หุ่นยนต์รับ-ส่งเอกสารในออฟฟิศ

OFFICE HELPER ROBOT

นาย กิตินันต์ อยู่เสนาสน์ รหัส 41360223

นาย อธิปัตย์ ล้อวงศ์งาม รหัส 41360546

นาย เริงชัย คงมี รหัส 41360595

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 30 พ.ย. 2544
เลขทะเบียน กศ 440605
เลขเรียกหนังสือ ว11 ก676ห
มหาวิทยาลัยนเรศวร

ปร.
ก676ห
2544
ค.2

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2544

ใบรับรองโครงงานวิจัย

หัวข้อโครงงาน หุ่นยนต์รับ-ส่งเอกสารในออฟฟิศ

ผู้เสนอโครงงาน

นาย กิตินันต์	อยู่เสนาสน์	41360223
นาย อธิปัตย์	ล้อวงศ์งาม	41360546
นาย เริงชัย	คงมี	41360595

อาจารย์ที่ปรึกษา อาจารย์ ยงยุทธ ชนบดีเฉลิมรุ่ง

อาจารย์ที่ปรึกษาร่วม อาจารย์ ศิริพร เดชะศิลารักษ์

สาขา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2543

.....

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิจัย

.....ประธานกรรมการ
(อาจารย์สิทธิโชค เชาวกุล)

.....กรรมการ
(อาจารย์ประทีป ตรีรณโอภาส)

.....กรรมการ
(อาจารย์วัชรวีร์ พืชพันธ์)

.....กรรมการ
(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงงาน	หุ่นยนต์รับ-ส่งเอกสารในออฟฟิศ
ผู้เสนอโครงงาน	นาย กิตินันต์ อยู่เสนาสน์ 41360223
	นาย อธิปัตย์ ล้อวงศ์งาม 41360546
	นาย เริงชัย คงมี 41360595
อาจารย์ที่ปรึกษา	อาจารย์ ยงยุทธ ชนบดีเฉลิมรุ่ง
อาจารย์ที่ปรึกษาร่วม	อาจารย์ ศิริพร เดชะศิลารักษ์
สาขา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2543

.....

บทคัดย่อ

โครงงานนี้เป็นการศึกษาและพัฒนาหุ่นยนต์ให้สามารถจัดการกับเอกสาร โดยการรับ-ส่งเอกสารที่ต้องการส่งไปยังโต๊ะทำงานที่ถูกต้อง การเคลื่อนไหวของหุ่นจะเป็นการเดินตามเส้น สำหรับโปรแกรมควบคุมหุ่นให้เคลื่อนไหวตามที่เราต้องการ นั้นจะใช้ภาษา c ในการควบคุม และหุ่นจะเลือกเส้นทางที่สั้นที่สุดในการนำเอกสารไปส่ง สำหรับโครงสร้างของหุ่นนั้นจะต้องทำให้สามารถที่จะใช้ประโยชน์จากมอเตอร์ให้มากที่สุด เพื่อผลประโยชน์ในการรับน้ำหนักบรรทุกหนังสือให้ได้มากที่สุดเท่าที่มอเตอร์สามารถทำงานได้

ผลที่คาดว่าจะได้รับจากการทำโครงงานนี้ คือ หุ่นที่สามารถจะแบ่งเบาภาระการทำงานของมนุษย์ได้ในการจัดการส่งเอกสารภายในออฟฟิศได้ และเป็นต้นแบบให้ผู้ที่สนใจสามารถพัฒนาขีดความสามารถของหุ่นยนต์ต่อไปได้

Project Title	Office Helper Robot	
Name	Mr. Kittinan Yousaenart	ID. 41360223
	Mr. Atipat Lorwong-ngam	ID. 41360546
	Mr. ReaungChai Kongme	ID. 41360595
Project Advisor	Mr. Yongyut Chanabodeechalermrung	
Co-Project Advisor	Miss Siriporn Dachasilaruk	
Major	Computer Engineering	
Department	Electrical and Computer Engineering	
Academic Year	2001	

Abstract

This project is to study and to develop robot so that the robot could handle paper by taking and placing them at a right desk! The processes begin with the robot get a request signal from a user. Then the robot will compare the signal with its Database, which desk to go to, then the robot will go to that desk. The user, who calls the robot, puts paper on the robot and tells the robot where to place the paper, by using the panel on the robot. The robot takes the paper to the right desk. The robot will move by following a line. We would use C Language Programming to program this robot. For searching algorithm we would use Best-First Search Algorithm to search for the shortest path. And for the robot construction, we need to design it to use the motor as much as we could, so the robot could take as many books as the motor could.

The result of this project is that this robot will relieve human works and save time on taking paper from place to place. And the robot will be the original one for others, who want to develop this type of robots later on.

กิตติกรรมประกาศ

การจัดทำโครงงานวิศวกรรมศาสตร์ครั้งนี้สำเร็จลุล่วงด้วยดี เนื่องมาจากการแนะนำและความช่วยเหลือจากอาจารย์ยงยุทธ ชนบดีเฉลิมรุ่ง และอาจารย์ศิริพร เดชะศิลารักษ์ ที่ได้ให้คำปรึกษาชี้แนวทางที่เป็นประโยชน์อย่างสูงในการทำโครงงานครั้งนี้และขอขอบใจเพื่อนๆ สำหรับความช่วยเหลือในด้านต่างๆ

นายกิตินันต์ อยู่เสนาสน์
นายอธิปัตย์ ล้อวงศ์งาม
นายเริงชัย คงมี

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา..... 1

1.2 วัตถุประสงค์ของโครงงาน..... 1

1.3 ขอบข่ายของโครงงาน..... 2

1.4 ขั้นตอนการดำเนินโครงงาน..... 2

1.5 ผลที่คาดว่าจะได้รับ..... 3

1.6 รายละเอียดงบประมาณตลอดโครงงาน.....3

บทที่ 2 หลักการและคณิตศาสตร์ที่เกี่ยวข้องกับการควบคุม

2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051 4

2.1.1 คุณลักษณะพื้นฐานของ 8051 4

2.1.2 การสื่อสารข้อมูลอนุกรม 5

2.1.3 การจัดข้อมูลอนุกรมของ 8051 6

2.1.4 การอินเตอร์รัปต์ของการสื่อสารอนุกรม 8

2.1.5 กระบวนการรับและส่งของมูลอนุกรมของ 8051 9

2.1.6 พอร์ตอนุกรม โหมด 0 9

2.2 การควบคุมจากระยะไกล 11

2.2.1 CP-SB31 (SINGLE BOARD 31 ON PC) 11

2.2.2 คุณลักษณะพิเศษของ CP-SB31 12

2.3 วงจร Supply 12

2.4 วงจร Sensor 13

2.5 Interface Board 14

สารบัญ (ต่อ)

หน้า

2.6 ไดรเวอร์มอเตอร์ 14

2.6.1 อุปกรณ์อิเลคทรอนิกที่ใช้ 16

2.6.2 การทำงานของบอร์ดไดรเวอร์ 17

บทที่ 3 การออกแบบและสร้างหุ่นยนต์

3.1 ศึกษาและออกแบบวงจรฮาร์ดแวร์ 21

3.2 โปรแกรมควบคุมการทำงานของหุ่นยนต์ 22

3.2.1 รูปแบบของโปรแกรมควบคุม 22

3.2.2 การค้นหาเส้นทาง 23

3.3 โครงสร้างของหุ่นเมื่อสร้างสำเร็จ 24

3.3.1 การนำเซ็นเซอร์มาใช้งาน 24

บทที่ 4 ผลการทดลองและผลการวิเคราะห์

4.1 จุดประสงค์ของการทดลอง 25

4.2 ขั้นตอนการทดลอง

4.2.1 การทดสอบประสิทธิภาพ ดีซีมอเตอร์ควบคู่กับไดรเวอร์มอเตอร์ PWM 25

4.2.2 การทดสอบประสิทธิภาพการทำงานของ เซนเซอร์ 25

4.2.3 การทดสอบการทำงานของโปรแกรม 26

4.3 ผลการทดลอง

4.3.1 ผลการทดสอบประสิทธิภาพ ดีซีมอเตอร์ควบคู่กับไดรเวอร์มอเตอร์ PWM ... 26

4.3.2 ผลการทดสอบประสิทธิภาพการทำงานของ เซนเซอร์ 27

4.3.3 ผลการทดสอบการทำงานของโปรแกรม 27

4.4 ผลการวิเคราะห์

4.4.1 ผลการวิเคราะห์จากการทดสอบประสิทธิภาพ ดีซีมอเตอร์ควบคู่กับ ไดรเวอร์มอเตอร์ PWM 27

4.4.2 ผลการวิเคราะห์จากการทดสอบประสิทธิภาพการทำงานของ เซนเซอร์ ... 28

4.4.3 ผลการวิเคราะห์จากการทดสอบการทำงานของโปรแกรม 28

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุป

5.1 สรุปผลการทดลอง 29

5.2 ปัญหาในการทดลองและแนวทางแก้ไข 29

5.3 ปัญหาที่พบในการทำงานของหุ่นและแนวทางแก้ไข 30

5.4 แนวทางในการพัฒนาหุ่นยนต์รับส่งเอกสารในออฟฟิศต่อไป 30

เอกสารอ้างอิง 31

ภาคผนวก ก 32

ภาคผนวก ข 39

ประวัติผู้เขียน 44

สารบัญตาราง

ตารางที่		หน้า
1.1	การศึกษา	2
1.2	การปฏิบัติงาน	3
2.1	โหมดการทำงานของพอร์ตอนุกรม	7
2.2	รีจิสเตอร์ควบคุมการทำงานและบอกสถานะการสื่อสารข้อมูลอนุกรม SCON	8
2.3	FUNCTION TABLE ของ 74LS123	18
4.1	ผลการทดลองหาค่าความต่างศักย์ที่ได้จากดีซีมอเตอร์เพื่อกำหนดค่า duty cycle ค่าต่าง ๆ	26
4.2	ผลการทดลองการใส่สีดำและสีขาวแล้ววัดค่าความต่างศักย์ที่ได้จากเซนเซอร์	27
4.3	ผลการทดลองเมื่อกำหนดค่า x, y และค่าต่าง ๆ	27

สารบัญรูป

รูปที่		หน้า
2.1	แผนภาพบล็อกแสดงหน่อยทำงานพื้นฐานของ MCS-51	5
2.2(a)	ข้อมูลสื่อสารแบบอนุกรม	6
2.2(b)	ข้อมูลสื่อสารแบบขนาน	6
2.3	แผนภาพแสดงการทำงานของวงจรส่วนการรับและส่งข้อมูลอนุกรมของ 8051	7
2.4	แผนภาพแสดงเวลาของสัญญาณอนุกรมโหมด 0	10
2.5	วงจร Supply	12
2.6	วงจร Sensor	13
2.7	วงจร Interface	14
2.8	วงจร Driver Motor	15
3.1	แผนผังการทำงานของวงจรทั้งหมด	21
3.2	แผนผังลำดับการทำงานของโปรแกรม	22
3.3	สถานที่จำลอง	23
3.4	ตัวอย่างพิกัดของโต๊ะ	23
3.5	ภาพของหุ่นยนต์ที่สร้างเสร็จแล้ว	24

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการทำงานในสำนักงานทั่วไป ในแผนกหนึ่งๆจะมีเจ้าหน้าที่หลายคนทำหน้าที่แตกต่างกันไป การทำงานในแต่ละวันแต่ละคนก็ต้องทำงานตามหน้าที่ที่ได้รับมอบหมาย และก็จะต้องมีการติดต่อสื่อสารระหว่างกัน เช่น การส่งเอกสาร แฟ้มรายงาน หรืออุปกรณ์ต่างๆ การที่จะเดินไปส่งถึงยังผู้รับก็อาจทำให้เสียเวลาในการทำงานได้ โดยเฉพาะอย่างยิ่งการทำงานในปัจจุบันจะต้องทำงานแข่งกับเวลาด้วย ทำให้อาจจะต้องมีการจ้างคนเพิ่มเพื่อทำหน้าที่นี้ แต่ก็จะทำให้บริษัทรต้องเสียเงินเพิ่มขึ้น ทางคณะผู้เสนอโครงงานจึงขอเสนอหุ่นยนต์เพื่อทำหน้าที่ส่งเอกสาร แฟ้ม หรือ อุปกรณ์อื่นๆ ซึ่งจะทำงานแทนคนได้ เทคโนโลยีหุ่นยนต์นี้ในปัจจุบันได้มีการพัฒนาไปมากกว่าในอดีตมากซึ่งจะทำให้พนักงานมีเวลาในการทำงานเพิ่มมากขึ้น หุ่นยนต์นี้จะทำหน้าที่ส่งเอกสารระหว่างโต๊ะทำงาน โดยจะเป็นการเดินตามเส้นและจะมีการเลือกเส้นทางที่ดีที่สุด เพื่อประหยัดเวลาในการส่งเอกสาร โดยการสั่งให้หุ่นยนต์ทำงานนี้ จะใช้ระบบคีย์บอร์ดในการสั่งให้หุ่นยนต์ทำงานผู้ส่งและผู้รับเอกสารจะอยู่ที่โต๊ะทำงาน

ทางคณะผู้เสนอโครงงานจึงมีความประสงค์ที่จะพัฒนาหุ่นยนต์ส่งเอกสาร ที่จะสามารถส่งเอกสารไปถึงผู้รับที่เป็นเป้าหมายปลายทาง โดยที่เอกสารไม่สูญหายไประหว่างทาง และพัฒนาให้หุ่นยนต์มีความสามารถมากขึ้น เพื่อเป็นต้นแบบให้ผู้ที่สนใจนำไปพัฒนาต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อให้หุ่นยนต์ช่วยงานแทนมนุษย์ได้จริง
2. เพื่อศึกษาการใช้ ระบบคีย์บอร์ด
3. เพื่อศึกษาการติดต่อระบบฐานข้อมูล เพื่อกำหนดเส้นทางของหุ่นยนต์
4. เพื่อศึกษาเกี่ยวกับการออกแบบโปรแกรมที่จะทำให้หุ่นยนต์สามารถคิดเองได้ในการที่จะ
   - เลือกเส้นทางที่ดีที่สุดที่ผู้รับและผู้ส่งเอกสารจะสามารถติดต่อสื่อสารกันได้
   - เพื่อศึกษาเกี่ยวกับโครงสร้างที่จะทำให้หุ่นยนต์อำนวยความสะดวกได้มากที่สุด

1.3 ขอบข่ายของโครงงาน

1. ศึกษาการใช้คีย์บอร์ดในการควบคุมหุ่นยนต์
2. ศึกษาการนำข้อมูลที่จำเป็นจากฐานข้อมูลเพื่อระบุเส้นทางของหุ่นยนต์
3. ศึกษาทางด้าน ซอฟต์แวร์ ในการควบคุมหุ่นยนต์
4. ศึกษาทางด้าน ฮาร์ดแวร์ ที่ใช้ในหุ่นยนต์
5. ออกแบบโครงสร้าง
6. ออกแบบวงจรควบคุม
7. ออกแบบ ซอฟต์แวร์
8. ทำการพัฒนาหุ่นยนต์
9. ทำการทดลองใช้งาน
10. วิเคราะห์ และ สรุปผลการทดลอง

