ระบบควบคุมการเปิดปิดสวิตซ์ภายในบ้านสั่งงานผ่านโทรศัพท์

TELEPHONE BASED CONTROL OF ELECTRONIC SWITCHES

นายจักรกฤษ เพ็งภู่ รหัส 46361580

นายชัยณรงค์ วัฒนาพงศ์สกุล รหัส 46361606

i 508 1142 ค.2

ห้องสมุดคณะวิศวกรรมศาสตร์
รับที่รับ 17 พ.ย. 2559
เลขทะเบียน 5000112
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร

ปร.
9216ร
2549

ปริญญานิพนธ์ เป็นส่วนหนึ่งของหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	ระบบควบคุมการเปิดปิดสวิตซ์ภายในบ้านสั่งงานผ่านโทรศัพท์	
ผู้ดำเนินโครงงาน	นายจักรกฤษ เพ็งภู่	รหัส 46361580
	นายชัยณรงค์ วัฒนาพงศ์สกุล	รหัส 46361606
อาจารย์ที่ปรึกษา	ดร.สมยศ เกียรติวนิชวิไล	
สาขาวิชา	วิศวกรรมไฟฟ้า	
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์	
ปีการศึกษา	2549	

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(ดร.สมยศ เกียรติวนิชวิไล)

.....กรรมการ
(ดร.สมพร เรืองสินชัยวานิช)

.....กรรมการ
(ดร. ชัยรัตน์ พินทอง)

หัวข้อโครงงาน	ระบบควบคุมการเปิดปิดสวิตซ์ภายในบ้านสั่งงานผ่านโทรศัพท์
ผู้ดำเนินโครงงาน	นายจักรกฤษ เพ็งภู่ รหัสนิสิต 46361580
	นายชัยณรงค์ วัฒนาพงศ์สกุล รหัสนิสิต 46361606
อาจารย์ที่ปรึกษา	ดร.สมยศ เกียรติวนิชวิไล
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2549

บทคัดย่อ

การออกแบบการควบคุมระบบไฟฟ้าด้วยโทรศัพท์มีวัตถุประสงค์เพื่อให้สามารถควบคุมเครื่องใช้ไฟฟ้าภายในบ้าน ซึ่งนำไปควบคุมอุปกรณ์ไฟฟ้าในระยะไกลได้

โครงงานนี้สามารถที่จะควบคุมได้โดยการใช้การส่งสัญญาณจากมือถือแล้วผ่านชุมสายเข้าสู่คู่สายโทรศัพท์เพื่อสั่งให้ทำงาน การทำงานอาศัยการเปลี่ยนสัญญาณโทรศัพท์หรือสัญญาณ DTMF โดยอุปกรณ์ถอดรหัสความถี่จากการกดแป้นคีย์โทรศัพท์เป็นเลขฐานสอง แล้วส่งต่อไปยังไมโครคอนโทลเลอร์ที่ทำงานผ่านวงจรขับรีเลย์ เพื่อควบคุมอุปกรณ์ไฟฟ้าในส่วนเอาท์พุทต่อไป ผลการทดสอบพบว่าเครื่องควบคุมการเปิด- ปิด สวิตซ์ภายในบ้านสั่งงานผ่านโทรศัพท์สามารถควบคุมอุปกรณ์ไฟฟ้าได้

Project Title	Telephone based control of electronic switches	
Name	Mr. Jukkrit Phengphu	ID 46361580
	Mr. Chainarong Watnaphongsakul	ID 46361606
Project Advisor	Somyot Kaitwanidvilai, Ph.D.	
Major	Electrical Engineering	
Department	Electrical and Computer Engineering	
Academic Year	2006	

ABSTRACT

This project is to make an electric a lines control . The control of electric appliances in project can done . The control set consists and electric circuit mobile .

This project can control by send the signal of mobile and through telephone exchange in line telephone in receiver for control can be done . The controller change telephone signal or DTMF signal to binary code decimal by a decoder circuit . A binary code decimal signal to a microcontroller is through drive relay circuit for control electric appliances to pass it to the output unit . The electric appliance can controlled via the telephone

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี โดยความช่วยเหลือจากหลายๆ ท่านด้วยกัน ผู้จัดทำขอถือโอกาสนี้ ขอกราบขอบพระคุณ อาจารย์ พนัส นัณฤทธิ์ และ ดร.สมยศ เกียรติวนิชวิไล ซึ่งเป็นอาจารย์ที่ปรึกษา และคณะกรรมการสอบโครงงานทุกท่านที่ได้ให้คำปรึกษาชี้แนะแนวทาง และข้อคิดเห็นต่างๆ ในการแก้ปัญหาที่เป็นประโยชน์อย่างสูงในการทำโครงงานนี้ให้สำเร็จลุล่วงได้ด้วยดี อาจารย์ ซึ่งได้ให้ความช่วยเหลือในด้านอุปกรณ์ตลอดจนคำปรึกษาชี้แนะแนวทางต่างๆ เกี่ยวกับการทำโครงงานนี้เป็นอย่างสูง

ท้ายนี้ผู้จัดทำโครงงานขอกราบขอบพระคุณบิดา มารดา ที่คอยสนับสนุนในด้านการเงิน และให้กำลังใจแก่ผู้จัดทำเสมอมาจนสำเร็จการศึกษา

คณะผู้จัดทำโครงงาน
นายจักกฤษ เพ็งภู่
นายชัยณรงค์ วัฒนาพงศ์สกุล

สารบัญ

หน้า

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

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ 1
1.3 ขอบข่ายของโครงงาน 1
1.4 ขั้นตอนการดำเนินงาน 1
1.5 แผนการดำเนินโครงงาน 2
1.6 ผลที่คาดว่าจะได้รับ 2
1.7 งบประมาณของโครงงาน 2

บทที่ 2 หลักการและทฤษฎี 3
2.1 รหัสสัญญาณโทรศัพท์ 3
2.1.1 ชนิดของรหัสสัญญาณ 3
2.1.2 การเข้ารหัส 4
2.1.3 การถอดรหัส (DECODER) 7
2.2 การถอดรหัสโดย IC MT8870 8
2.2.1 คุณสมบัติของ MT8870 8
2.2.2 โครงสร้างภายในของ MT8870 8
2.2.3 หน้าที่และการทำงานของแต่ละขาของ IC MT8870 10

สารบัญ (ต่อ)

หน้า

2.2.4 ฟังก์ชันการทำงานภายในของ MT8870 10
2.2.4.1 ภาคกรองสัญญาณความถี่ 10
2.2.4.2 ภาคถอดรหัส 10
2.2.4.3 ภาคตรวจสอบสัญญาณ 11
2.2.4.4 ภาคขยายความแตกต่าง 13
2.2.4.5 ภาคกำเนิดความถี่ 13
2.3 ไมโครคอนโทรเลอร์ AT89C2051 13
2.3.1 คุณสมบัติของไอซีไมโครคอนโทรลเลอร์ AT89C2051 13
2.3.2 การทำงานของไมโครคอนโทรลเลอร์ AT89C2051 14
2.3.2.1 การทำงานของแต่ละขา 15
2.3.2.2 การจักสรรหน่วยความจำข้อมูล 15
2.3.2.3 รีจีสเตอร์ใน AT89C2051 16

บทที่ 3 การเขียนโปรแกรมภาษาแอสเซมบลี 19
3.1 รูปแบบของภาษาแอสเซมบลี 19
3.2 Assemble-time Expression Evalution 24

บทที่ 4 วงจรรวมและหลักการทำงาน 29
4.1 ส่วน INPUT 29
4.1.1 วงจรรับโทรศัพท์อัตโนมัติ 29
4.1.2 วงจรถอดรหัสสัญญาณ DTMF 30
4.2 ส่วน CONTRO PROCESSING UNIT 32
4.3 ส่วน OUTPUT 33
4.4 วงจรภาคจ่ายไฟ 33
4.5 วงจรรวมและหลักการทำงาน 34

สารบัญ (ต่อ)

หน้า

บทที่ 5 การทดลองการทำงาน 35

5.1 การทดสอบวงจรตอบรับโทรศัพท์อัตโนมัติ 35

5.2 การทดสอบวงจรถอดรหัสสัญญาณ DTMF เป็น Binary Code 35

5.3 การทดลองเกี่ยวกับวงจรจ่ายไฟ 36

บทที่ 6 สรุปและข้อเสนอแนะ 37

6.1 สรุปผลการดำเนินการ 37

6.2 สรุปผลการทดลอง 37

6.3 ปัญหาที่พบในการทดลอง 37

6.4 ข้อเสนอแนะ 38

เอกสารอ้างอิง 39

ภาคผนวก 40

ประวัติผู้เขียนโครงงาน

สารบัญตาราง

ตารางที่ หน้า

2.1 ตารางความถี่ไดอัลโทนมัลติฟรีเควนซี่ 3
2.2 แสดงค่ารหัสที่ถอดได้จากความถี่ 11
3.1 คำสั่งกระโดดแบบมีเงื่อนไข 23
3.2 ตัวดำเนินการทางคณิตศาสตร์ 25
3.3 ตัวดำเนินการทางลอจิก 26
3.4 ตัวดำเนินการพิเศษ 26
3.5 ตัวดำเนินการเปรียบเทียบ 27
3.6 ลำดับการทำงานของตัวดำเนินการ 28
4.1 เลขฐานสอง และเลขฐานสิบ จากการกดโทรศัพท์ 31
5.1 เลขฐานสอง และเลขฐานสิบ จากการกดโทรศัพท์ 36

สารบัญรูป

รูปที่		หน้า
2.1	แสดงลักษณะการต่อโทรศัพท์กับชุมสาย	4
2.2	แสดงการต่อปากพูดหูฟังเข้ากับวงจร	4
2.3	วงจรรวมเครื่องโทรศัพท์	5
2.4	ก. แสดงลักษณะแรงดันในสายโทรศัพท์	5
	ข. แสดงลักษณะของกระแสในสายโทรศัพท์	5
2.5	บล็อกไดอะแกรม พื้นฐานของไอซีสร้างรหัสพัลส์	6
2.6	แสดงวงจรพื้นฐานของเครื่องโทรศัพท์แบบ DTMF	7
2.7	แสดงตำแหน่งสวิตช์เลือกความถี่ต่าง ๆ	7
2.8	วงจรถอดรหัสโทน	8
2.9	แสดงโครงสร้างภายในของ MT8870	9
2.10	แสดงโครงสร้างภายนอกของ MT8870	9
2.11	แสดงไทรมมิ่งไดอะแกรม	12
2.12	แสดงบล็อกไดอะแกรมของ AT89C2051	14
2.13	แสดงรีจีสเตอร์ใน AT89S8252	16
4.1	BLOCK DIAGRAM	29
4.2	วงจรรับโทรศัพท์อัตโนมัติ	30
4.3	วงจรถอดรหัสสัญญาณ DTMF	31
4.4	แสดงขาต่าง ๆ ของ AT89C2051	32
4.5	วงจรการเชื่อมต่อระหว่าง AT89C2051 กับอุปกรณ์อื่น ๆ	32
4.6	วงจรการทำงานของรีเลย์	33
4.7	วงจรภาคจ่ายไฟ	34
4.8	วงจรรวม	34
5.1	การต่อ output เข้ากับหลอด LED	35

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ปัจจุบันนี้เครื่องใช้ไฟฟ้าเครื่องอำนวยความสะดวกต่างๆในชีวิตประจำวันของเรา ซึ่งแน่นอนว่าการใช้ไฟฟ้าต้องมีมากตามไปด้วยเช่นกัน สำหรับคนขี้ลืมแล้วนั้นอาจจะทำให้เปลืองค่าใช้ไฟฟ้าเช่นกัน ถ้าหากเปิดเครื่องใช้ไฟฟ้าแล้วทิ้งไว้ หรือ มีธุระเร่งรีบ แล้วเปิดเครื่องใช้ไฟฟ้าทิ้งไว้ในบ้านแล้วรีบออกไปทำธุระแน่นอนว่าจะต้องเปลืองค่าไฟฟ้าไปโดยเปล่าประโยชน์

ชุดควบคุมเครื่องใช้ไฟฟ้าผ่านทางโทรศัพท์ ปิดเครื่องใช้ไฟฟ้าที่เปิดทิ้งไว้ หรือ ต้องการเปิดเครื่องใช้ไฟฟ้าก็สามารถทำได้โดยไม่จำเป็นว่าจะอยู่ที่ไหนเพียงแค่มีโทรศัพท์ ก็สามารถโทรมาเพื่อสั่งงานชุดควบคุมได้

1.2 วัตถุประสงค์

1.2.1 ช่วยอำนวยความสะดวกในการ เปิด-ปิด เครื่องใช้ไฟฟ้า

1.2.2 ช่วยประหยัดพลังงานไฟฟ้า

1.3 ขอบข่ายของงาน

1.3.1 ศึกษาการทำงานของระบบโทรศัพท์รูปแบบสัญญาณโทรศัพท์

1.3.2 ศึกษาวงจรการถอดรหัสสัญญาณโทรศัพท์

1.3.3 ศึกษาการทำงานโดยการควบคุมจากไมโครคอนโทรลเลอร์

1.3.4 ออกแบบวงจรรวมที่ใช้ควบคุมอุปกรณ์ไฟฟ้า

1.4 วิธีการดำเนินการ

1.4.1 ศึกษาและเก็บข้อมูลเกี่ยวกับหลักการทำงานของระบบโทรศัพท์และไมโครคอนโทรเลอร์

1.4.2 ออกแบบระบบควบคุม โดยใช้ไมโครคอนโทรเลอร์

1.4.3 สร้างระบบควบคุมที่ออกแบบขึ้นให้เหมาะสมกับการใช้งาน

1.4.4 ทดสอบการทำงาน

1.4.5 เก็บข้อมูลการปรับปรุงคุณภาพระบบควบคุมที่ออกแบบขึ้น

1.4.6 เปรียบเทียบข้อมูลที่ได้จากการทดลองกับทฤษฎี

1.4.7 สรุปผลการทดลองและจัดทำรูปเล่มโครงงาน

1.5 ตารางการดำเนินงาน

กิจกรรม	พ.ย. 2548	ธ.ค. 2548	ม.ค. 2549	ก.พ. 2549	มี.ค. 2549	เม.ย. 2549	พ.ค. 2549	มิ.ย. 2549	ก.ค. 2549	ส.ค. 2549	ก.ย. 2549
1. เขียน โครงการทำงาน	↔										
2. รวบรวมข้อมูลและเอกสาร		↔	↔								
3. ศึกษาระบบการทำงานของโทรศัพท์ ไมโครคอนโทรลเลอร์ และวงจรร่วม ต่าง ๆ วงจรแปลงสัญญาณ			↔	↔							
4. ศึกษาภาษาแอสเซมบลีและเขียน					↔	↔					
โปรแกรม											
5. ลงลายวงจรบัดกรี							↔	↔			
6. ทำการทดสอบและวิเคราะห์								↔	↔		
7. ทำโครงงานฉบับร่าง									↔		
8. ปรับปรุงแก้ไขโครงงานให้สมบูรณ์										↔	
9. ส่งโครงงานฉบับสมบูรณ์											↔

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถใช้งานอุปกรณ์ไฟฟ้าได้อย่างมีประสิทธิภาพ

1.6.2 สามารถใช้ประหยัดพลังงานได้

1.6.3 ประหยัดเวลา และประหยัดค่าใช้จ่าย

1.6.4 เพื่อเป็นแนวทางในการสร้างเทคโนโลยีใหม่

1.7 งบประมาณ

1.7.1 ค่าอุปกรณ์ ต่างๆ	1300 บาท
1.7.2 ค่าแผ่นซีดี	20 บาท
1.7.3 ค่าถ่ายเอกสารและค่าเข้าเล่มโครงงาน	680 บาท
รวมเป็นเงิน	2,000 บาท

บทที่ 2
หลักการและทฤษฎีพื้นฐาน

ในบทนี้จะกล่าวถึงทฤษฎีต่าง ๆ ที่ได้ทำการศึกษา เพื่อใช้ประกอบการทำโครงงานนี้ เนื้อหาในบทนี้จะประกอบด้วยรายละเอียดดังต่อไปนี้

2.1 รหัสสัญญาณโทรศัพท์

2.1.1 ชนิดของรหัสสัญญาณ รหัสสัญญาณที่เครื่องโทรศัพท์สร้างขึ้นมานี้ มีอยู่ 2 ชนิด คือ

สัญญาณแบบไดอัลพัลส์

สัญญาณแบบไดอัลพัลส์ (DP : Dial Pulse) รหัสสัญญาณที่เป็นแบบพัลส์นี้ จะมีลักษณะเป็นพัลส์ ช่วงบวกเท่านั้น

