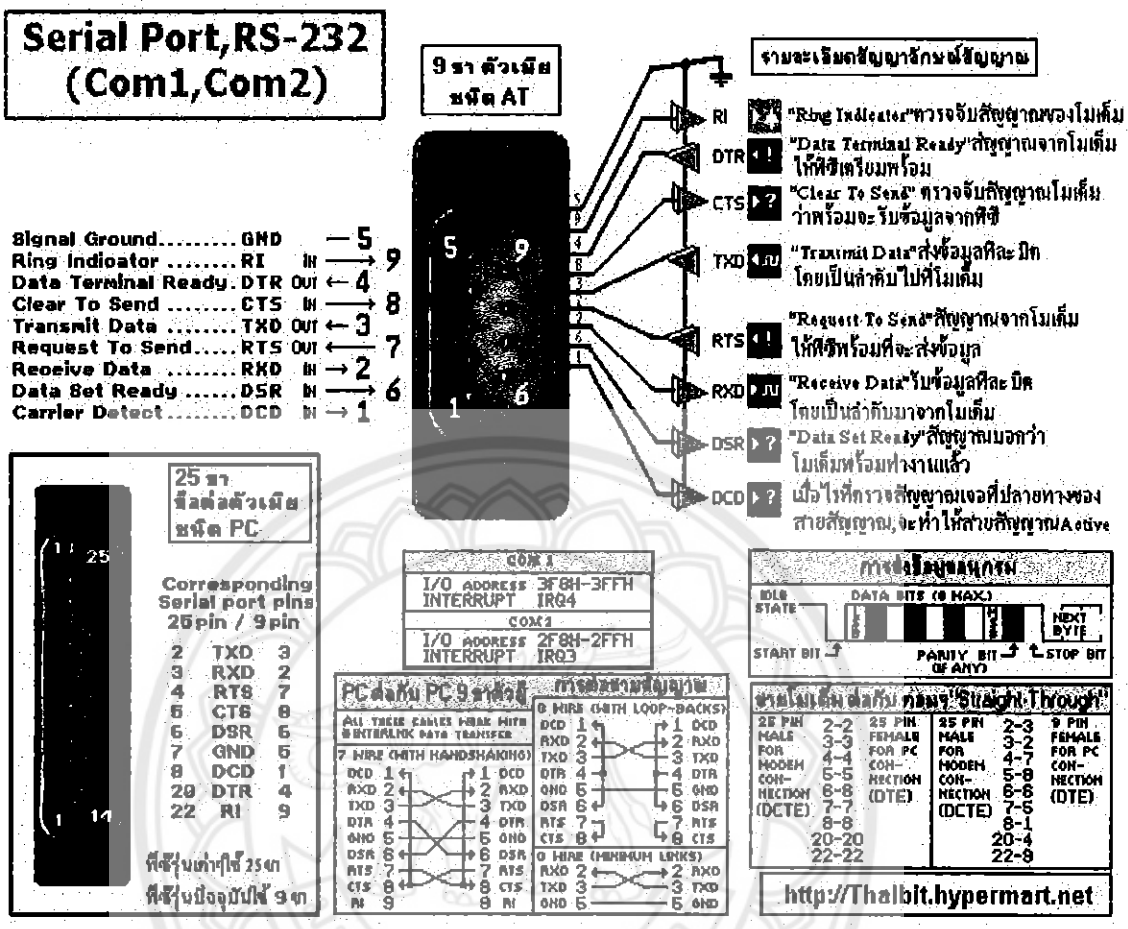
ส่วนการอินเตอร์เฟสทางลอจิกของพอร์ต RS-232 ออกแบบให้การสื่อสารมีความยาวไม่เกิน 50 ฟุต ที่ความเร็ว 20,000 บิตต่อวินาที (สามารถใช้งานที่ความยาวสายสื่อสารมากกว่านี้ หรือที่ความเร็วสูงกว่านี้ หรือทั้งสองอย่างได้ แต่มาตรฐานแล้วกำหนดไว้ที่ 50 ฟุต) การสื่อสารจะติดต่อโดยผ่านสายไฟ 25 เส้น ซึ่งแยกหน้าที่แตกต่างกันออกไป พอร์ต RS-232 สามารถรองรับการสื่อสารได้ทั้งแบบซิงโครนัส หากใช้ในการสื่อสารแบบอะซิงโครนัสแล้วเส้นควบคุมบางเส้นใน 25 เส้นนี้ ไม่จำเป็นต้องใช้

ส่วนสำคัญของพอร์ต RS-232 คือ Flow Control หรือส่วนควบคุมการไหลของข้อมูล ตัวอย่างเช่น เครื่องพริ้นเตอร์ส่งสัญญาณออกมาว่า " หยุด บัฟเฟอร์เต็ม รอสักครู่จะแจ้งกลับไปภายหลัง " หลังจากนั้น พริ้นเตอร์ก็จะพิมพ์งานในบัฟเฟอร์ต่อไป แล้วจะแจ้งกลับไปว่า " พร้อมที่จะรับข้อมูลต่อไปแล้ว ส่งข้อมูลเข้ามาได้ " เป็นต้น

RS-232 มีการจัดขา 2 แบบ คือ 9 และ 25 ขา แต่ในโครงงานนี้ได้ใช้แบบ 9 ขา ซึ่งมีการจัดการขาและรายละเอียดฟังก์ชันการทำงานดังนี้

ตารางที่ 2.4 รายละเอียดฟังก์ชันการทำงานของคอนเน็กเตอร์ของพอร์ต RS-232 แบบ 9 ขา

ขาที่	ลักษณะการทำงาน	คำอธิบาย
1	Input	Data Carrier detect (DCD)
2	Input	Receive Data (RX)
3	Output	Transmit Data (TX)
4	Output	Data Terminal Ready (DTR)
5	GND	Signal Ground
6	Input	Data Set Ready (DSR)
7	Output	Request To Send (RTS)
8	Input	Clear To Send (CTS)
9	Input	Ring Indicator (RI)



รูปที่ 2.3 การจัดการขาคอนเน็กเตอร์ของพอร์ต RS-232 แบบ 9 ขา

2.3 ET-EM PLUS (ET- EPROM EMULATOR)

ET-EM PLUS คือ บอร์ดวงจรที่ถูกสร้างขึ้นมาเพื่อใช้แทนส่วนของตัว EPROM หรือ RAM ซึ่งจะมีความง่ายและสะดวกกว่าการใช้ EPROM หรือ RAM จริง ๆ ทำให้เหมาะกับการพัฒนาระบบไมโครต่าง ๆ หรือในการแก้ไขเปลี่ยนแปลงโปรแกรมการทำงานของเครื่องเป็นไปโดยง่าย

เราสามารถเขียนข้อมูลเข้าไปในส่วนของ ET-EM PLUS ได้โดยตรงซึ่งต่างกับการที่เราต้องนำ EPROM นั้นออกมาเขียนเปลี่ยนแปลงแล้วจึงนำกลับเข้าไปใส่ในวงจรใหม่

2.3.1 งานที่นำไปใช้

ET-EM PLUS สามารถนำไปใช้งานได้หลายรูปแบบขึ้นอยู่กับผู้ใช้งาน ยกตัวอย่างเช่น

1. ROM MONITOR ใช้ในการเขียนโปรแกรมในส่วนของ MONITOR PROGRAM ของเครื่องไมโครต่าง ๆ เพราะในการเริ่มต้นพัฒนาระบบไมโครอะไรก็ตามนั้นยังไม่มีส่วนของโปรแกรมในการเขียนอ่านข้อมูลจากภายนอก ซึ่งถ้านำ ET-EM PLUS มาใช้ก็จะสะดวกกว่าใช้ EPROM เป็น MONITOR จริง ๆ มาก
2. CHARACTER GENERATOR สามารถใช้ ET-EM PLUS ในการทำเป็น CHARACTER GENERATOR ในการเปลี่ยนแปลงหรือเพิ่มเติมตัวอักษรเข้าไป เช่น ทำภาษาไทยในเครื่องพิมพ์ ซึ่งสามารถทดสอบเขียนแล้วทดสอบทำงานได้โดยตรง
3. BIOS เป็น ROM BIOS ในเครื่อง PC ใช้ทดสอบหรือแก้ไขตัวโปรแกรม BIOS ของเครื่องใช้หรือใช้ทดสอบเขียนโปรแกรมตรวจสอบการทำงานเป็นส่วน ๆ ของเครื่องก็ได้ ฯลฯ

2.3.2 คุณสมบัติของ ET-EM PLUS

1. สามารถส่งผ่านข้อมูลจากเครื่องพีซีได้ทาง PRINTER PORT ทำให้ง่ายในการจะหาเครื่องพีซีมาใช้ เพราะเครื่องพีซีต่าง ๆ นั้นจะมีส่วนของ PRINTER PORT อยู่แล้ว
2. ใช้กับไฟล์ได้หลายรูปแบบ เช่น

   - BINARY FILE

   - INTEL HEX FILE

   - MOTOLORA FILE (S FORMAT)

3. สามารถเปลี่ยนแปลงข้อมูลใน ET-EM PLUS ทั้งหมดหรือบางไบต์ก็ได้โดยไม่จำเป็นต้องโหลดไฟล์ใหม่หมด
4. สามารถตั้ง OFFSET คือ ค่าตำแหน่งที่อยู่ของข้อมูลจากไฟล์ได้โดยตรงอย่างอิสระ
5. สามารถเลือกรูปแบบการส่งข้อมูลจากไฟล์ได้หลายแบบ เช่น ส่งเต็มข้อมูล ส่งเฉพาะไบต์คู่ ส่งข้อมูลเฉพาะไบต์คี่ ซึ่งทำให้เราสามารถนำ ET-EM PLUS 2 ตัว มา ต่อกับ PRINTER PORT 2 ชุด
6. สามารถต่อกับ PRINTER PORT ได้ทั้ง LPT1, LPT2 หรือ MONO CARD ได้โดยอิสระต่อกัน
7. มีความเร็วในการส่งผ่านข้อมูลสูงมาก
8. สามารถ RESET CPU ได้เมื่อโหลดข้อมูลแล้ว โดยมีแบบ RESET HIGH และ RESET LOW