1.4 ขั้นตอนการดำเนินโครงงาน

1.4.1 การศึกษา

ตารางที่ 1.1 การศึกษา

กิจกรรม	เดือน - ปี		
	ม.ค. 44	ก.พ. 44	มี.ค. 44
ศึกษาด้านโครงสร้าง	←	→	
ศึกษาด้าน ฮาร์ดแวร์		←	→
ศึกษาด้าน ซอฟต์แวร์			← →

1.4.2 การปฏิบัติงาน

ตารางที่ 1.2 การปฏิบัติงาน

กิจกรรม	เดือน – ปี							
	ก.พ. 44	มี.ค. 44	เม.ย. 44	พ.ค. 44	มิ.ย. 44	ก.ค. 44	ส.ค. 44	ก.ย. 44
1.หาข้อมูล	←	→						
2.ออกแบบ		←	→					
3.พัฒนา			←	→				
4.ทดสอบ						←	→	
5.สรุปผล								↔

1.5 ผลที่คาดว่าจะได้รับ

1. ความรู้ความเข้าใจเกี่ยวกับ
   1.1 ระบบคีย์บอร์ด
   1.2 การจัดเก็บฐานข้อมูล และการดึงข้อมูลมาใช้
   1.3 การออกแบบ ที่ทำให้หุ่นยนต์สามารถคิดได้
2. หุ่นยนต์สามารถทำงานแทนคนได้จริง
3. เพื่อให้ผู้ที่สนใจศึกษาและทำการพัฒนาต่อไป

1.6 รายละเอียดงบประมาณตลอดโครงงาน

ค่ากระดาษรายงาน	75	บาท
ค่ารถเพื่อไปซื้ออุปกรณ์	500	บาท
ค่าอุปกรณ์เพื่อใช้ในวงจร	1,525	บาท
ค่าอุปกรณ์ทำตัวหุ่นยนต์	900	บาท
รวมเป็นเงินทั้งสิ้น	**3,000**	**บาท**

(สามพันบาทถ้วน)

บทที่ 2
หลักการและคณิตศาสตร์ที่เกี่ยวข้องกับการควบคุม

2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051 [2]

บรรดาไมโครคอนโทรเลอร์ที่มีการผลิตจากบริษัทต่าง ๆ จำนวนมากนั้น ไมโครคอนโทรเลอร์จากบริษัท อินเทล (Intel Corporation) ในตระกูล MCS-51 ได้มีการนำไปใช้งานกันแพร่หลายมากนับตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา ในระยะที่ผ่านมาก็ได้มีอีกหลายบริษัท เช่น บริษัท Phillips และ Siemens เป็นต้น ได้รับสิทธิ์ในการผลิตจำหน่าย และได้มีการเพิ่มประสิทธิภาพหน่วยการทำงานต่าง ๆ มากขึ้น ทำให้ปัจจุบันมีไมโครคอนโทรเลอร์จากผู้ผลิตต่าง ๆ ที่มีพื้นฐานมาจากไมโครคอนโทรเลอร์ MCS-51 ของบริษัทอินเทลอยู่เป็นจำนวนมาก

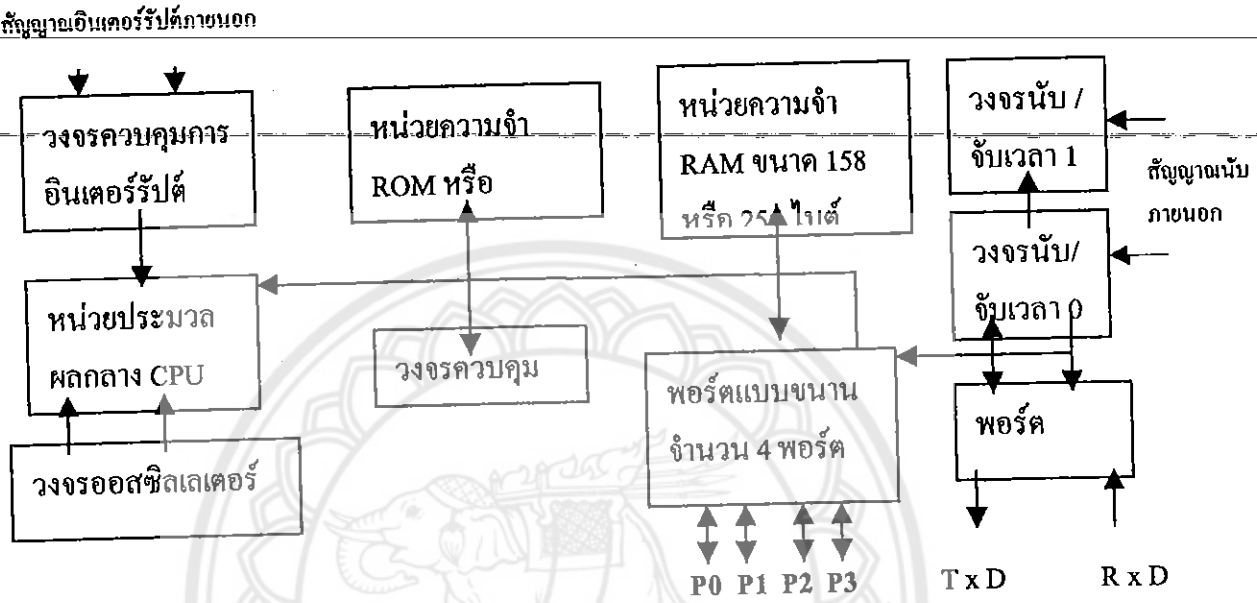
ไมโครคอนโทรเลอร์ตระกูล MCS-51 ประกอบด้วย ไมโครคอนโทรเลอร์หลายรุ่น (version) ซึ่งมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหรือจำนวนของหน่วยทำงานภายในที่แตกต่างกันออกไป เพื่อ ความเหมาะสมในงานประยุกต์ต่าง ๆ ตามความต้องการ ดังแสดงให้เห็นในตารางของรูปที่ 2.1 โดยมีทั้งลักษณะที่ใช้เทคโนโลยีการผลิตไอซีวงจรรวมความจุสูงมาก (LSI) แบบ HMOS หรือ CHMOS ซึ่งมีคุณลักษณะที่สูงมากขึ้นและสิ้นเปลืองกำลังไฟฟ้าน้อยกว่ามาก อย่างไรก็ตามการอ้างถึงไมโครคอนโทรลเลอร์ MCS-51 ในที่นี้จะเรียกรวมกันว่า 8051 แทน

2.1.1 คุณลักษณะพื้นฐานของ 8051 [2]

จากแผนภาพในรูปที่ 2.1 แสดงให้เห็นถึงหน่วยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ที่จัดอยู่ภายในตระกูล MCS-51 นี้ประกอบด้วย

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบอีพรอม (EPROM) เบอร์ 8751 หรือแบบรอม (ROM) เบอร์ 8051
- พอร์ตอินพุต/เอาท์พุตแบบขนานจำนวน 32 เส้นซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวน 2 วงจร
- วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)

- วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้สองระดับ
- วงจรออสซิลเลเตอร์ภายใน



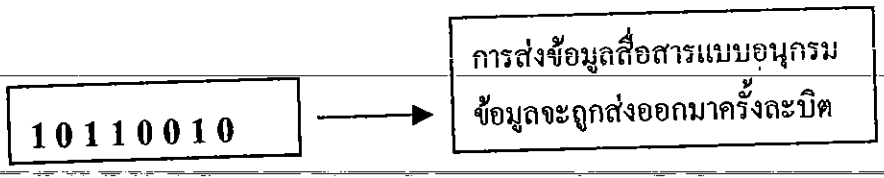
รูปที่ 2.1 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของ MCS-51

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา ดังแผนภาพในรูปที่ 2.2 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า Alternate Functions ในรูปที่ 2.2) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่น ขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อ P3.0) อาจจะใช้เป็นสัญญาณขาเอาท์พุต/อินพุต ตามปกติ หรืออาจจะทำหน้าที่เป็นสัญญาณขาอินพุตของข้อมูลสื่อสารแบบอนุกรม (R x D) ให้กับวงจรสื่อสารแบบอนุกรมของ 8051 ได้ ซึ่งการกำหนดว่าจะทำงานในลักษณะใดก็ขึ้นอยู่กับการเชื่อมต่อวงจรเข้ากับขาสัญญาณและโปรแกรมควบคุมของระบบนั้น

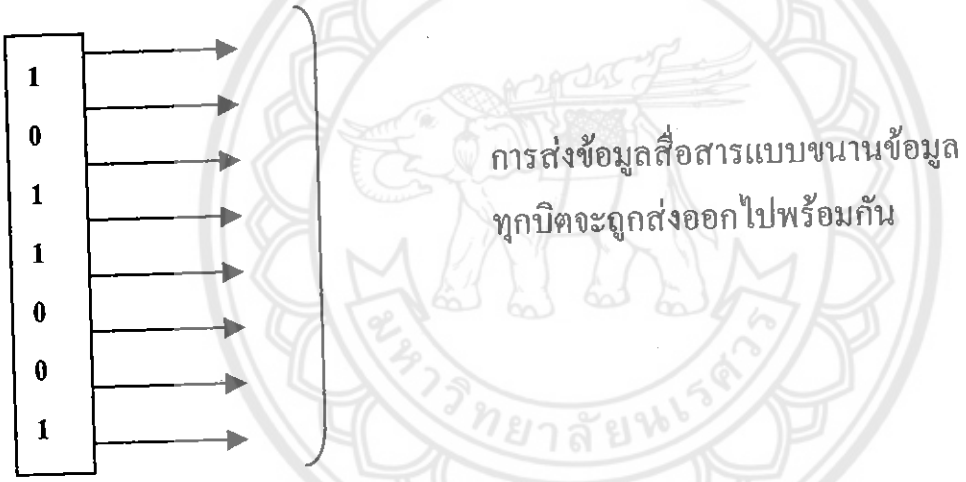
2.1.2 การสื่อสารข้อมูลอนุกรม [2]

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะของบิต คราวละหนึ่งบิตเป็นลำดับเป็นลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมากเนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกันจึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วยในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียง

สองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกล เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก ลองพิจารณาเปรียบเทียบการสื่อสารทั้งสองประเภทได้จากรูปที่ 2.2(a) และ 2.2(b)



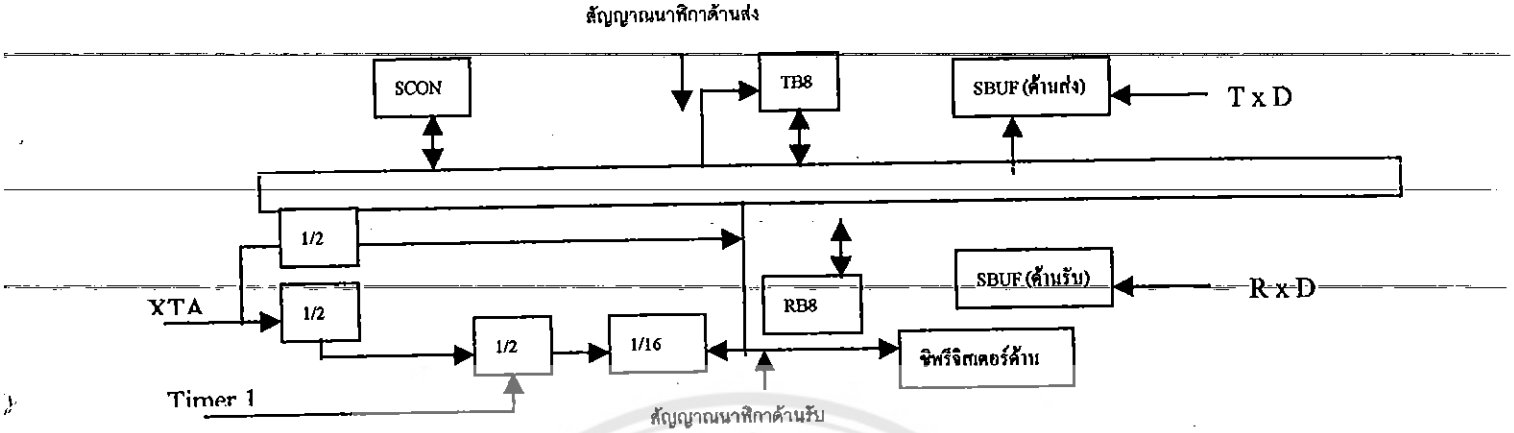
รูปที่ 2.2 (a) ข้อมูลสื่อสารแบบอนุกรม ข้อมูลหนึ่งไบต์จะถูกส่งออกคราวละบิตเป็นลำดับจนครบทั้ง 8 บิต



รูปที่ 2.2 (b) ข้อมูลสื่อสารแบบขนาน ข้อมูลแต่ละบิตภายในหนึ่งไบต์จะถูกส่งออกไปพร้อมกัน

2.1.3 การจัดข้อมูลอนุกรมของ 8051 [2]

พอร์ตอนุกรมของ 8051 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ซึ่งหมายถึงความสามารถในการรับและส่งข้อมูลอนุกรมในเวลาเดียวกัน จากรูปที่ 2.3 แสดงให้เห็นถึงการทำงานอย่างง่ายของวงจรส่วนจัดการข้อมูลอนุกรมของ 8051 โดยทางด้วนวงจรตัวส่ง ประกอบด้วยรีจิสเตอร์ SBUF ทำหน้าที่เก็บข้อมูลที่จะส่งออก การใช้คำสั่งเขียนหรือโอนย้ายข้อมูลมายังรีจิสเตอร์นี้ จะเป็นการส่งข้อมูลนั้นออกไปยังพอร์ตอนุกรมทางขาสัญญาณ T x D โดยอัตโนมัติ ส่วนวงจรด้านตัวรับ ประกอบด้วยรีจิสเตอร์ SBUF เช่นเดียวกัน แต่ทำหน้าที่เก็บข้อมูลที่นำมาจากส่วนของวงจรเลื่อนบิตหรือชิฟรีจิสเตอร์ของวงจรจัดการข้อมูลอนุกรมภายในสัญญาณข้อมูลอนุกรมที่รับเข้าจะผ่านมาทางขาสัญญาณ R x D



รูปที่ 2.3 แผนภาพแสดงการทำงานของวงจรส่วนการรับและส่งข้อมูลอนุกรมของ 8051

พอร์ตอนุกรมของ 8051 สามารถโปรแกรมให้ทำหน้าที่ในรูปแบบต่าง ๆ กันสี่แบบ โดยการกำหนดค่าบิต SM0 และ SM1 ซึ่งอยู่ภายในรีจิสเตอร์ควบคุมและบอกสถานะ SCON ดังตารางที่ 2.2 โหมดการทำงานทั้ง 4 แบบของพอร์ตอนุกรมมีดังนี้