สัญญาณแบบไดอัลโทนมัลติฟรีเควนซี

สัญญาณแบบไดอัลโทน (DTMF : Dial Tone Multi Frequency) เครื่องที่ให้รหัสเป็นไดอัลโทนนี้จะเป็นเครื่องแบบกดปุ่มเท่านั้น เครื่องแบบหมุนไม่สามารถทำได้ลักษณะของสัญญาณไดอัลโทนจะเป็นความถี่ 2 ความถี่วิ่งคู่กันไปที่ชุมสาย โดยความถี่ที่เกิดขึ้นนี้จะเป็นค่ามาตรฐานสากล ตามตารางที่ 2.1

ตารางที่ 2.1 ตารางความถี่ไดอัลโทนมัลติฟรีเควนซี่

Hz	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	O	#

ตัวอย่าง

ถ้าเรากดเลข 1 จะได้ความถี่ 697 กับ 1209

ถ้าเรากดเลข 5 จะได้ความถี่ 770 กับ 1336

ถ้าเรากดเลข * จะได้ความถี่ 941 กับ 1209

ถ้าเรากดเลข # จะได้ความถี่ 941 กับ 1477

2.1.2 การเข้ารหัส

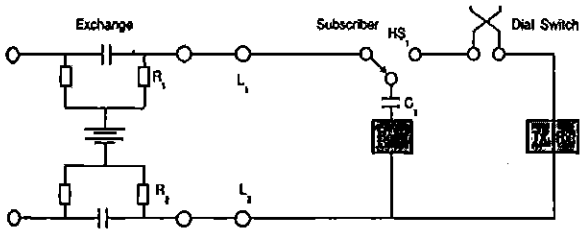
การเข้ารหัส (Encoder) ของเครื่องโทรศัพท์ก็จะสอดคล้องกับชนิดของรหัสไดอัล ซึ่งได้กล่าวมาแล้ว โดยแบ่งการเข้ารหัสได้ 2 แบบ คือ การเข้ารหัสแบบพัลส์ (Pulse Encoder) และการเข้ารหัสแบบโทน (Tone Encoder)

1. การเข้ารหัสแบบพัลส์

การเข้ารหัสแบบพัลส์ สามารถแยกวิธีการเข้ารหัสตามชนิดของเครื่องได้ 2 วิธี คือ การเข้ารหัสของเครื่องแบบหมุน (Rotary Encoder)และ การเข้ารหัสของเครื่องแบบกดปุ่ม (Push Bottom Encoder)

การเข้ารหัสของเครื่องแบบหมุน

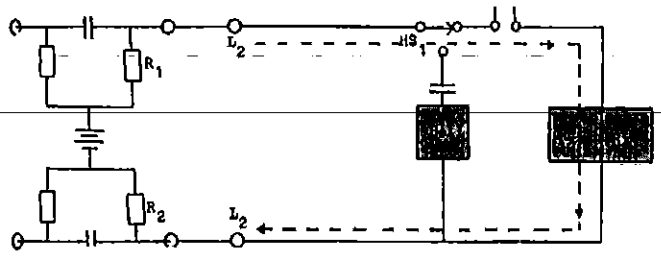
โดยทั่วไป เครื่องแบบหมุนจะไม่มีวงจรอิเล็กทรอนิกส์ที่ยุ่งยากมาก จึงไม่ได้ใช้วงจรอิเล็ก- ทรอนิกส์เป็นตัวเข้ารหัสโดยตรง แต่จะอาศัยการทำงานของระบบกลไกที่หน้าปัดเป็นตัวเข้ารหัสแทน



รูปที่ 2.1 แสดงลักษณะการต่อโทรศัพท์กับชุมสาย

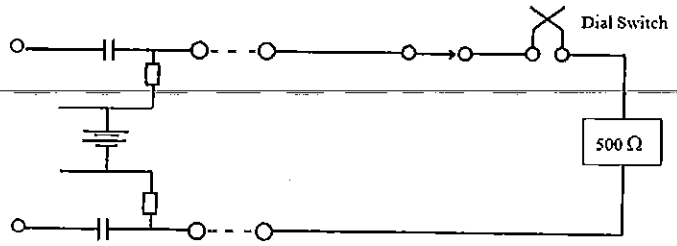
จากรูปที่ 2.1 เครื่องโทรศัพท์จะยังไม่ได้ถูกยกหูสายโทรศัพท์จะต่ออยู่กับวงจรกระดิ่ง ซึ่งมี C_1 ต่อกั้นอยู่ ดังนั้น ไฟกระแสตรง 48 โวลต์ จึงไม่สามารถผ่านเข้าในวงจรกระดิ่งได้ กระแสไม่ไหล ไฟในสายโทรศัพท์จึงยังคงมี 48 โวลต์ ตามปกติ

แต่เมื่อมีการยกหูฮุกสวิตช์จะตัดสายออกจากกระดิ่งแล้วต่อสายเข้ากับวงจรปากพูดหูฟังตามรูปที่ 2.2

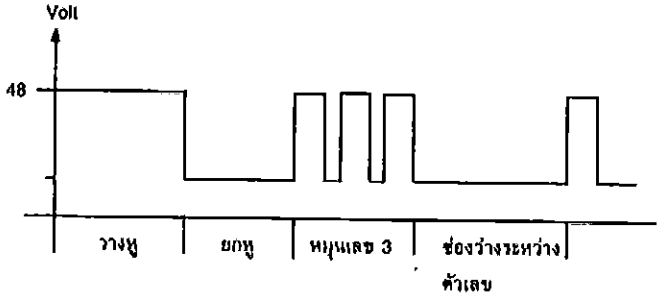


รูปที่ 2.2 แสดงการต่อปากพูดหูฟังเข้ากับวงจร

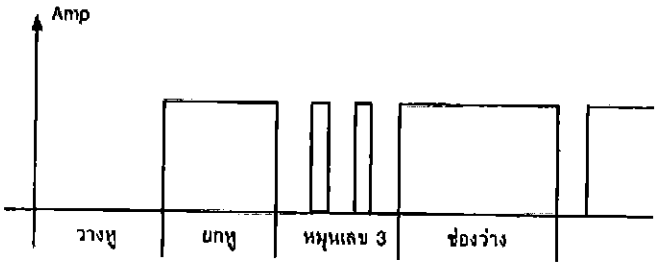
โดยทั่วไป ความต้านทานรวมของวงจรโทรศัพท์ จะประมาณ 500 โอห์ม ดังนั้น เมื่อยกหูขึ้นไฟดีซี 48 โวลต์ จากชุมสายจึงตกคร่อม R_1 R_2 และ 500 โอห์ม (เครื่องโทรศัพท์) รวมทั้งตกคร่อมความต้านทานของสายด้วย ไฟที่ตกคร่อมเครื่องโทรศัพท์ขณะยกหู จะมีประมาณ 6-12 โวลต์ แล้วแต่สายจะยาวมากน้อยเท่าใด (ถ้าไฟต่ำกว่า 5-6 โวลต์ เครื่องโทรศัพท์อาจไม่ทำงาน โดยเฉพาะเครื่องรุ่นใหม่ ๆ ที่มีวงจรอิเล็กทรอนิกส์ภายใน)



รูปที่ 2.3 วงจรรวมเครื่องโทรศัพท์



(ก)



(ข)

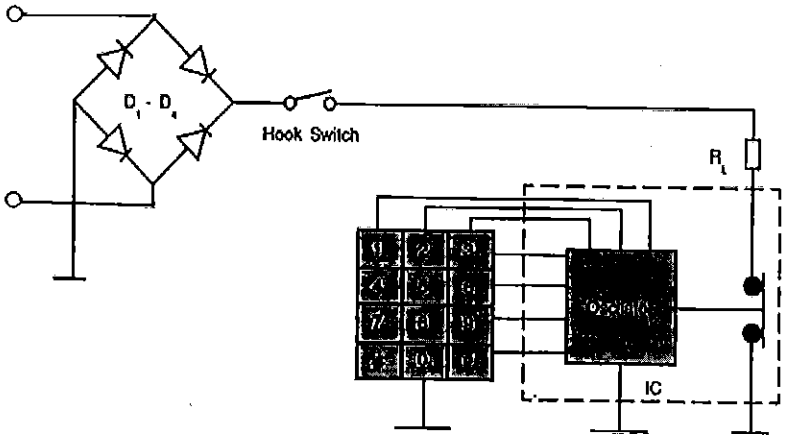
รูปที่ 2.4 ก. แสดงลักษณะแรงดันในสายโทรศัพท์
ข. แสดงลักษณะของกระแสในสายโทรศัพท์

จากรูปที่ 2.3 จะเห็นว่าสวิตช์ไดอัล(Dial Switch) ต่ออนุกรมอยู่กับสายโทรศัพท์ ดังนั้นเมื่อหมุนโทรศัพท์ในช่วงที่หน้าปัดหมุนกลับ จะมีกระเดื่องไปเขี่ยสวิตช์ไดอัลให้เปิดวงจร (Open Circuit) จำนวนครั้งของการตัดวงจรจะเท่ากับตัวเลขที่หมุน ในช่วงที่ถูกตัดวงจร ก็จะไม่มีกระแสไหลเป็น

ช่วงๆ ตามรูปที่ 2.4 ข แรงดันไฟฟ้าก็จะเพิ่มขึ้นเป็นช่วงๆ (เพราะไม่มีโหลด 500 โอห์ม) ตามรูปที่ 2.4 ก ส่วนคาบเวลาของพัลส์ต่างๆนั้น จะถูกตั้งด้วยความเร็วของลวดสปริงที่ทำหน้าที่ ดึงหน้าปัดกลับและเบรก โดยเบรกจะเป็นตัวลดความเร็วของสปริง

ทางด้านชุมสายโทรศัพท์ก็จะมีวงจรถอดรหัสพัลส์ที่ส่งมานี้ เช่น อาจเป็นวงจรนับ (Counter) พัลส์หรืออาจมีรีเลย์ (Relay) มาต่ออนุกรมกับสายโทรศัพท์ เพื่อให้ทำงานตามการไหลหรือไม่ไหลของกระแส

การเข้ารหัสของเครื่องแบบกดปุ่ม

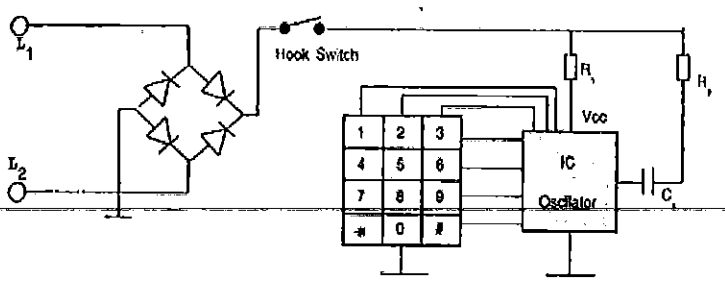


รูปที่ 2.5 บล็อกไดอะแกรม พื้นฐานของไอซีสร้างรหัสพัลส์

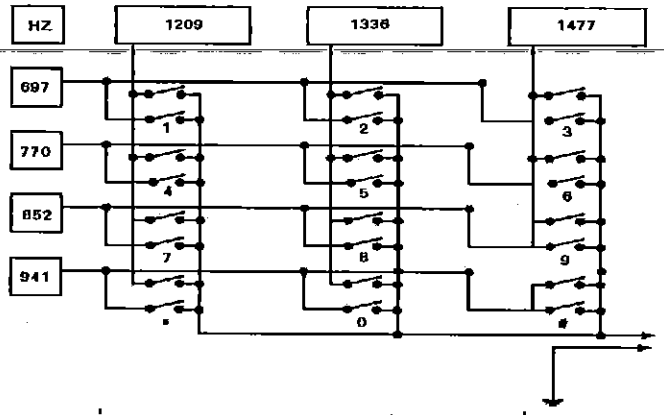
การเข้ารหัสของเครื่องแบบกดปุ่ม ภายในเครื่องจะมีไอซีสร้างพัลส์ ดังแสดงตาม รูปที่ 2.5 เมื่อยกหู ฮุกสวิตช์จะต่อไฟเข้ามา ทำให้มีกระแสไหลผ่าน R_L ผ่านสวิตช์ ในไอซีแล้วลงกราวด์ เพราะฉะนั้นเมื่อสายมีโหลด ไฟในสายก็จะตกลงจาก 48 โวลต์ เหลือ 5-12 โวลต์ (โดยปกติแล้วจะมีวงจรอื่น ๆ อีกขนานอยู่กับ R_L แต่จะมีความต้านทานรวมมากกว่า R_L มาก) เมื่อกดเลขหมายใด ๆ วงจรออสซิลเลเตอร์ก็จะสร้างพัลส์ขึ้นมา โดยจำนวนพัลส์จะเท่ากับค่าของตัวเลขนั้น ๆ พัลส์ที่สร้างขึ้นมาก็จะมาทำให้สวิตช์เปลี่ยนสภาวะจากปิด (Close) เป็นเปิด (Open) ตามจังหวะและจำนวนพัลส์ที่เข้ามา เมื่อสวิตช์เปิดวงจรก็จะไม่มีกระแสไหล R_L ก็ไม่ได้เป็นโหลดของสายทำให้แรงดันไฟฟ้าในสายเพิ่มขึ้นตามจังหวะเปิดและปิดของสวิตช์ นั่นคือเกิดพัลส์ขึ้นมาตามค่าของตัวเลข หรือถ้าจะมองในด้านของกระแส ช่วงสวิตช์เปิดและปิด ก็จะทำให้มีกระแสไหลเป็นช่วง ๆ เช่นกัน ส่วนคาบเวลาของพัลส์ ก็กำหนดได้ที่ภาคออสซิลเลเตอร์ซึ่งไปควบคุมสวิตช์ลักษณะพัลส์ที่ได้แสดงตามรูปที่ 2.4

2. การเข้ารหัสแบบโทน

การเข้ารหัสแบบโทนนี้ จะมีเฉพาะเครื่องแบบกดปุ่มเท่านั้น ดังแสดงในรูปที่ 2.6 และรูปที่ 2.7



รูปที่ 2.6 แสดงวงจรพื้นฐานของเครื่องโทรศัพท์แบบ DTMF



รูปที่ 2.7 แสดงตำแหน่งสวิตช์เลือกความถี่ต่าง ๆ

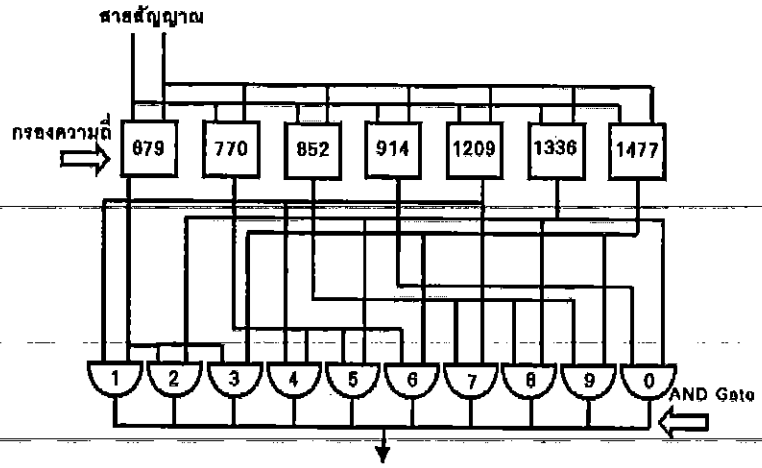
จากรูปที่ 2.6 จะมีไอซีผลิตความถี่ขึ้นมาตาม รูปที่ 2.7 โดยจะมีการผลิตความถี่ตลอดเวลาที่ยกหูและมีไฟมาจ่ายให้วงจร เมื่อกดปุ่มสวิตช์หมายเลขต่าง ๆ ก็จะได้ความถี่ออกไปหมายเลขละ 2 ความถี่ เช่น กดหมายเลข 5 จะได้ความถี่ 770 เฮิรตซ์และ 1,336 เฮิรตซ์ผ่านสวิตช์ออกไปได้ ปุ่มหมายเลขอื่นๆ ก็เช่นเดียวกัน

2.1.3 การถอดรหัส (DECODER)

การถอดรหัสแบ่งออกได้ 2 แบบ คือ การถอดรหัสแบบพัลส์ (Pulse Decoder) และ การถอดรหัสแบบโทน (Tone Decoder)

การถอดรหัสแบบพัลส์ วงจรถอดรหัสนี้จะอยู่ที่ชุมสายโทรศัพท์ วิธีถอดรหัสแบบพัลส์ที่ง่ายที่สุดก็คือ ใช้วงจรนับ (Counter) มานับจำนวนพัลส์ที่ส่งเข้ามา เอาต์พุต (Output) ของวงจรนับก็คือตำแหน่งของผู้รับที่ต้องการ