2.4 CP-SB31 (SINGLE BOARD 31 ON PC)

CP-SB31 ถูกสร้างขึ้นเพื่อใช้งานควบคุม ซึ่งตรงกับหน้าที่หลักของ CPU ในตระกูล MCS51 คือ เป็นไมโครคอนโทรลเลอร์ โครงสร้างทางกายภาพของบอร์ด CP-SB31 มีดังนี้

ลักษณะของบอร์ด CP-SB31

- CPU 8031 (ON COARD) หรือ 8032, 8052, 8751

MEMORY

- มี SOCKET ขนาด 28 PIN 2 ตัว สามารถใส่หน่วยความจำได้สูงสุด 96 KB

I/O

- 8 x 3 บิต INPUT / OUTPUT (8255)
- 8 x 1 บิต INPUT / OUTPUT (PORT 1)
- 1 SERAIL PORT (RS-232)

POWER

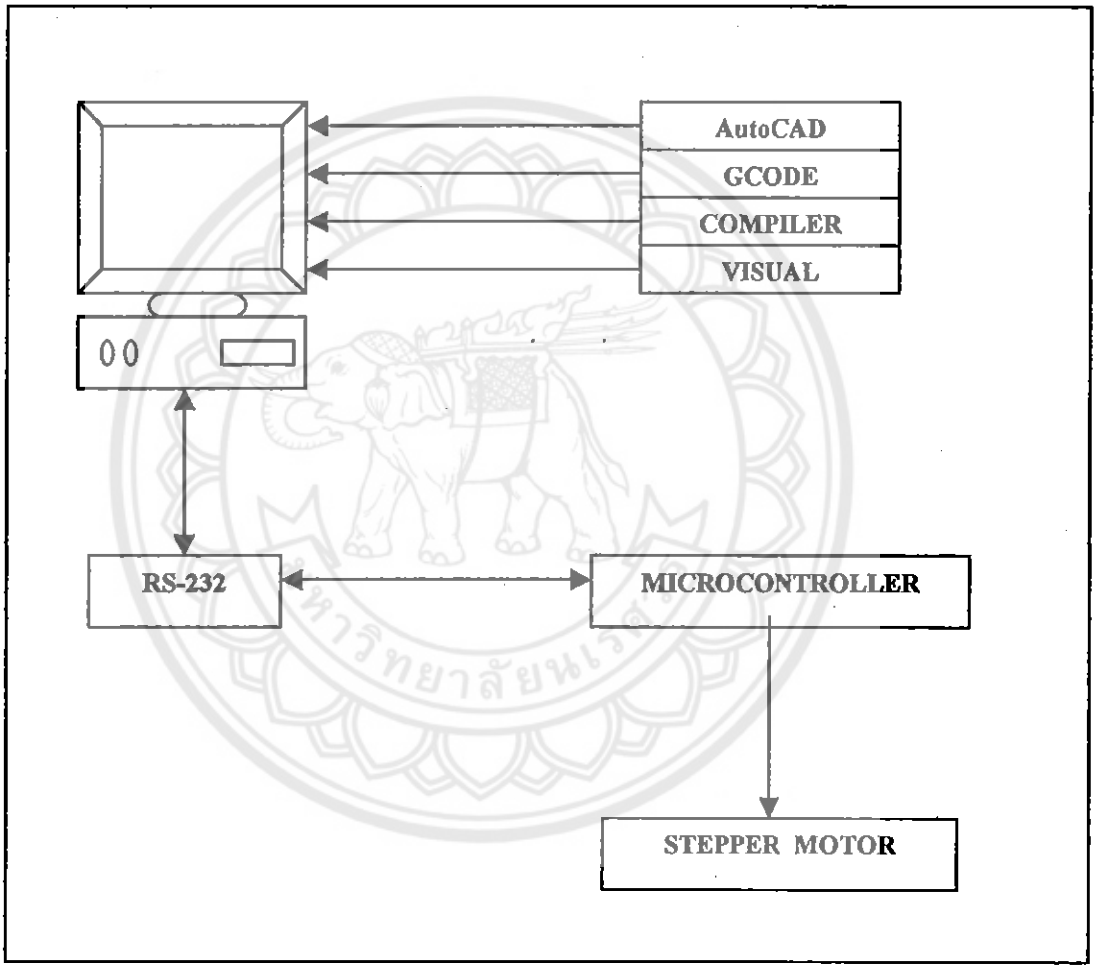
- 10 VDC POWER SUPPLY JACK
- 5 VDC (REGULATE) 7805 ON BOARD

คุณลักษณะพิเศษของ CP-SB31

1. หน่วยความจำสามารถเลือกได้ทั้งขนาด ตำแหน่ง และลักษณะการทำงาน (DATA MEMORY, CODE MEMORY, CODE & DATA MEMORY)
2. สามารถพัฒนาโปรแกรมได้ทั้งภาษาแอสเซมบลี้ (ร่วมกับ SB31 – DEBUGGER) หรือ ภาษาเบสิค (เพื่อใช้ 8052 AH – BASIC) หรือ ET EPROM EMULATOR ก็ได้
3. ต่อกับ LCD ได้ทันที โดยไม่ต้องใช้ I/O พอร์ต
4. มี I/O พอร์ต ขนาด 8 บิต ถึง 4 พอร์ต
5. ต่อร่วมกับอุปกรณ์สนับสนุนของบริษัทอีทีทีได้ทันที เช่น SSRAC, RTC, 72IO, ET-AD

บทที่ 3
การออกแบบเครื่องตัดโลหะอัตโนมัติ
จากข้อมูลภาพต้นแบบ

3.1 โครงสร้างภาพรวมการทำงานของเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ

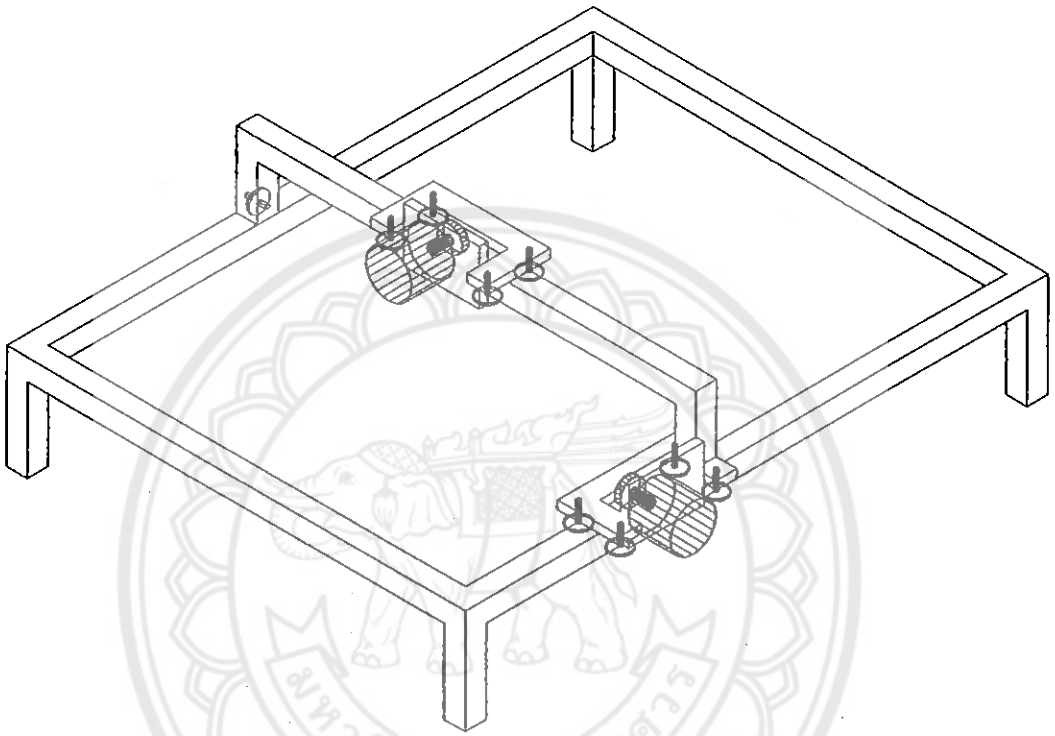


รูปที่ 3.1 โครงสร้างภาพรวมของโครงงานเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ

3.2 ศึกษาและออกแบบโครงสร้าง

1. ทำการออกแบบโครงสร้างของเครื่องตัดโลหะ โดยคำนึงถึงการใช้งานที่เหมาะสม
2. ศึกษาการทำงานของสเต็ปเปอร์มอเตอร์ เพื่อเลือกใช้สเต็ปเปอร์มอเตอร์ให้เหมาะสมกับงาน
3. ออกแบบลักษณะของโครงสร้าง
   - ตัวโครงสร้าง ทำด้วยเหล็ก

- ขนาดของโครงสร้าง คือ กว้าง 54 เซนติเมตร ยาว 65 เซนติเมตร และสูง 28 เซนติเมตร
- การเคลื่อนที่ของตัวโครงสร้าง จะมีส่วนที่เคลื่อนที่ได้ 2 ส่วน คือ
  - ส่วนที่ทำหน้าที่ในการตัดโลหะ จะเคลื่อนที่ได้ในแนวแกน X
  - ส่วนที่ทำการรองรับคานของตัวตัดโลหะ จะเคลื่อนที่ได้ในแนวแกน Y



รูปที่ 3.2 ภาพ Isometric View ของส่วนโครงสร้าง

3.3 ส่วนของโปรแกรม

ในส่วนของโปรแกรมของโครงงานนี้มีวิธีการดำเนินงาน คือ

3.3.1 โปรแกรม AutoCAD[8]

โปรแกรม AutoCAD นี้จะช่วยในการสร้างภาพต้นแบบที่จะใช้กับเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ ให้ออกมาเป็นชิ้นงานที่จะนำมาใช้งาน