ตารางที่ 2.1 โหมดการทำงานของพอร์ตอนุกรม

โหมดการทำงาน	คำอธิบาย
• โหมด 0	• เป็นการขยายพอร์ตอินพุต/เอาท์พุต โดยทำงานร่วมกับไอซีชิพรีจิสเตอร์ภายนอก ประเภททีทีแอลหรือซีมอส
• โหมด 1	• ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal asynchronous receiver / Transmitter) โดยการใช้กลุ่มข้อมูลแบบ 10 บิต และสามารถเปลี่ยนแปลงอัตราเร็วในการส่งข้อมูลได้
• โหมด 2	• ใช้สำหรับการเชื่อมต่อแบบอนุกรมแบบ UART โดยการใช้กลุ่มของข้อมูลแบบ 11 บิตและให้อัตราเร็วในการส่งข้อมูลคงที่
• โหมด 3	• ใช้สำหรับต่อเชื่อมอนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิตและสามารถเปลี่ยนแปลงอัตราเร็วในการส่งข้อมูลได้

นอกจากนี้โหมด 2 และโหมด 3 ยังมีการดำเนินการแบบพิเศษออกไป โดยสามารถนำมาใช้ประโยชน์ในการสื่อสารข้อมูลแบบที่มีในตัวไมโครโปรเซสเซอร์หลายตัวทำงานร่วมกันได้ ซึ่งจะได้อธิบายรายละเอียดเป็นลำดับไป

จากแผนภาพตารางที่ 2.2 ชิพรีจิสเตอร์ภายในตัวส่งจะทำหน้าที่เลื่อนบิตข้อมูลออกไปภายนอกโดยไม่มีการบัฟเฟอร์ และเมื่อใดที่มีการเขียนข้อมูลให้กับรีจิสเตอร์ SBUF แสดงว่ามีความต้องการที่จะส่งข้อมูลนี้ออกไปแบบอนุกรม สำหนับชิพรีจิสเตอร์ทางด้านรับจะทำการเลื่อนบิตข้อมูลนี้ออกไปแบบอนุกรม สำหนับชิพรีจิสเตอร์ทางด้านรับจะทำการเลื่อนบิตข้อมูลที่รับเข้ามาเก็บไว้ เมื่อบิตของข้อมูลที่รับเข้ามาครบถ้วนตามจำนวนที่กำหนดไว้ตามลักษณะโหมดการทำงานต่าง ๆ แล้ว จะถูกย้ายไปเก็บยังรีจิสเตอร์ SBUF ต่อไป อย่างไรก็ตามการย้ายข้อมูลนี้จะเกิดขึ้นก็ต่อเมื่อรีจิสเตอร์ SBUF นั้นไม่มีข้อมูลที่จะทำการส่งหรือได้ส่งข้อมูลภายนอกออกไปเสร็จสิ้นแล้ว

ตารางที่ 2.2 รีจิสเตอร์ควบคุมการทำงานและบอกสถานะการสื่อสารข้อมูลอนุกรม SCON

ชื่อบิต	ตำแหน่ง	ความหมาย
SM0	SCON.7	บิตเลือกโหมดการทำงาน
SM1	SCON.6	บิตเลือกโหมดการทำงาน
SM2	SCON.5	แฟล็กกำหนดการทำงานแบบมัลติโปรเซสเซอร์
REN	SCON.4	แฟล็กยอมให้มีการรับข้อมูล
TB8	SCON.3	ค่าของบิตที่ 9 สำหรับการส่งข้อมูลออก
RB8	SCON.2	ค่าของบิตที่ 9 ของข้อมูลที่รับเข้า
T1	SCON.1	แฟลกซ์แสดงการอินเตอร์รัปต์ภายหลังการส่งข้อมูล
RI	SCON.0	แฟลกซ์แสดงการอินเตอร์รัปต์เมื่อมีข้อมูลรับเข้า

2.1.4 การอินเตอร์รัปต์ของการสื่อสารอนุกรม

เนื่องจากการส่งหรือรับข้อมูลอนุกรมในการส่งข้อมูลไบต์หนึ่งๆ ค่อนข้างจะใช้เวลานานหลายมิลิวินาที ดังนั้นเพื่อการจัดการเกี่ยวกับการสื่อสารแบบนี้เป็นไปอย่างมีประสิทธิภาพ 8051 จึงได้กำหนดให้มีบิตหรือแฟลกซ์สถานะที่เกี่ยวข้องทั้งหมดจัดรวมอยู่ภายในรีจิสเตอร์ SCON เท่านั้น เช่น แฟลกซ์ TI ซึ่งมีค่าเป็นหนึ่ง เพื่อแจ้งให้ทราบว่าได้รับข้อมูลเข้ามาทางพอร์ตอนุกรม เมื่อแฟลกซ์ตัวใดตัวหนึ่งนี้มีค่าเป็นหนึ่ง จะมีผลทำให้เกิดการอินเตอร์รัปต์ขึ้น ดังนั้นภายในโปรแกรมจะต้องทำการตรวจสอบจากสภาวะของแฟลกซ์นั้น ลักษณะดังกล่าวนี้จะมีความแตกต่างไปจากการอินเตอร์รัปต์จากสัญญาณอื่น ๆ เช่น วงจรนับ วงจรจับเวลา เป็นต้น ซึ่งจะมีการกำหนดค่าศูนย์ ให้

กับแฟลกซ์สถานะที่เกี่ยวข้องโดยอัตโนมัติ ภายหลังจากที่ได้เข้าไปทำงานยังส่วนของโปรแกรมย่อยบริการอินเตอร์รัปต์ดังนั้นจึงขอให้สังเกตความแตกต่างในส่วนนี้ไว้ด้วยล

2.1.5 กระบวนการรับและส่งข้อมูลอนุกรมของ 8051 [1, 2]

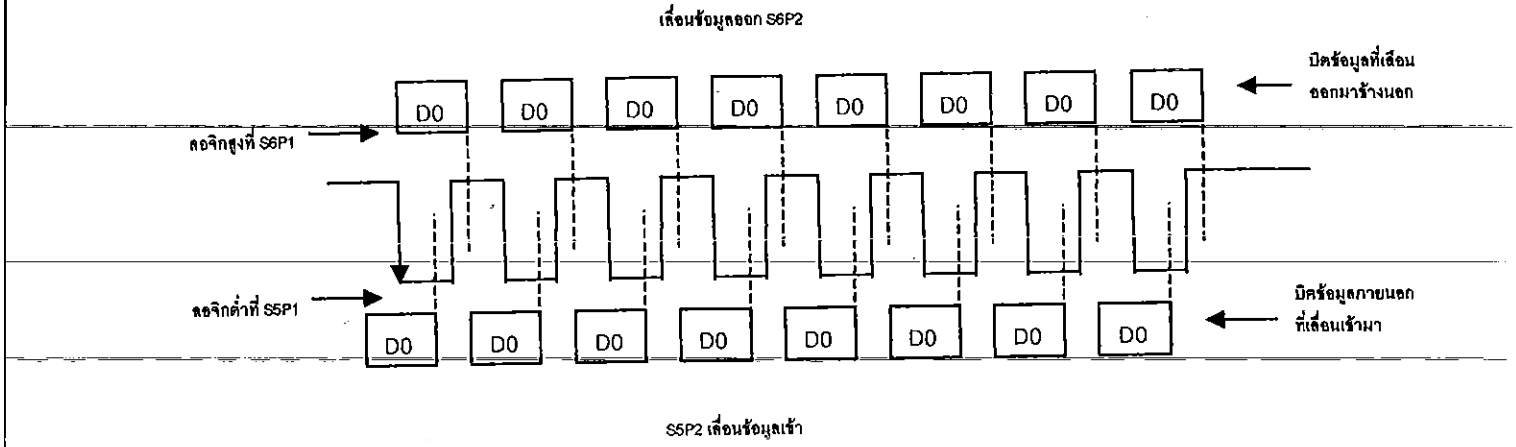
การส่งข้อมูลออกทางพอร์ตอนุกรมของ 8051 จะเริ่มต้นขึ้น ภายหลังเมื่อมีการเขียนข้อมูลลงในรีจิสเตอร์ SBUF ข้อมูลนี้จะถูกจัดการด้วยวิธีการด้านฮาร์ดแวร์ในการเลื่อนบิตและการส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่าได้ส่งออกครบถ้วนแล้วจึงทำการกำหนดค่าของแฟลกซ์ TI ให้เป็นหนึ่งเพื่อแจ้งให้ทราบว่า ขณะนั้นรีจิสเตอร์ SBUF ว่าง และพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่ต้องรอให้แฟลกซ์ TI มีค่าเป็นหนึ่งก่อนจะมีผลทำให้ข้อมูลที่ส่งออกไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่าบิ REN (Receiver Enable) ให้มีค่าเป็นหนึ่งก่อน หลังจากนั้นเมื่อมีบิตของข้อมูลถูกส่งเข้ามาจากภายนอกระบบฮาร์ดแวร์ของ 8051 จึงจะทำให้การเลื่อนบิตเหล่านี้เข้ามาโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนเข้าเรียบร้อยแล้วข้อมูลเหล่านั้นจะถูกย้ายมาเก็บยังรีจิสเตอร์ SBUF และทำการกำหนดให้แฟลกซ์ RI ให้มีค่าเป็นหนึ่ง ซึ่งมีผลทำให้เกิดการอินเตอร์รัปต์โปรแกรมขึ้น

2.1.6 พอร์ตอนุกรมโหมด 0

การทำงานของพอร์ตอนุกรมในโหมด 0 เป็นการขยายพอร์ตอินพุตหรือเอาต์พุตให้มีจำนวนมากขึ้น โดยจะทำการสร้างสัญญาณนาฬิกาขึ้นเพื่อให้จังหวะของการทำงานที่พร้อมกันสำหรับการเลื่อนบิตเข้าหรือออกจากไอซีรีจิสเตอร์ภายนอก เมื่อมีการโอนย้ายข้อมูลมายังรีจิสเตอร์ในแต่ละครั้งจะมีผลทำให้เกิดการส่งบิตข้อมูลทั้ง 8 บิตออกมาแม้ว่าแฟลกซ์ สถานะ TI ยังคงมีค่าเป็นหนึ่งอยู่ก็ตาม นอกจากนี้แล้วเมื่อใดก็ตามค่าของแฟลกซ์สถานะ RI ให้เป็นศูนย์เพื่อรับข้อมูลต่อไป

การทำงานของพอร์ตอนุกรมในโหมดศูนย์ เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิต โดยเพียงสัญญาณ R x D เท่านั้น ส่วนขาสัญญาณ T x D จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะการเลื่อนข้อมูลกับวงจรเลื่อนบิตภายนอก สำหรับอัตราการเลื่อนบิต จะถูกกำหนดไว้ที่ค่าคงที่ที่ค่า 1/12 ของความถี่ออสซิลเลเตอร์ จากรูปที่ 2.4 แสดงให้เห็นถึงแผนภาพเวลาสัญญาณต่างๆ ในโหมดศูนย์เมื่อมีการรับและส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกา โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดขึ้นในตัวของ 8051 เอง และมีจุดประสงค์เพื่อนำไปใช้ในวงจรชิพรีจิสเตอร์ภายนอกเท่านั้น



รูปที่ 2.4 แผนภาพแสดงเวลาของสัญญาณอนุกรมโหมดศูนย์

สัญญาณนาฬิกาที่สร้างขึ้นทางขาสัญญาณ T x D นี้จะสลับค่าไปมาจากระดับลอจิกสูงไปต่ำในราวช่วงใกล้เคียงกับเวลาขอบขาลงของสัญญาณ ALE ซึ่งอยู่ในคาบเวลาออสซิลเลเตอร์ ที่ 15 ภายหลังจากที่ได้ทำคำสั่งการโอนย้ายข้อมูลมายังรีจิสเตอร์ SBUF หรือคำสั่งที่ทำให้ค่าแฟลกซ์สถานะ RI เป็นศูนย์ หลังจากนั้นสัญญาณนาฬิกาก็จะเปลี่ยนแปลงอีกครั้งราวช่วงใกล้เคียงกับเวลาขอบขาลงของสัญญาณ ALE ในคาบเวลาออสซิลเลเตอร์หลังจากนั้นอีก 6 คาบ และจะดำเนินไปในลักษณะอย่างนี้จนกระทั่งข้อมูลทั้ง 8 บิต ได้ถูกส่งออกไปเรียบร้อยแล้ว เมื่อสัญญาณขอบขาขึ้นของสัญญาณนาฬิกานี้ขึ้นครบจำนวน 8 ครั้งแล้ว จึงจะมีผลทำให้แฟลกซ์สถานะ TI หรือ RI มีค่าเป็นหนึ่งขึ้นและสถานะของสัญญาณ T x D ก็จะเป็นระดับลอจิกสูงไปตลอด

ข้อมูลที่จะถูกส่งออกไปภายนอกจะถูกเลื่อนบิตนัยสำคัญต่ำออกไปก่อนเป็นอันดับแรก โดยจะเริ่มขึ้นในเวลาเริ่มต้นของคาบเวลาออสซิลเลเตอร์ ภายหลังจากที่ได้ทำคำสั่งการโอนย้ายข้อมูลมายังรีจิสเตอร์ SBUF สำหรับบิตแรกของข้อมูลที่รับเข้ามานั้นจะถูกแลตช์ไว้ด้วยขอบขาขึ้นของสัญญาณนาฬิกาในคาบเวลาออสซิลเลเตอร์ที่ 24 ภายหลังจากที่ได้มีการกำหนดให้แฟลกซ์สถานะ RI เป็นศูนย์ หลังจากนั้นในคาบเวลาออสซิลเลเตอร์อีก 12 คาบถัดมาก็จะได้รับบิตต่อไปซึ่งจะดำเนินการในลักษณะเช่นนี้จนกระทั่งได้จำนวนบิตข้อมูลครบทั้ง 8 บิต

2.2 การส่งสัญญาณโดยผ่านคีย์บอร์ด [2]

จะเป็นการใช้ระบบคีย์บอร์ดคอนโทรลระบบผ่านสายอนุกรมผ่านทางบอร์ด 8031 ระบบคีย์บอร์ดคอนโทรลนี้จะประกอบไปด้วยตัวส่งสัญญาณผ่านสายอนุกรม และตัวรับสัญญาณอนุกรม เนื่องจากจะเป็นการสะดวกเพื่อใช้ในการพัฒนาให้สามารถมีการควบคุมจากระยะไกลได้ต่อไป

2.2.1 CP-SB31 (SINGLE BOARD 31 ON PC)

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

2.2.1.1 คุณลักษณะพิเศษของ CP-SB312.4 CP-SB31 (SINGLE BOARD 31 ON PC)

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

2.2.1.2 ลักษณะของบอร์ด CP-SB31

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

2.2.1.3 หน่วยความจำ (Memory)