การถอดรหัสแบบโทน ในการถอดรหัสแบบโทนนั้น เนื่องจากสัญญาณที่ส่งมา มีลักษณะเป็น 2 ความถี่วิ่งคู่กันมาที่ชุมสาย การถอดรหัสเราจะต้องเปลี่ยนความถี่ 2 ความถี่นั้นให้เป็นแรงดันหรือกระแส หรือสภาวะ 0 กับ 1 ให้ได้ ทั้งนี้ขึ้นอยู่กับเราจะนำเอาผลของการถอดรหัสนี้ไปใช้ทำอะไร วงจรที่จะได้อธิบายต่อไปนี้จะเป็นการเปลี่ยนความถี่ให้เป็น 0 กับ 1 เพื่อป้อนให้คอมพิวเตอร์ทำงาน



รูปที่ 2.8 วงจรถอดรหัสโทน

สมมติเรากดเลข 1 ก็จะมีความถี่ 697 เฮิรตซ์ กับ 1209 เฮิรตซ์ วิ่งคู่กันมาเข้าที่อินพุตของฟิลเตอร์ทุกตัว แต่จะมีเอาต์พุตออกมาเฉพาะฟิลเตอร์ที่ตรงกับความถี่ที่เข้ามาเท่านั้น นั่นคือจะมีเอาต์พุตออกจาก ฟิลเตอร์ 697 และ 1209 เท่านั้น จากฟิลเตอร์ 697 และ 1209 ก็จะต่อมาถึง เกต (Gate) หมายเลข 1 เมื่ออินพุตของ แอนด์เกต (AND Gate) หมายเลข 1 เป็น 1 ทั้ง 2 ก็จะให้เอาต์พุตเป็น 1 ออกมา และจากเอาต์พุตของ Gate ก็จะต่อไปให้คอมพิวเตอร์จัดการตามลำดับขั้นของโปรแกรมต่อไป เลขอื่น ๆ ก็เช่นเดียวกัน

2.2 การถอดรหัสโดย IC MT8870

โดยปกติแล้ว วงจรถอดรหัสตามรูปที่ 2.8 นี้จะมีชิพไอซีสำเร็จรูปขายตามท้องตลาด เช่น เบอร์ MT8870 ที่ออกแบบมาเพื่อเป็นตัวถอดรหัสนี้โดยเฉพาะ

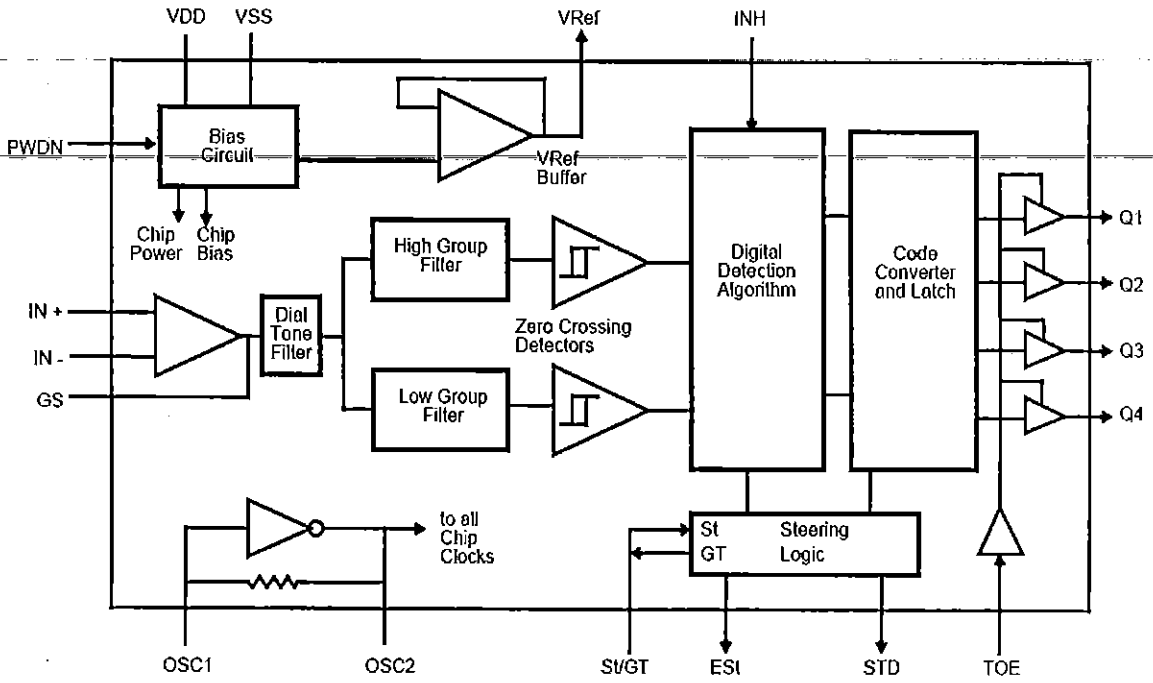
2.2.1 คุณสมบัติของ MT8870

- เป็นตัวรับและถอดรหัสความถี่
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการ์ดไทรม
- เป็นไอซีคุณภาพสูง

2.2.2 โครงสร้างภายในของ MT8870

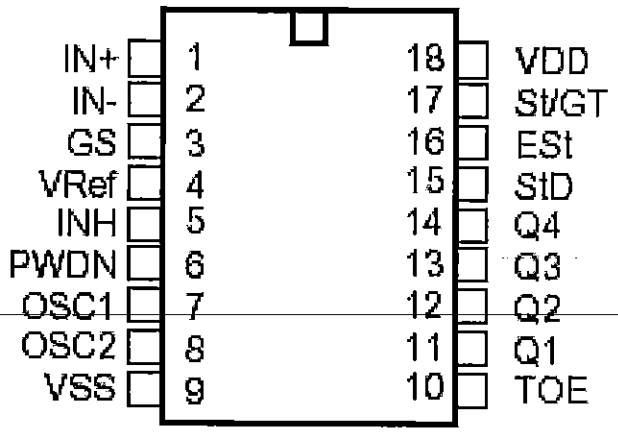
โครงสร้างภายในประกอบไปด้วยวงจรกรองความถี่ และ วงจรกรองถอดรหัสฟังก์ชันทางดิจิตอลเป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO-CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิตซ์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับ

ทางดิจิตอลเพื่อตรวจจับ และถอดรหัสทั้ง16 ความถี่ออกเป็นเลขฐานสองขนาด 4 บิตและเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาคอินพุตเป็นออปแอมป์ ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอกเอาต์พุตเป็นวงจรแลตซ์ 3 สถานะ ทำให้สามารถที่จะส่งสัญญาณ DTMF มีช่วงเวลาที่เท่ากันตลอดดูได้จากโครงสร้างภายในของ MT8870 รูปที่ 2.9



รูปที่ 2.9 แสดงโครงสร้างภายในของ MT8870

สำหรับลักษณะภายนอกของไอซี MT8870 มีรูปแบบและการใช้งานดูได้ในรูปที่ 2.10



18 PIN PLASTIC DIP/SOIC

รูปที่ 2.10 แสดงโครงสร้างภายนอกของ MT8870

2.2.3 หน้าที่และการทำงานของแต่ละขาของ IC MT8870

ขาที่ 1 เป็นขาขยายสัญญาณอินพุตแบบ non-inverting

ขาที่ 2 เป็นขาขยายสัญญาณอินพุตแบบ inverting

ขาที่ 3 เป็นขาขยายสัญญาณความแตกต่างแบบป้อนกลับ

ขาที่ 4 เป็นขาแรงดันอ้างอิง

ขาที่ 5 เป็นขาป้องกันสัญญาณอินพุตที่เป็น logic สูง

ขาที่ 6 เป็นขาป้องกันกำลังไฟฟ้าอินพุทเกินและoscillator

ขาที่ 7 เป็นขาอินพุทของ oscillator

ขาที่ 8 เป็นขาเอาท์พุตของ oscillator คริสตอลที่ใช้ต่อระหว่างขาอินพุท และ เอาท์พุทของ oscillator มีค่าเท่ากับ 3.579545MHz

ขาที่ 9 เป็นขา ground

ขาที่ 10 เป็นTone Output Enable (Input) ใช้คุม Q1-Q4 ให้เป็น H impedance

ขาที่ 11-14 เป็นขาข้อมูลใช้ในการส่งข้อมูลระหว่าง MT8870 และ microcontroller

ขาที่ 15 เป็นDelayed Steering Output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

ขาที่ 16 เป็นขาสัญญาณ Early steering output จะเป็นลอจิก 1 เมื่อมีการตรวจพบสัญญาณ tone pair

ขาที่ 17 เป็นขาสัญญาณ Steering input/Guard time output เป็นแบบสองทิศทาง มีการตรวจสอบ tone pair และ เปลี่ยนสถานะ output latch

ขาที่ 18 เป็นขาไฟบวก

2.2.4 ฟังก์ชันการทำงานภายในของ MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วน คือ

2.2.4.1 ภาคกรองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือ ช่วงความถี่สูงและช่วงความถี่ต่ำโดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิตซ์คาปาซิเตอร์ ซึ่งความถี่ที่แยกได้มี 2 ช่วง คือ ช่วงความถี่สูงและช่วงความถี่ต่ำ

2.2.4.2 ภาคถอดรหัส

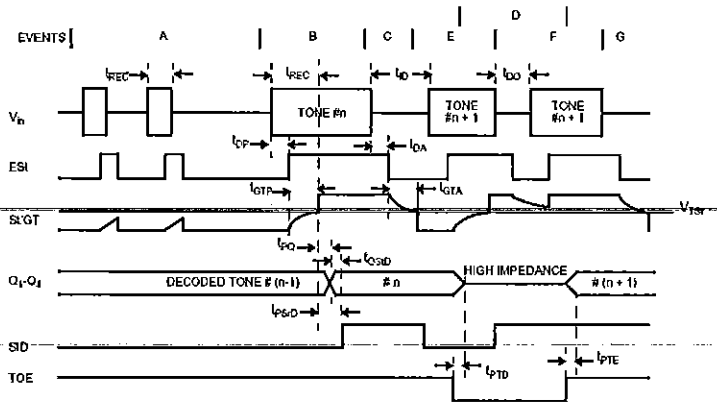
ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิตอลและมีการตรวจสอบความถี่ที่เข้ามาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสมเมื่อตรวจสอบว่าความถี่นั้นถูกต้องสัญญาณที่ขา Est ก็จะแอคทีฟ สำหรับค่ารหัสที่ถอดได้จากความถี่ต่างเป็นไปตามตารางที่ 2.2 แสดงค่ารหัสความถี่ที่ถอดได้

ตารางที่ 2.2 แสดงค่ารหัสที่ถอดได้จากความถี่

F_{LOW}	F_{HIGH}	DIGIT	D_3	D_2	D_1	D_0
697	1209	1	0	0	0	1
697	1336	2	0	0	1	0
697	1477	3	0	0	1	1
770	1209	4	0	1	0	0
770	1336	5	0	1	0	1
770	1447	6	0	1	1	0
852	1209	7	0	1	1	1
852	1336	8	1	0	0	0
852	1447	9	1	0	0	1
941	1336	0	1	0	1	0
941	1209	*	1	0	1	1
941	1477	#	1	1	0	0
697	1633	A	1	1	0	1
770	1633	B	1	1	1	0
852	1633	C	1	1	1	1
941	1633	D	0	0	0	0

2.2.4.3 ภาคตรวจสอบสัญญาณ

ก่อนที่จะมีการถอดรหัสความถี่ออกไปที่เอาต์พุต จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควร มิฉะนั้นวงจรนี้จะไม่รับ โดยจะถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา Est จะเป็น High นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามาเมื่อขา Est เป็น High ทำให้ Vc สูงขึ้นตัวเก็บประจุ C คายประจุทำให้แรงดัน Vc สูงขึ้นจนถึงค่าเทรชโอลต์ วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลขขนาด 4 บิต สำหรับคำว่า การ์ดไทม์ (gard time) นั้น หมายถึง ช่วงคาบเวลาของความถี่เข้ามา ซึ่งจะต้องนานเท่ากันหรือมากกว่าเวลาที่ตั้งไว้จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่าก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป



รูปที่ 2.11 แสดงไทมมิ่งไดอะแกรม

อธิบายการทำงานของไทมิ่งไดอะแกรม

A ตรวจพบความถี่เข้ามา แต่ควบเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน

B ความถี่ #n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแลตช์ไว้ที่เอาต์พุต

C จบความถี่ #n ช่วงห่างถูกต้องเอาต์พุตยังคงแลตช์อยู่จนกว่าจะได้ความถี่ที่ถูกต้องใหม่

D เอาต์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์

E ความถี่ #n + 1 ถูกตรวจพบ คาบเวลาถูกต้องความถี่ถูกถอดรหัสและแลตช์ไว้

F ความถี่ #n + 1 หายไปช่วงห่างไม่ถูกต้องเอาต์พุตยังคงแลตช์อยู่

สัญญาณต่าง ๆ

Vin สัญญาณความถี่ DTMF ที่เข้ามา

ESt Early Steering output ใช้แสดงความที่ถูกต้อง

St/GT Steering input/Guard Time output สำหรับต่อกับ RC ภายนอก

Q1-Q4 ถอดรหัสข้อมูลในรีจีสเตอร์

StD Delayed Steering Output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE Tone Output Enable (Input) ใช้คุม Q1-Q4 ให้เป็น high impedance

Trec คาบเวลานานที่สุดที่ตรวจพบสัญญาณความถี่ DTMF แล้วยังไม่ถูกต้อง

Trec คาบเวลาที่สั้นที่สุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

Tid เวลาสั้นสุดระหว่างสัญญาณ DTMF ที่ถูกต้องสองสัญญาณ

Tdo เวลานานที่สุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

Tdp เวลาที่ใช้ในการตรวจสอบสัญญาณความถี่ DTMF ที่ถูกต้อง

Tda เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DTMF ที่ถูกต้อง

Tgtp การ์ดไทม์ของการปรากฏความถี่ DTMF

Tgta การ์ดไทม์ของการหายไปของความถี่ DTMF

2.2.4.4 ภาคขยายความแตกต่าง

วงจรส่วนอินพุตของ MT8870 เป็นการขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้าไป

2.2.4.5 ภาคกำเนิดความถี่

ตัวส่งสัญญาณ DTMF ใน MT8870 มีความสามารถในการกำเนิดสัญญาณโทรศัพท์ได้ทั้งหมด 16 สัญญาณ ตั้งแต่ 0 – 9 และ A – D เป็นไปตามรหัสเลขฐาน 16 มีสัญญาณรบกวนต่ำ และ มีความถูกต้องสูง สัญญาณ ความถี่ทั้งหมดสร้างจากสัญญาณความถี่ที่ได้รับจากคริสตอลภายนอกซึ่งมีความถี่เท่ากับ 3.579545 MHz สัญญาณซายส์แต่ละค่าเกิดจากการสังเคราะห์แบบดิจิตอลโดยใช้ความถี่ของแถวและหลักที่ได้ทำการโปรแกรมไว้ แล้วใช้สวิทช์คาปาซิเตอร์ดิจิตอลทูอนาล็อกคอนเวอร์เตอร์ ความถี่ของแถวและหลักจะถูกผสมกัน แล้วจะถูกกรองเพื่อที่จะให้สัญญาณที่ได้มีความถูกต้องมากขึ้น สำหรับค่าของความถี่ของแต่ละ แถว และ หลัก ก็เหมือนกับตอนถอดสัญญาณ DTMF สามารถดูได้จากตารางเดียวกัน