3.3.2 โปรแกรม GCODE

โปรแกรม GCODE จะเป็นซอฟต์แวร์ในระบบ CAM(Computer Aided Manufacturing) ที่จะทำงานร่วมกับโปรแกรม AutoCAD เพื่อใช้ประโยชน์ในการสร้างรหัสตัวเลข ตัวอักษร เพื่อควบคุมการเคลื่อนที่ของเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ

3.3.3 โปรแกรม Visual Basic[3]

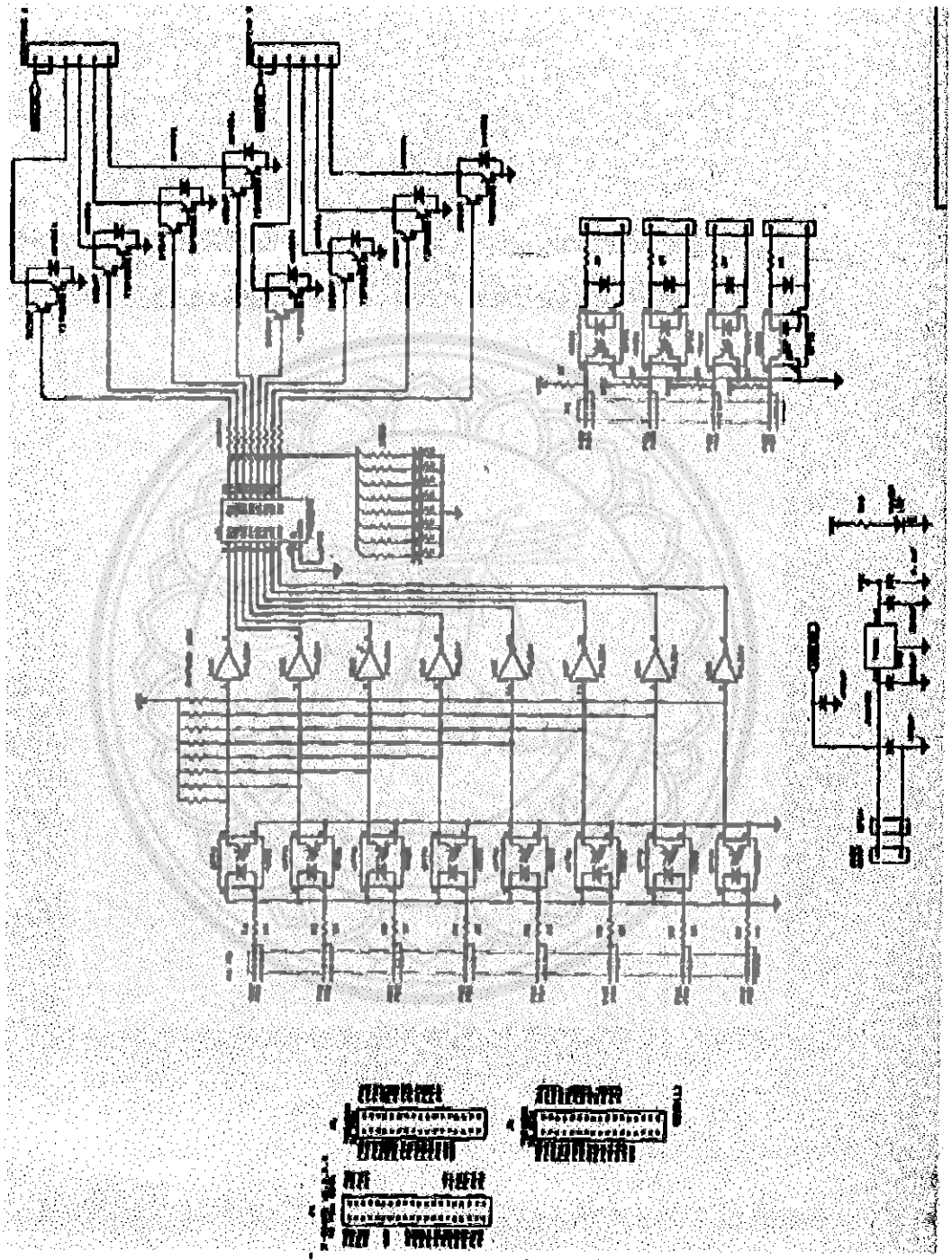
โปรแกรม Visual Basic จะมาเป็นเครื่องมือใช้งานในการติดต่อระหว่างเครื่องตัวโลหะอัตโนมัติกับผู้ใช้งาน โดยผู้ใช้งานสามารถสั่งงานผ่านทางโปรแกรม Visual Basic ได้เลย

3.3.4 โปรแกรมภาษา C[6]

ภาษา C จะเป็นตัวที่ใช้ในการติดต่อกับตัวไมโครคอนโทรลเลอร์ให้สามารถใช้งานกันได้และทำการควบคุมการทำงานของสเต็ปเปอร์มอเตอร์

3.4 การเลือกสเต็ปเปอร์มอเตอร์

การเลือกใช้สเต็ปเปอร์มอเตอร์ควรเลือกให้เหมาะสมกับงานและหาซื้อได้ง่าย โดยในโครงงานนี้จึงเลือกใช้สเต็ปเปอร์มอเตอร์แบบยูนิโพลาร์ 6 สาย 4 เฟส ซึ่งจะใช้สเต็ปเปอร์มอเตอร์ประเภท ET-SMCC (STEPPING MOTOR CONTROL CARD) ดังรูปที่ 3.3 โดย ET-SMCC จะเป็นการ์ดต่อเข้ากับพอร์ตของ8255 หรือพอร์ตอื่น ๆ ด้วยก็ได้โดยจะใช้ในการต่อควบคุมการทำงานของสเต็ปเปอร์มอเตอร์ โดยสามารถจะต่อกับสเต็ปเปอร์มอเตอร์ได้ 2 ตัวต่อหนึ่งการ์ด หรือจะต่อพ่วงอีกหนึ่งการ์ดเพื่อใช้กับสเต็ปเปอร์มอเตอร์ 4 ตัวต่อพอร์ต 8255 จำนวน 1 ตัวด้วยก็ได้ นอกจากนี้ ET-SMCC จะมี อินพุตในรูปลักษณะ OPTO อีก 4 อินพุต สามารถต่อเข้ากับสวิตช์ต่าง ๆ ได้อีกด้วย เช่น ต่อเข้ากับ MICRO SW. ตรวจการหมุนของสเต็ปเปอร์มอเตอร์ว่าถึงจุดที่ต้องการแล้วหรือยัง เป็นต้น



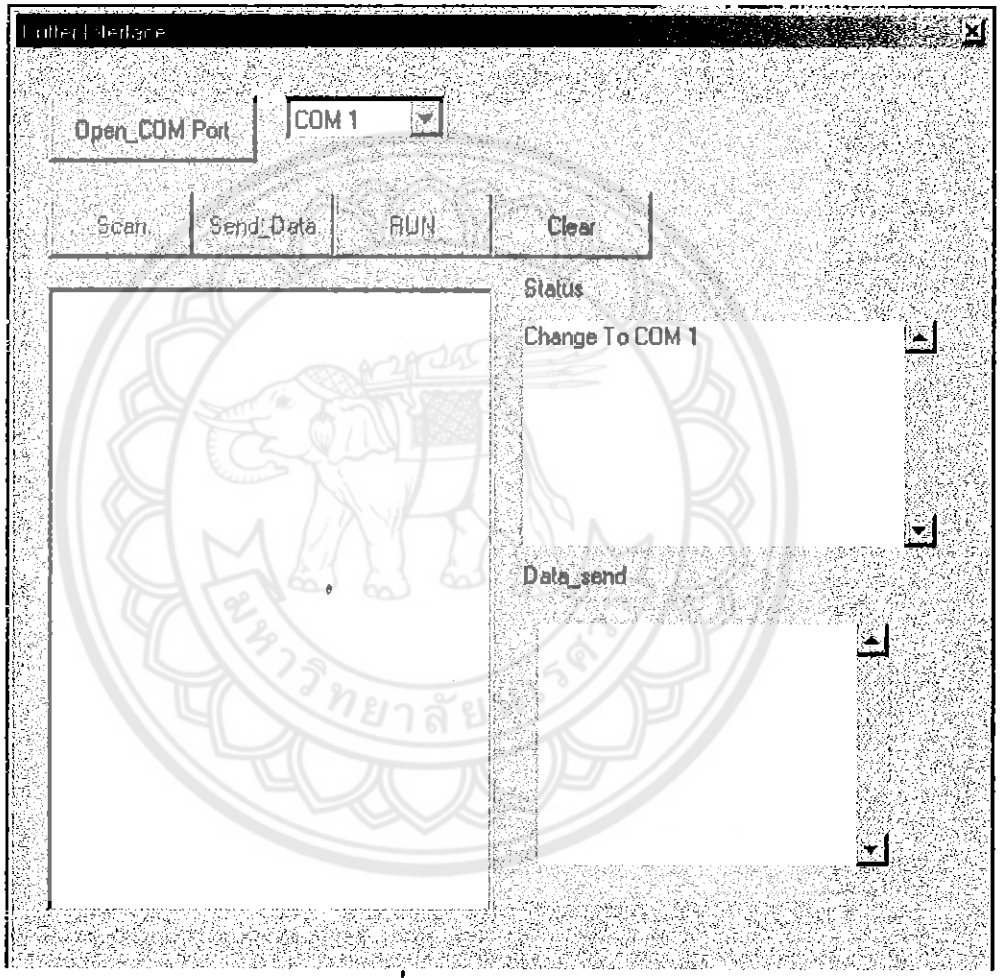
รูปที่ 3.3 วงจรควบคุมสเต็ปเปอร์มอเตอร์

3.5 ส่วนของโปรแกรม

ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบนี้ จะแบ่งออกเป็น 2 ส่วน คือ ส่วนของการติดต่อกับผู้ใช้ จะใช้ภาษา Visual Basic ในการพัฒนา และส่วนของการควบคุมการทำงานของสเต็ปเปอร์มอเตอร์จะใช้ภาษาซีในการพัฒนา