· มี SOCKET ขนาด 28 PIN 2 ตัว สามารถใส่หน่วยความจำได้สูงสุด 96 KB

2.2.1.4 I/O

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

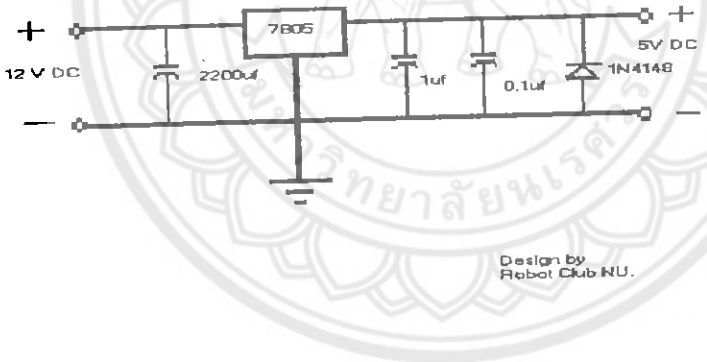
2.2.1.5 POWER

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

2.2.2 คุณลักษณะพิเศษของ CP-SB31

1. หน่วยความจำสามารถเลือกได้ทั้งขนาด ตำแหน่ง และลักษณะการทำงาน (DATA MEMORY, CODE MEMORY, CODE & DATA MEMORY)
2. สามารถพัฒนาโปรแกรมได้ทั้งภาษาแอสเซมบลี้ (ร่วมกับ SB31 – DEBUGGER) ห หรือภาษาเบสิค (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. ต่อร่วมกับอุปกรณ์สนับสนุนของบริษัทอีทีที่ได้ทันที เช่น SSRAC, RTC, 72IO, ET-AD

2.3 วงจร SUPPLY [2]

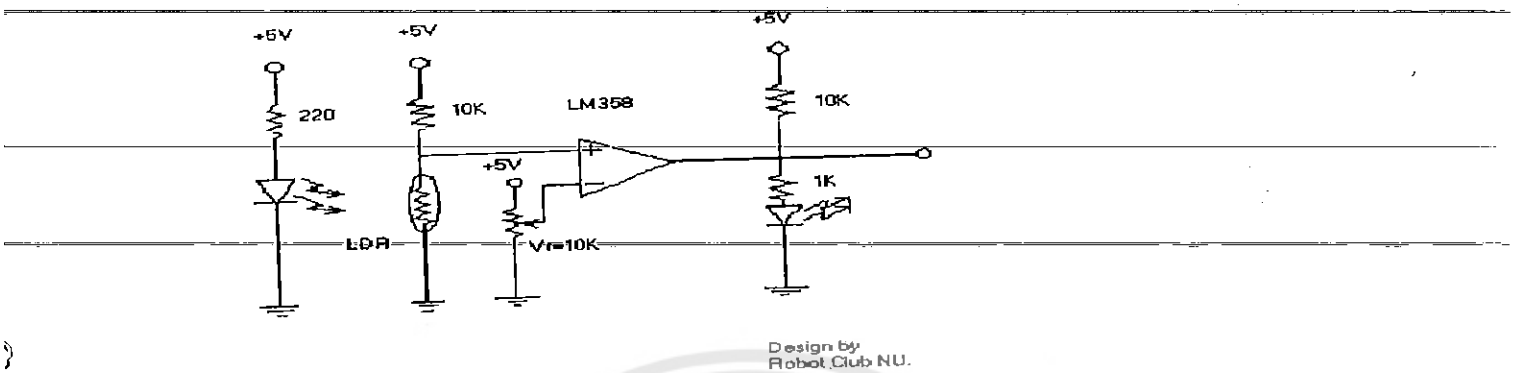


รูปที่ 2.5 วงจร Supply

หลักการทำงาน เนื่องจากวงจรทั้งหมดจะใช้ไฟ 5V DC แต่แหล่งพลังงานจะจ่ายไฟ 12V DC วงจร Supply จึงทำหน้าที่ในการแปลงไฟ จาก 12V DC ไปเป็น 5V DC

จากวงจร จะมี IC 7805 ทำหน้าที่แปลงไฟจาก 12V DC เป็น 5V DC ซึ่งจะมี ตัวเก็บประจุค่าต่างๆ เพื่อทำให้ไฟที่จ่ายออกไปสม่ำเสมอ

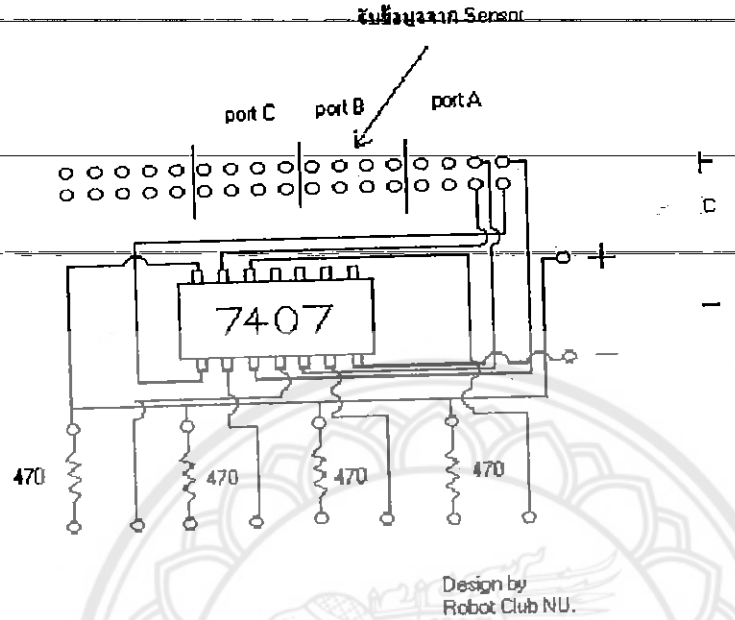
2.4 วงจร SENSOR [2]



รูปที่ 2.6 วงจร Sensor

หลักการทำงาน Sensor จะอาศัยหลักการสะท้อนของแสง โดยจะใช้ LED เป็นแหล่งกำเนิดแสง และจะใช้ LDR เป็นตัวรับแสง ซึ่งคุณสมบัติของ LDR จะมีสภาพความต้านทานไฟฟ้าไปตามความสว่างของแสง กล่าวคือ สภาพความต้านทานจะแปรผกผันกับความสว่างของแสง ซึ่งจากวงจรจะมี OP-AMP เบอร์ LM358 ซึ่ง เป็น mode comparetor ทำการเปรียบเทียบค่าศักย์ไฟฟ้าที่ได้รับจาก LDR แล้วมาแปลงเป็นสัญญาณ Digital โดยจะส่งไปให้ Board Interface ต่อไป

2.5 Interface Board [2]



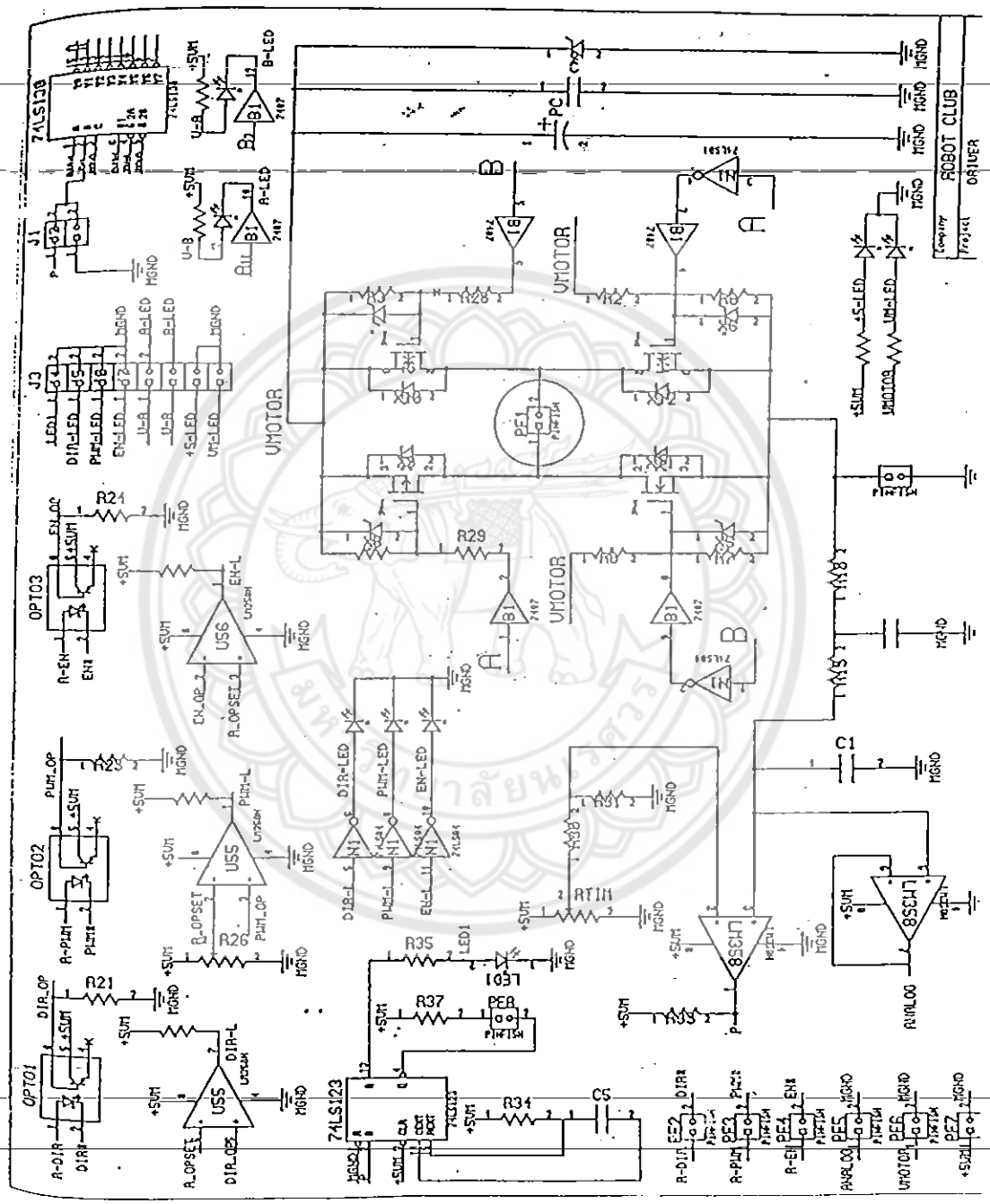
รูปที่ 2.7 วงจร Interface

หลักการทำงาน Input ข้อมูล ก็คือข้อมูลที่ได้จาก Sensor จะเข้าทาง Port B ดังรูป ซึ่ง Port B นี้เป็น Input ของ Microcontroller และ Output ของ Microcontroller ก็คือ Port A ดังรูป จาก Port A จะต่อเข้ากับ IC 7407 ซึ่งเป็น High voltage Buffer ซึ่งสัญญาณที่ส่งออกไปนี้จะไปทำการควบคุม Driver Motor ต่อไป

2.6 ไดรเวอร์มอเตอร์ [1]

หลักการทำงาน มีหน้าที่รับอินพุต จาก คอนโทรลเลอร์ ซึ่งประกอบด้วยสัญญาณพัลซ์ (Pulse Width Modulation, PWM) ทิศทาง (Dir) และสัญญาณ EN (Enable) เมื่อได้รับสัญญาณดังกล่าวแล้ว ก็จะควบคุมกระแสไฟฟ้าที่จะนำไปขับมอเตอร์ตามโปรแกรม

จากวงจรไดรเวอร์มอเตอร์จะมีการไอโซเลต (Isolate) ด้วย Optoisolater (4N25) ทั้ง 3 อินพุต เพื่อทำการแยกสัญญาณจากบอร์ดคอนโทรลเลอร์กับบอร์ดไดรเวอร์มอเตอร์ ซึ่งจะสามารถป้องกันสัญญาณรบกวนจากมอเตอร์ไปรบกวนบอร์ดคอนโทรลเลอร์และบอร์ดอื่น



รูปที่ 2.8 วงจร Driver Motor

2.6.1 อุปกรณ์อิเล็กทรอนิกส์ที่ใช้มีดังนี้ [1]

มอสเฟท MTP 2P10	2 ตัว
มอสเฟท MTP 15N06	2 ตัว
OPTO 4N25	3 ตัว
IC 74LS04	1 ตัว
IC 74LS07	1 ตัว
IC 74LS138	1 ตัว
IC 74LS123	1 ตัว
IC LM358	3 ตัว
Zener Diode IN4735A	4 ตัว
Diode IN4007	4 ตัว
ตัวต้านทาน 1.5 kΩ 0.5 วัตต์	4 ตัว
ตัวต้านทาน 1 kΩ 0.5 วัตต์	4 ตัว
ตัวต้านทาน 1 kΩ 0.25 วัตต์	11 ตัว
ตัวต้านทาน 10 kΩ 0.25 วัตต์	4 ตัว
ตัวต้านทาน 5 kΩ 0.25 วัตต์	1 ตัว
ตัวต้านทาน 10 Ω 0.25 วัตต์	2 ตัว
ตัวต้านทาน 0.2 Ω 5 วัตต์	1 ตัว
ตัวเก็บประจุอิเล็กทรอไลท์ 2200 μF	1 ตัว
ตัวเก็บประจุอิเล็กทรอไลท์ 100 μF	1 ตัว
ตัวเก็บประจุอิเล็กทรอไลท์ 47 μF	1 ตัว
ตัวเก็บประจุอิเล็กทรอไลท์ 10 μF	2 ตัว
ตัวต้านทานปรับค่าได้ 1 kΩ	2 ตัว
Jumper 2 Pin	2 ตัว
Connector 16 Pin	1 ตัว
LED	8 ตัว
Terminal 2 Pin	8 ตัว

2.6.2 การทำงานของบอร์ด Driver [1]

จากวงจรการทำงานของ DRIVER สัญญาณที่รับมาจากบอร์ด PWM จะถูก isolate ด้วย opto 4N25 สัญญาณ EN# จะเป็นตัวกำหนดให้ IC 74LS138 ทำงาน คือ สัญญาณ EN-L ที่ต่อกับขา G2A ของ 74LS138 จะต้องมีระดับสัญญาณลอจิกต่ำ(Low) ฉะนั้นสัญญาณ EN-OP จะต้องมีระดับสัญญาณลอจิกสูง(High) นั่นคือ opto ต้องทำงาน เพราะฉะนั้น EN# ที่ส่งมาจากบอร์ด PWM ต้องมีระดับสัญญาณลอจิกต่ำ(Low)

สัญญาณ DIR# เป็นตัวกำหนดทิศทางของ output ว่าเป็น A หรือ B คือถ้ามีสัญญาณ DIR-L ที่ต่อกับขา 1 ของ 74LS138 มีระดับสัญญาณลอจิกต่ำ(Low) output ของ 74LS138 ก็จะ active ที่ Y0 ทำให้ทิศทางคือ A สัญญาณ DIR-OP จะต้องมีระดับสัญญาณลอจิกต่ำ(Low) นั่นคือ opto ไม่ทำงาน ฉะนั้นสัญญาณ DIR# ที่ส่งมาจากบอร์ด PWM ก็จะต้องมีระดับสัญญาณลอจิกสูง(High) ในทางกลับกันถ้าต้องการให้ทิศทางเป็น B output ของ 74LS138 ก็ต้อง active ที่ Y1 สัญญาณ DIR-L ต้องมีระดับสัญญาณสูง(High) DIR-OP ก็ต้องมีระดับสัญญาณลอจิกสูง(High)ด้วย นั่นคือ opto ทำงาน DIR# ที่ส่งมาจากบอร์ด PWM จะต้องมีระดับสัญญาณลอจิกต่ำ(Low)