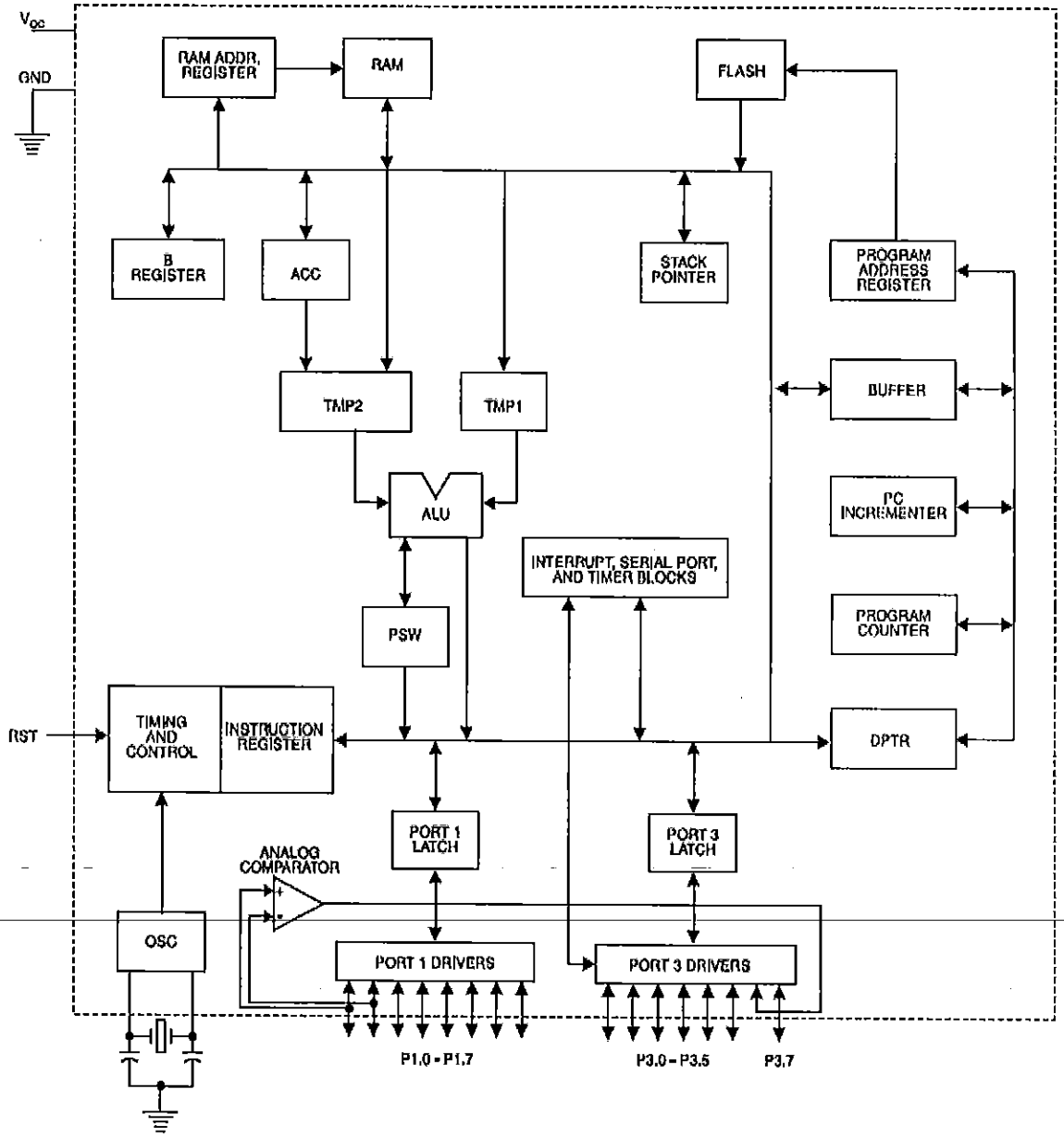
2.3 ไมโครคอนโทรเลอร์ AT89C2051

2.3.1 คุณสมบัติของไอซีไมโครคอนโทรลเลอร์ AT89C2051

1. แหล่งจ่ายไฟใช้ได้ตั้งแต่ 2.7 โวลท์ ถึง 6โวลท์
2. มีหน่วยความจำโปรแกรมชนิด Flash Memory ขนาด 2KBytes
3. มีหน่วยความจำแบบแรม 8 Bit ขนาด 128 Byte
4. ทำงานที่ความเร็วสัญญาณนาฬิกาได้สูงสุดถึง 24 MHz
5. มีอินพุตเอาต์พุตพอร์ต* ขนาด 15 บิต
6. พอร์ตสามารถ SINK * กระแสได้ 20 mA
7. มีสัญญาณการ อินเตอร์รัพท์* ได้ 6 แหล่ง
8. มีพอร์ตสื่อสารแบบอนุกรม 1 ช่อง (UART)
9. มีวงจรตั้งเวลาและวงจรนับขนาด 16 บิตจำนวน 2 ช่อง
10. สามารถโปรแกรมข้อมูล เพื่อป้องกันการอ่านเขียนหรือคัดลอกโปรแกรมได้ 2 ระดับ
11. มีวงจรเปรียบเทียบสัญญาณอนาล็อก (Analog Comparator Input)1 ช่อง
12. มีระบบประหยัดพลังงาน (Low Power Idle And Power Down Mode)

2.3.2 การทำงานของไมโครคอนโทรลเลอร์ตระกูลMCS 51 เบอร์ AT89C2051

AT89C2051 เป็นไมโครคอนโทรลเลอร์ ขนาด 8 บิต ชนิด CMOS มีหน่วยความจำชนิดแฟลชขนาด 2 กิโลไบท์ สามารถลบได้ด้วยสัญญาณไฟฟ้า และ ใส่ข้อมูลลงไปใหม่ ชุดคำสั่งและขาสัญญาณเข้ากันได้กับไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ซึ่งเป็นไมโครคอนโทรลเลอร์มาตรฐานในอุตสาหกรรม จุดเด่นของ AT89C2051 คือ มีหน่วยความจำชนิดแฟลชขนาด 2 กิโลไบท์ หน่วยความจำข้อมูล 128 ไบท์ อินพุท เอาท์พุทรวม 15 ขา มีตัวตั้งเวลาและตัวนับขนาด 16 บิต สองตัว อินเตอร์รัพท์ 5 ตำแหน่ง มีวงจรรับข้อมูลอนุกรมชนิดฟลูดูเพล็กและมีวงจรกำเนิดสัญญาณนาฬิกา และ ใช้พลังงานต่ำสามารถดูได้จากรูปที่ 2.12 ด้านล่าง



รูปที่ 2.12 แสดงบล็อกไดอะแกรมของ AT89C2051

2.3.2.1 การทำงานของแต่ละขา

Vcc เป็นขาไฟบวก

GND เป็นขากราวด์

Port 1 เป็น port แบบ 2 ทิศทาง เป็นได้ทั้งอินพุทและเอาท์พุท มีขนาด 8 บิต สามารถต่อกับไอซี TTL ได้โดยตรง และ ทำให้เป็นไฮอิมพีแดนได้ โดยการเขียนลอจิก 1 ไปที่ port 1 นอกจากนี้แล้วยังกำหนดให้เป็นแบบมัลติเพล็กระหว่าง data และ low-order address ได้ด้วย ซึ่งจะใช้ในการทำ memory map กับข้อมูลหรือโปรแกรมภายนอก ได้มีการพูลอัพไว้ภายในแล้ว

Port 3 เป็น port แบบ 2 ทิศทางเป็นได้ทั้งอินพุทและเอาท์พุท มีขนาด 8 บิต มีการพูลอัพไว้ภายในแล้ว สามารถต่อกับไอซี TTL ได้โดยตรง หน้าที่พิเศษอื่น ๆ ของ port 3

P3.0 เป็นขารับข้อแบบ serial

P3.1 เป็นขาส่งข้อมูลแบบ serial

P3.2 เป็นขาที่ใช้กระตุ่นอินเตอร์ 0 จากสัญญาณภายนอก

P3.3 เป็นขาที่ใช้กระตุ่นอินเตอร์ 1 จากสัญญาณภายนอก

P3.4 เป็นขานับสัญญาณของ timer 0

P3.5 เป็นขานับสัญญาณของ timer

RST เป็นขารีเซตอินพุท ลอจิก 1 จะทำการรีเซตการทำงานภายในตัว microcontroller ให้เริ่มทำงานใหม่ ลอจิก 0 จะทำงานปรกติ

XTAL1 เป็นขาอินพุทของ oscillator

XTAL2 เป็นขาเอาท์พุทของ oscillator

2.3.2.2 การจักสรรหน่วยความจำข้อมูล

หน่วยความจำข้อมูลแบ่งออกเป็นสองพื้นที่ คือ พื้นที่หน่วยความจำข้อมูลบนซิพ 256 ไบท์ จากตำแหน่ง 00H ถึง FFH และหน่วยความจำข้อมูลภายนอก จากตำแหน่ง 0000H ถึง FFFFH หรือ มีขนาด 64 กิโลไบท์ การอ่านหน่วยความจำข้อมูลภายนอก CPU จะกำเนิดสัญญาณ RD ส่วนการเขียนข้อมูล CPU กำเนิดสัญญาณ WR พื้นที่ของหน่วยความจำบนซิพ จะพิเศษกว่าหน่วยความจำภายนอก กล่าวคือ ตำแหน่ง 00H ถึง 1FH จะเป็นพื้นที่ของรีจีสเตอร์ใช้งานทั่วไป R0 – R7 จำนวน 4 ชุด ตำแหน่ง 20H – 2FH เป็นพื้นที่ที่สามารถกระทำได้ในระดับบิตโดยสามารถระบุข้อมูลตำแหน่งเป็นบิตได้ มีค่าตำแหน่งบิตเป็น 00H-7FH เช่นที่ตำแหน่ง 20H ภายในมีตำแหน่งบิตเป็น 00H-07H ส่วนพื้นที่ตำแหน่ง 30H-7FH เป็นพื้นที่ใช้งานทั่วไป ที่เหลือ 128 ส่วนบนจะเป็นที่อยู่ของรีจีเตอร์ต่าง

2.3.2.3 รีจีสเตอร์ใน AT89C2051

AT89C2051 มีโปรแกรมเค้าเตอร์ PC ที่ใช้พื้นที่แยกจากหน่วยความจำข้อมูล โปรแกรมมิ่งรีจีสเตอร์ใช้เป็นตัวเก็บ ข้อมูลก่อน และ หลัง การประมวลผลที่หน่วยประมวล ALU สถานะทางคณิตศาสตร์และตรรกศาสตร์จะเก็บไว้ในรีจีสเตอร์ PSW คำสั่งที่ใช้ในการเขียนโปรแกรมก็จะประกอบไปด้วยรีจีสเตอร์พื้นที่หน่วยความจำส่วนนี้อยู่ที่ 80H – FFH และเรียกว่ารีจีสเตอร์หน้าที่พิเศษ ดูได้จากรูปที่ 2.13

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

รูปที่ 2.13 แสดงรีจีสเตอร์ใน AT89S8252

ACC

เป็นแอคคิวมูลเลเตอร์รีสจีสเตอร์ขนาด 8 บิตใช้พักข้อมูลก่อนและหลังการประมวลผลทางคณิตศาสตร์แนะตรรกศาสตร์ในโปรแกรมเราสามารถเขียนสั้น ๆ เป็น A ตำแหน่งของแอคคิวมูเลเตอร์อยู่ที่แอดเดรส 0E0H สามารถเข้าถึงข้อมูลเป็นบิตได้ ตำแหน่งบิตคือ E0H – E7H

PC

เป็นรีจีสเตอร์ขนาด 16 บิต ใช้เป็นตัวชี้ตำแหน่งของคำสั่งจะมีค่าเป็น 0000H หลังการรีเซท

B

เป็นรีจีสเตอร์ขนาด 8 บิตใช้ในคำสั่งคูณและหารตำแหน่งของรีจีสเตอร์อยู่ที่แอดเดรส 0F0H สามารถเข้าถึงข้อมูลเป็นบิตได้ ตำแหน่งบิตคือ 0F0H – 0F7H

PSW

เป็นรีจีสเตอร์ขนาด 8 บิต ใช้เก็บสถานะภายหลังการกระทำคำสั่งทางคณิตศาสตร์ และตรรก- ศาสตร์สถานะที่เกิดขึ้นจะเก็บเป็นข้อมูล 1 บิต หรือใช้เรียกเป็นแฟลกบอกสถานะ เช่น CY แฟลกตัวทด นอกจากนี้บางบิตยังใช้ทำหน้าที่อื่น ๆ เช่น RS0 และ RS1 ที่ใช้ในการเลือกกลุ่มของส่วน F0 เป็นที่สามารถใช้งานได้ทั่วไป R0-R7 รูปที่ 2.19 แสดงภาพ และ ตำแหน่งของ PSW

SUBF,SCON

SUBF เป็นรีจีสเตอร์ขนาด 8 บิตใช้พักข้อมูลเพื่อการรับและส่งข้อมูลแบบอนุกรมข้อมูลจะส่ง ออกทางขา TXD และรับทางขา RXD ส่วน SCON เป็นรีจีสเตอร์กำหนดโหมดและควบคุมการทำงานของการรับข้อมูลแบบอนุกรม

SP

เป็นรีจีสเตอร์ขนาด 8 บิตใช้บอกตำแหน่งของหน่วยความจำที่เป็นพื้นที่สแตก พื้นที่สแตกมีการใช้งานแบบชนิดเข้าที่หลังออกก่อนเปรียบได้กับการว่างจานกระเบื้องซ้อนกันเวลาเราหยิบใช้เราจะหยิบจานที่ว่างอยู่บนสุดก่อน คำศัพท์ทางเทคนิคเรียกว่า FILO ย่อมาจาก First In Last Out การทำงานของโปรแกรมที่มีการเรียกโปรแกรมย่อยด้วยคำสั่ง call นั้นตำแหน่งแอดเดรสถัดจากคำสั่ง call จะได้รับการเก็บลงในหน่วยความจำข้อมูลบนชิปที่ตำแหน่งระบุใน SP รีจีสเตอร์ เมื่อโปรแกรมย่อยทำงานเสร็จคำสั่ง ret จะคืนค่าตำแหน่งถัดจากคำสั่ง call ให้กับโปรแกรมเค้าเตอร์ ปรกติเมื่อรีเซต CPU จะทำให้ SP ชี้อยู่ที่ตำแหน่ง 07H เมื่อมีการใช้ตำแหน่งเริ่มต้นของพื้นที่สแตกจะเป็น 08H กล่าวคือสแตกจะเพิ่มตัวเองขึ้นหนึ่งก่อนใส่ข้อมูลหนึ่งไบท์ แล้วดึงข้อมูลกับขึ้นมาก่อนจึงจะลดค่าตัวเองลงหนึ่ง นอกจากการใช้สแตกรีจีสเตอร์ในการเรียกโปรแกรมย่อยแล้ว SP ยังใช้คำสั่ง PUSH และ POP ในการใส่ข้อมูลและดึงข้อมูลตามลำดับ

DPTR

เป็นรีจีสเตอร์ขนาด 16 บิตประกอบด้วย DPH และ DPL มีตำแหน่งอยู่ที่ 83H และ 82H ตามลำดับ ใช้สำหรับการอ้างหน่วยความจำในการอ่านและเขียนข้อมูล

Port P1, P3

เป็นรีจีสเตอร์พิเศษต่างมาจากรีจีสเตอร์ที่กล่าวมาข้างต้นตรงที่ข้อมูลที่อ่านและเขียนที่รีจีสเตอร์เหล่านี้สามารถวัดได้ด้วยลอจิกโพรบที่ขาของชิปโดยตรง กล่าวอีกนัยหนึ่งก็คือรีจีสเตอร์เหล่านี้เป็นเส้นทางติดต่อระหว่าง CPU กับ โลกภายนอกนั้นเองเราเรียกรีจีสเตอร์เหล่านี้เป็นอินพุทเอาท์พุทพอร์ทก็ได้ พอร์ททั้งสองเป็นชนิดสองทิศแต่ละพอร์ทประกอบด้วย แลตช์ วงจรขับภาค

เอาท์พุท และวงจรอินพุทบัพเฟอร์ เอาท์พุทแต่ละบิตทำงานด้วย D Flip-Flop สามารถคงค่าที่เขียนมายังพอร์ท ส่วนการใช้เป็นอินพุทพอร์ท คำสั่งบางคำสั่งจะอ่านค่าจากเอาท์พุท Q ของ ฟลิบฟล๊อบ บางคำจะอ่านจากขาอินพุทโดยตรง พอร์ทสามารถเข้าถึงข้อมูลได้เป็นบิตได้

TH0

Timer 0 เป็นรีจีสเตอร์ขนาด 16 บิตประกอบด้วย TH0 และ TL0 รับสัญญาณนาฬิกาภายนอกที่ขา T0 หรือ รับสัญญาณนาฬิกาจากออสซิเลเตอร์ภายใน การนับเป็นการนับแบบเพิ่มครั้งละหนึ่งเมื่อค่าที่นับเกินขนาดของรีจีสเตอร์ คือ จาก FFFFH ไป 0 บิต TF0 จะเซทเป็น ลอจิก 1

TH1

Timer 1 เป็นรีจีสเตอร์ขนาด 16 บิตประกอบด้วย TH1 และ TL1 มีการทำงานเช่นเดียวกับไทมเมอร์ 0 TMOD,TCON เป็นรีจีสเตอร์ใช้เลือกโหมดการทำงานของไทมเมอร์ 0 และ 1 ส่วน TCON เป็นรีจีสเตอร์ควบคุมการทำงานของรีจีสเตอร์ทั้งสอง