3.5.1 ส่วนประกอบของโปรแกรมในการติดต่อกับผู้ใช้

เมื่อทำการเปิดโปรแกรมจะเจอกับหน้าจอที่มีลักษณะดังต่อไปนี้

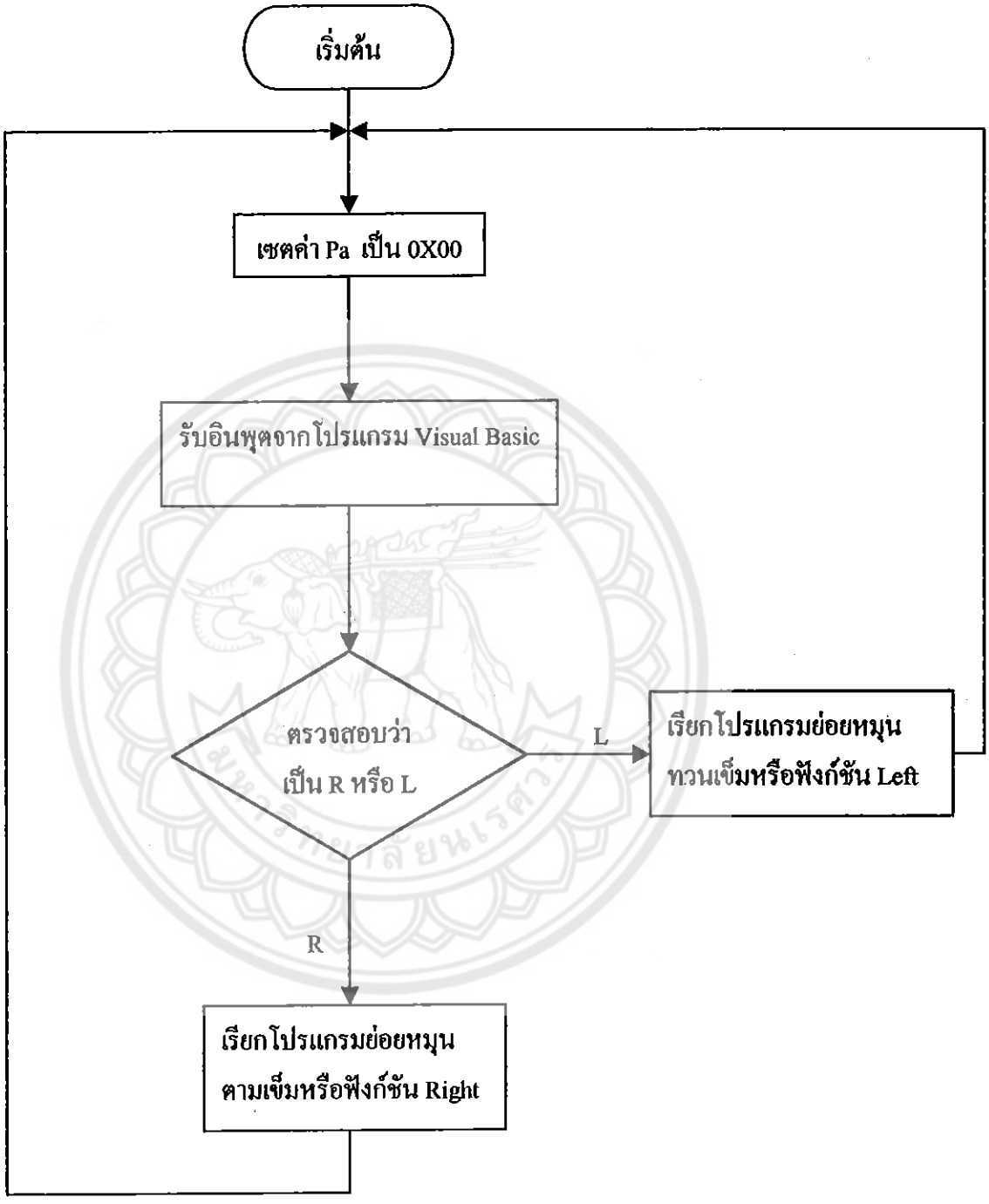


รูปที่ 3.4 หน้าจอของโปรแกรม

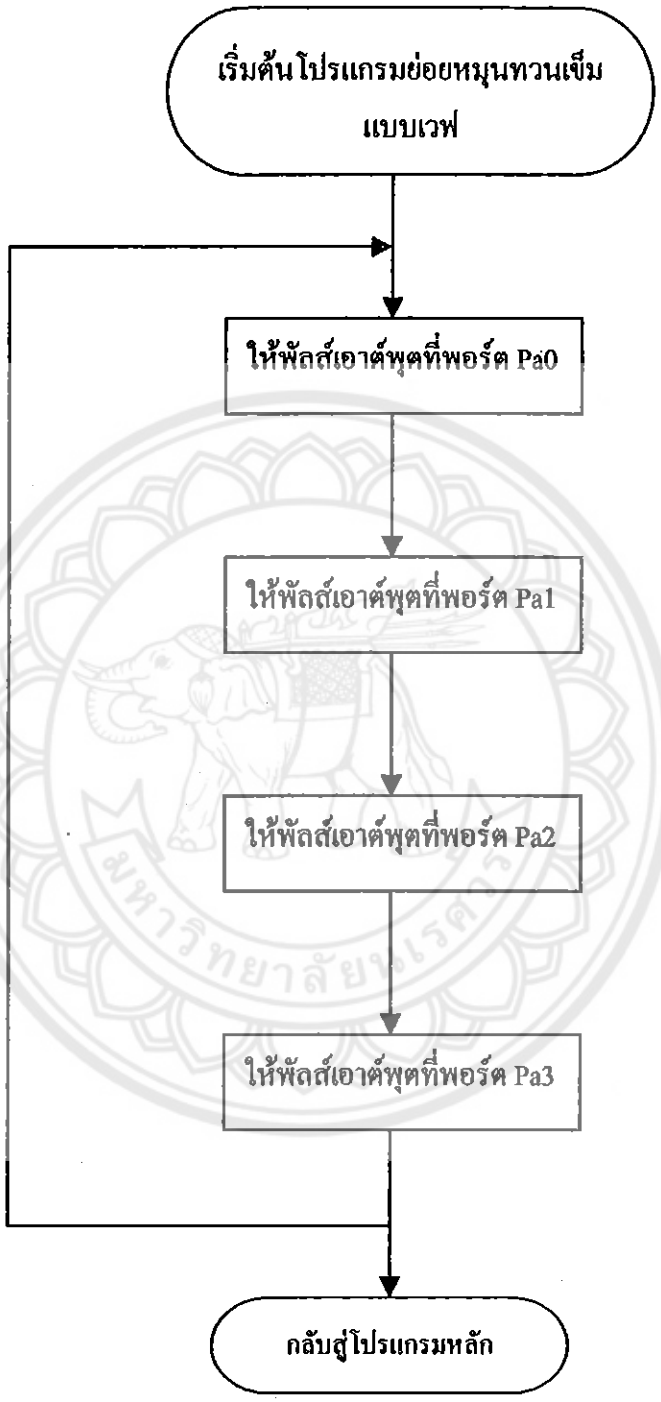
3.5.2 ส่วนประกอบของโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์

ลักษณะการเขียนโปรแกรมควบคุมการหมุนของสเต็ปเปอร์มอเตอร์ที่ใช้ในโครงงานนี้เป็นแบบเวฟ (Wave) ซึ่งเป็นการกระตุ้นแบบที่ง่ายที่สุดแบบหนึ่ง

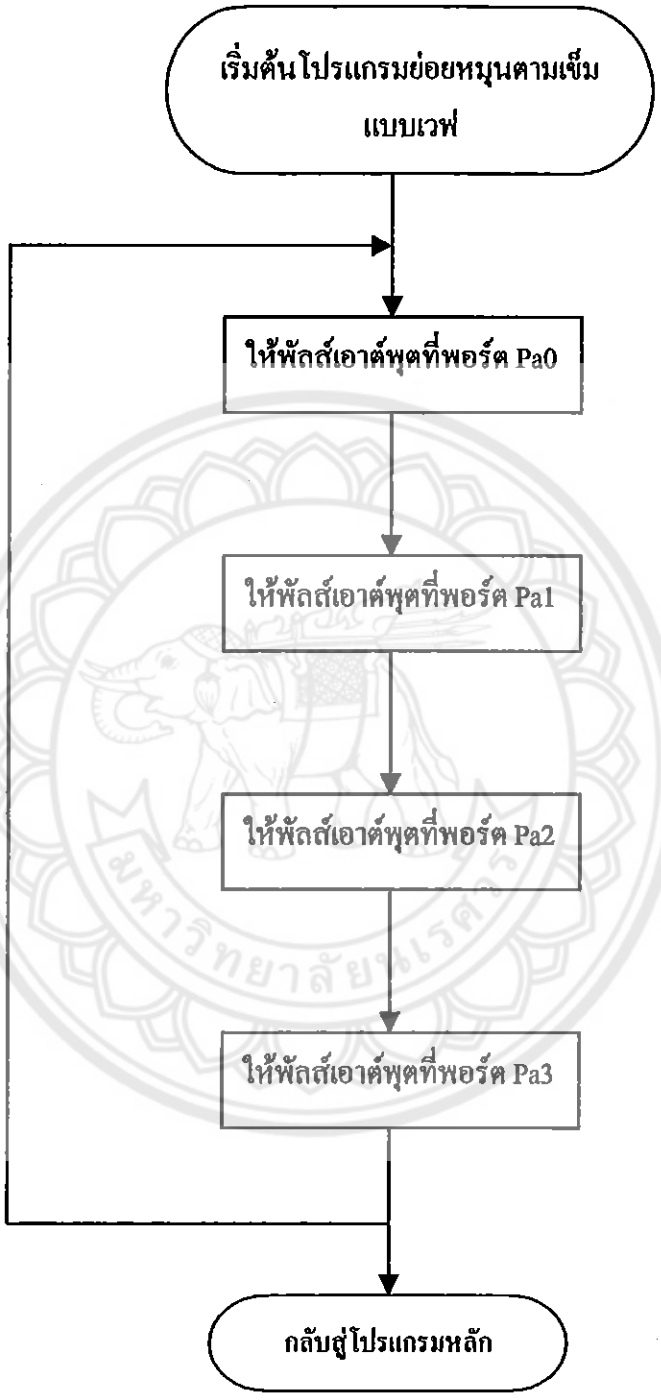
เราสามารถเขียนโฟลว์ชาร์ตการทำงานของโปรแกรมนี้ ได้ดังนี้