สัญญาณ PWM# ที่เข้ามาก็มีขั้นตอนการทำงานเช่นเดียวกับสัญญาณทั้งสองข้างต้น โดยสัญญาณ PWM-L ที่ต่อกับขา G1 ของ 74LS138 จะมีสัญญาณลอจิกสูง(High) หรือต่ำ(Low) ตาม pulse width ที่ส่งมาจากบอร์ด PWM ทำให้ 74LS138 ทำงานหรือไม่ทำงาน output ที่ออกมาทีก็จะเป็น pulse width ตามนั้น สัญญาณทั้ง 3 ข้างต้นจะมีการแสดงผลโดย LED โดยผ่าน 74LS04 ก่อนเพื่อกลับเฟสให้สัมพันธ์กับการทำงานและมี LED แสดงทิศทางว่าเป็น A หรือ B ด้วย

นอกจากนั้นยังมีการตรวจเช็คกระแสเกินโดยมีตัวต้านทานวัตต์สูงๆ มาต่ออนุกรม แล้วดึงสัญญาณที่ได้มาเข้าขาบวกของ OP-AMP LM358 โดยผ่าน R, C ค่าน้อยๆ เพื่อกรองสัญญาณรบกวนขาลบของ OP-AMP ตั้ง V_{ref} ไว้ในการทำงานปกติ ถ้าไม่เกิดกระแสเกิน Voltage ที่ขาบวกจะน้อยกว่าขาลบ ฉะนั้น output P จะมีระดับสัญญาณลอจิกต่ำ(Low) ไปต่อเข้ากับขา 2 ของ 74LS138 และทำงานตามปกติ แต่เมื่อเกิดกระแสเกิน หรือบางครั้งจะเป็นภาวะที่ไม่แน่นอน Output P จะเกิดเป็น pulse เล็กๆ ขึ้น จะใช้ IC 74LS123 เป็นตัวตรวจเช็คดังนี้

ตารางที่ 2.3 FUNCTION TABLE ของ 74LS123 [1]

INPUT			OUTPUT	
CLEAR	A	B	Q	Q
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⌐¬	⌐¬
H	↓	H	⌐¬	⌐¬
↑	L	H	⌐¬	⌐¬

จาก Function Table ของ 74LS123 ในสภาวะปกติ ขา A และ B (ซึ่งต่อกับ P) มีระดับสัญญาณลอจิกต่ำ(Low) ฉะนั้น Q จึงมีระดับสัญญาณลอจิกต่ำ(Low) ด้วย LED จึงไม่สว่างแต่เมื่อเกิดกระแสเกินหรือภาวะที่ไม่แน่นอน ขา B เปลี่ยนจากระดับสัญญาณลอจิกต่ำ(Low) มาเป็นระดับสัญญาณลอจิกสูง(High) ทำให้ขา Q ซึ่งมีระดับสัญญาณลอจิกต่ำ(Low) อยู่เกิด pulse มีความกว้างขึ้นอยู่กับค่า R และ C ที่ต่อเข้ากับขา CEXT และ REXT ซึ่งถ้าเป็นภาวะที่ไม่แน่นอน ขา B จะเปลี่ยนระดับสัญญาณจากต่ำ(Low) เป็นสูง(High) อยู่ตลอดเวลาจึงทำให้ขา Q มีระดับสัญญาณสูง (High) อยู่ตลอดเวลาด้วย ทำให้ LED สว่าง แสดงให้รู้ว่าการทำงานผิดปกติหรือมีกระแสเกิน

กระแสไฟฟ้าอินพุต จะไหลผ่านไดโอดเปล่งแสงภายในออฟโต้ไอโซเลตเมื่อไดโอดเปล่งแสงสว่างจะทำให้เกิดกระแสปริมาณเล็กน้อย ไหลออกจากออปโต้ทรานซิสเตอร์ผ่านความต้านทานที่มีค่าสูง ทำให้เกิดความต่างศักย์ขึ้น แล้วนำความต่างศักย์ดังกล่าวไปเปรียบเทียบด้วยออปติคอล โดยจะต้องปรับค่าความต้านทานออฟเซต ให้มีแรงดันประมาณครึ่งหนึ่งของแรงดันที่คร่อมตัวต้านทาน สัญญาณที่ได้จากออปจอมปลอมจะเป็นสัญญาณลอจิก โดยสัญญาณทั้งสาม จะไปควบคุมการทำงานของ IC Encoder (74LS138) ซึ่งสัญญาณทิศทางจะเข้าที่ขา 1A ของไอซี สัญญาณพัลซ์เข้าที่ขา 6(G1) และสัญญาณปิด-เปิดเข้าที่ขา 4(G2A)

ผลของสัญญาณทิศทางจะสามารถทำให้ขา 15 (Y0) และขา 14 (Y1) ทำงานสับกัน ขึ้นอยู่กับลอจิกของสัญญาณ สำหลับสัญญาณพัลซ์จะทำให้เอาต์พุตของไอซีทำงาน และไม่ทำงานเป็นจังหวะตามลักษณะของพัลซ์ ซึ่งจะมีผลต่อการควบคุมกระแสเฉลี่ยที่นำไปขับมอเตอร์ ส่วนสัญญาณปิด-เปิด นั้นหากมีลอจิก 1 จะทำให้ไดรเวอร์ไม่ทำงาน นั้นคือ ไม่มีกระแสไปขับ มอเตอร์ แต่ถ้ามีลอจิก 0 จะทำงาน

สำหรับวงจรที่ใช้ในการจ่ายกระแสไฟให้กับมอเตอร์ จะใช้ MOSFET ต่อกันแบบ H-bridge ขณะทำงานจะสั่งให้มอสเฟตทำงานที่ละ 2 ตัว ทะแยงมุมกัน เพื่อให้กระแสไหลผ่านมอเตอร์ ซึ่งวงจรได้ทำการออกแบบไม่ให้มอสเฟตข้างเดียวกันทำงานพร้อมกัน การขอบคุมการทำงานของมอสเฟตทั้ง 4 ตัว จะรับสัญญาณมาจาก ไอซีถอดรหัส 74LS138 ขา 15 (Y0) และ ขา 14 (Y1) ซึ่งปกติแล้วจะทำงานไม่พร้อมกัน โดยจะควบคุมมอสเฟทไขว้กัน

สัญญาณที่ควบคุมมอสเฟต ชนิด P หรือ (P-Channel) จะผ่านบัฟเฟอร์ (7407) หากอินพุตเป็นลอจิก 0 เอาท์พุตของบัฟเฟอร์จะต่อลงกราวด์ ทำให้กระแสไหลผ่านตัวต้านทานเกิดแรงดันตกคร่อมประมาณ –5V ซึ่งทำให้มอสเฟตทำงาน หากเป็นลอจิก 1 ก็จะไม่ทำงานเนื่องจาก Vgs เท่ากับ 0V

สัญญาณที่ควบคุมมอสเฟตชนิดเอ็น (N-Channel) จะผ่านนอตเกต 7404 และเข้าบัฟเฟอร์ 7407 หากอินพุตของนอตเกต เป็น0 ทำให้มอสเฟต ทำงาน หากเป็น1 ก็จะไม่ทำงาน

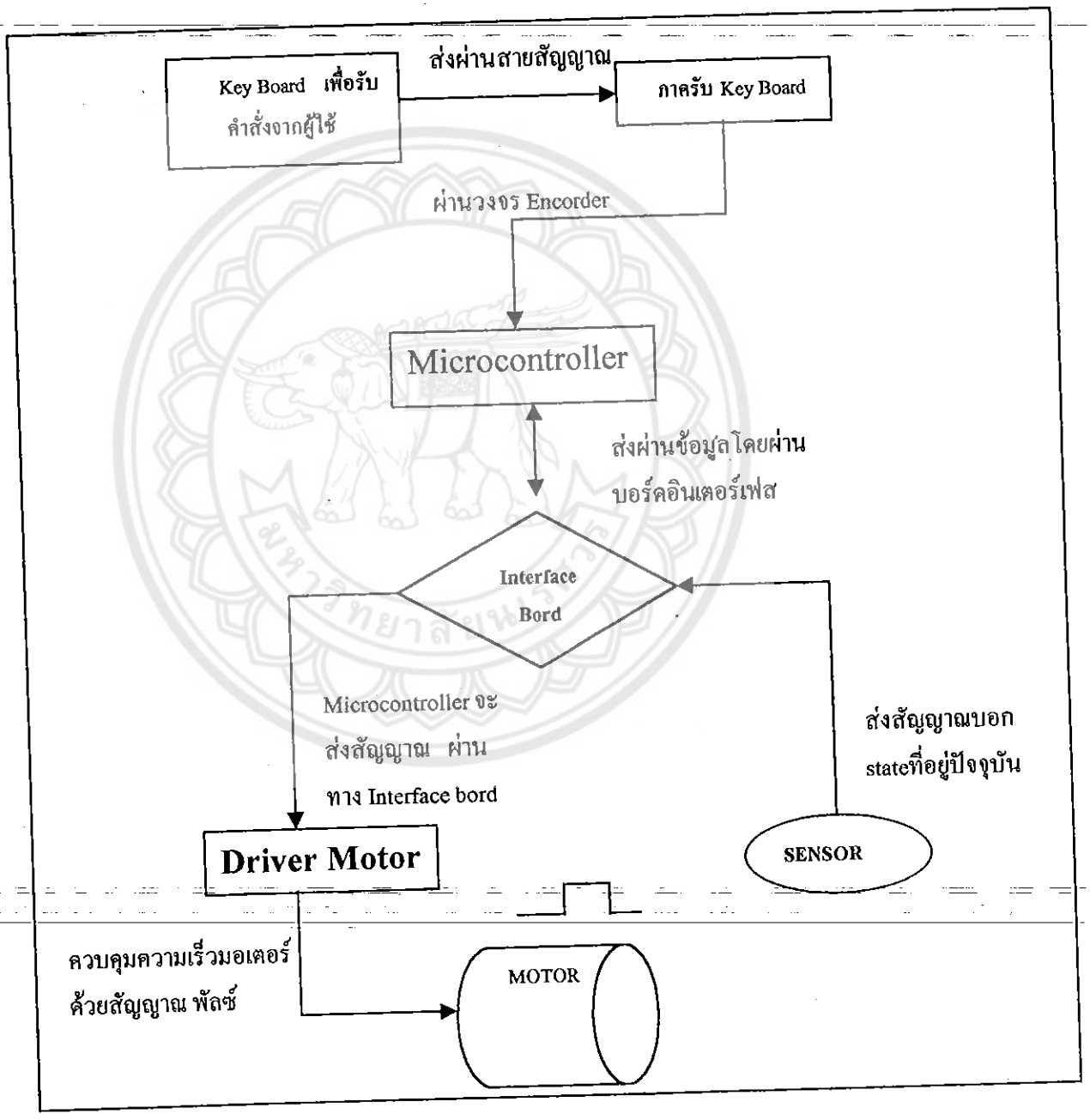
ก่อนที่กระแสไฟฟ้าที่ขับมอเตอร์จะต่อลงกราวด์ จะมีการต่อตัวต้านทานที่มีค่าน้อยมากและมีวัตต์สูงมาคั่นก่อน เพื่อนำโวลต์ที่เกิดขึ้นไปตรวจสอบปริมาณกระแสไฟฟ้าที่ไหลผ่านมอเตอร์ เพื่อป้องกันอันตรายจากกระแสโอเวอร์โหลด (Over load) โดยการใช้ตัวต้านทานและตัวเก็บประจุค่าน้อยๆ ทำให้ลดสัญญาณลบกวนและนำไปเปรียบเทียบกับ โวลต์จากตัวต้านทานปรับค่าได้ด้วยออปแอมป์ แล้วนำสัญญาญที่ได้ไปเข้าไอซีถอดรหัส 74138 ทำงานที่ขา 13 (Y2)หรือขา 14(Y3)

แทน ซึ่งเป็นผลให้มอสเฟตไม่จ่ายกระแสให้มอเตอร์ เป็นการรักษาความปลอดภัยขั้นหนึ่ง นอกจากนั้น สัญญาณที่ออกจากออปแอมป์ดังกล่าว ยังถูกขยายความกว้างของพัลซ์ที่เกิด จากกระแสเกินเพื่อขับไดโอดเปล่งแสงแสดงให้ผู้ใช้ เห็นชัดเจนขึ้น โดย ไอซีโมโนสเตเบิล 74123 ซึ่งจะทำให้ไดโอดเปล่งแสงสว่างนานขึ้น

บทที่ 3
การออกแบบและสร้างหุ่นยนต์

3.1 ศึกษาและออกแบบวงจรฮาร์ดแวร์

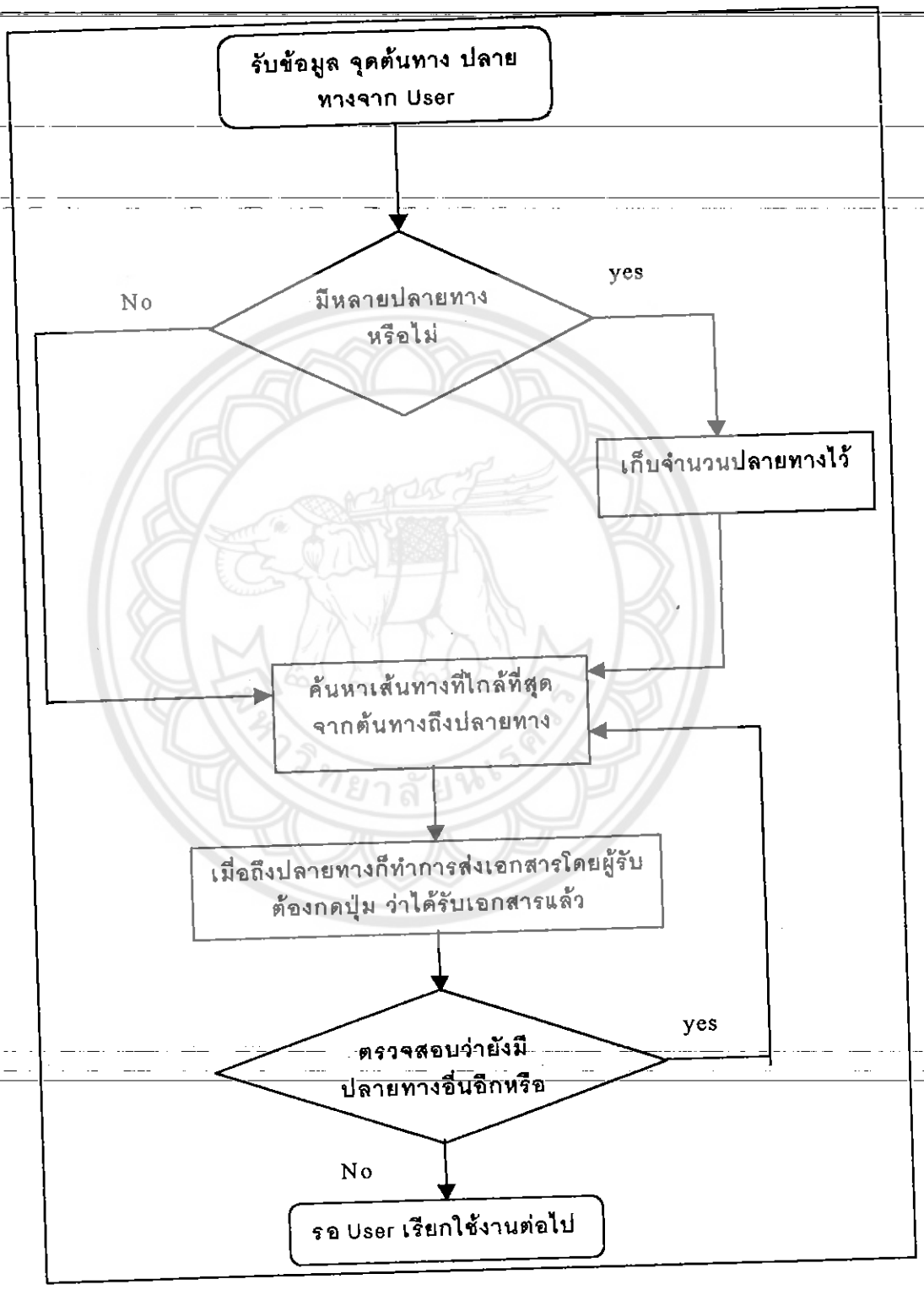
ลักษณะการทำงานโดยทั่วไปของวงจรทั้งหมดคือ