IP,IE,PCON

IP เป็นรีจีสเตอร์ที่ควบคุมลำดับความสำคัญของการอินเทอร์รัพท์

IE เป็นรีจีสเตอร์ที่ควบคุมการยอมให้เกิดการอินเทอร์รัพท์ได้หรือไม่

PCON เป็นรีจีสเตอร์ควบคุมโหมดประหยัดพลังงาน

บทที่ 3
การเขียนโปรแกรมภาษาแอสเซมบลี

การที่จะให้ MCS-51 ทำงานได้จะต้องป้อนคำสั่งให้กับ MCS-51 การนำรหัสคำสั่งหลายๆ รหัสคำสั่งมาเรียงกันให้ MCS-51ทำงานเรียกว่าโปรแกรมภาษาแอสเซมบลี ซึ่งภาษานี้เป็นภาษาที่อยู่ระหว่างภาษาเครื่องกับภาษาระดับสูง โดยภาษาระดับสูงเช่น Pascal C หรือ C++ จะใช้ชุดคำสั่งที่ใกล้เคียงกับภาษาที่มนุษย์ใช้ ส่วนภาษาเครื่องจะเป็นภาษาที่ใช้ในรูปของเลขฐานสองซึ่งเป็นภาษาที่คอมพิวเตอร์จะเข้าใจ

โปรแกรมแอสเซมบลียังไม่สามารถนำมาใช้งานกับไมโครคอนโทรลเลอร์ได้ทันที แต่ต้องนำมาทำการแปลงจากซอร์สโค้ดไฟล์เป็นภาษาเครื่อง (machine code) ทั้งคำสั่งต่างๆ ข้อมูลและการอ้างแอดเดรสทั้งหมดถูกแปลงไปเป็นโปรแกรมภาษาเครื่องที่ไมโครคอนโทรลเลอร์สามารถเข้าใจได้ หรือเรียกว่าออปเจ็กต์โค้ด (object code) โดยใช้โปรแกรมแอสเซมเบลอร์ เอาต์พุตไฟล์ที่ได้จากการแอสเซมเบลอร์จะมีนามสกุลเป็น .HEX ไฟล์ที่ถูกแปลงเป็นภาษาเครื่องแล้วจะอยู่ในรูปของเลขฐานสิบหกจำนวน 2 หลัก เรียงตามลำดับคำสั่งต่าง ๆ ที่เขียนขึ้น ไฟล์ที่ได้สามารถนำไปประมวลผลได้ทันที เมื่อมันถูกส่งผ่านไปเก็บไว้ที่หน่วยความจำบน MCS-51 บอร์ด นอกจากนั้นยังจะทำการสร้างไฟล์เอกสาร (list file) ขึ้นมามีนามสกุล .LST เพื่อรวบรวมและแสดงออปเจ็กต์โค้ดที่สร้างโดยโปรแกรมแอสเซมเบอร์จากซอร์สโค้ดโปรแกรมและข้อมูลอื่น ๆ ที่สำคัญ เช่น

คอลัมน์ LINE แสดงตำแหน่งบรรทัดของซอร์สโปรแกรม

คอลัมน์ LOC แสดงตำแหน่งของออปเจ็กต์โค้ด (object code) ที่เก็บในหน่วยความจำโปรแกรม

คอลัมน์ OBJ แสดงข้อมูลออปเจ็กต์โค้ด

คอลัมน์ T แสดงเวลาที่ใช้ไปในการเอ็กซิคิวต์ในแต่ละคำสั่งนั้น ๆ โดยมีหน่วยเป็นไมโครวินาที

คอลัมน์ SOURCE แสดงข้อมูลเหมือนกับไฟล์ต้นฉบับที่เขียนโปรแกรมขึ้น จะประกอบด้วยส่วน
นีโมนิก, ชื่อหรือลาเบล, แอดเดรส

3.1 รูปแบบของภาษาแอสเซมบลี

โปรแกรมภาษาแอสเซมบลีประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

1. Machine Instructions คือชุดคำสั่งภาษาเครื่องโดยแทนด้วยรหัสนีโมนิก เช่น MOV, ANL

2.Assembler Directives หรือคำสั่งเทียม เป็นคำสั่งที่กำหนดขึ้นในภาษาแอสเซมบลีโดยจะไม่ถูกเปลี่ยนไปเป็นรหัสภาษาเครื่อง มีไว้สำหรับกำหนดโครงสร้างโปรแกรม, สัญญาลักษณ์, ข้อมูลต่าง ๆ เช่น ORG, EQU เป็นต้น

3. Assembler Controls เป็นชุดคำสั่งในการควบคุมของตัวแอสเซมเบอร์

4. Comment เป็นการเขียนคำอธิบายเข้าไปในโปรแกรม

ในการเขียนโปรแกรมในภาษาแอสเซมบลีในแต่ละบรรทัดมีส่วนประกอบที่สำคัญอยู่ 4 ส่วนคือ Label, Mnemonics, Operand, Comment โดยรูปทั่วไปจะเป็นดังนี้

⟶ [Label:] Mnemonics [Operand] [;Comment]

Label Field

ในภาษาแอสเซมบลี Label จะเป็นตัวแทนของตำแหน่งของชุดคำสั่ง หรือข้อมูลที่ตามหลัง Label อยู่ในคำสั่งกระโดดหรือคำสั่งต่าง ๆ ที่ต้องการอ้างตำแหน่ง สามารถแทนค่าตำแหน่งด้วย Label ได้เลย เช่น SJMP เป็นการกระโดดไปยังตำแหน่งที่แทนด้วย Label SKIP

โดยทั่วไปแล้ว Label Field แทนได้ สองความหมายคือ เป็น "Label" แทนค่าตำแหน่ง และเป็น "Symbol" ซึ่งจะแทนค่าสัญญาลักษณ์ ถ้าหากจะใช้เป็น Label จะตามด้วยเครื่องหมาย " : " (Colon) แต่ถ้าจะเป็น Symbol จะตามด้วยคำสั่งเทียม เช่น EQU, SEGMENT, BIT, DATA เป็นต้น

Mnemonic Field

ส่วน Mnemonic Field จะเป็นชุดคำสั่งหรือคำสั่งเทียมก็ได้ ซึ่งจะเขียนตามหลัง Label ตัวอย่างของชุดคำสั่งได้แก่ ADD, MOV, DIV หรือ INC ตัวอย่างของคำสั่งเทียมได้แก่ ORG, EQU หรือ DB

Operand Field

ในภาษาแอสเซมบลี Operand เป็นตัวบอกว่าจะให้ Mnemonic ทำอะไร เช่น MOV จะเป็นตัวบอกว่าให้ย้ายข้อมูล แต่ยังไม่มีตำแหน่งที่จะย้ายว่าจะย้ายจากไหนไปไหน Operand จะเขียนตามหลังรหัส Mnemonic ใน Operand Field นี้จะประกอบด้วย Address หรือ Data ที่จะใช้กับชุดคำสั่ง ซึ่งอาจจะเป็น Label ที่แทนตำแหน่งของข้อมูลหรือ Symbol ที่แทนค่าข้อมูลด้วยก็ได้คำสั่งส่วนใหญ่จะมี Operand Field ตามหลัง แต่บางคำสั่งจะไม่มี เช่น คำสั่ง RET, NOP เป็นต้น

Comment Field

ส่วนนี้จะเป็นส่วนที่เขียนขึ้นในภาษาแอสเซมบลี เพื่อเป็นคำอธิบายส่วนต่าง ๆ ในโปรแกรมโดยการเขียนจะใช้เครื่องหมาย " ; " (Semicolon) นำหน้าส่วน Comment นี้ ตัวแอสเซมเบอร์จะไม่แปลงเป็นภาษาเครื่อง

โหมดการอ้างแอดเดรส (Addressing Modes) สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะมีโหมดการอ้างแอดเดรสอยู่ 8 โหมด ได้แก่ Register addressing modes, Direct addressing modes, Immediate addressing modes, Indirect addressing modes, Based-indexed indirect addressing modes, Relative addressing modes, Extended addressing modes และ Implied addressing modes

1. Register Addressing Modes (การอ้างแอดเดรสด้วยรีจีสเตอร์)

เป็นการอ้างแอดเดรสโดยใช้รีจีสเตอร์ ข้อมูลที่จะนำมาประมวลผลจะเป็นข้อมูลที่เก็บอยู่ในรีจีสเตอร์ โดยตรง ในภาษาแอสเซมบลีของ MCS-51 การอ้างตำแหน่งของรีจีสเตอร์ต่าง ๆ สาสารถเขียนเป็นตัวอักษรได้โดยตรง เช่น A, R0 ถึง R7, DPTR ส่วนสัญญาลักษณ์ $ (Dollar sign) จะแทนตำแหน่งที่รีจีสเตอร์ PC ชี้อยู่โดยมากมักจะใช้กับคำสั่งกระโดด

2. Direct Addressing Modes (การอ้างแอดเดรสโดยตรง)

เป็นโหมดการอ้างแอดเดรสที่ติดต่อกับข้อมูลที่อยู่ในหน่วยความจำโดยตรง ซึ่งจะรวมถึงรีจีสเตอร์ใช้ งานพิเศษ (SFR) ด้วย ในส่วนนี้มีหน่วยความจำขนาด 128 ไบต์ ดังนั้นจึงอ้างได้ 128 ตำแหน่ง (00H-7FH) เช่น MOV 10H,20H จะเป็นการอ่านไบต์ข้อมูลที่อยู่ในตำแหน่งแอดเดรส 20H มาเก็บในตำแหน่งแอดเดรส 10H

3. Immediate Addressing Modes (การอ้างแอดเดรสด้วยค่าตำแหน่งแบบทันที)

การอ้างแอดเดรสแบบนี้ข้อมูลที่จะนำมาประมวลผลจะอยู่ในรหัสคำสั่งโดยตรง การเขียนรหัสนีโมนิกจะใช้เครื่องหมาย # (pound sign) นำหน้าค่าของข้อมูล เช่น MOV 10H,#22H จะเป็นการนำค่าข้อมูล 22H มาเก็บในแอดเดรส 10H

4. Indirect Addressing Modes (การอ้างแอดเดรสโดยอ้อม)

จะเป็นการอ้างแอดเดรสโดยอ้อมโดยกำหนดให้ใช้รีจีสเตอร์ R0, R1 หรือ DPTR เก็บค่าตำแหน่งของหน่วยความจำที่ต้องการดำเนินการ โดยถ้าหากต้องการติดต่อกับหน่วยความจำข้อมูลภายในใช้รีจีสเตอร์ R0 หรือ R1 เป็นตัวชี้ตำแหน่งของหน่วยความจำ ถ้าหากต้องการติดต่อกับหน่วยความจำข้อมูลภายนอกใช้รีจีสเตอร์ DPTR เป็นตัวชี้ ในการเขียนรหัสนีโมนิกจะใช้สัญญาลักษณ์ @ เป็นตัวบอกว่ารีจีสเตอร์นี้เก็บค่าตำแหน่งหน่วยความจำ เช่น

MOV A , @R1 นำข้อมูลจากแอดเดรสที่ชี้โดย R1 มาเก็บในรีจีสเตอร์ A

MOVX @R0,A นำข้อมูลจากรีจีสเตอร์ A ไปเก็บยังหน่วยความจำภายนอกที่ชี้โดย R0

MOVX @DPTR,A นำข้อมูลจากรีจีสเตอร์ A ไปเก็บยังหน่วยความจำภายนอกที่ชี้โดย DPTR

5. Based-Indexed Indirect Addressing Modes

การอ้างแอดเดรสในโหมดนี้ จะใช้รีจีสเตอร์สองตัวในการอ้างแอดเดรส โดยจะอ่านข้อมูลออกมาในลักษณะของการเปิดตาราง ซึ่งจะมีคำสั่งอยู่ 2 รูปแบบคือ

MOVC A,@A+DPTR

MOVC A,@A+PC

6. Relative Addressing Modes

การอ้างแอดเดรสประเภทนี้จะใช้กับคำสั่งควบคุมการทำงานของโปรแกรม เช่น คำสั่งกระโดด คำสั่งเรียกโปรแกรมย่อย โดยจะเป็นการระบุตำแหน่งของหน่วยความจำที่อยู่ด้านหน้าหรือด้านหลังของตำแหน่งที่รีจีสเตอร์ PC ชี้อยู่ โดยนำค่าในรีจีสเตอร์ PC บวกกับค่าตัวเลขแบบ 2's complement

7. Extended Addressing Modes

จะเป็นการเข้าถึงข้อมูลที่อยู่ในแอดเดรสในหน่วยความจำทั้งหมดที่ MCS-51 สามารถมีได้ ตามที่ทราบมาแล้วว่า MCS-51สามารถมีหน่วยความจำข้อมูลได้ 64 กิโลไบต์ และมีหน่วยความจำโปรแกรมได้ 64 กิโลไบต์

8. Implied Addressing Modes

ในโหมดนี้จะไม่มีการอ้างตำแหน่งของข้อมูล หมายถึงคำสั่งที่ไม่มีโอเปอร์แรนด์ (operands) เช่นคำสั่ง NOP, RET เป็นต้น

คำสั่งทำซ้ำและคำสั่งกระโดด

การให้ MCS-51 ทำงานใด ๆ ซ้ำ ๆ สามารถทำได้หลายวิธี แต่จะเรียกรวมกันว่าการทำลูป (loop) ในการทำซ้ำแต่ละครั้งจะมีการตรวจสอบเงื่อนไขด้วยว่าจะต้องทำงานนั้นซ้ำหรือไม่ สำหรับการทำคำสั่งกระโดดจะมีสองประเภทใหญ่ ๆ คือ การกระโดดแน่ ๆ ถ้าโปรแกรมทำงานมาถึงคำสั่งนั้น และจะกระโดดถ้าเงื่อนไขเป็นไปตามที่กำหนด

การเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์ตระกูล MCS-51 จะใช้คำสั่ง DJNZ reg, label เมื่อ MCS-51 ทำคำสั่งนี้แต่ละครั้งจะลดค่าในรีจีสเตอร์ reg ลงค่าหนึ่ง และคำสั่งนี้จะทำให้โปรแกรมกระโดดไปยังแอดเดรส label ถ้าค่าในรีจีสเตอร์ reg ยังไม่เป็นศูนย์ แต่ถ้าค่าในรีจีสเตอร์ reg เป็นศูนย์ โปรแกรมจะทำคำสั่งต่อจากคำสั่ง DJNZ นี้

การกระโดดแบบมีเงื่อนไข

คำสั่งกระโดดแบบมีเงื่อนไขของ MCS-51 สรุปได้ดังตารางที่ 3.1

ตารางที่3.1 คำสั่งกระโดดแบบมีเงื่อนไข

คำสั่ง	การทำงาน
JZ	การกระโดด A เท่ากับ 0
JNZ	การกระโดด A ไม่เท่ากับ 0
DJNZ	ลดค่าทีละหนึ่งและกระโดดถ้า A ไม่เท่ากับ 0
CJNE A,byte	กระโดดถ้า A ไม่เท่ากับ byte
CJNZ reg,#data	กระโดดถ้า reg ไม่เท่ากับ #data
JC	กระโดดถ้า (ตัวทด) CY เท่ากับ 1
JNC	กระโดดถ้า (ตัวทด) CY เท่ากับ 0
JB	กระโดดถ้าบิตที่กำหนดเท่ากับ 1
JNB	กระโดดถ้าบิตที่กำหนดเท่ากับ 0
JBC	กระโดดถ้าบิตที่กำหนดเท่ากับ 1และเคลียร์บิต

คำสั่งกระโดดแบบไม่มีเงื่อนไข

การกระโดดแบบไม่มีเงื่อนไข (unconditional jump) จะเป็นคำสั่งที่ทำให้โปรแกรมกระโดดไปยังตำแหน่งที่กำหนดแน่ ๆ ใน MCS-51 มีคำสั่งกระโดดแบบไม่มีเงื่อนไขสองคำสั่งคือ คำสั่ง LJMP (long jump) และ SJMP (short jump) โดยทั้งสองคำสั่งต่างกันที่ระยะทางการกระโดดของโปรแกรม

LJMP (long jump)

คำสั่งนี้จะมีขนาด 3 ไบต์ โดยไบต์แรกจะเป็นออปโค้ด อีกสองไบต์ต่อมาจะแทนตำแหน่งแอดเดรสขนาด 16 บิต ซึ่งสามารถกระโดดไปยังหน่วยความจำใด ๆ ได้ในช่วง 0000 ถึง FFFFH

SJMP (short jump)

คำสั่งนี้จะมีขนาด 2 ไบต์ โดยไบต์แรกจะเป็นออปโค้ด ไบต์ต่อมาจะเป็นแอดเดรสที่จะกระโดดไปโดยอ้างแอดเดรสแบบสัมพัทธ์ ซึ่งจะอ้างตำแหน่งได้ในช่วง 00 ถึง FFH หรือกระโดดไปข้างหลังได้ -128 ไบต์ และกระโดดไปข้างหน้าได้ +127 ไบต์ ข้อดีของคำสั่งนี้คือประหยัดออปเจ็กต์โค้ดที่ใช้งานลงอย่างมากเมื่อต้องมีการกระโดดระยะสั้น ๆ บ่อยครั้ง