รูปที่ 3.5 โฟลว์ชาร์ตโปรแกรมหลักของโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์



รูปที่ 3.6 โฟลว์ชาร์ตโปรแกรมย่อยหมุนทวนเข็มนาฬิกาแบบเวฟ

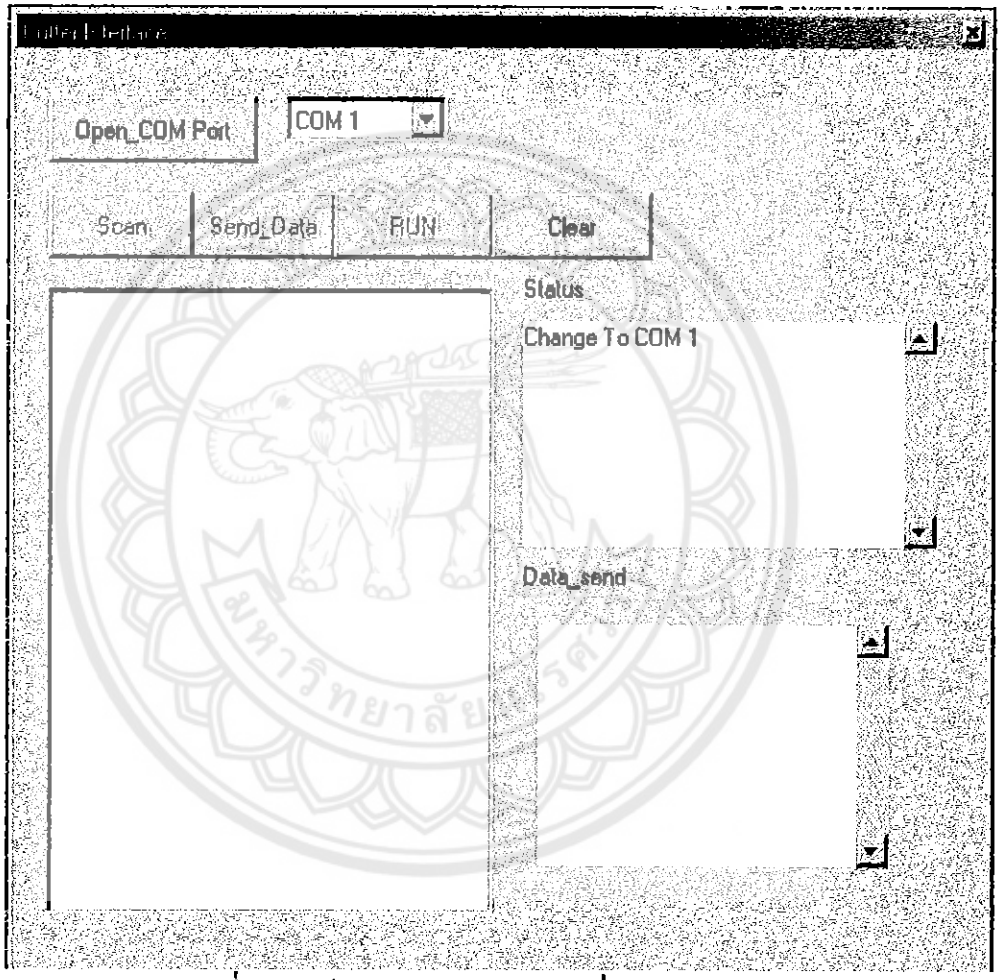


รูปที่ 3.7 โฟลว์ชาร์ตโปรแกรมย่อยหมุนตามเข็มนาฬิกาแบบเวฟ

3.6 การใช้งานเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบ

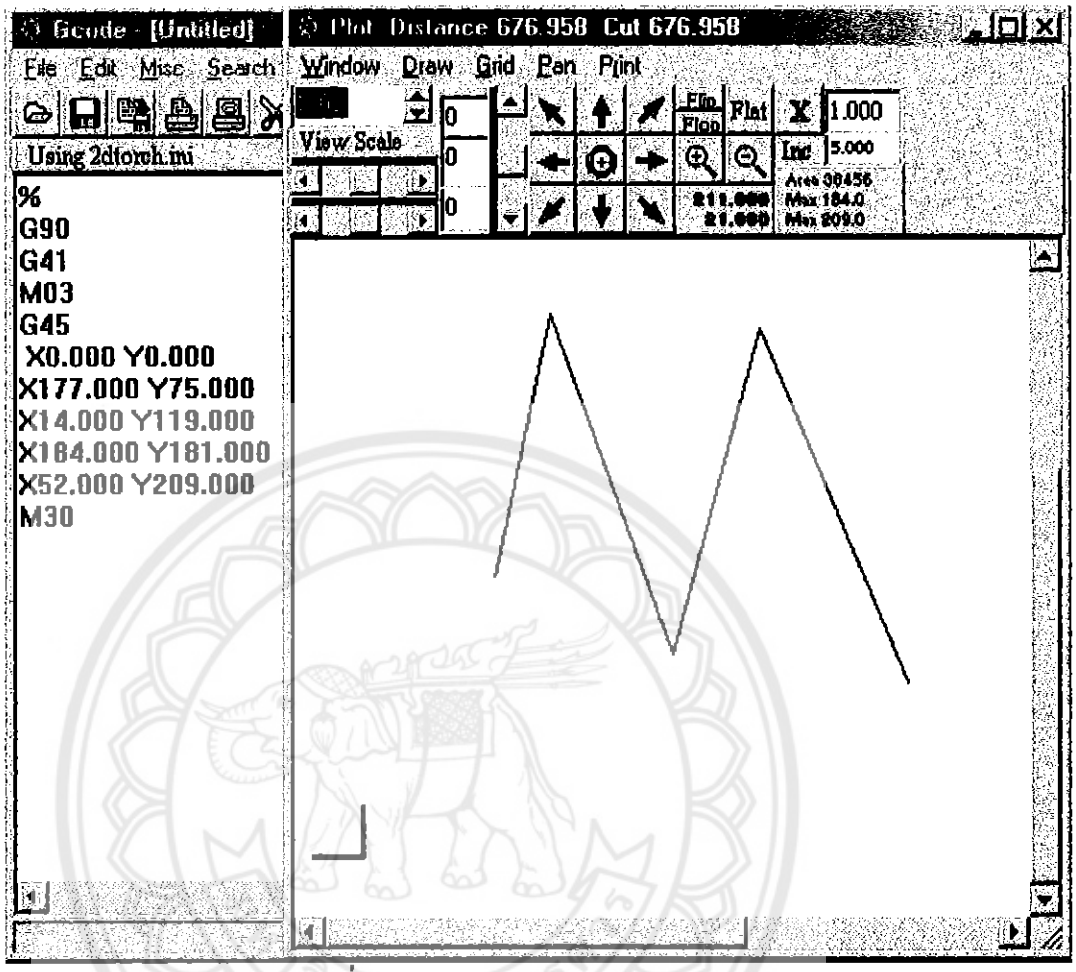
1. ติดตั้งเครื่องตัดโลหะอัตโนมัติจากข้อมูลภาพต้นแบบให้มีการเชื่อมต่อกับเครื่องคอมพิวเตอร์โดยการติดต่อผ่านทางพอร์ตอนุกรม RS-232

2. เมื่อทำการเชื่อมต่อแล้วให้ทำการเรียกโปรแกรมเครื่องตัดโลหะที่ใช้ในการติดต่อกับผู้ใช้งานขึ้นมา จะได้หน้าจอหลักที่มีลักษณะการใช้งาน ดังนี้



รูปที่ 3.8 หน้าจอของโปรแกรมก่อนที่จะมีการทำงาน

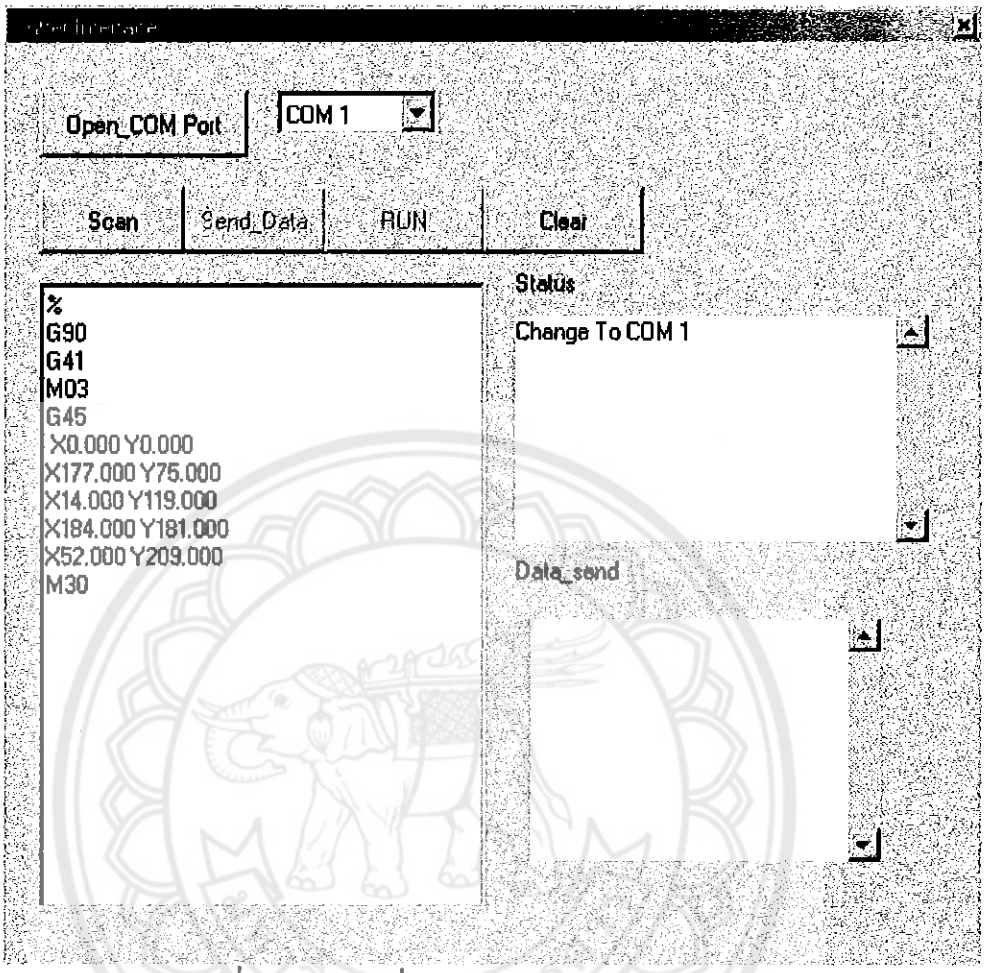
3. ทำการเปิดโปรแกรม GCODE ที่ได้มีการวาดภาพไว้แล้ว ซึ่งจะมีลักษณะการทำงานดังนี้