รูปที่ 3.1 แผนผังการทำงานของวงจรทั้งหมด

3.2 โปรแกรมควบคุม

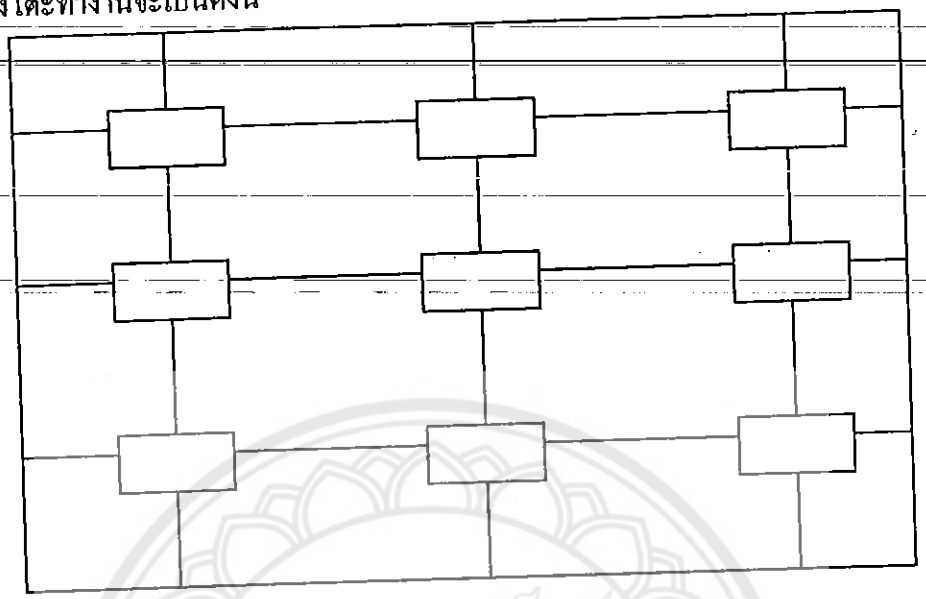
3.2.1 รูปแบบของโปรแกรมควบคุมจะเป็นไปตาม Flowchart ดังนี้



รูปที่3.2 แผนผังลำดับการทำงานของโปรแกรม

3.2.2 การค้นหาเส้นทาง

1. การจัดวางโต๊ะทำงานจะเป็นดังนี้



หมายเหตุ

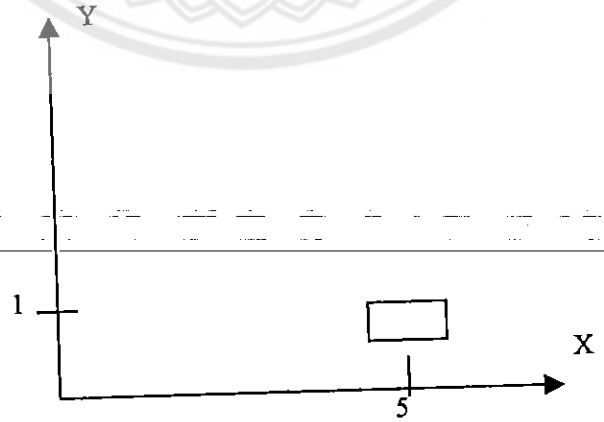
: โต๊ะทำงาน

: เส้นทางเดินของหุ่นยนต์

รูปที่ 3.3 สถานที่จำลอง

2. ลำดับขั้นตอนในการหาเส้นทางของหุ่น

จะคิดว่าทั้งระบบอยู่บนระนาบ XY โดยตำแหน่งของโต๊ะทำงานแต่ละโต๊ะ จะคิดเป็นพิกัด ดังรูป



รูปที่ 3.4 ตัวอย่างพิกัดของโต๊ะ

จากรูปโต๊ะดังกล่าวอยู่ที่พิกัด (5,1) เมื่อผู้ใช้ต้องการเรียกใช้จะระบุเป็นตำแหน่ง แล้วโปรแกรมจะมาทำการแปลความหมายเป็นพิกัดอีกครั้งหนึ่ง

เมื่อผู้ใช้ระบุต้นทางและปลายแล้ว โปรแกรมจะแปลเป็นพิกัดต้นทางและปลายทาง จากนั้นจะนำพิกัดมาเข้าสู่กระบวนการหาเส้นทางต่อไป

โดยจะยกตัวอย่างดังนี้

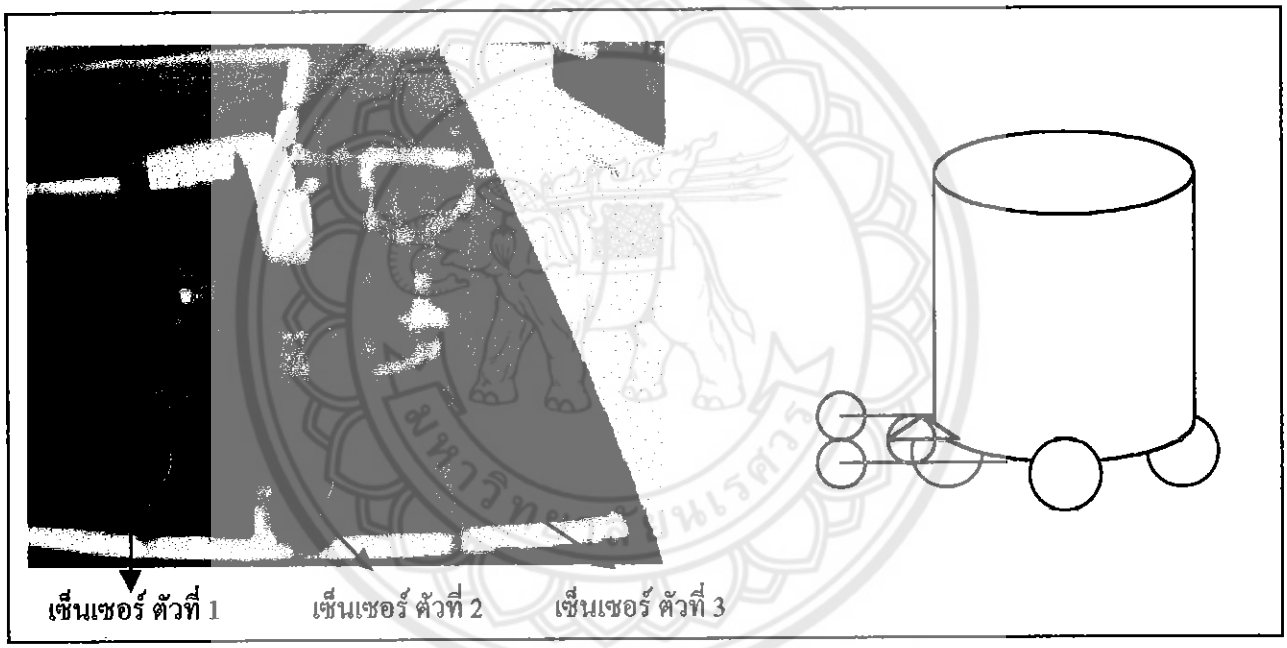
จากต้นทาง (5,2) ไปยังปลายทาง (15,4)

เราจะได้ระยะทางในแนวแกน X=15-5=5

และระยะทางในแนวแกนY=4-2=2

หุ่นก็จะเดินไปในแนวแกนX=5หน่วยแกนY=2หน่วย ตามลำดับ

3.3 โครงสร้างของหุ่นเมื่อสร้างสำเร็จ



รูปที่ 3.5 ภาพของหุ่นยนต์ที่สร้างเสร็จแล้ว

3.3.1 การนำเซ็นเซอร์มาใช้งาน

3.3.1.1 เซ็นเซอร์ตัวที่ 1 และ 3 ใช้ในการตรวจจับเส้นเส้นที่ตัดผ่านเพื่อหาจุดตัดของเส้น หากพบเส้นด้านข้างเซ็นเซอร์จะบอกให้หุ่นทราบว่าเป็นทางแยกแล้ว เพื่อนำไปวิเคราะห์ว่าอยู่ที่ตำแหน่ง (x,y) ใด ๆ

3.3.1.2 เซ็นเซอร์ตัวที่ 2 ใช้ในการตรวจจับเส้นทางที่หุ่นใช้ในการเดินตามเส้นทางในด้านหน้า

ห้องสมุดคณะวิศวกรรมศาสตร์

150941 36 4400609
มร. TJ
ก676ค. ก11
2544 ก676ค
ค.2 2544.C.2

บทที่ 4
ผลการทดลองและผลการวิเคราะห์

4.1 จุดประสงค์ของการทดลอง

4.1.1 เพื่อวัดประสิทธิภาพการทำงานของ ดีซี มอเตอร์ ควบคู่กับ ไดรเวอร์มอเตอร์ PWM

4.1.2 เพื่อวัดประสิทธิภาพการทำงานของเซ็นเซอร์

4.1.3 ทดสอบการทำงานของโปรแกรม

4.2 ขั้นตอนการทดลอง

4.2.1 การทดสอบประสิทธิภาพของ ดีซี มอเตอร์ ควบคู่กับไดรเวอร์มอเตอร์ PWM

วิธีการทดลอง

การทดสอบประสิทธิภาพการทำงานของ ดีซีมอเตอร์ ควบคู่กับ ไดรเวอร์มอเตอร์ PWM จะทำการปล่อยสัญญาณ PWM จากไมโครคอนโทรลเลอร์ ซึ่งมีค่า duty cycle 1-100 ระดับ

1. ใช้ duty cycle ระดับที่ 100 แล้ววัดค่าความต่างศักย์ไฟฟ้าที่ขั้วมอเตอร์
2. ใช้ duty cycle ในระดับที่ลดลงมาคราวละ ¼ ของค่าเดิม
3. นำค่าความต่างศักย์จากข้อ 1 และ ข้อ 2 มาทำการวิเคราะห์สัดส่วน

4.2.2 การทดสอบประสิทธิภาพการทำงานของ เซ็นเซอร์

วิธีการทดลอง

เซ็นเซอร์ที่ใช้จะใช้หลักความสามารถการสะท้อนแสงของแสงสีต่างๆ

1. ทดลองใช้สีขาวกับเซ็นเซอร์แล้ววัดค่าความต่างศักย์ที่ได้จากเซ็นเซอร์
2. ทดลองใช้สีดำกับเซ็นเซอร์แล้ววัดค่าความต่างศักย์ที่ได้จากเซ็นเซอร์
3. นำค่าความต่างศักย์ที่ได้จากข้อ 1 และข้อ 2 มาวิเคราะห์เพื่อแสดงเป็นค่าตรรกศาสตร์

4.2.3 ทดสอบการทำงานของโปรแกรม

วิธีการทดลอง

ทดลองการกำหนดค่าพิกัดบนระนาบสองมิติ x, y ของจุดต้นทางและจุดปลายทาง แล้วนำค่าพิกัดมาหาระยะทางในแนวแกน x และแกน y

1. กำหนดค่าพิกัดให้ระยะทางในแนวแกน x เป็น ลบและในแนวแกน y เป็นบวก
2. กำหนดค่าพิกัดให้ระยะทางในแนวแกน x เป็น บวกและในแนวแกน y เป็นลบ
3. กำหนดค่าพิกัดให้ระยะทางในแนวแกน x และ y เป็นลบ
4. กำหนดค่าพิกัดให้ระยะทางในแนวแกน x และ y เป็นบวก
5. สังเกตการเคลื่อนที่ของข้อ 1, 2, 3 ตามลำดับ

4.2 ผลการทดลอง

4.2.1 ผลการทดสอบประสิทธิภาพการทำงานของ ดีซีมอเตอร์ ควบคู่กับ ไดรเวอร์มอเตอร์

ตารางที่ 4.1 ผลการทดลองหาค่าความต่างศักย์ที่ได้จาก ดีซีมอเตอร์ เพื่อกำหนดค่า Duty cycle ค่าต่าง ๆ

Duty cycle	ค่าความต่างศักย์ (โดยประมาณ)
100	24 V.
75	18 V.
50	12 V.
25	6 V.
0	0 V.

4.2.2 ผลการทดสอบประสิทธิภาพการทำงานของ เซ็นเซอร์

ตารางที่ 4.2 ผลการทดลองการใส่สีดำและสีขาวแล้ววัดค่าความต่างศักย์ที่ได้จาก เซ็นเซอร์

สี	ค่าความต่างศักย์(โดยประมาณ)
ดำ	5 V.
ขาว	0 V.