การเรียกโปรแกรมย่อย

การเขียนโปรแกรมประเภทโปรแกรมย่อย Subroutine สามารถถูกเรียกขึ้นมาใช้งานได้ด้วยคำสั่ง CALL จะมีอยู่สองคำสั่งคือ LCALL (long call) และ ACALL (absolute call) โดยทั้งสองคำสั่งต่างกันตรงขนาดของคำสั่ง และระยะห่างของโปรแกรมย่อยที่จะเรียกใช้งาน

LCALL (long call)

คำสั่งนี้จะมีขนาด 3 ไบต์ โดยไบต์แรกจะเป็นออปโค้ด อีกสองไบต์ต่อมาจะเป็นแอดเดรสที่จะกระโดดไปทำงานโปรแกรมย่อย ซึ่งสามารถเรียกโปรแกรมย่อยได้ในระยะ 64 กิโลไบต์ เมื่อโปรแกรมย่อยถูกเรียกใช้ MCS-51 จะนำค่าแอดเดรสของคำสั่งถัดไปหรือค่า PC เก็บลงหน่วยความจำสแตกโดยอัตโนมัติ แล้วโหลดค่าแอดเดรสของโปรแกรมย่อยให้กับรีจีสเตอร์ PC และเมื่อทำโปรแกรมย่อยจนพบคำสั่ง RET (return) MCS-51 จะกลับไปทำคำสั่งของโปรแกรมที่อยู่ต่อจากคำสั่งเรียกโปรแกรมย่อย โดยคืนค่าที่เก็บอยู่ในสแตกให้กับรีจีสเตอร์ PC ตามเดิม

ACALL (absolute call)

เป็นคำสั่งที่มีขนาด 2 ไบต์ โดยใช้ข้อมูล 11 บิตในคำสั่งเป็นค่าแอดเดรสทำให้สามารถเรียกโปรแกรมย่อยได้ในช่วง 2 กิโลไบต์ การทำงานกับสแตกของ ACALL และ LCALL นี้จะไม่แตกต่างกัน แต่การใช้งานจะต่างกันตรงที่คำสั่ง LCALL เรียกโปรแกรมย่อยได้ในช่วง 64 กิโลไบต์ ส่วนคำสั่ง ACALL เรียกโปรแกรมย่อยได้ในช่วง 2 กิโลไบต์

3.2 Assemble-time Expression Evalution

ในที่นี้จะกล่าวการกำหนดค่าคงที่และตัวแปรต่าง ๆ

Number Bases

การกำหนดค่าตัวเลขหรือค่าคงที่ต่าง ๆ ให้กับตัวแปรในการเขียนภาษาแอสเซมบลีนั้น ถ้าเป็นเลขไบนารีจะลงท้ายด้วยตัว "B" ถ้าเป็นเลขฐานสิบหกจะใส่ตัวเลข 0 นำหน้าและลงท้ายด้วย "H" ถ้าเป็นเลขฐานแปดจะลงท้ายด้วย "O" ถ้าเป็นเลขฐานสิบจะลงท้ายด้วย "D" หรือไม่ต้องใส่อะไรก็ได้ ตัวอย่างเช่น

```
MOV   A,#1111B
MOV   A,#0FFH
MOV   A,#10D หรือ  MOV    A,#10
```

Character Strings

การกำหนดตัวอักษรจะใช้ 1 หรือ 2 ตัว โดยการกำหนดจะใช้เครื่องหมาย (') ปิดตัวอักษร ตัวแปรภาษาแอสแซมเบอร์จะแปลงตัวอักษรเป็นรหัส ASCII ดังตัวอย่างต่อไปนี้

```
CJNE   A, # 'Q' , AGAIN
SUBB   A, # 'Q'
MOV    DPTR,# 'AB'
```

ห้องสมุดคณะวิศวกรรมศาสตร์

i 5081142

5000112

ฝร.
9216
2649
c.2

Arithmetic Operators

ตัวดำเนินการทางคณิตศาสตร์ ที่ใช้ในภาษาแอสแซมบลีมีดังนี้

ตารางที่ 3.2 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	ความหมาย
+	บวก
-	ลบ
*	คูณ
/	หาร
MOD	หารแบบ module (เอาเศษที่ได้จากการหาร)

การเขียนโปรแกรมทางคณิตศาสตร์ของสองบรรทัดต่อไปนี้ให้ผลเหมือนกัน โดยบรรทัดแรกจะบวกค่า10 ในเลขฐานสิบ กับ 10 ในเลขฐานสิบหกแล้วนำค่าที่ได้มาเก็บไว้ในรีจีสเตอร์ A ซึ่งมีค่าเท่ากับการเก็บค่า 1AH ลงในรีจีสเตอร์ A โดยตรง

```
MOV   A, #10 + 10H
MOV   A, #1AH
```

การเขียนโปรแกรมทางคณิตศาสตร์ของสองบรรทัดต่อไปนี้ให้ผลเหมือนกัน โดยบรรทัดแรกจะเอาเศษที่ได้จากการหารคือ 5 ใส่ในรีจีสเตอร์ A

```
MOV   A, #37 MOD 8
MOV   A, #5
```

Logical Operators

ตัวดำเนินการทางลอจิกมีดังนี้

ตารางที่ 3.3 ตัวดำเนินการทางลอจิก

ตัวดำเนินการ	ความหมาย
OR	การกระทำ OR
AND	การกระทำ AND
XOR	การกระทำ Exclusive
NOT	การกระทำ Complement

การเขียนโปรแกรมสามารถใส่ตัวดำเนินการเข้าไปในคำสั่งได้เลย ดังตัวอย่างต่อไปนี้

```
MOV    A, '#9' AND 0FH
```

ตัวอย่างต่อไปนี้จะแสดงให้เห็นว่าตัวดำเนินการ NOT สามารถใส่เข้าไปในตัว Operands ได้ทันที ซึ่งคำสั่ง MOV ทั้งสามคำสั่งผลที่ได้จะเหมือนกัน

```
THREE          EUQ   3
MINUS_THREE    EQU   -3
               MOV   A, # (NOT THREE) + 1
               MOV   A, # MINUS_THREE
               MOV   A, # 11111101B
```

Special Operators

ตัวดำเนินการพิเศษมีดังนี้ตารางที่

3.4 ตัวดำเนินการพิเศษ

ตัวดำเนินการ	ความหมาย
SHR	เลื่อนไปทางขวา
SHL	เลื่อนไปทางซ้าย
HIGH	ค่าไบต์สูง
LOW	ค่าไบต์ต่ำ
( )	ให้ทำค่านี้ก่อน

ตัวอย่างเช่น

```
MOV    A, #8 SHL 1
MOV    A, #10H
```

ตัวอย่างจะเป็นคำสั่งเลื่อนค่า 8 บิต ไปหนึ่งครั้งแล้วเก็บไว้ในรีจีสเตอร์ A ค่า 8 มีเลขไบนารีเป็น 00001000 เมื่อถูกเลื่อนไปทางซ้ายหนึ่งครั้งจะมีค่าเป็น 00010000 ซึ่งผลลัพท์จะเท่ากับ 10H ซึ่งจะทำให้คำสั่งทั้งสองมีผลที่เท่ากัน

```
MOV    A, #HIGH 1234H
MOV    A, #12H
```

จากตัวอย่างคำสั่งแรกจะเป็นการนำค่าไบต์สูงของ 1234H มาใส่ในรีจีสเตอร์ A ซึ่งค่าไบต์สูงคือ 12H ทำให้คำสั่งทั้งสองมีค่าเท่ากัน

Relational Operators

ตัวดำเนินการแสดงความสัมพันธ์จะใช้กับ Operand สองตัวเปรียบเทียบกัน ซึ่งมีดังนี้

ตารางที่ 3.5 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ		ความหมาย
=	EQ	เท่ากัน
<>	NE	ไม่เท่ากัน
<	LT	น้อยกว่า
<=	LE	น้อยกว่าหรือเท่ากับ
>	GT	มากกว่า
>=	GE	มากกว่าหรือเท่ากับ

การใช้ตัวดำเนินการให้เรียกใช้อย่างใดอย่างหนึ่ง (" = " หรือ "EQ") โดยผลลัพท์ที่ได้จะเป็นจริงหรือเท็จถ้าเป็นจริงค่าในรีจีสเตอร์ A จะเป็น FFH ถ้าเป็นเท็จจะเป็น 00H พิจารณาตัวอย่างต่อไปนี้

```
MOV    A, #5 = 5
MOV    A, 5 NE 4
MOV    A, # 'X' LT 'Z'
```

ผลลัพท์ทุกคำสั่งจะมีค่าเท่ากับ

```
MOV    A, #0FFH
```

ตัวอย่างตัวดำเนินการต่าง ๆ ผลลัพท์ที่ได้จะเป็นดังนี้

การกระทำ	ผลที่ได้
'B' – 'A'	0001H
8 / 3	0002H
155 MOD 2	0001H
4*4	0010H
8 AND 7	0000H
NOT 1	FFFEH
'A' SHL 8	4100H
LOW 65535	00FFH
(8+1) * 2	0012H

ถ้าหากมีการใช้ตัวดำเนินการหลาย ๆ ตัวในนิพจน์เดียวกัน จะมีลำดับการทำงานก่อนหลังดังนี้

ตารางที่ 3.6 ลำดับการทำงานของตัวดำเนินการ

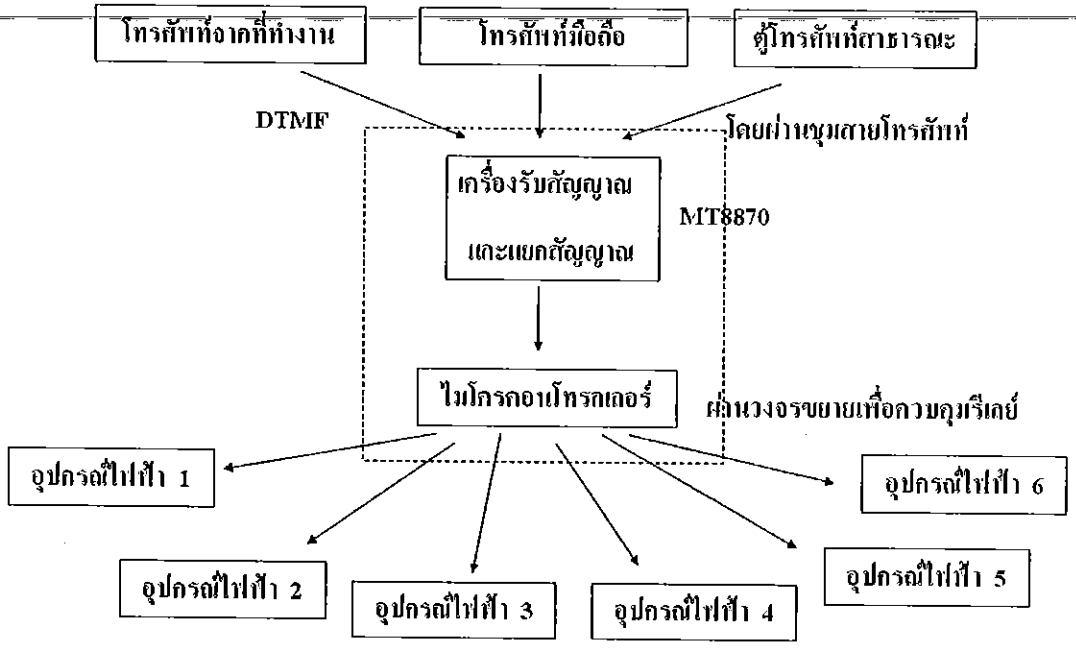
ลำดับ	ตัวดำเนินการ
1	( )
2	HIGH LOW
3	* / MOD SHL SHR
4	+ -
5	EQ NE LT LE GT GE
6	= <> < <= > >=
7	NOT
8	OR XOR

บทที่ 4

วงจรรวมและหลักการทำงาน

หลักการทำงานของโครงงานนี้สามารถอธิบายคร่าว ๆ ได้จาก BLOCK DIAGRAM นี้

ไดอะแกรมของระบบการทำงาน



รูปที่ 4.1 BLOCK DIAGRAM

ในที่นี้ได้แบ่งการทำงานของระบบออกเป็น 3 ส่วน ได้แก่

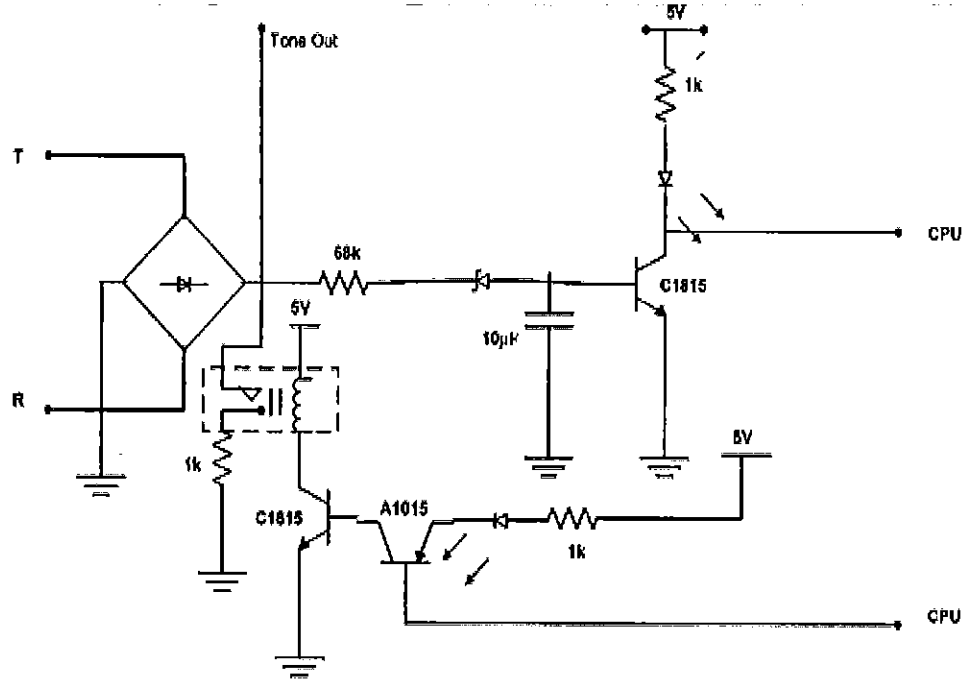
1. ส่วน Input
2. ส่วน Control processing unit
3. ส่วน Output

4.1 ส่วน อินพุท (INPUT)

4.1.1 วงจรรับโทรศัพท์อัตโนมัติ

หลักการทำงานของวงจรรับโทรศัพท์อัตโนมัติ เมื่อมีสัญญาณโทรศัพท์เข้ามาจะมีแรงดันไฟฟ้ากระแสสลับและสัญญาณไฟฟ้ากระแสตรงผสมกันมา โดยจะสนใจที่สัญญาณไฟฟ้ากระแสตรง จึงนำสัญญาณมาผ่านวงจรบริด เพื่อให้ได้สัญญาณไฟฟ้ากระแสตรงออกมาซึ่งมีระดับแรงดันอยู่ที่ 48 โวลต์ เมื่อปลายทางไม่มีโหลดหรือเปิดวงจรอยู่ แต่เมื่อสัญญาณนี้ผ่านวงจรรับโทรศัพท์จะ

มีโหลดในวงจรทำให้แรงดันไฟฟ้ากระแสตรงจะตกลงเหลือที่ 5 - 10 โวลต์ แล้วนำไปป้อนสู่ไมโครคอนโทลเลอร์ เมื่อไมโครคอนโทลเลอร์รับสัญญาณเข้ามาทาง อินพุท ก็จะแสดงผลออกทางเอาท์พุท ไปไบแอสทรานซิสเตอร์โดยใช้ทรานซิสเตอร์เป็นสวิตซ์เพื่อนำไฟฟ้าไปขับรีเลย์ให้ทำงานยกหูโทรศัพท์ แล้วถ้ามีการวางหูโทรศัพท์ที่ต้นทางไมโครคอนโทลเลอร์จะยกเลิกการแสดงผล จึงไม่มีกระแสไบแอสจึงไม่มีกระแสไฟไหลผ่านทรานซิสเตอร์ รีเลย์จะไม่ทำงานจึงเป็นการวางหูโทรศัพท์เหมือนเดิม