รูปที่ 3.9 หน้าจอของโปรแกรม GCODE

ห้องสมุดคณะวิศวกรรมศาสตร์

4. นำรหัสตัวเลขที่ได้มาใส่ไว้ใน Textbox ของหน้าจอเครื่องตัดโลหะ ซึ่งจะมีลักษณะดังนี้



รูปที่ 3.10 หน้าจอที่มีการเรียกใช้โปรแกรม GCODE

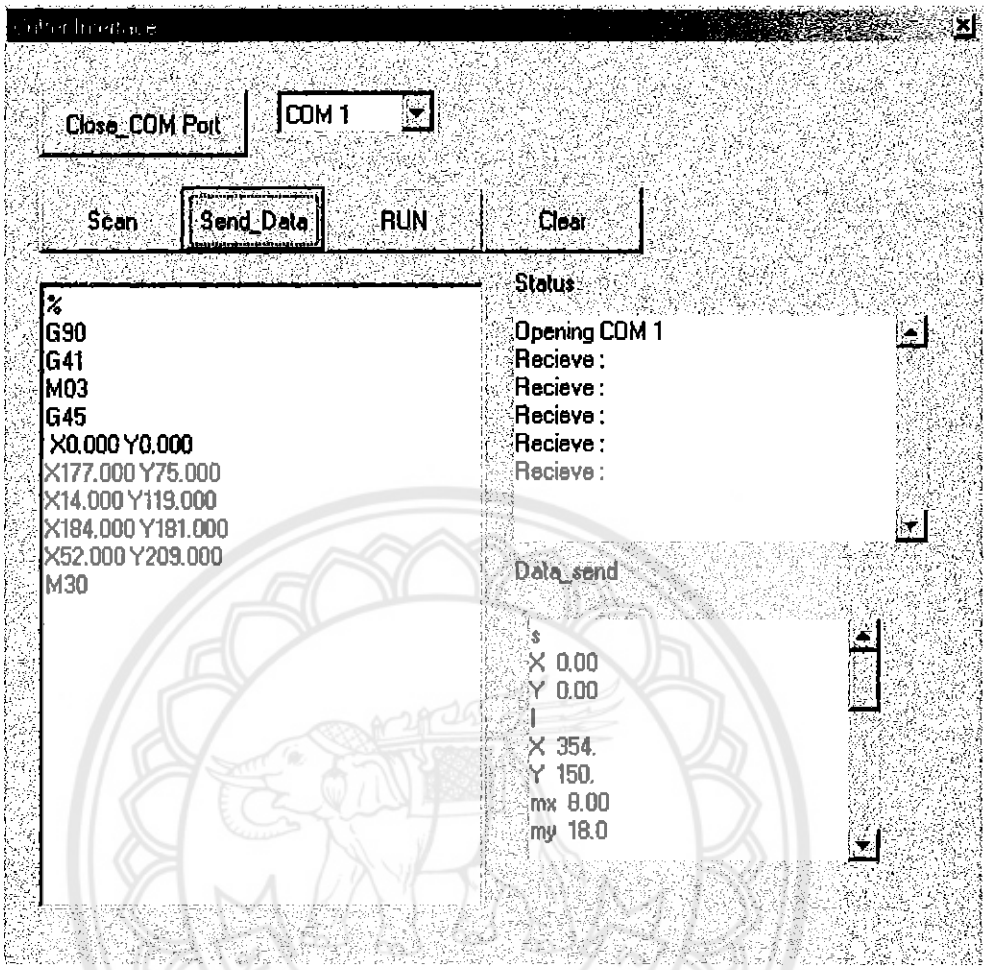
5. เมื่อได้หน้าจอดังรูปที่ 3.7 แล้วให้ทำดังต่อไปนี้

5.1 กดปุ่ม Open_COM Port เพื่อทำการเปิดพอร์ตที่จะใช้งาน

5.2 กดปุ่ม Scan เพื่อให้โปรแกรมทำการตรวจสอบหาคำที่จะสามารถระบุได้ว่าเป็นโปรแกรม GCODE ที่จะใช้ในการวาดเส้นตรง ได้แก่ G01 วงกลมที่มีลักษณะตามเข็มนาฬิกา ได้แก่ G02 หรือวงกลมที่มีลักษณะทวนเข็มนาฬิกา ได้แก่ G03

5.3 กดปุ่ม Send_Data เพื่อทำการส่งข้อมูลไปให้กับ Microcontroller ได้ทำงานโดยผ่านทางพอร์ตอนุกรม RS-232 ซึ่งจะมีลักษณะของหน้าจอดังนี้

4400613
TJ
1185
.N277g
2544 c.1



รูปที่ 3.11 หน้าจอที่ได้ทำการ Send_Data แล้ว

6. เมื่อต้องการให้เครื่องตัดโลหะทำงานให้ทำการกดปุ่ม RUN แล้วเครื่อง**ตัดโลหะอัตโนมัติจากข้อมูล**ภาพต้นแบบก็จะทำงานตามที่ต้องการ

บทที่ 4
ผลการทดลองและการวิเคราะห์

4.1 การทดลองบอร์ดไดเวอร์กับโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์

การทดลองที่ 1 : ทดสอบความถูกต้องของวงจรไดเวอร์สเต็ปเปอร์มอเตอร์

ก่อนที่จะทำการต่อบอร์ดไดเวอร์ ได้ทดลองต่อวงจรเพียงส่วนเดียวจากทั้งหมดสองส่วนก่อน พบว่าได้ผล จึงเริ่มทำการต่อวงจรทั้งหมดแล้วต่อเอาต์พุตเข้ากับ LED 2 ตัว พบว่าได้ผล จึงต่อบอร์ดไดเวอร์เข้ากับตัวสเต็ปเปอร์มอเตอร์ แล้วลองป้อนอินพุตเข้าไป พบว่าได้เอาต์พุตออกมาตรงตามที่ต้องการ

การทดลองที่ 2 : ทดสอบโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์

ทำการเขียนโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์ด้วย MCS51 โดยส่วนของโปรแกรมจะมีหน้าที่รับข้อมูลจาก Visual Basic มาวิเคราะห์ว่าสั่งให้สเต็ปเปอร์มอเตอร์หมุนไปทางขวาหรือทางซ้ายทีละกี่สเต็ป หรืออาจสั่งให้หยุดก็ได้ แล้วส่งค่าไปที่บอร์ดไดเวอร์ผ่านทางพอร์ต RS – 232 โดยแบ่งเป็นการทดลองย่อยดังนี้

การทดลองที่ 2.1 : ทดสอบความละเอียดของการหมุนของสเต็ปเปอร์มอเตอร์

- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนตามเข็มไป 1 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไป 1 สเต็ปแรกเท่านั้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนตามเข็มไป 2 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไป 2 สเต็ปแรกเท่านั้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนตามเข็มไป 3 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไป 3 สเต็ปแรกเท่านั้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนตามเข็มไป 4 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไปได้ทุกครั้งที่มีการส่งค่าให้

การทดลองที่ 2.2 : ทดสอบความละเอียดของการหมุนของสเต็ปเปอร์มอเตอร์

- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนทวนเข็มไป 1 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไป 1 สเต็ปแรกเท่านั้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนทวนเข็มไป 2 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไป 2 สเต็ปแรกเท่านั้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนทวนเข็มไป 3 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไป 3 สเต็ปแรกเท่านั้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนทวนเข็มไป 4 สเต็ป หลายครั้ง
  พบว่า : สเต็ปเปอร์มอเตอร์เคลื่อนไปได้ทุกครั้งที่มีการส่งค่าให้

การทดลองที่ 2.3 : ทดสอบความถูกต้องของจำนวนสเต็ปการหมุนของสเต็ปเปอร์มอเตอร์

- ก่อนการทดลองให้ทำเครื่องบอกตำแหน่งไว้เป็นจุดเริ่มต้น
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนตามเข็มไป 200 สเต็ป
  พบว่า : หมุนไปได้ระยะหนึ่งแล้วหยุด (ทำเครื่องหมายบอกตำแหน่งไว้)
- ทดลอง : ส่งค่าให้สเต็ปเปอร์มอเตอร์หมุนทวนเข็มไป 200 สเต็ป
  พบว่า : หมุนไปได้ระยะหนึ่งแล้วหยุด (สังเกตเครื่องหมายบอกตำแหน่ง) พบว่าตรงกับตำแหน่งเริ่มต้น
- ทำการทดลองซ้ำ และลองเปลี่ยนค่าของสเต็ปการหมุน พบว่าได้ผลเช่นกัน