4.2.3 ผลการทดสอบการทำงานของโปรแกรม

ตารางที่ 4.3 ผลการทดลองเมื่อกำหนดค่า x, y และค่าต่าง ๆ

ต้นทาง		ปลายทาง		ΔX	ΔY	การเคลื่อนย้ายของหุ่นยนต์
X	Y	X	Y			
0	0	4	5	4	5	จะเคลื่อนที่ไปในแนวแกน x ทางบวก 4 หน่วย จะเคลื่อนที่ไปในแนวแกน y ทางบวก 5 หน่วย
4	5	3	7	-1	2	จะเคลื่อนที่ไปในแนวแกน x ทางลบ 1 หน่วย จะเคลื่อนที่ไปในแนวแกน y ทางบวก 2 หน่วย
3	7	6	2	3	-5	จะเคลื่อนที่ไปในแนวแกน x ทางบวก 3 หน่วย จะเคลื่อนที่ไปในแนวแกน y ทางลบ 5 หน่วย
6	2	0	0	-6	-2	จะเคลื่อนที่ไปในแนวแกน x ทางลบ 6 หน่วย จะเคลื่อนที่ไปในแนวแกน y ทางบวก 2 หน่วย

4.4 ผลการวิเคราะห์

4.4.1 จากผลการทดลองประสิทธิภาพการทำงานของ ดีซีมอเตอร์ ควบคู่กับมอเตอร์ไดรเวอร์ PWM จะพบว่า duty cycle ที่ทำการป้อนเข้าไปมีผลกับค่าความต่างศักย์ที่ขั้ว

ของมอเตอร์ ซึ่งค่า duty cycle ยิ่งค่ามากค่าความต่างศักย์ที่ขั้วมอเตอร์ก็จะยิ่งมาก ตามไปด้วยตามสมการ (4.1)

duty cycle = k V (4.1)

ซึ่งหาค่า k ได้จาก k = V/duty cycle (4.2)

มีผลทำให้ความเร็วมอเตอร์แปรผันตรงกับค่า duty cycle ซึ่งสามารถควบคุมความเร็วของมอเตอร์ได้ 100 ระดับ

4.4.2 จากผลการทดลองประสิทธิภาพการทำงานของเซนเซอร์ สีดำจะทำให้เซนเซอร์มีความต่างศักย์มากที่สุดและสีขาวจะทำให้เซนเซอร์มีความต่างศักย์น้อยที่สุดซึ่งจะมีผลทำให้

สีดำ จะให้ค่าทางตรรกศาสตร์ เป็น " 1 "
สีขาวจะให้ค่าทางตรรกศาสตร์ เป็น " 0 "

ซึ่งในทางปฏิบัติจริงแล้วสามารถใช้ได้ทุก ๆ สี แต่จะต้องทำการเปลี่ยนแปลงฟังก์ชั่นของเซนเซอร์ ให้ควบคู่กับสถานที่ปฏิบัติงาน

4.4.3 จากผลการทดสอบของโปรแกรมจะพบว่า

เมื่อจุดต้นทางและจุดปลายทางซึ่งอยู่ในรูปพิกัด (x, y) มาผ่านกระบวนการทางคณิตศาสตร์แล้วได้เป็นค่า ลบ หุ่นยนต์จะเคลื่อนไปทางซ้ายเป็นระยะทาง dx และ dy ตามลำดับ

ในกรณีที่ค่าออกมาเป็นบวก หุ่นยนต์จะเคลื่อนที่ไปทางขวาเป็นระยะทาง dx และ dy ตามลำดับ

บทที่ 5
บทสรุป

5.1 สรุปผลการทดลอง

5.1.1 การใช้ไดรเวอร์ PWM ไปทำการควบคุมมอเตอร์ทำให้สามารถควบคุมความเร็วของมอเตอร์ได้ ซึ่งจะทำให้ควบคุมการเคลื่อนที่ของหุ่นยนต์ได้สะดวก

5.1.2 ใช้เซนเซอร์แสงทำให้สามารถหาเส้นทางจากจุดต้นทางไปยังจุดปลายทางได้ โดยเซนเซอร์จะอาศัยหลักการสะท้อนของแสงและความสามารถในการดูดกลืนแสงของสีต่าง ๆ

5.1.3 การทำงานของโปรแกรมจะเป็นตัวบอกเส้นทางจากพิกัดต้นทางไปยังปลายทางและยังควบคุมการเคลื่อนที่ของหุ่นยนต์อีกด้วย

5.2 ปัญหาในการทดลองและแนวทางแก้ไข

ปัญหา : เกิดจากการถูกสัญญาณรบกวนของไมโครคอนโทรลเลอร์จากไดรเวอร์มอเตอร์

สาเหตุ : เนื่องจากมอเตอร์เป็นสัญญาณ อนาลอก คือมีค่าตั้งแต่ 0-24 โวลต์ แต่ไมโครคอนโทรลเลอร์เป็นสัญญาณดิจิตอล

ผล : ไมโครคอนโทรเลอร์เกิดค่าความผิดพลาด

วิธีแก้ไข: ใช้ IC 4N25 เพื่อแยกสัญญาณ อนาลอก ออกจากดิจิตอล

ปัญหา : เซนเซอร์มีความละเอียดในการตรวจจับน้อย

สาเหตุ : อาจมีแสงรบกวนจากสถานที่อื่น

ผล : หุ่นยนต์หาเส้นทางไม่พบ

วิธีแก้ไข :ปรับเปลี่ยนฟังก์ชั่นการทำงานให้มีความละเอียดในการตรวจจับเส้นทางมากขึ้น

ปัญหา : หุ่นยนต์เมื่อเจอเส้นทางที่มีทางแยกจะเกิดความผิดพลาด

สาเหตุ : ไม่มีตัวตรวจจับว่าเป็นทางแยก

ผล : เกิดความผิดพลาดในการหาเส้นทาง

วิธีแก้ไข :เพิ่มเซนเซอร์ในการตรวจจับแยก

5.3 ปัญหาที่พบในการทำงานของหุ่นและแนวทางแก้ไข

ปัญหา : ไดรเวอร์มอเตอร์มีปัญหาทำให้ไม่สามารถ หมุนล้อถอยหลังได้

ข้อสันนิษฐาน : อาจมีกระแสไฟฟ้าเกินในไดรเวอร์

การแก้ปัญหา : ปรับค่าความต้านทานปรับค่าได้ให้มีกระแสที่เหมาะสม

ปัญหา : การเลี้ยวของหุ่นในมุมกว้าง

สาเหตุ : เนื่องจากการที่ล้อของหุ่นไม่สามารถหมุนย้อนกลับได้ทำให้ต้องใช้พื้นที่ในการเลี้ยวมาก

วิธีแก้ปัญหา : แก้ปัญหาการทำงานของไดรเวอร์ และปรับปรุงการเขียนโปรแกรมเพื่อรับรองการทำงาน

ปัญหา : หุ่นไม่สามารถกลับตัว 180 องศาได้

สาเหตุ : เนื่องจากหุ่นเลี้ยวในมุมที่กว้างทำให้มีการตกเส้น หรือ วิเคราะห์เส้นผิดพลาด

วิธีแก้ปัญหา : แก้ปัญหาการเลี้ยวของหุ่นในมุมกว้าง และปรับปรุงการเขียนโปรแกรมเพื่อรับรองการทำงาน

ปัญหา : การวิเคราะห์เส้นผิดพลาด

ข้อสันนิษฐาน : 1. มีแสงภายนอกมารบกวนการทำงานของเซ็นเซอร์
2. เมื่อมีสีที่ใกล้เคียงกันเกินไปทำให้ไม่สามารถตรวจเช็คได้

วิธีแก้ปัญหา : 1. หาวัสดุมากันแสงรบกวน
2. ปรับค่าความต้านทานปรับค่าได้ให้มีความละเอียดมากขึ้น

ปัญหา : การเดินทางที่มีค่า dx และ dy เป็นค่าลบผิดพลาด

สาเหตุ : โปรแกมมีขนาดใหญ่มากทำให้เกิด Error ไม่สามารถแปลงค่า "ลบ" เป็น "บวก" ได้

วิธีแก้ปัญหา : ทำการเพิ่มค่า Interupt ให้แก่ CPU

5.4 แนวทางในการพัฒนาหุ่นยนต์รับส่งเอกสารในออฟฟิศต่อไป

1. ควรเพิ่มการทำงานระยะไกล(ระบบรีโมทคอนโทรล)เพื่อให้สามารถนำไปใช้งานได้สะดวกยิ่งขึ้น
2. ควรลดรัศมีการเลี้ยวให้แคบลงเพื่อการทำงานอย่างมีประสิทธิภาพ
3. ควรออกแบบวงจรและอุปกรณ์ให้รองรับการทำงานได้หลายรูปแบบเพื่อประโยชน์ในการพัฒนาหุ่นยนต์ให้เป็นไปในรูปแบบต่าง ๆ ต่อไป

เอกสารอ้างอิง

[1] ชมรมโรบอท มหาวิทยาลัยนเรศวร, คู่มือชมรมโรบอท, มหาวิทยาลัยนเรศวร, 1998

[2] สุนทร วิทูสุรพจน์, การใช้งานไมโครคอนโทรเลอร์ตระกูล 8051

ภาคผนวก ก

แสดงโปรแกรมการทำงานของหุ่นยนต์ รวมทั้งเป็นโปรแกรมภาครับของคีย์บอร์ด

```
#include <reg51.h>
#include <string.h>
#include <absacc.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>

#define Pa1    XBYTE [0xE0E0]
#define Pb1    XBYTE [0xE0E1]
#define Pc1    XBYTE [0xE0E2]
#define Pcon1  XBYTE [0XE0E3]

#define LCD_CONTROL   XBYTE[0xE0C0]
#define LCD_BUSY_FLAG XBYTE[0xE0C1]
#define LCD_WRITEDATA XBYTE[0xE0C2]
#define LCD_READDATA  XBYTE[0xE0C3]

xdata unsigned int quantum;
xdata unsigned int duty1,duty2,period;
xdata unsigned char mPa,mPb;

xdata float m1;

xdata char screen[5];
data unsigned int x,y;

typedef struct table
{
  unsigned int num_table;
  unsigned int x_poin;
  unsigned int y_poin,
};

xdata struct table data_table[20];

void serch_data_table(unsigned int input)
{
 unsigned int i;
 data unsigned int MAX;

 data_table[0].num_table=0;  data_table[1].num_table=1;
 data_table[0].x_poin=0;     data_table[1].x_poin=1;
 data_table[0].y_poin=0;     data_table[1].y_poin=1;

 data_table[2].num_table=2;  data_table[3].num_table=3;
 data_table[2].x_poin=2;     data_table[3].x_poin=3;
 data_table[2].y_poin=0;     data_table[3].y_poin=0;

 data_table[4].num_table=4;  data_table[5].num_table=5;
 data_table[4].x_poin=0;     data_table[5].x_poin=1;
 data_table[4].y_poin=1;     data_table[5].y_poin=1;

 data_table[6].num_table=6;  data_table[7].num_table=7;
 data_table[6].x_poin=2;     data_table[7].x_poin=2;
 data_table[6].y_poin=1;     data_table[7].y_poin=1;

 data_table[8].num_table=8;  data_table[9].num_table=9;
 data_table[8].x_poin=2;     data_table[9].x_poin=2;
 data_table[8].y_poin=1;     data_table[9].y_poin=2;

 MAX=9;
 for(i=0;i<MAX+1;i++)
  {
  if(input==data_table[i].num_table)
    {
    x=data_table[i].x_poin;
```

```
    y=data_table[i].y_poin;
    }
  }
}

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

//start lcd
void wait_for_lcd_ready (void)
{
  while ((LCD_BUSY_FLAG & 0x80) != 0)
    {
    }
}

void specified_lcd(void)
{
  LCD_CONTROL=0x38;
  wait_for_lcd_ready ();
}
void clear_lcd(void)
{
  LCD_CONTROL=0x01;
  wait_for_lcd_ready ();
}

void cursor_on(void)
{
  LCD_CONTROL=0x0f;
  wait_for_lcd_ready ();
}

void cursor_off(void)
{
  LCD_CONTROL = 0x0c;
  wait_for_lcd_ready ();
}

void entry_mode_set(void)
{
  LCD_CONTROL=0x02;
  wait_for_lcd_ready ();
}

void init_lcd(void)
{
  specified_lcd();
  cursor_off();
  clear_lcd();
  entry_mode_set();
  cursor_off();
}
void WChar(char ch)
{
  LCD_WRITEDATA = ch;
  wait_for_lcd_ready();
}

void gotoxy(unsigned char row, unsigned char col)
{
  if (row == 1) LCD_CONTROL = 0x80+col-1; else
  if (row == 2) LCD_CONTROL = 0xc0+col-1; else
  if (row == 3) LCD_CONTROL = 0x90+col-1; else
  if (row == 4) LCD_CONTROL = 0xd0+col-1;
  wait_for_lcd_ready ();
}

void WCharxy(unsigned char row,unsigned char
col,unsigned char ch)
{
  gotoxy(row,col);
  WChar(ch);
}

void string_to_lcd (unsigned char row,unsigned char
col,char *st)
{
  data unsigned char mcol,colfinish,len;
  len = strlen(st);
  colfinish=col+len-1;
```

```
    if (colfinish > 16)  colfinish=16;
    for (mcol=col; mcol<=colfinish; mcol++)
                {
                WCharxy(row,mcol,s[mcol-col]);
                }
}
void ftoa(float mfloat,char *mstring)
{
  xdata char mchar,i,j,ii,iii;
  xdata long mlong,mlong1;
  xdata float mfloat1,mfloat2,mfloat3;
  xdata char sint[16];
  xdata char sfloat[16];
  xdata char msign;
  if (mfloat<0.0)
     {
     mfloat=-mfloat;
     msign='-';
     }
   else
     {
     msign=' ';
     }
   i=0;
   mfloat1=mfloat;
   mlong=mfloat1;
   mchar=(mlong%10)+'0';
   sint[i]=mchar;
   while (mlong!=0)
      {
      mfloat1=mfloat1/10.0;
      mlong=mfloat1;
      if (mlong==0)
         {
         break;
         }
      mchar=(mlong%10)+'0';
      i=i+1;
      sint[i]=mchar;
      }
    mlong1=mfloat;
    mfloat2=mlong1;
    mfloat1=mfloat-mfloat2;
for (j=0;j<=15;j++)
   {
   mfloat1=mfloat1*10.0;
   mlong=mfloat1;
   mchar=(mlong%10)+'0';
   sfloat[j]=mchar;
   }
mstring[0]=msign;
for (ii=i;ii>=0;ii--)
   {
   mstring[i-ii+1]=sint[ii];
   }
mstring[i+2]='.';
for (j=0;j<=15;j++)
   {
   iii=i+j+3;
   if (iii==8)
      {
      break;
      }
   mstring[iii]=sfloat[j];
   }
mstring[iii]=0;
}//end of lcd function.

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 drive_pwm1(unsigned char n_pwm)
{
  if (n_pwm==0)
     {
     mPa=mPa | 0x01;
     }
  else
     {
     mPa=mPa & 0xfe;
     }
  Pc1=mPa;
}

void drive_pwm2(unsigned char n_pwm)
{
  if (n_pwm==0)
     {
     mPa=mPa | 0x04;
     }
  else
     {
     mPa=mPa & 0xfb;
     }
  Pc1=mPa;
}

void init_timer(void)
{
  TMOD=0x01;  //timer1=mode0,timer0=mode1
  TH0=0xff;  //8bit high
  TL0=0x80;  //8bit low
  ET0=1;     //let timer0 interrupt
  TR0=1;     //start timer0
  EA=1;      //let all interrupt
}

void int_quantum(void) interrupt 1
{
//Interrupt Timer0
  if (quantum<=duty1)
     {
     drive_pwm1(1);
     }
  else
     {
     drive_pwm1(0);
     }
  if (quantum<=duty2)
     {
     drive_pwm2(1);
     }
  else
     {
     drive_pwm2(0);
     }
  if (quantum>=period)
     {
     quantum=0;
     }
  quantum=quantum+1;
  TH0 = 0xff;      //8bit high
  TL0 = 0x80;      //8bit low
}

void motor (unsigned char left,unsigned char right)
{
  duty1=left;
  duty2=right;
}

void forward(void)
{
  xdata unsigned char sensor;
  data unsigned int fastl=75,slowl=31,
    fastr=100,slowr=40,stop=0;
  sensor=Pa1&0x0e;
  if(sensor==0x02)
     {
     motor(fastl,slowr);
     }
  else if(sensor==0x06)
     {
//     motor(stop,fastr);
```

```
  motor(fastl,stop);
  }
 else if(sensor==0x08)
  {
  motor(slowl,fastr);
  }
 else if(sensor==0x0a)
  {
  motor(fastl,fastr);
  }
 else if(sensor==0x0c)
  {
//   motor(fastl,stop);
   motor(stop,fastr);
  }
 else if(sensor==0x0e)
  {
  motor(fastl,fastr);
  }
 else
  {
  motor(slowl,slowr);
  }
}


void stop(void)
{
  motor(0,0);
}

void rotate_left(void)
{
  xdata unsigned char sensor;
  motor(0,100);
  delay(1000);
 // delay(5000);
  sensor=Pa1&0x11;
  while(sensor==0x11)
    {
    sensor=Pa1&0x11;
    motor(0,100);
    }
  sensor=Pa1&0x02;
  while(sensor==0x02)
    {
    sensor=Pa1&0x02;
    motor(0,100);
    }
}

void rotate_right(void)
{
  xdata unsigned char sensor;
  motor(100,0);
  delay(1000);
//  delay(5000);
  sensor=Pa1&0x11;
  while(sensor==0x11)
    {
    sensor=Pa1&0x11;
    motor(100,0);
    }
  sensor=Pa1&0x08;
  while(sensor==0x08)
    {
    sensor=Pa1&0x08;
    motor(100,0);
    }
}

x1_to_x2(unsigned int x1,unsigned int x2)
{
  data int dx;
  dx=x2-x1;
  return dx;
}

y1_to_y2(unsigned int y1,unsigned int y2)
{
  data int dy;
  dy=y2-y1;
  return dy;
}

void main(void)
```

```
{
 char ch[1];
 xdata char insensor,screen[5];
 data unsigned int num_table;
 xdata unsigned int
x_axis,y_axis,initial_x=0,initial_y=0,z;
 delay(1000);
 Pcon1=0x90;
 init_timer();
 init_serial();
 init_lcd();
 period=100;
 duty1=0;
 duty2=0;
 init_lcd();
 delay(1000);

 string_to_lcd(1,1,"Hello");
 string_to_lcd(2,1,"I'm Office Robot");

 while(1)
   {
   z=1;ch[0]=' ';
   while(z)
     {
     ch[0]=recv_char();
     if(ch[0]!=' ')
       z=0;
     }

   num_table=atoi(ch);  //change char to integer
   serch_data_table(num_table);
   init_lcd();
   string_to_lcd(2,1,"X=");
   sprintf(screen,"%d",x);
   string_to_lcd(2,4,screen);

   string_to_lcd(2,9,"Y=");
   sprintf(screen,"%d",y);
   string_to_lcd(2,12,screen);

   x_axis=x1_to_x2(initial_x,x);//input point x1 to x2
   y_axis=y1_to_y2(initial_y,y);//input point y1 to y2
   string_to_lcd(1,1,"DX=");
   sprintf(screen,"%d",x_axis);
   string_to_lcd(1,4,screen);
   initial_x=x;

   string_to_lcd(1,7,"DY=");
   sprintf(screen,"%d",y_axis);
   string_to_lcd(1,10,screen);
   initial_y=y;

   delay(2000); //delay for ready

   init_lcd();
   string_to_lcd(1,1,"Let,s go");

   if(x_axis<0)
     {
     x_axis=abs(x_axis);
     stop();
     delay(2000);
//     delay(10000);
     init_lcd();
     string_to_lcd(1,1,"rotate_left");
     rotate_left();
     }
   else if(x_axis>0)
     {
     stop();
     delay(2000);
//     delay(10000);
     init_lcd();
     string_to_lcd(1,1,"rotate_right");
     rotate_right();
     }
   init_lcd();
   string_to_lcd(1,1,"stop");

   stop();
//   delay(10000);
   delay(2000);
   forward();
   delay(700);
```

```
init_lcd();
sprintf(screen,"%d",x_axis);
string_to_lcd(1,1,screen);

 while(x_axis>0)
  {
  insensor=Pa1&0x1f;
  if((insensor&0x11)==0x11)
    forward();
  else
    {
//    delay(4000);
    delay(100);
    x_axis--;
    init_lcd();
    sprintf(screen,"%d",x_axis);
    string_to_lcd(1,1,screen);
    }
  }
 init_lcd();
 string_to_lcd(1,1,"stop");

  stop();
  delay(2000);
//  delay(10000);

  if(y_axis>0) //y value is +
    {
 init_lcd();
 string_to_lcd(1,1,"rotate_left");

    rotate_left();
    }
  else if(y_axis<0) //y value is -
    {
    y_axis=abs(y_axis);
 init_lcd();
 string_to_lcd(1,1,"rotate_right");
    rotate_right();
    }
init_lcd();
string_to_lcd(1,1,"stop");

  stop();
  delay(2000);
//  delay(10000);
  forward();
  delay(700);

  init_lcd();
  string_to_lcd(1,1,"DY=");
  sprintf(screen,"%d",y_axis);
  string_to_lcd(1,4,screen);

  while(y_axis>0)
   {
   insensor=Pa1&0x1f;
   if((insensor&0x11)==0x11)
     forward();
   else
     {
//     delay(4000);
     delay(100);
     y_axis--;
     init_lcd();
     sprintf(screen,"%d",y_axis);
     string_to_lcd(1,1,screen);
     }
   }
  stop();
  delay(2000);
  }
}
```