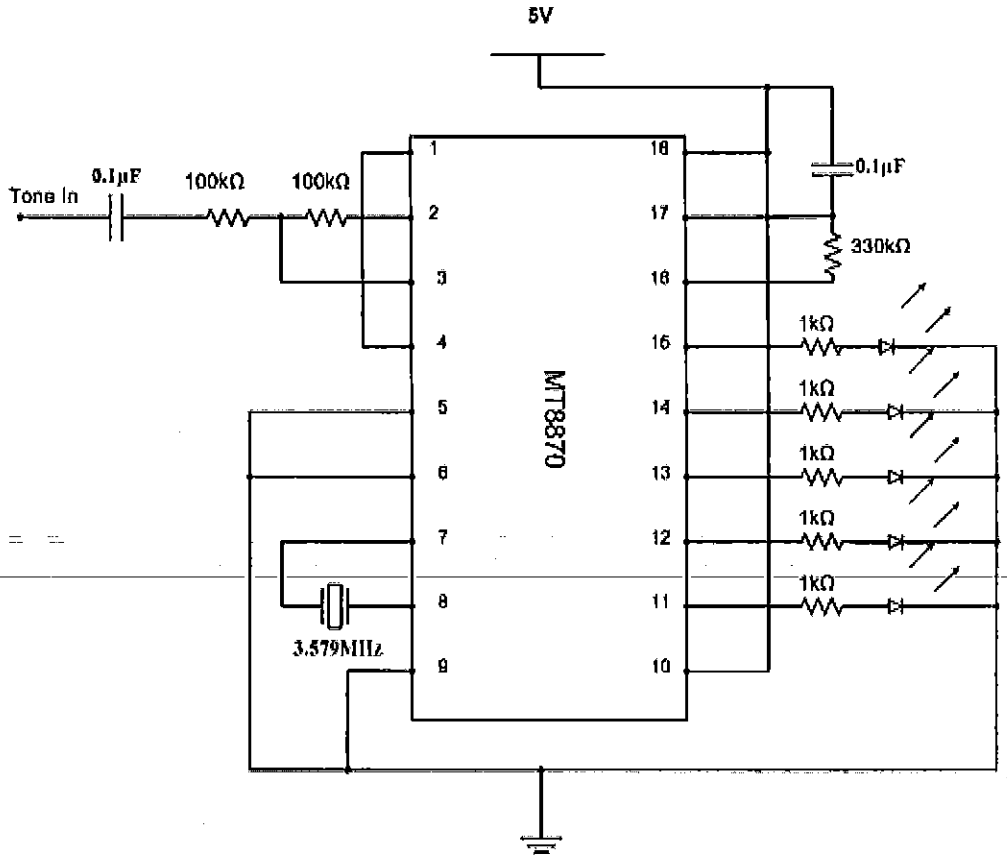
รูปที่ 4.2 วงจรรับโทรศัพท์อัตโนมัติ

4.1.2 วงจรถอดรหัสสัญญาณ DTMF

เป็นวงจรถอดรหัสสัญญาณ DTMF ซึ่งใช้ IC เบอร์ MT8870 ซึ่งเป็นไอซี ถอดรหัสสุญญาณ DTMF ที่จะแปลงเป็นเลขฐานสองขนาด 4 บิต โดยสัญญาณ DTMF ที่ต่ออยู่กับคู่สายโทรศัพท์คับปลิ้งเข้าสู่ IC เบอร์ MT8870 โดยตัวเก็บประจุขนาด 0.01 µF ทั้งสองตัว และตัวต้านทาน 100 kΩ นอกจากนี้ยังทำหน้าที่ลดแรงดันสัญญาณที่อาจทำให้เกิดการเสียหารได้ เมื่อ IC เบอร์ MT8870 ได้รับสัญญาณ DTMF แล้วก็จะทำการถอดรหัสสัญญาณออกมาเป็นข้อมูลเลขฐานสอง ขนาด 4 บิต ออกมาทาง output (D0 - D3) ส่งไปให้กับไมโครคอนโทรลเลอร์เมื่อถอดรหัสเสร็จสมบูรณ์ในแต่ละชุดสัญญาณ MT8870 จะส่งพัลส์ ลอจิก " 1 " ออกมาทางขอ STD (ขาที่ 15) จากนั้นข้อมูลเลขฐานสอง จะปรากฏที่ขา output (D0 - D3) โดยข้อมูล output ของ MT8870 เมื่อเทียบกับสัญญาณปุ่มโทรศัพท์มีรายละเอียดดังนี้

ตารางที่ 4.1 เลขฐานสอง และเลขฐานสิบ จากการกดโทรศัพท์

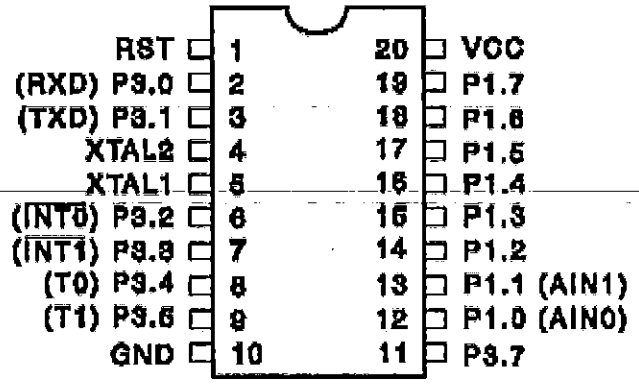
ปุ่มโทรศัพท์	ข้อมูล output เลขฐานสอง (D0 – D3)	ข้อมูล output เลขฐานสิบ
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
0	1010	10
*	1011	11
#	1100	12



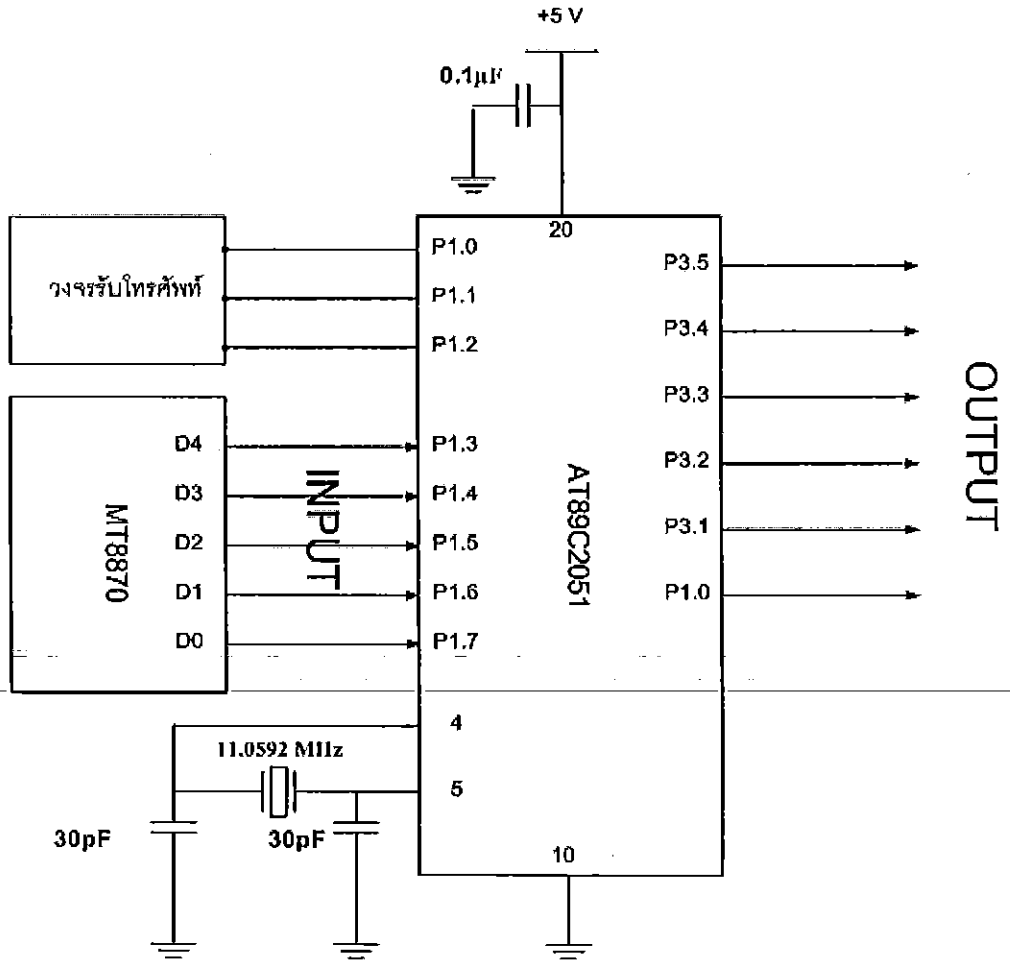
รูปที่ 4.3 วงจรถอดรหัสสัญญาณ DTMF

4.2 ส่วน CONTROL PROCESSING UNIT

ในส่วนนี้เราได้ใช้ชิฟไมโครคอนโทรลเลอร์ AT89C2051 ในการ control การทำงานต่าง ๆ ซึ่งการใช้ขา input และ output ดังตัวอย่างรูปที่ 4.4



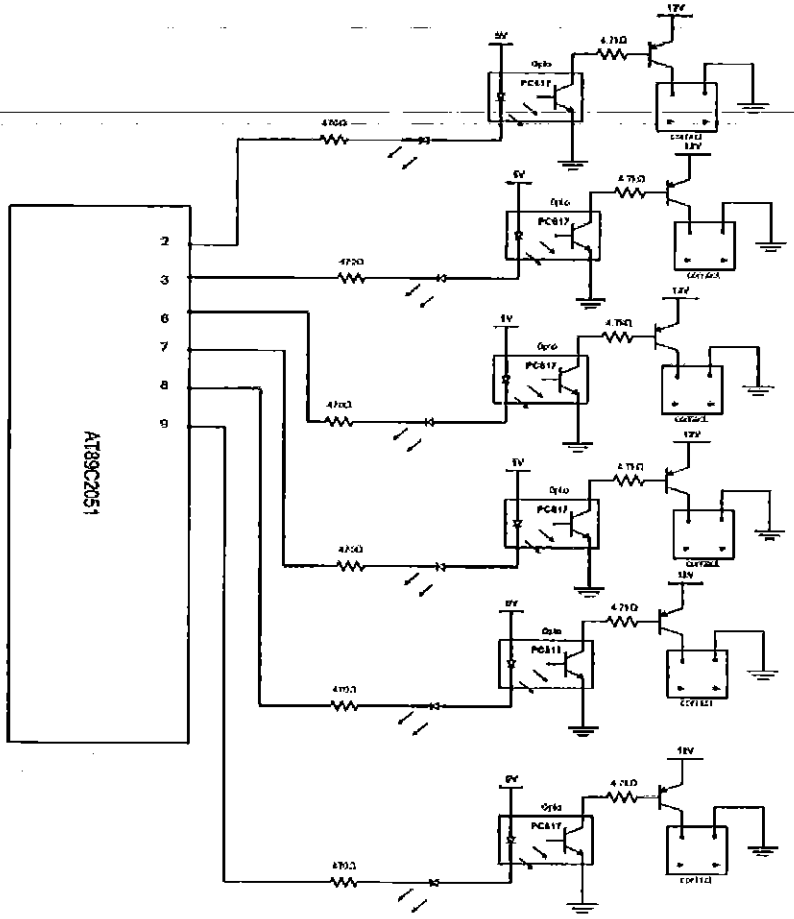
รูปที่ 4.4 แสดงขาต่าง ๆ ของ AT89C2051



รูปที่ 4.5 วงจรการเชื่อมต่อระหว่าง AT89C2051 กับอุปกรณ์อื่น ๆ

4.3 ส่วนเอ้าท์พุท (OUTPUT)

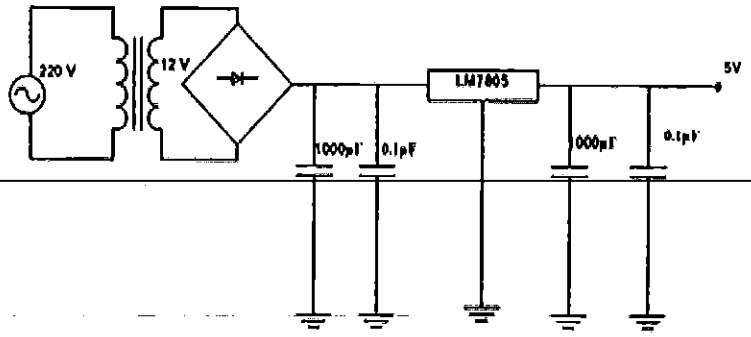
วงจรทางด้าน output จะเป็นส่วนที่รับสัญญาณจากไมโครคอนโทรลเลอร์มาเพื่อใช้ในการควบคุมอุปกรณ์ไฟฟ้า โดยการต่อกับอุปกรณ์ภายนอก โดยวงจรได้ทำเป็นโซลิดสเตทรีเลย์ 6 ช่องทางโดยใช้ออปโต้คอยควบคุมให้รีเลย์เปิด-ปิดสวิตซ์เพื่อนำไปควบคุมอุปกรณ์ไฟฟ้าภายนอกที่ต้องการต่อไป



รูปที่ 4.6 วงจรการทำงานของรีเลย์

4.4 วงจรภาคจ่ายไฟ

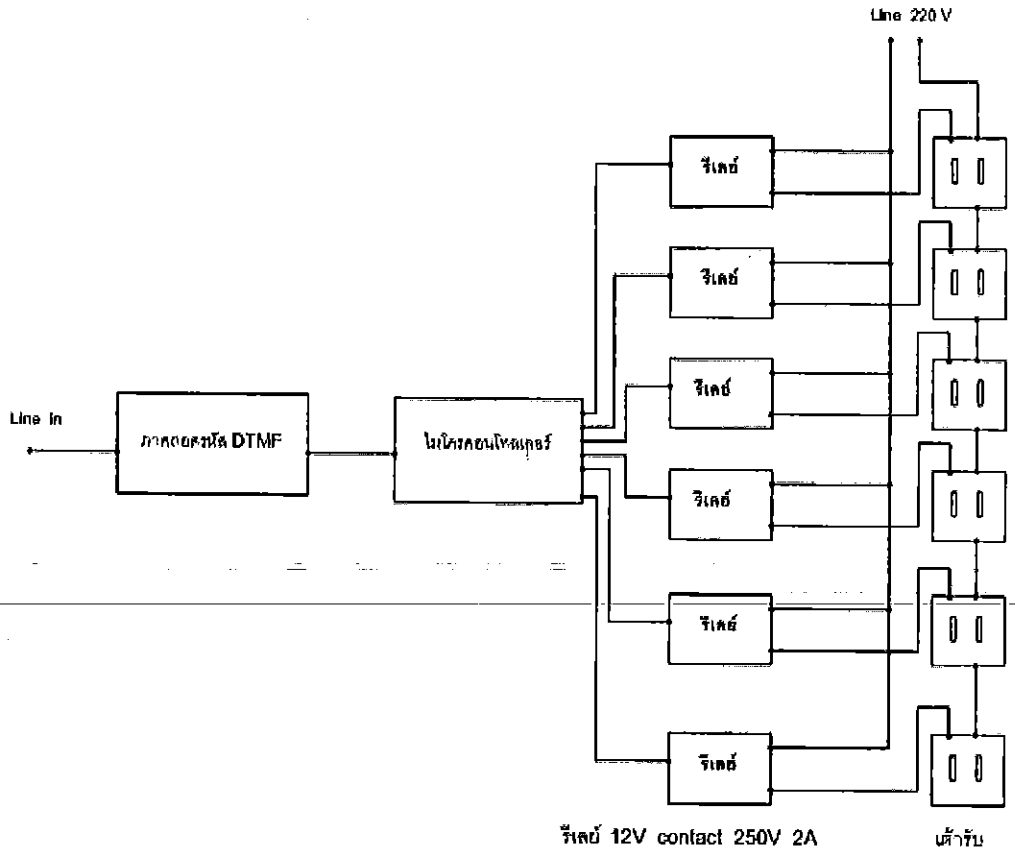
ในวงจรรวมทั้งหมดเราต้องการใช้ไฟฟ้ากระแสตรงที่มีแรงดัน 5 โวลต์ และ 12 โวลต์ จึงได้ออกแบบวงจรของภาคจ่ายไฟดังรูป 4.7 โดยเริ่มจากไฟฟ้ากระแสสลับ 220 โวลต์ ผ่านหม้อแปลงไฟฟ้า เหลือแรงดันกระแสสลับ 12 โวลต์ แล้วนำไปผ่านวงจรบริดไดโอด เพื่อแปลงเป็นไฟฟ้ากระแสตรง แล้วมีตัวเก็บประจุต่อคร่อมเพื่อเพิ่มเสถียรภาพของแรงดันไฟฟ้า ตอนนี้เราจะได้ไฟฟ้ากระแสตรง 12 โวลต์ แล้วเรายังต้องการไฟฟ้ากระแสตรงที่มีแรงดันไฟฟ้า 5 โวลต์ โดยผ่าน IC regulator เบอร์ 7805 เพื่อแปลงไฟฟ้า 12 VDC เป็น 5VDC เพื่อนำไปใช้เป็นแหล่งจ่ายต่อไป