การทดลองที่ 2.4 : ทดสอบความสามารถของโปรแกรมในการสั่งให้สเต็ปเปอร์มอเตอร์หยุดหมุน

ทดลอง : สั่งให้สเต็ปเปอร์มอเตอร์หยุดหมุน

พบว่า : สามารถหยุดหมุนได้

การทดลองที่ 2.5 : ทดสอบโปรแกรมที่สั่งให้สเต็ปเปอร์มอเตอร์หมุนวนต่อเนื่องกัน

ทดลอง : เขียนโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์แบบอินเตอร์รัพท์ (Interrupt) เพื่อสั่งให้สเต็ปเปอร์มอเตอร์หมุนตามเข็ม 100 สเต็ป หมุนทวนเข็ม 200 สเต็ป หมุนตามเข็ม 300 สเต็ป หมุนทวนเข็ม 100 สเต็ป

พบว่า : สเต็ปเปอร์มอเตอร์สามารถทำงานได้ตามที่ต้องการ

การทดลองที่ 3 : การคำนวณหาขอบเขตของสเต็ปเปอร์มอเตอร์ในระนาบแกน X และแกน Y

ทดลอง : X = 810 สเต็ป/ 43 เซนติเมตร

พบว่า : 1 สเต็ป / 0.0531 เซนติเมตร

ทดลอง : Y = 720 สเต็ป/ 39 เซนติเมตร

พบว่า : 1 สเต็ป / 0.0542 เซนติเมตร

4.2 การทดลองในส่วนของโปรแกรมวาดภาพ

การทดลองที่ 4 : ทดลองโปรแกรม AutoCAD และโปรแกรม GCODE กับฟังก์ชั่นถอดโค้ด GCODE ใน Visual Basic

ทดลอง : นำโปรแกรม AutoCAD ในการวาดภาพให้มาปรากฏที่โปรแกรม GCODE แล้วให้ทำการคัดลอก Text File จากฟังก์ชั่น GCODE มาวางไว้ในโปรแกรม Visual Basic ให้สามารถทำงานได้

พบว่า : สามารถทำงานได้

ทดลอง : นำโปรแกรม GCODE มาทำการวาดภาพเลยแล้วให้ฟังก์ชั่นถอดโค้ด GCODE ใน Visual Basic ทำงานได้

พบว่า : สามารถทำงานได้

4.3 การทดลองการรับส่งข้อมูลระหว่าง Visual Basic กับ MCS-51

เพื่อให้โปรแกรม Visual Basic และ MCS-51 สามารถติดต่อสื่อสารกันได้

การทดลองที่ 5 : ทดลองการส่งค่าจากโปรแกรม Visual Basic ไปยัง MCS-51

ทดลอง : โปรแกรมในการส่งค่าการขจัดที่ได้ (แปลงเป็นจำนวนสเต็ปแล้ว) ไปยังโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์เพื่อสั่งให้มอเตอร์หมุนไปซ้ายหรือขวากี่สเต็ป

พบว่า : 1. ทำการทดลองส่งค่าผ่านทาง หน้าจอ LCD ก่อนพบว่าได้ผล

2. ทำการทดลองจริง พบว่าได้ผลเช่นเดียวกัน

4.4 การทดลองฟังก์ชั่นต่าง ๆ ระหว่าง โครงสร้างกับ MCS-51

การทดลองที่ 6 : ทดลองด้วยฟังก์ชั่นต่าง ๆ ใน MCS-51 กับโครงสร้าง

ทดลอง : ทำการวาดฟังก์ชั่นต่าง ๆ เช่น วาดเส้นตรง สามเหลี่ยม สี่เหลี่ยม เป็นต้น ใน MCS-51 กับโครงสร้างเครื่องตัดโลหะ

พบว่า : สามารถวาดฟังก์ชั่นต่าง ๆ ได้ แต่จะมีความผิดพลาด (Error)

การทดลองที่ 7 : การทดสอบโปรแกรมว่าสามารถทำงานร่วมกันได้

ทดลอง : ทำการทดสอบโปรแกรมว่าสามารถทำงานร่วมกันได้และสามารถทำงานแบบอัตโนมัติได้

พบว่า : โปรแกรมสามารถทำงานร่วมกันและทำงานแบบอัตโนมัติได้

บทที่ 5

บทสรุป

จากการทดลองในบทที่ 4 นั้นจะนำผลการทดลองมาสรุปและวิจารณ์ พร้อมกับข้อเสนอแนะต่าง ๆ ที่จะเป็นประโยชน์ต่อไป

5.1 สรุปและวิจารณ์ผลการทดลอง

5.1.1 การทดลองบอร์ดไดเวอร์กับโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์

สรุปผลการทดลองที่ 1 : จากการทดลองสามารถสรุปได้ว่าบอร์ดไดเวอร์สเต็ปเปอร์มอเตอร์สามารถนำมาใช้งานได้ตรงตามความต้องการของเครื่องตัดโลหะ

สรุปผลการทดลองที่ 2.1 และ 2.2 : จากการทดลองจะพบว่าเมื่อเราสั่งให้โปรแกรมส่งค่าสเต็ปการหมุนเป็น 1 สเต็ปหลายครั้งติดต่อกันจะมีผลทำให้สเต็ปเปอร์มอเตอร์หมุนเพียงสเต็ปแรกสเต็ปเดียวเท่านั้น เนื่องจากการเขียนโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์เป็นแบบ WAVE หากต้องการให้สเต็ปเปอร์มอเตอร์หมุนได้ละเอียดมากขึ้นควรเขียนโปรแกรมแบบที่สามารถสั่งงานได้ทีละสเต็ป

สรุปผลการทดลองที่ 2.3 : จากการทดลองนี้ต้องการทดสอบการทำงานของโปรแกรมว่าสามารถทำให้สเต็ปเปอร์มอเตอร์หมุนได้ตามสเต็ปที่ต้องการหรือไม่ ซึ่งผลจากการทดลองพบว่าสเต็ปเปอร์มอเตอร์สามารถหมุนได้ตามที่ต้องการ

สรุปผลการทดลองที่ 2.4 : จากผลการทดลองทำให้สามารถสั่งให้สเต็ปเปอร์มอเตอร์หยุดหมุนได้

สรุปผลการทดลองที่ 2.5 : จากการทดลองนี้ต้องการทดสอบว่าการทำงานของโปรแกรมว่าสามารถทำให้สเต็ปเปอร์มอเตอร์หมุนได้ตามสเต็ปที่ต้องการหรือไม่ ซึ่งจากผลการทดลองพบว่าสเต็ปเปอร์มอเตอร์สามารถหมุนได้ตามสเต็ปที่ต้องการ

สรุปผลการทดลองที่ 3 : จากการทดลองนี้ต้องการคำนวณหาขอบเขตของสเต็ปเปอร์มอเตอร์ในระนาบแกน X และแกน Y ซึ่งจากผลการทดลองพบว่าสามารถหาออกมาได้ตามที่ต้องการ

5.1.2 การทดลองในส่วนของโปรแกรมวาดภาพ

สรุปผลการทดลองที่ 4 : จากการทดลองนี้ต้องการให้โปรแกรม AutoCAD และโปรแกรม GCODE กับฟังก์ชั่นถอดโค้ด GCODE ใน Visual Basic ทำงานด้วยกันได้ ซึ่งผลจากการทดลองพบว่าทั้งโปรแกรม GCODE และโปรแกรม AutoCAD สามารถทำงานแทนกันได้ในบางฟังก์ชั่น เช่น เส้นตรง วงกลม เป็นต้น

5.1.3 การทดลองการรับส่งข้อมูลระหว่าง Visual Basic กับ MCS-51

สรุปผลการทดลองที่ 5 : จากผลการทดลองนี้ พบว่าโปรแกรม Visual Basic สามารถส่งค่าการขจัดไปยังโปรแกรมควบคุมสเต็ปเปอร์มอเตอร์ได้

5.1.4 การทดลองฟังก์ชั่นต่าง ๆ ระหว่าง โครงสร้างกับ MCS-51

สรุปผลการทดลองที่ 6 : จากผลการทดลองนี้ จะพบว่าสามารถทำการวาดฟังก์ชั่นต่าง ๆ ได้กับโครงสร้างเครื่องตัดโลหะ แต่ก็จะมีความผิดพลาด (Error) อยู่บ้างเพราะความละเอียดของสเต็ปเปอร์มอเตอร์และความสมดุลของโครงสร้างไม่ดีพอ

สรุปผลการทดลองที่ 7 : จากผลการทดลองนี้ จะพบว่าโปรแกรมสามารถทำงานร่วมกันได้และสามารถติดตามวัตถุได้อย่างอัตโนมัติ

5.1.5 สรุปผลการทดลองของโครงงานเครื่องตัดโลหะอัตโนมัติจากของข้อมูลภาพต้นแบบ

1. มีความเข้าใจในหลักการการทำงานของไมโครคอนโทรลเลอร์
2. มีความเข้าใจในหลักการทำงานและการควบคุมของสเต็ปเปอร์มอเตอร์
3. มีความเข้าใจในหลักการของการอินเทอร์เฟส
4. มีความเข้าใจในการทำงานร่วมกันของส่วนต่าง ๆ ของโครงงานนี้
5. มีความสามารถนำคอมพิวเตอร์มาช่วยงานอุตสาหกรรมได้

5.2 ปัญหาและวิธีการแก้ปัญหา

1. ปัญหา : ทางด้านเทคนิคของโครงสร้างที่จะทำให้สมดุลกันยาก
   วิธีแก้ : ทำการศึกษาหาความรู้เพิ่มเติมในด้านเครื่องกล
2. ปัญหา : ความรู้ที่ไม่เพียงพอในการเขียนฟังก์ชั่นการทำงานที่ยุ่งยาก
   วิธีแก้ : ทำการศึกษาหาความรู้เพิ่มเติมในเรื่องนั้น ๆ ให้มากขึ้น
3. ปัญหาทางด้านการเงิน
   วิธีแก้ : ทำการเก็บสะสมเงินไว้ล่วงหน้า

5.3 ข้อเสนอแนะ

1. ควรจะมีการพัฒนาในส่วนของโครงสร้างให้มีความสมดุลเพิ่มมากขึ้น
2. ควรจะมีการพัฒนาให้สามารถนำไปใช้งานได้จริง

เอกสารอ้างอิง

[1] ไกรวุฒิ โรจน์ประเสริฐสุด. รวมโครงงานอิเล็คทรอนิกส์ไมโครคอมพิวเตอร์. กรุงเทพฯ : สำนักพิมพ์ซีเอ็ดยูเคชัน. 2539.