ภาคผนวก ข

แสดงโปรแกรมที่ใช้ในอุปกรณ์ การควบคุมทำงานของหุ่นยนต์ในระยะไกล

```
#include <reg51.h>
#include <string.h>
#include <absacc.h>
#include <stdlib.h>
#include <math.h>
#include <stdio.h>

#define Pa1    XBYTE [0xE0E0]
#define Pc1    XBYTE [0xE0E1]
#define Pb1    XBYTE [0xE0E2]
#define Pcon1  XBYTE [0XE0E3]

#define LCD_CONTROL   XBYTE[0xE0C0]
#define LCD_BUSY_FLAG XBYTE[0xE0C1]
#define LCD_WRITEDATA XBYTE[0xE0C2]
#define LCD_READDATA  XBYTE[0xE0C3]

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

void wait_for_lcd_ready (void)
{
  while ((LCD_BUSY_FLAG & 0x80) != 0)
     {
     }
}

void specified_lcd(void)
{
  LCD_CONTROL=0x38;
  wait_for_lcd_ready 0;
}

void shift_lcd(void)
{
  LCD_CONTROL=0x1c;
  wait_for_lcd_ready 0;
}

void clear_lcd(void)
{
  LCD_CONTROL=0x01;
  wait_for_lcd_ready 0;
}

void cursor_on(void)
{
  LCD_CONTROL=0x0f;
  wait_for_lcd_ready 0;
}

void cursor_off(void)
{
  LCD_CONTROL = 0x0c;
  wait_for_lcd_ready 0;
}
```

```
void entry_mode_set(void)
{
  LCD_CONTROL=0x06;
  wait_for_lcd_ready();
}

void init_lcd(void)
{
  specified_lcd();
  cursor_off();
  clear_lcd();
  entry_mode_set();
  cursor_off();
}

void WChar(char ch)
{
  LCD_WRITEDATA = ch;
  wait_for_lcd_ready();
}

void gotoxy(unsigned char row, unsigned char col)
{
  if (row == 1) LCD_CONTROL = 0x80+col-1; else
  if (row == 2) LCD_CONTROL = 0xc0+col-1; else
  if (row == 3) LCD_CONTROL = 0x90+col-1; else
  if (row == 4) LCD_CONTROL = 0xd0+col-1;
  wait_for_lcd_ready();
}

void WCharxy(unsigned char row,unsigned char
col,unsigned char ch)
{
  gotoxy(row,col);
  WChar(ch);
}

void string_to_lcd (unsigned char row,unsigned char
col,char *st)
{
  data unsigned char mcol,colfinish,len;

  len = strlen(st);
  colfinish=col+len-1;
  if (colfinish > 16)  colfinish=16;

  for (mcol=col; mcol<=colfinish; mcol++)
        {
        WCharxy(row,mcol,st[mcol-col]);
        }
}

char waitkey(void)
{
  data char key1;
  key1=Pb1|0xf0;
  if (key1!=0xff)
        {
        delay(10000);
        if ((Pb1|0xf0)==key1)
          {
          }
        else
          {
          key1=0xff;
          }
        }
  else
        {
        }
  return (key1);
}

char inkey(void)
{
  data char key1,key2;
  key2='';
```

```
Pc1=0xfe;
key1=waitkey();
if ((key1&0x01)==0)
        {
        key2='0';
        }
if ((key1&0x02)==0)
        {
        key2='1';
        }
if ((key1&0x04)==0)
        {
        key2='2';
        }
if ((key1&0x08)==0)
        {
        key2='3';
        }
if ((key1&0x10)==0)
        {
        key2='4';
        }
if (key2==' ')
        {
        Pc1=0xfd;
        key1=waitkey();
        if ((key1&0x01)==0)
           {
           key2='5';
           }
        if ((key1&0x02)==0)
           {
           key2='6';
           }
        if ((key1&0x04)==0)
           {
           key2='7';
           }
        if ((key1&0x08)==0)
           {
           key2='8';
           }
        if ((key1&0x10)==0)
           {
           key2='9';
           }
        }
if (key2==' ')
        {
        Pc1=0xfb;
        key1=waitkey();
        if ((key1&0x01)==0)
           {
           key2='a';
           }
        if ((key1&0x02)==0)
           {
           key2='b';
           }
        if ((key1&0x04)==0)
           {
           key2='c';
           }
        if ((key1&0x08)==0)
           {
           key2='d';
           }
        if ((key1&0x10)==0)
           {
           key2='e';
           }
        }
if (key2==' ')
        {
        Pc1=0xf7;
        key1=waitkey();
        if ((key1&0x01)==0)
           {
           key2='f';
           }
        if ((key1&0x02)==0)
           {
           key2='g';
           }
        if ((key1&0x04)==0)
           {
```

```
                key2='h';
                }
            if ((key1&0x08)==0)
                {
                key2='i';
    }
            if ((key1&0x10)==0)
                {
                key2='j';
                }
            }

if (key2==' ')
            {
            Pc1=0xef;
            key1=waitkey();
            if ((key1&0x01)==0)
                {
                key2='k';
                }
            if ((key1&0x02)==0)
                {
                key2='l';
                }
            if ((key1&0x04)==0)
                {
                key2='m';
                }
            if ((key1&0x08)==0)
                {
                key2='n';
    }
            if ((key1&0x10)==0)
                {
                key2='o';
                }
            }
if (key2==' ')
            {
            Pc1=0xdf;
            key1=waitkey();
            if ((key1&0x01)==0)
                {
                key2='p';
                }
            if ((key1&0x02)==0)
                {
                key2='q';
                }
            if ((key1&0x04)==0)
                {
                key2='r';
                }
            if ((key1&0x08)==0)
                {
                key2='s';
    }
            if ((key1&0x10)==0)
                {
                key2='t';
                }
            }
        return (key2);
}

keybord(void)
{
 xdata unsigned char key2;
 data unsigned int tab;
 key2=inkey();
 if (key2=='0')
    {
    string_to_lcd(1,16,"4");
    tab=4;
    }
 else if (key2=='1')
    {
    string_to_lcd(1,16,"1");
    tab=1;
    }
 else if (key2=='3')
    {
    string_to_lcd(1,16,"7");
    tab=7;
    }
 else if (key2=='5')
```

```
    {
    string_to_lcd(1,16,"5");
    tab=5;
    }
  else if (key2=='6')
    {
    string_to_lcd(1,16,"2");
    tab=2;
    }
  else if (key2=='2')
    {
    string_to_lcd(1,16,"9");
    tab=9;
    }
  else if (key2=='8')
    {
    string_to_lcd(1,16,"8");
    tab=8;
    }
  else if (key2=='d')
    {
    string_to_lcd(1,16,"OK");
    tab=13;
    }
  else if (key2=='f')
    {
    string_to_lcd(1,16,"3");
    tab=3;
    }
  else if (key2=='g')
    {
    string_to_lcd(1,16,"0");
    tab=0;
    }
  else if (key2=='i')
    {
    string_to_lcd(1,16,"6");
    tab=6;
    }
 return tab;
}

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


main(void)
{
data unsigned char key,screen[5];
data unsigned int point,wait;
delay(1000);
Pcon1=0x81;
init_lcd();
init_serial();
delay(1000);
key=inkey();
if(key=='q')
 {
 Pa1=0xff;
 while(1)
   {
   init_lcd();
  string_to_lcd(1,1,"Sent to table:");
  while(wait!=13)
    {
    wait=keybord();
    if(wait!=13)
      point=wait;
    }
   init_lcd();
    string_to_lcd(1,1,"Sending to table");
    delay(20000);
    printf("%d",point);
    sprintf(screen,"%d",point);
    string_to_lcd(2,7,screen);

    wait=inkey();
    }
  }
}
```

ประวัติผู้เขียน

ชื่อ นายกิตินันต์ อยู่เสนาสน์

วัน/เดือน/ปีเกิด 8 กรกฎาคม 2522

ภูมิลำเนา 17/4 หมู่ 5 ต.นาเฉลียง อ.เมือง จ.เพชรบูรณ์

ประวัติการศึกษาระดับประถมศึกษา โรงเรียนลาซาลโชติรวี จ.นครสวรรค์

ระดับมัธยมศึกษา โรงเรียนลาซาลโชติรวี จ.นครสวรรค์

ปัจจุบัน กำลังศึกษาอยู่ ณ สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จ.พิษณุโลก

ชื่อ นายอธิปัตย์ ล้อวงศ์งาม

วัน/เดือน/ปีเกิด 17 สิงหาคม 2522

ภูมิลำเนา 378 ต.ท่าเรือ อ.ท่ามะกา จ.กาญจนบุรี

ประวัติการศึกษาระดับอนุบาลศึกษา โรงเรียนอุดมวิทยา จ.ราชบุรี

ระดับประถมศึกษา โรงเรียนอุดมวิทยา จ.ราชบุรี

ระดับมัธยมศึกษา โรงเรียนเทพศิรินทร์ จ.กรุงเทพมหานครฯ

ปัจจุบัน กำลังศึกษาอยู่ ณ สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จ.พิษณุโลก

รางวัลที่เคยได้รับ 1. เหรียญเงินกีฬาคาราเต้ Gojukai All Thailand ปี 2536 ณ มหาวิทยาลัย ABAC ประเภทบุคคลเดี่ยวไม่จัดน้ำหนัก

2. เหรียญเงินกีฬาคาราเต้ Gojukai All Thailand ปี 2544 ณ ศาลาเฉลิมพระเกียรติ จ.กาญจนบุรี ประเภทบุคคลเดี่ยวไม่จัดน้ำหนัก

ทุนการศึกษาที่เคยได้รับ ได้รับทุนการศึกษาไปเรียน ณ เมืองชาลส์ตัน รัฐเซาท์คาโรไลน่าประเทศสหรัฐอเมริกา เป็นเวลา 1 ปี (พ.ศ. 2537 – พ.ศ. 2538)

ชื่อ นายเริงชัย คงมี

วัน/เดือน/ปีเกิด 24 ตุลาคม 2522

ภูมิลำเนา 278/6 หมู่ 5 ต.ทุ่งทอง อ.หนองบัว จ.นครสวรรค์

ประวัติการศึกษาระดับประถมศึกษา โรงเรียนหนองคู่ จ.นครสวรรค์

ระดับมัธยมศึกษา โรงเรียนนครสวรรค์ จ.นครสวรรค์

ปัจจุบัน กำลังศึกษาอยู่ ณ สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จ.พิษณุโลก