รูปที่ 4.7 วงจรภาคจ่ายไฟ

4.5 วงจรรวมและหลักการทำงาน

เมื่อเราส่งงานปิด เปิด สวิตซ์ทางโทรศัพท์ จะมีสัญญาณเข้ามาที่ภาคถอดรหัสส่วนนี้จะทำการแปลง รหัส DTMF เป็นรหัสดิจิตอล 4 บิต จากนั้นจะส่งให้ภาคไมโครคอนโทลเลอร์ตัดสินใจว่าจะไปสั่งงานรีเลย์ตัวไหนแล้วรีเลย์จะทำหน้าที่ตัดต่อไฟจากแหล่งจ่าย 220 โวลต์ให้กับอุปกรณ์เครื่องใช้ไฟฟ้า



รูปที่ 4.8 วงจรรวม

บทที่ 5

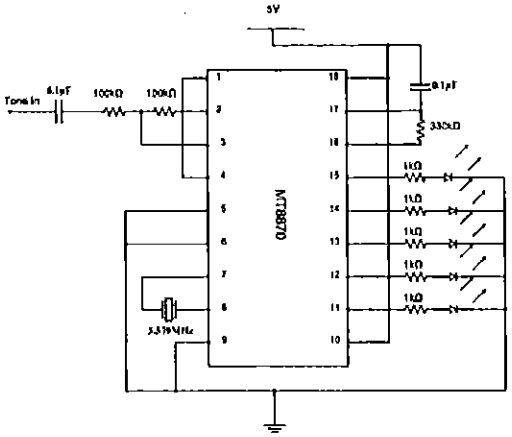
การทดลองการทำงาน

5.1 การทดสอบวงจรตอบรับโทรศัพท์อัตโนมัติ

หลักการทำงานของวงจรรับโทรศัพท์อัตโนมัติ เมื่อมีสัญญาณโทรศัพท์เข้ามาจะมีแรงดันไฟฟ้ากระแสสลับและสัญญาณไฟฟ้ากระแสตรงผสมกันมาโดยเราจะสนใจที่สัญญาณไฟฟ้ากระแสตรงเราจึงนำสัญญาณมาผ่านวงจรบริด เพื่อให้ได้สัญญาณไฟฟ้ากระแสตรงออกมาซึ่งมีระดับแรงดันอยู่ที่ 48 โวลต์ เมื่อปลายทางไม่มีโหลดหรือเปิดวงจรอยู่ แต่เมื่อสัญญาณนี้ผ่านวงจรรับโทรศัพท์จะมีโหลดในวงจรทำให้แรงดันไฟฟ้ากระแสตรงจะตกลงเหลือที่ 5 - 10 โวลต์

5.2 การทดสอบวงจรถอดรหัสสัญญาณ DTMF เป็น Binary Code

ทำการต่อวงจรตามรูปเข้ากับคู่สายโทรศัพท์ แล้วต่อ Output ทั้ง 4 วงจร (D0-D3) และ STD ของ IC เบอร์ MT8870DE เข้ากับหลอด LED ดังรูป โดยการทดลองจะต่อสายโทรศัพท์เข้ากับอุปกรณ์ทดลอง



รูปที่ 5.1 การต่อ output เข้ากับหลอด LED

ในการทดลองโดยค่าที่ได้จากวงจรคิดเป็นเลขฐานสอง คือถ้า LED ติดให้ Logic เป็น "1" แต่ถ้า LED ดับให้ Logic เป็น "0" จะกำหนดให้ขาที่ 11 เป็น D0 ขาที่ 12 เป็น D1 ขาที่ 13 เป็น D2 ขาที่ 14 เป็น D3 และขาที่ 15 เป็นขา STD เมื่อสังเกตที่ขา STD จะติดเมื่อมีการกดปุ่มโทรศัพท์ โดยผลการทดลองที่ได้จะเป็นดังแสดงในตารางที่ 5.1

ตารางที่ 5.1 เลขฐานสอง และเลขฐานสิบ จากการกดโทรศัพท์

ปุ่มโทรศัพท์	ข้อมูล output เลขฐานสอง (D0 – D3)	ข้อมูล output เลขฐานสิบ
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
0	1010	10
*	1011	11
#	1100	12

5.3 การทดลองเกี่ยวกับวงจรจ่ายไฟ

ในการทดลองต่อเกี่ยวกับภาคตัดไฟเพื่อให้ได้ค่าเอาต์พุทที่ 5.10V ซึ่งค่าที่ได้มีค่าใกล้เคียงกับแรงดันที่ต้องการซึ่งสาเหตุที่ได้แรงดันไม่ตรงกับค่าที่เราต้องการ คือ อาจจะมาจากตัวของไอซีเองหรืออาจจะเป็นเกี่ยวกับสภาพแวดล้อมโดยใช้ไอซีเรกูเลตในการรักษาให้มีค่าคงที่

บทที่ 6
สรุปและข้อเสนอแนะ

6.1 สรุปผลการดำเนินการ

จากการที่ได้ศึกษาโครงงานนี้ ได้ทำการศึกษาอยู่ 2 เรื่องใหญ่ ๆ คือ

1. โทรศัพท์ ได้ศึกษาและทำการออกแบบวงจรที่เกี่ยวกับกระบวนการควบคุมทางโทรศัพท์โดยประกอบด้วยวงจรตรวจจับสัญญาณกระดิ่ง วงจรยกหูและวางโทรศัพท์ วงจรถอดรหัสแบบ DTMF

2. ไมโครคอนโทรลเลอร์ MCS-51 AT89C2051 ได้ศึกษาสถาปัตยกรรมของวงจรไมโครคอนโทรลเลอร์ AT89C2051 และหลักการทำงานต่าง ๆ โดยใช้ภาษาแอสเซมบลีในการเขียนคำสั่งควบคุมการทำงาน

ในการทำโครงงานโทรศัพท์ควบคุมอุปกรณ์ไฟฟ้านี้เราได้นำสัญญาณที่บอกสถานการณ์ทำงานต่าง ๆของโทรศัพท์มาประยุกต์ใช้ในการควบคุมอุปกรณ์ไฟฟ้า โดยสัญญาณที่ใช้ ได้แก่ สัญญาณความถี่ (ความถี่สูงผสมความถี่ต่ำ) จากการกดปุ่มโทรศัพท์ โดยในการทำโครงงานโทรศัพท์ควบคุมอุปกรณ์ไฟฟ้านี้ได้นำสัญญาณความถี่มาแปลงเป็น Binary Code ด้วย IC เบอร์ MT8870 แล้วนำค่า Binary Code ไปประมวลผลในไมโครคอนโทรลเลอร์ MCS-51 AT89C2051 ซึ่งเป็นการควบคุมให้สวิตซ์ไฟฟ้า ON, OFF ตามการกดหมายเลขโทรศัพท์

6.2 สรุปผลการทดลอง

จากการทดลองการสั่งอุปกรณ์ไฟฟ้าทางโทรศัพท์ระยะไกล ได้ผลเป็นที่น่าพอใจ สามารถตอบสนองคำสั่งที่สั่งออกไปได้อย่างถูกต้อง

- สามารถเปิดอุปกรณ์ทีละอย่างได้
- สามารถสั่งปิดอุปกรณ์ทีละอย่างได้

6.3 ปัญหาที่พบในการทดลอง

- เมื่อเราสั่งงานทางโทรศัพท์ระยะไกล เราไม่สามารถทราบได้ว่าอุปกรณ์ไฟฟ้าที่เราสั่ง ปิด – เปิด นั้นทำงานได้ตามที่เราสั่งหรือไม่ นั่นก็คืออุปกรณ์ตัวนี้ไม่มีระบบตรวจสอบเพื่อจะส่งผลลัพธ์ไปยังต้นทางได้

- เมื่อเราคิดไปถึงขั้นที่ว่าถ้ามีคนที่ไม่ประสงค์ดี ต้องการแกล้งโดยแอบมาสั่งการ ปิด – เปิด อุปกรณ์ไฟฟ้า ภายในบ้านเรา ซึ่งสามารถทำได้โดยง่ายเพียงแค่รู้เบอร์โทรศัพท์บ้าน และ รู้วิธีการสั่งงาน

6.4 ข้อเสนอแนะ

1. อยากให้มีการพัฒนาอุปกรณ์นี้ต่อไปโดยให้มีระบบตรวจสอบการทำงานของอุปกรณ์ไฟฟ้าแล้วแจ้งไปที่ต้นทาง เพื่อให้เกิดความมั่นใจว่าอุปกรณ์นั้น ๆ ทำงานตามคำสั่งแล้วจริง

2. อยากให้พัฒนาอุปกรณ์นี้ต่อไปโดยให้มีระบบใส่พาสเวิร์ด เพื่อป้องกันผู้อื่นแอบมาใช้งานอุปกรณ์

หนังสืออ้างอิง

[1] ชัยวัฒน์ ลิ้มพรจิตรวิไล .**คู่มือนักอิเล็คทรอนิก.** กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2538

[2] ธีรวัฒน์ ประกอบผล .**การประยุกต์ใช้งานไมโครคอนโทรลเลอร์.** กรุงเทพมหานคร : สำนักพิมพ์ สสท. 2543

[3] ธนัท ชัยยุทธ และ กณพ แก้วพิชัย .**ดิจิตอลพื้นฐาน.** กรุงเทพมหานคร :ซีเอ็ดยูเคชั่น, 2540

ภาคผนวก

โปรแกรมควบคุมการทำงานไมโครคอนโทรลเลอร์

```
  DTMF DECODER
;     8 RELAY OUTPUT
;     DTMF6.PCB
;

SPAC:     DS        36
RING:     DS        1
HOLD:     DS        1
BUFF:     DS        1
DBUF:     DS        1
CT00:     DS        1
CT01:     DS        1

KLOC      BIT       00H
HLOC      BIT       01H

          ORG       0000H
          LJMP      INIT

          ORG       000BH
          LJMP      TIME

          ORG       0050H
INIT:     MOV       P0,#0FFH        ; Initial Port
          MOV       P1,#0FFH
          MOV       P2,#0FFH
          MOV       P3,#0FFH

          MOV       R0,#00
          MOV       R1,#00
          MOV       R2,#00
          MOV       R3,#00
          MOV       R4,#00
          MOV       R5,#00
          MOV       R6,#00
          MOV       R7,#00

          CLR       KLOC
          SETB      HLOC

          MOV       CT00,#40
          MOV       CT01,#100
          MOV       HOLD,#30
          MOV       RING,#08
```

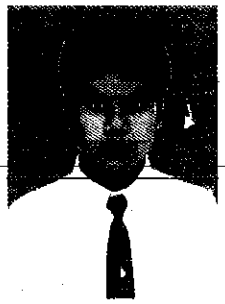
```
        MOV     TMOD,#22H       ; Timer 0-1 8 Bit Auto Reload
        MOV     TH0,#256-231    ; 1.000.000
        MOV     TL0,#256-231    ;  20.000
        CLR     TR0             ;  2.550
        SETB    IE.1            ;
        SETB    IE.7            ; Enable INT

        MOV     SCON,#50H       ; Serial Mode 1
        MOV     TH1,#0FDH       ; 9600 Bps
        MOV     TL1,#0FDH
        SETB    TR1             ; Start Timer
        CLR     RI              ; Clear Receive Bit
        CLR     TI              ; Clear Transmitt Bit

        LCALL   HHLY
        LCALL   HHLY
        LCALL   HHLY
        SETB    TR0

MAIN:   JNB     TR0,HANG

        JB      P1.0,MAIN       ; Ring In
        LCALL   DLAY

        SETB    TR0             ; Start Timer

        JNB     P1.0,$          ; Ring Hold
        LCALL   DLAY

        DJNZ    RING,MAIN       ; Ring Count
        CLR     P1.2            ; On Hook
        LJMP    MN10

HANG:   SETB    P1.2            ; Off Hook
        LCALL   HHLY
        LCALL   HHLY
        LCALL   HHLY

        JB      P1.0,$          ; Ring In
        LJMP    MAIN

MN10:   JNB     TR0,HANG

        JNB     P1.3,MN10       ; Data In
        CLR     P1.0            ; LED
        LCALL   HLAY
        JB      P1.3,$
        SETB    P1.0
```

```
                MOV     A,BUFF
                MOV     FBUF,A

                MOV     A,P1            ; Read Data
                ANL     A,#11110000B
                SWAP    A
-------------------------------------------------------------------------------
MN20:           CJNE    A,#08,MN21      ; 1
                MOV     BUFF,#'1'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
MN21:           CJNE    A,#04,MN22      ; 2
                MOV     BUFF,#'2'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
MN22:           CJNE    A,#12,MN23      ; 3
                MOV     BUFF,#'3'

MN23:           CJNE    A,#02,MN24      ; 4
                MOV     BUFF,#'4'

MN24:           CJNE    A,#10,MN25      ; 5
                MOV     BUFF,#'5'

MN25:           CJNE    A,#06,MN26      ; 6
                MOV     SBUF,#'6'

MN26:           CJNE    A,#14,MN27      ; 7
                MOV     BUFF,#'7'

MN27:           CJNE    A,#01,MN28      ; 8
                MOV     BUFF,#'8'

MN28:           CJNE    A,#09,MN29      ; 9
                MOV     BUFF,#'9'

MN29:           CJNE    A,#05,MN2A      ; 0
                MOV     BUFF,#'0'

MN2A:           CJNE    A,#13,MN2B      ; *
                MOV     BUFF,#'*'
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
MN2B:           CJNE    A,#03,MN2C      ; #
                MOV     BUFF,#'#'

MN2C:           LCALL   CONT

MN30:           LJMP    MN10            ; End Main
```

```
;---> Control ----------------------------------------------------

CONT:   MOV     A,DBUF

CT30:   CJNE    A,#'*',CT40     ; Control On
        MOV     A,DBUF

CT31:   CJNE    A,#'1',CT32     ; CH 1
        CLR     P3.5
CT32:   CJNE    A,#'2',CT33     ; CH 2
        CLR     P3.4
CT33:   CJNE    A,#'3',CT34     ; CH 3
        CLR     P3.3
CT34:   CJNE    A,#'4',CT35     ; CH 4
        CLR     P3.2
CT35:   CJNE    A,#'5',CT36     ; CH 5
        CLR     P3.1
CT36:   CJNE    A,#'6',CT37     ; CH 6
        CLR     P3.0
CT37:   NOP
        LJMP    ECON

CT40:   CJNE    A,#'#',CT50     ; Control Off
        MOV     A,DBUF

CT41:   CJNE    A,#'1',CT42     ; CH 1
        SETB    P3.5
CT42:   CJNE    A,#'2',CT43     ; CH 2
        SETB    P3.4
CT43:   CJNE    A,#'3',CT44     ; CH 3
        SETB    P3.3
CT44:   CJNE    A,#'4',CT45     ; CH 4
        SETB    P3.2
CT45:   CJNE    A,#'5',CT46     ; CH 5
        SETB    P3.1
CT46:   CJNE    A,#'6',CT47     ; CH 6
        SETB    P3.0
CT47:   NOP
        LJMP    ECON

CT50:   NOP
ECON:   RET
```

```
;---> Timer 0 Interrupt ----------------------------------------

TIME:   DJNZ    CT00,ETIM       ; 1 Sec
        MOV     CT00,#40
        DJNZ    CT01,ETIM
        MOV     CT01,#100

        DJNZ    HOLD,ETIM

ETIM:   RETI

DLAY:   MOV     R7,#50          ; Delay
        DJNZ    R7,$
        RET

HLAY:   MOV     R6,#100
HL01:   MOV     R7,#250
        DJNZ    R7,$
        DJNZ    R6,HL01
        RET

HHLY:   MOV     R5,#06
HH01:   MOV     R6,#00
HH02:   MOV     R7,#00
        DJNZ    R7,$
        DJNZ    R6,HH02
        DJNZ    R5,HH01
        RET

        END
```

ประวัติผู้เขียนโครงงาน



ชื่อ นายจักรกฤษ เพ่งภู่

ภูมิลำเนา 25 หมู่ 5 ต. บ้านใหม่คลองเดียน อ. บ้านไร่ จ. อุทัยธานี 61180

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนบ้านทุ่งนาวิทยา
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : toneeen@hotmail.com

ชื่อ นายชัยณรงค์ วัฒนาพงศ์สกุล

ภูมิลำเนา 243 หมู่ 5 ต. บ้านปิน อ. ดอกคำใต้ จ. พะเยา 56120

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนถ้ำปินวิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : watnaphongsakul@hotmail.com