[2] สุนทร วิฑูสุรพจน์. การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051. กรุงเทพฯ : สำนักพิมพ์ เอช.เอ็นกรุ๊ป จำกัด.

[3] กิตติ ภักดีวัฒนะกุล, จำลอง ครูอุตสาหะ. Visual Basic6 ฉบับโปรแกรมเมอร์. พิมพ์ครั้งที่ 7. กรุงเทพฯ : บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด. 2543.

[4] ฉัททวุฒิ พืชผล, พิชิต สันติกุลานนท์. คู่มือเรียน Visual Basic 6. กรุงเทพฯ : บริษัท โปรวิชั่น จำกัด. 2542.

[5] ยืน ภู่วรวรรณ, วัฒนา เชียงกูล. ไมโครโปรเซสเซอร์ไมโครคอมพิวเตอร์. กรุงเทพฯ : บริษัทซีเอ็ดยูเคชั่นจำกัด. 2542.

[6] บุญเลิศ เอี่ยมทัศนา, ยืน ภู่วรวรรณ. โปรแกรมคอมพิวเตอร์ภาษาซี. กรุงเทพฯ : บริษัทซีเอ็ดยูเคชั่นจำกัด. 2537.

[7] วิทยา เรืองพรวิสุทธิ์. คู่มือโปรแกรมภาษา C. กรุงเทพฯ : บริษัทซีเอ็ดยูเคชั่นจำกัด. 2537.

[8] ภาราไดย ลดาวัลย์, กฤษณ์ไกรพ์ สิทธิเสรีประทีป. การเขียนแบบด้วยคอมพิวเตอร์. พิมพ์ครั้งที่ 7. กรุงเทพฯ : บริษัทไทยฟรีมาจำกัด. 2540.

ภาคผนวก

โปรแกรมสเต็ปเปอร์มอเตอร์

```
#include <reg51.h>
#include <absacc.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#define LCD_CONTROL   XBYTE[0xE0C0]
#define LCD_BUSY_FLAG XBYTE[0xE0C1]
#define LCD_WRITEDATA XBYTE[0xE0C2]
#define LCD_READDATA  XBYTE[0xE0C3]

#define  ID   0x30

#define Pa    XBYTE[0xE0E0]
#define Pb    XBYTE[0xE0E1]
#define Pc    XBYTE[0xE0E2]
#define Pcon  XBYTE[0xE0E3]

#define Startx  22
#define Starty  -13
#define Startz  0

int itimex,itimey,l,k,m;
unsigned char msBUF,inkey;
unsigned char equ[10];
int kx,ky,kxm,kym;
typedef struct StructTableCommand
{
   int X;
```

```
    int Y;
    int mx;
    int my;
    int j;
    int command;
};

xdata struct StructTableCommand TableCommand[100];
xdata char messx[4],messy[4],messi[4],messj[4],messmx[4],messmy[4],mess[4],comm,check_send;

void delay(unsigned int time)
{
    data unsigned int i;
    for(i= 0;i < time;i++)
                {
                }
}

void init_serial(void)
{
    SCON = 0x52;
    PCON=0x00;
    TMOD = 0x20;
    TH1 = 0xfd;
    TR1 = 1;
}

unsigned char recv_char(void)
{
  while (!RI)
     {
     }
```

```
  RI = 0;
  return(SBUF);
}
void recv_messx(void)
{
   unsigned char i,mBUF;
   for (i=0;i<=4;i++)
         {
           mBUF=recv_char();
           messx[i]=mBUF;
         }
}

void recv_messy(void)
{
   unsigned char i,mBUF;
   for (i=0;i<=4;i++)
         {
           mBUF=recv_char();
           messy[i]=mBUF;
         }
}

void recv_messmx(void)
{
   unsigned char i,mBUF;
   for (i=0;i<=4;i++)
         {
           mBUF=recv_char();
           messmx[i]=mBUF;
         }
 }
```

```
void recv_messmy(void)
{
  unsigned char i,mBUF;
  for (i=0;i<=4;i++)
        {
          mBUF=recv_char();
          messmy[i]=mBUF;
        }
}

int recv_command(void)
{
  unsigned char mBUF;
  int com ;

  mBUF =recv_char();
  switch (mBUF)
        {
          case 'U':
                {
                  com=1;
                  return (com);
                }

          case 'S':
                {
                  com=2;
                  return (com);
                }

          case 'P':
```

```
                {
                  com=3;
                  return (com);
                }

        case 'N':
                {
                  com=4;
                  return (com);
                }

        case '#':
                {
                  com=5;
                  return (com);
                }

        default : com=6;

        return (com);
      }
}

unsigned char recv_data(void)
{
  unsigned char i,mBUF;
  unsigned char ready=0;
  unsigned char j;
  int command;
  static int q=0;
  if (msBUF=='s')
    {
```

```
        equ[q]=msBUF;
        recv_messx();
        recv_messy();
        recv_messmx();
        recv_messmy();
        command=recv_command();
        if (command==5)
          {
            check_send='0';
          }
        else
          {
            ready=1;
          }
    }
else
        if (msBUF=='1')
        {
          equ[q]=msBUF;
          recv_messx();
          recv_messy();
          recv_messmx();
          recv_messmy();
          command=recv_command();
          if (command==5)
                {
                  check_send='0';
                }
          else
                {
                  ready=1;
                }
```

```
        }
    else
        {
          ready=1;
        }
  q=q+1;
  return(ready);
}

void init_timer(void)
{
  TMOD=0x01;  //timer1=mode1,timer0=mode1
  ET0=1;     //let timer0 interrupt
  TR0=1;     //start timer0
  EA=1;      //let all interrupt
}

void int_stepping(void) interrupt 1
{
  char fullx[4]={0x01,0x02,0x04,0x08};
  char fully[4]={0x10,0x20,0x40,0x80};
  static  char i=0;
  static  char j=0;
   int ii;
   int jj;
   if (ii>=abs(itimey))
     {
       if (itimey>0)
         {
           Pa=fullx[i--];
           ky=ky+1;
         }
```

```
        else
          if (itimey<0)
          {
            Pa=fullx[i++];
            ky=ky+1;
          }
        else
        if (itimey=0)
          {
            Pa=fullx[i];
          }
        if (i>3) i=0;
        if (i<0) i=3;
        ii=0;
      }
    else
      {
        ii++;
      }
    if (jj>=abs(itimex))
      {
        if (itimex>0)
          {
            Pb=fully[j--];
            kx=kx+1;
          }
        else
          if (itimex<0)
          {
            Pb=fully[j++];
            kx=kx+1;
          }
```

```
        else
          if (itimex=0)
          {
            Pb=fully[j];
          }
        if (j>3) j=0;
        if (j<0) j=3;
        jj=0;
      }
    else
      {
        jj++;
      }

    TH0=0xf7;  //8bit high timer0
    TL0=0x00;  //8bit low  timer0
}

void line( int x1,int y1,int mx ,int my)
{
    int i,j,im,jm;
    itimex=mx;
    itimey=my;
    while ((i<x1)&&(j<y1))
        {
        i=kx;
        j=ky;
        }
     itimex=0;
     itimey=0;
     kx=0;
     ky=0;
```

```
}

init_initial(void)
{
  itimex=10;
  while(inkey!=0xa0)
    {
      inkey=Pc&0xf0;
      if (inkey==0xb0)
        {
          itimex=0;
          itimey=10;
          while (inkey!=0xa0)
                {
                  inkey=Pc&0xf0;
                  if (inkey==/*0xe0*/0xa0)
                        {
                          itimey=0;
                        }
                   if (inkey==/*0xd0*/0x90)
                        {
                          itimey=-10;
                        }
                }
        }
        if (inkey==0x70)
        {
          itimex=-10;
        }
  }
  kx=0;
  ky=0;
```

```
}

void main(void)
{
  unsigned char mBUF,pout,recv_err;
  int i,j,m,t=0;
  delay(1000);
  init_timer();
  Pcon=0x88;
  init_serial();
  delay(1000);
  init_initial();
  Pc=0x0e;
  while(msBUF!='r')
    {
    msBUF=recv_char();
    recv_err=recv_data();
    if ((recv_err==0)&&(check_send=='0'))
      {
      TableCommand[m].X=atoi(messx);
      TableCommand[m].Y=atoi(messy);
      TableCommand[m].mx=atoi(messmx);
      TableCommand[m].my=atoi(messmy);
      m++;
      }
    else
      if (recv_err==1)
      {
      }

    }
  while (1)
```

```
    {
  if (equ[t]=='s')
    {
    itimex=0;
    itimey=0;
    kx=0;
    ky=0;
    }
  else
  if (equ[t]=='l')
    {
    delay(5000);
    Pc=0x0f|Pc;
    delay(1000);
    Pc=0x0e|Pc ;
    itimey=5*TableCommand[t].my;
    itimex=5*TableCommand[t].mx;
   while ((kx<TableCommand[t].X)&&(ky<TableCommand[t].Y))
      {
      }
    itimey=0;
    itimex=0;
    kx=0;
    ky=0;
    }
    delay(1000);
    t++;
    }
}
```

ประวัติผู้เขียน

1. นายประทีป กันทา

   รหัสประจำตัว 41360355

   เกิด 21 เมษายน 2522

   จบการศึกษาจาก โรงเรียนพะเยาวิทยาคม

2. นางสาวหทัยรัตน์ ธรรมาธิวัฒน์

   รหัสประจำตัว 41360538

   เกิด 4 เมษายน 2523

   จบการศึกษาจาก โรงเรียนหล่มสักวิทยาคม

มหาวิทยาลัยนเรศวร